SymSpice

User Guide

Product version 2.xx
October 2012

Table of Contents

Part I

SymSpice manual

Introduction

Overview

SymSpice is a "golden" SPICE standard simulation program.

Symspice Turbo is a fast-SPICE simulation program. SymSpice Turbo implements modified common used "golden" SPICE approaches for circuit simulation, that allows to preserve simulation accuracy while simulation speed increases.

The both SymSpice and SymSpice Turbo are combined in one executable module "symspice" but they need to different licenses.

While SymSpice suits to simulate small, medium and large scale integrated circuits, SymSpice Turbo intended to simulate large and very large scale integrated circuits represented at transistor level.

SymSpice Turbo has its best results in time latency circuits, particularly in large analog-to-digital circuits, containing up to 1M elements. Such circuits are usually simulated faster than the industry average level (5-10 times faster for typical analog circuits and up to 1000 times faster for special types of circuits).

SymSpice and SymSpice Turbo are compatible with the most popular SPICE netlists and foundry-supplied device models.

Website: http://www.symica.com

System requirements

Verified Operating Systems

1. Windows

   32bit: Windows 7, Windows Vista, Windows XP

   64bit: Windows 7, Windows Vista, Windows XP

2. Linux

   32bit: OpenSuSE 12.x, Ubuntu 11.10, Debian 6.x, CentOS/RHEL 4.x, 5.x, 6.x, Fedora 16

   64bit (with 32bit system libraries installed): CentOS/RHEL 5.x, 6.x, Fedora 14

Using multithreading

SymSpice and SymSpice Turbo are specially designed to take advantage of the multiprocessing system. To speed up simulation SymSpice use specially designed parallel numerical methods and concurrent model evaluation. SymSpice Turbo additionally use the unique technology called dynamical decomposition which allows partitioning the full matrix of equations into independent sub-matrix blocks at certain time interval. The dynamically separated sub-matrix blocks can be classified as active or latent and be simulated independently, so that dynamical decomposition allows to calculate each active sub-matrix more effectively.

By using multithreading, you can speed-up simulations with no loss of accuracy according to the accuracy options to be specified. However, multithreading may lead to slightly different results from run to run. This peculiarity is quite normal for parallel numerical methods used in SPICE programs. Turn off multithreading if you want to get consistent result.

The acceleration rate depends on circuit and it's typical values are:
2-core system ~ 1.5 times
4-core system ~ 2.5 times.

To use certain number of threads during simulation you have to specify variable 'multithread' in 'spice.ini' file. The default value of this parameter is 128, that makes SymSpice to activate all available threads automatically.

Hint

Typically Hyper-Threading architecture makes simulation faster, but for some computational systems it might result to slower simulation.

Analog fast-SPICE

SymSpice Turbo uses three fast-SPICE methods to speed up simulation:
- dynamic decomposition;
- multi-rate;
- multi-threading.

SymSpice implements a set of original methods allowing to classify and to separate large system of algebraic equations into a set of independent subsystems (Figure. 1), which are generated dynamically at each integration step (time interval), that allows:
- to simulate circuit active parts only;
- to apply particular mathematical algorithms to the each subsystem;
- to use multithread calculations naturally.

Hint

Using dynamic decomposition keeps Gold SPICE accuracy and simulation speed should be higher for the major sorts of circuits, so that it is recommended to use this method for all circuits by default. However, for a quite small fully active circuits these methods might do some additional job and may slightly increase simulation time.

You may switch on/off the dynamic decomposition by setting option 'fast_spice' (1 - switches fast spice methods on, 0 - switches them off).

'fast_spice' option can be set in:
1. netlist to use for each particular circuit (e.g., 'opt_name options fast_spice=1');
2. 'spice.ini' file to use for all simulated circuits by default.

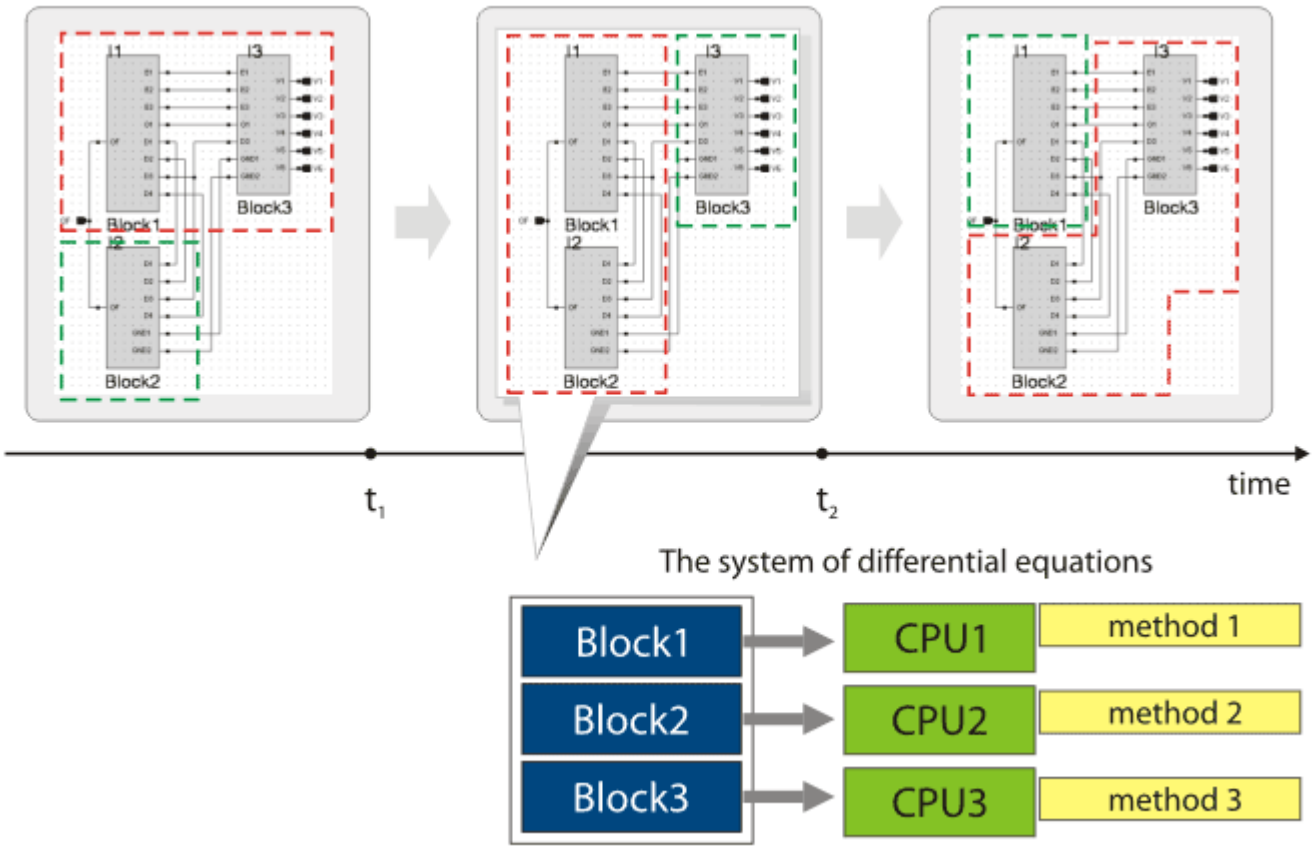


Figure 1. Dynamic decomposition.

Running SymSpice

Command Line

Syntax

symspice --help

symspice --version

symspice [options] [-i] [path] <input_file> [[-o] [path] <output_file>]

Options

-v, --version	Displays the version number, build date and copyrights of SymSpice.
-h, --help	Outputs message of usage SymSpice.
-i <input_file>	Specifies the input netlist file name.
-o <output_file>	Specifies the output file name or directory. In case a directory <output_file> exists SymSpice uses <output_file> as output directory otherwise <output_file> is considered as output file name. If no extension for output file name is given, .lis is assigned.
-format <fmt> [<fmt2>..<fmtN>]	Produces raw data in the format <fmt>. Possible values for <fmt> are 'apb', 'csdf', 'csv', 'nutascii', 'nutbin', 'none'.
-probe	Runs SymProbe to visualize simulation result in runtime.

Running SymProbe with SymSpice

To visualize simulation results during runtime you can run SymProbe simultaneously with SymSpice, using command line option '-probe'.
At first run SymProbe shows output traces pointed in the netlist statement 'save'. You can adjust your special view: traces, colors, axis scaling etc, and save these settings in file "view.ppw" by clicking "Save view to disk". This file will be placed in the same folder with simulation results. When SymProbe starts up, the file "view.ppw" is being looked at and the last session is restored.

SymSpice Input/Output formats

SymSpice accepts the most common SPICE netlist formats. The names and extensions of output files depend on simulation mode. See chapter "Mixed SPICE format" for more details.

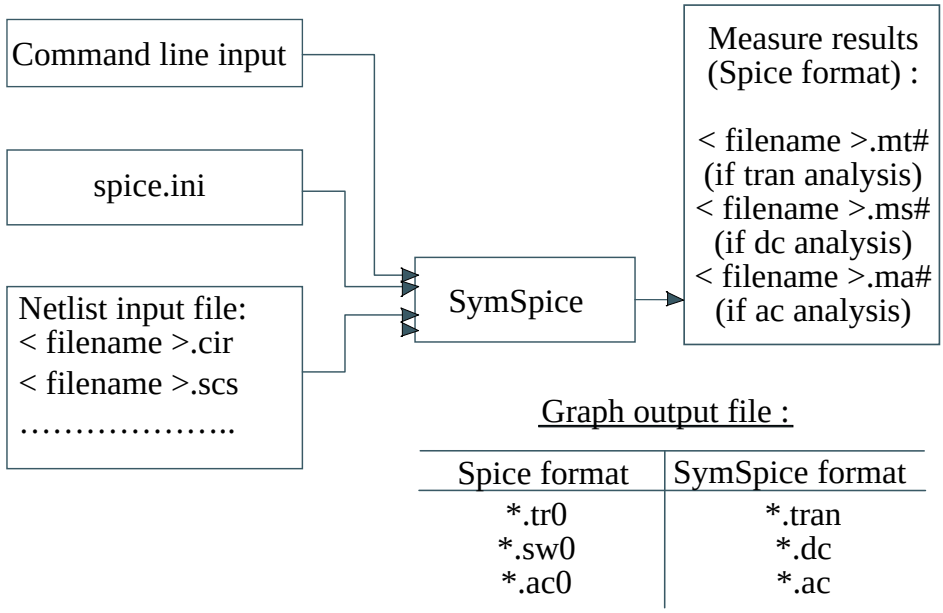


Figure 2. Overview of SymSpice data flow

SymSpice output formats

*.apb	file with results of TRAN analysis in APB format (BIN)
*.tr0	file with results of TRAN analysis in CSDF format (ASCII) if input file is in 'spice' mode;
*.sw0	file with results of DC analysis in CSDF format (ASCII) if input file is in 'spice' mode;
*.ac0	file with results of AC analysis in CSDF format (ASCII) if input file is in 'spice' mode;
*.tran	file with results of TRAN analysis in APB format (BIN);
*.dc	file with results of DC analysis in APB format (BIN);
*.ac	file with results of AC analysis in APB format (BIN);
*.csd	file with results of TRAN, DC, AC analysis in CSDF format (BIN\ASCII);

*.csv	file with results of TRAN, DC, AC analysis in tabulated ASCII format (raw-format), it can be viewed with many freeware visualization programs, such as gwave, etc;
*.raw	file with results of TRAN, DC, AC analysis in NUTMEG format (BIN/ASCII);
*.lis	listing file containing information about simulation process;
runObjFile, logFile	supplemental files for viewing simulation results in some result browsers.

Configuration file spice.ini

Configuration file 'spice.ini' is used to preset simulation options that effect all the following runnings of SymSpice. When SymSpice starts up the file 'spice.ini' is being looked consequently in the following directories:

Linux
- <current directory>/
- <home>/.config/symica/
- /etc/symica/
- <instdir>/share/symica/

Windows
- <current directory>/
- <home>/Application Data/symica/
- <instdir>/share/

Configuration file can contain any netlist statements and is included in front of the top netlist file. However 'spice.ini' file does have an effect on the determine of the base netlist format and netlist mode (see chapter "Mixed SPICE format").

Example

```
**********************************
 .options multithread=on nthreads=128
 .options fast_spice=1
**********************************
```

Simulation Log File

During simulation log file is created including the following info:

- errors and warning during parsing input netlist and included files;
- short circuit description (number of elements, circuit nodes, etc);
- models info;
- type of analysis and simulation process indicator;
- time spent on simulation.

Input netlist

Mixed SPICE format

SymSpice accepts the most common SPICE netlist formats, which can be mixed in one netlist. Symica Design Environment generates SymSpice specific file format to run SymSpice.

For every top netlist SymSpice determines a **base netlist format** depending on which simulator may change its behaviour, e.g names of the output files.

To determine the base netlist format SymSpice uses the following rules:
1. Name of the top netlist file extension.
2. Type of the first meaningful statement in the project root file.

Based on these rules simulator also determines **netlist mode** to parse its content. There are three possible netlist modes:
- 'local' mode switches on SymSpice netlist format compatibility.
- 'specre' mode switches on Cadence® Spectre® netlist format compatibility.
- 'spice' mode switches on Synopsys® HSPICE® netlist format compatibility.

To explicitly switch between different netlist formats use statement :

simulator lang={local | spectre | spice}.

Note:
Every model in the netlist has its own attribute in which mode it was specified. The behaviour of the model, such as a set of formulas and default parameters, will depend on this attribute.

Case sensitivity

SymSpice and Spectre netlist formats ('local' and 'spectre' modes) are case-sensitive by default. Spice netlist format ('spice' mode) is case insensitive by default. To switch on/off case sensitivity use statement *'simulator insensitive=yes/no'*. This option affects all netlist content after the position where it is defined and can be used together with other *'simulator'* options.

Syntax

'local', 'spectre' and 'spice' modes

simulator insensitive={yes | no}

simulator lang={local | spectre | spice}' insensitive={yes | no}

Note: If parameter *insensitiveis set up to yes* all netlist content is converted to lowercase.

Note: Spice netlist being included from the netlist in 'local' or 'spectre' mode becomes case-sensitive.

There are two special options to switch on/off case sensitivity for the current and all included files .

Syntax

'local', 'spectre' and 'spice' modes

**@ spectre_insensitive={yes | no}*

**@ spice_insensitive={yes | no}*

Example

simulator lang={local | spectre} insensitive=no
subckt **teST1** **-** *is not converted to lowercase*
ends

simulator lang={local | spectre} insensitive=yes
subckt **teST2** *- all content is converted to lowercase*
ends

simulator lang={spice} insensitive=yes

subckt **teST3** - *all content is converted to lowercase*
ends

simulator lang={spice} insensitive=no
subckt **teST4** **-** *is not converted to lowercase*
ends

simulator lang={local | spectre | spice} insensitive=no
x 1_n **test1** **-** *incorrect*
x 1_n **teST1** **-** *correct*

x 1_n **test2** **-** *correct*
x 1_n **teST2** **-** *incorrect*

x 1_n **test3** **-** *correct*
x 1_n **teST3** **-** *incorrect*

x 1_n **test4** **-** *incorrect*
x 1_n **teST4** **-** *correct*
...

simulator lang={local | spectre | spice} insensitive=yes
x 1_n **test1** **-** *incorrect*
x 1_n **teST1** **-** *incorrect*

x 1_n **test2** **-** *correct*
x 1_n **teST2** **-** *correct*

x 1_n **test3** **-** *correct*
x 1_n **teST3** **-** *correct*

x 1_n **test4** **-** *incorrect*
x 1_n **teST4** **-** *incorrect*
...

Ground node

In 'local' and 'spectre' modes
The node named '0' is treated as a ground.
For Cadence® Spectre® netlist compatibility the first node mentioned in GLOBAL statement, which must be the first statement of the top-level netlist, is also treated as a ground.

In 'spice' mode
All nodes '0', 'gnd', 'gnd!', 'ground' (in upper and lower case) are treated as a ground.

Instance statements

Instance statements describe the device and source components. Each instance statement consist of:

- name of the component;
- nodes to which the component is connected;
- model of the component;
- parameter values.

Syntax

'local' and 'spectre' modes

Name <(> node1 ... nodeN <)> <eltype> << pname1= val1> ...< pnameN= valN>>

Name <(> node1 ... nodeN <)> <mname> << pname1= val1> ...< pnameN= valN>>

'spice' mode

Name <node1 node2 ... nodeN> <mname> <pname1=val1>

+<pname2=val2> <M=val>

Name <node1 node2 ... nodeN> <mname> <val1 val2 ... valn>

Parameters

Parameter name	Description
Name	In 'local', 'spice' and 'spectre' mode you can use any name. Though in classical SPICE netlist name must begin with a specific letter for each component type: R Resistor C Capacitor L Inductor D Diode Q BJT M MOSFET I Current source V Voltage source X Subcircuit call

Parameter name	Description
node1 ... nodeN	Node names are identifiers of the nodes to which the component is connected. Node names can begin with a number or with an identifier which can be alphabetic character or any of the following characters: # _ ! %.
mname/eltype	Model reference name is required for all components except passive devices. In 'local' and 'spectre' modes if You don't use model reference name, You must declare an eltype as: resistor Resistor capacitor Capacitor inductor Inductor diode Diode bjt BJT mos3 MOSFET level=3 bsim3v3 MOSFET level=49 isource Current source vsource Voltage source subckt Subcircuit call
pname1 ... pnameN	Component parameter name used to identify the parameter value that follows this name.
val1 ... valN	Value assigned to the parameter pname. The value can be a number or an algebraic expression.
M	The multiply parameter M is a special keyword common to all components (except for voltage sources). It multiplies the internal component values to give the effect of making parallel copies of the component.

Example 1

```
q_pnp n1 n2 n3 gnd! vpnp
```

In the example above collector of bipolar junction transistor q_pnp is connected to node n1, base is connected to node n2, emitter is connected to node n3, substrate is connected to gnd! node. The transistor parameters are described in the model statement referenced by the name vpnp.

Example 2

```
m_nmos n1 n2 n3 gnd! nmos1 l=2.4e-7 w=6e-7
```

The previous example specifies a MOSFET called m_nmos, whose drain, gate, source, and substrate nodes are named n1, n2, n3 and gnd!, respectively. The element statement calls an associated model statement, nmos1. MOSFET channel length=2.4e-7 and channel width=6e-7.

Example 3

```
R12 1 2 3k
```

The previous example specifies a resistor r12 connected from node 1 to node 2 with a resistance of 3 kilo Ohms.

Example 4

```
t3_t1_1 (vd n3 n5) bjt area=1 m=1
```

Collector of bipolar junction transistor t3 is connected to node vd, base is connected to node n3, emitter is connected to node n5.

Example 5

```
r1 (1 2) res1 l=10u w=1u
```

Resistor r1 is connected to nodes 1 and 2. Resistor length=10u, resistor width=1u. Other parameters are specified in the model statement referenced by the name res1.

Comments

SymSpice uses the following characters as a comment:

'local' and 'spectre' modes

A double slash (//) and an asterisk (*) indicate a comment line:

* *<comment until end of line>*

// <comment until end of line>

< SymSpice statements > // <comment>

'spice' mode

An asterisk (*) as the first nonblank character or dollar sign ($), or semicolon (;) placed after statements in the netlist:

* *<comment until end of line>*

or

< SymSpice statements > $ <comment>

or

<SymSPice statement> ; <comment>

Note: Comment can be used inside netlist description if a comment content is denoted by "+" sign before and after comment. That is, the construction "+ <comments> +" marks as a comment several symbols, not the whole line.

Examples in 'local' and 'spectre' mode

1. *// comment1 //SymSpice format*
2. ** comment 2*
3. *r1 5 9 r=20000 // comment 3*
4. *saveOptions options +this is a single comment+ save=all*

Examples in 'spice' mode

1. ** this is a comment 1 //SPICE format*
2. *$ this is a comment 2*
3. *r1 5 9 r=20000 $ this is a comment 3*
4. *.options +this is a single comment+ probe=0*

Names of components

You can create component names with the follow rules:

'local' and 'spectre' modes
Component names can begin with an alphabetic character or with an underline character ("_"), then alphabetic or numeric characters follow or any other character preceding by backslash (\).

'spice' mode
Component names can begin both with an alphabetic character and with a numeric character, but thereafter can contain numbers and the following characters:
! # $ % * + - / < > [ ] _@|

Before and after the name must be token delimiter. See chapter "Delimiters".

Examples
1) For 'local' mode

```
q_a\#b12 c b e s modelq
```

2) For 'spice' mode

```
m_mos1|2 d g s b modelm
```

Splicing symbols

Line splicing symbols, compatible with SymSpice

In SymSpice for line splicing '\' symbol is used.
Splicing symbol is situated at the end of the line. Besides, two backslashes in a row represent masking of one '\' symbol. Therefore splicing is fulfilled in case of sequence of odd number of backslashes.
The blank after splicing symbol is not allowable. Otherwise the rule of symbol masking is fulfilled (i.e. masked symbol is processed as name).
In SymSpice-simulator you can use blank or tab after backslash which will be processed as lines splicing only if element or model name is separated from single '\ ' symbols combination or several such combinations by blank. In this case '\ ' symbols combinations will be processed as separate structure, not as a part of name.

Examples

1) *model default_pmos_1 bsim3v3'\'' ''\'' ''\'' ''\'' '*

type=p

- program will process this example as separate lines:

model default_pmos_1 bsim3v3

type=p.

2) *model default_pmos_1 bsim3v3' ''\'' ''\'' ''\'' ''\'' '*

type=p

- in this case the second line is a continuation of the first line, i.e. program will process this example as:

model default_pmos_1 bsim3v3 type=p,
because model element name is separated from backslash by blank.

Splicing in SymSpice is fulfilled the following way.
The line next to the line, containing splicing symbol is the continuation of this line. However, during two lines splicing between the last symbol of the first line and the first symbol of the next line the blank is placed, therefore element or model name must not be separated on different lines.
Splicing process is completed when empty line or line without closing '\' symbol is found.
In SymSpice-format the splicing with the end of the file is allowable.

Line splicing symbols, compatible with SPICE.

In SPICE for line splicing '+' symbol is used. This symbol is coincides with addition operation symbol used in expressions. Therefore, splicing symbol must be the first character in the line. Unlike SymSpice, '+' symbol is placed at the beginning of the line and indicates splicing with previous line.

1. Blanks and tabs are allowable around '+' symbol. However, element and model names must not be separated, because lines are always separated by blank.
2. Multiple lines with comments are allowable between lines containing '+' symbol.
3. Multiple lines containing more than one blanks and tabs are allowable. (In 'local' mode multiple empty lines are allowable).
4. One empty line is allowable between lines containig '+' symbol.
5. Line containing single '+' symbol is allowable.

In SymSpice-format there are no limitations on number of intermediate empty lines and feature is excluded identifying intermediate empty line and line containing blanks. In this way SymSpice-format requires existence of nonempty first line and thus guarantees more correct work with lines separation methods.

Note: SymSpice simulator allows to splice lines by both methods, but if you use both methods simultaneously SymSpice method will be priority.

Example

1) r1 1 gnd type=resistor
   + r=1k
   - result string: r1 1 gnd type=resistor r=1k

2) r1 1 gnd type=resistor r=1k \
   + 2k
   - result string: r1 1 gnd type=resistor r=1k+2k

1) r1 1 gnd type=resis \
   tor r=1k
   - error.

Delimiters

- Input token delimiters are: tab, blank, comma, equal sign (=), and parentheses “( ) ”.
- Expressions and file names are enclosed in single or double quotes.
- Element attributes are delimited by colons (“Q3:subs”, for example).
- Hierarchy is indicated by periods. For example, “X1.K3.I” is the I node on subcircuit K3 of circuit X1.

Nodes

- Node names can begin with a number or with an identifier which can be alphabetic character or any of the following characters: # _ ! %
- Letters that follow numbers in node names are ignored.
- Node names can contain characters: + - * / $ # [] ! <> _ %
- The following characters are not allowed in node names: () , = ' blank
- The sorting order for operating point nodes is: a-z, !, #, $, %, *, +, -, /
- Zero as the first character of node number is ignored.
- Global statement specifies global nodes across all subcircuits.
- The global SymSpice ground includes nodes 0, gnd, GND and GND!

Node names can also be integers.

Examples

1) 'local' mode

```
name 1_2_n\!h gnd resistor r=100
```

2) 'spice' mode

```
r1 1 2_!d  gnd 1k
```

Instance names

- The names of element instances begin with the element key letter (for example, Q for a BJT element, C for a capacitance, V for a voltage source, and so on), except in subcircuits.
- Subcircuit instance names begin with “X”. (Subcircuits are sometimes called macros or modules.)

Examples

in SPICE and SymSpice format

```
x_block 1 2 3 q
```

Hierarchy paths

- A path name begins with the highest level subcircuit call, then subcircuit names of lower hierarchy level are listed, at the end an element or bottom level node is placed.
- Path hierarchy is indicated by a period.

Numbers

- SymSpice accepts integer or real numbers.
- Numbers can use exponential format or engineering key letter format (see table below).
- Exponents are designated by D or E.
- Alphabetic characters at the end of number considered to be units comments.
- Units comments are not checked.

Suffix	Scale parameter	Name
f	1e-15	femto
p	1e-12	pico
n	1e-9	nano
u	1e-6	micro
mil	25.4e-6	mil (0.001 of inch)
m	1e-3	milli
k	1e3	kilo
meg	1e6	mega
g	1e9	gigo
t	1e12	tera

Example

1. *c=1fF*

means that parameter c has value of 1e-15 and 'F' is a comment, that is the unit of this parameter is Farad.

2. *z=1zzz*

'zzz' is a unit comment and it is not verified as a valid unit name.

3. *.tran 1n 100n*

4. *dcsweep dc dev=vpulse start=0.0e+0000 stop=5.0e+00 step=1e-004*

Parameters and expressions

- Parameter names follow SymSpice name syntax rules.
- If netlist contains several definitions for one parameter, SymSpice takes the last definition.
- Format or other simulation options can be specified using .OPTIONS statement (for SPICE format), or using options statement (for SymSpice format).
- Expressions are enclosed in single or double quotes.

Input netlist file structure

To restructure the input netlist file modules use the Include statement, Lib call statements. These statements can call netlists, model parameters, test vectors, analysis, and option macros into a file from library files or other files. Parameterized data for element sources and models can be stored in external data file, which also can be called by the input netlist file.

Netlist statements

TITLE statement

The first line of circuit input netlist which is launched to simulation can be described as a comment to the circuit. Then, if the file is described in SPICE-format this line is omitted, for SymSpice-netlist the first line must be a comment.

GLOBAL statement

All circuit nodes are local by default. Use the global statement to designate frequently used nodes as common to the main circuit and all subcircuits. This means that all references to a global node name combine within a project and connect to the same node.

Syntax in 'local' or 'spectre' modes

GLOBAL node1 node2 node3

Syntax in 'spice' mode

.GLOBAL node1 node2 node3 ...

where:

Parameter name	Description
node1 …	Specifies global nodes, such as supply and clock names, overrides local subcircuit definitions.

For example, you need to specify ground node with 0 V value, i.e. it is necessary to switch ground node through the resistor and use this node afterwards as a "ground" concept:

Example in 'local' mode

```
global gndA
res1 (gnd gndA) resistor r=1
```

Example in 'spice' mode

```
.global gndA
res1 gnd gndA 1
```

SAVE statement

The SAVE statement saves output variables into the graphic files.
By default SymSpice saves all voltages and supply currents in addition to the output variables. Set options 'SAVE' to save output variables only.

Syntax 'local' or 'spectre' modes

opt OPTIONS SAVE

SAVE out_var ...

Syntax 'spice' mode

.OPTIONS PROBE

.PROBE antype out_var ...

where:

SPICE format	SymSpice format	Description
.OPTIONS PROBE	OPTIONS SAVE	activate save option
out_var …	*out_var …*	Output variables to be plotted. These are voltage, current, or element template variables from a DC, AC, TRAN analysis. Several save options statements may be used.
antype	-	Type of analysis for the specified plots. Analysis types are: DC, AC, TRAN.

Examples in 'local' mode

1) *save 3 r1:1*

Examples in 'spice' mode

1) *.probe v(3) v(5,0) i1(r1)*
2) *.probe ac vm(2) vp(2)*

Note: The following table shows some specific rules concerning data outputting and displaying simulation results for input netlist in SPICE format.

	PROBE (- or +)	PROBE (-)	PROBE (+)
.PROBE ovn	-	nodal voltages and currents through elements branches.	ovn

		If in .probe ovn statement those viewpoints are specified which shouldn't be outputted by default, they also will be outputted in the output file.	
.PROBE	-	nodal voltages and currents through elements branches	-

where:

PROBE (+ or -)	denotes whether statement .option probe is set or not respectively.
all	denotes that all node voltages and branch currents will be displayed
ovn	only output variables specified by user in the .probe statement will be displayed.

PARAMETERS statement

The PARAMETERS statement are used to simplify references to often used variables, or can be used in sweep or statistical analysis.

Netlist parameters can be referenced with anywhere that a numeric value is normally specified on the right-hand side of an "=" sign or within a vector, where the vector itself is on the right-hand side of an "=" sign.

You can use parameter in:
- expressions containing parameter references to specify device instance parameter values,
- model parameter values in model cards (for example specifying "bf=p1*0.8" for a bipolar model parameter, bf)
- initial conditions
- nodesets for individual circuit nodes.

Numeric values, expressions enclosed in single quotes or symbols can be assigned to parameters in SymSpice. You can use any of the following methods to define parameters in input netlist format.

Syntax 'local' or 'spectre' modes

PARAMETERS param = value

PARAMETERS param = expression

Syntax 'spice' mode

.PARAM param = value

.PARAM param = 'expression'

where:

Parameter name	Description
param	parameter name which is encountered on the top hierarchy level or in the subcircuit
value	is a constant or other param
expression	mathematical function (abs, log, sin, cos, ...), calculated as a result of algebraic expressions over parameter defined in the list

Examples

1) *PARAMETERS parA=2*
2) *.PARAM parA=2*

In the example the parameter **parA** assumes value 2, even if other parameter definitions exists higher in the netlist.

INCLUDE statement

Syntax in 'local' or 'spectre' modes

INCLUDE "<filepath> filename"

Syntax in 'spice' mode

.INCLUDE '<filepath> filename'

.INCLUDE ''<filepath> filename''

where:

Parameter name	Description
filepath	Path name of a file.
filename	Name of a file to include in the data file.

Example in 'local' mode

```
include "stimul.txt"
```

Example in 'spice' mode

```
.include 'd:\!Work!\Tests\model.lib'
```

AHDL_INCLUDE statement

Syntax in 'local' or 'spectre' modes

AHDL_INCLUDE "<filepath> filename"

Syntax in 'spice' mode

.AHDL_INCLUDE "<filepath> filename"

where:

Parameter name	Description
filepath	Path name of a file.
filename	Name of file contained VerilogA modules.

TEMP statement

SymSpice allows you to specify three temperatures:

- Model reference temperature, specified in a Model statement using the tref parameter (or temp or tnom, for some models). This is the temperature, in °C, at which the model parameters are measured and extracted. The default value of tnom is 27°C.
- Circuit temperature, specified using a temp statement or the temp parameter. This is the temperature, in °C, at which all elements are simulated. The default circuit temperature is the value of tnom.
- Individual element temperature, specified as the circuit temperature plus an optional amount specified using the dtemp parameter (in SPICE input netlist).

Syntax in 'local' or 'spectre' modes

Name OPTIONS <TEMP> <TNOM>

Syntax in 'spice' mode

.TEMP t1 <t2 <t3 ...>>

where:

Parameter name	Description
t1 t2	Specifies temperatures, in °C, at which the circuit is to be simulated
temp=27 C	Temperature
tnom=27 C	Default component parameter measurement temperature

Example in 'local' mode

t_diod options temp=27

Examples in 'spice' mode

1) *.temp 0 22 60*

The '.TEMP' statement sets the circuit temperatures for the entire circuit simulation. SymSpice uses the temperature set in the '.TEMP' statement, along with the 'TNOM' option setting (or the tref model parameter) and the dtemp element temperature.

2) *.temp 70*

diode1 n1 n2 diod_model dtemp=40

res1 np nn 100 tc1=1 dtemp=-5

.MODEL diod_model D IS=1E-15 VJ=0.6 CJA=1.2E-13

+ CJP=1.3E-14 TREF=50.0

From the '.TEMP' statement, the circuit simulation temperature is given as 70°C. Since tnom is not specified, it defaults to 27°C. The temperature of the diode is given as 40°C above the circuit temperature by the dtemp parameter. That is, for diode1 dtemp = 70°C + 40°C = 110°C. res1 is simulated at 65°C. Since tref is specified at 50°C in the diode model statement, the diode model parameters given are derated by 60°C (110°C - 50°C) for diode1. The value of res1 is derated by 38°C (65°C - tnom).

MODEL statement

The MODEL statement describes element model parameters.
Model can be included in a separate file which is called using include statement (for SymSpice-format) or '.INCLUDE' and '.LIB' (for SPICE format), also model can be described inside the input netlist.
If the model is not specified, default parameters values are used.

Syntax in 'local' or 'spectre' modes

MODEL mname type <pname1=val1> ... <pnameN=valN>

Syntax in 'spice' mode

.MODEL mname type <pname1=val1> ... <pnameN=valN>

where:

Parameter name	Description
mname	Model name reference. Elements must refer to the model by this name. Note: Model names that contain periods (.) can cause the SymSpice automatic model selector to fail under certain circumstances.

type	Selects the model type, which must be one of the following: SymSpice resistor Resistor capacitor Capacitor inductor Inductor diode Diode bjt BJT mos3 MOSFET level=3 bsim3v3 MOSFET level=49 isource Current source vsource Voltage source subckt Subcircuit call SPICE C Capacitor model D Diode model L Inductor model NMOS n-channel MOSFET model NPN npn BJT model PMOS p-channel MOSFET model PNP pnp BJT model R Resistor model
pname1=val1... pnameN=valN	Parameter name. The model parameter name assignment list (pname1) must be from the list of parameter names for the appropriate model type. Default values are given in each model section. The parameter assignment list can be enclosed in parentheses and each assignment can be separated by either blanks or commas for legibility.

Examples in 'local' mode

model npn_mod bjt type=npn is=10e-13 bf=200 va=58.8

model mos_mod mos3 type=p vto=-0.8 ld=0.5e-6

tox=0.4e-7 nsub=0.3e17

Examples in 'spice' mode

.model rload r r=2 tc1=0.2 tc2=0.005

.model d101 d is=1e-10

.model Mod NPN is=10e-14A bf=549 rc=19 rb=39

```
.model C12 c c=1 dev=0.1
.model ModA nmos Level=3 L=2e-6 RS=0.01 RD=0.01 VTO=3
+ TOX=2e-6 RG=5 IS=1e-14 n=1 RB=0.001
```

END statement

The END statement is the last statement in the netlist file.

Syntax in 'local' or 'spectre' modes

END <comment>

Syntax in 'spice' mode

.END <comment>

In 'local' mode the 'END' statement might be omited.
In 'spice' mode the '.END' statement is a required part.

Any text that follows the END statement is treated as a comment and has no effect on that simulation.

Examples

1) *.dc v1 1 2 0.01*

   v1 1 0 dc 0

   d1 1 0 modelD

   .end simulation

2) *.include 'model.txt'*

   q_q1_0 c b e s mod

   .tran 1n 100n

   .end

ALTER

.ALTER statement is used to alter netlist statements or provide multiple analysis.

Syntax

Syntax in 'local' or 'spectre' modes

This statement is allowed in 'spice' mode only.

Use statement 'simulator lang= spice' to switch on 'spice' mode.

Syntax in 'spice' mode

.ALTER

The .ALTER section can contain the following statements: .PARAM , .MEAS, .TRAN, .DC, .AC.
If you use multiple .ALTER sections, they are performed one after another.

Note

You cannot alter the circuit structure in ALTER section.

When multiple analysis is defined each analysis defined in ALTER section is followed by all foregoing analyses, for example:

```
...
.param par=0
.tran                 //  result file = .tr0
.alter
.param par=1
.dc                   // result file = .tr1 + .dc0
.alter
.param par=2
.ac                   //  result file = .tr2 + .dc1 + .ac0
.alter
.param par=3
.tran                 //  result file = .tr3 + .dc2 + .ac1 + .tr4

.end
```

Example

```
.PARAM _frequency = 10Meg
.PARAM _capacity   = Opt1 (100p, 50p, 120p)
.MODEL OptMod Opt Method = bisection
.PARAM C =  1e-011

c_cap1_0 n_2 n_1 c= C
c_cap2_1 gnd n_2 c=_capacity
l_l1_2 gnd n_2 l=2.5e-006 r=1e-6
r_res1_3 gnd n_2 r = 20k
v_vsin1_4 n_1 gnd sin 0 1 _frequency ac=1

.ac dec 1000 5Meg  12Meg sweep
+ OPTIMIZE = Opt1
+ RESULTS  = freq $ Look at measure
+ MODEL    = OptMod

.measure ac  maxVout MAX v(n_2)
.measure ac freq when v(n_2) = maxVout goal = 10Meg $_frequency

.alter
.param C = 0.5e-11
.alter
.param C = 2e-11
.alter
.param C = 5e-11
```

OPTIONS statement

Control options are used to change default behaviour of simulator.

Syntax

Syntax in 'local' or 'spectre' modes

<Name> OPTIONS <opt1> <opt2 opt3 ...>

Syntax in 'spice' mode

.OPTIONS <opt1> <opt2 opt3 ...>

where:

Parameter name	Description
opt1 opt2 ...	Specifies control options. Options can be represented in two forms: 1. <opt>, where <opt> - name of the option. If option is specified its value assumed to be 'true'. 2. <opt>=<val>, where <opt> - name of the option and <val> - value of the option.
Name	The name of certain statement in SymSpice format.

General Control Options

SPICE format	SymSpice format	Description
apb, csdf, csv	rawfmt= {apb|csdf|csv|none}	Specifies output file format, where APB - internal Symica binary format, CSDF and CSV - ASCII format. The default behavior if no output file format specified: in SPICE format - no output file will be generated . in SymSpice format - APB format will be generated.
probe = {0 | 1}	save= {all|selected| none} currents= {all|selected}	Specifies set of variables to be saved in output file. SAVE=ALL Saves all node voltages and all currents of element. SAVE=SELECTED Saves specified markers only. SAVE=NONE Does not save any data.

		By default, SAVE=SELECTED. CURRENT=ALL Save all currents. CURRENT=SELECTED Save specified markers only.
ingold = {1 | 2}	-	Specifies data format in .lis file. INGOLD=1 Fixed format for values between 0.1 and 999, and exponential format for values greater than 999 or less than 0.1. INGOLD=2 Exponential format for all values. By default ingold=1.
measdgt	-	Specifies number of digits representing values in .lis and .mt0 file (*.ma0, *mt0, *ms0, etc). By default MEASDGT=8.
numdgt	numdgt	Specifies number of digits representing values in output file. By default NUMDGT=8.
acout	acout	Specifies method to calculate the values of real or imaginary parts for complex voltages/currents of AC analysis. If acout=0 VR(N1,N2)=REAL(V(N1,0)-V(N2,0)) VI(N1,N2)=IMAG(V(N1,0)-V(N2,0)) $VM(N1,N2)=(VR(N1,N2)^2+VI(N1,N2)^2)^{0.5}$ VP(N1,N2)=ARCTAN(VI(N1,N2)/VR(N1,N2)) VDB(N1,N2)=20*LOG10(VM(N1,N2)) If acout=1 (default value) VR(N1,N2)= REAL(V(N1,0)) - REAL(V(N2,0)) VI(N1,N2)= IMAG(V(N1,0)) - IMAG(V(N2,0)) $VM(N1,0)=(VR(N1,0)^2 + VI(N1,0)^2)^{0.5}$ $VM(N2,0)=(VR(N2,0)^2 + VI(N2,0)^2)^{0.5}$ VM(N1,N2)= VM(N1,0) - VM(N2,0) VP(N1,0)= ARCTAN(VI(N1,0)/VR(N1,0)) VP(N2,0)= ARCTAN(VI(N2,0)/VR(N2,0)) VP(N1,N2)= VP(N1,0) - VP(N2,0) VDB(N1,0)=20*LOG10(VM(N1,0)) VDB(N1,N2)= 20*LOG10(VM(N1,0)/VM(N2,0))

		here: VR, VI - real and imaginary parts; VM - magnitude; VP - phase; VDB - decibel.

Analysis Options

SPICE format	SymSpice format	Description
homotopy	homotopy	Specifies method used for DC analysis; possible values are none, gmin, source, ptran, tran, or all (by default).
gmin	gmin	Specifies conductances that are placed in parallel with all p-n junctions for DC analysis.
delmax	See analysis statement	Specifies maximum integration step for TRAN analysis.
method	method	Specifies integration method in transient analysis; possible values are EULER, TRAPONLY, TRAP, GEAR2ONLY, GEAR2, GEAR. By default METHOD=TRAPONLY.
itl4 (imax)	See analysis statement	Maximum number of Newton iterations at one time point, before taking a smaller time step. Default value is 8.

Fast-SPICE Options

SPICE format	SymSpice options	Description
fast_spice= {0 | 1}	fast_spice= {0 | 1}	Sets the different modes to simulate circuit: 0 – dynamic decomposition algorithms are disabled. 1 – dynamic decomposition algorithms are enabled.

Multithreading Options

SPICE format	SymSpice options	Description
multithread	multithread	This option turns on/off multithreading capability in TRAN analysis. Possible values are *on*, *off*. Default value is *on*.

nthreads	nthreads	Specifies the number of threads.
parallel_sweeps	parallel_sweeps	This option turns on/off capability to simulate parameterized analyses in parallel. This option can be applied for multiple analyses simulation, secondary sweep simulation, .MONTE analysis and .ALTER group. This option disables option 'multithread'. Possible values are *on*, *off*. Default value is *off*.
nsweeps	nsweeps	Specifies the number of parameterized analyses run in parallel.

Accuracy Control Options

SPICE format	SymSpice format	Description
absv (vntol) absi (abstol)	vabstol, iabstol	Convergence criteria setting the absolute tolerances for the computed values in the last two Newton iterations. The default value for vabstol/absv is 1e-6, and for iabstol/absi is 1e-12.
reltol	reltol	Convergence criteria setting the maximum relative tolerance for the computed values in the last two Newton iterations. Default value is 0.001.
Iteratio	Iteratio	Criteria setting the local truncation error tolerances. To increase accuracy (conservative mode) set value to 1, to increase simulation speed (liberal mode) set value to 10. Default value (moderate mode) is 3.5. It can be set either in options or in analysis statement.

Model Options

SPICE format	SymSpice format	Description
dcap	-	Selects method for depletion capacitance calculation for level=1 and 3 diodes, and BJT transistors
expli	-	Model parameter which specifies current explosion for BJT and Diode
scale	scale	Element scaling factor. Scales parameters in element cards, by their value. By default in SymSpice scale=1.

scalm	scalem	Model scaling factor. Scales model parameters by their value. By default in SymSpice scalm=1
aspec	-	Set this option for SymSpice to read aspec netlists and models and the results are compatible. By default aspec=0.

Parasitic Reduction

To simulate circuits extracted from layout with high quantity of RC parasitic elements (parasitics) you can use reduction statement *@ prs_options.

All reduction statements begin with two symbols "*@", so that these statements are omitted in other simulation programs processing asterisk as a comment.

Global nodes (global) and voltage markers (probe) are not effected with *@ prs_options.

Syntax

*@ prs_options <fold all>

+ <fold-acc>

+ <fold-serial-resistances> <fold-resistances-nodes>

+ <remove-floating-elements> <remove-short-elements> <remove-capacitances>

+ <remove-gate-small-resistances> <fold-signal-line-rc-series>

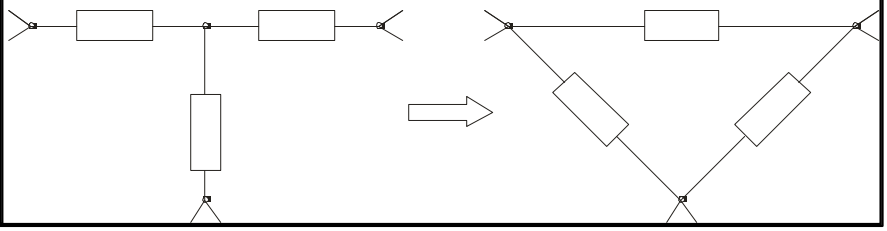
+ <fold-power-line-rc-series> <fold-power-line-small-resistances> <remove-power-line-r-series>

Multiplexed Parasitic Options

Parameter name	Description
fold-all	Implements all options.
fold-acc	Implements all options realizing equivalent circuit transformation: fold-serial-resistors, fold-resistances-nodes, remove-floating-elements, remove-short-elements

Fold Parasitic Elements and Nodes

Parameter name	Description
fold-serial-resistances	Option to reduce R parasitic serial resistors.

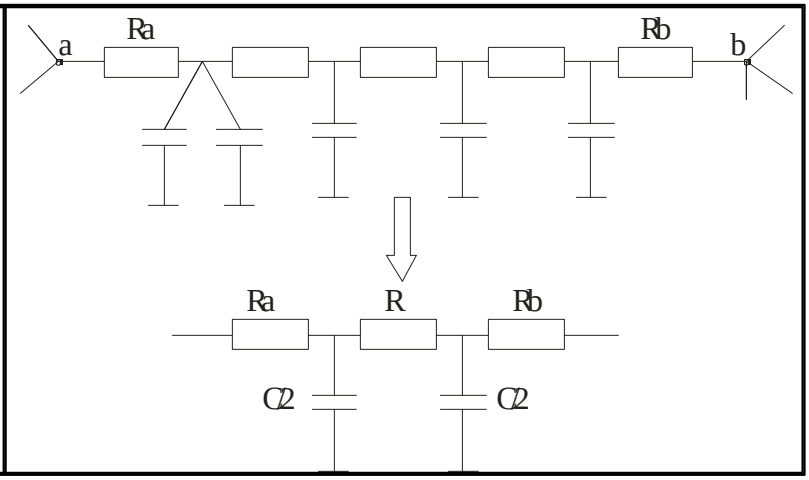
fold-resistances-nodes	Option to reduce R parasitic which form "star" shape to "triangle" shape. 

Remove Parasitic Elements

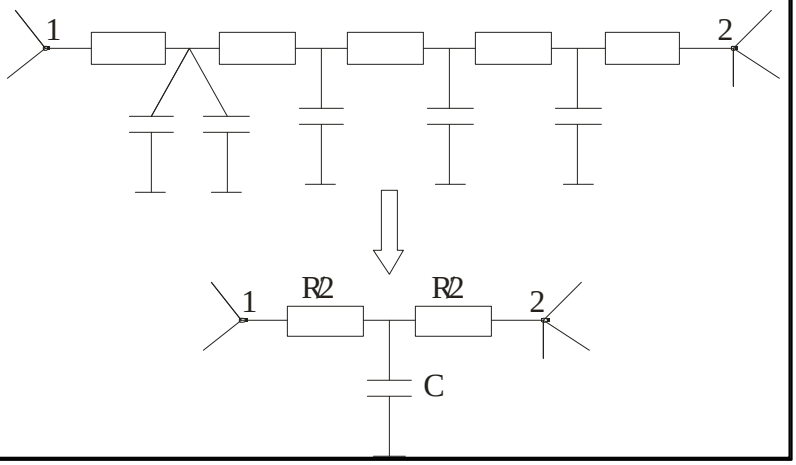
Parameter name	Description
remove-floating-elements	Option to delete R parasitic with one pin hanging.
remove-short-elements	Option to delete R parasitic with both pins connected to the same node.
remove-capacitances	Option to delete parallel C parasitic in power line and signal line. This option is supplemental to remove-power-line-r-series.

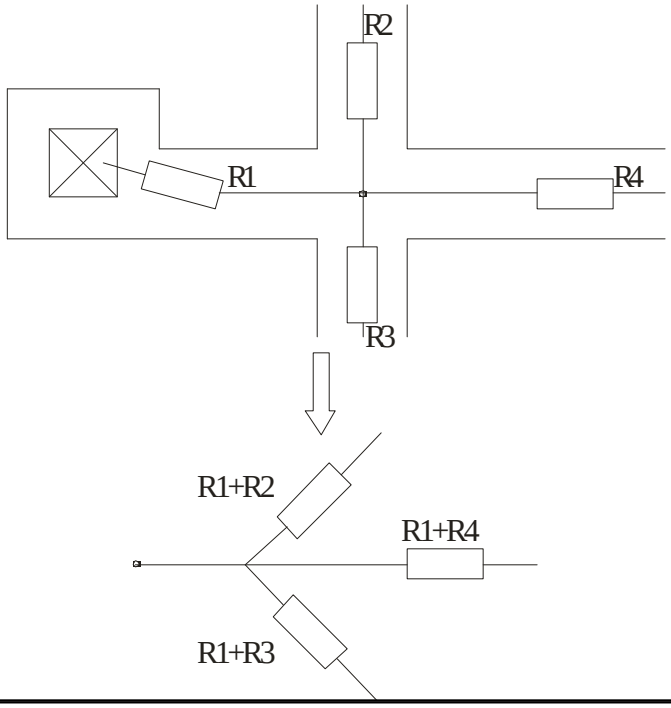
Signal Line Parasitic Options

Parameter name	Description
remove-gate-small-resistances	Option to delete from the whole circuit R parasitic connected to gates of MOS49–transistors (independently of *@ prs_r_models statement).
fold-signal-line-rc-series	This option reduces RC parasitic chains which form signal line into TT-type circuit, where Ra, Rb – resistors connected to nodes a and b; R – the total resistance of the chain (P-arm), C – the total capacitance of the chain. Signal line is a chain connected to all pins of MOS49 transistor. R = R1 + R2 + ... + Rn C = C1 + C2 + ... + Cn



Power Line Parasitic Options

Parameter name	Description
fold-power-line-rc-series	This option reduces RC parasitic chains which form power line into T-type circuit, where R – the total resistance of the chain, C – the total capacitance of the chain. R = R1 + R2 + ... + Rn C = C1 + C2 + ... + Cn 

fold-power-line-small-resistances	Option to reduce in circuit's supply lines small R parasitic (value of resistance five time less than values of neighboring resistances). 
remove-power-line-r-series	Option to delete R parasitic chains in power and ground lines. All C parasitics will be folded. If remove-capacitances is specified C parasitics will be removed.

Defining R and C parasitics

R-parasitics

R parasitics are defined by element models specified in prs_r_models.

Syntax

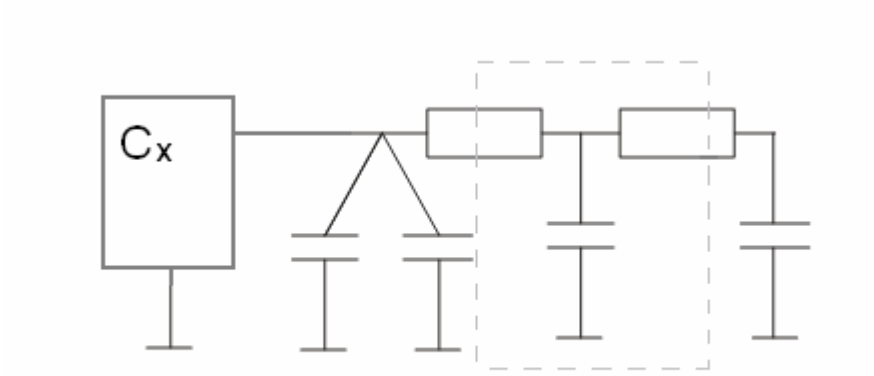
@ prs_r_models mname type

where:

Parameter name	Description
mname	Model name. These models are used to identify R.
type	Selects the model type, which must be one of the following: R Resistor model PHY Physical resistor model

C-parasitics

C-parasitic is defined as an any capacitor with one node connected to R-parasitics and another node connected to ground.



Defining power lines

Recognizing of the parasitic chains of power or ground bus is implemented by R parasitics (defined in *@ prs_r_models).
An algorithm of the power/ground bus forming:
1. User specifies nodes connected to power/ground bus in prs_pwr_nodes statement.
2. R parasitics are used to generate RC parasitic chains starting from specified nodes.

Syntax

*@ prs_pwr_nodes nname

where:

Parameter name	Description
nname	Node name.

Active Blocks

During simulation SymSpice dynamically divides circuit into active (where state is being switched at the present time) and passive (where state is not being switched or being switched slowly) parts. SymSpice uses the automatic method of dividing, however you can choose any part of circuit which will be processed as an active one.
An active block is the block which is active permanently.
For example, You can choose an inner clock generator as permanently active block. This will reduce time on automatic search and time to divide circuit into active and passive parts.

Syntax

*@ *active_blocks SubcktInstance1...*

where:

Parameter name	Description
active_blocks	Reserved word.
SubcktInstance1...	Subcircuit block.

Example:

@analog_blocks X1

MEASURE

The MEASURE statement is used to modify information and define the results of successive simulations
Note: the '.MEASURE' statement is used for input netlist in SPICE format.
The MEASURE statement prints user-defined electrical specifications of a circuit and is used extensively in optimization. The specifications include propagation, delay, rise time, fall time, peak-to-peak voltage, minimum and maximum voltage over a specified period, and a number of other user-defined variables. With either the error function or GOAL parameter, MEASURE is also used extensively for optimization of circuit component values and curve fitting measured data to model parameters.
There are several different formats for MEASURE statement , which depend on the application. They can be used for either DC sweep, AC, or transient analysis.
Fundamental measurement modes are:

- *Rise, fall, and delay*
- *Find-when*
- *Equation evaluation*
- *Average, RMS, MIN, MAX, and Peak-to-Peak*
- *Integral evaluation*
- *Derivative evaluation*

When a MEASURE statement fails to execute, SymSpice writes 'not found' to measure output file.

Measure Parameter Types

Measurement parameter names cannot conflict with standard parameter names. Values, obtained with MEASURE statement, are not the parameters so they cannot be used in expressions as parameters. The following example illustrates the way SymSpice works with MEASURE statement parameters.

```
...
.MEASURE tran length TRIG v(clk) VAL=1.4 TD=11ns RISE=1
+ TARGv(neq) VAL=1.4 TD=11ns RISE=1
.SUBCKT path out in width=0.9u length=600u
+ rm1 in m1 m2mg w='width' l='length/6'
...
.ENDS
```

In the above listing, the 'length' in the resistor statement
rm1 in m1 m2mg w='width' l='length/6'
does not inherit its value from the length in the MEASURE statement
MEASURE tran length ...
since they are of different types. The correct value of l in rm1 should be
l = length/6 = 100u
instead of a value derived from the measured value in transient analysis.

Rise, Fall, and Delay

This format is used to measure independent-variable (time, frequency, or any parameter or temperature) differential measurements such as rise time, fall time, slew rate, and any measurement that requires the determination of independent variable values. The format specifies substatements TRIG and TARG. These two statements specify the beginning and ending of a voltage or current amplitude measurement. The rise, fall, and delay measurement mode computes the time, voltage, or frequency between a trigger value and a target value. Examples for transient analysis include rise/fall time, propagation delay, and slew rate measurement. Applications for AC analysis are the measurement of the bandwidth of an amplifier or the frequency at which a certain gain is achieved.

Syntax in 'local' or 'spectre' modes

Not available. Use insertion in 'spice' mode to use MEASURE functionality.

Syntax in 'spice' mode

.MEASURE <DC|AC|TRAN> result TRIG … TARG … <GOAL=val>

+<MINVAL=val> <WEIGHT=val>

where:

Parameter name	Description
MEASURE	Specifies measurements. A shortening MEAS also can be used.
result	Name that is associated with the measured value in the SymSpice output. The item measured is the independent variable beginning at the trigger and ending at the target: for transient analysis it is time; for AC analysis it is frequency; for DC analysis it is the DC sweep variable. If the target is reached before the trigger is activated, the resulting value is negative. Note: The terms "DC", "TRAN", and "AC" are illegal for result name.
TRIG..., TARG...	Identifies the beginning of trigger and target specifications, respectively.
<DC|AC|TRAN>	Specifies the analysis type of the measurement. If omitted, the last analysis mode requested is assumed.
GOAL	Specifies the desired measure value in optimization.
MINVAL	If the absolute value of GOAL is less than MINVAL, the GOAL value is replaced by MINVAL in the denominator of the ERRfun expression. Default=1.0e-12.

WEIGHT	The calculated error is multiplied by the weight value. Used in optimization. Default=1.0.

Syntax

TRIG (Trigger) Syntax

TRIG trig_var VAL=trig_val <TD=time_delay>

+<CROSS=c> <RISE=r> <FALL=f>

or

TRIG AT=val

TARG (Target) Syntax

TARG targ_var VAL=targ_val <TD=time_delay>

+<CROSS=c | LAST> <RISE=r | LAST> <FALL=f | LAST>

where:

Parameter name	Description
TRIG	Indicates the beginning of the trigger specification
trig_val	Value of trig_var at which the counter for crossing, rises, or falls is incremented by one
trig_var	Specifies the name of the output variable, which determines the logical beginning of measurement. If the target is reached before the trigger is activated, MEASURE reports a negative value.
TARG	Indicates the beginning of the target signal specification.
time_delay	Amount of simulation time that must elapse before the measurement is enabled. The number of crossings, rises, or falls is counted only after time_delay value. The default trigger delay is zero.
CROSS=c RISE=r FALL=f	The numbers indicate which occurrence of a CROSS, FALL, or RISE event causes a measurement to be performed. For RISE=r, the WHEN condition is met and measurement is performed when the designated signal has risen r rise times. For FALL =f, measurement is performed when the designated signal has fallen f fall times. A crossing is either a rise or a fall, so for CROSS=c, measurement is performed when the designated signal has achieved a total of c crossing times, as a result of either rising or falling. For TARG, the last event is specified with the LAST keyword.

LAST	Measurement is performed when the last CROSS, FALL, or RISE event occurs. For CROSS = LAST, measurement is performed the last time the WHEN condition is true for either a rising or falling signal. For FALL = LAST, measurement is performed the last time the WHEN condition is true for a falling signal. For RISE = LAST, measurement is performed the last time the WHEN condition is true for a rising signal. LAST is a reserved word and cannot be chosen as a parameter name in the above MEASURE statements.
AT=val	Special case for trigger specification. The "val" is the time for TRAN analysis, the frequency for AC analysis, or the parameter for DC analysis, at which measurement is to start.

Examples

1) *.MEASURE TRAN timedelay TRIG v(vin) VAL=1.5 TD=5n FALL=1*

+ TARG v(node1) VAL=1.5 RISE=1

In this example a propagation delay measurement is taken between nodes 'vin' and 'node1' for a transient analysis. The delay is measured from the first falling edge of the voltage at node 'vin' to the first rising edge of node 'node1'. The measurement is specified to begin when the first falling voltage at node vin is 1.5 V and to end when the first rising voltage at node node1 reaches 1.5 V. The TD=5n parameter does not allow count time delay until after 5 ns has elapsed.

2) *.MEASURE TRAN rise_t TRIG I(Q1) VAL=0.5m RISE=3*

+ TARG I(Q1) VAL=4.5m RISE=3

.MEASURE pwidth TRIG AT=10n TARG V(IN) VAL=2.5 CROSS=3

The last example uses the short form of TRIG. AT=10n specifies that the time measurement is to begin at time t=10 ns in the transient analysis. The TARG parameters specify that the time measurement is to end when V(IN)=2.5 V on the third crossing.
Note: If the .TRAN statement is used in conjunction with a .MEASURE statement, using a nonzero START time in the .TRAN statement can result in incorrect .MEASURE results. Do not use nonzero START times in .TRAN statements when .MEASURE is also being used, use TD instead.

FIND and WHEN Functions

The FIND and WHEN functions allow any independent variables (time, frequency, parameter), any dependent variables (voltage or current, for example), or the derivative of any dependent variables to be measured when some specific event occurs. These measure statements are useful in unity gain frequency or phase measurements, as well as for measuring the time, frequency, or any parameter value when two signals cross each other, or when a signal crosses a constant value. The measurement starts after a specified time delay, TD. It is possible to find a specific event by setting RISE, FALL, or CROSS to a value (or parameter) or LAST for last event. LAST is a reserved word and cannot be chosen as a parameter name in the above measure statements.

Syntax

Syntax in 'local' or 'spectre' modes

Not available. Use insertion in 'spice' mode to use MEASURE functionality.

Syntax in 'spice' mode

.MEASURE <DC|TRAN|AC> result WHEN out_var = val <TD = val>

+< RISE=r | LAST > < FALL=f | LAST > < CROSS=c | LAST >

+<GOAL=val> <MINVAL=val> <WEIGHT=val>

or

.MEASURE <DC|TRAN|AC> result WHEN out_var1=out_var2

+< TD=val > < RISE=r | LAST > < FALL=f | LAST >

+< CROSS=c| LAST > <GOAL=val> <MINVAL=val> <WEIGHT=val>

or

.MEASURE <DC|TRAN|AC> result FIND out_var1 WHEN out_var2=val

+< TD=val > < RISE=r | LAST > < FALL=f | LAST > < CROSS=c| LAST >

+<GOAL=val> <MINVAL=val> <WEIGHT=val>

or

.MEASURE <DC|TRAN|AC> result FIND out_var1 WHEN out_var2 = out_var3

+<TD=val > < RISE=r | LAST > < FALL=f | LAST > <CROSS=c | LAST>

+<GOAL=val> <MINVAL=val> <WEIGHT=val>

or

.MEASURE <DC|TRAN|AC> result FIND out_var1 AT=val <GOAL=val>

+<MINVAL=val> <WEIGHT=val>

where:

Parameter name	Description
CROSS=c RISE=r FALL=f	The numbers indicate which occurrence of a CROSS, FALL, or RISE event causes a measurement to be performed. For RISE=r, the WHEN condition is met and measurement is performed when the designated signal has risen r rise times. For FALL =f, measurement is performed when the designated signal has fallen f fall times. A crossing is either a rise or a fall, so for CROSS=c, measurement is performed when the designated signal has achieved a total of c crossing times, as a result of either rising or falling.
<DC|TRAN|AC>	Specifies the analysis type of the measurement. If omitted, the last analysis type requested is assumed.
FIND	Selects the FIND function.
GOAL	Specifies the desired .MEASURE value. It is used in optimization.
LAST	Measurement is performed when the last CROSS, FALL, or RISE event occurs. For CROSS = LAST, measurement is performed the last time the WHEN condition is true for either a rising or falling signal. For FALL = LAST, measurement is performed the last time the WHEN condition is true for a falling signal. For RISE = LAST, measurement is performed the last time the WHEN condition is true for a rising signal. LAST is a reserved word and cannot be chosen as a parameter name in the above .MEASURE statements.
MINVAL	If the absolute value of GOAL is less than MINVAL, the GOAL value is replaced by MINVAL in the denominator of the ERRfun expression. Default=1.0e-12.
out_var(1,2,3)	Variables used to establish conditions at which measurement is to take place.
result	Name which is associated with the measured value in the SymSpice output.
TD	Identifies the time at which measurement is to start.
WEIGHT	Calculated error is multiplied by the weight value. Default=1.0
WHEN	Selects the WHEN function.

Examples

1) *.MEASURE TRAN result1 FIND v(node3) WHEN v(out)=1.5 RISE=4*

The example above defines a voltage value in the 'node3' when the voltage value in the node 'out' is equal to 1.5 at the fourth rising edge.

2) *.MEASURE TRAN result2 WHEN v(node2)=1.5 TD=100n RISE=1*
The example above calculates a voltage value at the first rising edge when a voltage value in the 'node2' equal to 1.5 after 100ns after simulation started.

Equation Evaluation

This statement is used to calculate an equation that is a function of the results of previous MEASURE statements. The equation must not be a function of node voltages or branch currents.

Syntax

Syntax in 'local' or 'spectre' modes

Not available. Use insertion in 'spice' mode to use MEASURE functionality.

Syntax in 'spice' mode

.MEASURE <DC|TRAN|AC> result PARAM='equation'

+<GOAL=val> <MINVAL=val>

Examples

1) *.MEASURE TRAN result_1 WHEN v(node1)=v(node2)*

2) *.MEASURE TRAN result_2 FIND v(node4) WHEN v(out)=1.5 FALL=3*

3) *.MEASURE TRAN result_total param='result_1 + result_2'*

The example above calculates value that is the function of previous measured parameters.

Average, RMS, MIN, MAX, INTEG, and Peak-to-Peak

The average (AVG), RMS, MIN, MAX, and peak-to-peak (PP) measurement modes report statistical functions of the output variable rather than the analysis value.

Syntax

Syntax in 'local' or 'spectre' modes

Not available. Use insertion in 'spice' mode to use MEASURE functionality.

Syntax in 'spice' mode

.MEASURE <DC|TRAN|AC> result func out_var <FROM=val> <TO=val>

+<GOAL=val> <MINVAL=val> <WEIGHT=val>

where:

Parameter name	Description
<DC|TRAN|AC>	Specifies the analysis type of the measurement. If omitted, the last analysis mode requested is assumed.
FROM	Specifies the initial value for the "func" calculation. For transient analysis, value is in units of time.
TO	Specifies the end of the "func" calculation.
GOAL	Specifies the desired .MEASURE value. It is used in optimization.
MINVAL	If the absolute value of GOAL is less than MINVAL. Default=1.0e-12.
func	Indicates one of the following types for measure statement: • AVG (average): Calculates the area under the out_var divided by the periods of interest • MAX (maximum): Reports the maximum value of the out_var over the specified interval • MIN (minimum): Reports the minimum value of the out_var over the specified interval • PP (peak-to-peak): Reports the maximum value minus the minimum value of the out_var over the specified interval • RMS (root mean squared): Calculates the square root of the area under the out_var2 curve divided by the period of interest

result	Name that is associated with the measured value in the SymSpice output. The value is a function of the variable specified (out_var) and func.
out_var	Name of any output variable whose function ("func") is to be measured in the simulation.
WEIGHT	The calculated error is multiplied by the weight value. Default=1.0.

Examples

1) *.MEAS TRAN MAXVAL MAX V(1,2) FROM=35ns TO=90ns*

The example above finds the maximum voltage difference between nodes 1 and 2 for the time period from 35 ns to 90 ns.

2) *.MEAS TRAN avg_value AVG V(10) FROM=50ns TO=100ns*

The example above calculates the average nodal voltage value for node 10 during the transient sweep from the time 50 ns to 100 ns and prints out the result as "avg_value".

INTEGRAL Functions

The INTEGRAL function calculates the integral of an output variable over a specified period.

Syntax

Syntax in 'local' or 'spectre' modes

Not available. Use insertion in 'spice' mode to use MEASURE functionality.

Syntax in 'spice' mode

.MEASURE <DC|AC|TRAN> result INTEGRAL out_var <FROM=val> <TO=val>

+<GOAL=val> <MINVAL=val> <WEIGHT=val>

The same syntax used for the average (AVG), RMS, MIN, MAX, and peak-to-peak (PP) measurement mode is used for the INTEGRAL function with func to be defined as INTEGRAL (INTEG).

Examples

The following example calculates the integral of P(r1) from 30 ns to 100 ns.

.MEAS TRAN power INTEG P(r1) FROM=30ns TO=100ns

DERIVATIVE Function

The DERIVATIVE function provides the derivative of an output variable at a given time or frequency or for any sweep variable, depending on the type of analysis. It also provides the derivative of a specified output variable when some specific event occurs.

Syntax

Syntax in 'local' or 'spectre' modes

Not available. Use insertion in 'spice' mode to use MEASURE functionality.

Syntax in 'spice' mode

.MEASURE <DC|TRAN|AC> result DERIVATIVE out_var AT=val <GOAL=val>

+<MINVAL=val> <WEIGHT=val>

or

.MEASURE <DC|TRAN|AC> result DERIVATIVE out_var WHEN var2=val

+ <RISE=r | LAST> <FALL=f | LAST> <CROSS=c | LAST> <TD=tdval>

+ <GOAL=goalval> <MINVAL=minval> <WEIGHT=weightval>

or

.MEASURE <DC|TRAN|AC> result DERIVATIVE out_var WHEN var2=var3

+ <RISE=r | LAST> <FALL=f | LAST> <CROSS=c | LAST> <TD=tdval>

+ <GOAL=goalval> <MINVAL=minval> <WEIGHT=weightval>

where:

Parameter name	Description
AT=val	Value of out_var at which the derivative is to be found
CROSS=c RISE=r FALL=f	The numbers indicate which occurrence of a CROSS, FALL, or RISE event causes a measurement to be performed. For RISE=r, the WHEN condition is met and measurement is performed when the designated signal has risen r rise times. For FALL =f, measurement is performed when the designated signal has fallen f fall times. A crossing is either a rise or a fall, so for CROSS=c, measurement is performed when the designated signal has achieved a total of c crossing times, as a result of either rising or falling.

<DC|TRAN|AC>	Specifies the analysis type of the measurement. If omitted, the last analysis mode requested is assumed.
DERIVATIVE	Selects the derivative function. A shortening DERIV also can be used.
GOAL	Specifies the desired .MEASURE value. It is used in optimization.
LAST	Measurement is performed when the last CROSS, FALL, or RISE event occurs. For CROSS = LAST, measurement is performed the last time the WHEN condition is true for either a rising or falling signal. For FALL = LAST, measurement is performed the last time the WHEN condition is true for a falling signal. For RISE = LAST, measurement is performed the last time the WHEN condition is true for a rising signal. LAST is a reserved word and cannot be chosen as a parameter name in the above .MEASURE statements.
MINVAL	If the absolute value of GOAL is less than MINVAL, the GOAL value is replaced by MINVAL in the denominator of the ERRfun expression. Default=1.0e-12.
out_var	Variable for which the derivative is to be found
result	Name which is associated with the measured value in the SymSpice output
TD	Identifies the time at which measurement is to start
var(2,3)	Variables used to establish conditions at which measurement is to take place
WEIGHT	The calculated error between result and GOAL is multiplied by the weight value. Default=1.0.
WHEN	Selects the WHEN function

Examples

1) The following example calculates the derivative of v(node1) when v(node2) is equal to 1.2 at the third rising edge:

```
.MEAS TRAN result_4 derivative v(node1) WHEN v(node2)=1.2 RISE=3
```

2) The following example calculates the derivative of VP(output)/360.0 when the frequency is 10 kHz:

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

SUBCKT

Input netlist structure includes initial comment used as circuit title, circuit description, hierarchical block description (if it's presented in the circuit) and ends with netlist description ending statement (in case of SPICE format the .end statement is necessary, though SymSpice can simulate circuits without this statement).

SUBCKT statement

The SUBCKT statement is used to define a subcircuit.
Subcircuit represents hierarchical block inside which the set of its own elements, nodes and circuit ports is defined. Through the external nodes subcircuit is embedded in the netlist, terminal device circuit is built, outside signals are connected to it and simulation mode can be defined.
Several hierarchy levels can be used in the circuit.

Syntax in 'local' or 'spectre' modes

SUBCKT subnam <(> node1 ... nodeN <)> < parameters name1= value1 ... <nameN=valueN>>

Syntax in 'spice' mode

.SUBCKT subnam node1 < node2 node3 ...> < parnam=val ...>

or

.MACRO subnam node1 < node2 node3 ...> < parnam=val ...>

where:

Parameter name	Description
subckt	The keyword subckt (.subckt is used in 'spice' mode).
subnam	Specifies reference name for the subcircuit model call.
node1 ... nodeN	The external or connecting nodes of the subcircuit to the main circuit; cannot be ground node (zero). Any element nodes appearing in the subcircuit but not included in this list are strictly local, with three exceptions: • the ground node (zero); • nodes assigned using BULK=node in the MOSFET or BJT models; • nodes assigned using the global statement.

parameters name1=value1... nameN=valueN	This is an optional parameter specification field. You can specify default values for subcircuit calls that refer to this subcircuit. The field contains the keyword parameters followed by the names and values of the parameters you want to specify.
parnam	A parameter name set to a value. For use only in the subcircuit, overridden by an assignment in the subcircuit call or by a value set in a Param statement.

Examples:

1) *subckt and2_not in1 in2 out*

m_pmos1 out n3 g_vdd! g_vdd! pmos l=1e-6 w=1e-5

m_pmos2 out n2 g_vdd! g_vdd! pmos l=1e-6 w=1e-5

...

.macro or2_not in1 in2 out

m_nmos1 out n2 gnd! gnd! nmos l=1e-6 w=1e-5

m_nmos2 out n3 gnd! gnd! nmos l=1e-6 w=1e-5

...

2) *subckt d_trigger (D C Ra Q nQ)*

x_and21 D C R and2_not

x_and22 R C S and2_not

...

Subcircuit Call statement

To connect all local nodes on the subcircuit level with external nodes, use list of circuit nodes in hierarchical block description on the top project level. They are described sequentially and correspond to those local subcircuit nodes that follow the subckt statement.

Syntax in 'local' or 'spectre' modes

Name <(> < node1> ... < nodeN> <)> subnam

Syntax in 'spice' mode

Xyyy node1 <node2 node3 ...> subnam <parnam=val ...> <M=val>

where:

SPICE mode	SymSpice mode	Description
Xyyy	Name	Subcircuit element name. For SPICE format netlist it must begin with an "X", which may be followed by alphanumeric characters.
node1 … nodeN	node1 … nodeN	Node names for external reference
subnam	subnam	Subcircuit model reference name
parnam	-	A parameter name set to a value for use only in the subcircuit. It overrides a parameter value assigned in the subcircuit definition, but is overridden by a value set in a Param statement.
M	-	Multiplier. Makes the subcircuit appear as M subcircuits in parallel. This is useful in characterizing circuit loading. No additional calculation time is needed to evaluate multiple subcircuits.

Examples

1) *X1 n1 n2 inv CAPOUT=CINV*
The above example calls a subcircuit model named inv. It assigns the parameter CAPOUT = CINV the parameters CAPOUT given in the .subckt statement (not shown). The subcircuit name is X1. All subcircuit names must begin with X.

2) *X1 n1 n2 inv*

Subcircuit Parameter Definition

Parameter statement can define parameters values which are used on different hierarchy levels including subcircuits.
Parameters defined in the subcircuit description can serve as value for measurement of individual parameters of element model specified inside circuit.

Syntax in 'local' or 'spectre' modes

SUBCKT subname <(>pinlist<)> < paramnamexxx=> <value>

Syntax in 'spice' mode

.SUBCKT <subname> <pinlist> <paramnamexxx => <value>

where

Parameter name	Description
.SUBCKT/subckt	the keyword subcircuit
pinlist	the external or connecting nodes of the subcircuit to the main circuit
paramnamexxx	parameter name which will be used inside subcircuit
value	parameter value

Example in 'spice' mode

The following example implements an inverter with a Mult parameter. By default, the inverter can drive three devices. By entering a new value for the parameter Mult in the element line, the user can select larger or smaller inverters to suit the application.

```
.SUBCKT Inverter in out vdd Mult = 2
MPMOS1 out in vdd vdd Pmos_mod L=1.5u W='Mult * 2e-6 + 3e-6'
MNMOS1 out in 0 0 Mmos_mod L=1.5u W='Mult * 1e-6 + 3e-6'
.ENDS
...
x_inv0 in o0 vdd1 Inverter $ Default values: mpmos1 = 6u, mnmos1=3u
x_inv1 in o1 vdd2 Inverter Mult=3 $ mpmos1 = 9u, mnmos1 = 6u
x_inv2 in o2 vdd3 Inverter Mult=1 $ mpmos1 = 5u, mnmos1 = 4u
. . .
```

Example in 'local' mode

```
subckt opamp (n1 n2)
parameters ku=2, r=5k, v=5
ends opamp
```

Ends statement

After subcircuit description specify ends statement. This statement indicates end of the description.

Syntax in 'local' or 'spectre' modes

ENDS <SubcircuitName>

Syntax in 'spice' mode

.ENDS <SubcircuitName>

or

.EOM <SubcircuitName>

Examples of Subcircuit Test

1) *.macro or2_not in1 in2 out*

m_nmos1 out n2 gnd! gnd! nmos l=1e-6 w=1e-5

m_nmos2 out n3 gnd! gnd! nmos l=1e-6 w=1e-5

m_pmos1 out n2 n6 g_vdd! pmos l=1e-6 w=1e-5

m_pmos2 n6 n3 g_vdd! g_vdd! pmos l=1e-6 w=1e-5

.ends

The above example describes two subcircuits: and2_not and or2_not. They are called with the x_block1 and x_block2 statements.
Figures below show the subcircuit and the scheme to which this subcircuit is connected.

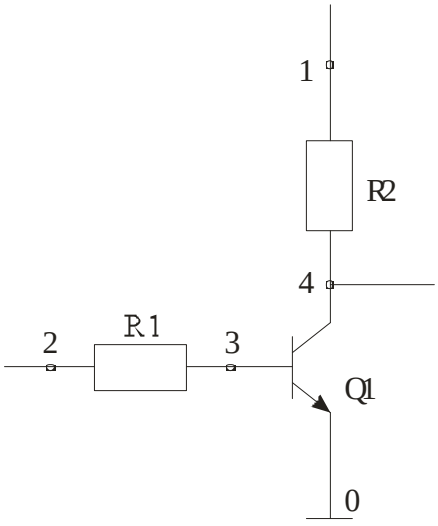


Figure 25. Subcircuit "rtl"

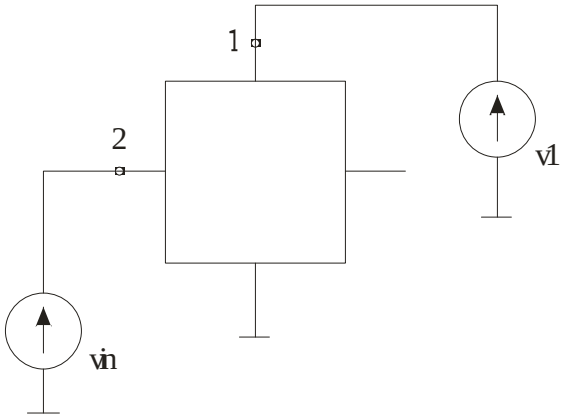


Figure 26. Scheme for simulation

2) *EXA*

```
.dc vin 0 5 0.01
vin 2 0 dc 0
v1 1 0 dc 5
x1 1 2 3 rtl
.subckt rtl 1 2 4
q1 4 3 0 0 MODT
r1 2 3 10k
r2 1 4 1k
.model MODT NPN(bf=100 vaf=200 ise=1.e-15 br=.1 rb=200
+ rc=10 cje=2p vje=.6 tf=.1n cjc=2p vjc=.5 tr=10n cjs=2p)
.ends
.end
```

This example specifies the subcircuit rtl that contains two resistors (r1 and r2) and the bipolar transistor (q1). It is connected to the general scheme with the statement x1.

3) *subckt d_trigger (D C Ra Q nQ)*

```
x_and21 D  C R      and2_not
x_and22 R  C S      and2_not
x_and31 Ra R nQ Q and3_not
x_and23 Q  S nQ    and2_not
ends d_trigger
```

LIB

In SPICE format netlist you can place commonly used commands, device models, subcircuit analysis and statements in library files by using the '.LIB' call statement. As each '.LIB' call name is encountered in the main data file, the corresponding entry is read in from the designated library file. The entry is read in until an '.ENDL' statement is encountered.

LIB library call statement

Syntax in 'local' or 'spectre' modes

LIB '<filepath> filename' entryname

Syntax in 'spice' mode

.LIB '<filepath> filename' entryname

where:

Parameter name	Description
filepath	Path to a file. Used where a computer supports tree structured directories. When the LIB file (or alias) resides in the same directory in which SymSpice is run, no directory path need to be specified; the netlist runs on any machine. You can use the "../" syntax in the filepath to designate the parent directory of the current directory.
filename	Name of a file to include in the data file. The combination of filepath plus filename may be up to 256 characters long, structured as any valid filename for the computer's operating system. File path and name must be enclosed in single or double quotation marks. You can use the "../" syntax in the filename to designate the parent directory of the current directory.
entryname	Entry name for the section of the library file to include. The first character of an entryname cannot be an integer.

Example

1) *lib 'C:\Spice\LIB\Qnom.lib' npn3*
2) *.lib 'MODELS' Cmod*

LIB library file definition statement

You can build libraries by using the 'LIB' statement in a library file. The 'LIB' statement begins the library macro, and the 'ENDL' statement ends the library macro.

lib entryname1
$ ANY VALID SET OF SymSpice STATEMENTS
endl entryname1

lib entryname2
$ ANY VALID SET OF SymSpice STATEMENTS
endl entryname2

lib entryname3
$ ANY VALID SET OF SymSpice STATEMENTS
endl entryname3

The text following a library file entry name must consist of valid SymSpice statements.

LIBnested library calls

Library calls may call other libraries, provided they are different files.

Example

.lib npn12

.lib 'one' npn11

.lib 'two' npn10

.lib 'one' npn12 $ This call is illegal within library npn12

.endl

Library calls are nested to any depth. This capability allows the construction of a sequence of model runs composed of similar components with different model parameters, without duplicating the entire SymSpice input file.

Library building rules

- A library may contain nested .lib calls to itself or other libraries. The depth of nested calls is only limited by the constraints of your system configuration.
- A library cannot contain a call to a library of its own entry name within the same

library file.

- A library cannot contain the .end statement.

The simulator accesses the models and skew parameters through the .lib statement and the .include statement. The library contains parameters that modify .model statements. The following example of a .lib of model skew parameters features both worst case and statistical distribution data.

Example in 'spice' mode

```
*** Library files "nmos.lib" ***
.lib aa
.model nmos_mod nmos lmin= 2.4e-07 lmax= 3.0e-07
+ wmin= 3.0e-07 wmax=4.0e-07
...
.endl aa


*** Cir-files "scheme.cir" ***
...
.lib "nmos.lib" aa
m_pmos_1 n_4 n_2 n_3 n_3 nmos_mod l=1u w=20u
+ ad=1e-12 as=1e-12 pd=4e-6 ps=4e-6
...
.end
```

Analyses

TRAN

This analysis computes the transient response of a circuit. The initial condition is taken to be the DC steady-state solution if not otherwise given.

Syntax in 'local' or 'spectre' modes

Name TRAN parameter= value ...

Syntax in 'spice' mode

.TRAN tincr1 tstop1 <tincr2 tstop2 ...tincrN tstopN>

+<START=start> <UIC>

or

.TRAN DATA=datanm

Keywords and parameters are defined as:

SPICE mode	SymSpice mode	Description
tincr1 ...	step	A time used in integration step calculation. Its behaviour may depend on simulation mode, tstop value and others. To explicitly control maximum value of integration step use 'maxstep' parameter.
See OPTIONS statement	maxstep	Specifies maximum value of integration step.
tstop1 ...	stop	Time when simulation stops.
start	start	Time when simulation starts.
UIC	ic	To set initial condition. For SymSpice netlist possible values are dc, node, dev, or all. For SPICE netlist uses the nodal voltages specified in the .IC statement (or by the “IC=” parameters in the various element statements) to calculate the initial transient conditions, rather than solving for the quiescent operating point.
DATA=datanm	-	Data name referred to in the '.TRAN' statement

Use UIC option	skipdc	If yes, there will be no dc analysis for transient. Possible values are no, yes, waveless, rampup, autodc, or sigrampup.
-	Iteratio	It is a criteria setting the local truncation error tolerances. To increase accuracy (conservative mode) set value to 1, to increase simulation speed (liberal mode) set value to 10. Default value (moderate mode) is 3.5. It can be set either in options or in analysis statement.
See OPTIONS statement	maxiters	Maximum number of Newton iterations at one time point, before taking a smaller time step. The typical value is 8.

Examples

1) The following example performs and prints the transient analysis every 5 ms for 500 ms for 'spice' mode

```
.TRAN 5ms 500ms
```

2) For 'local' mode previous example will be in following syntax:

```
nameSweep tran start=0.000 stop=500ms maxstep=5ms
```

Secondary sweep

Syntax in 'local' or 'spectre' modes

```
Name SWEEP parameter=value {
primary_sweep
}
```

Syntax in 'spice' mode

```
.TRAN PRIMARY_SWEEP SWEEP var pstart pstop pincr
or
.TRAN PRIMARY_SWEEP SWEEP var2 type np start2 stop2
or
.TRAN PRIMARY_SWEEP SWEEP DATA=datanm
```

For SymSpice format you can specify analyses statements. These statements should be bound within braces. The opening brace is required at the end of the line defining the

sweep. Sweep statements can be nested.

where:

SPICE mode	SymSpice mode	Description
PRIMARY_SWEEP	primary_sweep	For SymSpice, as primary_sweep parameter can be: ac, dc, tran. For SPICE, the parameter defined as PRIMARY_SWEEP parameters.
SWEEP	SWEEP	Keyword to indicate a second sweep is specified in the TRAN statement.
var ...	dev/mod/param	For SymSpice format the parameter is • dev — Device instance whose parameter value is to be swept • mod — Model whose parameter value is to be swept • param — Name of parameter to sweep For SPICE format is the name of an independent voltage or current source, any element value. However, if a parameter sweep, a .DATA statement, and a temperature sweep are specified, a parameter name must be chosen for the source value and subsequently referred to in the TRAN statement. The parameter name cannot begin with V or I.
TEMP	param=temp values=[...]	However, in 'local' mode you should use the next expression as parameter of secondary sweep analysis: param=temp values=[...] where: • param=temp — defines temperature sweep analysis as secondary sweep • values=[...] — array of sweep values If you use temperature as parameter (indicating a temperature sweep), then in 'spice' mode you can use TEMP parameter as var.
pstart	start	Starting voltage, current, temperature, any element value. Note: If type variation “POI” is used (list of points), a list of parameter values is specified instead of “pstart pstop”.
pstops	stop	Final voltage, current, temperature, any element value.
pincr	maxstep	Voltage, current, element parameter, or temperature increment value. Note: If “type” variation is used, the “np” (number of points) is specified instead of “pincr”.

np	-	Number of points or number of points per decade or octave, depending on the preceding keyword.
start2 ...	-	Time at which printing or plotting is to begin. The START keyword is optional: you can specify start time without preceding it with "Start=".
DATA=datanm	-	Data name referred to in the .TRAN statement.

Examples for 'spice' mode

Tran sweep analysis

.tran v_1 0 5 .1n .1n .1n 25n 50n sweep v_vdc_2 2 3 0 0.1n 0.1n 0.1n 10n 15n

Examples for 'local' mode

1) Temperature sweep

sweepsecond sweep param=temp values=[-75 50 75 100 300] {

timesweep tran start=0.000000e+000 stop=1.000000e-007 maxstep=1.000000e-009 oppoint=logfile

}

2) Tran analysis as secondary sweep and dc analysis as primary sweep

sweepsecond sweep dev=v_vds_2 start=0.000000e+000 stop=3.500000e+000 step=1.000000e-002 {

transweep tran start=0.000000e+000 stop=100n maxstep=1n

}

Simulation Accuracy Control for Transient and DC analyses

Numerical algorithms, used in SymSpice simulation program, don't generally require default values changes. These options (except **fast_spice**) are used for special case or for non-typical circuits. In order to adjust these options user has to be familiar with the numerical integration algorithms of nonlinear differential equations.

Acceleration Level

options: **fast_spice**

This option sets the acceleration mode for the circuit simulation.

- Level 0 – acceleration algorithms are disabled.
- Level 1 – acceleration algorithms are enabled (used by default).

DC

DC-analysis is the time independent type of analysis. It is defined by the amplitude value of the entering signal source.

Syntax in 'local' or 'spectre' modes

Name DC parameter=value ...

Syntax in 'spice' mode

Single-point:

.DC var1 START = start1 STOP = stop1 STEP = incr1

or

.DC var1 start1 stop1 incr1

or

.DC var1 type np start1 stop1

Data driven:

.DC DATA=datanm

where:

SPICE mode	SymSpice mode	Description
start1	start	Start sweep limit. It could be specified as a number, also as an expression.
stop1	stop	Stop sweep limit. It could be specified as a number, also as an expression.
incr1	step	Step size in linear sweep. It could be specified as a number, also as an expression.
var1	dev/mod/param	Name of an independent voltage or current source, any element or parameter. For SymSpice format the parameter is • dev — Device instance whose parameter value is to be swept • mod — Model whose parameter value is to be swept • param — Name of parameter to sweep

		For SPICE format the parameter name cannot begin with V or I.
type	dec/lin/log	For 'local' mode the parameter is • dec — number of steps, log sweep • lin — number of steps, linear sweep • log — points per decade For 'spice' mode it can be any of the following keywords: • DEC — decade variation • OCT — octave variation • LIN — linear variation • POI — list of points
np	-	Number of points per decade or per octave, or just number of points depending on the preceding keyword.
DATA=datanm	-	Datanm is the reference name of a '.DATA' statement.

Examples in 'spice' mode

1) Dc analysis by voltage v

```
.dc v start=0.0 stop=5.0 step=0.1
```

2) Dc analysis by current i. Current value varies from -1mA to 1mA, with step 0.01mA

```
.dc i -1m 1m 0.01m
```

Examples in 'local' mode

1) Dc analysis by voltage source

```
namesweep dc dev=v start=0.0 stop=1.0 step=1n
```

2) Dc analysis by temperature

```
dcSweep dc param=temp values=[0 15 25]
```

Primary sweep

Secondary sweep

Syntax in 'local' or 'spectre' modes

Name SWEEP parameter=value {

PRIMARY_SWEEP

}

Syntax in 'spice' mode

.DC PRIMARY_SWEEP SWEEP var2 type np start2 stop2

or

.DC PRIMARY_SWEEP SWEEP DATA=datanm

or

.DC var1 start1 stop1 incr1 <var2 start2 stop2 incr2>

For SymSpice format you can specify analyses statements. These statements should be bound within braces. The opening brace is required at the end of the line defining the sweep. Sweep statements can be nested.

where:

SPICE mode	SymSpice mode	Description
PRIMARY_SWEEP	primary_sweep	For SymSpice, as primary_sweep parameter uses any type of analysis: ac, dc, tran. For SPICE, the parameter defined as PRIMARY_SWEEP parameters.
SWEEP	sweep	Keyword to indicate a secondary sweep has different type of variation (DEC, OCT, LIN, POI, DATA statement).
var2 …	dev/mod/param	For SymSpice format the parameter is • dev — Device instance whose parameter value is to be swept • mod — Model whose parameter value is to be swept • param — Name of parameter to sweep For SPICE format the parameter is the name of an independent voltage or current source, any element or parameter. The parameter name cannot begin with V or I.

TEMP	param=temp values=[...]	However, in 'local' mode you should use the next expression as parameter of secondary sweep analysis: param=temp values=[...] where: • param=temp — defines temperature sweep analysis as secondary sweep • values=[...] — array of sweep values If you use temperature as parameter (indicating a temperature sweep), then in 'spice' mode you can use TEMP parameter as var.
type	dec/lin/log	For 'local' mode the parameter is • dec — number of steps, log sweep • lin — number of steps, linear sweep • log — points per decade For 'spice' mode it can be any of the following keywords: • DEC — decade variation • OCT — octave variation • LIN — linear variation • POI — list of points
np	-	Number of points per decade or per octave, or just number of points depending on the preceding keyword.
start2 ...	start	Start sweep limit.
stop2 ...	stop	Stop sweep limit.
incr2 ...	step	Step size, linear sweep.
DATA=datanm	-	Datanm is the reference name of a '.DATA' statement.
-	values=[...]	Array of sweep values

Examples for 'local' mode

Temperature sweep

```
sweepsecond sweep  param=temp values=[ -75 50 75 100 300] {
dcsweep dc dev=i_ipul1_1 start=0.0 stop=0.35m step=0.01e-6
}
```

Examples for 'spice' mode

1) The first example causes the value of the voltage source vin to be swept from 0.5 volts to 5.0 volts in increments of 0.25 volts.

```
.dc vin 0.5 5.0 0.25
```

2) In this example temperature specifying by list is varied.

```
.dc temp 0 27 1
```

3) The third example invokes a DC sweep of the parameter A from 7.5v to 12v in 0.5v increments.

```
.dc A 7.5v 12v .5v
```

4) Dc sweep analysis

```
.dc v_vdc2_4 dec 1.000000e+002 1.000000e-001 5.000000e+000 sweep v_vdc1_3
dec 10 1k 100k
```

Initial Condition

Initial Condition Statements

The IC statement is used to set transient initial conditions. How it initializes depends upon whether the UIC (IC) parameter is included in the tran analysis statement.
When the UIC(IC) parameter is specified in the tran statement, SymSpice does not calculate the initial DC operating point. In this case, the transient analysis is entered directly. The transient analysis uses the IC initialization values as part of the solution for timepoint zero (a fixed equivalent voltage source is applied during the calculation of the timepoint zero). The IC statement is equivalent to specifying the IC parameter on each element statement, but is more convenient.
You can still specify the IC parameter, but it does not take precedence over values set in the IC statement.
When the UIC(IC) parameter is not specified in the tran statement, the DC operating point solution is computed before the transient analysis. In this case, the node voltages specified in the IC statement are fixed for the determination of the DC operating point. For the transient analysis, the initialized nodes are released for the calculation of the second timepoint and later.

Syntax in 'local' or 'spectre' modes

IC node=value

Syntax in 'spice' mode

.IC V(node1) = val1 V(node2) = val2 ...

where:

Parameter name	Description
val1(value) ...	Specifies voltages.
node1 …	Node numbers or node names can include full path names or circuit numbers.

Example

1) For 'local' mode

ic 7=0 out=1 OpAmp1.comp=5 L1:1=1.0u

node 7 should start at 0V, node out should start at 1V, node comp in subcircuit OpAmp1 should start at 5V, and the current through the first terminal of L1 should start at 1uA.

2) For 'spice' mode

```
.IC V(11)=5 V(4)=-5 V(2)=2.2
```

Optimization

Optimization statements

Using Bisection
To use bisection, the following is required:
A user-specified pair of upper and lower boundary input variable values.
For a solution to be found, one of these values must result in an output variable result |goal value| and the other in a result < |goal value|
Specified goal value
Error tolerance value. The bisection process stops when the difference between successive test values = error tolerance. If the other criteria are met, see below.
Related variables. Variables must be related by a monotonic transfer function, where a steadily progressing time (increase or decrease) results in a single occurrence of the "goal" value at the "target" input variable value The error tolerance is included in a relation used as a process-termination criterion.
Note: Bisection search is applied to only one parameter

Syntax
To perform Bisection optimization using HSPICE compatibility syntax, the .PARAM, .TRAN, .AC, .DC, .MODEL and .MEASURE statements must be used.

.PARAM parname = OPTxxx(initval lower upper)
.MODEL modname OPT <METHOD = BISECTION | PASSFAIL>
+ <RELIN=val> <ITROPT=val>
Optimizing analyses are .TRAN, .AC and .MEASURE

initval: used for multidimensional optimizing
parname: Name of the Bisection optimization parameter.
OPTxxx: Optimization parameter reference name must agree with the OPTxxx name given in the .TRAN statement associated with the keyword OPTIMIZE.
lower: Left bound for bisection optimization parameter.
upper: Right bound for bisection optimization parameter.
MODEL: This keyword is followed by the optimization model.
modname: The model name. It is used by Bisection optimization to reference a particular model.
METHOD: Keyword to indicate which bisection optimization method to use.
BISECTION: The measure results for lower and upper bounds of optimization parameter, must be on opposite sides of goal value.
PASSFAIL: The measure must pass for one limit and fail for the other limit.

RELIN=val: Relative optimization parameter error tolerance. When the difference between the two latest test input values is smaller, then:
interval x RELIN
where
Interval = max(upper – lower),
Default is RELIN value is 0.001.
ITROPT=val: Maximum number of iterations. The Bisection optimization process will be terminated, when the number of iterations reaches. Default is NUMITER=20.

Optimizing analyses

.TRAN tstep tstop <tstart> <tmax> <UIC>
+ <CALLV> <SAVEV < =tsave>> <TRANOP < =top>> <STORE=num>
+ SWEEP OPTIMIZE = OPTxxx RESULT = measname MODEL=modname
.AC
.DC
.MEASURE TRAN measname ... <GOAL < |=| > val>

OPTIMIZE: This keyword is followed by the Bisection optimization parameter.
RESULT: This keyword is followed by the target for optimization.
measname: Name of the measure calculated by a .MEASURE statement.
modname: The model name. It is used by Bisection optimization to reference a particular model.

Example:

```
.PARAM _frequency = 10Meg
.PARAM _capacity   = Opt1 (100p, 50p, 120p)
.MODEL OptMod Opt
+ Method = bisection

c_cap1_0 n_2 n_1 c= 1e-011
c_cap2_1 gnd n_2 c=_capacity
l_l1_2 gnd n_2 l=2.5e-006 r=1e-6
r_res1_3 gnd n_2 r = 20k
v_vsin1_4 n_1 gnd sin 0 1 _frequency ac=1

.measure ac  maxVout MAX v(n_2)
.measure ac freq when v(n_2) = maxVout goal = 10Meg $_frequency

.ac dec 1000 5Meg  12Meg  sweep
```

```
+ OPTIMIZE = Opt1
+ RESULTS  = freq $ Look at measure
+ MODEL    = OptMod
```

AC Small Signal Analysis

The AC analysis linearizes the circuit about the DC operating point and computes the response to a given small sinusoidal stimulus.

Single-point analysis

Syntax in 'local' or 'spectre' modes

Name AC parameter= value ...

Syntax in 'spice' mode

.AC type np fstart fstop

or

.AC var1 START=start1 STOP=stop1 STEP=incr1

or

.AC DATA=datanm

where:

Parameter name	SymSpice compatible	Description
type	dec/lin/log	For 'local' mode the parameter is • dec — number of steps, log sweep • lin — number of steps, linear sweep • log — points per decade For 'spice' mode it can be any of the following keywords: • DEC – decade variation • OCT – octave variation • LIN – linear variation • POI – list of points
np	-	Number of points per decade or per octave, or just number of points, depending on the preceding keyword.
var1	dev/mod/param/freq	For 'local' format the parameter is • dev — Device instance whose parameter value is to be swept • mod — Model whose parameter value is to be swept • param — Name of parameter to sweep

		• freq - You may use a frequency as name of independent source. Name of an independent voltage or current source, any element parameter. For 'spice' format the parameter name cannot begin with V or I.
start1	start	Start sweep limit.
stop1	stop	Stop sweep limit.
incr1	step	Step size, linear sweep.
-	values=[...]	Array of sweep values
fstart	-	Starting frequency. Note: If type variation "Poi" (list of points) is used, a list of frequency values is specified instead of "fstart fstop".
fstop	-	Final frequency.
DATA=datanm	-	Data name referred to in the '.AC' statement.

Examples

1) For SymSpice format

nameSweep ac dev=i lin start=1 stop=1MEG step=1K

2) For SPICE format

.ac vIn dec 100 1 1MEG

Secondary sweep

Syntax in 'local' or 'spectre' modes

Name SWEEP parameter=value {

PRIMARY_SWEEP

}

Syntax in 'spice' mode

.AC PRIMARY_SWEEP SWEEP var start stop incr

or

.AC PRIMARY_SWEEP SWEEP var type np start stop

or

.AC PRIMARY_SWEEP SWEEP DATA=datanm

For SymSpice format you can specify analyses statements. These statements should be bound within braces. The opening brace is required at the end of the line defining the sweep. Sweep statements can be nested.

Keywords and parameters are defined as:

SPICE mode	SymSpice mode	Description
PRIMARY_SWEEP	primary_sweep	For SymSpice, as primary_sweep parameter uses any type of analysis: ac, dc, tran. For SPICE, the parameter defined as PRIMARY_SWEEP parameters.
SWEEP	sweep	Keyword to indicate a second sweep is specified in the .AC statement.
var	dev/mod/param	For SymSpice format the parameter is • dev — Device instance whose parameter value is to be swept • mod — Model whose parameter value is to be swept • param — Name of parameter to sweep For SPICE is the name of an independent voltage or current source, any element parameter. The parameter name cannot begin with V or I.
TEMP	param=temp values=[...]	However, in 'local' mode you should use the next expression as parameter of secondary sweep analysis: param=temp values=[...] where: • param=temp — defines temperature sweep analysis as secondary sweep • values=[...] — array of sweep values If you use temperature as parameter (indicating a temperature sweep), then in 'spice' mode you can use TEMP parameter as var.
start	start	Starting voltage, current, any element value.
stop	stop	Final voltage, current, any element value.

incr	step	Voltage, current, element increment value. Note: If "type" variation is used, the "np" (number of points) is specified instead of "incr".
type	dec/lin/log	For 'local' mode the parameter is • dec — number of steps, log sweep • lin — number of steps, linear sweep • log — points per decade For 'spice' mode it can be any of the following keywords: • Dec – decade variation • Oct – octave variation • Lin – linear variation • Poi – list of points
np	-	Number of points per decade or per octave, or just number of points, depending on the preceding keyword.
DATA=datanm	-	Data name referred to in the '.AC' statement.

Examples for 'local' mode

1) Ac parametric sweep analysis and tran analysis as the primary sweep analysis

```
transweep sweep dev=i_igs_1 start=0 stop=100n maxstep=1n {
acsweep ac start=1.000000e+003 stop=1.000000e+009 dec=5.000000e+001
}
```

2) Temperature sweep

```
sweepsecond sweep  param=temp values=[ -75 50 75 100 300] {
acsweep ac start=1.000000e+001 stop=1.000000e+009 dec=5.000000e+001
}
```

Examples for 'spice' mode

1) The following example calls for a 21 point frequency sweep from 2 Hz to 1000 Hz.

```
.AC lin 21 2 1000Hz
```

2) The example below performs a frequency sweep by 10 points per decade from 1 Hz to 300 kHz.

```
.AC DEC 10 1 300kHz
```

3) The following example performs an AC analysis for each value of r12, which results from a linear sweep of r12 between 1k and 5k (20 points), sweeping frequency by 10

points per decade from 1 Hz to 300 kHz.

```
.AC DEC 10 1 300kHz sweep r12 lin 20 1k 5k
```

4)The last example increases the temperature by 10°C through the range -50°C to 20°C and performs AC analysis for each temperature.

```
.AC DEC 10 1 300kHz sweep temp –50 20 10
```

For an AC analysis, at least one independent AC source element statement must be in the data file (for example, V1 INPUT GND AC 1V). SymSpice checks for this condition and reports a fatal error if no such AC sources have been specified.

S-Parameter Analysis

The S-parameter analysis is performed for input files in 'local' mode. To perform S-Parameter analysis of the circuit means to compute the S-parameters of the nodes.

S-Parameter

Voltage and current are measured using so-called scattering parameters (S-parameters). S-parameters completely describe the behavior of a linear device. The individual parameters, dimensionless, complex numbers normally expressed as magnitude and phase.
S-parameters are computed for circuit nodes, defined as ports. A possibility for performing this is the following: each step computes S-parameters of each port provided they are stimulated sequentially.

For a 2-port network:
S_{n1n2} , where

n1- output port, where response is taken,

n2- input port which is stimulated.

S_{11}: input reflection coefficient (response of the port1, when it is stimulated).
S_{21}: forward transmission coefficient (gain) (response of the port2, when prt1 is stimulated).
S_{12}: reverse transmission coefficient (isolation) (response of the port1, when port2 is stimulated).
S_{22}: output reflection coefficient (response of the port2, when it is stimulated).

The S-parameters are measured by sending a signal (per frequency selected) into the device and detecting what returns as a response. The output (and other ports, if any) of the device is/are terminated with a 50-ohm resistor (Z=50 Ohm) during the measurement.
The S-parameters are presented as a table of numbers.

Definition

The S-parameter analysis linearizes the circuit about the DC operating point and computes S-parameters of the circuit taken as an N-port. The port statements define the ports of the circuit. Each active port is turned on one after one, and a linear small-signal analysis is performed. SymSpice converts the response of the circuit at each active port into S-parameters and outputs these parameters. There must be at least one active port

statement in the circuit.
If a filename is specified using the file parameter, the S-parameter analysis generates an ASCII file containing the S-parameters of the circuit that can later be read-in by the nport component. The output file is in SymSpice format.
SymSpice can perform the analysis while sweeping a parameter. The parameter can be frequency, temperature, component instance parameter, component model parameter, or global parameter. If changing a parameter leads to the DC operating point changing, the operating point is recomputed on each step. You can sweep the circuit temperature by giving the parameter name as temp with no dev or mod parameter. You can sweep a netlist parameter by giving the parameter name with no dev, or mod parameter. After the analysis has completed, the modified parameter returns to its original value.

SP Statement

Syntax in 'local' or 'spectre' modes

Primary sweep:

Name sp parameter=value ...

Secondary sweep:

Name sweep parameter=value ...

Keywords and parameters are defined as:

Sweep interval parameters

Parameter name	Description
start	Start sweep limit.
stop	Stop sweep limit.
step	Step size, linear sweep.
lin	Number of steps, linear sweep.
dec	Points per decade.
log	Number of steps, log sweep.
values=[...]	Array of sweep values.

Sweep variable parameters

Parameter name	Description
dev	Device instance whose parameter value is to be swept.

param	Name of parameter to sweep (temp, freq, global parameter).
freq (Hz)	Frequency when parameter other than frequency is being swept.

Output parameters

Parameter name	Description
file="..."	S-parameters output file name.

Note: If the list of active ports is specified with the ports parameter, then the ports are numbered from one to one in the order given. Otherwise, all ports present in the circuit are active, and the port numbers used are those that were assigned on the port statements.

Steps can be linear or logarithmic, and the number of steps can be specified or the size of each step. To determine type of sweep (linear or logarithmic) you can give a step size parameter (step, lin, log, dec). If a step size parameter is not given, the sweep is linear when the ratio of stop to start values is less than 10, and logarithmic when this ratio is 10 or greater. All frequencies are in Hertz.
The small-signal analysis begins by linearizing the circuit about an operating-point. By default this analysis computes the operating-point if it is not known, or recomputes it if any significant component or circuit parameter has changed.

Example

The following example performs S-Parameter analysis for RC-circuit. Temperature is a parameter to be swept when frequency=1 MHz. Two active ports are numbered.

```
output_file_name sp start=0.100000e+000 stop=1.000000e+009 lin=100
+ ports=[port0 port1] file="output_file_name.csd" param=temp freq=1meg
```

The simulation results are represented in .csd file:

```
reference resistance
                                port2 = 50.000000
                                port1 = 50.000000

format temp:  s1:1(real,imag)          s2:1(real,imag)
```

s1:2(real,imag) s2:2(real,imag)

.sp file is generated automatically:

```
...
TRACE
"s11" "S-Param"
"s21" "S-Param"
"s12" "S-Param"
"s22" "S-Param"
VALUE
"temp" 0.100000
"s11" (1.000000 -6.28319e-13)
"s21" (4.34263e-24 6.28319e-13)
"s12" (4.34263e-24 6.28319e-13)
"s22" (1.00000 -6.28319e-13)
"temp" 1.00000e+07
"s11" (1.000000 -6.28319e-13)
"s21" (4.34263e-24 6.28319e-13)
"s12" (4.34263e-24 6.28319e-13)
"s22" (1.00000 -6.28319e-13)
...
```

Ports

Description

A port is a resistive source that is tied between pos and neg. It is equivalent to a voltage source in series with a resistor, and the reference resistance of the port is the value of the resistor. The DC value given for the port voltage specifies the DC voltage across the port when it is terminated in its reference resistance (in other words, the DC voltage of the internal voltage source is double the user specified DC value, dc). The same is true for the values for the transient and AC signals of the port.
While generally useful as a stimulus in high frequency circuits, the port has three unique capabilities. First, it acts to define the ports of the circuit to the S-parameter analysis. Second, it has an intrinsic noise source, and so allows the noise analysis to directly compute the noise figure of the circuit. And finally, it is the only source for

which the amplitude can be specified in terms of power.

Sample Instance Statement

r_res1_0 n_1 in default_1 r=1000
c_cap1_1 out n_1 default_2 c=1e-012
port0 (in gnd) port num=2 type=dc
port1 (out gnd) port num=1 type=dc

Syntax in 'local' or 'spectre' modes

Name p n port parameter=value ...

Parameters are the following:

Parameter name	Unit	Description
num	-	Port number.
dc	V	DC value.
ac	V	AC voltage or power source value.
r	Ohm	Reference resistance.
phase	degrees	Small signal phase.
rdc	Ohm	Series resistance (overrides r) for DC analysis.
rttran	Ohm	Series resistance (overrides r) for DC analysis.

Simulation output

Specifying Output Data Format

There are two ways for specifying output data format.

1. Using netlist options.
For specifying output data format with netlist options see "SymSpice Netlist Options".

2. Through command line.

symspice <input_netlist> -format <apb/csdf/csv/none>

The default behavior:
If the netlist base format is SymSpice, then the output file format is APB.
If the netlist base format is SPICE, then no output file will be generated.

For more information see "Command Line Input".

Note:
1. The format specified in command line overwrites the format specified in the netlist.
2. If more than one output file format is specified, either in the command line or in the netlist options, only one output file will be generated.

Output Commands

SymSpice output statements are .probe (or save, for 'local' format netlist) and .measure. Statements are included in the input netlist file.

Output Statement	Description
.probe (SPICE netlist) save (SymSpice netlist)	Outputs data to post-processor output files but not to the output listing (used with option probe to limit output).
.measure (only in 'spice' mode)	Prints to output listing file the results of specific user-defined analyses.

Output Variables

The output format statements require special output variables to probe analysis results for nodal voltages and branch currents. There are three groups of output variables: DC and transient analysis, AC analysis and MEASURE statement.
Dc and Transient analysis displays individual nodal voltages, branch currents, and element power dissipation.
AC analysis displays imaginary and real components of a nodal voltage or branch current, as well as the phase of a nodal voltage or branch current.
The MEASURE statement variables are user-defined. They represent the electrical specifications measured in a MEASURE statement analysis.
The following table shows all the output variables, associated with DC, TRAN, or AC. For input netlists in SPICE and SymSpice formats the sets of output variables are different.

Input Netlist Format	Analysis Type	
	DC and TRAN	AC
SPICE	nodal voltage output current output (voltage sources) current output (element branches) measure user-defined variables	nodal voltage output current output (independent voltage sources) current output (element branches) group time delay output measure user-defined variables
SymSpice	nodal voltage output current output (voltage sources) current output (element branches)	Pair of numbers is always on the output: real and complex representation of viewpoint

DC and Transient Analysis

DC and transient analysis displays:

- Voltage differences between specified nodes (one specified node and ground).
- Current output for an independent voltage source.
- Current output for any element.

Nodal Voltage Output

Syntax in 'local' or 'spectre' modes

Name options save=value

SAVE node

Syntax in 'spice' mode

.PROBE <DC | TRAN> V (n1)

where:

Parameter name	Description
n1	Defines the voltage difference between n1 and ground (node 0) is given. If omitted, all nodal voltages will be outputted
value	Possible values all, selected or none. By default, SymSpice use all
save	The save statement indicates that the values of specific nodes should be saved in the output file Note: if save=all, all nodal voltages will be outputted.
node	List of nodes to be saved

Example in 'local' mode

saveoptions options save=selected
save n_1

Example in 'spice' mode

.probe v(n_1) i(m_m1_0)

Outputs voltage in node n_1.

Current Output: Element Branches

Syntax in 'local' or 'spectre' modes

Name options currents=value

SAVE compName:currents

Syntax in 'spice' mode

.PROBE <DC | TRAN> In (Wwww)

where:

Parameter name	Description
n	Node position number in the element statement. For example, if the element contains four nodes, I3 denotes the branch current output for the third node; if n is not specified, the first node is assumed
Wwww	Element name. If the element is within a subcircuit, its current output is accessed by appending a dot and the subcircuit name to the element name, for example, I3(xl.mos1_1). If omitted, all currents through elements branches will be outputted
value	Possible values all, nonlinear or selected. By default, SymSpice use selected
compName	The netlist name of a component whose signals you want to save
currents	Reserved word indicated that currents through elements branches will be saved in the output file. Note: if currents=all, all currents through elements branches will be outputted.

Example in 'local' mode

save m1:currents m2:1 m3:b
saveoptions options currents=selected
Outputs all currents of the MOSFET m1, drain current of m2 and bulk current of m3.

Examples in 'spice' mode

.probe I1(R12)
This example specifies the current through the first node of resistor R12.
.probe I2(XSUB.Q1)
The above example specifies the current through the second node (the base node) of the bipolar transistor Q1, which is defined in subcircuit XSUB.
.probe I3(PMOS3)

The last example specifies the current through the third node (the source node) of the MOSFET PMOS3.

Branch current direction for the elements in the figures below is defined in terms of arrow notation (current direction) and node position number (terminal type).

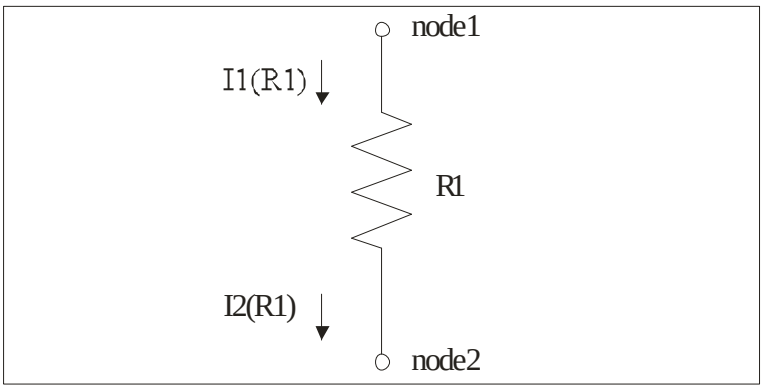


Figure 35. Resistor (node1, node2)

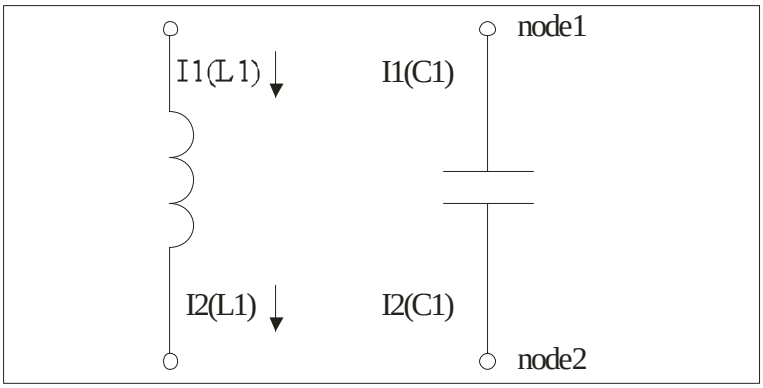


Figure 36. Capacitor (node1, node2); Inductor (node 1, node2)

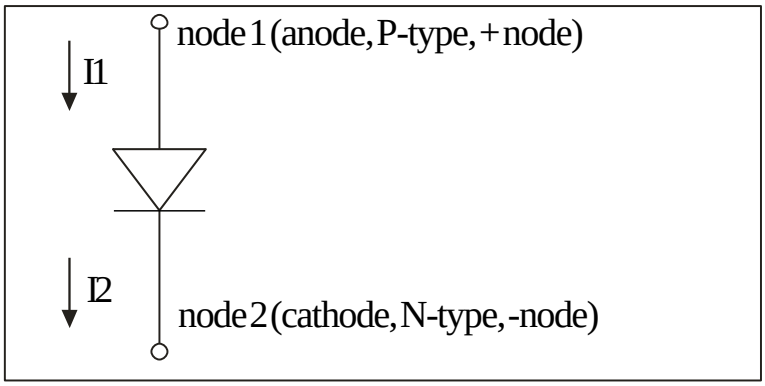


Figure 37. Diode (node1, node2)

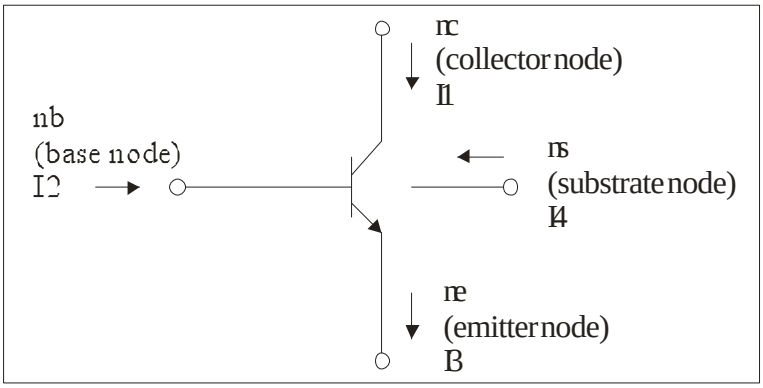


Figure 38. BJT (node1, node2, node3, node4) – npn

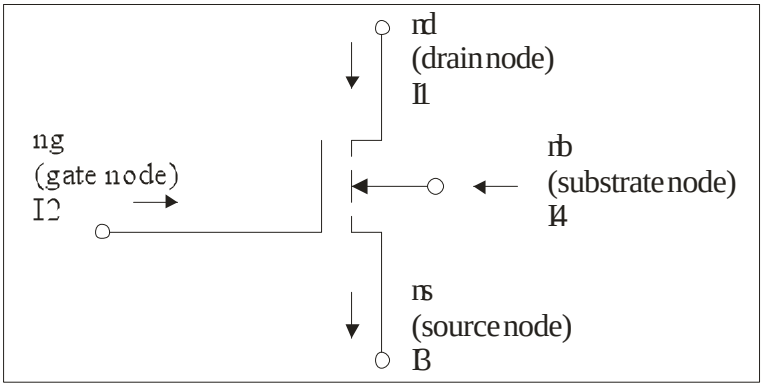


Figure 39. MOSFET (node1, node2, node3, node4) – n-channel

Current Output: Voltage Sources

Syntax in 'local' or 'spectre' modes

SAVE Vxxx:pin_name

Syntax in 'spice' mode

.PROBE <DC | TRAN> I(Vxxx)

where:

Parameter name	Desctription
Vxxx	Voltage source element name. If an independent power supply is within a subcircuit, its current output is accessed by appending a dot and the subcircuit name to the element name, for example, I(X1.Vxxx)
pin_name	index or name of pin

Example in 'local' mode

save v1:1 v2:p

saveoptions options currents=selected

Examples in 'spice' mode

.probe i(v1)

AC Analysis

In case of SPICE input netlist, AC analysis displays:

- Voltage differences between specified nodes (one specified node and ground).
- Element branch current.

To specify real or imaginary parts, magnitude, phase, decibels, and group delay for voltages and currents the following AC output variable types can be used.

AC Output Variable Types Table

Type Symbol	Variable Type
DB	decibel
I	imaginary part
M	magnitude
P	phase
R	real part
T	group delay

The output variable name is formed by variable symbol and the type symbol appended to it. For example, VM is the magnitude part of the voltage, or IDB is the decibel of the current.

Note: In 'local' mode AC analysis result will be pair of values (real part, imaginary part) of outputted Viewpoint.
However, if frequency response or phase response characteristics should be outputted, you can insert section describing SPICE-prefixes (db, m or p) to the symspice-netlist.

Example

```
simulator lang=local
r1 1 0 resistor
...
simulator lanng=spice
.PROBE VDB(1)
```

Nodal Voltage Output

Syntax in 'local' or 'spectre' modes

Name options save=value

save node

Syntax in 'spice' mode

.PROBE Vx(n1)

where:

Parameter name	Description
x	Specifies the voltage output type (see AC Output Variable Types Table)
n1	Defines the voltage difference between n1 and ground (node 0) is given. If omitted, all nodal voltages will be outputted
value	Possible values all, selected or none. By default, SymSpice use all
save	The save statement indicates that the values of specific nodes should be saved in the output file. Note: if save=all, all nodal voltages will be outputted.
node	List of nodes to be saved

Examples for SPICE format

1) *.PROBE AC VT(10) VT(2,25) IT(RL)*
2) *.PROBE AC IT1(Q1) IT3(M15) IT(D1)*
3) *.probe vdb(n_2)*

Outputs the decibel of the voltage between the n_2 node and the ground.

Example for SymSpice format

simulator lang=local

r 1 2 resistor 1k

.....

simulator lang=spice

.probe vt(1)

Current Output: Element Branches

Syntax in 'local' or 'spectre' modes

Name options currents=value

save compName:currents

Syntax in 'spice' mode

.PROBE Izn (Wwww)

where:

Parameter name	Description
z	Current output type (see AC Output Variable Types Table)
n	Node position number in the element statement. For example, if the element contains four nodes, IP2 denotes the phase of the branch current output for the second node.
Wwww	Element name. If the element is within a subcircuit, its current output is accessed by appending a dot and the subcircuit name to the element name, for example, IP2(subciruit1.Wwww).
value	possible values *all*, *nonlinear* or *selected*. By default, SymSpice use *selected*
compName	is the netlist name of a element whose signals you want to save
currents	reserved word indicated that currents through elements branches will be saved in the output file. Note: if currents=all, all currents through elements branches will be outputted.

Example

1. Outputs the magnitude of the current through resistor in subcircuit x_block and the phase of the current through the second pin of the resistor res1.
Current source is a two pins element. Therefore in this case: ip(res1:2), current can be outputted through the both pins of resistor: ip(res1:1) and ip(res1:2).

probe im1(x_block1.res1) ip(res1:2)

2. Outputs the real part of the current through the MOSFET m1 within subcircuit x_block1 and the real part of the current through the current source isin1.
Element pin with index 1 is taken by default.

probe ir2(x_block1.m1) ir(isin1)

Group Time Delay Output Equations

The group time delay, TD, is associated with AC analysis and is defined as the negative derivative of phase, in radians, with respect to radian frequency. In SPICE, the difference method is used to compute TD, as follows

$$TD = -\frac{1}{360} \cdot \frac{(\text{phase2} - \text{phase1})}{(\text{f2} - \text{f1})}$$

where phase1 and phase2 are the phases, in degrees, of the specified signal at the frequencies f1 and f2, in Hertz.

Current Output: Independent Voltage Sources

Syntax in 'local' or 'spectre' modes

Syntax in 'spice' mode

.PROBE Iz(Vxxx)

where:

Parameter name	Description
z	Current output type (see AC Output Variable Types Table)
Vxxx	Voltage source element name. If an independent power supply is within a subcircuit, its current output is accessed by appending a dot and the subcircuit name to the element name, for example, IP(subcircuit1.Vxxx)

Note: SymSpice doesn't support current output of input voltage sources.

Example

.probe IP(x_block1.V1)

Outputs the phase of the current through V1, which is within a subcircuit x_block1.

Part II

SymSpice Models

SymSpice Models Compatibility

Model	SymSpice	SPICE
Passive devices		
Resistor	V	V
Physical Resistor	V	-
Capacitor	V	V
Inductor	V	V
Diodes		
Level 1	V	V
Level 2	V	V
Level 3	V	V
Juncap (spice level=4)	V	+
Juncap200 (spice level=6)	V	+
Transistors		
BJT Gummel Poon	V	V
HICUM 2.1	V	V
JFET (spice level=1, 2, 3)	V	+
MEXTRAM (bjt504, bjt504t, bjtd504, bjtd504t)	V	+
MOSFET MOS1, MOS3 (spice level=1,3)	V	V
MOSFET BSIM3v3.2 (spice level=49, 53)	V	V
MOSFET BSIM4 , BSIM4.5.0 (spice level=54)	V	V
MOSFET BSIM3-SOI (spice level=59, 60)	V	V
MOSFET EKV 2.6 (spice level=55)	V	+
MOSFET EKV3 (Electronics Laboratory–TUC) (ekv3rf, ekv3nqs, ekv3r4, ekv3s)	V	+
MOSFET HiSIM HV version 1.2.2	V	+

(spice level=73, 172)		
MOSFET PSP (psp102e, pspnqs102e, psp1020, pspnqs1020) (spice level=69)	V	V
VBIC 1.2	V	+
Independent Sources		
PULSE	V	V
SIN	V	V
EXP	V	V
PWL	V	V
SFFM	V	V
AM	V	V
Dependent Sources		
VCVS	V	V
CCCS	V	V
VCCS	V	V
CCVS	V	V

Denotement:
'V' - the model is supported and compatible
'+' - the model is supported and implemented according to standard description
'-' - model isn't supported

Passive devices

Resistors

Resistor is a two-pins passive element defined by the value of resistance.
The element can be described by two types of parameters: individual and group parameters.
Individual parameters are used for description of element.
Element's group parameters are defined in its model specified by the model statement.

Syntax in 'local' or 'spectre' modes

without model:

Name (n1 n2) resistor parameter=value ...

with model:

Name (n1 n2) modelname parameter=value ...

model modelname resistor pname=val

Syntax in 'spice' mode

Rxxx n1 n2 <mname>
<R=>resistance

+<<TC1=>val> <<TC2=>val>

+<SCALE=val> +<M=val> <AC=val>

+ <DTEMP=val> <L=val>

+<W=val> <C=val>

where:

SPICE mode	SymSpice mode	Default	Description
Rxxx	Name	-	Resistor element name. For SPICE netlist it must begin with "R", which can be followed by alphanumeric characters.
n1	n1	-	Positive terminal node name.

n2	n2	-	Negative terminal node name.
mname	resistor/model name	-	Resistor model name. This name is used in elements to reference a resistor model.
R	r	-	Resistance value at room temperature. This may be a numeric value or parameter in Ohms or expression.
TC1	tc1	-	First order temperature coefficient for the resistor.
TC2	tc2	-	Second order temperature coefficient for the resistor.
SCALE	scale	-	Element scale parameter; scales resistance by its value. Default=1.0.
M	m	-	Multiplier used to simulate parallel resistors. Default=1.0.
AC	rac	-	AC resistance used in the AC analysis. Default=Reff.
DTEMP	trise	0.0	Temperature difference between the element and the circuit in Celsius. Default=0.0.
L	l	-	Resistor length in meters. Default=0.0, if L is not specified in a resistor model.
W	w	-	Resistor width. Default=0.0, if W is not specified in the model.
C	c	-	Capacitance connected from node n2 to bulk. Default=0.0, if C is not specified in a resistor model.
-	scaler	-	Resistance scaling factor.
-	scalec	-	Capacitance scaling factor.
-	tc1c	-	First-order temperature coefficient for capacitance
-	tc2c	-	Second-order temperature coefficient for capacitance

Wire Model (Group) Parameters

SPICE mode	SymSpice mode	Units	Default	Description
cap	c	F	0.0	Default capacitance
capsw	cjsw	F/m	0.0	Sidewall fringing capacitance
cox	cj	F/m^2	0.0	Bottomwall capacitance
di	di		0.0	Relative dielectric constant
dlr(dl)	etchl	m	0.0	Difference between drawn length and actual length (for resistance calculation only). For capacitance calculation, dw is used DLReff=dlr*scalm.
dw	etch	m	0.0	Difference between drawn width and actual width DWeff=dw*scalm
l	l	m	0.0	Default length of wire lscaled=l*shrink*scalm
rac	rac	Ohm		Ac resistance (RACeff default is reff)
res	r	Ohm	0.0	Default resistance
rsh	rsh	Ohm/sqr	0.0	Sheet resistance/square
shrink	shrink		1.0	Shrink factor
tc1c	tc1c	1/C°	0.0	First-order temperature coefficient for capacitance
tc2c	tc2c	1/C°2	0.0	Second-order temperature coefficient for capacitance
tc1r	tc1	1/C°	0.0	First-order temperature coefficient for resistance
tc2r	tc2	1/C°2	0.0	Second-order temperature coefficient for resistance
thick	thick	m	0.0	Dielectric thickness
tref (tnom)	tref (tnom)	C°	Tnom	Temperature reference for model parameters
w	w	m	0.0	Default width of wire wscaled=w*shrink*scalm
coeffs	coeffs	-	-	Vector of polynomial resistance coefficients
-	scaler		-	Resistance scaling factor

-	scalec		-	Capacitance scaling factor

Resistor Model Equations

Resistance Calculation

The element width and length are scaled by the option Scale and the model parameter Shrink. The model width and length are scaled by the option Scalm and the model parameter Shrink.

The effective width and length are calculated as follows:

$$Weff = Wscaled - 2 \cdot DWeff$$

$$Leff = Lscaled - 2 \cdot DLReff$$

If element resistance is specified:

$$Reff = \frac{R \cdot Scale(elem\ ent)}{M}$$

Otherwise, if $Weff \cdot Leff \cdot Rsh > 0$, then:

$$Reff = \frac{Leff \cdot Rsh \cdot Scale(elem\ ent)}{M \cdot Weff}$$

If $Weff \cdot Leff \cdot Rsh = 0$, then:

$$Reff = \frac{Res \cdot Scale(elem\ ent)}{M}$$

If Ac resistance is specified in the element, then:

$$RACeff = \frac{Ac \cdot Scale(elem\ ent)}{M}$$

Otherwise, if Rac is specified in the model, Rac is used:

$$RACeff = \frac{Rac \cdot Scale(elem\ ent)}{M}$$

If neither are specified, it defaults to:

$$RACeff = Reff$$

Resistor Temperature Equations

The resistor and capacitor values are modified by temperature values as follows:

$$R(T) = R \cdot (1.0 + Tc1 \cdot \Delta t + Tc2 \cdot \Delta t^2)$$

$$Rac(T) = Rac \cdot (1.0 + Tc1 \cdot \Delta t + Tc2 \cdot \Delta t^2)$$

Δt	$t - tnom$
t	Element temperature in °K: $t = circuttemp + \Delta temp + 273.15$
tnom	Nominal temperature in °K: $tnom = 273.15 + Tnom$

Note: Circuit temperature is default set by options.

Physical Resistor (phy_res)

Physical resistor model is netlist supported for input netlist in SymSpice format.

Syntax in 'local' or 'spectre' modes

Name (n1 n2 <n3>) modelname parameter=value ...

model modelname phy_res pname=val

Syntax in 'spice' mode

model phy_res isn't supported in 'spice' mode

A physical resistor consists of a linear resistor (tied between n1 and n2) and two junction diodes (tied between n1-n3 and n2-n3). By default, these two diodes are reverse biased. The direction of the diodes an be reversed by the parameter SUBTYPE. If you do not specify n3, ground is assumed.

The arguments are defined as:

Parameter name	Default	Description
modelname	-	Resistor model name. This name is used in elements to reference a resistor model.
r	-	Resistance value at room temperature. This may be a numeric value or parameter in Ohms.
c	-	Linear capacitance.
l	-	Line length.
w	-	Line width.
tc1	0.0	First order temperature coefficient for the physical resistor.
tc2	0.0	Second order temperature coefficient for the physical resistor.
tc1c	0.0	First order temperature coefficient for linear capacitor.
tc2c	0.0	Second order temperature coefficient for linear capacitor.
trise	0.0	Temperature rise from ambient.
m	1.0	Multiplicity factor.

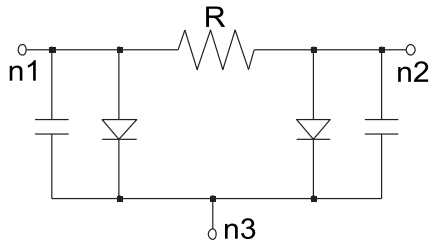


Figure 27. Physical resistor equivalent circuit, Subtype = n

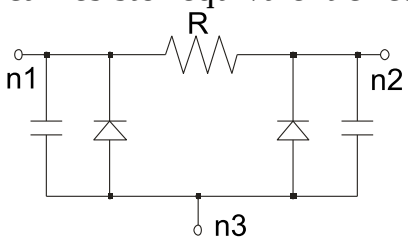


Figure 28. Physical resistor equivalent circuit, Subtype = p

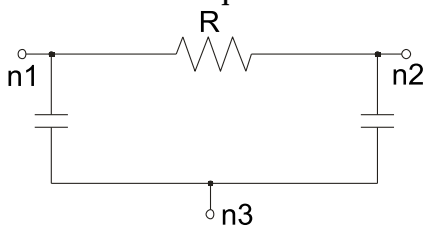


Figure 29. Physical resistor equivalent circuit, Subtype = poly

Substrate Type Parameters

Parameter name	Units	Default	Description
subtype		p	Substrate type. Possible values are n, p or poly.

Resistance Parameters

Parameter name	Units	Default	Description
res	Ohm	¥	Default resistance
rsh	Ohm/square	¥	Sheet resistance
minr	Ohm	0.1	Minimum resistance
coeffs		[…]	Vector of polynomial conductance coefficients.
polyarg		diff	Polynomial model argument type. Possible values are sum or diff.

Temperature Effects Parameters

Parameter name	Units	Default	Description
tc1	1/°C	0.0	First order temperature coefficient for resistor
tc2	°C-2	0.0	Second order temperature coefficient for resistor
tc1c	1/°C	0.0	First order temperature coefficient for linear capacitor
tc2c	°C-2	0.0	Second order temperature coefficient for linear capacitor
tnom	°C		Parameters measurement temperature. Default set by options.
trise	°C	0.0	Temperature rise from ambient

Junction Capacitance Model Parameters

Parameter name	Units	Default	Description
c	F	0.0	Default linear capacitance
cj	F/m2	0.0	Zero-bias junction bottom capacitance density
cjsw	F/m	0.0	Zero-bias junction sidewall capacitance density
mj		0.5	Junction bottom grading coefficient
mjsw		0.33	Junction sidewall grading coefficient
pb	V	0.8	Junction bottom built-in potential
pbsw	V	0.8	Junction sidewall built-in potential
fc		0.5	Junction bottom capacitor forward-bias threshold
fcsw		0.5	Junction sidewall capacitor forward-bias threshold
tt	s	0.0	Transit time

Junction diode model parameters

Parameter name	Units	Default	Description
is	A		Saturation current
js	A/m^2	0.0	Saturation current density
nn		1.0	Emission coefficient
eg	V	1.11	Band gap
xti		3.0	Saturation current temperature exponent

imelt	A	imax	Explosion current, diode is linearized beyond this current to aid convergence.
jmelt	A/m^2	jmax	Explosion current density, diode is linearized beyond this current to aid convergence.
imax	A	1.0	Maximum current, current above this limit generate a warning
jmax	A/m^2	1e8	Maximum current density, currents above this limit generate a warning
dskip		yes	Use simple piecewise linear model for diode currents below 0.1 iabstol. Possible values are no or yes.
bvj	V	¥	Junction reverse breakdown voltage

Device Size Parameters

Parameter name	Units	Default	Description
l	m	¥	Default line length
w	m	1e-6	Default line width
etch	m	0	Narrowing due to etching
etchl	m	0	Length reduction due to etching
etchc	m	Etch	Narrowing due to etching for capacitances
etchlc	m	Etchl	Length reduction due to etching for capacitances
scaler		1.0	Resistance scaling factor
scalec		1.0	Capacitance scaling factor

Noise model parameters

Parameter name	Units	Default	Description
kf	p	0.0	Flicker noise coefficient
af	p	1.0	Flicker noise exponent

The W and L parameters are scaled by the option parameters Scale and Scalem. If you set Is=0 or Subtype=poly, Physical Resistor will not contain diodes. If parameters Mj and Mjsw are set to zero or Subtype=poly the capacitance is linear, but is still calculated from the device geometry.

Physical Resistor Device Equations

Physical resistor was designed to provide for compatibility with the 'local' mode.
If R(element) is not given and R(model) is given,
R(element) = R(model)
Otherwise,

$$R(element) = Rsh * \frac{(L - 2*Etchl)}{(W - 2*Etch)}$$

The resistance as a function of temperature is given by

$$R(T) = R(Tnom) * [1 + Tc1 * (T - Tnom) + Tc2 * (T - Tnom)^2]$$

where

$$T = Trise(elem\ ent) + Temp$$

if Trise(element) is given.

$$T = Trise(mode\ l) + Temp$$

otherwise.

If you do not specify the junction leakage current (Is) and Js is specified, the leakage current is calculated from Js and the device dimensions:

$$Is = Js * 0.5 * (L - 2*Etchl) * (W - 2*Etch)$$

If you specify the instance capacitance or the linear model capacitance, linear capacitors are used between n1-n3 and n2-n3 (Figure 29). Otherwise, nonlinear junction capacitors are used and the zero bias capacitance values are calculated from the model parameters.
If C(element) is not given and C(model) is given,
C(element) = C(model)
Otherwise,

$$C(element) = 0.5 * Cj * (L - 2*Etchlc) * (W - 2*Etchc)$$
$$+ Cjsw * (W + L - 2*Etchc - 2*Etchlc)$$

If the capacitance is nonlinear, the temperature model for the junction capacitance is used. Otherwise, the following equation is used.

$$C(T) = C(Tnom) * [1 + Tc1c * (T - Tnom) + Tc2c * (T - Tnom)^2]$$

Capacitors

Capacitor is a passive component defined by the value of capacity.

Syntax in 'local' or 'spectre' modes
without model:

Name (n1 n2) capacitor parameter=value ...

with model:

Name (n1 n2) modelmame parameter=value ...

model modelname capacitor pname=val

Syntax for Polynomial Form

Name (n1 n2) modelmame coeffs=[c0 c1..] parameter=value

Syntax in 'spice' mode

Cxxx n1 n2 <mname> <C=>capacitance

+<<TC1=>val> <<TC2=>val>

+<SCALE=val> <IC=val> <M=val>

+<W=val> <L=val> <DTEMP=val>

or

Cxxx n1 n2 <C='equation'

+<CTYPE=val><<TC1=>val> <<TC2=>val>

+ <SCALE=val> <IC=val> <M=val>

+ <W=val> <L=val> <DTEMP=val>

Syntax for Polynomial Form

Cxxx n1 n2 POLY c0 c1...

+<<TC1=>val> <<TC2=>val>

+ <SCALE=val> <IC=val> <M=val>

+ <W=val> <L=val> <DTEMP=val>

The arguments are defined as:

SPICE mode	SymSpice mode	Default	Description
Cxxx	Name	-	Capacitor element name. For SPICE netlist it must begin with a "C", which can be followed by alphanumeric characters.
n1	n1	-	Positive terminal node name.
n2	n2	-	Negative terminal node name.
mname	capacitor/modelname	-	Capacitor model name. This name is used in elements to reference a capacitor model.
C	c	-	Capacitance at room temperature as a numeric value or parameter in Farads or expression.
TC1	tc1	-	First order temperature coefficient for the capacitor.
TC2	tc2	-	Second order temperature coefficient for the capacitor.
SCALE	scale	-	Element scale parameter, scales capacitance by its value. Default=1.0.
M	m	1.0	Multiplier used to simulate multiple parallel capacitors. Defaults 1.0.
W	w	-	Capacitor width in meters. Default=0.0, if W is not specified in a capacitor model.
L	l	-	Capacitor length in meters. Default=0.0, if L is not specified in a capacitor model.
DTEMP	trise	0.0	Element temperature difference with respect to the circuit temperature in Celsius. Default=0.0.
POLY	-		Keyword to specify capacitance given by a polynomial.
c0 c1...	coeffs=[...]		Coefficients of a polynomial in voltage describing the capacitor value. c0 represents the magnitude of the 0th order term, c1 represents the magnitude of the 1st order term, and so on.
-	area	-	capacitor area.
-	perim	-	capacitor perimeter.
-	scalec	-	Capacitance scaling factor.

Capacitance Parameters

SPICE mode	SymSpice mode	Units	Default	Description
cap	cap	F		Default capacitance value
capsw	cjsw	F/m	0.0	Sidewall fringing capacitance
cox	cj	F/m^2		Bottomwall capacitance
del	etch	m	0.0	Difference between drawn width and actual width or length DELeff = del*scalm
di	di		0.0	Relative dielectric constant
l	l		0.0	Default length of capacitor lscaled = l*shrink*scalm
shrink	shrink		1	Shrink factor
tc1	tc1	1/C°	0.0	First temperature coefficient for capacitance
tc2	tc2	1/C°2	0.0	Second temperature coefficient for capacitance
thick	thick	m	0.0	Insulator thickness
tref(tnom)	tref(tnom)	C°		Reference temperature
w	w	m	0.0	Default width of capacitor wscaled = w*shrink*scalm
-	scalec	F/m^2		Capacitance scaling factor.
-	coeffs		v	Vector of polynomial capacitance coefficients

Capacitor Device Equations

SPICE formulas for computation the capacitance are the followings.

Effective Capacitance Calculation

The option Scale and the model parameter Shrink scale the element width and length. The option Scalm and the model parameter Shrink scale the model width and length. The effective width and length are calculated as follows:

$$Weff = Wscaled - 2 \cdot DELeff$$

$$Leff = Lscaled - 2 \cdot DELeff$$

If the element capacitance *C* is specified:

$$CAPeff = C \cdot Scale(elem\ ent) \cdot M$$

Otherwise, the capacitance is calculated from the Leff, Weff and Cox.

$$CAPeff = M \cdot Scale(elem\ ent) \cdot [Leff \cdot Weff \cdot Cox + 2 \cdot (Leff + Weff) \cdot Capsw]$$

If Cox is not specified, but Thick is not zero, then:

$$Cox = \frac{\Delta i \cdot \varepsilon_o}{Thick}$$ – if Δi not zero

or

$$Cox = \frac{\varepsilon_{ox}}{Thick}$$ – if Δi = 0

where

ε_o = 8.8542149e-12 F/m

ε_{ox} = 3.453148e-11 F/m

If only model capacitance Cap is specified, then

$$CAPeff = Cap \cdot Scale(elem\ ent) \cdot M$$

Capacitance Temperature Equations

The capacitance as a function of temperature is calculated as follows:

$$C(t) = C \cdot (1.0 + Tc1 \cdot \Delta t + Tc2 \cdot \Delta t^2)$$

Δt	$t - tnom$
t	Element temperature in °K: $t = circuttemp + \Delta temp + 273.15$
tnom	Nominal temperature in °K: $tnom = 273.15 + Tnom$

For SymSpice compatible we used the follow formulas:

Description

You can assign the capacitance or let SymSpice compute it from the physical length and width of the capacitor. In either case, the capacitance can be a function of temperature or applied voltage.

This device is supported within altergroups.

If the C(inst) is not given, C(inst) = C(model)

if C(model) is given and , if Area(inst) or Perim(inst) is given

$$Areaeff = Area - (Perim) \cdot Etch + 4 \cdot Etch^2$$

$$Perimeff = Perim - 8 \cdot Etch$$

else

$$Areaeff = (L - 2 \cdot Etch) \cdot (W - 2 \cdot Etch)$$

$$Perimeff = 2 \cdot (W + L - 4 \cdot Etch)$$

$$C(inst) = Cj \cdot Areaeff + Cjsw \cdot Perimeff$$

if C(model) is not given.

If the polynomial coefficients vector (coeffs=[c1 c2 ...]) is specified, the capacitor is nonlinear and the capacitance is

$$C(V) = dQ(V)/dV = C(inst) \cdot (1 + c1 \cdot V + c2 \cdot V^2 + ...)$$

or

$$Q(V) = C(inst) \cdot V \cdot (1 + 1/2 \cdot c1 \cdot V + 1/3 \cdot c2 \cdot V^2 + ...)$$

where ck is the kth entry in the coefficient vector.

The value of the capacitor as a function of the temperature is given by:

$$C(T) = C(Tnom) \cdot [1 + Tc1 \cdot (T - Tnom) + Tc2 \cdot (T - Tnom)^2]$$

where

T = Trise(inst) + temp

if Trise(inst) is given, and

T = Trise(model) + temp

if Trise(inst) is not given.

Inductors

Inductor is a passive component defined by the value of inductance. To calculate the value of inductance it is necessary to calculate the magnetic field produced by inductor.

Syntax in 'local' or 'spectre' modes

without model:

Name (n1 n2) inductor parameter=value ...

with model:

Name (n1 n2) modelname parameter=value ...

model modelname inductor pname=val

Syntax in 'spice' mode

Lxxx n1 n2 <L=>inductance

+<<TC1=>val> <<TC2=>val>

+<SCALE=val> <IC=val> <M=val>

+<DTEMP=val> <R=val>

or

Lxxx n1 n2 L='equation'

+<LTYPE=val> <<TC1=>val> <<TC2=>val>

+ <SCALE=val> <IC=val> <M=val>

+ <DTEMP=val> <R=val>

Syntax for Polynomial Form

Lxxx n1 n2 POLY c0 c1...

+<<TC1=>val> <<TC2=>val>

+ <SCALE=val> <IC=val> <M=val>

+<DTEMP=val> <R=val>

Syntax for Magnetic Winding form

Lxxx n1 n2 NT=turns

+<<TC1=>val> <<TC2=>val>

+ <SCALE=val> <IC=val> <M=val>

+<DTEMP=val> <R=val>

The arguments are defined as:

SPICE mode	SymSpice mode	Default	Description
Lxxx	Name	-	Inductor element name. For SPICE netlist it must begin with "L", which can be followed by alphanumeric characters.
n1	n1	-	Positive terminal node name.
n2	n2	-	Negative terminal node name.
-	inductor/modellname	-	Inductor model name. This name is used in elements to reference an inductor model.
Tc1	tc1	0.0	First order temperature coefficient for the inductor.
Tc2	tc2	0.0	Second order temperature coefficient for the inductor.
SCALE	scale	-	Element scale parameter; scales inductance by its value. Default=1.0.
L	l	-	Inductance value. This may be a numeric value or parameter in Henries, or expression.
M	m	1.0	Multiplier used to simulate parallel inductors. Default=1.0.
DTEMP	-	0.0	Temperature difference between the element and the circuit in Celsius. Default=0.0.
R	r	0.0	Resistance of inductor in Ohms. Default=0.0.
LTYPE	-	0	Determines inductance flux calculation for elements with inductance equations.
POLY	-	-	Keyword to specify inductance given by a polynomial.
NT	-	-	Number representing the number of turns of an inductive magnetic winding.

c0 c1...	-	-	Coefficients of a polynomial in voltage describing the inductor value. c0 represents the magnitude of the 0th order term, c1 represents the magnitude of the 1st order term, and so on.

Inductor Core Model Parameters

Parameter name	Units	Default	Description
l	H	0	Default inductance
r	Ohm	0	Default resistance
tc1	1/C°	0	Linear temperature coefficient
tc2	C°-2	0	Quadratic temperature coefficient
trise	C°	0	Default 'trise' value for instance
tnom	C°	-	Parameters measurement temperature
rforce	Ohm2	1e9	Resistance used when forcing nodesets and initial conditions
coeffs	-	-	Vector of polynomial inductance coefficients
skalei	-	1	Inductance scaling factor
kf	-	0	Flicker (1/f) noise coefficient
af	-	2	Flicker (1/f) noise exponent

Inductor Device Equations

SPICE expressions for calculating inductance.
Inductance Temperature Equations

The effective inductance as a function of temperature is provided by the following equation:

$$L(t) = L \cdot (1.0 + Tc1 \cdot \Delta t + Tc2 \cdot \Delta t^2)$$

Δt	$t - tnom$
t	Element temperature in °K: $t = circuittemp + \Delta temp + 273.15$
tnom	Nominal temperature in °K: $tnom = 273.15 + Tnom$

Create coupling between inductors with a separate coupling element. Specify mutual inductance between two inductors by the coefficient of coupling, kvalue,

defined by the equation:

$$K = \frac{M}{(L_1 \cdot L_2)^{1/2}}$$

L1, L2 the inductances of the two coupled inductors
M the mutual inductance between the inductors

Linear branch relation for transient analysis:

$$v_1 = L_1 \cdot \frac{di_1}{dt} + M \cdot \frac{di_2}{dt}$$

$$v_2 = M \cdot \frac{di_1}{dt} + L_2 \cdot \frac{di_2}{dt}$$

Linear branch relation for AC analysis:

$$V_1 = (j \cdot \omega \cdot L_1) \cdot I_1 + (j \cdot \omega \cdot M) \cdot I_2$$

$$V_2 = (j \cdot \omega \cdot M) \cdot I_1 + (j \cdot \omega \cdot L_2) \cdot I_2$$

Note: You must define an inductor reference by a mutual inductor statement; otherwise SPICE displays an error message and terminates.

SymSpice compatible equations

If the polynomial coefficients vector (coeffs=[c1 c2 ...]) is specified, the inductor is nonlinear and the inductance is

$$L(I) = L(inst) \cdot (1 + c1 \cdot I + c2 \cdot I^2 + ...)$$

The branch flux as a function of current is

$$Flux(I) = L(inst) \cdot I \cdot (1 + \frac{1}{2} \cdot c1 \cdot I + \frac{1}{3} \cdot c2 \cdot I^2 + ...)$$

where ck is the kth entry in the coefficient vector.
The value of the inductor as a function of the temperature is given by:

$$L(T) = L(tnom) \cdot [1 + tc1 \cdot (T - tnom) + tc2 \cdot (T - tnom)^2]$$

where

$$T = trise(inst) + temp$$

if trise(inst) is given, and

$$T = trise(model) + temp$$

otherwise.

Active devices

Diodes

Syntax in 'local' or 'spectre' modes

without model:

Name (nplus nminus) diode parameter=value ...

with model:

Name (nplus nminus) modelname parameter=value ...

model modelname diode pname=val

Syntax in 'spice' mode

Dxxx nplus nminus mname

+<<AREA=>area> <<PJ=>val>

+<WP=val> <LP=val> <WM=val>

+<LM=val> <OFF> <IC=vd>

+<M=val> <DTEMP=val>

or

Dxxx nplus nminus mname

+<W=width> <L=length> <WP=val>

+ <LP=val> <WM=val> <LM=val>

+<OFF> <IC=vd> <M=val> <DTEMP=val>

or

Dxxx nplus nminus mname

+<W=val <L=val>> <WP=val> <M=val>

The arguments are as follows:

SPICE mode	SymSpice mode	Default	Description

Dxxx	Name	-	Diode element name. For SPICE format netlist it must begin with "D", which can be followed by alphanumeric characters.
nplus	nplus	-	Positive terminal (anode) node name. The series resistor of the equivalent circuit is attached to this terminal.
nminus	nminus	-	Negative terminal (cathode) node name.
mname	diode/modelname	-	Diode model name reference.
AREA	area	1.0	Area of the diode (square meters for diode model level=3). This affects saturation currents, capacitances and resistances (diode model parameters Ik, Ikr, Js, Cjo and Rs). Default=1.0. Overrides Area from the diode model. If unspecified, is calculated from width and length specifications.
PJ	perim	0.0	Periphery of junction (meters for diode model level=3). Overrides PJ from the diode model. If unspecified, calculated from the width and length specifications.
WP	wp	0.0	Width of polysilicon capacitor in meters (for diode model level=3 only). Overrides Wp in diode model. Default=0.0.
LP	lp	0.0	Length of polysilicon capacitor in meters (for diode model level=3 only). Overrides LPpin diode model. Default=0.0.
WM	wm	0.0	Width of metal capacitor in meters (for diode model level=3 only). Overrides Wm in diode model. Default=0.0.
LM	lm	0.0	Width of metal capacitor in meters (for diode model level=3 only). Overrides Lm in diode model. Default=0.0.
M	m	1.0	Multiplier to simulate multiple diodes in parallel. All currents, capacitances and resistances are affected by the setting of M. Default=1.0.
DTEMP	trise	0.0	The difference between the element temperature and the circuit temperature in Celsius. Default=0.0.
W	w	0.0	Width of the diode in meters (diode model level=3 only).
L	l	0.0	Length of the diode in meters (diode model level=3 only).

Using Diodes

Diode models are used to describe p–n junction diodes within MOS and bipolar integrated circuit environments and discrete devices. Three types of diode models are available in SymSpice:

- Zener diodes
- Silicon diffused junction diodes
- Schottky barrier diodes

Diode Types

Geometric junction diode is used to model IC-based standard silicon diffused diodes, Schottky barrier diodes, and Zener diodes. The geometric parameter lets you specify p–n junction poly and metal capacitance dimensions for a particular IC process technology.

Nongeometric junction diode is used to model discrete diode devices such as standard and Zener diodes. The nongeometric model lets you scale currents, resistances, and capacitances using dimensionless area parameters.

Diode Model Statements

Use model and element statements to select the diode models. Select the type of diode model specifying Level parameter:

- Level=1 selects the nongeometric junction diode model
- Level=3 selects the geometric junction diode model

You can design Zener, Schottky barrier, and silicon diffused diodes by altering model parameters for both Level=1 and Level=3. For Zener diodes, the BV parameter is set for an appropriate Zener breakdown voltage.If Level parameter is not specified, the model defaults to the nongeometric junction diode model, Level=1.

Junction Diode Models

Diode element parameters include: temperature, geometric junction, and capacitance parameters. Both Level=1 and Level=3 junction diode models share the same element parameter set. Poly and metal capacitor parameters of Lm, Lp, Wm and Wp do not share the same element parameter.

Geometric Scaling for Diode Models

Level=1 Scaling

Scaling for Level=1 involves the use of the Area and M element parameters. These parameters are not affected by the Scale and Scalm options.

equations

$$PJeff = Pj \cdot M$$

$$CJPeff = PJeff \cdot Cjp$$
$$JSWeff = PJeff \cdot Jsw$$

$$AREAeff = Area \cdot M$$

$$IKeff = AREAeff \cdot Ik$$
$$IKReff = AREAeff \cdot Ikr$$
$$IBVeff = AREAeff \cdot Ibv$$
$$ISeff = AREAeff \cdot Is$$
$$RSeff = \frac{Rs}{AREAeff}$$
$$CJOeff = AREAeff \cdot Cjo$$

LEVEL=3 Scaling

The element parameters affected by Scale include:
Area, Lm, Lp, Pj, Wm, Wp, W, L.
The model parameters affected by Scalm include:
Area, Lm, Lp, Pj, Wm, Wp, W, L, Ibv, Ik, Ikr, Is, Jsw, Rs, Cjo, Cjp, Xm, Xp, Xw.

equations

$$AREAeff = Area \cdot M \cdot Scale^2 \cdot Shrink^2$$
$$PJeff = PJ \cdot Scale \cdot M \cdot Shrink$$

or, if W and L are specified:

$$AREAeff = Weff \cdot Leff \cdot M$$

$$PJeff = (2 \cdot Weff + 2 \cdot Leff) \cdot M$$

where

$$Weff = W \cdot Scale \cdot Shrink + XWeff$$
$$Leff = L \cdot Scale \cdot Shrink + XWeff$$

$$JSWeff = PJeff \cdot \frac{Jsw}{Scalm}$$
$$CJPeff = PJeff \cdot \frac{Cjp}{Scalm}$$

$$LMeff = Lm \cdot Scale \cdot Shrink$$
$$WMeff = Wm \cdot Scale \cdot Shrink$$
$$LPeff = Lp \cdot Scale \cdot Shrink$$
$$WPeff = Wp \cdot Scale \cdot Shrink$$
$$XPeff = Xp \cdot Scalm$$
$$XMeff = Xm \cdot Scalm$$

$$IKeff = AREAeff \cdot Ik$$
$$IKReff = AREAeff \cdot Ikr$$
$$IBVeff = \frac{AREAeff \cdot Ibv}{Scalm^2}$$
$$ISeff = \frac{AREAeff \cdot Is}{Scalm^2}$$
$$RSeff = \frac{Rs}{AREAeff \cdot Scalm^2}$$
$$CJOeff = \frac{AREAeff \cdot Cjo}{Scalm^2}$$

Junction Model (Group) Parameters

The model (group) parameters specify the type of diode model used (Level=1 or 3), as well as DC, capacitance, temperature, resistance and geometric parameters.

Junction Capacitance Parameters

SPICE mode	SymSpice mode	Units	Default	Description

cj(cjo,cja,cj0)	cj(cjo,cja,cj0)	F/[area]	0.0	Zero-bias junction capacitance per unit area
cjp(cjsw)	cjp(cjsw)	F/[pj]	0.0	Zero-bias junction capacitance per unit perimeter
fc	fc		0.5	Forward-bias depletion capacitance coefficient for area
fcs	fcsw		0.5	Forward-bias depletion capacitance coefficient for perimeter
mj(exa)	mj(exa)		0.5	Area junction grading coefficient
mjsw(exp)	mjsw(exp)		0.33	Periphery junction grading coefficient
pb(phi,vj,pha)	vj	V		Area junction contact potential
php	vjsw	V		Periphery junction contact potential
tt	tt	s	0.0	Transit time
tm1	tm1	1/C°	0.0	First-order temperature coefficient for mj
tm2	tm2	1/C°2	0.0	Second-order temperature coefficient for mj
tpb(tvj)	pta	V/C°	0.0	Temperature coefficient for pb
tphp	ptp	V/C°	0.0	Temperature coefficient for php
tref(tnom)	tref(tnom)	C°	Tnom	Model reference temperature
trs	trs	1/C°	0.0	Resistance temperature coefficient
ttt1	ttt1	1/C°	0.0	First order temperature coefficient for tt
ttt2	ttt2	$1/C^{o2}$	0.0	Second order temperature coefficient for tt
xti	xti		3.0	Saturation current temperature exponent
ns	ns			Emission coefficient for side-wall diode
cd	cd	F		Linear capacitance
rsw	rsw	Ohm		Sidewall resistance
imax	imax	A	1	Maximum current
gleak	gleak	1/Ohm		
gleaksw	gleaksw	1/Ohm		
minr	minr	Ohm		Mimimum resistance
nz	nz		1	Emission coefficient for Zener diode

Metal and Poly Capacitor Parameters for Level=3

Parameter name	Units	Default	Description
Lm	m	0.0	Length of metal capacitor
Lp	m	0.0	Length of poly capacitor
Wm	m	0.0	Width of metal capacitor
Wp	m	0.0	Width of poly capacitor
Xm	m	0.0	Accounts for masking and etching effects
Xoi	1e-10 m	1E4	Thickness of the poly to bulk oxide
Xom	1e-10 m	1E4	Thickness of the metal to bulk oxide
Xp	m	0.0	Accounts for masking and etching effects

Temperature Effects on Junction Diodes

Level=1 and Level=3 model statements contain parameters for the calculation of temperature effects. Tlev and Tlevc select different temperature equations for the calculation of temperature effects on energy gap, leakage current, breakdown voltage, contact potential, junction capacitance, and grading.

Temperature Effect Parameters (Level=1 and 3)

Parameter name	Units	Default	Description
cta(ctc)	1/C°	0.0	Temp coefficient for cjo
ctp	1/C°	0.0	Temp coefficient for cjp
eg	eV	1.11	Energy gap for p-n junction diode
gap1	eV/C°	7.02E-4	Band gap temp coeff
gap2	C°	1108	Band gap offset
tcv	1/C°	0.0	Breakdown voltage temperature coefficient
tlev			Temperature equation selector for diode
tlevc			Level selector for diode temperarure equations, junction capacitances and contact potentials

Junction DC equations

For SPICE format netlist SymSpice uses the following expressions.

The basic diode is modeled in three regions:

- Forward bias
- Reverse bias
- Breakdown region

Forward Bias: $vd > -10 \cdot vt$

$$id = ISeff \cdot \left(e^{\frac{vd}{N \cdot vt}} - 1 \right)$$

$$vd = v_{node1} - v_{node2}$$

Reverse Bias: $BVeff < vd < -10 \cdot vt$

$$id = -ISeff$$

For breakdown, the parameter *Bv (Vb)* is set, inducing reverse breakdown or avalanche. This effect is seen in Zener diodes and occurs when the anode – cathode voltage is less than *Bv*. Model this action by measuring the voltage *(Bv)* and the current *(Ibv)* at the reverse knee or onset of avalanche.

Note: *Bv* always described as a positive number.

Breakdown: $vd < -BVeff$

$$id = -ISeff \cdot e^{-\left(\frac{vd + BVeff}{N \cdot vt} \right)}$$

The *BV* parameter is adjusted as follows to obtain *BVeff*:

$$ibreak = -ISeff \cdot \left(e^{\frac{-BV}{N \cdot vt}} - 1 \right)$$

If $IBVeff > ibreak$, then

$$BVeff = Bv - N \cdot vt \cdot \ln\left(\frac{IBVeff}{ibreak} \right)$$

Otherwise,

$$IBVeff = ibreak$$

Most diodes do not behave as ideal diodes. The parameters *Ik* and *Ikr* are called high-level injection parameters. They tend to limit the exponential current increase.

Note: The exponential equation is used in both the forward and reverse regions.

Forward Bias:

$$id = \frac{id1}{1+\sqrt{\frac{id1}{IKeff}}}$$

Reverse Bias:

$$id = \frac{id1}{1+\sqrt{\frac{id1}{IKReff}}}$$

where id1 is

For : $vd \geq -BVeff$

$$id1 = ISeff \cdot (e^{\frac{vd}{N \cdot vt}} - 1)$$

Otherwise:

$$id1 = ISeff \cdot (e^{\frac{vd}{N \cdot vt}} - 1) - ISeff \cdot (e^{\frac{(vd+BVeff)}{N \cdot vt}} - 1)$$

You can estimate the reverse saturation current *Is*, emission coefficient *N*, and model parameter *Rs* from DC measurements of the forward biased diode characteristics. You can determine *N* from the slope of the diode characteristic in the ideal region. In most cases, the emission coefficient is the value of unit, but is closer to 2 for MOS diodes. In practice, at higher levels of bias, the diode current deviates from the ideal exponential characteristic. This deviation is due to the presence of Ohmic resistance in the diode as well as high-level injection effects. The deviation of the actual diode voltage from the ideal exponential characteristic at a specific current determines the value of *Rs*. In practice, *Rs* is estimated at several values of *id* and averaged, since the value of Rs depends upon diode current.

For SymSpice compatible SymSpice uses the following expressions.

Level-1 Model

The following effects are included in the junction diode model: forward characteristics, reverse leakage current, breakdown, parasitic resistance, diffusion capacitance, depletion capacitance, and overlap capacitance.

To achieve better scaling, a sidewall (peripheral) diode can also be specified. The peripheral diode shares most model parameters with the main diode except for the following parameters: isw, ns, cjsw, mjsw, fcsw, rsw, gleaksw, ctp, and ptp. If the sidewall parasitic resistance (rsw) is not specified, the peripheral diode shares the same parasitic resistance with the main diode. That is, both the main and the peripheral diodes are connected to the same internal nodes. If rsw is specified, the peripheral diode creates its own internal node. In this case, the peripheral and main diode are equivalent to connecting two diodes in parallel.

The leakage currents are modeled by putting two small conductances (gleak and gleaksw) in parallel with the intrinsic diodes.

DC Current

$$I_{jtot} = I_j + I_{jsw}$$

where V_t is the thermal voltage given by

$$I_j = \begin{cases} is\left(e^{\frac{V}{nV_t}} - 1\right) & \text{if } V \le V_{Expl} \\ I_{offset} + G_{Expl}V & \text{otherwise} \end{cases}$$

$$V_t = \frac{kT}{q}$$

$$V_{Expl} = nV_t \ln\left[1 + \frac{imelt}{is}\right]$$

is the forward explosion voltage,

$$G_{Expl} = \frac{(imelt + is)}{nV_t}$$

is the conductance at V_{Expl}, and

$$I_{offset} = imelt - V_{Expl}G_{Expl}$$

is the current linearly extrapolated to V = 0 from V_{Expl}.

$$I_{jsw} = \begin{cases} isw\left(e^{\frac{V}{n_sV_t}} - 1\right) & \text{if } V \le V_{Explsw} \\ I_{offsetsw} + G_{Explsw}V & \text{otherwise} \end{cases}$$

$$V_{Explsw} = n_sV_t \ln[1 + \frac{imelt}{is}]$$

$$G_{Explsw} = \frac{(imelt + isw)}{n_sV_t}$$

$$I_{offsetsw} = imelt - V_{Explsw}G_{Explsw}$$

Diode Capacitance Equations

For SPICE format netlist SymSpice uses the following expressions.

The capacitance *cd* consists of diffusion capacitance (*cdiff*), depletion capacitance (*cdep*), metal (*cmetal*), and poly capacitances (*cpoly*).

$$cd = cdiff + cdep + cmetal + cpoly$$

Diffusion Capacitance Equations

The transit time (*Tt*) models the diffusion capacitance, caused by injected minority carriers. In practice, *Tt* is estimated from pulsed time-delay measurements.

$$cdiff = Tt \cdot \frac{\partial id}{\partial vd}$$

Depletion Capacitance Equations

The depletion capacitance is modeled by junction bottom and junction periphery

capacitances. The formula for both bottom area and periphery capacitances is similar, except each has its own model parameters. There are two equations for forward bias junction capacitance that are selected using *Dcap.*

Dcap=1

The junction bottom area capacitance formula is:

$$vd < Fc \cdot Pb$$

$$cdepa = CJeff \cdot (1 - \frac{vd}{Pb})^{-Mj}$$

$$vd \geq Fc \cdot Pb$$

$$cdepa = CJeff \cdot \frac{[1 - Fc \cdot (1 + Mj) + Mj \cdot \frac{vd}{Pb}]}{[1 - Fc]^{(1+Mj)}}$$

The junction periphery capacitance formula is:

$$vd < Fcs \cdot Php$$

$$cdepp = CJPeff \cdot (1 - \frac{vd}{Php})^{-Mjsw}$$

$$vd \geq Fcs \cdot Php$$

$$cdepp = CJPeff \cdot \frac{[1 - Fcs \cdot (1 + Mjsw) + Mjsw \cdot \frac{vd}{Php}]}{[1 - Fcs]^{(1+Mjsw)}}$$

then, $cdep = cdepa + cdepp$

Dcap=2 (default)

The total depletion capacitance formula is:

$$vd < 0$$

$$cdep = CJeff \cdot (1 - \frac{vd}{Pb})^{-Mj} + CJPeff \cdot (1 - \frac{vd}{Php})^{-Mjsw}$$

$$vd \geq 0$$

$$cdep = CJeff \cdot (1 + Mj \cdot \frac{vd}{Pb}) + CJPeff \cdot (1 + Mjsw \cdot \frac{vd}{Php})$$

Dcap=3

Limits peak depletion capacitance to $Fc \cdot CGDeff$ or $Fc \cdot CGSeff$, with proper fall-off when forward bias exceeds Pb ($Fc \geq 1$).

Metal and Poly Capacitance Equations (Level=3 Only)

To determine the metal and poly capacitances, use the equations:

$$\text{cmetal} = \frac{\varepsilon_{ox}}{\text{Xom}} \cdot (\text{WMeff} + \text{XMeff}) \cdot (\text{LMeff} + \text{XMeff}) \cdot \text{M}$$

$$\text{cpoly} = \frac{\varepsilon_{ox}}{\text{Xoi}} \cdot (\text{WPeff} + \text{XPeff}) \cdot (\text{LPeff} + \text{XPMeff}) \cdot \text{M}$$

For SymSpice **compatible SymSpice uses the following expressions.**

Level-1 Model

Junction Capacitance

$$C_j(V) = \begin{cases} \dfrac{\text{cjo}}{\left[1 - \dfrac{V}{\text{vj}}\right]^{m}} & \text{if } V \le \text{fc} \cdot \text{vj} \\ \dfrac{\text{cjo}}{(1-\text{fc})^{m}}\left[1 + \dfrac{\text{m}(V - \text{vj} \cdot \text{fc})}{\text{vj}(1-\text{fc})}\right] & \text{otherwise} \end{cases}$$

Peripheral Junction Capacitance

$$C_{jsw}(V) = \begin{cases} \dfrac{\text{cjsw}}{\left[1 - \dfrac{V}{\text{vjsw}}\right]^{\text{mjsw}}} & \text{if } V \le \text{fcsw} \cdot \text{vjsw} \\ \dfrac{\text{cjsw}}{(1-\text{fcsw})^{\text{mjsw}}}\left[1 + \dfrac{\text{mjsw}(V - \text{vjsw} \cdot \text{fcsw})}{\text{vjsw}(1-\text{fcsw})}\right] & \text{otherwise} \end{cases}$$

Diffusion Capacitance

$$C_{diff} = \frac{\text{tt}[I_j + \text{is}]}{nV_t} + \frac{\text{tt}[I_{jsw} + \text{isw}]}{n_s V_t}$$

Total Capacitance

$$C_{tot} = C_j + C_{jsw} + C_{diff} + C_d$$

Temperature Compensation Equations

For SPICE format netlist SymSpice uses the following expressions.

Energy Gap Temperature Equations

Tlev=0 or **1**

$$\text{egnom} = 1.16 - 7.02\text{e}{-4} \cdot \frac{\text{tnom}^2}{\text{tnom} + 1108}$$

$$\text{eg(t)} = 1.16 - 7.02\text{e}{-4} \cdot \frac{t^2}{t + 1108}$$

Tlev=2

$$eg(t) = Eg - Gap1 \cdot \frac{t^2}{t + Gap2}$$

Leakage Current Temperature Equations

$$Js(t) = Js \cdot e^{\frac{facln}{N}}$$

$$Jsw(t) = Jsw \cdot e^{\frac{facln}{N}}$$

Tlev=0 or **1**

$$facln = \frac{Eg}{vt(tnom)} - \frac{Eg}{vt(t)} + Xti \cdot \ln(\frac{t}{tnom})$$

Tlev=2

$$facln = \frac{egnom}{vt(tnom)} - \frac{eg(t)}{vt(t)} + Xti \cdot \ln(\frac{t}{tnom})$$

Breakdown Voltage Temperature Equations

Tlev=0

$$Bv(t) = Bv - Tcv \cdot \Delta t$$

Tlev=1 or **2**

$$Bv(t) = Bv \cdot (1 - Tcv \cdot \Delta t)$$

Transit Time Temperature Equations

$$Tt(t) = Tt \cdot (1 + Ttt1 \cdot \Delta t + Ttt2 \cdot \Delta t^2)$$

Contact Potential Temperature Equations

Tlevc=0

$$Pb(t) = Pb \cdot (\frac{t}{tnom}) - vt(t) \cdot \left[3 \cdot \ln(\frac{t}{tnom}) + \frac{egnom}{vt(tnom)} - \frac{eg(t)}{vt(t)} \right]$$

$$Php(t) = Php \cdot (\frac{t}{tnom}) - vt(t) \cdot \left[3 \cdot \ln(\frac{t}{tnom}) + \frac{egnom}{vt(tnom)} - \frac{eg(t)}{vt(t)} \right]$$

Tlevc=1 or **2**

$$Pb(t) = Pb - Tpb \cdot \Delta t$$

$$Php(t) = Php - Tphp \cdot \Delta t$$

Tlevc=3

$$Pb(t) = Pb + dpbdt \cdot \Delta t$$

$$Php(t) = Php + dphpdt \cdot \Delta t$$

where **Tlev=0** or **1**

$$dpbdt = -\frac{\left[egnom + 3 \cdot vt(tnom) + (1.16 - egnom) \cdot (2 - \frac{tnom}{tnom + 1108}) - Pb\right]}{tnom}$$

$$dphpdt = -\frac{\left[egnom + 3 \cdot vt(tnom) + (1.16 - egnom) \cdot (2 - \frac{tnom}{tnom + 1108}) - Php\right]}{tnom}$$

and **Tlev=2**

$$dpbdt = -\frac{\left[egnom + 3 \cdot vt(tnom) + (Eg - egnom) \cdot (2 - \frac{tnom}{tnom + Gap2}) - Pb\right]}{tnom}$$

$$dphpdt = -\frac{\left[egnom + 3 \cdot vt(tnom) + (Eg - egnom) \cdot (2 - \frac{tnom}{tnom + Gap2}) - Php\right]}{tnom}$$

Junction Capacitance Temperature Equations

Tlevc=0

$$Cj(t) = Cj \cdot \left[1 + Mj \cdot (4.0e{-}4 \cdot \Delta t - \frac{Pb(t)}{Pb} + 1)\right]$$

$$Cjsw(t) = Cjsw \cdot \left[1 + Mjsw \cdot (4.0e{-}4 \cdot \Delta t - \frac{Php(t)}{Php} + 1)\right]$$

Tlevc=1

$$Cj(t) = Cj \cdot (1 + Cta \cdot \Delta t)$$

$$Cjsw(t) = Cjsw \cdot (1 + Ctp \cdot \Delta t)$$

Tlevc=2

$$Cj(t) = Cj \cdot (\frac{Pb}{Pb(t)})^{Mj}$$

Note: In the above equation Mj is not Mj(t).

$$Cjsw(t) = Cjsw \cdot (\frac{Php}{Php(t)})^{Mjsw}$$

Tlevc=3

$$Cj(t) = Cj \cdot (1 - 0.5 \cdot dpbdt \cdot \frac{\Delta t}{Pb})$$

$$Cjsw(t) = Cjsw \cdot (1 - 0.5 \cdot dphpdt \cdot \frac{\Delta t}{Php})$$

Grading Coefficient Temperature Equation

$$Mj(t) = Mj \cdot (1 + Tm1 \cdot \Delta t + Tm2 \cdot \Delta t^2)$$

Resistance Temperature Equation

$$Rs(t) = Rs \cdot (1 + Trs \cdot \Delta t)$$

For SymSpice compatible SymSpice uses the following expressions.

Level-1 Model

Band Gap

$$E_g(T) = \begin{cases} 1.16 - \dfrac{7.02 \cdot 10^{-4} T^2}{1108 + T} & \text{if tlev} = 0 \text{ or } 1 \\ eg - \dfrac{gap1 T^2}{gap2 + T} & \text{if tlev} = 2 \end{cases}$$

Junction Currents

$$is = \begin{cases} is_{nom} \left(\dfrac{T}{Tnom} \right)^{\frac{xti}{n}} \exp\left[\dfrac{eg}{V_{t,nom}} - \dfrac{eg}{V_t} \right] & \text{if tlev} = 0 \text{ or } 1 \\ is_{nom} \left(\dfrac{T}{Tnom} \right)^{\frac{xti}{n}} \exp\left[\dfrac{E_g(Tnom)}{V_{t,nom}} - \dfrac{E_g(T)}{V_t} \right] & \text{if tlev} = 2 \end{cases}$$

$$isw = \begin{cases} isw_{nom} \left(\dfrac{T}{Tnom} \right)^{\frac{xti}{n,}} \exp\left[\dfrac{eg}{V_{t,nom}} - \dfrac{eg}{V_t} \right] & \text{if tlev} = 0 \text{ or } 1 \\ isw_{nom} \left(\dfrac{T}{Tnom} \right)^{\frac{xti}{n,}} \exp\left[\dfrac{E_g(Tnom)}{V_{t,nom}} - \dfrac{E_g(T)}{V_t} \right] & \text{if tlev} = 2 \end{cases}$$

Breakdown Voltage

$$bv = bv_{nom}(1 + tbv \cdot \Delta T + tbv2 \cdot \Delta T^2)$$

Parasitic Resistance

$$rs = rs_{nom}(1 + trs \cdot \Delta T + trs2 \cdot \Delta T^2)$$

$$rsw = rsw_{nom}(1 + trs \cdot \Delta T + trs2 \cdot \Delta T^2)$$

Leakage Conductance

$$gleak = gleak_{nom}(1 + tgs \cdot \Delta T + tgs2 \cdot \Delta T^2)$$

$$gleaksw = gleaksw_{nom}(1 + tgs \cdot \Delta T + tgs2 \cdot \Delta T^2)$$

Junction Potential

$$vj = \begin{cases} vj_{nom}\left(\frac{T}{Tnom}\right) - \Delta V_j & \text{if } tlevc = 0 \\ vj_{nom} - pta \cdot \Delta T & \text{if } tlevc = 1 \text{ or } 2 \\ vj_{nom} - \frac{dV_j}{dT} \cdot \Delta T & \text{if } tlevc = 3 \end{cases}$$

$$vjsw = \begin{cases} vjsw_{nom}\left(\frac{T}{Tnom}\right) - \Delta V_j & \text{if } tlevc = 0 \\ vjsw_{nom} - ptp \cdot \Delta T & \text{if } tlevc = 1 \text{ or } 2 \\ vjsw_{nom} - \frac{dV_{jsw}}{dT} \cdot \Delta T & \text{if } tlevc = 3 \end{cases}$$

where

$$\Delta V_j = V_t\left[3\ln\left(\frac{T}{Tnom}\right) + \frac{E_g(Tnom)}{V_{t,nom}} - \frac{E_g(T)}{V_t}\right]$$

$$\frac{dV_j}{dT} = \begin{cases} \frac{\left[E_g(Tnom) + 3V_{t,nom} + (1.16 - E_g(Tnom))\left(2 - \frac{Tnom}{Tnom + 1108}\right) - vj\right]}{Tnom} & \text{if } tlev = 0 \text{ or } 1 \\ \frac{\left[E_g(Tnom) + 3V_{t,nom} + (eg - E_g(Tnom))\left(2 - \frac{Tnom}{Tnom + gap2}\right) - vj\right]}{Tnom} & \text{if } tlev = 2 \end{cases}$$

$$\frac{dV_{jsw}}{dT} = \begin{cases} \frac{\left[E_g(Tnom) + 3V_{t,nom} + (1.16 - E_g(Tnom))\left(2 - \frac{Tnom}{Tnom + 1108}\right) - vjsw\right]}{Tnom} & \text{if } tlev = 0 \text{ or } 1 \\ \frac{\left[E_g(Tnom) + 3V_{t,nom} + (eg - E_g(Tnom))\left(2 - \frac{Tnom}{Tnom + gap2}\right) - vjsw\right]}{Tnom} & \text{if } tlev = 2 \end{cases}$$

Junction Capacitance

$$cjo = \begin{cases} cjo_{nom}\left[1 + m\left(0.0004 \cdot \Delta T - \frac{vj}{vj_{nom}} + 1\right)\right] & \text{if } tlevc = 0 \\ cjo_{nom}(1 + cta \cdot \Delta T) & \text{if } tlevc = 1 \\ cjo_{nom}\left(\frac{vj_{nom}}{vj}\right)^m & \text{if } tlevc = 2 \\ cjo_{nom}\left(1 - 0.5\frac{dV_j}{dT}\frac{\Delta T}{vj_{nom}}\right) & \text{if } tlevc = 3 \end{cases}$$

$$\text{cjsw} = \begin{cases} \text{cjsw}_{\text{nom}}\left[1+\text{m}\left(0.0004\cdot\Delta\text{T}-\dfrac{\text{vjsw}}{\text{vjsw}_{\text{nom}}}+1\right)\right] & \text{if tlevc} = 0 \\ \text{cjsw}_{\text{nom}}\left(1+\text{ctp}\cdot\Delta\text{T}\right) & \text{if tlevc} = 1 \\ \text{cjsw}_{\text{nom}}\left(\dfrac{\text{vjsw}_{\text{nom}}}{\text{vjsw}}\right)^{\text{mjsw}} & \text{if tlevc} = 2 \\ \text{cjsw}_{\text{nom}}\left(1-0.5\dfrac{\text{dV}_{\text{jsw}}}{\text{dT}}\dfrac{\Delta\text{T}}{\text{vjsw}_{\text{nom}}}\right) & \text{if tlevc} = 3 \end{cases}$$

Bipolar junction transistors

Syntax in 'local' or 'spectre' modes

without model:

Name nc nb ne <ns> bjt parameter=value ...

with model:

Name nc nb ne <ns> modelname parameter=value ...

model modelname bjt pname=val

Syntax in 'spice' mode

Qxxx nc nb ne <ns> mname

+<area> <OFF> <IC=vbeval,vceval>

+ <M=val> <DTEMP=val>

or

Qxxx nc nb ne <ns> mname

+<AREA=area> <AREAB=val>

+ <AREAC=val> <OFF>

+<VBE=vbeval> <VCE=vceval>

+<M=val> <DTEMP=val>

The arguments are the following:

SPICE mode	SymSpice mode	Default	Description
Qxxx	Name	-	BJT element name. For SPICE netlist it must begin with "Q", which can be followed by alphanumeric characters.
nc	nc	-	Collector terminal node name.
nb	nb	-	Base terminal node name.
ne	ne	-	Emitter terminal node name.
ns	ns	-	Substrate terminal node name, optional. Can also be set in the BJT model with the parameter Bulk.

mname	bjt/modelname	-	BJT model name reference.
AREA	area	1.0	Emitter area multiplying factor which affects currents, resistances and capacitances. Default=1.0.
M	m	1	Multiplier to simulate multiple BJTs in parallel. All currents, capacitances and resistances are affected by the setting of M. Default=1.0.
DTEMP	trise	-	The difference between the element temperature and the circuit temperature in Celsius. Default=0.0.
AREAB	areab	1.0	Base area multiplying factor, which affects currents, resistances and capacitances. Default=Area.
AREAC	areac	1.0	Collector area multiplying factor, which affects currents, resistances and capacitances. Default=Area.

Using BJT models

The bipolar-junction transistor (BJT) model in SymSpice is an adaptation of Gummel-Poon model. The SymSpice model extends the original Gummel-Poon model to include several effects at high bias levels. This model automatically simplifies to the Ebers-Moll model when certain parameters (Vaf, Var, Ikf and Ikr) are not specified.

The BJT model is used to develop BiCMOS, Ttl, and Ecl circuits. For BiCMOS devices, use the high current Beta degradation parameters, Ikf and Ikr, to modify high injection effects. The model parameter Subs facilitates the modeling of both vertical and lateral geometrics.

BJT basic model (group) parameters

BJT model (group) parameters are divided into several subsets. DC model parameters include the most basic Ebers-Moll parameters. This model is effective for modeling low-frequency large-signal characteristics.

Low current Beta degradation effect parameters Isc, Ise, Nc and Ne aid in modeling the drop in the observed Beta, caused by the following mechanisms:

- recombination of carriers in the emitter-base space charge layer
- recombination of carriers at the surface

- formation of emitter-base channels

Low base and emitter dopant concentrations, found in some BIMOS type technologies, use the high current Beta degradation parameters, Ikf and Ikr. Use the base-width modulation parameters, that is, Early effect parameters Vaf and Var, to model high-gain, narrow-base devices. The model calculates the slope of the I-V curve for the model in the active region with Vaf and Var. If Vaf and Var are not specified, the slope in the active region is zero.

The parasitic resistor parameters Re, Rb, and Rc are the most frequently used second-order parameters since they replace external resistors. This simplifies the input netlist file. All of the resistances are functions of the BJT multiplier M value. The resistances are divided by M to simulate parallel resistances. The base resistance is also a function of base current, as is often the case in narrow-base technologies.

Transient model parameters for BJTs are composed of two subsets: junction capacitor parameters and transit time parameters. The base-emitter junction is modeled by Cje, Vje, and Mje. The base-collector junction capacitance is modeled by Cjc, Vjc, and Mjc. The collector-substrate junction capacitance is modeled by Cjs, Vjs, and Mjs.

Tf is the forward transit time for base charge storage. Tf can be modified to account for bias, current, and phase, by Xtf, Vtf, Itf, and Ptf. The base charge storage reverse transit time is set by Tr. There are several sets of temperature equations for the BJT model parameters that you can select by setting Tlev and Tlevc.

BJT basic DC model parameters

SPICE mode	SymSpice mode	Units	Default	Description
bf(bfm)	bf(bfm)		100.0	Ideal maximum forward beta
br(brm)	br(brm)		1.0	Ideal maximum reverse beta
bulk(nsub)	bulk(nsub)		Gnd	Bulk node name
Ibc	Ibc	A	0.0	Reverse saturation current between base and collector
expli	expli	A	1E15	Current explosion model parameter
Ibe	Ibe	A	0.0	Reverse saturation current between base and emitter
Is	Is	A	1E-16	Transport saturation current

Iss	Iss		0.0	Reverse saturation current bulk-to-collector or bulk-to-base
level	level		1.0	Model selector
nf	nf		1.0	Forward current emission coefficient
nr	nr		1.0	Reverse current emission coefficient
ns	ns		1.0	Substrate current emission coefficient
subs	struct		1.0	Substrate connection selector: +1 for vertical geometry, -1 for lateral
update	update		0.0	Base charge equation selector

Low Current Beta Degradation Effect Parameters

Parameter name	Units	Default	Description
isc(C4,jlc)	A	0.0	Base-collector leakage saturation current
ise(C2, jle)	A	0.0	Base-emmiter leakage saturation current
nc(nlc)		2.0	Base-collector leakage emission coefficient
ne(nle)		1.5	Base-emitter leakage emission coefficient

High Current Beta Degradation Effect Parameters

Parameter name	Units	Default	Description
ikf(ik,jbf)	A	0.0	Corner for forward Beta high current roll off
ikr(jbr)	A	0.0	Corner for reverse Beta high current roll off
nkf		0.5	Exponent for high current Beta roll-off

Base Width Modulation Parameters

Parameter name	Units	Default	Description
vaf(va,vbf)	V	0.0	Forward early voltage
var(vb,vrb,bv)	V	0.0	Reverse early voltage

Parasitic Resistance Parameters

Parameter name	Units	Default	Description
irb(jrb,iob)	A	0.0	Base current , where base resistance falls half-way to rbm
rb	Ohm	0.0	Base resistance
rbm	Ohm	0.0	Minimum high-current base resistance
re	Ohm	0.0	Emitter resistance
rc	Ohm	0.0	Collector resistance

Junction Capacitor Parameters

Parameter name	Units	Default	Description
cjc	F	0.0	Base-collector zero-bias depletion capacitance
cje	F	0.0	Base-emitter zero-bias depletion capacitance
cjs(ccs,csub)	F	0.0	Zero-bias collector substrate capacitance
fc		0.5	Coeff. for forward bias depletion capacitance
mjc(mc)		0.33	Base-collector junction exponent
mje(me)		0.33	Base-emitter junction exponent
mjs(esub)		0.5	Substrate junction exponent
vjc(pc)	V	0.75	Base-collector built-in potential
vje(pe)	V	0.75	Base-emitter built-in potential
vjs(psub)	V	0.75	Substrate junction built-in potential
xcjc(cdis)	F	1.0	Internal base fraction of base-collector depletion capacitance

Parasitic Capacitances

Parameter name	Units	Default	Description
cbcp	F	0.0	External base-collector constant capacitance
cbep	F	0.0	External base-emitter constant capacitance
ccsp	F	0.0	External collector-substrate constant capacitance

Transit Time Parameters

Parameter name	Units	Default	Description
itf(jtf)	A	0.0	Tf high current parameter
ptf		0.0	Frequency multiplier to determine excess phase
tf	s	0.0	Base forward transit time
tr	s	0.0	Base reverse transit time
vtf	V	0.0	Tf base-collector voltage dependence coeff.
xtf		0.0	Tf bias dependence coeff.

BJT LEVEL=2 Model Parameters

Parameter name	Units	Default	Description
brs		1.0	Reverse beta for substrate BJT
gamma		0.0	Epitaxial doping factor
nepi		1.0	Emission coeficient
qco	Coul	0.0	Epitaxial charge factor
vo	V	0.0	Carrier velocity saturation voltage. Use zero to indicate an infinite value

Handling BJT Model Temperature Effects

Several temperature parameters control derating of the BJT model parameters. They include temperature parameters for junction capacitance, Beta degradation (DC), and base modulation (Early effect) among others.

BJT Temperature Parameters

SPICE mode	SymSpice mode	Units	Default	Description
bex	bex		2.42	Rc temperature exponent (level 2)
bexv	bexv		1.90	Vo temperature exponent (level 2)
ctc	ctc	1/C°	0.0	Temperature coeff. for zero-bias base collector capacitance

cte	cte	1/C°	0.0	Temperature coeff. for zero-bias base emitter capacitance
cts	cts	1/C°	0.0	Temperature coeff. for zero-bias substrate capacitance
eg	eg	eV	1.11	Energy gap for p-n junction
gap1	gap1	eV/C°	7.02E-4	First bandgap correction factor
gap2	gap2	C°	1108.0	Second bandgap correction factor
tbf1	tbf1	1/C°	0.0	First order temperature coeff for bf
tbf2	tbf2	$1/C^{o2}$	0.0	Second order temperature coeff for br
tbr1	tbr1	1/C°	0.0	First order temperature coeff for br
tbr2	tbr2	$1/C^{o2}$	0.0	Second order temperature coeff for br
tikf1	tikf1	1/C°	0.0	First order temperature coeff for Ikf
tikf2	tikf2	$1/C^{o2}$	0.0	Second order temperature coeff for Ikf
tikr1	tikr1	1/C°	0.0	First order temperature coeff for Ikr
tikr2	tikr2	$1/C^{o2}$	0.0	Second order temperature coeff for Ikr
tirb1	tirb1	1/C°	0.0	First order temperature coeff for Irb
tirb2	tirb1	$1/C^{o2}$	0.0	Second order temperature coeff for Irb
tisc1	tisc1	1/C°	0.0	First order temperature coeff for Isc
tisc2	tisc2	$1/C^{o2}$	0.0	Second order temperature coeff for Isc
tis1	tis1	1/C°	0.0	First order temperature coeff for Is
tis2	tis2	$1/C^{o2}$	0.0	Second order temperaturecoeff for Is
tise1	tise1	1/C°	0.0	First order temperature coeff for Ise
tise2	tise2	$1/C^{o2}$	0.0	Second order temperature coeff for Ise
tiss1	tiss1	1/C°	0.0	First order temperature coeff for Iss
tiss2	tiss2	$1/C^{o2}$	0.0	Second order temperature coeff for Iss
titf1	titf1	1/C°	0.0	First order temperature coeff for Itf
titf2	titf2	$1/C^{o2}$	0.0	Second order temperature coeff for Itf
tlev	tlev		1.0	Temperature equation level selector
tlevc	tlevc		1.0	Temperature equation level selector for capacitances and potentials
tmjc1	tmjc1	1/C°	0.0	First order temperature coeff for mjc
tmjc2	tmjc2	$1/C^{o2}$	0.0	Second order temperature coeff for mjc
tmje1	tmje1	1/C°	0.0	First order temperature coeff for mje

tmje2	tmje2	$1/C^{o2}$	0.0	Second order temperature coeff for mje
tmjs1	tmjs1	1/C°	0.0	First order temperature coeff for mjs
tmjs2	tmjs2	$1/C^{o2}$	0.0	Second order temperature coeff for mjs
tnc1	tnc1	1/C°	0.0	First order temperature coeff for nc
tnc2	tnc2	$1/C^{o2}$	0.0	Second order temperature coeff for nc
tne1	tne1	1/C°	0.0	First order temperature coeff for ne
tne2	tne2	$1/C^{o2}$	0.0	Second order temperature coeff for ne
tnf1	tnf1	1/C°	0.0	First order temperature coeff for nf
tnf2	tnf2	$1/C^{o2}$	0.0	Second order temperature coeff for nf
tnr1	tnr1	1/C°	0.0	First order temperature coeff for nr
tnr2	tnr2	$1/C^{o2}$	0.0	Second order temperature coeff for nr
tns1	tns1	1/C°	0.0	First order temperature coeff for ns
tns2	tns2	$1/C^{o2}$	0.0	Second order temperature coeff for ns
trb1(trb)	trb1(trb)	1/C°	0.0	First order temperature coeff for rrb
trb2	trb2	$1/C^{o2}$	0.0	Second order temperature coeff for rb
trc1(trc)	trc1(trc)	1/C°	0.0	First order temperature coeff for rc
trc2	trc2	$1/C^{o2}$	0.0	Second order temperature coeff for rc
tre1(tre)	tre1(tre)	1/C°	0.0	First order temperature coeff for re
tre2	tre2	$1/C^{o2}$	0.0	Second order temperature coeff for re
trm1	trm1	1/C°	0.0	First order temperature coeff for rbm
trm2	trm2	$1/C^{o2}$	0.0	Second order temperature coeff for rbm
ttf1	ttf1	1/C°	0.0	First order temperature coeff for tf
ttf2	ttf2	$1/C^{o2}$	0.0	Second order temperature coeff for tf
ttr1	ttr1	1/C°	0.0	First order temperature coeff for tr
ttr2	ttr2	$1/C^{o2}$	0.0	Second order temperature coeff for tr
tvaf1	tvaf1	1/C°	0.0	First order temperature coeff for vaf
tvaf2	tvaf2	$1/C^{o2}$	0.0	Second order temperature coeff for vaf
tvar1	tvar1	1/C°	0.0	First order temperature coeff for var
tvar2	tvar2	$1/C^{o2}$	0.0	Second order temperature coeff for var
tvjc	tvjc	V/C°	0.0	Temperature coeff for vjc
tvje	tvje	V/C°	0.0	Temperature coeff for vje
tvjs	tvjs	V/C°	0.0	Temperature coeff for vjs

xtb(tb,tcb)	xtb(tb,tcb)		0.0	Forward and reverse Beta temperature exponent
xti	xti		3.0	Saturation current temperature exponent
tref	tref	C°	tnom	Reference temperature
af	af	-	1.0	Flicker noise exponent
kf	kf	-	0.0	Flicker noise coefficient
-	c2	-	0.0	Forward leakage saturation current coefficient
-	c4	-	0.0	Reverse leakage saturation current coefficient
-	cbo	A	0.0	Extrapolated 0-volt B-C leakage current (*area)
-	gbo	S	0.0	Slope of Icbo vs. Vbc above Vbo (*area)
-	vbo	V	-	Slope of Icbo vs. Vbc at Vbc=0
-	tcbo	1/C°	0.0	Temperature coefficient of cbo
-	tgbo	1/C°	0.0	Temperature coefficient of gbo
-	ke	1/V	-	B-E space-charge integral multiplier
-	kc	1/V	-	B-C space-charge integral multiplier
-	rbmod	-	spice	Nonlinear Rb model. Possible values are local or spice
-	rcv	Ohm	0	Variable collector resistance (/area)
-	rcm	Ohm	0	Minimum collector resistance (/area)
-	dope	cm^{-3}	1e15	Collector background doping concentration
-	gex	-	1	Current crowding exponent
-	gco	A	1	Current crowding normalization constant (*area)
-	minr	Ohm	0.1	Minimum parasitic resistance
-	xcjc2	-	1	Fraction of B-C capacitance tied to collector and fraction of B-C tied to internal node
-	td	sec	0	Intrinsic base delay time
-	trise	C°	0	Temperature rise from ambient
-	tvtf1	1/C°	0	Linear temperature coefficient for vtf
-	tvtf2	$(C°)^{-2}$	0	Quadratic temperature coefficient for vtf
-	txtf1	1/C°	0	Linear temperature coefficient for xtf

-	txtf2	$(C°)^{-2}$	0	Quadratic temperature coefficient for xtf
-	dskip	-	yes	Skip junction calculations if they are reverse-saturated. Possible values are no or yes
-	bvbe	V	-	B-E breakdown voltage
-	bvbc	V	-	B-C breakdown voltage
-	bvce	V	-	C-E breakdown voltage
-	bvsub	V	-	Substrate junction breakdown voltage
-	vbefwd	V	0.2	B-E forward voltage
-	vbcfwd	V	0.2	B-C forward voltage
-	vsubfwd	V	0.2	Substrate junction forward voltage
-	imax	A	1e3	Maximum allowable base current (*area)
-	imax1	A	-	Maximum allowable collector current (*area)
-	alarm	-	none	Forbidden operating region. Possible values are none, off, fwd, rev, or sat
-	kb	-	0	Burst noise coefficient
-	bnoisefc	-	1	Burst noise cutoff frequency
-	rbnoi	Ohm	-	Effective base noise resistance

Scaling

Scaling is controlled by the element (individual) parameters Area, Areab, Areac, and M. The Area parameter, the normalized emitter area, divides all resistors and multiplies all currents and capacitors. Areab and Areac scale the size of the base area and collector area. Either Areab or Areac is used for scaling, depending on whether vertical or lateral geometry is selected (using the Subs model parameter). For vertical geometry, Areab is the scaling factor for Ibc, Isc, and Cjc. For lateral geometry, Areac is the scaling factor. The scaling factor is Area for all other parameters.
The scaling of the DC model parameters (Ibe, Is, Ise, Ikf, Ikr, and Irb) for both vertical and lateral BJT transistors, is determined by the following formula:

$$Ieff = Area \cdot M \cdot I$$

where I is either Ibe, Is, Ise, Ikf, Ikr, or Irb.
For both the vertical and lateral, the resistor model parameters, Rb, Rbm, Re, and Rc are scaled by the following equation:

$$Reff = \frac{R}{Area \cdot M}$$

where R is either Rb, Rbm, Re, or Rc.

Understanding Transistor Geometry in Substrate Diodes

The substrate diode is connected to either the collector or the base depending on whether the transistor has a lateral or vertical geometry. Lateral geometry is implied when the model parameter $\text{Subs} = -1$, and vertical geometry when $\text{Subs} = +1$. The lateral transistor substrate diode is connected to the internal base and the vertical transistor substrate diode is connected to the internal collector.

DC Model Equations

For SPICE netlist SymSpice uses the following expressions.

DC model equations are for the DC component of the collector current (ic) and the base current (ib).

Current Equations – Is Only

If only Is is specified, without Ibe and Ibc:

$$ic = \frac{\text{ISeff}}{qb}\cdot\left(e^{\frac{vbe}{Nf\cdot vt}} - e^{\frac{vbc}{Nr\cdot vt}}\right) - \frac{\text{ISeff}}{Br}\cdot\left(e^{\frac{vbc}{Nr\cdot vt}} - 1\right) - \text{ISCeff}\cdot\left(e^{\frac{vbc}{Nc\cdot vt}} - 1\right)$$

$$ib = \frac{\text{ISeff}}{Bf}\cdot\left(e^{\frac{vbe}{Nf\cdot vt}} - 1\right) + \frac{\text{ISeff}}{Br}\cdot\left(e^{\frac{vbc}{Nr\cdot vt}} - 1\right) + \text{ISEeff}\cdot\left(e^{\frac{vbe}{Ne\cdot vt}} - 1\right) + \text{ISCeff}\cdot\left(e^{\frac{vbc}{Nc\cdot vt}} - 1\right)$$

Current Equations – Ibe and Ibc

If Ibe and Ibc are specified, instead of Is:

$$ic = \frac{\text{IBEeff}}{qb}\cdot\left(e^{\frac{vbe}{Nf\cdot vt}} - 1\right) - \frac{\text{IBCeff}}{qb}\cdot\left(e^{\frac{vbc}{Nr\cdot vt}} - 1\right) - \frac{\text{IBCeff}}{Br}\cdot\left(e^{\frac{vbc}{Nr\cdot vt}} - 1\right) - \text{ISCeff}\cdot\left(e^{\frac{vbc}{Nc\cdot vt}} - 1\right)$$

$$ib = \frac{\text{IBEeff}}{Bf}\cdot\left(e^{\frac{vbe}{Nf\cdot vt}} - 1\right) + \frac{\text{IBCeff}}{Br}\cdot\left(e^{\frac{vbc}{Nr\cdot vt}} - 1\right) + \text{ISEeff}\cdot\left(e^{\frac{vbe}{Ne\cdot vt}} - 1\right) + \text{ISCeff}\cdot\left(e^{\frac{vbc}{Nc\cdot vt}} - 1\right)$$

$\text{IBCeff} = \text{Ibc}\cdot\text{Areab}\cdot\text{M}$ Vertical

$\text{IBCeff} = \text{Ibc}\cdot\text{Areac}\cdot\text{M}$ Lateral

$\text{IBEeff} = \text{Ibe}\cdot\text{Area}\cdot\text{M}$ Vertical or Lateral

$\text{ISCeff} = \text{Isc}\cdot\text{Areab}\cdot\text{M}$ Vertical

$\text{ISCeff} = \text{Isc}\cdot\text{Areac}\cdot\text{M}$ Lateral

$\text{ISEeff} = \text{Ise}\cdot\text{Area}\cdot\text{M}$ Vertical or Lateral

The last two terms in the expression of the base current represent the components due to recombination in the base-emitter and base-collector space charge regions at low injection.

For SymSpice netlist SymSpice uses the following expressions.

$$I_c = \frac{is}{Q_B}\left(e^{\frac{V_{BE}}{nfV_t}} - e^{\frac{V_{BC}}{nrV_t}}\right) - \frac{is}{br}\left(e^{\frac{V_{BC}}{nrV_t}} - 1\right) - isc\left(e^{\frac{V_{BC}}{ncV_t}} - 1\right)$$

$$I_b = \frac{is}{bf}\left(e^{\frac{V_{BE}}{nfV_t}} - 1\right) + ise\left(e^{\frac{V_{BE}}{nsV_t}} - 1\right) + \frac{is}{br}\left(e^{\frac{V_{BC}}{nrV_t}} - 1\right) + isc\left(e^{\frac{V_{BC}}{ncV_t}} - 1\right)$$

where V_t is the thermal voltage given by

$$V_t = \frac{kT}{q}$$

Substrate Current Equations

For SymSpice netlist SymSpice uses the following expressions.

The substrate current is substrate to collector for vertical transistors and substrate to base for lateral transistors.

Vertical Transistors

$$\text{isc} = \text{ISSeff} \cdot \left(e^{\frac{\text{vsc}}{\text{NS}\cdot\text{vt}}} - 1\right), \quad \text{vsc} > -10 \cdot \text{Ns} \cdot \text{vt}$$

$$\text{isc} = -\text{ISSeff}, \quad \text{vsc} \le -10 \cdot \text{Ns} \cdot \text{vt}$$

Lateral Transistors

$$\text{ibs} = \text{ISSeff} \cdot \left(e^{\frac{\text{vbs}}{\text{NS}\cdot\text{vt}}} - 1\right), \quad \text{vbs} > -10 \cdot \text{Ns} \cdot \text{vt}$$

$$\text{ibs} = -\text{ISSeff}, \quad \text{vbs} \le -10 \cdot \text{Ns} \cdot \text{vt}$$

If both Ibe and Ibc are not specified:

$$\text{ISSeff} = \text{Iss} \cdot \text{Area} \cdot \text{M}$$

If both Ibe and Ibc are specified:

$$\text{ISSeff} = \text{Iss} \cdot \text{Areac} \cdot \text{M} \quad \text{Vertical}$$

$$\text{ISSeff} = \text{Iss} \cdot \text{Areab} \cdot \text{M} \quad \text{Lateral}$$

For SymSpice netlist SymSpice uses the following expressions.

If *ise* (*isc*) is not given and *C*2 (*C*4) is specified, *ise* (*isc*) is calculated from

$$ise = C2 \cdot is$$

$$isc = C4 \cdot is$$

Base Charge Equations

For SPICE netlist SymSpice uses the following expressions.

Vaf and *Var* are, respectively, forward and reverse early voltages. *Ikf* and *Ikr* determine the high current Beta roll-off. *Ise*, *Isc*, *N*, and *Nc* determine the low current Beta roll-off with ic.

If *UPDATE*=0 or $\frac{\text{vbc}}{\text{Vaf}} + \frac{\text{vbe}}{\text{Var}} < 0$, then

$$q1 = \frac{1}{\left(1 - \frac{vbc}{Vaf} - \frac{vbe}{Var}\right)}$$

Otherwise, if *UPDATE*=1 and $\frac{vbc}{Vaf} + \frac{vbe}{Var} \geq 0$, then

$$q1 = \left(1 + \frac{vbc}{Vaf} + \frac{vbe}{Var}\right)$$

$$q2 = \frac{ISEeff}{IKFeff} \cdot \left(e^{\frac{vbe}{Nf \cdot vt}} - 1\right) + \frac{ISCeff}{IKReff} \cdot \left(e^{\frac{vbc}{Nr \cdot vt}} - 1\right)$$

$$qb = \frac{q1}{2} \cdot \left[1 + (1 + 4 \cdot q2)^{Nkf}\right]$$

For SymSpice netlist SymSpice uses the following expressions.

$$Q_B = \frac{Q_1}{2}(1 + \sqrt{1 + 4Q_2})$$

$$Q_1 = \begin{cases} \dfrac{1}{1 - \dfrac{V_{BC}}{vaf} - \dfrac{V_{BE}}{var}} & \text{if neither ke or kc is specified} \\ (1 + \int_0^{V_{BE}} f_{cj}(ke, vje, mje)dv + \int_0^{V_{BC}} f_{cj}(kc, vjc, mjc)dv) & \text{otherwise} \end{cases}$$

where f_{cj} is defined as follows:

$$f_{cj}(C, P, M) = \begin{cases} \dfrac{C}{(1 - \frac{v}{P})^M} & \text{if } v \leq fc \cdot P \\ \dfrac{C}{(1 - fc)^{1+M}}[1 - fc(1 + M) + \dfrac{M}{P}v] & \text{otherwise} \end{cases}$$

$$Q_2 = \frac{is}{ikf}\left(e^{\frac{V_{BE}}{nf V_t}} - 1\right) + \frac{is}{ikr}\left(e^{\frac{V_{BC}}{nr V_t}} - 1\right)$$

Variable Resistance Equations

For SPICE netlist SymSpice uses the following expressions.

SymSpice provides a variable base resistance model consisting of a low-current maximum resistance set by Rb and a high-current minimum resistance set by Rbm. Irb is the current when the base resistance is halfway to its minimum value. If Rbm is not specified, it is set to Rb.

If Irb is not specified:

$$rbb = RBMeff + \frac{RBeff - RBMeff}{qb}$$

If Irb is specified:

$$rbb = RBMeff + 3\cdot(RBeff - RBMeff)\cdot\frac{\tan(z) - z}{z\cdot\tan(z)\cdot\tan(z)}$$

$$z = \frac{-1+\left[1+144\cdot ib/\left(\pi^2\cdot IRBeff\right)\right]^{1/2}}{\frac{24}{\pi^2}\cdot\left(\frac{ib}{IRBeff}\right)^{1/2}}$$

For SymSpice netlist SymSpice uses the following expressions.

Nonlinear base resistance

$$R_B = rbm + \frac{rb - rbm}{Q_B} \quad if\ irb\ is\ not\ given$$

If *irb* is given and SPICE compatibility is required,

$$R_B = rbm + 3(rb - rbm)\frac{\tan(z) - z}{z\tan^2(z)}$$

where

$$z = \frac{-1+\sqrt{1+\frac{144 I_b}{irb\,\pi^2}}}{\frac{24}{\pi^2}\sqrt{\frac{I_b}{irb}}}$$

If *irb* is given and SPICE compatibility is required,

$$R_B = rbm + \frac{rb - rbm}{\sqrt{1+3\left(\frac{I_b}{irb}\right)^{0.852}}}$$

Nonlinear collector resistance (if *rcv* is specified)

$$R_C = rcv\left[1+\left(\frac{n_i}{dope}\right)^2\left(e^{\frac{V_{BC}}{V_t}}\right)\right]^{-1}\left[1+\left(\frac{I_c}{cco}\right)^{cex}\right] + rcm$$

BJT Capacitance Equations

Base-Emitter Capacitance Equations

The base-emitter capacitance contains a complex diffusion term with the standard depletion capacitance formula. The diffusion capacitance is modified by model parameters Tf, Xtf, Itf, and Vtf.

Determine the base-emitter capacitance cbe by the following formula:

$$cbe = cbediff + cbedep$$

where cbediff and cbedep are the base-emitter diffusion and depletion capacitances, respectively.

Note: When you run a DC sweep on a BJT, use Dcap to force the evaluation of the voltage-variable capacitances during the DC sweep.

Base-Emitter Diffusion Capacitance

Determine diffusion capacitance as follows:

$ibe \le 0$

$$cbediff = \frac{\partial}{\partial vbe}\left(Tf \cdot \frac{ibe}{qb}\right)$$

$vbe \ge 0$

$$cbediff = \frac{\partial}{\partial vbe}\left[Tf \cdot (1 + argtf) \cdot \frac{ibe}{qb}\right]$$

where:

$$argtf = Xtf \cdot \left(\frac{ibe}{ibe + Itf}\right)^2 \cdot e^{\frac{vbc}{1.44 \cdot Vtf}}$$

The forward part of the collector-emitter branch current is determined as follows:

$$ibe = ISeff \cdot \left(e^{\frac{vbe}{Nf \cdot vt}} - 1\right)$$

Base-Emitter Depletion Capacitance

There are two different equations for modeling the depletion capacitance. Select the proper equation by specifying option Dcap.

Dcap=1

The base-emitter depletion capacitance is determined as follows:

$vbe < Fc \cdot Vje$

$$cbedep = CJEeff \cdot \left(1 - \frac{vbe}{Vje}\right)^{-Mje}$$

$vbe \ge Fc \cdot Vje$

$$cbedep = CJEeff \cdot \frac{1 - Fc \cdot (1 + Mje) + Mje \cdot \frac{vbe}{Vje}}{(1 - Fc)^{(1+Mje)}}$$

Dcap2

The base-emitter depletion capacitance is determined as follows:

$vbe < 0$

$$cbedep = CJEeff \cdot \left(1 - \frac{vbe}{Vje}\right)^{-Mje}$$

$vbe \ge 0$

$$cbedep = CJEeff \cdot \left(1 + Mje \cdot \frac{vbe}{Vje}\right)$$

Dcap=3

Limits peak depletion capacitance to $Fc \cdot CJCeff$ or $Fc \cdot CJEeff$, with proper fall-off when forward bias exceeds Pb ($Fc \ge 1$).

Base-Collector Capacitance

Determine the base-collector capacitance cbc as follows:

$$cbc = cbcdiff + cbcdep$$

where cbcdiff and cbcdep are the base-collector diffusion and depletion capacitances, respectively.

Base-Collector Diffusion Capacitance

$$cbcdiff = \frac{\partial}{\partial vbc} \cdot (Tr \cdot ibc)$$

where the internal base-collector current ibc is:

$$ibc = ISeff \cdot \left(e^{\frac{vbc}{Nr \cdot vt}} - 1 \right)$$

Base-Collector Depletion Capacitance

There are two different equations for modeling the depletion capacitance. Select the proper equation by specifying option Dcap.

Dcap=1

Specify Dcap=1 to select one of the following equations:

$$vbc < Fc \cdot Vjc$$

$$cbcdep = Xcjc \cdot CJCeff \cdot \left(1 - \frac{vbc}{Vjc} \right)^{-Mjc}$$

$$vbc \geq Fc \cdot Vjc$$

$$cbcdep = Xcjc \cdot CJCeff \cdot \frac{1 - Fc \cdot (1 + Mjc) + Mjc \cdot \frac{vbc}{Vjc}}{(1 - Fc)^{(1+Mjc)}}$$

Dcap=2

Specify *Dcap*=2 to select one of the following equations:

$$vbc < 0$$

$$cbcdep = Xcjc \cdot CJCeff \cdot \left(1 - \frac{vbc}{Vjc} \right)^{-Mjc}$$

$$vbc \geq 0$$

$$cbcdep = Xcjc \cdot CJCeff \cdot \left(1 + Mjc \cdot \frac{vbc}{Vjc} \right)$$

External Base – Internal Collector Junction Capacitance

The base-collector capacitance is modeled as a distributed capacitance when the model parameter Xcjc is set. Since the default setting of Xcjc is one, the entire base-collector capacitance is on the internal base node cbc.

Dcap=1

Specify Dcap=1 to select one of the following equations:

$$vbcx < Fc \cdot Vjc$$

$$cbcx = CJCeff \cdot (1 - Xcjc) \cdot \left(1 - \frac{vbcx}{Vjc}\right)^{-Mjc}$$

$$vbcx \geq Fc \cdot Vjc$$

$$cbcx = CJCeff \cdot (1 - Xcjc) \cdot \frac{1 - Fc \cdot (1 + Mjc) + Mjc \cdot \frac{vbcx}{Vjc}}{(1 - Fc)^{(1+Mjc)}}$$

Dcap=2

Specify Dcap=2 to select one of the following equations:

$$vbcx < 0$$

$$cbcx = CJCeff \cdot (1 - Xcjc) \cdot \left(1 - \frac{vbcx}{Vjc}\right)^{-Mjc}$$

$$vbcx \geq 0$$

$$cbcx = CJCeff \cdot (1 - Xcjc) \cdot \left(1 + Mjc \cdot \frac{vbcx}{Vjc}\right)$$

where vbcx is the voltage between the external base node and the internal collector node.

Substrate Capacitance

The function of substrate capacitance is similar to that of the substrate diode. Switch it from the collector to the base by setting the model parameter, *Subs*.

Substrate Capacitance Equation – Lateral

Base to Substrate Diode

Reverse Bias $vbs < 0$:

$$cbs = CJSeff \cdot \left(1 - \frac{vbs}{Vjs}\right)^{-Mjs}$$

Forward Bias $vbs \geq 0$:

$$cbs = CJSeff \cdot \left(1 + Mjs \cdot \frac{vbs}{Vjs}\right)$$

Substrate Capacitance Equation – Vertical

Substrate to Collector Diode

Reverse Bias $vsc < 0$:

$$csc = CJSeff \cdot \left(1 - \frac{vsc}{Vjs}\right)^{-Mjs}$$

Forward Bias $vsc \geq 0$:

$$csc = CJSeff \cdot \left(1 + Mjs \cdot \frac{vsc}{Vjs}\right)$$

BJT Temperature Compensation Equations

Energy Gap Temperature Equations

Tlev=0, 1 or 3

$$egnom = 1.16 - 7.02e{-}4 \cdot \frac{tnom^2}{tnom + 1108.0}$$

$$eg(t) = 1.16 - 7.02e{-}4 \cdot \frac{t^2}{t + 1108.0}$$

Tlev=2

$$egnom = Eg - Gap1 \cdot \frac{tnom^2}{tnom + Gap2}$$

$$eg(t) = Eg - Gap1 \cdot \frac{t^2}{t + Gap2}$$

Saturation and Beta Temperature Equations

Tlev=0 or 2

The basic BJT temperature compensation equations for beta and the saturation currents when Tlev=0 or 2 (default Tlev=0):

$$Bf(t) = Bf \cdot \left(\frac{t}{tnom}\right)^{Xtb}$$

$$Br(t) = Br \cdot \left(\frac{t}{tnom}\right)^{Xtb}$$

$$Ise(t) = \frac{Ise}{\left(\frac{t}{tnom}\right)^{Xtb}} \cdot e^{\frac{facln}{Ne}}$$

$$Isc(t) = \frac{Isc}{\left(\frac{t}{tnom}\right)^{Xtb}} \cdot e^{\frac{facln}{Nc}}$$

$$Iss(t) = \frac{Iss}{\left(\frac{t}{tnom}\right)^{Xtb}} \cdot e^{\frac{facln}{Ns}}$$

The parameter Xtb usually should be set to zero for Tlev=2.

$$Is(t) = Is \cdot e^{facln}$$

$$Ibe(t) = Ibe \cdot e^{\frac{facln}{Nf}}$$

$$Ibc(t) = Ibc \cdot e^{\frac{facln}{Nr}}$$

Tlev=0, 1 or 3

$$facln = \frac{Eg}{vt(tnom)} - \frac{Eg}{vt(t)} + Xti \cdot \ln\left(\frac{t}{tnom}\right)$$

Tlev=2

$$facln = \frac{egnom}{vt(tnom)} - \frac{eg(t)}{vt(t)} + Xti \cdot \ln\left(\frac{t}{tnom}\right)$$

Saturation and Temperature Equations, Tlev=1

The basic BJT temperature compensation equations for beta and the saturation currents when *Tlev*=1:

$$Bf(t) = Bf \cdot (1 + Xtb \cdot \Delta t)$$

$$Br(t) = Br \cdot (1 + Xtb \cdot \Delta t)$$

$$Ise(t) = \frac{Ise}{1 + Xtb \cdot \Delta t} \cdot e^{\frac{facln}{Ne}}$$

$$Isc(t) = \frac{Isc}{1 + Xtb \cdot \Delta t} \cdot e^{\frac{facln}{Nc}}$$

$$Iss(t) = \frac{Iss}{1 + Xtb \cdot \Delta t} \cdot e^{\frac{facln}{Ns}}$$

$$Is(t) = Is \cdot e^{facln}$$

$$Ibe(t) = Ibe \cdot e^{\frac{facln}{Nf}}$$

$$Ibc(t) = Ibc \cdot e^{\frac{facln}{Nr}}$$

where

$$facln = \frac{Eg}{vt(tnom)} - \frac{Eg}{vt(t)} + Xti \cdot \ln\left(\frac{t}{tnom}\right)$$

Tlev=0, 1, 2

The parameters *Ikf*, *Ikr*, and *Irb* are also modified as:

$$Ikf(t) = Ikf \cdot (1 + Tikf1 \cdot \Delta t + Tikf2 \cdot \Delta t^2)$$

$$Ikr(t) = Ikr \cdot (1 + Tikr1 \cdot \Delta t + Tikr2 \cdot \Delta t^2)$$

$$Irb(t) = Irb \cdot (1 + Tirb1 \cdot \Delta t + Tirb2 \cdot \Delta t^2)$$

Saturation Temperature Equations, Tlev=3

$$Is(t) = Is^{(1 + Tis1 \cdot \Delta t + Tis2 \cdot \Delta t^2)}$$

$$Ibe(t) = Ibe^{(1 + Tis1 \cdot \Delta t + Tis2 \cdot \Delta t^2)}$$

$$Ibc(t) = Ibc^{(1 + Tis1 \cdot \Delta t + Tis2 \cdot \Delta t^2)}$$

$$Ise(t) = Ise^{(1 + Tise1 \cdot \Delta t + Tise2 \cdot \Delta t^2)}$$

$$Isc(t) = Isc^{(1 + Tisc1 \cdot \Delta t + Tisc2 \cdot \Delta t^2)}$$

$$Iss(t) = Iss^{(1 + Tiss1 \cdot \Delta t + Tiss2 \cdot \Delta t^2)}$$

The parameters Ikf, Ikr, and Irb are also modified as:

$$Ikf(t) = Ikf^{(1+Tikf1\cdot\Delta t+Tikf2\cdot\Delta t^2)}$$

$$Ikr(t) = Ikr^{(1+Tikr1\cdot\Delta t+Tikr2\cdot\Delta t^2)}$$

$$Irb(t) = Irb^{(1+Tirb1\cdot\Delta t+Tirb2\cdot\Delta t^2)}$$

The following parameters are also modified when corresponding temperature coefficients are specified, regardless of the Tlev value.

$$Bf(t) = Bf\cdot(1+Tbf1\cdot\Delta t+Tbf2\cdot\Delta t^2)$$

$$Br(t) = Br\cdot(1+Tbr1\cdot\Delta t+Tbr2\cdot\Delta t^2)$$

$$Vaf(t) = Vaf\cdot(1+Tvaf1\cdot\Delta t+Tvaf2\cdot\Delta t^2)$$

$$Var(t) = Var\cdot(1+Tvar1\cdot\Delta t+Tvar2\cdot\Delta t^2)$$

$$Itf(t) = Itf\cdot(1+Titf1\cdot\Delta t+Titf2\cdot\Delta t^2)$$

$$Tf(t) = Tf\cdot(1+Ttf1\cdot\Delta t+Ttf2\cdot\Delta t^2)$$

$$Tr(t) = Tr\cdot(1+Ttr1\cdot\Delta t+Ttr2\cdot\Delta t^2)$$

$$Nf(t) = Nf\cdot(1+Tnf1\cdot\Delta t+Tnf2\cdot\Delta t^2)$$

$$Nr(t) = Nr\cdot(1+Tnr1\cdot\Delta t+Tnr2\cdot\Delta t^2)$$

$$Ne(t) = Ne\cdot(1+Tne1\cdot\Delta t+Tne2\cdot\Delta t^2)$$

$$Nc(t) = Nc\cdot(1+Tnc1\cdot\Delta t+Tnc2\cdot\Delta t^2)$$

$$Ns(t) = Ns\cdot(1+Tns1\cdot\Delta t+Tns2\cdot\Delta t^2)$$

$$Mje(t) = Mje\cdot(1+Tmje1\cdot\Delta t+Tmje2\cdot\Delta t^2)$$

$$Mjc(t) = Mjc\cdot(1+Tmjc1\cdot\Delta t+Tmjc2\cdot\Delta t^2)$$

$$Mjs(t) = Mjs\cdot(1+Tmjs1\cdot\Delta t+Tmjs2\cdot\Delta t^2)$$

Capacitance Temperature Equations

Tlevc=0

$$Cje(t) = Cje\cdot\left[1+Mje\cdot\left(4.0e{-}4\cdot\Delta t-\frac{Vje(t)}{Vje}+1\right)\right]$$

$$Cjc(t) = Cjc\cdot\left[1+Mjc\cdot\left(4.0e{-}4\cdot\Delta t-\frac{Vjc(t)}{Vjc}+1\right)\right]$$

$$Cjs(t) = Cjs\cdot\left[1+Mjs\cdot\left(4.0e{-}4\cdot\Delta t-\frac{Vjs(t)}{Vjs}+1\right)\right]$$

where

$$Vje(t) = Vje\cdot\frac{t}{tnom}-vt(t)\cdot\left[3\cdot\ln\left(\frac{t}{tnom}\right)+\frac{egnom}{vt(tnom)}-\frac{eg(t)}{vt(t)}\right]$$

$$Vjc(t) = Vjc\cdot\frac{t}{tnom}-vt(t)\cdot\left[3\cdot\ln\left(\frac{t}{tnom}\right)+\frac{egnom}{vt(tnom)}-\frac{eg(t)}{vt(t)}\right]$$

$$Vjs(t) = Vjs\cdot\frac{t}{tnom}-vt(t)\cdot\left[3\cdot\ln\left(\frac{t}{tnom}\right)+\frac{egnom}{vt(tnom)}-\frac{eg(t)}{vt(t)}\right]$$

Tlevc=1

$$Cje(t) = Cje \cdot (1 + Cte \cdot \Delta t)$$
$$Cjc(t) = Cjc \cdot (1 + Ctc \cdot \Delta t)$$
$$Cjs(t) = Cjs \cdot (1 + Cts \cdot \Delta t)$$

and contact potentials determined as:

$$Vje(t) = Vje - Tvje \cdot \Delta t$$
$$Vjc(t) = Vjc - Tvjc \cdot \Delta t$$
$$Vjs(t) = Vjs - Tvjs \cdot \Delta t$$

Tlevc=2

$$Cje(t) = Cje \cdot \left(\frac{Vje}{Vje(t)} \right)^{Mje}$$
$$Cjc(t) = Cjc \cdot \left(\frac{Vjc}{Vjc(t)} \right)^{Mjc}$$
$$Cjs(t) = Cjs \cdot \left(\frac{Vjs}{Vjs(t)} \right)^{Mjs}$$

where

$$Vje(t) = Vje - Tvje \cdot \Delta t$$
$$Vjc(t) = Vjc - Tvjc \cdot \Delta t$$
$$Vjs(t) = Vjs - Tvjs \cdot \Delta t$$

Tlevc=3

$$Cje(t) = Cje \cdot \left(1 - 0.5 \cdot dvjedt \cdot \frac{\Delta t}{Vje} \right)$$
$$Cjc(t) = Cjc \cdot \left(1 - 0.5 \cdot dvjcdt \cdot \frac{\Delta t}{Vjc} \right)$$
$$Cjs(t) = Cjs \cdot \left(1 - 0.5 \cdot dvjsdt \cdot \frac{\Delta t}{Vjs} \right)$$
$$Vje(t) = Vje + dvjedt \cdot \Delta t$$
$$Vjc(t) = Vjc + dvjcdt \cdot \Delta t$$
$$Vjs(t) = Vjs + dvjsdt \cdot \Delta t$$

where for **Tlev=0, 1** or **3**

$$dvjedt = -\frac{egnom + 3 \cdot vt(tnom) + (1.16 - egnom) \cdot \left(2 - \frac{tnom}{tnom + 1108} \right) - Vje}{tnom}$$
$$dvjcdt = -\frac{egnom + 3 \cdot vt(tnom) + (1.16 - egnom) \cdot \left(2 - \frac{tnom}{tnom + 1108} \right) - Vjc}{tnom}$$
$$dvjsdt = -\frac{egnom + 3 \cdot vt(tnom) + (1.16 - egnom) \cdot \left(2 - \frac{tnom}{tnom + 1108} \right) - Vjs}{tnom}$$

and for **Tlev=2**

$$dvjedt = -\frac{egnom + 3 \cdot vt(tnom) + (Eg - egnom) \cdot \left(2 - \frac{tnom}{tnom + Gap2}\right) - Vje}{tnom}$$

$$dvjcdt = -\frac{egnom + 3 \cdot vt(tnom) + (Eg - egnom) \cdot \left(2 - \frac{tnom}{tnom + Gap2}\right) - Vjc}{tnom}$$

$$dvjsdt = -\frac{egnom + 3 \cdot vt(tnom) + (Eg - egnom) \cdot \left(2 - \frac{tnom}{tnom + Gap2}\right) - Vjs}{tnom}$$

Parasitic Resistor Temperature Equations

The parasitic resistors, as a function of temperature regardless of Tlev value, are determined as:

$$Re(t) = Re \cdot (1 + Tre1 \cdot \Delta t + Tre2 \cdot \Delta t^2)$$

$$Rb(t) = Rb \cdot (1 + Trb1 \cdot \Delta t + Trb2 \cdot \Delta t^2)$$

$$Rbm(t) = Rbm \cdot (1 + Trm1 \cdot \Delta t + Trm2 \cdot \Delta t^2)$$

$$Rc(t) = Rc \cdot (1 + Trc1 \cdot \Delta t + Trc2 \cdot \Delta t^2)$$

BJT Level=2 Temperature Equations

The model parameters of BJT Level=2 model are modified for temperature compensation as:

$$Gamma(t) = Gamma \cdot e^{(facln)}$$

$$Rc(t) = Rc \cdot \left(\frac{t}{tnom}\right)^{Bex}$$

$$Vo(t) = Vo \cdot \left(\frac{t}{tnom}\right)^{Bexv}$$

MOSFETs

Level 3 MOS Model

The MOS model is the level 3 model from Berkeley SPICE, and is a semi-empirical model. Three charge models are available. MOS3 transistors require that you use a model statement.
The arguments are defined as:

W	Channel width
L	Channel length
AS	Area of source diffusion
AD	Area of drain diffusion
PS	Perimeter of source diffusion
PD	Perimeter of drain diffusion
NRD	Number of squares of drain diffusion
NRS	Number of squares of source diffusion
LD	Length of drain diffusion region
LS	Length of source diffusion region
M	Multiplicity factor (number of MOSFETs in parallel)

Equation Constants

Variable	Definition
Δt	t-tnom
esi	1.0359e-10 F/m dielectric constant of silicon
k	1.38062e-23 (Boltzmann's constant)
q	1.60212e-19 (electron charge)
t	New temperature of model or element in °K
tnom	tnom = TNOM + 273.15. This variable represents the nominal temperature of parameter measurements in °K (user input in °C).
vt	k*t/q
vt(tnom)	k*tnom/q

Model (group) parameters

Device type parameters

Name	Units	Default	Description
TYPE		n	Transistor type. Possible values are n or p.

Process parameters

Name	Units	Default	Description
NSUB	cm-3	1.13e16	Channel doping concentration. spectre-compatible.
NSS	cm-2	0	Surface state density
NFS	cm-2	0	Fast surface state density
TPG		+1	Type of gate (+1=opposite of substrate, -1=same as substrate, 0=aluminium)
TOX	m	1e-7	Gate oxide thickness. spectre-compatible.
LD	m	0	Lateral diffusion
WD	m	0	Field-oxide encroachment
XW	m	0	Width variation due to masking and etching
XL	m	0	Length variation due to masking and etching
XJ	m	0	Source/drain junction depth. spectre-compatible.

Name	Units	Default	Description
VTO	V	0	Threshold voltage at zero body bias
KP	A/ V2	2.0718e-5	Transconductance parameter
THETA	1/ V	0	Mobility modulation coefficient. spectre-compatible.
PHI	V	0.7	Surface potential at strong inversion. spectre-compatible.
Gamma	V1/2	0	Body-effect parameter
UO	cm2/ Vs	600	Carrier surface mobility. spectre-compatible.

VMAX	m/s		Maximum carrier saturation velocity
ETA	1/ V	0	Static feedback coefficient. spectre-compatible.
KAPPA		0.2	Saturation field factor
DELTA		0	Width effect on threshold voltage

Charge model selection parameters

Name	Units	Default	Description
XQC		0	Drain/source channel charge partition in saturation for charge models, e.g. use 0.4 for 40/60, 0.5 for 50/50, or 0 for 0/100. spectre-compatible.

Junction diode model parameters

Name	Units	Default	Description
JS	A/m^2		Bulk junction reverse saturation current density. spectre-compatible.
IS	A	1e-14	Bulk junction reverse saturation current. spectre-compatible.
N		1	Junction emission coefficient. spectre-compatible.

Name	Units	Default	Description
CBS	F	0	Bulk-source zero-bias junction capacitance. spectre-compatible.
CBD	F	0	Bulk-drain zero-bias junction capacitance. spectre-compatible.
CJ	F/m2	0	Zero-bias junction bottom capacitance density. spectre-compatible.
MJ		1/2	Bulk junction bottom grading coefficient. spectre-compatible.
PB	V	0.8	Bulk junction built-in potential. spectre -compatible.
FC		0.5	Forward-bias depletion capacitance threshold. spectre-compatible.
CJSW	F/m	0	Zero-bias junction sidewall capacitance density. spectre-compatible.

MJSW		1/3	Bulk junction sidewall grading coefficient. spectre-compatible.

Overlap capacitance parameters

Name	Units	Default	Description
CGSO	F/m	0	Gate-source overlap capacitance. spectre-compatible.
CGDO	F/m	0	Gate-drain overlap capacitance. spectre-compatible.
CGBO	F/m	0	Gate-bulk overlap capacitance. spectre-compatible.
METO	m	0	Metal overlap in fringing field. spectre-compatible.

Parasitic resistance parameters

Name	Units	Default	Description
RS	Ohm	0	Source resistance. spectre-compatible.
RD	Ohm	0	Drain resistance. spectre-compatible.
RSH	Ohm/ square	0	Source/drain diffusion sheet resistance. spectre-compatible.
RSC	Ohm	0	Source contact resistance. spectre-compatible.
RDC	Ohm	0	Drain contact resistance. spectre-compatible.
LDIF	m	0	Lateral diffusion beyond the gate. spectre-compatible.
HDIF	m	0	Length of heavily doped diffusion. spectre-compatible.

Default instance parameters

Name	Units	Default	Description
W	m	3e-6	Default channel width. spectre-compatible.
L	m	3e-6	Default channel length. spectre-compatible.
AS	m2	0	Default area of source diffusion. spectre-compatible.

AD	m2	0	Default area of drain diffusion. spectre-compatible.
PS	m	0	Default perimeter of source diffusion. spectre-compatible.
PD	m	0	Default perimeter of drain diffusion. spectre-compatible.
NRD	m/m	0	Default number of squares of drain diffusion. spectre-compatible.
NRS	m/m	0	Default number of squares of source diffusion. spectre-compatible.

Auto model selector parameters

Name	Units	Default	Description
WMAX	m	1	Maximum channel width for which the model is valid. spectre-compatible.
WMIN	m	0	Minimum channel width for which the model is valid. spectre-compatible.
LMAX	m	1	Maximum channel length for which the model is valid. spectre-compatible.
LMIN	m	0	Minimum channel length for which the model is valid. spectre-compatible.

Auto Model Selection

Model selector is used for libraries with multiple models of a specific element. Many models need to be characterized for different geometries in order to obtain accurate results for model development. The model selector program automatically searches for a model with the length and width range specified in the instance statement and uses this model in the simulations. For the auto model selector program to find a specific model, the models to be searched should be grouped together within braces. Such a group is called a model group. An opening brace is required at the end of the line defining each model group. Every model in the group is given a name followed by a colon and the list of parameters. Also, the four geometric parameters LMAX, LMIN, WMAX, and WMIN should be given. The selection criteria to choose a model is as follows:

$LMIN \le Element\_LENGHT \le LMAX$ and $WMIN \le Element\_WIDTH \le WMAX$

During automatic model selection parameter BULK must be the same for all models in model group.

Example of the model selector syntax:

model_root_name nd ng ns <nb> cell_name <<L=>length> <<W=>width>

The following is an example of SymSpice syntax for MOSFET models:

*MOS1 drain gate source bulk MOSFET W=2u L=1u *element statement*

.MODEL MOSFET2 NMOS WMIN=1.5u WMAX=3u LMIN=0.8u LMAX=2u

.MODEL MOSFET3 NMOS WMIN=1.5u WMAX=3u LMIN=2u LMAX=6u

The auto model MOSFET2 will be selected as it fits within the length and width ranges specified in the element statement.

Temperature effects parameters

Name	Units	Default	Description
TLEV		0	DC temperature selector. spectre -compatible.
TLEVC		0	AC temperature selector. spectre -compatible.
EG	V	1.12452	Potential resulting from energy band gap. spectre-compatible.
GAP1	V/°C	7.02e-4	Band gap temperature coefficient. spectre-compatible.
GAP2	°C	1108	Band gap temperature offset. spectre -compatible.
F1EX		0	Temperature exponent for ucrit
LAMEX	1/°C	0	Temperature parameter for lambda and kappa
TRS	1/°C	0	Temperature parameter for source resistance. spectre-compatible.
TRD	1/°C	0	Temperature parameter for drain resistance. spectre-compatible.
XTI		3	Saturation current temperature exponent. spectre-compatible.
PTC	V/°C	0	Surface potential temperature coefficient. spectre-compatible.
TCV	V/°C	0	Threshold voltage temperature coefficient. spectre-compatible.
PTA	V/°C	0	Junction potential temperature coefficient. spectre-compatible.
PTP	V/°C	0	Sidewall junction potential temperature coefficient. spectre-compatible.

CTA	1/°C	0	Junction capacitance temperature coefficient. spectre-compatible.
CTP	1/°C	0	Sidewall junction capacitance temperature coefficient. spectre -compatible.

MOS Level 3 Model Equations

Channel Width and Length

$$W_{scaled} = W \cdot SCALE + XW \cdot SCALEM$$

$$W_{eff} = W \cdot SCALE + XW \cdot SCALEM - 2 \cdot WD \cdot SCALEM$$

$$L_{eff} = L \cdot SCALE + XL \cdot SCALEM - 2 \cdot LD \cdot SCALEM$$

Threshold Voltage

$$V_{TH} = V_{BI} + GAMMA \cdot F_S \sqrt{PHI - V_{BS}} - \sigma V_{DS} + F_N (PHI - V_{BS})$$

where

$$V_{BI} = VTO - GAMMA \sqrt{PHI}$$

$$\sigma = \frac{8.15 \times 10^{-22} ETA}{C_{ox} Leff^3}$$

$$F_S = 1 - \frac{X_j}{L_{eff}} \left\{ \left(\frac{LD}{XJ} + \frac{W_C}{XJ} \right) \sqrt{1 - \left(\frac{W_p}{1 + W_p} \right)^2} - \frac{LD}{XJ} \right\}$$

$$W_p = \frac{X_D}{XJ} \sqrt{PHI - V_{BS}}$$

$$X_D = \sqrt{\frac{2\varepsilon_{si}}{Q \cdot NSUB}}$$

$$\frac{W_c}{XJ} = 0.0631353 + 0.8013292 W_p - 0.01110777 W_p^2$$

$$F_N = \frac{DELTA \cdot \pi \cdot \varepsilon_{si}}{2 C_{ox} W_{eff}}$$

If either *XJ* or *NSUB* is zero, the short-channel effects on threshold voltage are not evaluated; that is, $F_S = 1$.

Drain Saturation Voltage

If V_{MAX} is not input, V_{DSAT} is determined by the pinchoff condition and is given by

$$V_{DSAT} = \frac{V_{GST}}{1 + F_B}$$

where

$$V_{GST} = V_{GS} - V_{TH}$$

$$F_B = \frac{GAMMA \cdot F_S}{4\sqrt{PHI - V_{BS}}} + F_N$$

If V_{MAX} is specified, V_{DSAT} is determined by the velocity saturation effect.

$$V_{DSAT} = \frac{V_{GST}}{1+F_B} + E_C L_{eff} - \sqrt{\left(\frac{V_{GST}}{1+F_B}\right)^2 + (E_C L_{eff})^2}$$

where

$$E_C = \frac{V_{MAX}}{\mu_{eff}}$$

μ_{eff} is the effective mobility that is defined later.

Drain Current for the Subthreshold Region

Note: These equations apply when $V_{GS} \le V_{ON}$.

You cannot use the subthreshold current equations without *NFS*.

$$V_{ON} = \begin{cases} V_{TH} + nV_t & \text{if } NFX \text{ is specified} \\ V_{TH} & \text{otherwise} \end{cases}$$

where

$$n = 1 + \frac{C_{FS} + C_D}{C_{ox}}$$

$$C_{FS} = Q(NFS)$$

$$C_D = \left[\frac{\gamma_s}{2\sqrt{PHI - V_{BS}}} - \frac{d\gamma_s}{dV_{BS}}\sqrt{PHI - V_{BS}} + F_N\right] C_{ox}$$

$$I_{DS} = I_{DS,ON} e^{(V_{GS} - V_{ON})/nV_t}$$

where *IDS, ON* is the drain current evaluated at $V_{GS} = V_{ON}$.

Drain Current for the Triode Region

Note: These equations apply when $V_{GS} \ge V_{ON}$ and $V_{DS} \le V_{DSAT}$.

$$DS = \frac{\beta\left[V_{GS} - V_{TH} - \frac{(1+F_B)}{2}V_{DS}\right]V_{DS}}{1 + \frac{\mu_{eff} V_{DS}}{V_{MAX} \cdot L_{eff}}}$$

where

$$\beta = \frac{\mu_{eff} C_{ox} W_{eff}}{L_{eff}}$$

$$\mu_{eff} = \frac{UO}{1 + THETA(V_{GS} - V_{TH})}$$

Drain Current for the Saturation Region

Note: These equations apply when $V_{GS} \ge V_{ON}$ and $V_{DS} \ge V_{DSAT}$.

$$I_{DS} = \frac{\beta\left[V_{GS} - V_{TH} - \frac{(1+F_B)}{2}V_{DSAT}\right]\cdot V_{DSAT}}{1+\frac{\mu_{eff}V_{DSAT}}{V_{MAX}\cdot L_{eff}}}$$

where

$$\beta = \frac{\mu_{eff}C_{ox}W_{eff}}{L_{eff}-\Delta L}$$

$$\Delta L = \begin{cases} \sqrt{KAPPA\cdot X_D^2(V_{DS}-V_{DSAT})} & \text{if } V_{MAX} \text{ is not specified} \\ \sqrt{E_x^2 + KAPPA\cdot X_D^2(V_{DS}-V_{DSAT})} - E_x & \text{otherwise} \end{cases}$$

$$E_x = \frac{E_p X_D^2}{2}$$

$$E_p = \frac{1+U_1V_{DSAT}}{L_{eff}\cdot U_1^2 V_{DSAT}}$$

$$U_1 = \frac{\mu_{eff}}{L_{eff}\cdot V_{MAX}}$$

If $\Delta L = \frac{L_{eff}}{2}$, a new value of ΔL (ΔL_{new}) is calculated to avoid device punch-through.

$$\Delta L_{new} = L_{eff} - \frac{L_{eff}^2}{4\Delta L}$$

Substrate Current

The substrate current results from impact ionization in the velocity saturation region near the drain. This impact-ionization induced current *(IDB)* flows between the drain and the substrate. You need both *aio* and *bio* to use the impact-ionization model.

$$I_{DB} = \begin{cases} I_{DS}A_i(V_{DS}-V_{DSAT})e^{-B_i/(V_{DS}-V_{DSAT})} & if\ V_{DS} \le V_{DSAT} \\ 0 & otherwise \end{cases}$$

where

$$A_i = AIO + \frac{LAIO\times 10^{-6}}{L_{eff}} + \frac{WAIO\times 10^{-6}}{W_{eff}}$$

$$B_i = BIO + \frac{LBIO\times 10^{-6}}{L_{eff}} + \frac{WBIO\times 10^{-6}}{W_{eff}}$$

Level 49 and 53 BSIM 3v3. MOS Models

BSIM 3v3 is the latest industry-standard MOSFET model for deep-submicron digital and analog circuit designs from the BSIM Group at the University of California at Berkeley.

The development of BSIM3v3 is based on Poisson's equations using gradual channel approximation and coherent quasi 2D analysis, taking into account the effects of device geometry and process parameters. BSIM 3v3 considers the following physical phenomena observed in MOSFET devices:

- Short and narrow channel effects on threshold voltage.
- Non-uniform doping effect (in both lateral and vertical directions).
- Mobility reduction due to vertical field.
- Bulk charge effect.
- Velocity saturation.
- Drain-induced barrier lowering (DIBL).
- Channel length modulation (CLM).
- Substrate current induced body effect (SCBE).
- Subthreshold conduction.
- Source/drain parasitic resistances.

Select BSIM 3v3 model for channel length about 0.15u micron. Minimum tox for BSIM 3v3 model is 4nm.

This MOSFET model is compatible with the model of level=49 (53) from SPICE and BSIM 3v3 from SymSpice.

Syntax and Individual Parameters

Version 3.2

In SymSpice models Level 49 and Level 53 were combained in one BSIM 3v3 MOS model.
In June, 1998 Berkeley released BSIM 3 Version 3.2, which contains many new features (see http://www-device.eecs.berkeley.edu/~bsim3/).

Syntax in 'local' or 'spectre' modes

without model:

Name <(>nd ng ns nb<)> bsim3v3 parameter=value ...

with model:

Name <(>nd ng ns nb<)> modelname parameter=value ...

model modelname bsim3v3 pname=val

Syntax in 'spice' mode

Mxxx nd ng ns <nb> mname

+<<L=>length> <<W=>width>

+<AD=val> <AS=val> <PD=val>

+<PS=val> <NRD=val> <NRS=val>

+<RDC=val> <RSC=val>

+<IC=vds,vgs,vbs> <M=val>

+ <DTEMP=val> <GEO=val>

+<DELVTO=val>

or

Mxxx nd ng ns <nb> mname

+<width> <length> <other options...>

where:

SPICE mode	SymSpice mode	Default	Desciption
Mxxx	Name	-	BSIM 3v3 element name. For SPICE format netlist it must begin with "M", which can be followed by alphanumeric charactesrs.
mname	bsim3v3/modelname	-	BSIM3v3 model name reference.
L	l	defl	MOSFET channel length
W	w	defw	MOSFET channel width
AD	ad	defad	Drain diffusion area
AS	as	defas	Source diffusion area
PD	pd	defpd	Perimiter of the drain junction, including the channel edge

PS	ps	defps	Perimiter of the source junction, including the channel edge
NRD	nrd	defnrd	Number of squares of drain diffusion for resistance calcualtions
NRS	nrs	defnrs	Number of squares of source diffusion for resistance calcualtions
RDC	rdc	-	Additional drain resistance due to contact resistance
RSC	rsc	-	Additional source resistance due to contact resistance
M	m	1	Multiplier to simulate multiple MOSFETs in parallel
DTEMP	trise	0.0	The difference between the element temperature and the circuit temperature
GEO	geo	0	Source/drain sharing selector for MOSFET model parameter value ACM=3
DELVTO	delvto	0.0	Zero-bias threshold voltage shift
NQSMOD	nqsmod	-	NQS flag
SA1	sa1	-	Irregular LOD device geometry parameter
SB1	sb1	-	Irregular LOD device geometry parameter
SW1	sw1	-	Irregular LOD device geometry parameter
SA2	sa2	-	Irregular LOD device geometry parameter
SB2	sb2	-	Irregular LOD device geometry parameter
SW2	sw2	-	Irregular LOD device geometry parameter
SA3	sa3	-	Irregular LOD device geometry parameter
SB3	sb3	-	Irregular LOD device geometry parameter
SW3	sw3	-	Irregular LOD device geometry parameter
SA4	sa4	-	Irregular LOD device geometry parameter
SB4	sb4	-	Irregular LOD device geometry parameter
SW4	sw4	-	Irregular LOD device geometry parameter
SA5	sa5	-	Irregular LOD device geometry parameter
SB5	sb5	-	Irregular LOD device geometry parameter
SW5	sw5	-	Irregular LOD device geometry parameter
SA6	sa6	-	Irregular LOD device geometry parameter
SB6	sb6	-	Irregular LOD device geometry parameter
SW6	sw6	-	Irregular LOD device geometry parameter

SA7	sa7	-	Irregular LOD device geometry parameter
SB7	sb7	-	Irregular LOD device geometry parameter
SW7	sw7	-	Irregular LOD device geometry parameter
SA8	sa8	-	Irregular LOD device geometry parameter
SB8	sb8	-	Irregular LOD device geometry parameter
SW8	sw8	-	Irregular LOD device geometry parameter
SA9	sa9	-	Irregular LOD device geometry parameter
SB9	sb9	-	Irregular LOD device geometry parameter
SW9	sw9	-	Irregular LOD device geometry parameter
SA10	sa10	-	Irregular LOD device geometry parameter
SB10	sb10	-	Irregular LOD device geometry parameter
SW10	sw10	-	Irregular LOD device geometry parameter

Level 49 and 53 BSIM 3v3 Model (Group) Parameters

Model Control Parameters

Parameter name	Units	Default	Description
version		3.2	BSIM 3v3 version selector
mobmod		1	Mobility model selector.
capmod		3	Capacitance model selector.
nqsmod		0	NQS model selector.
paramchk		1	Flag parameter for parameter cheking
bnunit		1	Flag parameter for the units of size parameters in binning approach

Parameter Binning

For SPICE netlist the expressions for parameter binning is followings.
Parameter binning is supported in the Berkeley BSIM3v3 release through the specification of Lwp parameters. That is, a subset of model parameters can be bilinearly interpolated over 1/Leff and 1/Weff by specifying four terms: the parameter Xo, a length term Xl, a width term Xw, and a product term Xp. The parameter value at a given L, W is then interpolated as:

$$X = Xo + \frac{Xl}{Leff} + \frac{Xw}{Weff} + \frac{Xp}{Leff \cdot Weff}$$

SymSpice adds parameters Lmin, Lmax, Wmin, Wmax and Lref, Wref to allow multiple cell binning. Lmin, Lmax, Wmin, Wmax define the cell boundary. Lref, Wref are offset values that provide a convenient interpolation scheme. Lref, Wref offsets are used when both values are defined and the model parameter Binflag > 0.9 is specified. The parameter value at a given L, W is then interpolated as:

$$X = X_0 + Xl \cdot \left(\frac{1}{Leff} - \frac{1}{Lref}\right) + Xw \cdot \left(\frac{1}{Weff} - \frac{1}{Wref}\right) + \frac{Xp}{\left(\frac{1}{Leff} - \frac{1}{Lref}\right) \cdot \left(\frac{1}{Weff} - \frac{1}{Wref}\right)}$$

For SymSpice compatible SymSpice uses the following expressions.

The binning equation is given by

$$P = P0 + Pl/Leff + Pw/Weff + Pp/(Leff \cdot Weff)$$

Only the P0 parameters are listed. Pl, Pw, and Pw are not shown but can be recognized. The names of Pl, Pw, and Pp are identical to that of P0 but with a prefix of l, w and p, respectively. BSIM3v3 transistors require that you use a model statement.

Note: For more information on this model You can consult the University of California at Berkeley BSIM3 home page at
http://www-device.eecs.berkeley.edu/~bsim3/index.html

Basic Model Parameters

Parameter name	Units	Default	Bin	Description
tox	m	1.5E-8		Gate oxide thickness
toxm	m	Tox		Normal tox at which parameter are extracted
xj	m		+	Junction depth
nch	$1/cm^3$		+	Channel doping concentration
nsub	$1/cm^3$		+	Substrate doping concentration
gamma1	$V^{1/2}$	Calculated	+	Body-effect coefficient near the interface
gamma2	$V^{1/2}$	Calculated	+	Body-effect coefficient in the bulk
xt	m	1.55E-7	+	Doping depth
vbx	V	0	+	vsb at which the depletion width equals xt
php	V			Sidewall bulk junction contact potential (used only with acm=0-3

cjgate	F/m	-		Zero-bias gate-edge sidewall bulk junction capacitance (used only with ACM=3!)
n		1		Emission coefficient,(used only with acm=0-3
cbd	F			Zero bias bulk-drain junction capacitance. Used only when cj and cjsw are 0 (used only with acm=0-3)
cbs	F			Zero bias bulk-source junction capacitance. Used only when cj and cjsw are 0 (used only with acm=0-3)
tt	sec	0		Transit time (used only with acm=0-3)
rdc	Ohm	0.0		Additional drain resistance due to contact resistance
lrd	Ohm/m			Drain resistance length sensitivity
wrd	Ohm/m			Drain resistance length sensitivity (used with lrd)
prd	Ohm/m^2			Drain resistance product (area) sensitivity (used with lrd)
lrs	Ohm/m			Source resistance length sensitivity. Use this parameter with automatic model selection in conjunction with wrs and prs to factor model for device size.
wrs	Ohm/m			Source resistance width sensitivity (used with lrs)
prs	Ohm/m^2			Source resistance product (area) sensitivity (used with lrs)
rsc	Ohm	0.0		Additional source resistance due to contact resistance
update	Ohm	0.0		Selector for rs and rd to be compatible with aspec
ld	m			Lateral diffusion into channel from source and drain diffusion
calcacm		0		Selector for calculate areas and perimeters. SymSpice equations for SPICE netlist are used if calcacm=1. Selector is available if acm=12
noimod		1		Noise model selector
trise	C°	0		Temperature rise from ambient
lgcs	m	0		Gate-to-contact length of source side
lgcd	m	0		Gate-to-contact length of drain side
rss	Ohm*m	0		Scalable source resistance
rdd	Ohm*m	0		Scalable drain resistance
sc	m			Spacing between contacts
minr	Ohm	0.1		Minimum source/drain resistance
jmelt	A/m^2			Explosion current density
meto	m	0		Metal overlap in fringing field

fc		0.5		Foward bias depletion capacitance threshold
fcsw				Sidewall foward bias depletion capacitance threshold
w	m	5e-6		Default channel width
l	m	5e-6		Default channel length
as	m^2			Default area of source diffusion
ad	m^2			Default area of drain diffusion
ps	m			Default perimeter of source diffusion
pd	m			Default perimeter of drain diffusion
nrd	m/m			Default number of squares of drain diffusion
nrs	m/m			Default number of squares of source diffusion
tlev		0		DC temperature selector
tlevc		0		AC temperature selector
eg	V	1.12452		Energy band gap
gap1	V/C°	7.02e-4		Band gap temperature coefficient
gap2	V	1108		Band gap temperature offset
trs	1/C°	0		Temperature parameter for source resistance
trd	1/C°	0		Temperature parameter for drain resistance
pta	V/K	0		Junction potential temperature coefficient
ptp	V/K	0		Sidewall junction potential temperature coefficient
cta	1/K	0		Junction capacitance temperature coefficient
ctp	1/K	0		Sidewall junction capacitance temperature coefficient
imax	A	1		Maximum allowable current
jmax	A/m^2	1e8		Maximum allowable current density
bvj	V			Junction reverse breakdown voltage
wnoi	m	1e-5		Chanel width at which noise parameters were extracted
flkmod		0		Flicker noise model
diomod		1		Diode model switch. diomod=0 Berkeley junction model is used

Parameters for substrate current model

Parameter name	Units	Default	Description
alpha0	m/V	0	The first parameter of impact ionization current

lalpha0	m^2/V	0	Length dependence of the first parameter of impact ionization current
walpha0	m^2/V	0	Width dependence of the first parameter of impact ionization current
palpha0	m^3/V	0	Cross term dependence of the first parameter of impact ionization current
alpha1	1/V	0	The length scaling parameter of substrate current model
lalpha1	m/V	0	Length dependence of the length scaling parameter of substrate current model
walpha1	m/V	0	Width dependence of the length scaling parameter of substrate current model
pAlpha1	m^2/V	0	Cross term dependence of the length scaling parameter of substrate current model
beta0	V	30	The second parameter of impact ionization current
lbeta0	m*V	0	Length dependence of the second parameter of impact ionization current
wbeta0	m*V	0	Width dependence of the second parameter of impact ionization current
pbeta0	$m^2 \cdot V$	0	Cross term dependence of the second parameter of impact ionization current

Parameters for effective channel length/width in I-V model

Parameter name	Units	Default	Description
wln		1.0	Power of length dependence of width offset
wl	m^{Wln}	0.0	Coefficient of length dependence for width offset
wwn		1.0	Power of width dependence of width offset
ww	m^{Wwn}	0.0	Coefficient of width dependence for width offset
wwl	$m^{(Wwn+Wln)}$	0.0	Coefficient of length and width cross term for width offset
lln		1.0	Power of length dependence for length offset
ll	m^{Lln}	0.0	Coefficient of length dependence for length offset
lwn		1.0	Power of width dependence for length offset
lw	m^{Lwn}	0.0	Coefficient of width dependence for length offset
lwl	$m^{(Lwn+Lln)}$	0.0	Coefficient of length and width cross term for length offset

Parameters for Threshold voltage model

Parameter name	Units	Default	Bin	Description
vth0(vtho)	V	Calculated	+	Treshold voltage for large l
vfb	V	Calculated	+	Flat band voltage
k1	$V^{1/2}$	Calculated	+	First-order body effect coefficient
k2		Calculated	+	Second-order body effect coefficient
k3		80.0	+	Narrow width coefficient
k3b	1/V	0	+	Body effect coefficient of k3
w0(wo)	m	2.5E-6	+	Narrow width parameter
nlx	m	1.74E-7	+	Lateral non-uniform doping coefficient
dvt0w		0	+	First coefficient of narow width effect on vth at small l
dvt1w	1/m	5.3E6	+	Second coefficient of narow width effect on vth at small l
dvt2w	1/V	-0.032	+	Body-bias coefficient of narow width effect on vth at small l
dvt0		2.2	+	First coefficient of short-channel effect on vth
dvt1		0.53	+	Second coefficient of short-channel effect on vth
dvt2	1/V	-0.032	+	Body-bias coefficient of short-channel effect on vth
vbm	V	-3	+	Maximum applied body bias in vth calculation

Parameters for I-V model

Parameter name	Units	Default	Bin	Description
u0(uo,ubo)	$cm^2/(V \cdot s)$	Calculated	+	Mobility at t=tnom
ua	m/V	2.25E-9	+	First-order mobility degradation coefficient
ub	$(m/V)^2$	5.87E-19	+	Second-order mobility degradation coefficient

uc	m/V^2 or 1/V	Calculated	+	Body-effect of mobility degradation coefficient
vsat	m/sec	8E4	+	Saturation velocity at t=tnom
a0		1.0	+	Bulk charge effect coefficient
ags	1/V	0.0	+	Gate bias coefficient of the bulk charge effect
b0	m	0.0	+	Bulk charge effect coefficient for channel width
b1	m	0.0	+	Bulk charge effect width offset
keta	1/V	-0.047	+	Body-bias coefficient of bulk charge effect
a1	1/V	0.0	+	First non-saturation effect parameter
a2		1.0	+	Second non-saturation effect parameter
rdsw	Ohm-micron	0.0	+	Parasitic resistance per unit width
prwg	1/V	0	+	Gate bias effect coefficient of rdsw
prwb	$1/V^{1/2}$	0	+	Body effect coefficient of rdsw
wr		1.0	+	Width offset from weff for rds calculation
wint	m	0.0		Width offset fitting parameter from I-V without bias effect
lint	m	0.0		Length offset fitting parameter from I-V without bias effect
dwg	m/V	0.0	+	Coefficient of weff's gate dependence
dwb	$m/V^{1/2}$	0.0	+	Coefficient of weff's substrate body bias dependence
voff	V	-0.08	+	Offset voltage in the subthreshold region at large w and l
nfactor		1.0	+	Subthreshold swing factor
eta0(etao)		0.08	+	Dibl coefficient in subthreshold region
etab	1/V	-0.07	+	Body-bias coefficient for the subthreshold dibl effect
pclm		1.3	+	Channel length modulation parameter
pdiblc1		0.39	+	First output resistance dibl effect correction parameter
pdiblc2		0.0086	+	Second output resistance dibl effect correction parameter
pdiblcb	1/V	0	+	Body effect coefficient of dibl correction parameters

drout		0.56	+	i dependence coefficient of the dibl correction parameter in rout
pscbe1	V/m	4.24E8	+	First substrate current body-effect parameter
pscbe2	m/V	1E-5	+	Second substrate current body-effect parameter
pvag		0.0	+	Gate dependence of early voltage
delta	V	0.01	+	Effective vds parameter
ngate	cm^{-3}	0	+	Poly gate doping concentration
dsub		drout	+	dibl coefficient exponent in subthreshold region
Cct	F/m^2	0.0	+	Interface trap capacitance
cdsc	F/m^2	2.4E-4	+	Drain/Source to channel coupling capacitance
cdscd	$F/(V \cdot m^2)$	0.0	+	Drain-bias sensitivity of cdsc
cdscb	$F/(V \cdot m^2)$	0.0	+	Body-bias sensitivity of cdsc
php	V			Sidewall bulk junction contact potential (used only with acm=0-3)
cjgate	F/m			Zero-bias gate-edge sidewall bulk junction capacitance(used only with acm=3!)
n		1		Emission coefficient,(used only with acm=0-3)
cbd	F			Zero bias bulk-drain junction capacitance. Used only when cj and cjsw are 0 (used only with acm=0-3)
cbs	F			Zero bias bulk-source junction capacitance. Used only when cj and cjsw are 0 (used only with acm=0-3)
tt	sec	0		Transit time (used only with acm=0-3)
rdc	Ohm	0.0		Additional drain resistance due to contact resistance
lrd	Ohm/m			Drain resistance length sensitivity
wrd	Ohm/m			Drain resistance length sensitivity (used with lrd)
prd	Ohm/m^2			Drain resistance product (area) sensitivity (used with lrd)
lrs	Ohm/m			Source resistance length sensitivity. Use this parameter with automatic model selection in conjunction with wrs and prs to factor model for device size
wrs	Ohm/m			Source resistance width sensitivity (used with lrs)

prs	Ohm/m^2			Source resistance product (area) sensitivity (used with lrs)
rsc	Ohm	0.0		Additional source resistance due to contact resistance
update	Ohm	0.0		Selector for rs and rd to be compatible with aspec
ld	m			Lateral diffusion into channel from source and drain diffusion
calcacm		0		Selector for calculate areas and perimeters. Symspice equations for spice netlist are used if calcacm=1. Selector is available if acm=12
noimod		1		Noise model selector
trise	C°	0		Temperature rise from ambient
lgcs	m	0		Gate-to-contact length of source side
lgcd	m	0		Gate-to-contact length of drain side
rss	Ohm*m	0		Scalable source resistance
rdd	Ohm*m	0		Scalable drain resistance
sc	m			Spacing between contacts
minr	Ohm	0.1		Minimum source/drain resistance
jmelt	A/m^2			Explosion current density
meto	m	0		Metal overlap in fringing field
fc		0.5		Foward bias depletion capacitance threshold
fcsw				Sidewall foward bias depletion capacitance threshold
w	m	5e-6		Default channel width
l	m	5e-6		Default channel length
as	m^2			Default area of source diffusion
ad	m^2			Default area of drain diffusion
ps	m			Default perimeter of source diffusion
pd	m			Default perimeter of drain diffusion
nrd	m/m			Default number of squares of drain diffusion
nrs	m/m			Default number of squares of source diffusion
tlev		0		DC temperature selector
tlevc		0		AC temperature selector
eg	V	1.12452		Energy band gap
gap1	V/C°	7.02e-4		Band gap temperature coefficient

gap2	V	1108		Band gap temperature offset
trs	1/C°	0		Temperature parameter for source resistance
trd	1/C°	0		Temperature parameter for drain resistance
pta	V/K	0		Junction potential temperature coefficient
ptp	V/K	0		Sidewall junction potential temperature coefficient
cta	1/K	0		Junction capacitance temperature coefficient
ctp	1/K	0		Sidewall junction capacitance temperature coefficient
imax	A	1		Maximum allowable current
jmax	A/m^2	1e8		Maximum allowable current density
bvj	V			Junction reverse breakdown voltage
wnoi	m	1e-5		Chanel width at which noise parameters were extracted
flkmod		0		Flicker noise model
diomod		1		Diode model switch. diomod=0 Berkeley junction model is used

Parameters for capacitance model

Parameter name	Units	Default	Bin	Description
xpart		1		Charge partitioning flag
cgso(cgs,cgs0,c1)	F/m	Calculated		Non ldd region source-gate overlap capacitance per channel length calculated
cgdo(cgd,cgd0,c2)	F/m	Calculated		Non ldd region drain-gate overlap capacitance per channel length calculated
cgbo(cgb,cgb0)	F/m			Gate bulk overlap capacitance per unit channel length
cgsl	F/m	0.0	+	Light doped source-gate region overlap capacitance
cgdl	F/m	0.0	+	Light doped drain-gate region overlap capacitance
ckappa	V	0.6	+	Coefficient for lightly doped region overlap capacitance Fringing field capacitance
cf	F/m	Calculated	+	Fringing field capacitance calculated
clc	m	1E-7	+	Constant term for the short channel model

cle		0.6	+	Exponential term for the short channel model
dlc	m			Length offset fitting parameter from C-V
dwc	m	wint		Width offset fitting parameter from C-V
vfbcv	V	-1	+	Flat-band voltage parameter (for capmod=0 only)
noff		1.0	+	C-V parameter for vgsteff
voffcv	V	0.0	+	Offset voltage parameter of vth from week to strong inversion in C-V model
acde	m/V	1.0	+	Exponential coefficient for charge thickness in capmod=3 for accumulation and depletion regions
moin		15.0	+	Coefficient for the gate-bias dependent surface potential

Parameters for NQS model

Parameter name	Units	Default	Bin	Description
elm		5.0	+	elmore constant of the channel

Parameters for models of temperature effects

Parameter name	Units	Default	Bin	Description
tref(tnom)	C°	tnom		Temperature at which parameters are extracted
prt	Ohm-micron	0.0	+	Temperature coefficient for rdsw
ute		-1.5	+	Mobility temperature exponent
kt1	V	-0.11	+	Temperature coefficient for threshold voltage
kt1l	V*m	0.0	+	Channel length dependence of the temperature coefficient for threshold voltage
kt2		0.022	+	Body-bias coefficient of vth temperature effect
ua1	m/V	4.31E-9	+	Temperature coefficient for ua
ub1	$(m/V)^2$	-7.61E-18	+	Temperature coefficient for ub
uc1	1/V	Calculated	+	Temperature coefficient for uc mobmod=1,2 or mobmod=3
at	m/sec	3.3E4	+	Temperature coefficient for saturation velocity

xti				Junction current temperature exponent coefficient
tcj	1/K	0.0		Temperature coefficient of cj
tcjsw	1/K	0.0		Temperature coefficient of cjsw
tcjswg	1/K	0.0		Temperature coefficient of cjswg
tpb	V/K	0.0		Temperature coefficient of pb
tpbsw	V/K	0.0		Temperature coefficient of pbsw
tpbswg	V/K	0.0		Temperature coefficient of pbswg

Parameters for models of parasitic components

Parameter name	Units	Default	Description
rsh(rl)	Ohm/sqr	0	Source drain sheet resistance
js	A/m^2		Saturation current density of bottom junction diode
jsw(isp,isw)	A/m	0	Saturation current density of sidewall junction diode
nj		1	Emission coefficient of junction
ijth(imelt)	A		Diode limiting current
cj(cjo,cj0,cdb,csb, cja)	F/m^2		Source/Drain bottom junction capacitance per unit area at zero bias
mj(exa,exj,exs,exd)		0.5	Source/Drain Bottom junction capacitance grating coefficient
pb(pha,phs,phd, phib)	V		Bottom junction built-in potential
cjsw(cjp)	F/m		Source/Drain sidewall junction capacitance per unit length at zero bias
mjsw(exp)		0.33	Source/Drain sidewall junction capacitance grading coefficient
pbsw	V		Source/drain sidewall junction built-in potential
cjswg	F/m		Source/drain gate edge sidewall junction capacitance per unit length at zero bias
mjswg		mjsw	Source/drain gate sidewall junction capacitance grading coefficient
pbswg	V		Built-in potential of the source/drain gate edge sidewall junction

Length dependent parameters

Parameter name	Units	Default	Description
xl(dl,ldel)	m		Length variation due to masking and etching

Width dependent parameters

Parameter name	Units	Default	Description
xw(wdel, dw)	m		Width variation due to masking and etching
ldif	m		Lateral diffusion beyond the gate
hdif	m		Length of heavily doped diffusion
lref	m	0.0	Reference channel length
wref	m	0.0	Reference channel width
lmlt		1.0	Length shrink factor
wmlt		1.0	Width shrink factor
rs	Ohm		Source resistance
rd	Ohm		Drain resistance
binflag		0.0	Uses wref, lref when set > 0.9
xlref	m		Difference between physical (on wafer) and drawn reference channel length
xwref	m		Difference between physical (on wafer) and drawn reference channel width
wmax	m	1.0	Maximum channel width for which the model is valid
wmin	m	0.0	Minimum channel width for which the model is valid
lmax	m	1.0	Maximum channel length for which the model is valid
lmin	m	0.0	Minimum channel length for which the model is valid

Calculating effective saturation currents

For SPICE netlist SymSpice uses the following formulas.
Source diode saturation current

Define:

$val = Js \cdot ASeff + JSW \cdot PSeff$

If $val > 0$ then, $Isbs = val$

Otherwise, $Isbs = M \cdot IS$

Drain diode saturation current

Define:

$val = Js \cdot ADeff + JSW \cdot PDeff$

If $val > 0$ then, $Isbd = val$

Otherwise, $Isbd = M \cdot IS$

$$Js = Js0 \cdot \exp\left[\frac{\frac{Eg0}{Vt0} - \frac{Eg}{Vt} + Xti \cdot \ln\left(\frac{T}{Tnom}\right)}{Nj}\right]$$

$$Jsw = Jsw0 \cdot \exp\left[\frac{\frac{Eg0}{Vt0} - \frac{Eg}{Vt} + Xti \cdot \ln\left(\frac{T}{Tnom}\right)}{Nj}\right]$$

For calculating the same parameters in SymSpice netlist consult the University of California at Berkeley BSIM3 home page at
http://www-device.eecs.berkeley.edu/~bsim3/index.html

MOS diode equations

DC current

If parameter $Ijth \le 0$

$$Ibs = Isbs \cdot \left[\exp\left(\frac{Vbs}{Nj \cdot vt}\right) - 1\right]$$

$$Ibd = Isbd \cdot \left[\exp\left(\frac{Vbd}{Nj \cdot vt}\right) - 1\right]$$

Otherwise ($Ijth > 0$):

For source

$$Vjsm = Njvt \cdot \ln\left(\frac{Ijth}{Isbs} + 1\right)$$

For drain

$$Vjsm = Njvt \cdot \ln\left(\frac{Ijth}{Isbd} + 1\right)$$

Drain and source diodes forward biased $Vbs < Vjsm$:

$$Ibs = Isbs \cdot \left[\exp\left(\frac{Vbs}{Nj \cdot vt} \right) - 1 \right]$$

$$Ibd = Isbd \cdot \left[\exp\left(\frac{Vbd}{Nj \cdot vt} \right) - 1 \right]$$

Drain and source diodes reverse biased $Vbs \geq Vjsm$:

$$Ibs = Ijth + \frac{Ijth + Isbs}{Nj \cdot vt} \cdot (Vbs - Vjsm)$$

$$Ibd = Ijth + \frac{Ijth + Isbd}{Nj \cdot vt} \cdot (Vbd - Vjsm)$$

MOS diode capacitance equations

Source diode capacitance

$$Capbs = Cjbst + Cjbsswgt + Cjbsswt$$

$$Cjbst = As \cdot Cjbs$$

if Ps>Weff'

$$Cjbsswgt = Weff' \cdot Cjbsswg$$

$$Cjbsswt = (Ps - Weff') \cdot Cjbssw$$

otherwise (PsJ Weff')

$$Cjbswgt = Ps \cdot Cjbsswg$$

$$Capbs = As \cdot Cjbs + Ps \cdot Cjbsswg$$

$$Cjbs = Cj \cdot \left(1 - \frac{Vbs}{Pb} \right)^{-Mj}, \quad Vbs < 0$$

$$Cjbs = Cj \cdot \left(1 + Mj \cdot \frac{Vbs}{Pb} \right), \quad Vbs \geq 0$$

$$Cjbssw = Cjsw \cdot \left(1 - \frac{Vbs}{Pbsw} \right)^{-Mjsw}, \quad Vbs < 0$$

$$Cjbssw = Cjsw \cdot \left(1 + Mjsw \cdot \frac{Vbs}{Pbsw} \right), \quad Vbs \geq 0$$

$$Cjbsswg = Cjswg \cdot \left(1 + \frac{Vbs}{Pbswg} \right)^{-Mjswfg}, \quad Vbs < 0$$

$$Cjbsswg = Cjswg \cdot \left(1 + Mjswg \cdot \frac{Vbs}{Pbswg} \right), \quad Vbs \geq 0$$

Drain diode capacitance

$$Capbd = Cjbdt + Cjbdswgt + Cjbdswt$$

$$Cjbdt = Ad \cdot Cjbd$$

if Pd>Weff'

$$Cjbdswgt = Weff' \cdot Cjbdswg$$
$$Cjbdswt = (Pd - Weff') \cdot Cjbdsw$$

otherwise (Pd≤Weff')

$$Cjbdwgt = Pd \cdot Cjbdswg$$
$$Capbd = Ad \cdot Cjbd + Pd \cdot Cjbdswg$$
$$Cjbd = Cj \cdot \left(1 - \frac{Vbd}{Pb}\right)^{-Mj}, \quad Vbd < 0$$
$$Cjbd = Cj \cdot \left(1 + Mj \cdot \frac{Vbd}{Pb}\right), \quad Vbd \geq 0$$
$$Cjbdsw = Cjsw \cdot \left(1 - \frac{Vbd}{Pbsw}\right)^{-Mjsw}, \quad Vbd < 0$$
$$Cjbdsw = Cjsw \cdot \left(1 + Mjsw \cdot \frac{Vbd}{Pbsw}\right), \quad Vbd \geq 0$$
$$Cjbdswg = Cjswg \cdot \left(1 + \frac{Vbd}{Pbswg}\right)^{-Mjswfg}, \quad Vbd < 0$$
$$Cjbdswg = Cjswg \cdot \left(1 + Mjswg \cdot \frac{Vbd}{Pbswg}\right), \quad Vbd \geq 0$$

Effective channel length and width

$$Leff = Ldrawn - 2dL$$
$$Weff = Wdrawn - 2dW$$
$$Weff' = Wdrawn - 2dW'$$
$$dW = dW' + dWgVgsteff + dWb \cdot (\sqrt{\Phi s - Vbseff} - \sqrt{\Phi s})$$
$$dW' = Wint + \frac{Wl}{L^{Wln}} + \frac{Ww}{W^{Wwn}} + \frac{Wwl}{L^{Wln} W^{Wwn}}$$
$$dL = Lint + \frac{Ll}{L^{Lln}} + \frac{Lw}{W^{Lwn}} + \frac{Lwl}{L^{Lln} W^{Lwn}}$$

Note: A detailed discussion of the BSIM3v3 equations is available from the BSIM3 site:
http://www.device.eecs.berkeley.edu/~bsim3/get.html

Drain/Source resistance

Calculating Effective Drain and Source Resistances

For ACM=0, the effective drain and source resistances are calculated as:

Source Resistance

Define:

$$val = Nrs \cdot Rsh$$

If val > 0 then,

$$Rseff = \frac{val + Rsc}{M}$$

Otherwise,

$$Rseff = \frac{Rs + Rsc}{M}$$

Drain Resistance

Define:

$$val = Nrd \cdot Rsh$$

If val > then,

$$RDeff = \frac{val + Rdc}{M}$$

Otherwise,

$$RDeff = \frac{Rd + Rdc}{M}$$

Temperature effects

$$Vth(T) = Vth(Tnom) + (Kt1 + Kt1l/Leff + Kt2 \cdot Vbseff) \cdot (T/Tnom - 1)$$

$$\mu 0(T) = \mu 0(Tnom) \cdot \left(\frac{T}{Tnom}\right)^{U}$$

$$Vsat(T) = Vsat(Tnom) - At \cdot (T/Tnom - 1)$$

$$Rdsw(T) = Rdsw(Tnom) + Prt \cdot \left(\frac{T}{Tnom} - 1\right)$$

$$Ua(T) = Ua(Tnom) + Ua1 \cdot (T/Tnom - 1)$$

$$Ub(T) = Ub(Tnom) + Ub1 \cdot (T/Tnom - 1)$$

$$Uc(T) = Uc(Tnom) + Uc1 \cdot (T/Tnom - 1)$$

$$Eg0 = 1.16 - \frac{7.02 \cdot 10^{-4} Tnom^{2}}{Tnom + 1108}$$

$$Eg = 1.16 - \frac{7.02 \cdot 10^{-4} T^{2}}{T + 1108}$$

$$Cj(T) = Cj \cdot [1 + Tcj \cdot (T - Tnom)]$$

$$Cjsw(T) = Cjsw \cdot [1 + Tcjsw \cdot (T - Tnom)]$$

$$Cjswg(T) = Cjswg \cdot [1 + Tcjswg \cdot (T - Tnom)]$$

$$Pb(T) = Pb - Tpb \cdot (T - Tnom)$$

$$Pbsw(T) = Pbsw - Tpbsw \cdot (T - Tnom)$$

$$Pbswg(T) = Pbswg - Tpbswg \cdot (T - Tnom)$$

Level 54 BSIM 4 Model

BSIM4, as the extension of BSIM3 model, addresses the MOSFET physical effects into sub-100nm regime. The continuous scaling of minimum feature size brought challenges to compact modeling in two ways: One is that to push the barriers in making transistors with shorter gate length, advanced process technologies are used such as non-uniform substrate doping. The second is its opportunities to RF applications.

To meet these challenges, BSIM4 has the following major improvements and additions over BSIM3v3:

- An accurate new model of the intrinsic input resistance for both RF, high-frequency analog and high-speed digital applications.
- Flexible substrate resistance network for RF modeling.
- A new accurate channel thermal noise model and a noise partition model for the induced gate noise.
- A non-quasi-static (NQS) model that is consistent with the Rg-based RF model and a consistent AC model that accounts for the NQS effect in both transconductances and capacitances.
- An accurate gate direct tunneling model for multiple layer gate dielectrics.
- A comprehensive and versatile geometry dependent parasitics model for various source/drain connections and multi-finger devices.
- Improved model for steep vertical retrograde doping profiles.
- Better model for pocket-implanted devices in Vth, bulk charge effect model, and Rout.
- Asymmetrical and bias-dependent source/drain resistance, either internal or external to the intrinsic MOSFET at the user's discretion.
- Acceptance of either the electrical or physical gate oxide thickness as the model input at the user's choice in a physically accurate manner.
- The quantum mechanical charge-layer thickness model for both IV and CV.
- A more accurate mobility model for predictive modeling.
- A gate-induced drain/source leakage (GIDL/GISL) current model, available in BSIM for the first time.
- An improved unified flicker (1/f) noise model, which is smooth over all bias regions and considers the bulk charge effect.
- Different diode IV and CV characteristics for source and drain junctions.
- Junction diode breakdown with or without current limiting.
- Dielectric constant of the gate dielectric as a model parameter.

- A new scalable stress effect model for process induced stress effect; device performance becoming thus a function of the active area Geometry and the location of the device in the active area.
- A unified current-saturation model that includes all mechanisms of current saturation- velocity saturation, velocity overshoot and source end velocity limit.
- A new temperature model format that allows convenient prediction of temperature effects on saturation velocity, mobility, and S/D resistances.

In 2004 Berkeley released BSIM4 Version 4.0, which contains many new features (see http://www-device.eecs.berkeley.edu/~bsim3/bsim4.html).

Syntax and Individual Parameters

Syntax in 'local' or 'spectre' modes

without model:

Name <(>nd ng ns nb<)> bsim4 parameter=value ...

with model:

Name <(>nd ng ns nb<)> modelname parameter=value ...

model modelname bsim4 pname=val

Syntax in 'spice' mode

Mxxx nd ng ns <nb> mname

+<<L=>length> <<W=>width>

+<AD=val> <AS=val> <PD=val>

+<PD=val> <M=val> <DTEMP=val>

+<RBPB=val> <RBPD=val> <RBPS=val>

+<RBDB=val> <RBSB=val> <NRD=val>

+<NRS=val> <MIN=val> <NF=val>

+<GEOMOD=val> <RGEOMOD=val>

+<RBODYMOD=val> <TRNQSMOD=val>

+<ACNQSMOD=val> <SA=val> +<SB=val>

+<SD=val> <DELVTO=val> <SCA=val>

+<SCB=val> <SCD=val> <SC=val>

where:

SPICE mode	SymSpice mode	Default	Desciption
Mxxx	Name	-	BSIM4 element name. For SPICE format netlist it must begin with "M", which can be followed by alphanumeric characters.
mname	bsim4/modelname	-	BSIM4 model name reference.
L	l	-	MOSFET channel length
W	w	-	MOSFET channel width
AD	ad	-	Drain diffusion area
AS	as	-	Source diffusion area
PD	pd	-	Perimiter of the drain junction, including the channel edge
PS	ps	-	Perimiter of the source junction, including the channel edge
M	m	1	Multiplier to simulate multiple MOSFETs in parallel
DTEMP	trise	0.0	The difference between the element temperature and the circuit temperature
RBPB	rbpb	-	Resistance connected between bNodePrime and bNode
RBPD	rbpd	-	Resistance connected between bNodePrime and dbNode
RBPS	rbps	-	Resistance connected between bNodePrime and sbNode
RBDB	rbdb	-	Resistance connected between dbNode and bNode
RBSB	rbsb	-	Resistance connected between sbNode and bNode
NRD	nrd	-	Number of squares of drain diffusion for resistance calcualtions
NRS	nrs	-	Number of squares of source diffusion for resistance calcualtions
MIN	min	0	Whether to minimize the number of drain or source diffusions for even-number fingered device
NF	nf	1	Number of device fingers
GEOMOD	geomod	0	Geometry dependent parasitics model selector - specifying how the end S/D diffusions are connected

RGEOMOD	rgeomod	-	Source/drain diffusion resistance and contact model selector - specifying the end S/D contact type:point,wide or merged, and how S/D parasitics resistance is computed
RBODYMOD	rbodymod	0	Substrate resistance network model selector
TMQSMOD	trnqsmod	0	Transient NQS model selector
ACNQSMOD	acnqsmod	0	AC small-signal NQS model selector
SA	sa	0	Distance between OD edge to Poly from one side
SB	sb	0	Distance between OD edge to Poly from the other side
SD	sd	0	Distance between neighbor fingers
DELVTO	delvto	0	Adding to Vth0
SCA	-	-	Integral of the first distribution function for scattered well dopant
SCB	-	-	Integral of the second distribution function for scattered well dopant
SCC	-	-	Integral of the third distribution function for scattered well dopant
SC	-	-	Distance to a single well edge

Level 54 BSIM 4 Model(Group) Parameters

Model Selectors

Parameter name	Default	Description
version	4.2	BSIM 4 version selector. BSIM 4.2, 4.3, 4.4, 4.5 supported
binunit	1	Binning unit selector
paramchk	1	Switch for parameter value chek
mobmod	0	Mobility model selector
rdsmod	0	Bias depending source/drain resistance model selector
igcmod	0	Gate-to-channel tunneling current model selector
igbmod	0	Gate-to-substrate tunneling current model selector
capmod	2	Capacitance model selector
rgatemod	0	Gate resistance model selector

rbodymod	0	Substrate resistance network model selector
trnqsmod	0	Transient nqs model selector
acnqsmod	0	AC small-signal nqs model selector
fnoimod	1	Flicker noise model selector
tnoimod	0	Thermal noise model selector
diomod	1	Source/drain junction diod I-V model selector
permod	1	Whether ps/pd (when given) includes the gate-edge perimeter
geomod	0	Geometry dependent parasitics model selector - specifying how the end s/d diffusions are connected
tempmod	0.0	Temperature mode selector

Process parameters

Parameter name	Units	Default	Bin	Description
epsrox		3.9		Gate dielectric constant relative to vacuum
toxe	m	3.0E-9		Electrical gate equivalent oxide thickness
toxp	m	toxe		Physical gate equivalent oxide thickness
toxm	m	toxe		tox at which parameters are extracted
dtox	m	0.0		Difference toxe and toxp
xj	m	1.5e-7	+	s/d junction depth
gamma1	$V^{1/2}$	Calculated	+	Body-effect coefficient near the interface
gamma2	$V^{1/2}$	Calculated	+	Body-effect coefficient in the bulk
ndep	$1/cm^3$		+	Channel doping concentration at depletion edge for zero body bias
nsub	$1/cm^3$	6E16	+	Substrate doping concentration
ngate	cm^{-3}	0	+	Poly Si gate doping concentration
nsd	$1/cm^3$	1.0e20	+	Source/drain doping concentration
vbx	V	Calculated		vbs at which the depletion width equals xt
xt	m	1.55E-7	+	Doping depth
rsh	Ohm/sqr	0		Source/drain sheet resistance

rshg	Ohm/sqr	0.1		Gate electrode sheet resistance

Parameters for effective channel length/width in I-V model

Parameter name	Units	Default	Description
wl	m^{Wln}	0.0	Coefficient of length dependence for width offset
wln		1.0	Power of length dependence of width offset
ww	m^{Wwn}	0.0	Coefficient of width dependence for width offset
wwn		1.0	Power of width dependence of width offset
wwl	$m^{(Wwn+Wln)}$	0.0	Coefficient of length and width cross term for width offset
ll	m^{Lln}	0.0	Coefficient of length dependence for length offset
lln		1.0	Power of length dependence for length offset
lw	m^{Lwn}	0.0	Coefficient of width dependence for length offset
lwn		1.0	Power of width dependence for length offset
lwl	$m^{(Lwn+Lln)}$	0.0	Coefficient of length and width cross term for length offset

Parameters for effective channel length/width in C-V model

Parameter name	Units	Default	Description
llc	m^{Lln}	ll	Coefficient of length dependence for C-V channel length offset
lwc	m^{Lwn}	lw	Coefficient of width depen-dence for C-V channel length offset
lwlc	$m^{(Lwn+Lln)}$	lwl	Coefficient of length and width-dependence for C-V channel length offset
wlc	m^{Wln}	wl	Coefficient of length dependence for C-V channel width offset

wwc	m^{Wwn}	ww	Coefficient of widthdependence for C-V channel width offset
wwlc	$m^{(Wwn+Wln)}$	wwl	Coefficient of length and width dependence for C-V channel width offset

Unified Current Saturation

Parameter name	Units	Default	Bin	Description
lambda		0.0	+	Velocity overshoot coefficient
vtl	m/s	2.0e5	+	Thermal velocity
lc	m	0.0	+	Velocity back scattering coefficient
xn		3.0		Velocity back scattering coefficient

Range parameters for model applications

Parameter name	Units	Default	Description
lmin	m	0.0	Minimum channel length for which the model is valid
lmax	m	1.0	Maximum channel length for which the model is valid
wmin	m	0.0	Minimum channel width for which the model is valid
wmax	m	1.0	Maximum channel width for which the model is valid

Gate-Induced Drain Leakage Model Parameters

Parameter name	Units	Default	Bin	Description
agidl	1/Ohm	0	+	Pre-exponential coefficient for gidl
bgidl	V/m	2.3e9	+	Exponential coefficient for gidl
cgidl	V^3	0.5	+	Parameter for body-bias effect on gidl
egidl	V	0.8	+	Fitting parameter for band-bending for gidl

Basic model parameters

Parameter name	Units	Default	Bin	Description
vth0	V		+	Threshold voltage for large l at vbs=0
vfb	V		+	Flat band voltage
phin	V	0.0	+	Non uniform vertical doping effect on surface potential
k1	$V^{1/2}$		+	First-order body effect coefficient
k2			+	Second-order body effect coefficient
k3		80.0	+	Narrow width coefficient
k3b	1/V	0	+	Body effect coefficient of k3
w0	m	2.5E-6	+	Narrow width parameter
lpe0	m	1.74E-7	+	Lateral non-uniform doping coefficient
lpeb	m	0.0	+	Lateral non-uniform doping effect on k1
vbm	V	-3.0	+	Maximum applied body bias in vth calculation
dvt0		2.2	+	First coefficient of short-channel effect on vth
dvt1		0.53	+	Second coefficient of short-channel effect on vth
dvt2	1/V	-0.032	+	Body-bias coefficient of short-channel effect on vth
dvtp0	m	0.0	+	First coefficient of drain-induced vth shift due to for long-channel pocket devices
ldvtp1	m/V	0	+	Length dependence of the Second coefficient of short-channel effect on vth
dvt0w		0	+	First coefficient of narow width effect on vth at small l
dvt1w	1/m	5.3E6	+	Second coefficient of narow width effect on vth at small l
dvt2w	1/V	-0.032	+	Body-bias coefficient of narow width effect on vth at small l
u0	$m^2/(V \cdot s)$		+	Low field mobility (0.025 for PMOS)

ua	m/V		+	First-order mobility degradation coefficient due to vertical field (1e-15 for mobmod=2)
ub	$(m/V)^2$		+	Second-order mobility degradation coefficient
uc	$(m/V)^2$ or 1/V		+	Coefficient of mobility degradation due to Body-bias effect (-0.0465 V**-1 for mobmod=1 or -0.0465e-9m/V**2 foe others)
eu				Exponent for mobility degradation of mobmod=2 (1.67 for NMOS and 1.0 for PMOS)
vsat	m/sec	8E4	+	Saturation velocity
a0		1.0	+	Coefficient of channel length dependence of bulk charge effect
ags	1/V	0.0	+	Gate bias coefficient of the bulk charge effect
b0	m	0.0	+	Bulk charge effect coefficient for channel width
b1	m	0.0	+	Bulk charge effect width offset
keta	1/V	-0.047	+	Body-bias coefficient of bulk charge effect
a1	1/V	0.0	+	First non-saturation effect parameter
a2		1.0	+	Second non-saturation effect parameter
wint	m	0.0	+	Channel-width offset fitting parameter from I-V without bias effect
lint	m	0.0	+	Channel-length offset fitting parameter from I-V without bias effect
dwg	m/V	0.0	+	Coefficient of weff's gate dependence
dwb	$m/V^{1/2}$	0.0	+	Coefficient of weff's substrate body bias dependence
voff	V	-0.08	+	Offset voltage in the subthreshold region at large W and l
voffl	m*V	0.0		Channel-length dependence of voff
minv		0.0	+	vgsteff fitting parameter for moderate inversion condition
nfactor		1.0	+	Subthreshold swing factor
eta0		0.08	+	dibl coefficient in subthreshold region

etab	1/V	-0.07	+	Body-bias coefficient for the subthreshold dibl effect
dsub		Drout	+	dibl coefficient exponent in subthreshold region
cit	F/m^2	0.0	+	Interface trap capacitance
cdsc	F/m^2	2.4E-4	+	Drain/Source to channel coupling capacitance
cdscb	$F/(V \cdot m^2)$	0.0	+	Body-bias sensitivity of cdsc
cdscd	$F/(V \cdot m^2)$	0.0	+	Drain-bias sensitivity of cdsc
pclm		1.3	+	Channel length modulation parameter
pdiblc1		0.39	+	First output resistance dibl effect correction parameter
pdiblc2		0.0086	+	Second output resistance dibl effect correction parameter
pdiblcb	1/V	0	+	Body effect coefficient of dibl correction parameters
drout		0.56	+	l dependence coefficient of the dibl correction parameter in rout
pscbe1	V/m	4.24E8	+	First substrate current body-effect parameter
pscbe2	m/V	1E-5	+	Second substrate current body-effect parameter
pvag		0.0	+	Gate dependence of early voltage
delta	V	0.01	+	Effective vds parameter
fprout	$V/m^{0.5}$	0.0	+	Effect of pocket implant in rout degradation
pdits	1/V	0.0	+	Impact of drain induced vth shift on Rout
pditsl	1/m	0.0		Channel-length dependence of drain induced vth shift for Rout
pditsd	1/V	0.0	+	vds dependence of drain induced vth shift for rout

Parameters for asymmetric and bias-dependent Rds model

Parameter name	Units	Default	Bin	Description

rdsw	Ohm*micron	200.0		Zero-bias ldd resistance per unit width for rdsmode=0
rdswmin	Ohm*micron	0.0		Zero-bias ldd resistance per unit width at high vgs for rdsmode=0
rdw	Ohm*micron	100.0	+	Zero-bias lightly-doped drain resistance per unit width for rdsmode=1
rdwmin	Ohm*micron	0.0		Zero-bias lightly-doped drain resistance per unit width for rdsmode=1
rsw	Ohm*micron	100.0	+	Zero-bias lightly-doped source resistance per unit width for rdsmode=1
êswmin	Ohm*micron	0.0		Zero-bias lightly-doped source resistance per unit width for rdsmode=1
prwg	1/V	1.0	+	Gate bias effect coefficient of ldd resistance
prwb	$1/V^{1/2}$	0.0	+	Body effect coefficient of ldd resistance
wr		1.0	+	Channel-width dependence parameter of ldd resistance

Parameters for substrate current model

Parameter name	Units	Default	Bin	Description
alpha0	A*m/V	0	+	The first parameter of impact ionization current
alpha1	A/V	0	+	The length scaling parameter of substrate current model
beta0	V	30	+	The second parameter of impact ionization current

Gate Dielectric Tunneling Current Model Parameters

Parameter name	Units	Default	Bin	Description
aigbacc	$(m^{-1})\cdot(F\cdot s^2/g)^{0.5}$	0.43	+	Parameter for igb in accumulation
bigbacc	$(V\cdot m^{-1})\cdot(F\cdot s^2/g)^{0.5}$	0.054	+	Parameter for igb in accumulation
cigbacc	1/V	0.075	+	Parameter for igb in accumulation
nigbacc		1.0	+	Parameter for igb in accumulation
aigbinv	$(m^{-1})\cdot(F\cdot s^2/g)^{0.5}$	0.35	+	Parameter for igb in inversion

bigbinv	$(V \cdot m^{-1}) \cdot (F \cdot s^2/g)^{0.5}$	0.03	+	Parameter for igb in inversion
cigbinv	1/V	0.006	+	Parameter for igb in inversion
eigbinv	1/V	1.1	+	Parameter for igb in inversion
nigbinv		3.0	+	Parameter for igb in inversion
aigc	$(m^{-1}) \cdot (F \cdot s^2/g)^{0.5}$		+	Parameter for igcs and igcd
bigc	$V^{-1} \cdot m^{-1} \cdot (F \cdot s^2/g)^{0.5}$		+	Parameter for igcs and igcd
cigc	V^{-1}		+	Parameter for igcs and igcd
aigsd	$(m^{-1}) \cdot (F \cdot s^2/g)^{0.5}$		+	Parameter for igs and igd
bigsd	$V^{-1} \cdot m^{-1} \cdot (F \cdot s^2/g)^{0.5}$		+	Parameter for igs and igd
cigsd	V^{-1}		+	Parameter for igs and igd
dlcig	m	Lint	+	Source/drain overlap length for igs and igd
nigc		1.0	+	Parameter for igcs, igcd, igs and igd
poxedge		1.0	+	Factor for the gate-oxide thickness in source/drain overlap regions
pigcd		1.0	+	vds dependence of igcs and igcd
ntox		1.0	+	Exponent for the gate-oxide ratio
toxref	m	3.0e-9		Nominal gate-oxide thickness for gate dielectric tunneling current model only

Charge and capacitance model parameters

Parameter name	Units	Default	Bin	Description
xpart		0		Charge partitioning flag
cgso	F/m			Non LDD region source-gate overlap capacitance per channel length calculated
cgdo	F/m			Non LDD region drain-gate overlap capacitance per channel length calculated
cgbo	F/m	0.0		Gate bulk overlap capacitance per unit channel length
cgsl	F/m	0.0	+	Light doped source-gate region overlap capacitance

cgdl	F/m	0.0	+	Light doped drain-gate region overlap capacitance
ckappas	V	0.6	+	Coefficient of bias-dependent overlap capacitance for the source side
ckappad	V	ckappas	+	Coefficient of bias-dependent overlap capacitance for the drain side
cf	F/m	Calculated	+s	Fringing field capacitance calculated
clc	m	1E-7	+	Constant term for the short channel model
cle		0.6	+	Exponential term for the short channel model
dlc	m			Length offset fitting parameter from C-V
dwc	m	wint		Width offset fitting parameter from C-V
vfbcv	V	-1.0	+	Flat-band voltage parameter (for capmod=0 only)
noff		1.0	+	C-V parameter for vgsteff
voffcv	V	0.0	+	Offset voltage parameter of vth from week to strong inversion in C-V model
acde	m/V	1.0	+	Exponential coefficient for charge thickness in capmod=3 for accumulation and depletion regions
moin	$V^{1/2}$	15.0	+	Coefficient for the gate-bias dependent surface potential

High-Speed/RF Model Parameters

Parameter name	Units	Default	Bin	Description
xrcrg1		12.0	+	Parameter for distributed channel-resistance effect for both intrinsic-input resistance and charge-deficit NQS models
xrcrg2		1.0	+	Parameter to account for the excess channel diffusion resistance for both intrinsic-input resistance and charge-deficit NQS models
rbpb	Ohm	50.0		Resistance connected between bNodePrime and bNode
rbpd	Ohm	50.0		Resistance connected between bNodePrime and dbNode

rbps	Ohm	50.0		Resistance connected between bNodePrime and sbNode
rbdb	Ohm	50.0		Resistance connected between dbNode and bNode
rbsb	Ohm	50.0		Resistance connected between dbNode and bNode
gbmin	1/Ohm	1.0e-12		Conductance in parallel with each of the five substrate resistances to avoid potential numerical instability due to unreasonably too large a substrate resistance

Flicker and Thermal Noise Model Parameters

Parameter name	Units	Default	Description
noia	$(eV^{-1})\cdot(s^{(1-Ef)})\cdot m^{-3}$	6.25e41	Flicker noise parameter A
noib	$(eV^{-1})\cdot(s^{(1-Ef)})\cdot m^{-1}$	3.125e26	Flicker noise parameter B
noic	$(eV^{-1})\cdot(s^{(1-Ef)})\cdot m$	8.75	Flicker noise parameter C
em	V/m	4.1E7	Saturation field
af		1	Flicker noise exponent
ef		1	Flicker noise frequency exponent
kf	$A^{(2-Ef)}\cdot s^{(1-Ef)}\cdot F$	0	Flicker noise coefficient
ntnoi		1.0	Noise factor for short-channel devices for tnoimod=0 only
tnoia		1.5	Coefficient of channel-length dependence of total channel thermal noise
tnoib		3.5	Parameter of channel-length dependence of channel thermal noise partitioning

Layout-Dependent Parasitics Model Parameters

Parameter name	Units	Default	Description
dmcg	m	0.0	Distance from S/D contact center to the gate edge
dmci	m	dmcg	Distance from S/D contact center to the isolation edge in the channel-length direction

dmdg	m	0.0	Same as dmcg but for merged device only
dmcgt	m	0.0	dmcg of test structures
dwj	m	dwc	Offset of S/D junction width
xgw	m	0.0	Distance from the gate contact to the channel edge
xgl	m	0.0	Offset of the gate length due to variations in pattering
xl	m		Channel length offset due to mask/etch effect
xw	m		Channel width offset due to mask/etch effect
ngcon		1	Number of gate contact

Asymmetric Source/Drain Junction Diode Model Parameters

Parameter name	Units	Default	Description
ljthsrev	A	0.1	Diode limiting current in reverse bias region of b-s junction
ljthdrev	A	ljthsrev	Diode limiting current in reverse bias region of b-d junction
ljthsfwd	A	0.1	Diode limiting current in forward bias region of b-s junction
ljthdfwd	A	ljthsfwd	Diode limiting current in forward bias region of b-d junction
xjbvs		1.0	Fitting parameter for diode breakdown of b-s junction
xjbvd		xjbvs	Fitting parameter for diode breakdown of b-d junction
bvs	V	10	Breakdown voltage of b-s junction
bvd	V	bvs	Breakdown voltage of b-d junction
jss	A/m^2	1E-4	Saturation current density of bottom b-s junction diode
jsd	A/m^2	jss	Saturation current density of bottom b-d junction diode
jsws	A/m	0	Saturation current density of Isolation edge sidewall b-s junction diode
jswd	A/m	jsws	Saturation current density of Isolation edge sidewall b-d junction diod

jswgs	A/m	0	Saturation current density of gate edge sidewall b-s junction diode
jswgd	A/m	jswgs	Saturation current density of gate edge sidewall b-d junction diode
cjs	F/m^2	5E-4	Source bottom junction capacitance per unit area at zero bias
cjd	F/m^2	cjs	Drain bottom junction capacitance per unit area at zero bias
mjs		0.5	Source Bottom junction capacitance grating coefficient
mjd		0.5	Drain Bottom junction capacitance grating coefficient
mjsws		0.33	Source isolation-edge sidewall junction capacitance grading coefficient
mjswd		mjsws	Drain isolation-edge sidewall junction capacitance grading coefficient
cjsws	F/m	5E-10	Source isolation-edge sidewall junction capacitance per unit area
cjswd	F/m	cjsws	Drain isolation-edge sidewall junction capacitance per unit area
cjswgs	F/m	cjsws	Source gate-edge sidewall junction capacitance per unit length at zero bias
cjswgd	F/m	cjsws	Drain gate-edge sidewall junction capacitance per unit length at zero bias
mjswgs		mjsws	Source gate-edge sidewall junction capacitance grading coefficient
mjswgd		mjsws	Drain gate-edge sidewall junction capacitance grading coefficient
pbs	V	1.0	Bottom source junction built-in potential
pbd	V	pbs	Bottom drain junction built-in potential
pbsws	V	1.0	Source isolation-edge sidewall junction built-in potential
pbswd	V	pbsws	Drain isolation-edge sidewall junction built-in potential
pbswgs	V	pbsws	Built-in potential of the source gate-edge sidewall junction
pbswgd	V	pbsws	Built-in potential of the drain gate-edge sidewall junction

Temperature Dependence Parameters

Parameter name	Units	Default	Bin	Description
tnom(tref)	C°	27		Temperature at which parameters are extracted
ute		-1.5	+	Mobility temperature exponent
kt1	V	-0.11	+	Temperature coefficient for threshold voltage
kt1l	V*m	0.0	+	Channel length dependence of the temperature coefficient for threshold voltage
kt2		0.022	+	Body-bias coefficient of vth temperature effect
ua1	m/V	1E-9	+	Temperature coefficient for ua
ub1	$(m/V)^2$	-1E-18	+	Temperature coefficient for ub
uc1	1E-9* m/V^2 or 1/V		+	Temperature coefficient for uc mobMod=1,2 or mobMod=3
at	m/s	3.3E4	+	Temperature coefficient for saturation velocity
prt	Ohm*m	0.0	+	Temperature coefficient for rdsw
njs		1		Emission coefficient of source junction
njd		1		Emission coefficient of drain junction
xtis		3.0		Source junction current temperature exponent coefficient
xtid		Xtis		Drain junction current temperature exponent coefficient
tpb	V/K	0.0		Temperature coefficient of pb
tpbsw	V/K	0.0		Temperature coefficient of pbsw
tpbswg	V/K	0.0		Temperature coefficient of pbswg
tcj	1/K	0.0		Temperature coefficient of cj
tcjsw	1/K	0.0		Temperature coefficient of cjsw
tcjswg	1/K	0.0		Temperature coefficient of cjswg

Stress Effect Model Parameters

Parameter name	Units	Default	Description
saref(sa0)	m	1e-6	Reference distance between OD edge to poly of one side
sbref(sb0)	m	1e-6	Reference distance between OD edge to poly of the other side
wlod	m	0.0	Width parameter for stress effect
ku0	m	0.0	Mobility degradation/enchancement coefficient for stress effect
kvsat	m	0.0	Saturation velocity degradation/enchancement parameter for stress effect
tku0		0.0	Temperature coefficient of ku0
lku0		0.0	Length dependence of ku0
wku0		0.0	Width dependence of ku0
pku0		0.0	Cross-term dependence of ku0
llodku0		0.0	Length parameter for u0 stress effect
wlodku0		0.0	Width parameter for u0 stress effect
kvth0	V*m	0.0	Threshold shift parameter for stress effect
lkvth0		0.0	Length dependence of Kvth0
wkvth0		0.0	Width dependence of Kvth0
pkvth0		0.0	Cross-term dependence of Kvth0
llodvth		0.0	Length parameter for Vth stress effect
wodvth		0.0	Width parameter for Vth stress effect
stk2	m	0.0	K2 shift factor related to Vth0 change
lodk2	m	0.0	K2 shift modification factor for stress effect
steta0	m	0.0	Eta0 shift factor related to Vth0 change
lodeta0	m	0.0	Eta0 shift modification factor for stress effect

Enhancment of version BSIM 4.4.0

Parameter name	Units	Default	Bin	Description
jtss	A/m^2	0.0		Bottom trap-assisted saturation current density of b-s junction diode

jtsd	A/m^2	0.0		Bottom trap-assisted saturation current density of b-d junction diode
jtssws	A/m	0.0		Bottom trap-assisted saturation current density of Isolation edge sidewall b-s junction diode
jtsswd	A/m	0.0		Bottom trap-assisted saturation current density of Isolation edge sidewall b-d junction diode
jtsswgs	A/m	0.0		Trap-assisted saturation current density of gate edge sidewall b-s junction diode
jtsswgd	A/m	0.0		Trap-assisted saturation current density of gate edge sidewall b-d junction diode
nts		20.0		Non-ideality factor for jtss, jtsd
njtssw		20.0		Non-ideality factor for jtssws, jtsswd
njtsswg		20.0		Non-ideality factor for jtswgs, jtswgd
xtss		0.02		Power dependence of jtss on temperature
xtsd		0.02		Power dependence of jtsd on temperature
xtssws		0.02		Power dependence of jtssws on temperature
xtsswd		0.02		Power dependence of jtsswd on temperature
xtsswgs		0.02		Power dependence of jtsswgs on temperature
xtsswgd		0.02		Power dependence of jtsswgd on temperature
vtss	V	10.0		Bottom trap-assisted voltage dependent parameter
vtsd	V	vtss		Bottom trap-assisted voltage dependent parameter
vtssws	V	10.0		Sti sidewall trap-assisted voltage dependent parameter
vtsswd	V	vtssws		Sti sidewall trap-assisted voltage dependent parameter
vtsswgs	V	10.0		Gate-edge sidewall trap-assisted voltage dependent parameter
vtsswgd	V	vtsswgs		Gate-edge sidewall trap-assisted voltage dependent parameter
tnjts		0.0		Temperature coefficient for njts
tnjtssw		0.0		Temperature coefficient for njtssw
tnjtsswg		0.0		Temperature coefficient for njtsswg
vfbsdoff	V	0.0	+	Flatband voltage offset parameter

lintnoi		0.0		Length reduction parameter offset
xlref	m			Difference between physical (on wafer) and drawn reference channel length
xwref	m			Difference between physical (on wafer) and drawn reference channel width
tempmod		0.0		Temperature mode selector
rnoia		0.577		Thermal noise coefficient
rnoib		0.37		Thermal noise coefficient

BSIM 4.5.0. Enhancements

- A mobility model which accounts for Coulomb scattering effect as well as the channel length dependence of mobility due to heavy halo-doping.
- A scalable substrate resistance model (rbodyMod=2) that is scalable with channel length, channel width and number of fingers.
- Gate resistance parameters Xgw, Ngcon that can now be specified as instance parameters (Xgl still model parameter).
- Additional temperature dependence of model parameters Voff, Vfbsdoff.
- Enhanced tempMod=2, where Vth(Dits) and gate tunnelig models are functions of nominal temperature dependence and the temperature dependence of zero-bias flat-band voltage is added.
- A new instance parameter Delvto that may be used to represent threshold voltage variation.
- A new well-proximity effect model developed by CMC companies.
- Igc Vbs dependence improvement with the full BSIM4 Vth model implemented.

Model Parameters

Parameter name	Units	Default	Bin	Description
ud	$1/m^2$	1E14	+	Mobility Coulomb scattering coefficient
up	$1/m^2$	0	+	Mobility channel length coefficient

lp	m	1e-8	+	Mobility channel length exponentional coefficient
rbps0	Ohm	50		Scaling prefactor for rbps
rbpsl		0		Length scaling parameter for rbps
rbpsw		0		Width scaling parameter for rbps
rbpsnf		0		Number of fingers scaling parameter for rbps
rbpd0	Ohm	50		Scaling prefactor for rbpd
rbpdl		0		Length scaling parameter for rbpd
rbpdw		0		Width scaling parameter for rbpd
rbpdnf		0		Number of fingers scaling parameter for rbpd
rbpbx0	Ohm	100		Scaling prefactor for rbpbx
rbpbxl		0		Length scaling parameter for rbpbx
rbpbxw		0		Width scaling parameter for rbpbx
rbpbxnf		0		Number of fingers scaling parameter for rbpbx
rbpby0	Ohm	100		Scaling prefactor for rbpby
rbpbyl		0		Length scaling parameter for rbpby
rbpbyw		0		Width scaling parameter for rbpby
rbpbynf		0		Number of fingers scaling parameter for rbpby
rbsbx0	Ohm	100		Scaling prefactor for rbsbx0
rbsby0	Ohm	100		Scaling prefactor for rbsby0
rbdbx0	Ohm	100		Scaling prefactor for rbdbx0
rbdby0	Ohm	100		Scaling prefactor for rbdby0
rbsdbxl		0		Length scaling parameter for rbsbx and rbdbx
rbsdbxw		0		Width scaling parameter for rbsbx and rbdbx
rbsdbxnf		0		Number of fingers scaling parameter for rbsbx and rbdb
rbsdbyl		0		Length scaling parameter for rbsby and rbdby
rbsdbyw		0		Width scaling parameter for rbsby and rbdby
rbsdbynf		0		Number of fingers scaling parameter for rbsby and rbdby

tvoff		0		Temperature coefficient of voff
tvfbsdoff	1/K	0		Temperature coefficient of vfbsdoff
web		0		Coefficient for scb must be > 0
wec		0		Coefficient for sscc must be > 0
kvth0we		0	+	Threshold shift factor for well proximity effect
k2we		0	+	K2 shift factor for well proximity effect
ku0we		0		Mobility degradation factor for well proximity effect
lkvth0we		0		Length dependence of the Mobility degradation factor for well proximity effect
wkvth0we		0		Width dependence of the Mobility degradation factor for well proximity effect
pkvth0we		0		Cross term dependence of the Mobility degradation factor for well proximity effect
scref	m	1e-6		Reference distance to calculate sca, scb, scc
wpemod		0		Flag to activate well proximity effect
rgeomod		0		Flag to activate well proximity effect

SymSpice specific parameters

Parameter name	Units	Default	Description
trise	C°	0	Temperature rise from ambient
minr	Ohm	0.001	Minimum source/drain resistance
jmelt	A/m^2		Explosion current density
meto	m	0	Metal overlap in fringing field
fc		0.5	Forward bias depletion capacitance threshold
fcsw		0.5	Sidewall forward bias depletion capacitance threshold
tlev		0	DC temperature selector
tlevc		0	AC temperature selector
eg	V	1.12452	Energy band gap
gap1	V/C°	7.02e-4	Band gap temperature coefficient
gap2	V	1108	Band gap temperature offset
pta	V/K	0	Junction potential temperature coefficient
ptp	V/K	0	Sidewall junction potential temperature coefficient

cta	1/K	0	Junction capacitance temperature coefficient
ctp	1/K	0	Sidewall junction capacitance temperature coefficient
imax	A	1	Maximum allowable current
jmax	A/m^2	1e8	Maximum allowable current density
bvj	V		Junction reverse breakdown voltage
is	A		Saturation current
js	A/m^2		Saturation current density
cj	F/m2	5e-4	Zero-bias junction bottom capacitance density
cjsw	F/m	5e-10	Zero-bias junction sidewall capacitance density
mj		0.5	Junction bottom grading coefficient
mjsw		0.33	Junction sidewall grading coefficient
pb	V	1.0	Junction bottom built-in potential
pbsw	V	1.0	Junction sidewall built-in potential
cjswg	F/m		Source/drain gate edge sidewall junction capacitance per unit length at zero bias
mjswg			Source/drain gate sidewall junction capacitance grading coefficient
pbswg			Built-in potential of the source/drain gate edge sidewall junction
n		1.0	Emission coefficient
xti		3.0	Saturation current temperature exponent
imelt	A		Explosion current

Mobility Model for BSIM4.5.0

Mobility Coulomb Scattering Model and dependence

mobMod=0

$$\mu eff = \frac{U0 \cdot f(Leff)}{1 + (Ua + Ucvbseff) \cdot \left(\frac{Vgsteff + 2Vth}{Toxe}\right) + Ub \cdot \left(\frac{Vgsteff + 2Vth}{Toxe}\right)^2 + Ud \cdot \left(\frac{Vth \cdot Toxe}{Vgsteff + 2Vth}\right)^2}$$

mobMod=1

$$\mu eff = \frac{U0 \cdot f(Leff)}{1 + \left[Ua \cdot \left(\frac{Vgsteff + 2Vth}{Toxe} \right) + Ub \cdot \left(\frac{Vgsteff + 2Vth}{Toxe} \right)^2 \right] \cdot (1 + Uc \cdot Vbseff) + Ud \cdot \left(\frac{Vth \cdot Toxe}{Vgsteff + 2Vth} \right)^2}$$

mobMod=2

$$\mu eff = \frac{U0 \cdot f(Leff)}{1 + (Ua + Uc \cdot Vbseff) \cdot \left[\frac{Vgsteff + C0 \cdot (Vth0 - Vfb - \Phi s)}{Toxe} \right]^{Eu} + Ud \cdot \left(\frac{Vth \cdot Toxe}{Vgsteff + 2Vth} \right)^2}$$

where:

$$f(Leff) = 1 - Up \cdot e^{-Leff/Lp}$$

The model is backward compatible with Up=0.0, Ud=0.0

Scalable Substrate Resistance Model

$R_X = R_{X\_HORI} \| R_{X\_VERT}$ where $R_{X\_H(V)} = R_0 L^{\alpha} W^{\beta} NF^{\gamma}$

Temperature Dependence for VOFF, VFBSDOFF

$$Voff(T) = Voff(Tnom) \cdot [1 + Tvoff \cdot (T - Tnom)]$$
$$Vfbsdoff(T) = Vfbsdoff(T\ nom) \cdot [1 + Tvfbsdoff \cdot (T - Tnom)]$$

New Temperature Mode(TempMod=2)

Share the same temperature equations as for TempMod=1

AND:

From:

$$\Delta Vth(Dits) = -nv_t \cdot \ln\left(\frac{Leff}{Leff + Dvtp0 \cdot (1 + e^{-Dvtp1Vds})} \right)$$

To:

$$\Delta Vth(Dits) = -nv_{tnom} \cdot \ln\left(\frac{Leff}{Leff + Dvtp0 \cdot (1 + e^{-Dvtp1Vds})} \right)$$

Vfbzb(T)

$$Vfbzb(T) = Vfbzb(Tnom) - Kt1 \cdot \left(\frac{T}{Tnom} - 1 \right)$$

Igate(T)

T is replaced by Tnom

DELVTO: an Instance Parameter

if Vth0 is given:

Vth0 = Vth0 + Delvto;

if Vth0 is not given,

Vfb = Vfb + Delvto;
Vth0 = Vfb + phi + k1*sqrt(k1)

Well-Proximity Effect Modeling

Instance parameters: Sca, Scb, Scc, Sc
Model parameters: Web, Wec, Kvth0we, K2we, Ku0we, Scref, Wpemod

Model equations:

Vth0 = Vth0org + Kvth0we*(Sca+Web*Scb+Wec*Scc)
K2 = K2org + K2we*(Sca+Web*Scb+Wec*Scc)

$$\mu eff = \mu eff,\ org \cdot (1 + Kuowe \cdot (Sca + Web \cdot Scb + Wec \cdot Scc))$$

- Deep well oping alters Vth of devices near the mask edge: Vth(Sc)
- Cmc well-proximity effect model is able to capture the distance effect

Gate Current Vbs Dependence

$$Igc0 = Weff \cdot Leff \cdot A \cdot ToxRatio \cdot Vgse \cdot Vaux \cdot \exp\left[-B \cdot Toxe(Aigc - Bigc \cdot Voxdepinv) \right] \cdot (1 + Cigc \cdot Voxdepinv)$$

Igcmod=1

$$Vaux - Nigc \cdot v_t \cdot \log\left(1 + \exp\left(\frac{Vgse - Vth0}{Nigc \cdot v_t} \right) \right)$$

Igcmod=2

$$Vaux - Nigc \cdot v_t \cdot \log\left(1 + \exp\left(\frac{Vgse - Vth}{Nigc \cdot v_t}\right)\right)$$

Implementing full BSIM4 Vth model into Igc enables the accurate prediction of Igs Vbs dependence.

Sources

Independent Source Elements

Independent sources determine the shape of the input characteristic, supplied to the input of circuit.

SymSpice provides the following types of independent source:

- Dc Source Element (DC-parameters)
- Pulse Source Function (PULSE-parameters)
- Sinusoidal Source Function (SIN-parameters)
- Exponential Source Function (EXP-parameters)
- Piecewise linear Source Function (PWL-parameters)
- Single-frequency FM Source Function (SFFM-parameters)

Note: do not use these reserved keywords in netlists to identify parameters as values in independent sources.

Note: 1. Voltage sources need not be grounded.
2. Positive current is assumed to flow from the positive node through the source to the negative node.
3. A positive current source forces current to flow out of the N+ node through the source and into the N- node.

Dc Source Element

Dc independent source is a source of direct current or voltage.

Syntax in 'local' or 'spectre' modes

voltage source:

Name <(>n+ n-<)> vsource parameter=value ...

current source:

Name <(>n+ n-<)> isource parameter=value ...

Syntax in 'spice' mode

voltage source:

Vxxx n+ n- <<DC=> dcval>

current source:

Iyyy n+ n- <<DC=> dcval>

The arguments are defined as follows:

SPICE mode	SymSpice mode	Description
Vxxx	Name	Independent voltage source element name. In 'spice' mode, it must begin with a "V", which can be followed by alphanumeric characters.
Iyyy	Name	Independent current source element name. For SPICE netlist it must begin with an "I", which can be followed by alphanumeric characters.
n+	n+	Positive node.
n-	n-	Negative node.
DC	type=dc	Waveform type.
dcval	dc	DC source keyword and value in volts. The "tranfun" value at time zero overrides the DC value. Default=0.0.
-	vsource/isource	Reserved keyword to designate independent voltage/current source.
-	tc1	First order temperature coefficient.
-	tc2	Second order temperature coefficient.

m	m	Multiplicity factor (only for current source).
acmag	mag	Magnitude (RMS) of the AC source.
acphase	phase	Phase of the AC source, in degrees.

Pulse Source Function

SymSpice source of pulse signal generates stimulus as a trapezoidal pulse source function defined by the following parameters: initial and plateau values of voltage or current, leading and back fronts of signal, period from onset to onset.

Syntax in 'local' or 'spectre' modes

voltage source:

Name <(>n+ n-<)> vsource parameter=value ...

current source:

Name <(>n+ n-<)> isource parameter=value ...

Syntax in 'spice' mode

voltage source:

Vxxx n+ n- PU<LSE> <(>v1 v2 <td <tr <tf <pw <per>>>>> <)>

current source:

Ixxx n+ n- PU<LSE> <(>v1 v2 <td <tr <tf <pw <per>>>>> <)>

The arguments are defined as:

SPICE mode	SymSpice mode	Description
Vxxx, Ixxx	Name	Independent voltage or current source which will exhibit the pulse response.
n+	n+	Positive node.
n-	n-	Negative node.
-	vsource/isource	Reserved keyword to designate independent voltage/current source.
PULSE	type=pulse	Keyword for a pulsed time-varying source. For 'spice' mode, You may use the short form is "PU".
v1	v1	Initial value of the voltage or current, before the pulse onset (units of volts or amps)
v2	v2	Pulse plateau value (units of volts or amps).

td	delay	Delay time in seconds from the beginning of transient interval to the first onset ramp. Default=0.0 and negative values are considered as zero.
tr	rise	Duration of the onset ramp in seconds, from the initial value to the pulse plateau value (reverse transit time). .
tf	fall	Duration of the recovery ramp in seconds, from the pulse plateau back to the initial value (forward transit time).
pw	width	Pulse width (duration of v2) in seconds.
per	period	Period of waveform.
-	tc1	First order temperature coefficient.
-	tc2	Second order temperature coefficient.
m	m	Multiplicity factor (only for current source).
acmag	mag	Magnitude (RMS) of the AC source.
acphase	phase	Phase of the AC source, in degrees.

Below is a table showing the time-value relationship for a PULSE source:

Time	*Value*
0	v1
td	v1
td + tr	v2
td + tr + pw	v2
td + tr + pw + tf	v1
tstop	v1

Intermediate points are determined by linear interpolation.

Note: TSTEP is the printing increment, and TSTOP is the final time.

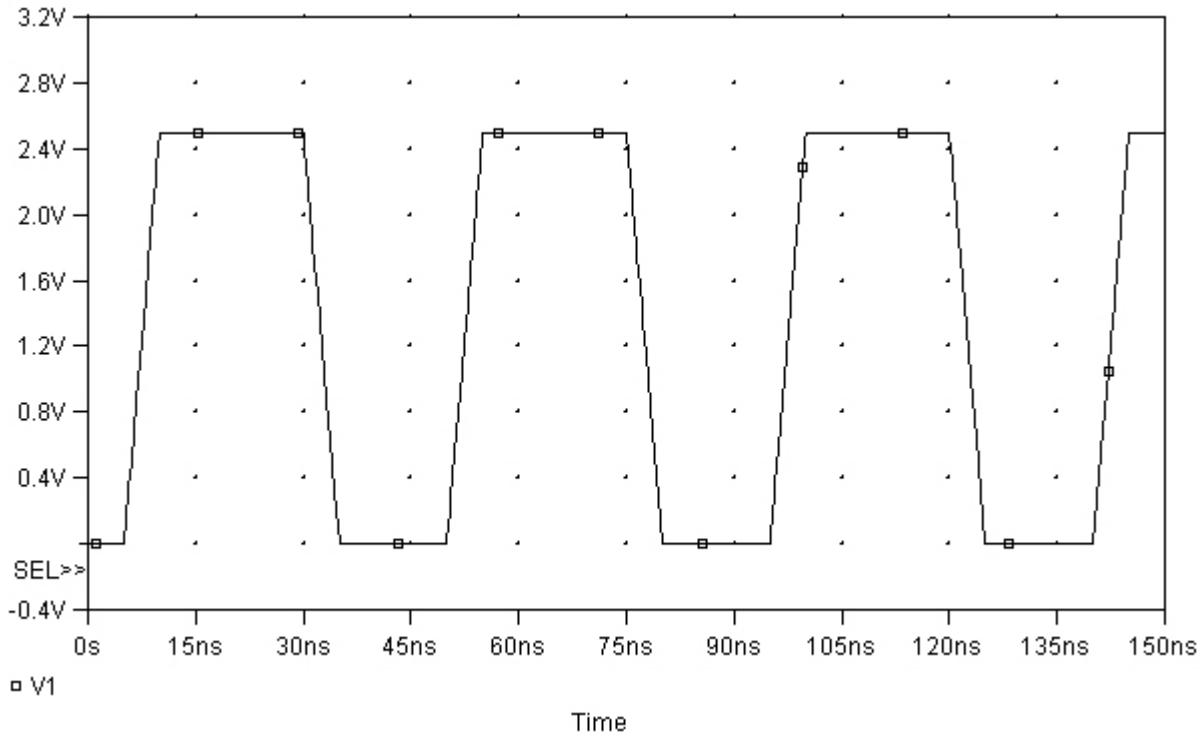


Figure 30. Pulse Source Function

Sinusoidal Source Function

SymSpice source of sinusoidal signal is defined by signal amplitude and frequency.

In case of TRAN-analysis the sinusoidal source will generate sinusoid signal.

For SymSpice sinusoidal signal is defined as:

Syntax in 'local' or 'spectre' modes

voltage source:

Name <(>n+ n-<)> vsource parameter=value ...

current source:

Name <(>n+ n-<)> isource parameter=value ...

Syntax of VSIN-source is defined by the following parameters:

Syntax in 'spice' mode

voltage source:

Vxxx n+ n- SIN <(> vo va <freq <td <damp <ph>>>> <)>

current source:

Ixxx n+ n- SIN <(> vo va <freq <td <damp <ph>>>> <)>

Source of AC-signal can be supplied to the input of circuit, simulating amplitude-frequency characteristic and phase-frequency characteristic.

In that case syntax will be defined by the following parameters:

Syntax in 'spice' mode

voltage source:

Vxxx n+ n- <<dc=> dcval> SIN <ac=acmag, <acphase>>

current source:

Iyyy n+ n- <<dc=> dcval> SIN <ac=acmag, <acphase>> <M=val>

The arguments are defined as:

SPICE mode	SymSpice mode	Description
acmag	mag	Magnitude (RMS) of the AC source.
acphase	phase	Phase of the AC source in degrees (for 'local' mode, when t=delay).
Vxxx, Ixxx	Name	Independent voltage or current source that will generate the sinusoidal response.
n+	n+	Positive node.
n-	n-	Negative node.
-	vsource/isource	Reserved keyword to designate independent voltage/current source.
SIN	type=sine	Keyword for a sinusoidal time-varying source.
vo	dc	Voltage or current offset in volts or amps.
va	ampl	First voltage or current amplitude in volts or amps.
freq	freq	First source frequency in Hz.
td	delay	Time delay before beginning the sinusoidal variation in seconds.
damp	damp	Damping factor in units of 1/seconds.
ph	sinephase	First phase delay in units of degrees.

dc	sinedc=dc	DC level for sinusoidal waveforms in volts or ampers.
m	m	Multiplicity factor (only for current source).
-	ampl2	Second voltage or current amplitude in volts or amps.
-	freq2	Second source frequency in Hz.
-	sinephase2	Second phase delay in units of degrees.
-	fmmodindex	FM index of modulation for sinusoidal waveform.
-	fmmodfreq	FM modulation frequency for sinusoidal waveform.
-	ammodindex	AM index of modulation for sinusoidal waveform.
-	ammodfreq	AM modulation frequency for sinusoidal waveform.
-	ammodphase	AM phase of modulation for sinusoidal waveform.
-	tc1	First order temperature coefficient.
-	tc2	Second order temperature coefficient.

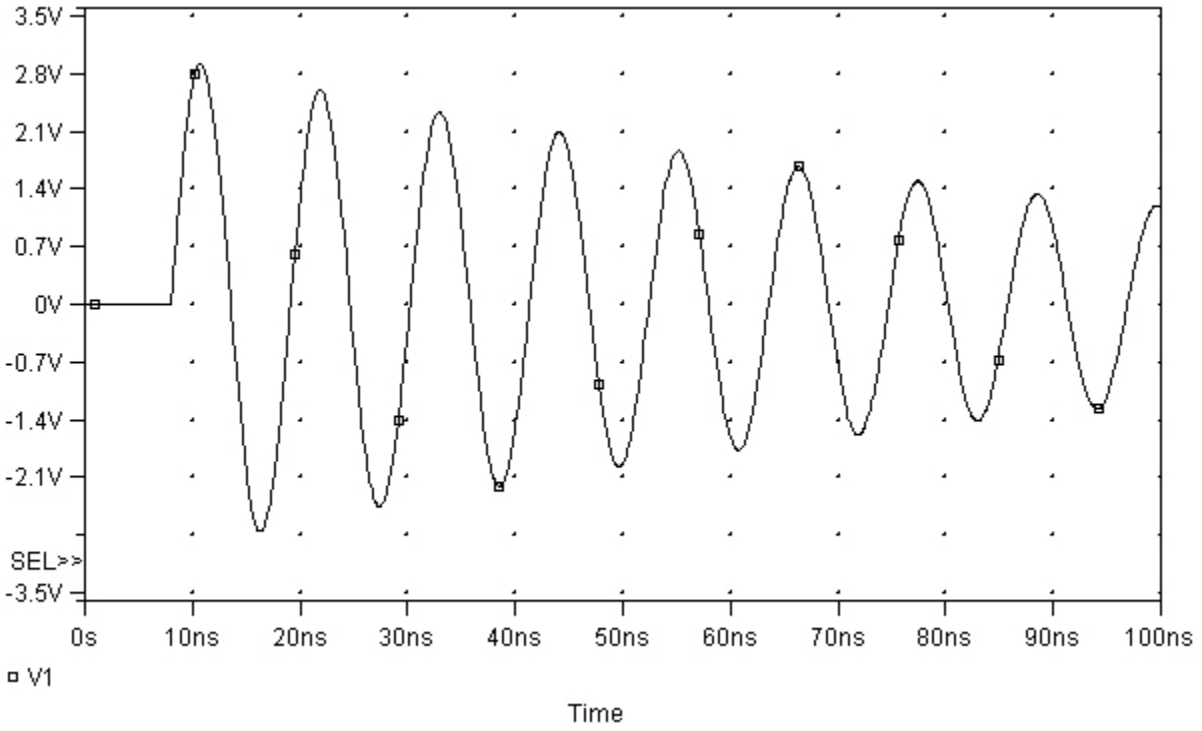


Figure 31. Sinusoidal Source Function

Examples

VIN 3 0 SIN (0 3 90meg 8ns 1e7 0)

Damped sinusoidal source connected between nodes 3 and 0. The waveform has a peak value of 3 V, an offset of 0 V, a 90 MHz frequency, a time delay of 8 ns, a damping factor of 1e7, and a phase delay of zero degree. See the figure for a plot of the source output.

Equations

In 'spice' mode SymSpice supports the following expressions for Sinosoidal Source Function:
The waveform shape is given by the following expressions:

Time *Value*

0 to td $vo + va \cdot SIN\left(\frac{2 \cdot \Pi \cdot \varphi}{360}\right)$

t to tstop $vo + va \cdot Exp\left[-(Time - td) \cdot \theta\right] \cdot SIN\left\{2 \cdot \Pi \cdot \left[freq \cdot (time - td) + \frac{\varphi}{360}\right]\right\}$

where TSTOP is the final time.

Exponential Source Function

SymSpice source of exponential signal generates stimulus as exponent which defined from one point to another.

Syntax in 'local' or 'spectre' modes

voltage source:

Name <(>n+ n-<)> vsource parameter=value ...

current source:

Name <(>n+ n-<)> isource parameter=value ...

Syntax in 'spice' mode

voltage source:

Vxxx n+ n- EXP <(> v1 v2 <td1 <tau1 <td2 <tau2>>>> <)>

current source:

Ixxx n+ n- EXP <(> v1 v2 <td1 <tau1 <td2 <tau2>>>> <)>

The arguments are defined as:

SPICE mode	SymSpice mode	Description

Vxxx, Ixxx	Name	Independent voltage or current source that will generate the exponential response.
n+	n+	Positive node.
n-	n-	Negative node.
EXP	type=exp	Keyword for an exponential time-varying source.
-	vsource/isource	Reserved keyword to designate independent voltage/current source.
v1	v1	Initial value of voltage or current in volts or amps
v2	v2	Pulsed value of voltage or current in volts or amps.
td1	td1	Rise delay time in seconds.
td2	td2	Fall delay time in seconds.
tau1	tau1	Rise time constant in seconds.
tau2	tau2	Fall time constant in seconds
-	delay	Waveform delay time.
-	tc1	First order temperature coefficient.
-	tc2	Second order temperature coefficient.
m	m	Multiplicity factor (only for current source).
acmag	mag	Magnitude (RMS) of the AC source, in volts
acphase	phase	

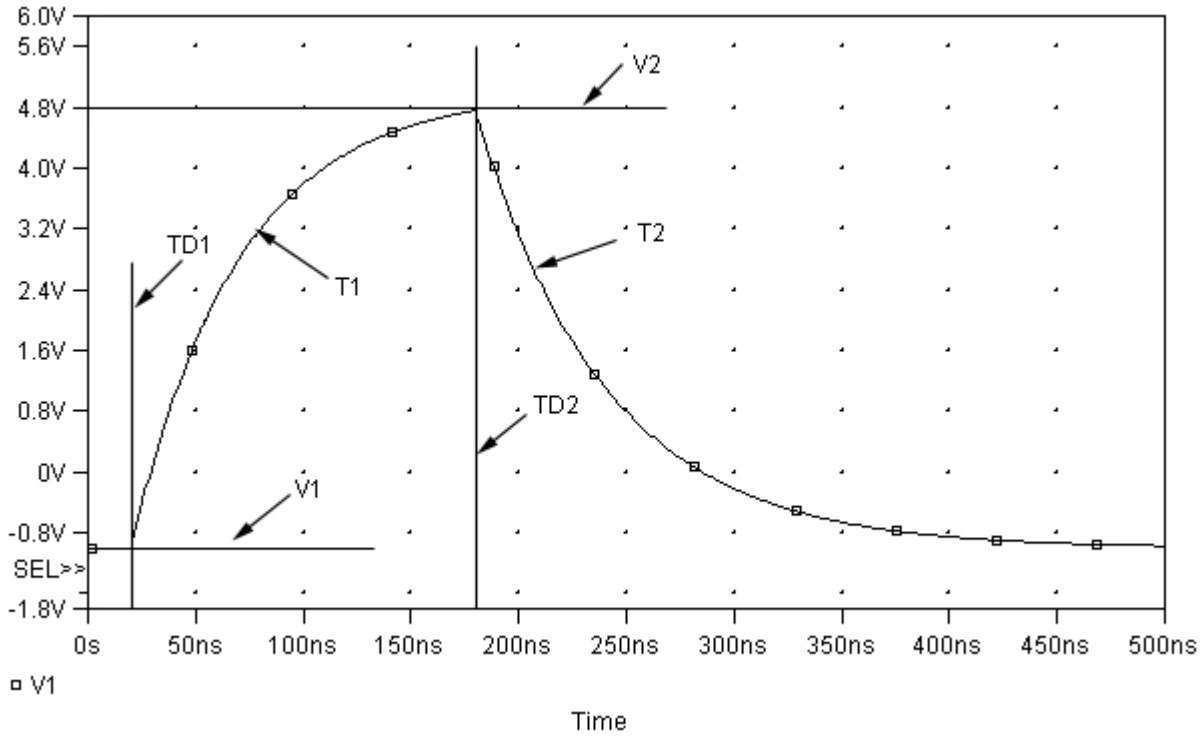


Figure 32. Exponential Source Function

Examples

VIN 3 0 EXP (-1 5 20ns 50ns 180ns 60ns)

The above example describes an exponential transient source that is connected between nodes 3 and 0. It has an initial t=0 voltage of -1 V and a final voltage of 5 V. The waveform rises exponentially from -1 V to 5 V with a time constant of 50 ns. At 180 ns it starts dropping to –1 V again, with a time constant of 60 ns.

```
*FILE: EXP.CIR THE EXPONENTIAL WAVEFORM
.PARAM V1=-4 V2=0 TD1=10ns TAU1=25ns TD2=60ns TAU2=30ns
V 1 0 EXP (V1 V2 TD1 TAU1 TD2 TAU2)
R 1 0 1
.TRAN 1ns 200ns START=0ns
.END
```

In the example a EXP voltage source has an initial t=0 voltage of -4 V and a final voltage of 0 V. The waveform rises exponentially from -4 V to 0 V with a time constant of 25 ns. At 60 ns it starts dropping to -4 V again, with a time constant of 30 ns.

Equations

The waveform shape is given by the following table of expressions:

Time	*Value*
0 to td1	v1
td1 to td2	$v1+(v2-v1)\cdot\left[1-\mathrm{Exp}\left(-\frac{\mathrm{Time}-td1}{\tau_1}\right)\right]$
td2 to tstop	$v1+(v2-v1)\cdot\left[1-\mathrm{Exp}\left(-\frac{td2-td1}{\tau_1}\right)\right]\cdot\mathrm{Exp}\left[\frac{-(\mathrm{Time}-td2)}{\tau_2}\right]$

Piecewise Linear Source Function

SymSpice source of piecewise linear signal generates stimulus which shape is defined by the following parameters: time and amplitude corresponded with time.

Syntax in 'local' or 'spectre' modes

voltage source:

Name <(>n+ n-<)> vsource parameter=value ...

current source:

Name <(>n+ n-<)> isource parameter=value ...

Syntax in 'spice' mode

voltage source:

Vxxx n+ n- PWL <(> t1 v1 <t2 v2 t3 v3...>

+<R <=repeat>> <TD=delay> <)>

current source:

Ixxx n+ n- PWL <(> t1 i1 <t2 i2 t3 i3...>

+<R <=repeat>> <TD=delay> <)>

ASPEC form

voltage source:

Vxxx n+ n- PL <(> v1 t1 <v2 t2 v3 t3...>

+<R <=repeat>> <TD=delay> <)>

current source

Ixxx n+ n- PL <(> i1 t1 <i2 t2 i3 t3...>

+<R <=repeat>> <TD=delay> <)>

The arguments are defined as:

SPICE mode	SymSpice mode	Description
Vxxx, Ixxx	Name	Independent voltage or current source that will generate the piecewise linear response.
n+	n+	Positive node.
n-	n-	Negative node.
PWL/PL	type=pwl	Keyword for a piecewise linear time-varying source.

-	vsource/isource	Reserved keyword to designate independent voltage/current source.
v1 v2 … vn or i1 i2 … in	-	Current or voltage values at corresponding timepoint.
t1 t2 … tn	-	Timepoint values where the corresponding current or voltage value is valid.
-	wave=[...]	Vector of time/value pairs that defines waveform.
-	offset	DC offset for the PWL waveform.
-	scale	Scale factor for the PWL waveform.
-	stretch	Scale factor for time given for the PWL waveform.
-	allbrkpns	All the points in the PWL waveform are breakpoints if set to yes. Possible values are no or yes.
-	pwlperiod	Period of the periodic PWL waveform.
-	twidth=pwlperiod/1000s	Transition width used when making PWL waveforms periodic.
r=repeat	pwlperiodstart	Keyword and time value to specify a repeating function. With no argument, the source repeats from the beginning of the function, "repeat" is time in units of seconds which specifies the start point of the waveform which is to be repeated. This time needs to be less than the greatest time point tn.
acmag	mag	Magnitude (RMS) of the AC source, in volts
acphase	phase	Phase of the AC source, in degrees.

Each pair of values (t1, v1) specifies that the value of the source is v1 (in volts or amps) at time t1. The value of the source at intermediate values of time is determined by linear interpolation between the time points.

Specify "pwlperiodstart"("R") to cause the function to repeat. You can specify a value after this "pwlperiodstart" to indicate the beginning of the function to be repeated: the repeat time must equal a breakpoint in the function. For example, if t1 = 1, t2 = 2, t3 = 3, and t4 = 4, "repeat" can be equal to 1, 2, or 3.

Examples

```
simulator lang=local

v1 1 0 source type=pwl coeffs=[0n 0 30n 0 45n 0 60n 5 75n 5 90n 0]
pwlperiodstarted=0n

r1 1 0 resistor r=1
```

v2 2 0 source type=pwl coeffs= [0n 0 30n 0 45n 0 60n 5 75n 5 90n 0] pwlperiodstarted=30n

r2 2 0 resistor r=1

save 1 2

tran tran start=0 stop=300n step=1n

In the examples two piecewise linear voltage sources have the same function. The first source has a repeat specified to start at the beginning of the function, whereas the second repeat starts at the first timepoint. See figure for the difference in responses.

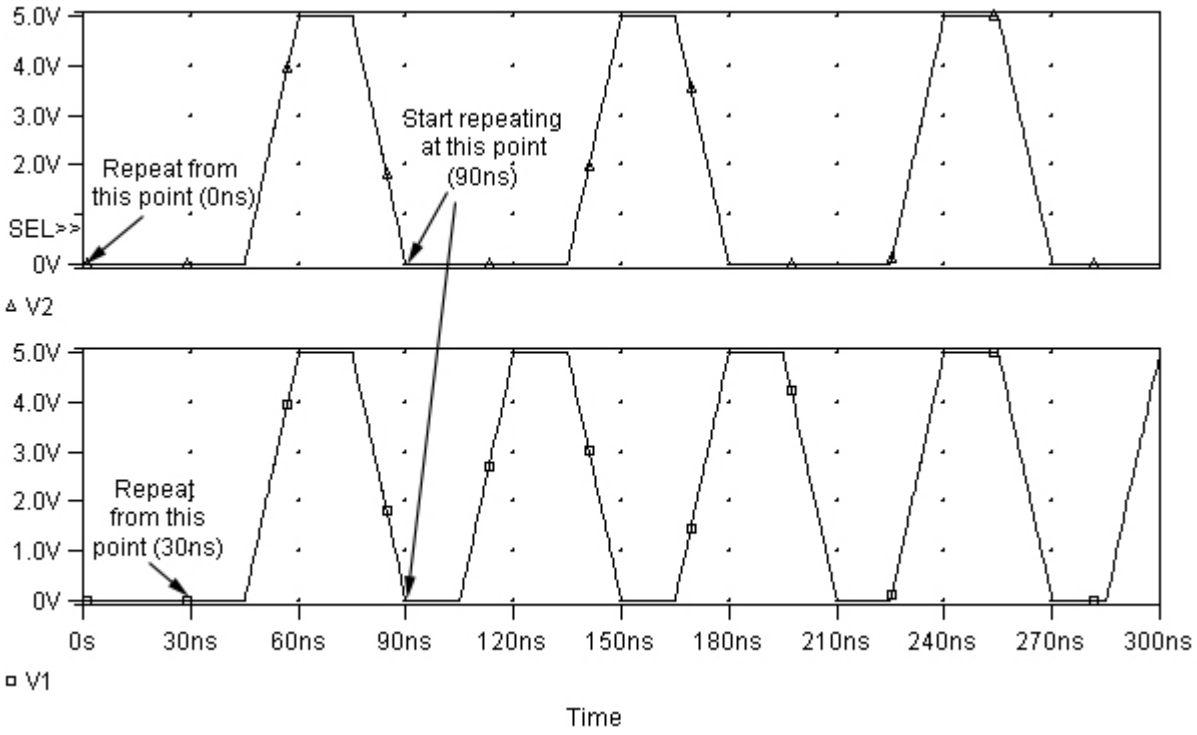


Figure 33. Results of Using the Repeat Function

Single-Frequency FM Source Function

SymSpice single-frequency frequency-modulated time-varying source is defined in the following way :

Syntax in 'local' or 'spectre' modes

use 'Sinusoidal Source Function'

Syntax in 'spice' mode

voltage source:

Vxxx n+ n- SFFM <(> vo va <fc <mdi <fs>>> <)>

current source:

Ixxx n+ n- SFFM <(> vo va <fc <mdi <fs>>> <)>

The arguments are as follows:

SPICE mode	SymSpice mode	Description
Vxxx, Ixxx	-	Independent voltage or current source which will generate the frequency-modulated response.
SFFM	-	Keyword for a single-frequency frequency-modulated time-varying source.
vo	-	Output voltage or current offset, in volts or amps.
va	-	Output voltage or current amplitude, in volts or amps.
fc	-	Carrier frequency in Hz. Default=1/TSTOP.
mdi	-	Modulation index which determines the magnitude of deviation from the carrier frequency. Values normally lie between 1 and 10. Default=0.0.
fs	-	Signal frequency in Hz. Default=1/TSTOP.
acmag	mag	Magnitude (RMS) of the AC source.
acphase	phase	Phase of the AC source, in degrees.

Examples

```
*FILE: SFFM.CIR THE SINGLE FREQUENCY FM SOURCE
V 1 0 SFFM (0, 5, 50meg, 15, 10meg)
R 1 0 1
.TRAN 1ns 150ns
.END
```

In the example a single-frequency frequency-modulated voltage source has an offset voltage of 0 volts, and a maximum voltage of 5 Volt. The carrier frequency is 50 MHz, and the signal is 10 MHz, with a modulation index of 15 (the maximum wavelength is roughly 15 times longer than the minimum).

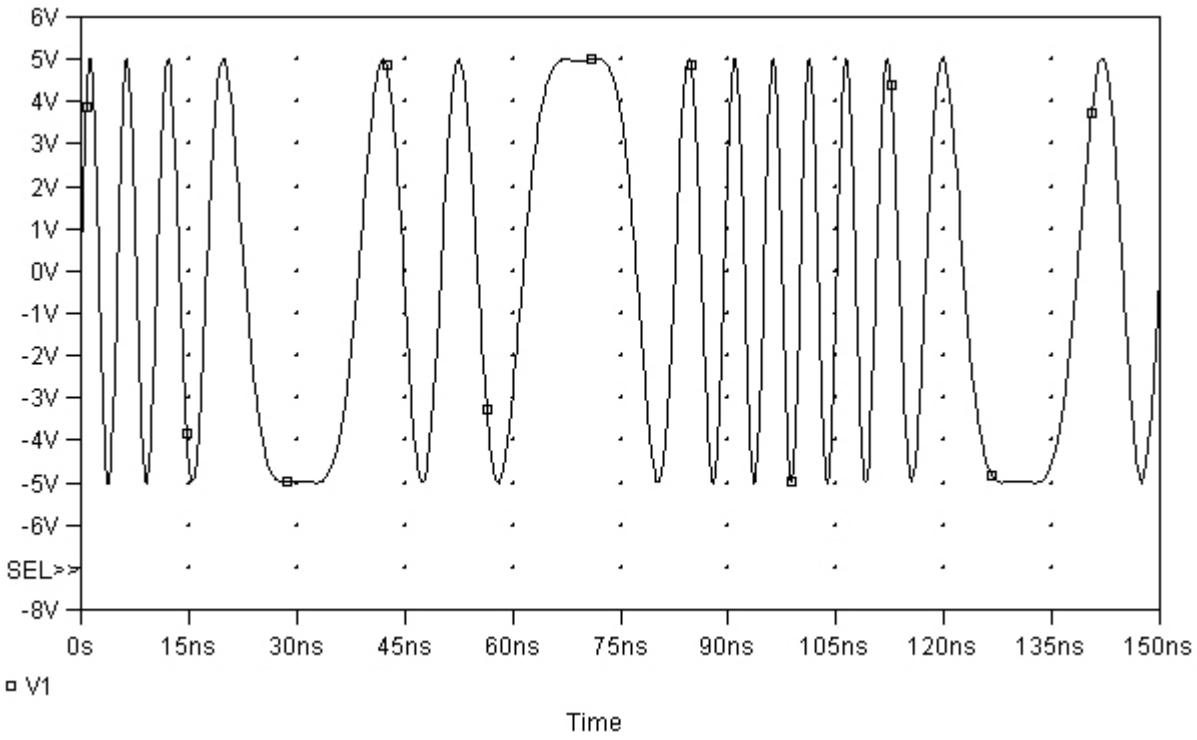


Figure 34. Single Frequency FM Sourcesss

Equations

The waveform shape is given by the following expression:

$$\text{soursevalu } e = v0 + va \cdot \text{SIN}\left[2 \cdot \pi \cdot fc \cdot \text{Time} + mdi \cdot \text{SIN}(2 \cdot \pi \cdot fc \cdot \text{Time})\right]$$

Single-Frequency AM Source

SymSpice single-frequency amplitude-modulated time-varying source is defined in the following way :

Syntax in 'local' or 'spectre' modes

use 'Sinusoidal Source Function'

Syntax in 'spice' mode

voltage source:

Vxxx n+ n- AM <(> sa oc fm fc <td><)>

current source:

Ixxx n+ n- AM <(> sa oc fm fc <td><)>

The arguments are as follows:

SPICE mode	SymSpice mode	Description
Vxxx, Ixxx	-	Independent voltage or current source which will generate the amplitude-modulated response.
AM	-	Keyword for a single-frequency amplitude-modulated time-varying source.
sa	-	Signal amplitude, in volts or amps. Default=0.0.
oc	-	Offset constant, a unitless constant that determines the absolute magnitude of the modulation. Default=0.0.
fm	-	Modulation frequency, in hertz. Default=1/TSTOP.
fc	-	Carrier frequency, in hertz. Default=0.0.
td	-	Delay time before the start of the signal, in seconds. Default=0.0.
acmag	mag	Magnitude (RMS) of the AC source.
acphase	phase	Phase of the AC source, in degrees.

Equations

The waveform shape is given by the following expression:

$$sourcevalue = sa \cdot [oc + SIN(2 \cdot \pi \cdot fm \cdot (Time - td))] \cdot SIN[2 \cdot \pi \cdot fc \cdot (Time - td)]$$

Dependent Sources Elements

A dependent source is either a voltage or current source whose value is proportional to some other voltage or current in the circuit. To the input of the dependent source signal is given directly from the independent source or through other circuit elements.

All input signals are defined by pairs: positive and negative values.The output signal is defined by the difference of node signals appearing across its inputs.

Depending on the number of controlled input signals dependent source elements can be conditionally divided into 1-, 2-, 3-, ..., n-pin. The number of input signals in the source description is indicated by own parameter and influence on calculation of mathematical expressions defining output dependences.
There are two types of dependent source elements controlled by voltage and two types of dependent source elements controlled by current.

The functions of the E, F, G, and H controlled elements are different.

- The E Element can be:
  – A voltage-controlled voltage source
  – A behavioral voltage source
  – An ideal op-amp.
  – An ideal transformer.
  – An ideal delay element.
  – A piecewise linear, voltage-controlled, multi-input AND, NAND, OR, or NOR gate.

- The F Element can be:
  – A current-controlled current source.
  – An ideal delay element.
  – A piecewise linear, current-controlled, multi-input AND, NAND, OR, or NOR gate.

- The G Element can be:
  – A voltage-controlled current source.
  – A behavioral current source.
  – A voltage-controlled resistor.
  – A piecewise linear, voltage-controlled capacitor.
  – An ideal delay element.
  – A piecewise linear, multi-input AND, NAND, OR, or NOR gate.

- The H Element can be:
  – A current-controlled voltage source.
  – An ideal delay element.
  – A piecewise linear, current-controlled, multi-input AND, NAND, OR, or NOR gate.

Polynomial Functions

Polynomial Function is the type of dependent source output characteristic dependence defined by specified coefficients.
The number of input signals during source description is indicated by NDIM parameter and influence on calculation of mathematical expressions defining output dependences. For example, view 3 types of polynomial functions POLY(NDIM) for circuit containing dependent source element: NDIM=1, 2, 3.

Value Description

POLY(1)	One-dimensional equation (function of one controlling variable).
POLY(2)	Two-dimensional equation (function of two controlling variables).
POLY(3)	Three-dimensional equation (function of three controlling variables).

Each polynomial equation includes polynomial coefficient parameters (P0, P1 … Pn), which you can set to explicitly define the equation.

One-Dimensional Function

If the function is one-dimensional (a function of one branch current or node voltage), the following expression determines the FV function value:

$$FV = P0 + (P1 \cdot FA) + (P2 \cdot FA^2) + (P3 \cdot FA^3) + (P4 \cdot FA^4) + (P5 \cdot FA^5) + \ldots$$

Parameter Description

FV	Controlled voltage or current, from the controlled source.
P0. . .PN	Coefficients of a polynomial equation.
FA	Controlling branch current, or nodal voltage.

Note: If you specify one coefficient in a one-dimensional polynomial, SymSpice assumes that the coefficient is P1 (P0 = 0.0). Use this as input for linear controlled sources.

The following controlled source statement is a one-dimensional function. This voltage-controlled current source connects to nodes n_5 and n_6.

```
g_vccs21_8 n_5 n_6 vccs poly ( 1 ) n_2 gnd 1 2
```

1. The single-dimension polynomial function parameter, POLY(1), informs SymSpice that g_vccs21_8 is a function of the difference of one nodal voltage pair. In this example, the voltage difference is between nodes n_2 and gnd, so FA=V(n_2,gnd).
2. The dependent source statement then specifies that P0=1 and P1=2. From the one-dimensional polynomial equation above, the defining equation for V(n_5,n_6) is:

$$V(n\_5, n\_6) = 1 + 2 \cdot V(n\_2, gnd)$$

Two-Dimensional Function

If the function is two-dimensional (that is, a function of two node voltages or two branch currents), the following expression determines FV:

$$FV = P0 + (P1 \cdot FA) + (P2 \cdot FB) + (P3 \cdot FA^2) + (P4 \cdot FA \cdot FB) + (P5 \cdot FB^2) + \\ + (P6 \cdot FA^3) + (P7 \cdot FA^2 \cdot FB) + (P8 \cdot FA \cdot FB^2) + (P9 \cdot FB^3) + \ldots$$

For a two-dimensional polynomial, the controlled source is a function of two nodal voltages or currents. To specify a two-dimensional polynomial, set POLY(2) in the controlled source statement. For example, generate a voltage-controlled source that specifies the controlled voltage, V(n_1,gnd), as:

$$V(n\_1, gnd) = 2 + 3 \cdot V(n\_3, gnd) + 2 \cdot V(gnd, gnd) + 2 \cdot V(n\_3, gnd)^2$$

To implement this function, use this controlled-source element statement:

e_vcvs21_4 n_1 gnd vcvs poly (2) n_3 gnd gnd gnd 2 3 2 2

This example specifies a controlled voltage source, which connects between nodes n_1 and gnd. Two differential voltages control this voltage source:

- Voltage difference between nodes n_3 and gnd.
- Voltage difference between nodes gnd and gnd.

That is, FA=V(n_3,gnd), and FB=V(gnd,gnd). The polynomial coefficients are:

- P0=2
- P1=3
- P2=2
- P3=2

Three-Dimensional Function

For a three-dimensional polynomial function, with FA, FB, and FC as its arguments, the following expression determines the FV function value:

$$FV = P0 + (P1 \cdot FA) + (P2 \cdot FB) + (F3 \cdot FC) + (P4 \cdot FA^2) + (P5 \cdot FA \cdot FB) + \\ + (P6 \cdot FA \cdot FC) + (P7 \cdot FB^2) + (P8 \cdot FB \cdot FC) + (P9 \cdot FC^2) + (P10 \cdot FA^3) + \\ + (P11 \cdot FA^2 \cdot FB) + (P12 \cdot FA^2 \cdot FC) + (P13 \cdot FA \cdot FB^2) + (P14 \cdot FA \cdot FB \cdot FC) + \\ + (P15 \cdot FA \cdot FC^2) + (P16 \cdot FB^3) + (P17 \cdot FB^2 \cdot FC) + (P18 \cdot FB \cdot FC^2) + \\ + (P19 \cdot FC^3) + (P20 \cdot FA^4) + \ldots$$

For example, generate a voltage-controlled source that specifies the voltage as:

$$V(n\_5, n\_6) = 1 + 2 \cdot V(n\_2, gnd) + 3 \cdot V(n\_3, gnd) + \\ + V(n\_2, gnd) \cdot V(n\_3, gnd)$$

Substitute these values into the voltage controlled voltage source statement:

g_vccs21_8 n_5 n_6 vccs poly (3) n_2 gnd n_1 gnd n_3 gnd 1 2 0 3 0 0 1

The preceding example specifies a controlled voltage source, which connects between nodes n_5 and n_6. Three differential voltages control this voltage source:

- Voltage difference between nodes n_2 and gnd.
- Voltage difference between nodes n_1 and gnd.
- Voltage difference between nodes n_3 and gnd.

That is:

- *FA*=*V*(n_2,gnd)
- *FB*=*V*(n_1,gnd)
- *FC*=*V*(n_3,gnd)

The statement defines the polynomial coefficients as:

- P0=1
- P1=2
- P3=3
- P6=4
- Other coefficients are zero.

Piecewise Linear Function

The piecewise linear function can be described by specifying measured data points. SymSpice automatically smooths the corners, to ensure derivative continuity and, as a result, better convergence.

To model bidirectional switch or transfer gates, G Elements use the NPWL and PPWL functions.

The piecewise linear function models multi-input AND, NAND,OR, and NOR gates. In this case, only one input determines the state of the output.

- In AND and NAND gates, the input with the smallest value determines the corresponding output of the gates.
- In OR and NOR gates, the input with the largest value determines the corresponding output of the gates.

The number of input control signals for multi-input type description is specified by `k` parameter.

Voltage-Controlled Voltage Source (VCVS)

Linear

Syntax in 'local' or 'spectre' modes

Name n+ n- in1+ in1- VCVS <gain=val> <tc1=val> <tc2=val>

+<scale=val> <max=val> <min=val> <abs=val> <delata=val>

Syntax in 'spice' mode

Exxx n+ n- <VCVS> in1+ in1- gain <tc1=val> <tc2=val> <scale=val>

+ <max=val> <min=val> <abs=val> <IC=val>

Polynomial (POLY)

Syntax in 'local' or 'spectre' modes

Name n+ n- in1+ in1- ... PVCVS <coeffs=[p0 p1 ...]> <gain=val> <tc1=val>

+<tc2=val> <scale=val> <max=val> <min=val> <abs=val> <delta=val>

Syntax in 'spice' mode

Exxx n+ n- <VCVS> POLY(NDIM) in1+ in1- ...

+ inndim+ inndim- <tc1=val> <tc2=val> <scale=val>

+ <min=val> <max=val> <abs=1> p0 <p1…> <IC=val>

Piecewise Linear (PWL)

Syntax in 'local' or 'spectre' modes

Name n+ n- in1+ in1- VCVS <pwl=[x1 y1 ...]> <tc1=val> <tc2=val>

+<scale=val> <stretch=val> <abs=val> <delta=val>

Syntax in 'spice' mode

Exxx n+ n- <VCVS> PWL(1) in+ in- <delta=val> <scale=val>

+<tc1=val> <tc2=val> x1,y1 x2,y2 ... x100,y100 <IC=val>

Multi-Input Gates

Syntax in 'local' or 'spectre' modes

Name n+ n- in1+ in1-... VCVS <type=val> parameter=value

You can use parameters 'line' or 'pwl' vcvs source's

Syntax in 'spice' mode

Exxx n+ n- <VCVS> gatetype(j) in1+ in1- ... inj+ inj-

+ <delta=val> <tc1=val> <tc2=val> <scale=val>

+ x1,y1 ... x100,y100 <IC=val>

Delay Element

Syntax in 'local' or "spectre modes

Syntax in 'spice' mode

Exxx n+ n- <VCVS> DELAY in+ in- td=val <scale=val>

+ <tc1=val> <tc2=val> <npdelay=val>

Behavioral Voltage Source

Syntax in 'local' or 'spectre' modes

Syntax in 'spice' mode

Exxx n+ n- VOL='equation' <MAX=val> <MIN=val>

Ideal Op-Amp

Syntax in 'local' or 'spectre' modes

Syntax in 'spice' mode

Exxx n+ n- OPAMP in+ in-

You can also substitute Level=1 in place of OPAMP:

Exxx n+ n- in+ in- level=1

Ideal Transformer

Syntax in 'local' or 'spectre' modes

Syntax in 'spice' mode

Exxx n+ n- TRANSFORMER in+ in- k

You can also substitute Level=2 in place of TRANSFORMER:

Exxx n+ n- in+ in- level=2 k

Parameter Description:

SPICE mode	SymSpice mode	Description
Exxx	Name	Voltage-controlled element name. For 'spice' mode must begin with E, followed by up to 1023 alphanumeric characters
n+/-	n+/-	Positive or negative node of a controlled element
in +/-	in +/-	Positive or negative controlling nodes. Specify one pair for each dimension.
VCVS	VCVS	Keyword for a voltage-controlled voltage source. VCVS is a reserved word; do not use it as a node name.
gain	gain	Voltage gain.
min	min	Minimum output voltage value. The default is undefined, and sets no minimum value.
max	max	Maximum output voltage value. The default is undefined, and sets no maximum value.
abs	abs	Output is an absolute value, if abs=on in 'local' mode or abs=1 in 'spice' mode;

delta	delta	Controls the curvature of the piecewise linear corners. This parameter defaults to one-fourth of the smallest distance between breakpoints. The maximum is one-half of the smallest distance between breakpoints
POLY(NDIM)	PVCVS	Keyword for the polynomial function. NDIM-Number of polynomial dimensions. If you do not specify POLY(ndim), SymSpice assumes a one-dimensional polynomial. Ndim must be a positive number.
p0 p1 ..	coeffs=[p0 p1...]	The polynomial coefficients. If you specify one coefficient, SymSpice assumes that it is p1 (p0=0.0). (For details, see Polynomial Functions).
PWL(1)	-	Keyword for the piecewise linear function.
x1,y1	pwl=[x1 y1 ...]	The PWL vector. x1- Controlling voltage. The x values must be in increasing order. y1 - Corresponding element values of x.
-	stretch	Scale factor for the PWL controlling voltage
gatetype(j)	type	Can be AND, NAND, OR, or NOR. j represents the number of inputs of the gate. x and y represent the piecewise linear variation of output, as a function of input. In multi-input gates, only one input determines the state of the output.
DELAY	-	Keyword for the delay element. Same as for the voltage-controlled voltage source, except it has an associated propagation delay, TD. This element adjusts propagation delay in macro (subcircuit) modeling.
td	-	Keyword for the time (propagation) delay
npdela	-	Sets the number of data points to use in delay simulations. The default value is the larger of either 10, or the smaller of TD/tstep and tstop/tstep. $NPDELAY_{default} = \max\left[\frac{\min\ < TD, tstop >}{tstep}, 10\right]$ That is, The TRAN statement specifies tstep and tstop values.
OPAMP or Level=1	-	The keyword for an ideal op-amp element. OPAMP is a SymSpice reserved word; do not use it as a node name.
TRANSFORMER or Level=2	-	Keyword for an ideal transformer. TRANSFORMER is a reserved word; do not use it as a node name.
k	-	Ideal transformer turn ratio: $V(in+,in-) = k \cdot V(n+,n-)$ or, number of gates input.

scale	scale	Multiplier for the element value.
tc1, tc2	tc1, tc2	First-order and second-order temperature coefficients. Temperature changes update the scale: $SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$
ic	-	Initial condition: initial estimate of controlling voltage value(s). If you do not specify ic, default=0.0.

Examples

Ideal OpAmp:

v_1 in_1 gnd pulse 0 -2.5 1n 10n 10n 10n 50n

r_1 in_1 in_2 100

e_1 out_3 gnd OPAMP in_2 gnd

r_2 in_2 out_3 1k

Polynomial:

e_vcvs21_4 n_1 gnd vcvs poly (2) n_3 gnd gnd gnd 2 3 2 2

The two-dimensional polynomial equation specifies P0=2, P1=3, P2=2, P3=2, FA=V(n_3,gnd), FB=V(gnd,gnd)

Ideal Transformer:

e_transformer1_3 gnd n_3 transformer n_2 n_1 4

Current-Controlled Current Source (CCCS)

Linear

Syntax in 'local' or 'spectre' modes

Name n+ n- CCCS probe=val <gain=val> <tc1=val> <tc2=val>

+<scale=val> <max=val> <min=val> <abs=val> <delata=val>

Syntax in 'spice' mode

Fxxx n+ n- <CCCS> vin1 gain <tc1=val> <tc2=val> <scale=val>

+ <max=val> <min=val> <abs=val> <m=val> <IC=val> <m=val>

Polynomial (POLY)

Syntax in 'local' or 'spectre' modes

Name n+ n- PCCCS probes=[...] <coeffs=[p0 p1 ...]> <gain=val>

+<tc1=val> <tc2=val> <scale=val> <max=val> <min=val>

+<abs=val> <delta=val> <m=val>

Syntax in 'spice' mode

Fxxx n+ n- <CCCS> POLY(NDIM) vin1 <... vinndim>

+ <tc1=val> <tc2=val> <scale=val> <min=val> <max=val>

+<abs=1> p0 <p1…> <IC=val> <m=val>

Piecewise Linear (PWL)

Syntax in 'local' or 'spectre' modes

Name n+ n- CCCS probe=val <pwl=[x1 y1 ...]> <tc1=val> <tc2=val>

+<scale=val> <stretch=val> <abs=val> <delta=val> <m=val>

Syntax in 'spice' mode

Fxxx n+ n- <CCCS> PWL(1) vin1 <delta=val>

+ <scale=val> <tc1=val> <tc2=val> x1,y1 x2,y2 ...

+ x100,y100 <IC=val> <m=val>

Multi-Input Gates

Syntax in 'local' or 'spectre' modes

Name n+ n- CCCS probes=[...] <type=val> parameter=value

You can use parameters 'line' or 'pwl' vcvs source's

Syntax in 'spice' mode

Fxxx n+ n- <CCCS> gatetype(j) vin1 ...vinj

+ <delta=val> <tc1=val> <tc2=val> <scale=val>

+ x1,y1 ... x100,y100 <IC=val> <m=val>

Delay Element

Syntax in 'local' or 'spectre' modes

Syntax in 'spice' mode

Fxxx n+ n- <CCCS> DELAY vin1 td=val <scale=val>

+ <tc1=val> <tc2=val> <npdelay=val> <m=val>

Parameter Description:

SPICE mode	SymSpice mode	Description
Fxxx	Name	Corrent-controlled element name. For 'spice' mode must begin with F, followed by up to 1023 alphanumeric characters
n+/-	n+/-	Positive or negative node of a controlled element
vin	probe probes=[...]	Names of voltage sources, through which the controlling current flows. Specify one name for each dimension. Use 'probe' if you specify one voltage source, and use 'probes=[...]' for several voltage sources.
CCCS	CCCS	Keyword for a voltage-controlled voltage source. CCCS is a reserved word; do not use it as a node name.
gain	gain	Voltage gain.
min	min	Minimum value of the current. The default is undefined, and sets no minimum value.
max	max	Maximum value of the current. The default is undefined, and sets no maximum value.
abs	abs	Output is an absolute value, if abs=on in 'local' mode or abs=1 in 'spice' mode;

delta	delta	Controls the curvature of the piecewise linear corners. This parameter defaults to one-fourth of the smallest distance between breakpoints. The maximum is one-half of the smallest distance between breakpoints
POLY(NDIM)	PCCCS	Keyword for the polynomial function. NDIM-Number of polynomial dimensions. If you do not specify POLY(ndim), SymSpice assumes a one-dimensional polynomial. Ndim must be a positive number.
p0 p1 ..	coeffs=[p0 p1...]	The polynomial coefficients. If you specify one coefficient, SymSpice assumes that it is p1 (p0=0.0). (For details, see Polynomial Functions).
PWL(1)	-	Keyword for the piecewise linear function.
x1,y1	pwl=[x1 y1 ...]	The PWL vector. x1- Controlling current. The x values must be in increasing order. y1 - Corresponding element values of x.
-	stretch	Scale factor for the PWL controlling current.
gatetype(j)	type	Can be AND, NAND, OR, or NOR. j represents the number of inputs of the gate. x and y represent the piecewise linear variation of output, as a function of input. In multi-input gates, only one input determines the state of the output.
DELAY	-	Keyword for the delay element. Same as for the voltage-controlled voltage source, except it has an associated propagation delay, TD. This element adjusts propagation delay in macro (subcircuit) modeling.
td	-	Keyword for the time (propagation) delay
npdela	-	Sets the number of data points to use in delay simulations. The default value is the larger of either 10, or the smaller of TD/tstep and tstop/tstep. $NPDELAY_{default} = \max\left[\frac{\min\ < TD, tstop >}{tstep}, 10\right]$ That is, The TRAN statement specifies tstep and tstop values.
scale	scale	Multiplier for the element value.
tc1, tc2	tc1, tc2	First-order and second-order temperature coefficients. Temperature changes update the scale: $SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$
m	m	Number of replications of the element, in parallel.
ic	-	Initial condition: initial estimate of controlling voltage value(s). If you do not specify ic, default=0.0.

Examples

Delay Element:

f_cccs1_0 n_3 gnd cccs delay v_vdc1_1 td=1e-009.

Voltage-Controlled Current Source (VCCS)

General Syntax

Voltage-Controlled Current Source (VCCS)

Linear

Syntax in 'local' or 'spectre' modes

Name n+ n- in1+ in1- VCCS <gm=val> <tc1=val> <tc2=val>

+<scale=val> <max=val> <min=val> <m=val> <abs=val> <delata=val>

Syntax in 'spice' mode

Gxxx n+ n- <VCCS> in1+ in1- transcond <tc1=val> <tc2=val> <scale=val>

+ <max=val> <min=val> <abs=val> <m=val> <IC=val>

Polynomial (POLY)

Syntax in 'local' or 'spectre' modes

Name n+ n- in1+ in1- ... PVCCS <coeffs=[p0 p1 ...]> <gm=val>

+<tc1=val> <tc2=val> <scale=val> <max=val> <min=val>

+ <m=val> <abs=val> <delta=val>

Syntax in 'spice' mode

Gxxx n+ n- <VCCS> POLY(NDIM) in1+ in1- ...

+ inndim+ inndim- <tc1=val> <tc2=val> <scale=val> <m=val>

+ <min=val> <max=val> <abs=1> p0 <p1…> <IC=val>

Piecewise Linear (PWL)

Syntax in 'local' or 'spectre' modes

Name n+ n- in1+ in1- VCCS <pwl=[x1 y1 ...]> <tc1=val> <tc2=val>

+<scale=val> <stretch=val> <m=val> <abs=val> <delta=val>

Syntax in 'spice' mode

Gxxx n+ n- <VCCS> PWL(1) in+ in- <delta=val>

+ <scale=val> <m=val> <tc1=val> <tc2=val> <m=val>

+ x1,y1 x2,y2 ... x100,y100 <IC=val> <smooth=val>

Gxxx n+ n- <VCCS> NPWL(1) in+ in- <delta=val>

+ <scale=val> <m=val> <tc1=val> <tc2=val> <m=val>

+ x1,y1 x2,y2 ... x100,y100 <IC=val> <smooth=val>

Gxxx n+ n- <VCCS> PPWL(1) in+ in- <delta=val>

+ <scale=val> <m=val> <tc1=val> <tc2=val> <m=val>

+ x1,y1 x2,y2 ... x100,y100 <IC=val> <smooth=val>

Multi-Input Gates

Syntax in 'local' or 'spectre' modes

Name n+ n- in1+ in1-... VCCS <type=val> parameter=value

You can use parameters 'line' or 'pwl' vcvs source's

Syntax in 'spice' mode

Gxxx n+ n- <VCCS> gatetype(j) in1+ in1- ... inj+ inj-

+ <delta=val> <tc1=val> <tc2=val> <scale=val> <m=val>

+ x1,y1 ... x100,y100 <IC=val>

Delay Element

Syntax in 'local' or 'spectre' modes

Syntax in 'spice' mode

Gxxx n+ n- <VCCS> DELAY in+ in- td=val <scale=val> <m=val>

+ <tc1=val> <tc2=val> <npdelay=val>

Behavioral Voltage Source

Syntax in 'local' or 'spectre' modes

Syntax in 'spice' mode

Gxxx n+ n- CUR='equation' <MAX=val> <MIN=val> <m=val> <scale=val>

Voltage-Controlled Resistor (VCR)

Syntax in 'local' mode

Syntax in 'spice' mode

The Level=1 syntax is:

Linear

Gxxx n+ n- VCR in+ in- transfactor <min=val> <max=val>

+ <scale=val> <m=val> <tc1=val> <tc2=val> <IC=val>

Polynomial (POLY)

Gxxx n+ n- VCR POLY(NDIM) in1+ in1- ...

+ <inndim+ inndim-> <min=val> <max=val> P0 <P1…>

+ <scale=val> <m=val> <tc1=val> <tc2=val> <IC=val>

Piecewise Linear (PWL)

Gxxx n+ n- VCR PWL(1) in+ in- <delta=val> <scale=val>

+ <m=val> <tc1=val> <tc2=val> x1,y1 x2,y2 ... x100,y100

+ <IC=val> <smooth=val>

Gxxx n+ n- VCR NPWL(1) in+ in- <delta=val> <scale=val>

+ <m=val> <tc1=val> <tc2=val> x1,y1 x2,y2 ... x100,y100

+ <IC=val> <smooth=val>

Gxxx n+ n- VCR PPWL(1) in+ in- <delta=val> <scale=val>

+ <m=val> <tc1=val> <tc2=val> x1,y1 x2,y2 ... x100,y100

+ <IC=val> <smooth=val>

Multi-Input Gates

Gxxx n+ n- VCR gatetype(k) in1+ in1- ... ink+ ink-

+ <deltla=val> <tc1=val> <tc2=val> <scale=val> <m=val>

+ x1,y1 ... x100,y100 <IC=val>

Voltage-Controlled Capacitor (VCCAP)

Gxxx n+ n- VCCAP PWL(1) in+ in- <DELTA=val>

+ <SCALE=val> <M=val> <TC1=val> <TC2=val>

+ x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Parameter Description

SPICE mode	SymSpice mode	Description
Gxxx	Name	Voltage-controlled element name. For 'spice' mode must begin with G, followed by up to 1023 alphanumeric characters
n+/-	n+/-	Positive or negative node of a controlled element
in +/-	in +/-	Positive or negative controlling nodes. Specify one pair for each dimension.
VCCS	VCCS	Keyword for a voltage-controlled voltage source. VCCS is a reserved word; do not use it as a node name.
transconductance	gm	Voltage-to-corrent conversion factor.
min	min	Minimum value of the current or resistance. The default is undefined, and sets no minimum value.

max	max	Maximum value of the current or resistance. The default is undefined, and sets no maximum value.
abs	abs	Output is an absolute value, if abs=on in 'local' mode or abs=1 in 'spice' mode;
delta	delta	Controls the curvature of the piecewise linear corners. This parameter defaults to one-fourth of the smallest distance between breakpoints. The maximum is one-half of the smallest distance between breakpoints
POLY(NDIM)	PVCCS	Keyword for the polynomial function. NDIM-Number of polynomial dimensions. If you do not specify POLY(ndim), SymSpice assumes a one-dimensional polynomial. Ndim must be a positive number.
p0 p1 ..	coeffs=[p0 p1...]	The polynomial coefficients. If you specify one coefficient, SymSpice assumes that it is p1 (p0=0.0). (For details, see Polynomial Functions).
PWL(1)	-	Keyword for the piecewise linear function.
NPWL	-	Models symmetrical bidirectional switch/transfer gate, NMOS.
PPWL	-	Models symmetrical bidirectional switch/transfer gate, PMOS.
x1,y1	pwl=[x1 y1 ...]	The PWL vector. x1- Controlling voltage. The x values must be in increasing order. y1 - Corresponding element values of x.
-	stretch	Scale factor for the PWL controlling voltage
gatetype(j)	type	Can be AND, NAND, OR, or NOR. j represents the number of inputs of the gate. x and y represent the piecewise linear variation of output, as a function of input. In multi-input gates, only one input determines the state of the output.
DELAY	-	Keyword for the delay element. Same as for the voltage-controlled voltage source, except it has an associated propagation delay, TD. This element adjusts propagation delay in macro (subcircuit) modeling.
td	-	Keyword for the time (propagation) delay
npdela	-	Sets the number of data points to use in delay simulations. The default value is the larger of either 10, or the smaller of TD/tstep and tstop/tstep. That is, $NPDELAY_{default} = \max\left[\frac{\min\ < TD, tstop >}{tstep}, 10\right]$

		The TRAN statement specifies tstep and tstop values.
smooth	-	For piecewise-linear, dependent-source elements, SMOOTH selects the curve-smoothing method. A curve-smoothing method simulates exact data points that you provide. You can use this method to make SymSpice simulate specific data points, which correspond to either measured data or data sheets.
VCR	-	Keyword for the voltage controlled resistor element. VCR is a reserved SymSpice keyword; do not use it as a node name.
transfactor	-	Voltage-to-resistance conversion factor.
VCCAP	-	Keyword for voltage-controlled capacitance element. VCCAP is a reserved SymSpice keyword; do not use it as a node name.
m	m	Number of replications of the elements in parallel.
m	m	
ic	-	Initial condition: initial estimate of controlling voltage value(s). If you do not specify ic, default=0.0.

NPWL and PPWL Functions

SymSpice uses either Level=2 (NPWL) or Level=3 (PPWL), based on the relationship of the (n+, n-) and (in+, in-) nodes.
Use the NPWL and PPWL functions to interchange the n+ and n- nodes, but use the same transfer function. The following summarizes this action:

NPWL Function

For the in- node connected to n+:

- If v(n+,n-) < 0, then the controlling voltage is v(in+,in-).
- Otherwise, the controlling voltage is v(in+,n-).

For the in- node connected to n-:

- If v(n+,n-) > 0, then the controlling voltage is v(in+,in-).
- Otherwise, the controlling voltage is v(in+,n+).

PPWL Function

For the in- node, connected to n+:

- If v(n+,n-) > 0, then the controlling voltage is v(in+,in-).
- Otherwise, the controlling voltage is v(in+,n-).

For the in- node, connected to n-:

- If v(n+,n-) < 0, then the controlling voltage is v(in+,in-).

- Otherwise, the controlling voltage is v(in+,n+).

If the in- node does not connect to either n+ or n-, then SymSpice changes NPWL and PPWL to PWL.

Examples

Voltage-Controlled Capacitor:
g_vccap1_2 n_2 gnd vccap pwl (1) n_1 gnd -2,4e-012 4,8e-012
The capacitance value across the *(n_2,gnd)* nodes varies linearly from 4e-012 to 8e-012, the voltage across the *(n_1,gnd)* nodes varies from -2 V to 4 V.

Polynomial:
g_vccs21_8 n_5 n_6 vccs poly (2) n_2 gnd n_1 gnd 1 2 0 3 0 0
The two-dimensional polynomial equation specifies P0=1, P1=2, P2=0, P3=3, P4=0, P5=0, FA=V(n_2,gnd), FB=V(n_1,gnd).

Multi-Input Gate:
g_vccs21_8 n_6 n_5 vccs and (2) n_2 gnd n_1 gnd 1,2 2,3 4,8
The above example represents two-input *AND* gate, the inputs are voltages at the *n_2* and *n_1* nodes.

Voltage-Controlled Resistor:
g_vcr1_3 n_4 gnd vcr npwl (1) n_1 gnd 2,4 4,8

Behavioral Current Source:
*g_bcs1_5 n_3 n_4 cur='5*v(n_2)+sqrt(v(n_2))'*

Current-Controlled Voltage Source (CCVS)

General Syntax

Linear

Syntax in 'local' or 'spectre' modes

Name n+ n- CCVS probe=val <gain=val> <tc1=val> <tc2=val>

+<scale=val> <max=val> <min=val> <abs=val> <delata=val>

Syntax in 'spice' mode

Fxxx n+ n- <CCVS> vin1 transresistance <tc1=val> <tc2=val>

+<scale=val> <max=val> <min=val> <abs=val> <m=val> <IC=val>

Polynomial (POLY)

Syntax in 'local' or 'spectre' modes

Name n+ n- PCCVS probes=[...] <coeffs=[p0 p1 ...]> <gain=val>

+<tc1=val> <tc2=val> <scale=val> <max=val> <min=val>

+<abs=val> <delta=val> <m=val>

Syntax in 'spice' mode

Hxxx n+ n- <CCVS> POLY(NDIM) vin1 <... vinndim>

+ <tc1=val> <tc2=val> <scale=val> <min=val> <max=val>

+<abs=1> p0 <p1…> <IC=val>

Piecewise Linear (PWL)

Syntax in 'local' or 'spectre' modes

Name n+ n- CCVS probe=val <pwl=[x1 y1 ...]> <tc1=val> <tc2=val>

+<scale=val> <stretch=val> <abs=val> <delta=val>

Syntax in 'spice' mode

Hxxx n+ n- <CCVS> PWL(1) vin1 <delta=val><scale=val>

+<tc1=val> <tc2=val> x1,y1 x2,y2 ... x100,y100 <IC=val>

Multi-Input Gates

Syntax in 'local' or 'spectre' modes

Name n+ n- CCVS probes=[...] <type=val> parameter=value

You can use parameters 'line' or 'pwl' vcvs source's

Syntax in 'spice' mode

Hxxx n+ n- <CCVS> gatetype(j) vin1 ...vinj

+ <delta=val> <tc1=val> <tc2=val> <scale=val>

+ x1,y1 ... x100,y100 <IC=val>

Delay Element

Syntax in 'local' or 'spectre' modes

Syntax in 'spice' mode

Hxxx n+ n- <CCVS> DELAY vin1 td=val <scale=val>

+ <tc1=val> <tc2=val> <npdelay=val>

Parameter Description:

SPICE mode	SymSpice mode	Description
Hxxx	Name	Corrent-controlled element name. For 'spice' mode must begin with H, followed by up to 1023 alphanumeric characters
n+/-	n+/-	Positive or negative node of a controlled element
vin	probe probes=[...]	Names of voltage sources, through which the controlling current flows. Specify one name for each dimension. Use 'probe' if you specify one voltage source, and use 'probes=[...]' for several voltage sources.
CCVS	CCVS	Keyword for a voltage-controlled voltage source. CCVS is a reserved word; do not use it as a node name.
gain	gain	Voltage gain.
min	min	Minimum output voltage value. The default is undefined, and sets no minimum value.
max	max	Maximum output voltage value. The default is undefined, and sets no maximum value.

abs	abs	Output is an absolute value, if abs=on in 'local' mode or abs=1 in 'spice' mode;
delta	delta	Controls the curvature of the piecewise linear corners. This parameter defaults to one-fourth of the smallest distance between breakpoints. The maximum is one-half of the smallest distance between breakpoints
POLY(NDIM)	PCCVS	Keyword for the polynomial function. NDIM-Number of polynomial dimensions. If you do not specify POLY(ndim), SymSpice assumes a one-dimensional polynomial. Ndim must be a positive number.
p0 p1 ..	coeffs=[p0 p1...]	The polynomial coefficients. If you specify one coefficient, SymSpice assumes that it is p1 (p0=0.0). (For details, see Polynomial Functions).
PWL(1)	-	Keyword for the piecewise linear function.
x1,y1	pwl=[x1 y1 ...]	The PWL vector. x1- Controlling current. The x values must be in increasing order. y1 - Corresponding element values of x.
-	stretch	Scale factor for the PWL controlling current.
gatetype(j)	type	Can be AND, NAND, OR, or NOR. j represents the number of inputs of the gate. x and y represent the piecewise linear variation of output, as a function of input. In multi-input gates, only one input determines the state of the output.
DELAY	-	Keyword for the delay element. Same as for the voltage-controlled voltage source, except it has an associated propagation delay, TD. This element adjusts propagation delay in macro (subcircuit) modeling.
td	-	Keyword for the time (propagation) delay
npdela	-	Sets the number of data points to use in delay simulations. The default value is the larger of either 10, or the smaller of TD/tstep and tstop/tstep. $NPDELAY_{default} = \max\left[\frac{\min < TD, tstop >}{tstep}, 10\right]$ That is, The TRAN statement specifies tstep and tstop values.
scale	scale	Multiplier for the element value.
tc1, tc2	tc1, tc2	First-order and second-order temperature coefficients. Temperature changes update the scale: $SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$

ic	-	Initial condition: initial estimate of controlling voltage value(s). If you do not specify ic, default=0.0.

Examples

Delay Element:

h_ccvs1_10 n_7 gnd ccvs delay v_vdc3_9 td=1.5e-008

Index

- . -

- C -

- D -

- E -

- P -

- R -

- S -

- T -