Spectre® Circuit Simulator Reference

Product Version 5.0
July 2002

Cadence Design Systems, Inc., 555 River Oaks Parkway, San Jose, CA 95134, USA

Contents

Preface

This manual assumes that you are familiar with the development, design, and simulation of integrated circuits and that you have some familiarity with SPICE simulation. It contains information about the Spectre® circuit simulator.

Spectre is an advanced circuit simulator that simulates analog and digital circuits at the differential equation level. The simulator uses improved algorithms that offer increased simulation speed and greatly improved convergence characteristics over SPICE. Besides the basic capabilities, the Spectre circuit simulator provides significant additional capabilities over SPICE. SpectreHDL (Spectre High-Level Description Language) and Verilog®-A use functional description text files (modules) to model the behavior of electrical circuits and other systems. SpectreRF adds several new analyses that support the efficient calculation of the operating point, transfer function, noise, and distortion of common RF and communication circuits, such as mixers, oscillators, sample holds, and switched-capacitor filters.

This preface discusses the following topics:

Related Documents

The following can give you more information about the Spectre circuit simulator and related products:

■ To learn more about the equations used in the Spectre circuit simulator, consult the *Spectre Circuit Simulator Device Model Equations* manual.

■ The Spectre circuit simulator is often run within the Cadence® analog circuit design environment, under the Cadence® design framework II. To see how the Spectre circuit simulator is run under the analog circuit design environment, read the *Cadence Analog Design Environment User Guide*.

■ For more information about using the Spectre circuit simulator with SpectreHDL, see the *SpectreHDL Reference* manual.

- For more information about using the Spectre circuit simulator with Verilog-A, see the *Verilog-A Language Reference* manual.
- If you want to see how SpectreRF is run under the analog circuit design environment, read *SpectreRF Help*.
- For more information about RF theory, see *SpectreRF Theory*.
- For more information about how you work with the design framework II interface, see *Design Framework II Help*.
- For more information about specific applications of Spectre analyses, see *The Designer's Guide to SPICE & Spectre*[1].

Typographic and Syntax Conventions

This list describes the syntax conventions used for the Spectre circuit simulator.

`literal` Nonitalic words indicate keywords that you must enter literally. These keywords represent command (function, routine) or option names, file names and paths, and any other sort of type-in commands.

`argument` Words in italics indicate user-defined arguments for which you must substitute a name or a value. (The characters before the underscore (_) in the word indicate the data types that this argument can take. Names are case sensitive.

| Vertical bars (OR-bars) separate possible choices for a single argument. They take precedence over any other character.

`[ ]` Brackets denote optional arguments. When used with OR-bars, they enclose a list of choices. You can choose one argument from the list.

`{ }` Braces are used with OR-bars and enclose a list of choices. You must choose one argument from the list.

`...` Three dots (...) indicate that you can repeat the previous argument. If you use them with brackets, you can specify zero or more arguments. If they are used without brackets, you must specify at least one argument, but you can specify more.

1. Kundert, Kenneth S. *The Designer's Guide to SPICE & Spectre.* Boston: Kluwer Academic Publishers, 1995.

Important

The language requires many characters not included in the preceding list. You must enter required characters exactly as shown.

References

Text within brackets ([ ]) are references. See Appendix A, “References,” for more detailed information.

1

Introducing the Spectre Circuit Simulator

This chapter discusses the following:

The Spectre® circuit simulator is a modern circuit simulator that uses direct methods to simulate analog and digital circuits at the differential equation level. The basic capabilities of the Spectre circuit simulator are similar in function and application to SPICE, but the Spectre circuit simulator is not descended from SPICE. The Spectre and SPICE simulators use the same basic algorithms—such as implicit integration methods, Newton-Raphson, and direct matrix solution—but every algorithm is newly implemented. Spectre algorithms, the best currently available, give you an improved simulator that is faster, more accurate, more reliable, and more flexible than previous SPICE-like simulators.

Improvements over SPICE

The Spectre circuit simulator has many improvements over SPICE.

Improved Capacity

The Spectre circuit simulator can simulate larger circuits than other simulators because its convergence algorithms are effective with large circuits, because it is fast, and because it is frugal with memory and uses dynamic memory allocation. For large circuits, the Spectre circuit simulator typically uses less than half as much memory as SPICE.

Improved Accuracy

Improved component models and core simulator algorithms make the Spectre circuit simulator more accurate than other simulators. These features improve Spectre accuracy:

- Advanced metal oxide semiconductor (MOS) and bipolar models
    - The Spectre BSIM 3v3 is a physics-based metal-oxide semiconductor field effect transistor (MOSFET) model for simulating analog circuits.
    - The Spectre models include the MOS0 model, which is even simpler and faster than MOS1 for simulating noncritical MOS transistors in logic circuits and behavioral models, MOS 9, EKV, BTA-HVMOS, BTA-SOI, VBIC95, TOM2, and HBT.
- Charge-conserving models

    The capacitance-based nonlinear MOS capacitor models used in many SPICE derivatives can create or destroy small amounts of charge on every time step. The Spectre circuit simulator avoids this problem because all Spectre models are charge-conserving.
- Improved Fourier analyzer

    The Spectre circuit simulator includes a two-channel Fourier analyzer that is similar in application to the SPICE `.FOURIER` statement but is more accurate. The Spectre simulator's Fourier analyzer has greater resolution for measuring small distortion products on a large sinusoidal signal. Resolution is normally greater than 120 dB. Furthermore, the Spectre simulator's Fourier analyzer is not subject to aliasing, a common error in Fourier analysis. As a result, the Spectre simulator can accurately compute the Fourier coefficients of highly discontinuous waveforms.
- Better control of numerical error

    Many algorithms in the Spectre circuit simulator are superior to their SPICE counterparts in avoiding known sources of numerical error. The Spectre circuit simulator improves the control of local truncation error in the transient analysis by controlling error in the voltage rather than the charge.

    In addition, the Spectre circuit simulator directly checks Kirchhoff's Current Law (also known as Kirchhoff's Flow Law) at each time step, improves the charge-conservation accuracy of the Spectre circuit simulator, and eliminates the possibility of false convergence.
- Superior time-step control algorithm

    The Spectre circuit simulator provides an adaptive time-step control algorithm that reliably follows rapid changes in the solution waveforms. It does so without limiting assumptions about the type of circuit or the magnitude of the signals.

- More accurate simulation techniques

  Techniques that reduce reliability or accuracy, such as device bypass, simplified models, or relaxation methods, are not used in the Spectre circuit simulator.

- User control of accuracy tolerances

  For some simulations, you might want to sacrifice some degree of accuracy to improve the simulation speed. For other simulations, you might accept a slower simulation to achieve greater accuracy. With the Spectre circuit simulator, you can make such adjustments easily by setting a single parameter.

Improved Speed

The Spectre circuit simulator is designed to improve simulation speed. The Spectre circuit simulator improves speed by increasing the efficiency of the simulator rather than by sacrificing accuracy.

- Faster simulation of small circuits

  The average Spectre simulation time for small circuits is typically two to three times faster than SPICE. The Spectre circuit simulator can be over 10 times faster than SPICE when SPICE is hampered by discontinuity in the models or problems in the code. Occasionally, the Spectre circuit simulator is slower when it finds ringing or oscillation that goes unnoticed by SPICE. This can be improved by setting the `macromodels` option to `yes`.

- Faster simulation for large circuits

  The Spectre circuit simulator is generally two to five times faster than SPICE with large circuits because it has fewer convergence difficulties and because it rapidly factors and solves large sparse matrices.

Improved Reliability

The Spectre circuit simulator offers you the following improvements in reliability:

- Improved convergence

  Spectre proprietary algorithms ensure convergence of the Newton-Raphson algorithm in the DC analysis. The Spectre circuit simulator virtually eliminates the convergence problems that earlier simulators had with transient simulation.

- Helpful error and warning messages

  The Spectre circuit simulator detects and notifies you of many conditions that are likely to be errors. For example, the Spectre circuit simulator warns of models used in forbidden operating regions, of incorrectly wired circuits, and of erroneous component parameter values. By identifying such common errors, the Spectre circuit simulator saves you the time required to find these errors with other simulators.

The Spectre circuit simulator lets you define soft parameter limits and sends you warnings if parameters exceed these limits.

- Thorough testing

  Automated tests, which include over 1,000 test circuits, are constantly run on all hardware platforms to ensure that the Spectre circuit simulator is consistently reliable and accurate.

- Benchmark suite

  There is an independent collection of SPICE netlists that are difficult to simulate. You can obtain these circuits from the Microelectronics Center of North Carolina (MCNC) if you have File Transfer Protocol (FTP) access on the Internet. You can also get information about the performance of several simulators with these circuits.

  The Spectre circuit simulator has successfully simulated all of these circuits. Sometimes the netlists required minor syntax corrections, such as inserting balance parentheses, but circuits were never altered, and options were never changed to affect convergence.

Improved Models

The Spectre circuit simulator has MOSFET Level 0–3, BSIM1, BSIM2, BSIM3, BSIM 3v3, EKV, MOS9, JFET, TOM2, GaAs MESFET, BJT, VBIC, HBT, diode, and many other models. It also includes the temperature effects, noise, and MOSFET intrinsic capacitance models.

The Spectre Compiled Model Interface (CMI) option lets you integrate new devices into the Spectre simulator using a very powerful, efficient, and flexible C language interface. This CMI option, the same one used by Spectre developers, lets you install proprietary models.

Spectre Usability Features and Customer Service

The following features and services help you use the Spectre circuit simulator easily and efficiently:

- You can use Spectre soft limits to catch errors created by typing mistakes.
- Spectre diagnosis mode, available as an options statement parameter, gives you information to help diagnose convergence problems.

- You can run the Spectre circuit simulator standalone or run it under the Cadence analog design environment. To see how the Spectre circuit simulator is run under the analog design environment, read the *Cadence Analog Design Environment User Guide*. You can also run the Spectre circuit simulator in the Composer-to-Spectre direct simulation environment. The environment provides a graphical user interface for running the simulation.
- The Spectre circuit simulator gives you an online help system. With this system, you can find information about any parameter associated with any Spectre component or analysis. You can also find articles on other topics that are important to use the Spectre circuit simulator effectively.
- The Spectre circuit simulator also includes a waveform display tool, Analog Waveform Display (AWD), to use to display simulation results. For more information about AWD, see the *Analog Waveform User Guide*.
- If you experience a stubborn convergence or accuracy problem, you can send the circuit to Customer Support to get help with the simulation. For current phone numbers and e-mail addresses, see the following web site: `http://sourcelink.cadence.com/supportcontacts.html`.

Analog HDLs

The Spectre circuit simulator works with two analog high-level description languages (AHDLs): SpectreHDL and Verilog®-A. These languages are part of the Spectre Verilog-A Simulation option. SpectreHDL is proprietary to Cadence and is provided for backward compatibility. The Verilog-A language is an open standard, which was based upon SpectreHDL. The Verilog-A language is preferred because it is upward compatible with Verilog-AMS, a powerful and industry-standard mixed-signal language.

Both languages use functional description text files (modules) to model the behavior of electrical circuits and other systems. Each programming language allows you to create your own models by simply writing down the equations. The AHDL lets you describe models in a simple and natural manner. This is a higher level modeling language than previous modeling languages, and you can use it without being concerned about the complexities of the simulator or the simulator algorithms. In addition, you can combine AHDL components with Spectre built-in primitives.

Both languages let designers of analog systems and integrated circuits create and use modules that encapsulate high-level behavioral descriptions of systems and components. The behavior of each module is described mathematically in terms of its terminals and external parameters applied to the module. Designers can use these behavioral descriptions in many disciplines (electrical, mechanical, optical, and so on).

Both languages borrow many constructs from Verilog and the C programming language. These features are combined with a minimum number of special constructs for behavioral simulation. These high-level constructs make it easier for designers to use a high-level description language for the first time.

RF Capabilities

SpectreRF adds several new analyses that support the efficient calculation of the operating point, transfer function, noise, and distortion of common analog and RF communication circuits, such as mixers, oscillators, sample and holds, and switched-capacitor filters.

SpectreRF adds four types of analyses to the Spectre simulator. The first is periodic steady-state (PSS) analysis, a large-signal analysis that directly computes the periodic steady-state response of a circuit. With PSS, simulation times are independent of the time constants of the circuit, so PSS can quickly compute the steady-state response of circuits with long time constants, such as high-Q filters and oscillators.

You can also embed a PSS analysis in a sweep loop (referred to as an SPSS analysis in the Cadence analog design environment), which allows you to easily determine harmonic levels as a function of input level or frequency, making it easy to measure compression points, intercept points, and voltage-controlled oscillator (VCO) linearity.

The second new type of analysis is the periodic small-signal analysis. After completing a PSS analysis, SpectreRF can predict the small-signal transfer functions and noise of frequency translation circuits, such as mixers or periodically driven circuits such as oscillators or switched-capacitor or switched-current filters. The periodic small-signal analyses–periodic AC (PAC) analysis, periodic transfer function (PXF) analysis, and periodic noise (Pnoise) analysis–are similar to Spectre's AC, XF, and Noise analyses, but the traditional small-signal analyses are limited to circuits with DC operating points. The periodic small-signal analyses can be applied to circuits with periodic operating points, such as the following:

- Mixers
- VCOs
- Switched-current filters
- Phase/frequency detectors
- Frequency multipliers
- Chopper-stabilized amplifiers
- Oscillators

- Switched-capacitor filters
- Sample and holds
- Frequency dividers
- Narrow-band active circuits

The third SpectreRF addition to Spectre functionality is periodic distortion (PDISTO) analysis. PDISTO analysis directly computes the steady-state response of a circuit driven with a large periodic signal, such as an LO (local oscillation) or a clock, and one or more tones with moderate level. With PDISTO, you can model periodic distortion and include harmonic effects. PDISTO computes both a large signal, the periodic steady-state response of the circuit, and also the distortion effects of a specified number of moderate signals, including the distortion effects of the number of harmonics that you choose. This is a common scenario when trying to predict the intermodulation distortion of a mixer, amplifier, or a narrow-band filter. In this analysis, the tones can be large enough to create significant distortion, but not so large as to cause the circuit to switch or clip. The frequencies of the tones need not be periodically related to each other or to the large signal LO or clock. Thus, you can make the tone frequencies very close to each other without penalty, which allows efficient computation of intermodulation distortion of even very narrow band circuits.

The fourth analysis that SpectreRF adds to the Spectre circuit simulator is the envelope-following analysis. This analysis computes the envelope response of a circuit. The simulator automatically determines the clock period by looking through all the sources with the specified name. Envelope-following analysis is most efficient for circuits where the modulation bandwidth is orders of magnitude lower than the clock frequency. This is typically the case, for example, in circuits where the clock is the only fast varying signal and other input signals have a spectrum whose frequency range is orders of magnitude lower than the clock frequency. For another example, the down conversion of two closely placed frequencies can also generate a slow-varying modulation envelope. The analysis generates two types of output files, a voltage versus time (td) file, and an amplitude/phase versus time (fd) file for each specified harmonic of the clock fundamental.

In summary, with periodic small-signal analyses, you apply a small signal at a frequency that might not be harmonically related (noncommensurate) to the periodic response of the undriven system, the clock. This small signal is assumed to be small enough so that the circuit is unaffected by its presence.

With PDISTO, you can apply one or two additional signals at frequencies not harmonically related to the large signal, and these signals can be large enough to drive the circuit to behave nonlinearly.

For complex nonlinear circuits, hand calculation of noise or transfer function is virtually impossible. Without SpectreRF, these circuits must be breadboarded to determine their performances. The SpectreRF simulator eliminates unnecessary breadboarding, saving time.

Mixed-Signal Simulation

You can use the Spectre circuit simulator coupled with the Verilog®-XL simulator in the Cadence analog design environment to simulate mixed analog and digital circuits efficiently. This mixed-signal simulation solution can easily handle complex designs with tens of thousands of transistors and tens of thousands of gates. The digital Verilog data can come from the digital designer as either an RTL block or gates out of synthesis.

Environments

The Spectre circuit simulator is fully integrated into the Cadence® design framework II for the Cadence analog design environment and also into the Cadence analog workbench design system. You can also use the Spectre circuit simulator by itself with several different output format options.

Assura♠ interactive verification, Dracula® distributed multi-CPU option, and Assura hierarchical physical verification produce a netlist that can be read into the Spectre circuit simulator. However, only interactive verification when used with the Cadence analog design environment automatically attaches the stimulus file. All other situations require a stimulus file as well as device models.

2

Spectre Command Options

This chapter lists the options you can use with the `spectre` command and gives a brief description of each. It also discusses the following topics:

- Default Values on page 24
- Default Parameter Values on page 24

The `spectre` command takes the following syntax at the command line:

```
spectre <options> <inputfile>
```

Note: The Spectre® circuit simulator reads default values for all the command line arguments marked with a dagger (†) from the UNIX environment variable `%S_DEFAULTS`. If you do not

`-help`	Lists command options and available components and analyses. You can use `-h` as an abbreviation of `-help`.
`-help` <*name*>	Gives a synopsis of the device or analysis *name*. If *name* is `all`, the synopses for all components and analyses are given. You can use `-h` as an abbreviation of `-help`.
`-helpsort` <*name*>	Gives a synopsis of the device or analysis *name* and sorts all the parameters by name. You can use `-hs` as an abbreviation of `-helpsort`.
`-helpfull` <*name*>	Gives a full synopsis of the component or analysis *name*, including parameter types and range limits. You can use `-hf` as an abbreviation of `-helpfull`.
`-helpsortfull` <*name*>	Gives a full synopsis of component or analysis *name*, including parameter types and range limits. Sorts all parameters by name. You can use `-hsf` as an abbreviation of `-helpsortfull`.

-param†	Does not read the file containing the suggested parameter range limits. You can use -p as an abbreviation of -param.
+param <*file*>†	Reads *file* for the suggested parameter range limits. You can use +p as an abbreviation of +param.
-log†	Does not copy all messages to a file. You can use -l as an abbreviation of -log.
+log <*file*>†	Copies all messages to *file*. You can use +l as an abbreviation of +log.
=log <*file*>†	Sends all messages to *file*. You can use =l as an abbreviation of =log.
-raw <*raw*>†	Puts results in a file or directory named *raw*. In *raw*, %C is replaced by a circuit name. You can use -r as an abbreviation of -raw.
-format <*fmt*>†	Produces raw data in the format *fmt*. You can use -f as an abbreviation of -format. Possible values for *fmt* are nutbin, nutascii, wsfbin, wsfascii, psfbin, psfascii, or awb.
+checkpoint†	Turns on the checkpoint capability. You can use +cp as an abbreviation of +checkpoint.
-checkpoint†	Turns off the checkpoint capability. You can use -cp as an abbreviation of -checkpoint.
-recover†	Does not restart the simulation, even if a checkpoint file exists. You can use -rec as an abbreviation of -recover.
+recover†	Restarts the simulation from a checkpoint file, if it exists. You can use +rec as an abbreviation of +recover.
-cols <*N*>†	Sets screen width in characters to *N*. You can use -c as an abbreviation of -cols. If not set, the Spectre simulator determines the screen width automatically.

`-env` *<env>*	Calls the Spectre simulator from the *env* simulation environment. Possible values for *env* are `artist2`, `artist4`, `awb`, `edge`, `opus`, or `solo`.
`-%`*X*	In quoted strings within the netlist, replaces `%`*X* with nothing where *X* is any uppercase or lowercase letter.
`+%`*X* *<string>*†	In quoted strings within the netlist, replaces `%`*X* with *string*, where *X* is an uppercase or owercase letter. You can modify the string by using the `:x` operators.
`+error`†	Prints error messages.
`-error`†	Does not print error messages.
`+warn`†	Prints warning messages.
`-warn`†	Does not print warning messages.
+note	Prints notices.
-note	Does not print notices.
`+info`†	Prints informational messages.
`-info`†	Does not print informational messages.
`+debug`†	Prints debugging messages.
`-debug`†	Does not print debugging messages.
`-slave` *<cmd>*	Starts the attached simulator using the command *cmd.*
`-slvhost` *<hostname>*	Runs the attached simulator on machine *hostname*. Defaults to local machine.
`-V`	Prints version information.
`-W`	Prints subversion information.
`-alias` *<name>*†	Gives *name* to the license manager as the name of the simulator.
`-E`†	Runs the C preprocessor on an input file. In SPICE mode, the first line in the file must be a comment.
`-D`<*x*>†	Defines string *x* and runs the C preprocessor.

`-D`<*x*=*y*>†	Defines string *x* to be *y* and runs the C preprocessor.
`-U`<*x*>†	Undefines string *x* and runs the C preprocessor.
`-I`<*dir*>†	Runs the C preprocessor and searches the directory *dir* for include files.
`-spp`†	Do not run the Spice netlist reader on the input file.
`+spp`†	Run the Spice netlist reader on the input file. Use `+spp -sppbin` on the command line option to read other spp binaries.
`-sppbin` *file*†	Specify the path to nondefault spp binary. Default provided.
`+sensdata` <*file*>	Sends the sensitivity analyses data to *file*.
`-interactive`	Run in the noninteractive mode, that is, process the input file and then return. You can use `-inter` as an abbreviation of `-interactive`.
`+interactive`	Run in the default interactive mode. You can use `+inter` as an abbreviation of `+interactive`.
`+interactive=`*type*	Run in the interactive mode of the type specified. You can use `+inter` as an abbreviation of `+interactive`. Possible values for *type* are `skill` or `mpsc`.
`+mpssession=`*sessionName*	The *sessionName* for an interactive session using multiprocess SKILL (MPS). This option is necessary for `+interactive=mpsc` and implies `+interactive=mpsc`.
`+mpshost=`*sessionHost*	The *sessionHost* for an interactive session using MPS.

specify an input file, the Spectr simulatorreads from standard input. When +/- pairs of `spectre` command options are available, the default is the first value given in the previous list. For further information about the percent code options, `+%` and `-%`, see Chapter 11, “Managing Files,” in the *Spectre Circuit Simulator User Guide*.

Note: To remain consistent with the C preprocessor, there is no space between the preprocessor flags (`D`, `U`, `I`) and their arguments. The C preprocessor is available on UNIX systems only and requires that the first line of the file (the SPICE title line) begin with a comment character (`*` or `//`).

Default Values

The Spectre simulator reads default values for all the command line arguments marked with a dagger (†) from the UNIX environment variable `%S_DEFAULTS`. The name of the simulator as called replaces `%S.` Typically, this name is `spectre`, and the Spectre simulator looks for `spectre_DEFAULTS`. However, the name can be different if you move the executable to a file with a different name or if you call the Spectre simulator through a symbolic or hard link with a different name. This feature lets you set different default values for each name you use to call the Spectre simulator.

If the variable `%S_DEFAULTS` does not exist, `SPECTRE_DEFAULTS` is used instead. The command line arguments always override any specifications from the `options` statement in the circuit file. The `options` statement specifications, in turn, override any specifications in the environment variable.

Default Parameter Values

Many Spectre parameters have default values, and sometimes you will need to know them so you can determine whether they are acceptable for your simulation. You can find the default values for component, analysis, and control statement parameters by consulting the documentation for the statement in Spectre online help (`spectre -h`). Values given for parameters in the online help are the default values.

The following examples show you some defaults for different types of parameters from the Spectre online help:

`nf=1.0`Forward emission coefficient

`etchc=etch m`Narrowing due to etching for capacitances

`homotopy=all`Method used when there is no convergence on initial attempt of DC analysis; possible values are `none`, `gmin`, `source`, `dptran`, `ptran`, or `all`

`rawfile="%C:r.raw"`
Output raw data filename

In this example, the default values for `nf`, `etchc`, `homotopy`, and `rawfile` are a real number (`1.0`), the value of a different parameter (`etch`), an enumerated type (`all`), and a character string with a percent code and a colon modifier that gives Spectre instructions for creating the output filename (`"%C:r.raw"`).

For more information about percent codes and colon modifiers, see "Description of Spectre Predefined Percent Codes," "Customizing Percent Codes," and "Creating Filenames from Parts of Input Filenames" in the *Spectre Circuit Simulator User Guide*.

3

Component Statements Part 1

This chapter discusses the following topics:

The rest of the component statements are in Chapter 4, “Component Statements Part 2.”

To examine the equations used for some of these components, consult the *Spectre Circuit Simulator Device Model Equations* manual.

Analog-to-Logic Converter (a2d)

Description

The analog-to-logic converter transfers analog waveforms to a logic simulator.

This device is not supported within altergroup.

To examine the equations used for this component, consult the *Spectre Circuit Simulator Device Model Equations* manual.

Sample Instance Statement

```
da99 (cmp_out 0) a2d dest=”99991” vl=0 vh=5 timex=200u
//  99991 is a digital net in the verilog netlist.
```

Instance Definition

```
Name  p  n a2d parameter=value ...
```

Instance Parameters

1 `dest`The foreign simulator name for the destination of the signal.

2 `nestlev=0`Number of nesting levels to ignore in the hierarchical name. This should be used skip over extra levels that do not exist in the co-simulator.

3 `vl=0` VVoltages below this will be logical 0.

4 `vh=5` VVoltages above this will be logical 1.

5 `timex=1` sTime signal can linger between `vl` and `vh` before the state becomes X.

B3SOI-PD Transistor (b3soipd)

Description

B3SOI is an SOI model developed by U.C. Berkeley based on bsim3v3. B3SOI devices require that you use a model statement. This is the B3SOI version-2.2 model

This device is not supported within altergroup.

Instance Definition

```
Name  d  g  s  e  [p]  [b]  [t] ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as` (m^2)Area of source diffusion.

4 `ad` (m^2)Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `nrb (m/m)`Number of body squares.

10 `m=1`Multiplicity factor (number of MOSFETs in parallel).

11 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, or `subth`.

12 `rth0` (Ω)Thermal resistance.

13 `cth0 (F)`Thermal capacitance.

14 `bjtoff=0`BJT off flag.

15 `nbc=0 m/m`Number of body contact isolation edge.

16 `nseg=1 m/m`Number of segments for channel width partitioning.

17 `pdbcp=0 m`Perimeter length for body contact parasitic at drain.

18 `psbcp=0 m`Perimeter length for body contact parasitic at source.

19 `agbcp=0 m`Gate to body overlap for body contact parasitic.

20 `aebcp=0 m`Gate to body overlap for body contact parasitic.

21 `vbsusr=0.0 V`Optional initial value of Vbs for transient.

22 `tnodeout=0`Temperature node flag associated with T node.

Model Definition

```
model modelName b3soipd parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

2 `version=2.2`Model version selector.

Threshold Voltage Parameters

3 `vtho (V)`Threshold voltage at zero body bias for long-channel devices. For enhancement-mode devices, vtho > 0 for n-channel and vth < 0 for p-channel. Default value is calculated from other model parameters.

4 `k1=0.5 √V` Body-effect coefficient.

5 `k1w1=0.0 m`First body effect width dependent parameter.

6 `k1w2=0.0 m`Second body effect width dependent parameter.

7 `k2=-0.0186`Charge-sharing parameter.

8 `k3=0`Narrow width coefficient.

9 `k3b=0 1/V`Narrow width coefficient.

10 `w0=2.5e-6 m`Narrow width coefficient.

11 `nlx=1.74e-7 m`Lateral nonuniform doping coefficient.

12 `gamma1 (√V )`Body-effect coefficient near the surface.

13 `gamma2 (√V )`Body-effect coefficient in the bulk.

14 `vbx (V)`Threshold voltage transition body voltage.

15 `vbm=-3 V`Maximum applied body voltage.

16 `dvt0=2.2`First coefficient of short-channel effects.

17 `dvt1=0.53`Second coefficient of short-channel effects.

18 `dvt2=-0.032 1/V`Body-bias coefficient of short-channel effects.

19 `dvt0w=0`First coefficient of narrow-width effects.

20 `dvt1w=5.3e6`Second coefficient of narrow-width effects.

21 `dvt2w=-0.032 1/V`Body-bias coefficient of narrow-width effects.

22 `a0=1`Nonuniform depletion width effect coefficient.

23 `b0=0 m`Bulk charge coefficient due to narrow width effect.

24 `b1=0 m`Bulk charge coefficient due to narrow width effect.

25 `a1=0`No-saturation coefficient.

26 `a2=1`No-saturation coefficient.

27 `ags=0` F/m^2 `V`Gate-bias dependence of abulk.

28 `keta=-0.6 1/V`Body-bias coefficient for non-uniform depletion width effect.

29 `ketas=0.0 V`Surface Potential adjustment for bulk charge effect.

Process Parameters

30 `nsub=6e16` cm^{-3}Substrate doping concentration.

31 nch=1.7e17 cm^{-3}Peak channel doping concentration.

32 ngate (cm^{-3})Poly-gate doping concentration.

33 xj=0.15e-6 mSource/drain junction depth.

34 lint=0 mLateral diffusion for one side.

35 wint=0 mWidth reduction for one side.

36 ll=0 mLength dependence of delta L.

37 lln=1Length exponent of delta L.

38 lw=0 mWidth dependence of delta L.

39 lwn=1Width exponent of delta L.

40 lwl=0 m^{2}Area dependence of delta L.

41 wl=0 mLength dependence of delta W.

42 wln=1Length exponent of delta W.

43 ww=0 mWidth dependence of delta W.

44 wwn=1Width exponent of delta W.

45 wwl=0 m^{2}Area dependence of delta W.

46 dwg=0 m/vGate-bias dependence of channel width.

47 dwb=0 m/$\sqrt{v}$ Body-bias dependence of channel width.

48 dwbc=0.0 mWidth offset for body contact isolation edge.

49 tox=1e-8 mGate oxide thickness.

50 tbox=3e-7 mBuried oxide thickness.

51 tsi=1e-7 mSilicon film thickness.

52 xt=1.55e-7 mDoping depth.

53 `rdsw=100 Ω µm`Width dependence of drain-source resistance.

54 `prwb=0 1/√v` Body-effect coefficient for Rds.

55 `prwg=0 1/V`Gate-effect coefficient for Rds.

56 `wr=1`Width offset for parasitic resistance.

57 `xl=0 m`Length variation due to masking and etching.

58 `xw=0 m`Width variation due to masking and etching.

59 `binunit=1`Bin parameter unit selector. 1 for microns and 2 for meters.

Mobility Parameters

60 `mobmod=1`Mobility model selector.

61 `u0=670` $cm^2/V\ s$Low-field surface mobility at `tnom`. Default is 250 for PMOS.

62 `vsat=8e4 m/s`Carrier saturation velocity at `tnom`.

63 `ua=2.25e-9 m/v`First-oder mobility reduction coefficient.

64 `ub=5.87e-19` m^2/v^2

Second-oder mobility reduction coefficient.

65 `uc=-4.65e-11` m/v^2

Body-bias dependence of mobility. Default is -0.046 and unit is 1/V for mobmod=3.

Output Resistance Parameters

66 `drout=0.56`DIBL effect on output resistance coefficient.

67 `pclm=1.3`Channel length modulation coefficient.

68 `pdiblc1=0.39`First coefficient of drain-induced barrier lowering.

69 `pdiblc2=8.6e-3`Second coefficient of drain-induced barrier lowering.

70 `pdiblcb=0 1/V`Body-effect coefficient for DIBL.

71 `pvag=0`Gate dependence of Early voltage.

72 `delta=0.01 V`Effective drain voltage smoothing parameter.

Subthreshold Parameters

73 `cdsc=2.4e-4 F/m`^{2}Source/drain and channel coupling capacitance.

74 `cdscb=0 F/m`^{2} `V`Body-bias dependence of `cdsc`.

75 `cdscd=0 F/m`^{2} `V`Drain-bias dependence of `cdsc`.

76 `nfactor=1`Subthreshold swing coefficient.

77 `cit=0 F`Interface trap parameter for subthreshold swing.

78 `voff=-0.08 V`Threshold voltage offset.

79 `dsub=drout`DIBL effect in subthreshold region.

80 `eta0=0.08`DIBL coefficient subthreshold region.

81 `etab=-0.07 1/V`Body-bias dependence of `et0`.

Substrate Current Parameters

82 `alpha0=0 m/v`Substrate current impact ionization coefficient.

83 `beta0=0 1/V`First Vds dependent parameter of impact ionization current.

84 `fbjtii=0.0`Fraction of bipolar current affecting the impact ionization.

85 `beta1=0`Second Vds dependent parameter of impact ionization current.

86 `beta2=0 V`Third Vds dependent parameter of impact ionization current.

87 `vdsatii0=0.9 V` Nominal drain saturation voltage at threshold for impact ionization current.

88 `tii=0`Temperature dependent parameter for impact ionization current.

89 `lii=0`Channel length dependent parameter at threshold for impact ionization current.

90 `esatii=1e7 V/m`Saturation channel electric field for impact ionization current.

91 `sii0=0.5 1/V`First Vgs dependent parameter for impact ionization current.

92 `sii1=0.1 1/V`Second Vgs dependent parameter for impact ionization current.

93 `sii2=0.0 1/V`Third Vgs dependent parameter for impact ionization current.

94 `siid=0 1/V`Vds dependent parameter of drain saturation voltage for impact ionization current.

Parasitic Resistance Parameters

95 `rbsh=0 Ω`Extrinsic body contact sheet resistance.

96 `rsh=0 Ω/sqr`Source/drain diffusion sheet resistance.

97 `rs=0 Ω`Source resistance.

98 `rd=0 Ω`Drain resistance.

99 `rbody=0 F`Body resistance.

100 `rsc=0 Ω`Source contact resistance.

101 `rdc=0 Ω`Drain contact resistance.

102 `rss=0 Ω m`Scalable source resistance.

103 `rdd=0 Ω m`Scalable drain resistance.

104 `hdif=0 m`Length of heavily doped diffusion.

105 `ldif=0 m`Lateral diffusion beyond the gate.

106 `minr=0.1 Ω`Minimum source/drain resistance.

Junction Diode Model Parameters

107 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`. Possible values are `no` or `yes`.

108 `imelt=`imaxA'`Explosion current.

Overlap Capacitance Parameters

109 `cgso (F/m)`Gate-source overlap capacitance.

110 `cgdo (F/m)`Gate-drain overlap capacitance.

111 `cgeo=0.0 F/m`Gate-substrate overlap capacitance.

112 `cgbo=2 Dwc Cox F/m`
Gate-bulk overlap capacitance. The default value is 0 if version=3.0.

113 `meto=0 m`Metal overlap in fringing field.

114 `cgsl=0 F/m`Gate-source overlap capacitance in LDD region.

115 `cgdl=0 F/m`Gate-drain overlap capacitance in LDD region.

116 `ckappa=0.6`Overlap capacitance fitting parameter.

Junction Capacitance Model Parameters

117 `cjswg=cjsw F/m`Zero-bias gate-side junction capacitance density.

118 `mjswg=0.5`Gate-side junction grading coefficient.

119 `pbswg=0.7 V`Gate-side junction built-in potential.

120 `tt=1e-12 s`Transit time.

121 `ndif=1`Power coefficient of channel length dependency for diffusion capacitance.

122 `ldif0=1`Power coefficient of channel length dependency for diffusion capacitance.

Charge Model Selection Parameters

123 `capmod=2`Intrinsic charge model.

124 `dwc=wint m`Delta W for capacitance model.

125 `delvt=0.0 V`Threshold voltage adjustment for C-V.

126 `fbody=1.0`Scaling factor for body charge.

127 `dlc=lint m`Delta L for capacitance model.

128 `dlcb=lint m`Length offset fitting parameter for body charge.

129 `dlbg=0.0 m`Length offset fitting parameter for backgate charge.

130 `clc=1e-8 m`Intrinsic capacitance fitting parameter.

131 `cle=0.0`Intrinsic capacitance fitting parameter.

132 `cf (F/m)`Fringe capacitance parameter.

133 `vfbcv=-1`Flat-band voltage for capmod=0.

134 `xpart=0`Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

Default Instance Parameters

135 `w=5e-6 m`Default channel width.

136 `l=5e-6 m`Default channel length.

137 `as=0` m^2Default area of source diffusion.

138 `ad=0` m^2Default area of drain diffusion.

139 `ps=0 m`Default perimeter of source diffusion.

140 `pd=0 m`Default perimeter of drain diffusion.

141 `nrd=0 m/m`Default number of squares of drain diffusion.

142 `nrs=0 m/m`Default number of squares of source diffusion.

143 `nrb=0 m/m`Default body squares.

Temperature Effects Parameters

144 tnom (C)Parameters measurement temperature. Default set by options.

145 tmax=500 CMaximum device temperature above ambient.

146 shmod=0Self-heating selector.

147 tlev=0DC temperature selector.

148 tlevc=0AC temperature selector.

149 eg=1.12452 VEnergy band gap.

150 gap1=7.02e-4 V/CBand gap temperature coefficient.

151 gap2=1108 CBand gap temperature offset.

152 kt1=-0.11 VTemperature coefficient for threshold voltage.

153 kt1l=0 v mTemperature coefficient for threshold voltage.

154 kt2=0.022Temperature coefficient for threshold voltage.

155 at=3.3e4 m/sTemperature coefficient for vsat.

156 tcjswg=0 1/KTemperature coefficient of Cjswg.

157 tpbswg=0 V/KTemperature coefficient of Pbswg.

158 ua1=4.31e-9 m/vTemperature coefficient for ua.

159 ub1=-7.61e-18 m^2/v^2
Temperature coefficient for ub.

160 uc1=-5.5e-11 m/v^2
Temperature coefficient for uc. Default is -0.056 for mobmod=3.

161 prt=0 ΩTemperature coefficient for Rds.

162 trs=0 1/CTemperature parameter for source resistance.

163 trd=0 1/CTemperature parameter for drain resistance.

164 `ute=-1.5`Mobility temperature exponent.

165 `dt1=0`First temperature coefficient for tau.

166 `dt2=0`Second temperature coefficient for tau.

167 `cth0=0 F`Self-heating thermal capacitance.

168 `rth0=0 Ω`Self-heating thermal resistance.

169 `ntrecf=0`Temperature coefficient of Ntrecf.

170 `ntrecr=0`Temperature coefficient of Ntrecr.

171 `xbjt=2`BJT current temperature exponent.

172 `xdif=2`Diffusion current temperature exponent.

173 `xrec=20`Recombination current temperature exponent.

174 `xtun=0`Tunneling current temperature exponent.

Noise Model Parameters

175 `noimod=1`Noise model selector.

176 `kf=0`Flicker (1/f) noise coefficient.

177 `af=1`Flicker (1/f) noise exponent.

178 `ef=1`Flicker (1/f) noise frequency exponent.

179 `noia=1e20`Oxide trap density coefficient. Default is 9.9e18 for pmos.

180 `noib=5e4`Oxide trap density coefficient. Default is 2.4e3 for pmos.

181 `noic=-1.4e-12`Oxide trap density coefficient. Default is 1.4e-8 for pmos.

182 `em=4.1e7 V/m`Maximum electric field.

Auto Model Selector Parameters

183 `wmax=1 m`Maximum channel width for which the model is valid.

184 `wmin=0 m`Minimum channel width for which the model is valid.

185 `lmax=1 m`Maximum channel length for which the model is valid.

186 `lmin=0 m`Minimum channel length for which the model is valid.

Operating Region Warning Control Parameters

187 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

188 `imax=1 A`Maximum allowable current.

189 `bvj=∞ V`Junction reverse breakdown voltage.

190 `vbox=1e9 tox V`Oxide breakdown voltage.

191 `warn=on`Parameter to turn warnings on and off.
Possible values are `off` or `on`.

SOI Specific Parameters

192 `vbsa=0 V`Vbs0t offset voltage.

193 `delp=0.02`Offset constant for limiting Vbseff to Phis.

194 `kb1=1`Backgate coupling coefficient at strong inversion.

195 `kb3=1`Backgate coupling coefficient at subthreshold.

196 `dvbd0=0 V`First coefficient of short-channel effect on Vbs0t.

197 `dvbd1=0`First coefficient of short-channel effect on Vbs0t.

198 `abp=1`Gate bias coefficient for Xcsat calculation.

199 `mxc=-0.9`A smoothing parameter for Xcsat calculation.

200 `agidl=0`GIDL constant.

201 `bgidl=0 V/m`GIDL exponential coefficient.

202 `ngidl=1.2 V`GIDL Vds enhancement coefficient.

203 `ntun=10`Reverse tunneling non-ideality factor.

204 `nrecf0=2.0`Recombination non-ideality factor at forward bias.

205 `nrecr0=10`Recombination non-ideality factor at reversed bias.

206 `vsdfb (F/m)`Source/Drain diffusion flatband voltage.

207 `vsdth`Source/Drain diffusion threshold voltage.

208 `csdmin (F)`Source/Drain diffusion bottom minimum capacitance.

209 `csdesw=0`Source/drain sidewall fringing constant.

210 `aii=0`First parameter for critical field.

211 `bii=0`Second parameter for critical field.

212 `cii=0`Gate dependence of critical field.

213 `dii=-1`Body dependence of critical field.

214 `ndiode=1`Diode non-ideality factor.

215 `asd=0.3`Source/Drain diffusion smoothing parameter.

216 `isbjt=1e-6 A`BJT saturation current.

217 `isdif=0 A`Diffusion saturation current.

218 `isrec=1e-5 A`Recombination saturation current.

219 `istun=0 A`Tunneling saturation current.

220 `ln=2e-6 m`Electron diffusion length.

221 `vrec0=0 V`Voltage dependent parameter for recombination current.

222 `vtun0=0 V`Voltage dependent parameter for tunneling current.

223 `nbjt=1`Power coefficient of channel length dependency for bipolar current.

224 `lbjt0=0.20e-6 m`Reference channel length for bipolar current.

225 `vabjt=10 V`Early voltage for bipolar current.

226 `aely=0 V`Channel length dependency of early voltage for bipolar current.

227 `ahli=0`High level injection parameter for bipolar current.

228 `kbjt1=0 m`Parasitic bipolor base width.

Gate Tunneling Parameters

229 `wth0=0.0 µm`Minimum width for thermal resistance calculation..

230 `rhalo=1.0e15 Ω/sqr`
Body halo sheet resistance.

231 `ntox=1.0`Body halo sheet resistance.

232 `toxref=2.5e-9 m`Target oxide thickness.

233 `ebg=1.2 V`Effective bandgap in gate current calculation.

234 `nevb=3.0`Valence-band electron non-ideality factor.

235 `alphagb1=0.35`First Vox dependent parameter for gate curent in inversion..

236 `betagb1=0.03`Second Vox dependent parameter for gate curent in inversion..

237 `vgb1=300`Third Vox dependent parameter for gate curent in inversion..

238 `alphagb2=0.43`First Vox dependent parameter for gate curent in accumulation..

239 `betagb2=0.05`Second Vox dependent parameter for gate curent in accumulation..

240 `necb=1.0`Conduction-band electron non-ideality factor.

241 `vgb2=17`Third Vox dependent parameter for gate curent in accumulation..

242 `toxqm=Tox m`Effective oxide thickness considering quantum effects..

243 `voxh=5.0 V` Limit of Vox in gate current calculation..

244 `deltavox=0.005 V` Smoothing parameter in the Vox smoothing function..

245 `igmod=0`Gate current model selector.

Cross-Term Dependent Parameters

246 `paramchk=1`Model parameter checking selector.

247 `noif=1`Floating body excess noise ideality factor.

The `jmelt` parameter is used to aid convergence and prevent numerical overflow. The junction characteristics of the FET are accurately modeled for current (density) up to `jmelt`. For current density above `jmelt`, the junction is modeled as a linear resistor and a warning is printed.

Auto Model Selection

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 <= inst_length < lmax  and   wmin <= inst_width  < wmax
```

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)`Effective channel width.

2 `leff (m)`Effective channel length.

3 `rtheff (Ω)`Effective thermal resistance.

4 `Ctheff (F)`Effective thermal capacitance.

5 `rseff (Ω)`Effective source resistance.

6 `rdeff (Ω)`Effective drain resistance.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are n or p.

2 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, or `subth`.

3 `reversed`Reverse mode indicator.
Possible values are no or `yes`.

4 `vgs (V)`Gate-source voltage.

5 `vds (V)`Drain-source voltage.

6 `vbs (V)`Bulk-source voltage.

7 `vbgs (V)`Back-Gate-source voltage.

8 `ids (A)`Resistive drain-to-source current.

9 `ic (A)`BJT collector current.

10 `isgidl (A)`Source GIDL current.

11 `idgidl (A)`Drain GIDL current.

12 `iii (A)`Impact ionization current.

13 `ibd (A)`Resistive bulk-to-drain junction current.

14 `igbt (A)`Gate-to-body tunneling current.

15 `ibs (A)`Resistive bulk-to-source junction current.

16 `vth (V)`Threshold voltage.

17 `vdsat (V)`Drain-source saturation voltage.

18 `gm (S)`Common-source transconductance.

19 `gds (S)`Common-source output conductance.

20 `gmb (S)`Body-transconductance.

21 `gmbg (S)`Back-gate-transconductance.

22 `ueff` ($cm^2/V\ s$)Effective mobility.

23 `betaeff` (A/V^2)Effective `beta`.

24 `qg (Coul)`Gate charge.

25 `qd (Coul)`Drain charge.

26 `qs (Coul)`Source charge.

27 `qb (Coul)`Body charge.

28 `qbg (Coul)`Back-Gate charge.

29 `cgg (F)`dQg_dVg.

30 `cgd (F)`dQg_dVd.

31 `cgs (F)`dQg_dVs.

32 `cgb (F)`dQg_dVbk .

33 `cdg (F)`dQd_dVg.

34 `cdd (F)`dQd_dVd.

35 `cds (F)`dQd_dVs.

36 `cdb (F)`dQd_dVb.

37 `csg (F)`dQs_dVg.

38 `csd (F)`dQs_dVd.

39 `css (F)`dQs_dVs.

40 `csb (F)`dQs_dVb.

41 `cbg (F)`dQb_dVg.

42 `cbd (F)`dQb_dVd.

43 `cbs (F)`dQb_dVs.

44 `cbb (F)`dQb_dVb.

45 `id (A)`Total resistive drain current.

46 `is (A)`Total resistive source current.

47 `ib (A)`Total resistive bulk current.

48 `pwr (W)`Power at op point.

49 `gmoverid (1/V)`Gm/Ids.

50 `tdev (C)`Temperature rise from ambient.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Bipolar Junction Transistor (bjt)

Description

The bipolar transistor model is adapted from the integral charge model of Gummel and Poon, and it includes several high bias-level effects. This model defaults to the simpler Ebers-Moll model if certain parameters are left unspecified. This model also includes a substrate junction that connects either to the collector or to the base to model vertical and lateral structures.

This model has the following enhancements over SPICE2G.6:

1. Two base resistance models are provided.
2. Nonlinear collector resistance is implemented.
3. The integral form of the Early voltage effect is available.
4. The substrate junction includes both the diode and the capacitor.

This device is supported within altergroups.

To examine the equations used for this component, consult the *Spectre Circuit Simulator Device Model Equations* manual.

Sample Instance Statement

```
q1 (vcc net3 minus) npn_mod region=fwd area=1 m=1
```

Sample Model Statement

```
model npn_mod bjt type=npn is=10e-13 bf=200 va=58.8 ikf=5.63e-3 rb=700 rbm=86
re=3.2 cje=0.352e-12 pe=0.76 me=0.34 tf=249e-12 cjc=0.34e-12 pc=0.55
```

Instance Definition

```
Name  c  b  e  [s] ModelName parameter=value ...
```

You do not have to specify the substrate terminal. If you do not specify it, the substrate is connected to ground.

Instance Parameters

1 `area=1`Transistor area factor.

2 m=1Multiplicity factor.

3 triseTemperature rise from ambient.

4 region=fwdEstimated operating region.
Possible values are off, fwd, rev, sat, or breakdown.

Model Definition

```
model modelName bjt parameter=value ...
```

Model Parameters

Structural parameters

1 type=npnTransistor type.
Possible values are npn or pnp.

2 struct=verticalTransistor structure. For pnp default=lateral.
Possible values are vertical or lateral.

Saturation current parameters

3 is=1e-16 ASaturation current (*area).

4 ise=0 AB-E leakage saturation current. Set to c2*is if not given. (*area).

5 isc=0 AB-C leakage saturation current. Set to c4*is if not given. (*area).

6 iss=0 ASubstrate leakage saturation current (*area).

7 c2=0Forward leakage saturation current coefficient.

8 c4=0Reverse leakage saturation current coefficient.

B-C leakage model parameters

9 cbo=0 AExtrapolated 0-volt B-C leakage current (*area).

10 gbo=0 SSlope of Icbo vs. Vbc above Vbo (*area).

11 `vbo=0 V`Slope of `Icbo` vs. `Vbc` at `Vbc=0`.

12 `tcbo=0 1/C`Temperature coefficient for cbo.

13 `tgbo=0 1/C`Temperature coefficient for gbo.

Emission coefficient parameters

14 `nf=1`Forward emission coefficient.

15 `nr=1`Reverse emission coefficient.

16 `ne=1.5`B-E leakage emission coefficient.

17 `nc=2`B-C leakage emission coefficient.

18 `ns=1`Substrate junction emission coefficient.

Current gain parameters

19 `bf=100 A/A`Forward current gain (`beta`).

20 `br=1 A/A`Reverse current gain (`beta`).

21 `ikf=`∞ AHigh current corner for forward `beta` (*area).

22 `ikr=`∞ AHigh current corner for reverse `beta` (*area).

Early voltage parameters

23 `vaf=`∞ VForward Early voltage.

24 `var=`∞ VReverse Early voltage.

25 `ke=0 1/V`B-E space-charge integral multiplier.

26 `kc=0 1/V`B-C space-charge integral multiplier.

Parasitic resistance parameters

27 `rb=0` ΩZero-bias base resistance (/area).

28 `rbm=rb` Ω Minimum base resistance for high currents (/area).

29 `irb=∞` A Current at base resistance midpoint (*area).

30 `rbmod=spice` Nonlinear Rb model.
Possible values are `spectre` or `spice`.

31 `rc=0` Ω Collector resistance (/area).

32 `rcv=0` Ω Variable collector resistance (/area).

33 `rcm=0` Ω Minimum collector resistance (/area).

34 `dope=1e15` cm^{-3} Collector background doping concentration.

35 `cex=1` Current crowding exponent.

36 `cco=1` A Current crowding normalization constant (*area).

37 `re=0` Ω Emitter resistance (/area).

38 `minr=0.1` Ω Minimum parasitic resistance.

Junction capacitance parameters

39 `cje=0` F B-E zero-bias junction capacitance (*area).

40 `vje=0.75` V B-E built-in junction potential.

41 `mje=1/3` B-E junction exponent.

42 `cjc=0` F B-C zero-bias junction capacitance (*area).

43 `vjc=0.75` V B-C built-in junction potential.

44 `mjc=1/3` B-C junction exponent.

45 `xcjc=1` Fraction of B-C capacitance tied to internal base node.

46 `xcjc2=1` Fraction of B-C capacitance tied to collector and fraction of B-C tied to internal node.

47 `cjs=0` F B-S zero-bias junction capacitance (*area).

48 `vjs=0.75 V`B-S built-in junction potential.

49 `mjs=0`B-S junction exponent.

50 `fc=0.5`Junction capacitor forward-bias threshold.

51 `cbcp=0 F`B-C parasitic capacitance.

52 `cbep=0 F`B-E parasitic capacitance.

53 `ccsp=0 F`C-S parasitic capacitance.

Transit time and excess phase parameters

54 `tf=0 s`Ideal forward transit time.

55 `td=0 s`Intrinsic base delay time.

56 `xtf=0`Coefficient for bias dependence of `tf`.

57 `vtf=`∞ `V`Voltage describing Vbc dependence of `tf`.

58 `itf=0 A`High current parameter for effect on `tf` (*area).

59 `tr=0 s`Ideal reverse transit time.

60 `ptf=0` $^\circ$Excess phase at freq = 1.0/(`tf`*2 pi) Hz.

Temperature effects parameters

61 `tnom (C)`Parameters measurement temperature. Default set by `options`.

62 `trise=0 C`Temperature rise from ambient.

63 `eg=1.11 V`Band-gap.

64 `xtb=0Beta` temperature exponent.

65 `xti=3`Temperature exponent for effect on `is`.

66 `trb1=0 1/C`Linear temperature coefficient for the base resistor.

67 trb2=0 C^{-2}Quadratic temperature coefficient for the base resistor.

68 trm1=0 1/CLinear temperature coefficient for the minimum base resistor.

69 trm2=0 C^{-2}Quadratic temperature coefficient for the minimum base resistor.

70 trc1=0 1/CLinear temperature coefficient for the collector resistor.

71 trc2=0 C^{-2}Quadratic temperature coefficient for the collector resistor.

72 tre1=0 1/CLinear temperature coefficient for the emitter resistor.

73 tre2=0 C^{-2}Quadratic temperature coefficient for the emitter resistor.

74 tlev=0DC temperature selector.

75 tlevc=0AC temperature selector.

76 gap1=7.02e-4 V/CBand-gap temperature coefficient.

77 gap2=1108 CBand-gap temperature offset.

78 tikf1=0 1/CLinear temperature coefficient for ikf.

79 tikf2=0 C^{-2}Quadratic temperature coefficient for ikf.

80 tikr1=0 1/CLinear temperature coefficient for ikr.

81 tikr2=0 C^{-2}Quadratic temperature coefficient for ikr.

82 tirb1=0 1/CLinear temperature coefficient for irb.

83 tirb2=0 C^{-2}Quadratic temperature coefficient for irb.

84 tis1=0 1/CLinear temperature coefficient for is.

85 tis2=0 C^{-2}Quadratic temperature coefficient for is.

86 tise1=0 1/CLinear temperature coefficient for ise.

87 tise2=0 C^{-2}Quadratic temperature coefficient for ise.

88 tisc1=0 1/CLinear temperature coefficient for isc.

89 `tisc2=0` `C`$^{-2}$Quadratic temperature coefficient for `isc`.

90 `tiss1=0 1/C`Linear temperature coefficient for `iss`.

91 `tiss2=0` `C`$^{-2}$Quadratic temperature coefficient for `iss`.

92 `tbf1=0 1/C`Linear temperature coefficient for `bf`.

93 `tbf2=0` `C`$^{-2}$Quadratic temperature coefficient for `bf`.

94 `tbr1=0 1/C`Linear temperature coefficient for `br`.

95 `tbr2=0` `C`$^{-2}$Quadratic temperature coefficient for `br`.

96 `tvaf1=0 1/C`Linear temperature coefficient for `vaf`.

97 `tvaf2=0` `C`$^{-2}$Quadratic temperature coefficient for `vaf`.

98 `tvar1=0 1/C`Linear temperature coefficient for `var`.

99 `tvar2=0` `C`$^{-2}$Quadratic temperature coefficient for `var`.

100 `titf1=0 1/C`Linear temperature coefficient for `itf`.

101 `titf2=0` `C`$^{-2}$Quadratic temperature coefficient for `itf`.

102 `ttf1=0 1/C`Linear temperature coefficient for `tf`.

103 `ttf2=0` `C`$^{-2}$Quadratic temperature coefficient for `tf`.

104 `ttr1=0 1/C`Linear temperature coefficient for `tr`.

105 `ttr2=0` `C`$^{-2}$Quadratic temperature coefficient for `tr`.

106 `tnf1=0 1/C`Linear temperature coefficient for `nf`.

107 `tnf2=0` `C`$^{-2}$Quadratic temperature coefficient for `nf`.

108 `tnr1=0 1/C`Linear temperature coefficient for `nr`.

109 `tnr2=0` `C`$^{-2}$Quadratic temperature coefficient for `nr`.

110 `tne1=0 1/C`Linear temperature coefficient for `ne`.

111 tne2=0 C^{-2}Quadratic temperature coefficient for ne.

112 tnc1=0 1/CLinear temperature coefficient for nc.

113 tnc2=0 C^{-2}Quadratic temperature coefficient for nc.

114 tns1=0 1/CLinear temperature coefficient for ns.

115 tns2=0 C^{-2}Quadratic temperature coefficient for ns.

116 tmje1=0 1/CLinear temperature coefficient for mje.

117 tmje2=0 C^{-2}Quadratic temperature coefficient for mje.

118 tmjc1=0 1/CLinear temperature coefficient for mjc.

119 tmjc2=0 C^{-2}Quadratic temperature coefficient for mjc.

120 tmjs1=0 1/CLinear temperature coefficient for mjs.

121 tmjs2=0 C^{-2}Quadratic temperature coefficient for mjs.

122 cte=0 1/CTemperature coefficient for cje.

123 ctc=0 1/CTemperature coefficient for cjc.

124 cts=0 1/CTemperature coefficient for cjs.

125 tvje=0 V/CTemperature coefficient for vje.

126 tvjc=0 V/CTemperature coefficient for vjc.

127 tvjs=0 V/CTemperature coefficient for vjs.

128 tvtf1=0 1/CLinear temperature coefficient for vtf.

129 tvtf2=0 C^{-2}Quadratic temperature coefficient for vtf.

130 txtf1=0 1/CLinear temperature coefficient for xtf.

131 txtf2=0 C^{-2}Quadratic temperature coefficient for xtf.

Junction diode model control parameters

132 `dskip=yes`Skip junction calculations if they are reverse-saturated.
Possible values are `no` or `yes`.

133 `imelt=imax` AJunction explosion current (*area).

Operating region warning control parameters

134 `bvbe=∞` VB-E breakdown voltage.

135 `bvbc=∞` VB-C breakdown voltage.

136 `bvce=∞` VC-E breakdown voltage.

137 `bvsub=∞` VSubstrate junction breakdown voltage.

138 `vbefwd=0.2` VB-E forward voltage.

139 `vbcfwd=0.2` VB-C forward voltage.

140 `vsubfwd=0.2` VSubstrate junction forward voltage.

141 `imax=1e3` AMaximum allowable base current (*area).

142 `imax1=imax` AMaximum allowable collector current (*area).

143 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `fwd`, `rev`, or `sat`.

Noise model parameters

144 `kf=0`Flicker (1/f) noise coefficient.

145 `af=1`Flicker (1/f) noise exponent.

146 `kb=0`Burst noise coefficient.

147 `bnoisefc=1`Burst noise cutoff frequency.

148 `rbnoi=rb` ΩEffective base noise resistance.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

Operating-Point Parameters

1 `type=npn`Transistor type.
Possible values are `npn` or `pnp`.

2 `struct=vertical`Transistor structure. For pnp default=lateral.
Possible values are `vertical` or `lateral`.

3 `region=fwd`Estimated operating region.
Possible values are `off`, `fwd`, `rev`, `sat`, or `breakdown`.

4 `vbe (V)`Base-emitter voltage.

5 `vbc (V)`Base-collector voltage.

6 `vce (V)`Collector-emitter voltage.

7 `vsub (V)`Substrate junction voltage.

8 `ic (A)`Resistive collector current.

9 `ib (A)`Resistive base current.

10 `isub (A)`Resistive substrate current.

11 `pwr (W)`Power dissipation.

12 `betadc (A/A)`Ratio of resistive collector current to resistive base current.

13 `betaac (A/A)`Small-signal common-emitter current gain.

14 `gm (S)`Common-emitter transconductance.

15 `rpi (Ω)`Common-emitter input resistance.

16 `ro (Ω)`Common-emitter output resistance.

17 `rb (Ω)`Parasitic base resistance.

18 `rc (Ω)`Parasitic collector resistance.

19 `cpi (F)`Common-emitter input capacitance.

20 `cmu (F)`Common-base output capacitance.

21 `cmux (F)`External common-base output capacitance.

22 `csub (F)`Substrate capacitance.

23 `ft (Hz)`Unity small-signal current-gain frequency.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Lateral PNP Transistor (bjt301)

Description

The bjt301 model provides an extensive description of a lateral integrated circuit junction-isolated PNP transistor. It is described in the Philips Bipolar Modelbook (Dec.93) as TPL level 301.

In extension to the modelbook description a minimum conductance `gmin` is inserted between the internal base and internal collector node, between the internal base and the internal emitter node, and between the external base and the substrate node to aid convergence. The value of `gmin` is set by an options statement, default = 1e-12 S.

The `imax` parameter is used to aid convergence and to prevent numerical overflow. The junction characteristics of the transistor are accurately modeled for currents up to `imax`. For currents above `imax`, the junction is modeled as a linear resistor and a warning is printed.

This device is supported within altergroups.

This device is dynamically loaded from the shared object /vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so

Sample Instance Statement

q2 (minus net3 vcc) pnp_mod region=fwd area=1 m=1

Sample Model Statement

model pnp_mod bjt301 type=pnp struct=lateral is=1e-14 bf=85 ilf=11e-9 ikf=95e-6 re=3.2 cje=0.352e-12

Instance Definition

Name `c b e` [s] ModelName parameter=value ...

Instance Parameters

1 `area=1`Area factor.

2 `mult=1`Alias of area factor.

3 `m=1`Multiplicity factor.

4 `region=fwd`Estimated DC operating region, used as a convergence aid. Possible values are `off`, `fwd`, `rev`, or `sat`.

Model Definition

model modelName bjt301 parameter=value ...

Model Parameters

Structural parameters

1 `type=pnp`Transistor type.
Possible values are `pnp` or `pnpl`.

2 `struct=lateral`Transistor structure.
Possible values are `lateral`.

Current parameters

3 `is=1.0e-15 A`Saturation current.

4 `imax=1.0 A`Explosion current.

5 `bf=100.0 A/A`Ideal forward common-emitter current gain (beta).

6 `ilf=10.0e-9 A`Low-level knee-current of forward beta.

7 `nlf=2.0`Emission coefficient of non-ideal forward base current.

8 `ikf=100.0e-6 A`High-injection knee-current of forward beta.

9 `nhf=1.0`Basewidening exponent.

10 `veaf=50.0 V`Early voltage related to collector junction.

11 `br=10.0 A/A`Ideal reverse common-collector current gain (beta).

12 `ilr=10.0e-9 A`Low-level knee-current of reverse beta.

13 `nlr=2.0`Emission coefficient of non-ideal reverse base current.

14 `ikr=100.0e-6 A`High-injection knee-current of reverse beta.

15 `iks=100.0e-6 A`High-injection current of substrate effect.

16 `xcs=1.0`Current fraction of c-b-s transistor.

17 `xes=0.01`Current fraction of e-b-s transistor.

Parasitic resistance parameters

18 `rc=1.0` ΩCollector resistance.

19 `rbc=10.0` ΩConstant part of base resistance.

20 `rbv=10.0` ΩVariable part of base resistance.

21 `re=1.0` ΩEmitter series resistance.

Junction capacitance parameters

22 `taub=25.0e-9` sForward transit time related to neutral base.

23 `taune=1.0e-9` sForward transit time related to neutral emitter in neutral e-b region.

24 `mtau=1.0`Coefficient of current dependence of taune.

25 `cje=100.0e-15` FZero bias emitter-base depletion capacitance.

26 `vde=0.55` VEmitter-base diffusion voltage.

27 `pe=0.333`Emitter-base grading coefficient.

28 `taur=100.0e-9` sIdeal reverse transit time.

29 `cjc=200.0e-15` FZero bias collector-base depletion capacitance.

30 `vdc=0.55` VCollector-base diffusion voltage.

31 `pc=0.333`Collector-base grading coefficient.

32 `cjs=1.0e-12` FZero bias substrate junction depletion capacitance.

33 `vds=0.55` VSubstrate junction diffusion voltage.

34 `ps=0.333`Substrate junction grading coefficient.

35 `exphi=0.3`Excess phase shift.

36 `fc=0.95`Coefficient for forward bias capacitance.

Temperature effects parameters

37 `tref (C)`Reference temperature. Default set by option `tnom`.

38 `tnom (C)`Alias of tref. Default set by option `tnom`.

39 `dta=0.0 K`Difference between device temperature and ambient temperature.

40 `trise=0.0 K`Alias of dta.

41 `ptbf=0.0`Power for temperature dependence of `bf`.

42 `ptbr=0.0`Power for temperature dependence of `br`.

43 `ptrc=0.0`Power for temperature dependence of `rc`.

44 `ptrb=0.0`Power for temperature dependence of `rbc` and `rbv`.

45 `vg=1.2 V`Band-gap voltage.

46 `pt=1.2`Power for temperature dependence of diffusion coefficient.

Noise model parameters

47 `kf=0.0`Flickernoise coefficient.

48 `af=1.0`Flickernoise exponent.

Output Parameters

1 `ist (A)`Saturation current.

2 `iole (A)`Non-ideal forward base saturation current.

3 `iolc (A)`Non-ideal reverse base saturation current.

4 `bft (A/A)`Ideal forward common-emitter current gain (beta).

5 `brt (A/A)`Ideal reverse common-collector current gain (beta).

6 `rct (Ω)`Collector resistance.

7 `rbct (Ω)`Constant part of base resistance.

8 `rbvt (Ω)`Variable part of base resistance.

9 `taubt (s)`Forward transit time related to neutral base.

10 `cjet (F)`Zero bias emitter-base depletion capacitance.

11 `vdet (V)`Emitter-base diffusion voltage.

12 `taurt (s)`Ideal reverse transit time.

13 `cjct (F)`Zero bias collector-base depletion capacitance.

14 `vdct (V)`Collector-base diffusion voltage.

15 `cjst (F)`Zero bias substrate junction depletion capacitance.

16 `vdst (V)`Substrate junction diffusion voltage.

Operating-Point Parameters

1 `ib (A)`Base current.

2 `ic (A)`Collector current.

3 `ie (A)`Emitter current.

4 `isub (A)`Substrate current.

5 `vbe (V)`Base-emitter voltage.

6 `vbc (V)`Base-collector voltage.

7 `vce (V)`Collector-emitter voltage.

8 `vsubj (V)`Substrate voltage.

9 `betadc (A/A)`Ratio of DC collector current to DC Base current.

10 `rb` (Ω) Base resistance at operating point.

11 `rc` (Ω) Collector resistance at operating point.

12 `re` (Ω) Emitter resistance at operating point.

13 `icb` (A) Collector-Base current.

14 `ieb` (A) Emitter-Base current.

15 `icsub` (A) Collector-Substrate current.

16 `iesub` (A) Emitter-Substrate current.

17 `pwr` (W) Power.

18 `gpi` (S) Conductance emitter-base junction.

19 `gmu` (S) Conductance collector-base junction.

20 `gf` (S) Forward transconductance.

21 `gr` (S) Reverse transconductance.

22 `gs` (S) Conductance substrate-base junction.

23 `g3` (S) Transconductance (parasitic PNP) c-b-s transistor.

24 `g4` (S) Transconductance (parasitic PNP) e-b-s transistor.

25 `ced` (F) Emitter diffusion capacitance.

26 `ccd` (F) Collector diffusion capacitance.

27 `cet` (F) Emitter junction depletion capacitance.

28 `cct` (F) Collector junction depletion capacitance.

29 `cst` (F) Substrate junction depletion capacitance.

30 `betaac` (A/A) Small-signal common-emitter current gain.

31 `ft` (Hz) Unity small-signal current-gain frequency.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

af M-48	gf OP-20	mtau M-24	struct M-2
area I-1	gmu OP-19	mult I-2	taub M-22
betaac OP-30	gpi OP-18	nhf M-9	taubt O-9
betadc OP-9	gr OP-21	nlf M-7	taune M-23
bf M-5	gs OP-22	nlr M-13	taur M-28
bft O-4	ib OP-1	pc M-31	taurt O-12
br M-11	ic OP-2	pe M-27	tnom M-38
brt O-5	icb OP-13	ps M-34	tref M-37
ccd OP-26	icsub OP-15	pt M-46	trise M-40
cct OP-28	ie OP-3	ptbf M-41	type M-1
ced OP-25	ieb OP-14	ptbr M-42	vbc OP-6
cet OP-27	iesub OP-16	ptrb M-44	vbe OP-5
cjc M-29	ikf M-8	ptrc M-43	vce OP-7
cjct O-13	ikr M-14	pwr OP-17	vdc M-30
cje M-25	iks M-15	rb OP-10	vdct O-14
cjet O-10	ilf M-6	rbc M-19	vde M-26
cjs M-32	ilr M-12	rbct O-7	vdet O-11
cjst O-15	imax M-4	rbv M-20	vds M-33
cst OP-29	iolc O-3	rbvt O-8	vdst O-16
dta M-39	iole O-2	rc M-18	veaf M-10
exphi M-35	is M-3	rc OP-11	vg M-45
fc M-36	ist O-1	rct O-6	vsubj OP-8
ft OP-31	isub OP-4	re OP-12	xcs M-16
g3 OP-23	kf M-47	re M-21	xes M-17
g4 OP-24	m I-3	region I-4	

Lateral PNP Transistor (bjt500)

Description

The `bjt500` model provides an extensive description of a lateral integrated circuit junction-isolated PNP transistor. It is described in the Philips Bipolar Modelbook (Dec.93) as TPL-level-500. Information on how to obtain this document can be found on Source Link by searching for Philips.

In extension to the modelbook description a minimum conductance `gmin` is inserted between the internal base and internal collector node, between the internal base and the internal emitter node, and between the external base and the substrate node to aid convergence. The value of `gmin` is set by an options statement, default is gmin = 1.0e-12 S .

The `imax` parameter is used to aid convergence and to prevent numerical overflow. The junction characteristics of the transistor are accurately modeled for currents up to `imax`. For currents above `imax`, the junction is modeled as a linear resistor, and a warning is printed.

This device is supported within altergroups.

This device is dynamically loaded from the shared object /vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so

Sample Instance Statement

q3 (minus net3 vcc) pnp_mod region=fwd area=1 m=1

Sample Model Statement

model pnp_mod bjt500 type=pnp struct=lateral is=1e-14 bf=85 ik=95e-6 reex=3.2 cje=0.352e-12

Instance Definition

Name `c b e` [s] ModelName parameter=value ...

Instance Parameters

1 `area=1`Area factor.

2 `mult=1`Alias of area factor.

3 `m=1`Multiplicity factor.

4 `region=fwd`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `fwd`, `rev`, or `sat`.

Model Definition

model modelName bjt500 parameter=value ...

Model Parameters

Structural parameters

1 `type=pnp`Transistor type.
Possible values are `pnp` or `pnpl`.

2 `struct=lateral`Transistor structure.
Possible values are `lateral`.

Current parameters

3 `is=1.8e-16 A`Collector-emitter saturation current.

4 `imax=1.0 A`Explosion current.

5 `bf=131.0 A/A`Ideal forward common-emitter current gain (beta).

6 `ibf=2.6e-14 A`Saturation current of non-ideal forward base current.

7 `vlf=0.54 V`Cross-over voltage of non-ideal forward base current.

8 `ik=1.1e-4 A`High injection knee current.

9 `xifv=0.43`Vertical fraction of forward current.

10 `eafl=20.5 V`Early voltage of the lateral forward current component.

11 `eafv=75.0 V`Early voltage of the vertical forward current component.

12 `br=25.0 A/A`Ideal reverse common-emitter current gain.

13 `ibr=1.2e-13 A`Saturation current of non-ideal reverse base current.

14 `vlr=0.48 V`Cross-over voltage of non-ideal reverse base current.

15 `xirv=0.43`Vertical fraction of reverse current.

16 `earl=13.1 V`Early voltage of the lateral reverse current component.

17 `earv=104.0 V`Early voltage of the vertical reverse current component.

18 `xes=2.7e-3`Ratio between saturation current of e-b-s transistor and e-b-c transistor.

19 `xhes=0.7`Fraction of substrate current of e-b-s transistor subject to high injection.

20 `xcs=3.0`Ratio between saturation current of c-b-s transistor and c-b-e transistor.

21 `xhcs=1.0`Fraction of substrate current of c-b-s transistor subject to high injection.

22 `iss=4.0e-13 A`Saturation current of substrate-base diode.

Parasitic resistance parameters

23 `rcex=5.0` ΩExternal part of the collector resistance.

24 `rcin=47.0` ΩInternal part of the collector resistance.

25 `rbcc=10.0` ΩConstant part of the base resistance `rbc`.

26 `rbcv=10.0` ΩVariable part of the base resistance `rbc`.

27 `rbec=10.0` ΩConstant part of the base resistance `rbe`.

28 `rbev=50.0` ΩVariable part of the base resistance `rbe`.

29 `reex=27.0` ΩExternal part of the emitter resistance.

30 `rein=66.0` ΩInternal part of the emitter resistance.

31 `rsb=1.0e15` ΩSubstrate-base leakage resistance.

Junction capacitance parameters

32 `tlat=2.4e-9 s`Low injection (forward and reverse) transit time of charge stored in the epilayer between emitter and collector.

33 `tfvr=3.0e-8 s`Low injection forward transit time due to charge stored in the epilayer under the emitter.

34 `tfn=2.0e-10 s`Low injection forward transit time due to charge stored in the emitter and the buried layer under the emitter.

35 `cje=6.1e-14 F`Zero-bias emitter-base depletion capacitance.

36 `vde=0.52 V`Emitter-base diffusion voltage.

37 `pe=0.3`Emitter-base grading coefficient.

38 `trvr=1.0e-9 s`Low injection reverse transit time due to charge stored in the epilayer under the collector.

39 `trn=3.0e-9 s`Low injection reverse transit time due to charge stored in the collector and the buried layer under the collector.

40 `cjc=3.9e-13 F`Zero-bias collector-base depletion capacitance.

41 `vdc=0.57 V`Collector-base diffusion voltage.

42 `pc=0.36`Collector-base grading coefficient.

43 `cjs=1.3e-12 F`Zero-bias substrate-base depletion capacitance.

44 `vds=0.52 V`Substrate-base diffusion voltage.

45 `ps=0.35`Substrate-base grading coefficient.

46 `exphi`Not used in model bjt500.

Temperature effects parameters

47 `tref (C)`Reference temperature. Default set by option `tnom`.

48 `tnom (C)`Alias of `tref`.

49 `tr (C)`Alias of `tref`.

50 `dta=0.0 K`Difference between the device temperature and the ambient analysis temperature.

51 `trise=0.0 K`Alias of `dta`.

52 `vgeb=1.206 V`Bandgap voltage of the emitter-base depletion region.

53 `vgcb=1.206 V`Bandgap voltage of the collector-base depletion region.

54 `vgsb=1.206 V`Bandgap voltage of the substrate-base depletion region.

55 `vgb=1.206 V`Bandgap voltage of the base between emitter and collector.

56 `vge=1.206 V`Bandgap voltage of the emitter.

57 `vgje=1.123 V`Bandgap voltage recombination emitter-base junction.

58 `ae=4.48`Temperature coefficient of `bf`.

59 `spb=2.853`Temperature coefficient of the epitaxial base hole mobility.

60 `snb=2.6`Temperature coefficient of the epitaxial base electron mobility.

61 `snbn=0.3`Temperature coefficient of buried layer electron mobility.

62 `spe=0.73`Temperature coefficient of emitter hole mobility.

63 `spc=0.73`Temperature coefficient of collector hole mobility.

64 `sx=1.0`Temperature coefficient of combined minority carrier mobility in emitter and buried layer.

Noise model parameters

65 `kf=0.0`Flickernoise coefficient.

66 `af=1.0`Flickernoise exponent.

Output Parameters

1 `ist (A)`Collector-emitter saturation current.

2 `bft (A/A)`Ideal forward common-emitter current gain (beta).

3 `ibft (A)`Saturation current of non-ideal forward base current.

4 `ikt (A)`High injection knee current.

5 `eaflt (V)`Early voltage of the lateral forward current component.

6 `eafvt (V)`Early voltage of the vertical forward current component.

7 `brt (A/A)`Ideal reverse common-emitter current gain.

8 `ibrt (A)`Saturation current of non-ideal reverse base current.

9 `earlt (V)`Early voltage of the lateral reverse current component.

10 `earvt (V)`Early voltage of the vertical reverse current component.

11 `isst (A)`Saturation current of substrate-base diode.

12 `rcint (Ω)`Internal part of the collector resistance.

13 `rbcct (Ω)`Constant part of the base resistance `rbc`.

14 `rbcvt (Ω)`Variable part of the base resistance `rbc`.

15 `rbect (Ω)`Constant part of the base resistance `rbe`.

16 `rbevt (Ω)`Variable part of the base resistance `rbe`.

17 `reint (Ω)`Internal part of the emitter resistance.

18 `tlatt (s)`Low injection (forward and reverse) transit time of charge stored in the epilayer between emitter and collector.

19 `tfvrt (s)`Low injection forward transit time due to charge stored in the epilayer under the emitter.

20 `tfnt (s)`Low injection forward transit time due to charge stored in the emitter and the buried layer under the emitter.

21 `cjet (F)`Zero-bias emitter-base depletion capacitance.

22 `vdet (V)`Emitter-base diffusion voltage.

23 `trvrt (s)`Low injection reverse transit time due to charge stored in the epilayer under the collector.

24 `trnt (s)`Low injection reverse transit time due to charge stored in the collector and the buried layer under the collector.

25 `cjct (F)`Zero-bias collector-base depletion capacitance.

26 `vdct (V)`Collector-base diffusion voltage.

27 `cjst (F)`Zero-bias substrate-base depletion capacitance.

28 `vdst (V)`Substrate-base diffusion voltage.

Operating-Point Parameters

1 `ic (A)`Resistive collector current.

2 `ib (A)`Resistive base current.

3 `ie (A)`Resistive emitter current.

4 `isub (A)`Resistive substrate current.

5 `iflat (A)`Lateral forward current.

6 `irlat (A)`Lateral reverse current.

7 `ifver (A)`Vertical forward current.

8 `irver (A)`Vertical reverse current.

9 `ire (A)`Ideal forward base current.

10 `ile (A)`Non-ideal forward base current.

11 `ise (A)`Forward substrate current.

12 `irc (A)`Ideal reverse base current.

13 `ilc (A)`Non-ideal reverse base current.

14 `isc (A)`Reverse substrate current.

15 `isf (A)`Reverse leakage current of the substrate-base junction.

16 `ip (A)`Main current.

17 `betadc (A/A)`Ratio of DC collector current to DC base current.

18 `vbc (V)`Base-collector voltage.

19 `vbe (V)`Base-emitter voltage.

20 `vce (V)`Collector-emitter voltage.

21 `vsb (V)`Substrate-base voltage.

22 `rcex` (Ω)External part of the collector resistance.

23 `rcint` (Ω)Internal part of the collector resistance.

24 `reex` (Ω)External part of the emitter resistance.

25 `reint` (Ω)Internal part of the emitter resistance.

26 `rbc` (Ω)Base resistance under the collector.

27 `rbe` (Ω)Base resistance under the emitter.

28 `rsb` (Ω)Ohmic leakage across the substrate-base junction.

29 `pwr (W)`Power.

30 `gfl (S)`Forward conductance, lateral path.

31 `grl (S)`Reverse conductance, lateral path.

32 `g11v (S)`Forward conductance, vertical path.

33 `g12v (S)`Collector Early-effect on Ifver.

34 `g21v (S)`Emitter Early-effect on Irver.

35 `g22v (S)`Reverse conductance, vertical path.

36 `gpiv (S)`Conductance emitter-base junction.

37 `gmuv (S)`Conductance collector-base junction.

38 `gbe (S)`Emitter-side: base conductance B1-B.

39 `gibe (S)`Emitter Early-effect on Ib1b.

40 `gbc (S)`Collector-side: base conductance B2-B.

41 `gibc (S)`Collector Early-effect on Ib2b.

42 `gise (S)`Transconductance (parasitic PNP) e-b-s transistor.

43 `gisc (S)`Transconductance (parasitic PNP) c-b-s transistor.

44 `gsb (S)`Conductance substrate-base junction.

45 `cpil (F)`Forward diffusion capacitance, lateral path.

46 `cipil (F)`Collector Early-effect on Qflat.

47 `cpiv (F)`Forward total capacitance, vertical path.

48 `cmul (F)`Reverse diffusion capacitance, lateral path.

49 `cimul (F)`Emitter Early-effect on Qrlat.

50 `cmuv (F)`Reverse total capacitance, vertical path.

51 `csb (F)`Total capacitance substrate-base junction.

52 `irbe (A)`Ideal total forward base current.

53 `irbc (A)`Ideal total reverse base current.

54 `irsb (A)`Substrate base leakage resistance current.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Vertical NPN/PNP Transistor (bjt503)

Description

The bjt503 model provides a detailed description of a vertical integrated NPN and PNP transistor. It is described in the Philips Bipolar Modelbook (Dec.95) as TN/TNS and TP/TPS level 503.

The NPN is also described in Nat.Lab. Unclassified Report Nr. 006/94 as Mextram Bipolar Transistor Model. Information on how to obtain this document can be found on Source Link by searching for Philips.

In addition to the model description a `level` parameter is added. Via the `level` parameter the user can switch between Philips Bipolar Modelbook (Dec.95) and Philips Bipolar Modelbook (Dec.94).

The `imax` parameter is used to aid convergence and to prevent numerical overflow. The junction characteristics of the transistor are accurately modeled for currents up to `imax`. For currents above `imax`, the junction is modeled as a linear resistor and a warning is printed.

The descriptions of the operating point derivatives are given for the NPN type. For the PNP type the terminal voltage in the descriptions has to be exchanged. E.g.:

NPN: gx = dIn/dVb2e1

PNP: gx = dIn/dVe1b2

This device is supported within altergroups.

This device is dynamically loaded from the shared object /vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so

Sample Instance Statement

q4 (vcc net3 minus) npn_mod region=fwd m=1 mult=1

Sample Model Statement

model npn_mod bjt503 type=npn level=2 exmod=1 is=1e-14 bf=85 ik=95e-6 rbc=50 cje=0.352e-12

Instance Definition

Name `c b e` [s] ModelName parameter=value ...

Instance Parameters

1 `area=1`Area factor.

2 `mult=1`Alias of area factor.

3 `m=1`Multiplication factor.

4 `region=fwd`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `fwd`, `rev`, or `sat`.

Model Definition

model modelName bjt503 parameter=value ...

Model Parameters

1 `type=npn`Transistor type.
Possible values are `npn`, `npnv`, `pnp`, or `pnpv`.

2 `level=2.0`Transistor Level. Possible values are `1` (Philips Bipolar Modelbook Dec.94) or `2` (Philips Bipolar Modelbook Dec.95).

3 `exmod=0`Flag for extended modeling of the reverse current gain.

4 `exphi=0`Flag for distributed high frequency effects.

5 `exavl=1`Flag for extended modeling of avalanche currents.

6 `is=5.0e-17 A`Collector-emitter saturation current.

7 `bf=140.0 A/A`Ideal forward current gain.

8 `xibi=0.0`Fraction of ideal base current that belongs to the sidewall.

9 `ibf=2.0e-14 A`Saturation current of the non-ideal forward base current.

10 `vlf=0.5 V`Cross-over voltage of the non-ideal forward base current.

11 `ik=15.0e-3 A`High-injection knee current.

12 `bri=16.0 A/A`Ideal reverse current gain.

13 `ibr=8.0e-15 A`Saturation current of the non-ideal reverse base current.

14 `vlr=0.5 V`Cross-over voltage of the non-ideal reverse base current.

15 `xext=0.5`Part of `Iex`, `Qex`, `Qtex` and `Isub` that depends on `Vbc1`.

16 `qbo=1.2e-12 Coul`Base charge at zero bias.

17 `eta=4.0`Factor of the built-in field of the base.

18 `avl=50.0`Weak avalanche parameter.

19 `efi=0.7`Electric field intercept (with `exavl=1`).

20 `ihc=3.0e-3 A`Critical current for hot carriers.

21 `rcc=25.0` ΩConstant part of the collector resistance.

22 `rcv=750.0` ΩResistance of the unmodulated epilayer.

23 `scrcv=1000.0` ΩSpace charge resistance of the epilayer.

24 `sfh=0.6`Current spreading factor epilayer.

25 `rbc=50.0` ΩConstant part of the base resistance.

26 `rbv=100.0` ΩVariable part of the base resistance at zero bias.

27 `re=2.0` ΩEmitter series resistance.

28 `taune=3.0e-10 s`Minimum delay time of neutral and emitter charge.

29 `mtau=1.18`Non-ideality factor of the neutral and emitter charge.

30 `cje=2.5e-13 F`Zero bias emitter-base depletion capacitance.

31 `vde=0.9 V`Emitter-base diffusion voltage.

32 `pe=0.33`Emitter-base grading coefficient.

33 `xcje=0.5`Fraction of the e-b depletion cap. that belongs to the sidewall.

34 `cjc=1.3e-13 F`Zero bias collector-base depletion capacitance.

35 `vdc=0.6 V`Collector-base diffusion voltage.

36 `pc=0.4`Collector-base grading coefficient variable part.

37 xp=0.2Constant part of cjc.

38 mc=0.5Collector current modulation coefficient.

39 xcjc=0.1Fraction of the collector-base depletion cap. under the emitter area.

40 tref (C)Reference temperature. Default set by option tnom.

41 tnom (C)Alias of tref. Default set by option tnom.

42 tr (C)Alias of tref. Default set by option tnom.

43 dta=0.0 KDifference of the device temperature to the ambient temperature.

44 trise=0.0 KAlias of dta.

45 vge=1.01 VBand-gap voltage of the emitter.

46 vgb=1.18 VBand-gap voltage of the base.

47 vgc=1.205 VBand-gap voltage of the collector.

48 vgj=1.1 VBand-gap voltage recombination emitter-base junction.

49 vi=0.04 VIonization voltage base dope.

50 na=3.0e17 cm^{-3}Maximum base dope concentration.

51 er=2.0e-3Temperature coefficient of vlf and vlr.

52 ab=1.35Temperature coefficient resistivity base.

53 aepi=2.15Temperature coefficient resistivity of the epilayer.

54 aex=1.0Temperature coefficient resistivity of the extrinsic base.

55 ac=0.4Temperature coefficient resistivity of the buried layer.

56 kf=2.0e-16Flickernoise coefficient ideal base current.

57 kfn=2.0e-16Flickernoise coefficient non-ideal base current.

58 af=1.0Flickernoise exponent.

59 `iss=6.0e-16 A`Base-substrate saturation current.

60 `iks=5.0e-6 A`Knee current of the substrate.

61 `cjs=1.0e-12 F`Zero bias collector-substrate depletion capacitance.

62 `vds=0.5 V`Collector-substrate diffusion voltage.

63 `ps=0.33`Collector-substrate grading coefficient.

64 `vgs=1.15 V`Band-gap voltage of the substrate.

65 `as=2.15`For a closed buried layer: `as=ac`. For an open buried layer: `as=aepi`.

66 `imax=1.0 A`Explosion current.

Output Parameters

1 `ist (A)`Collector-Emitter saturation current.

2 `bft (A/A)`Ideal forward current gain.

3 `ibft (A)`Saturation current of the non-ideal forward base current.

4 `vlft (V)`Cross-over voltage of the non-ideal forward base current.

5 `ikt (A)`High-injection knee current.

6 `ibrt (A)`Saturation current of the non-ideal reverse base current.

7 `vlrt (V)`Cross-over voltage of the non-ideal reverse base current.

8 `qbot (Coul)`Base charge at zero bias.

9 `avlt`Weak avalanche parameter.

10 `rcct (Ω)`Constant part of the collector resistance.

11 `rcvt (Ω)`Resistance of the unmodulated epilayer.

12 `rbct (Ω)`Constant part of the base resistance.

13 `rbvt (Ω)`Variable part of the base resistance at zero bias.

14 `taunet (s)`Minimum delay time of neutral and emitter charge.

15 `mtaut`Non-ideality factor of the neutral and emitter charge.

16 `cjet (F)`Zero bias emitter-base depletion capacitance.

17 `vdet (V)`Emitter-base diffusion voltage.

18 `cjct (F)`Zero bias collector-base depletion capacitance.

19 `vdct (V)`Collector-base diffusion voltage.

20 `xpt`Constant part of `cjc`.

21 `isst (A)`Base-substrate saturation current.

22 `ikst (A)`Knee current of the substrate.

23 `cjst (F)`Zero bias collector-substrate depletion capacitance.

24 `vdst (V)`Collector-substrate diffusion voltage.

Operating-Point Parameters

1 `ib (A)`Base current.

2 `ic (A)`Collector current.

3 `ie (A)`Emitter current.

4 `is (A)`Substrate current.

5 `vbe (V)`Base-emitter voltage.

6 `vbc (V)`Base-collector voltage.

7 `vce (V)`Collector-emitter voltage.

8 `vsc (V)`Substrate voltage.

9 `re` (Ω)Constant emitter resistance.

10 `rcc` (Ω)Constant collector resistance.

11 `rbc` (Ω)Constant part of base resistance.

12 `betadc` (A/A)DC current gain.

13 `pwr` (W)Power.

14 `Vb1e1` (V)Internal voltage.

15 `Vb2e1` (V)Internal voltage.

16 `Vb2c1` (V)Internal voltage.

17 `Vb2c2` (V)Internal voltage.

18 `Vb1b2` (V)Internal voltage.

19 `Vb1c1` (V)Internal voltage.

20 `Vbc1` (V)Internal voltage.

21 `in` (A)Main current.

22 `ic1c2` (A)Variable collector resistance current.

23 `ib1` (A)Bulk component of ideal base current.

24 `ib1s` (A)Sidewall component of ideal base current.

25 `ib2` (A)Non-ideal base current.

26 `iavl` (A)Weak avalanche current.

27 `ib1b2` (A)Variable base resistance current.

28 `ib3` (A)Non-ideal reverse base current.

29 `iex` (A)Internal extrinsic base current.

30 `isub` (A)Internal base-substrate current.

31 `isf` (A)Substrate-collector current.

32 `xiex` (A)External extrinsic base current.

33 `Xisub (A)`External base-substrate current.

34 `gx (S)`dIn/dVb2e1.

35 `gy (S)`dIn/dVb2c2.

36 `gz (S)`dIn/dVb2c1.

37 `grcvy (S)`dIc1c2/dVb2c2.

38 `grcvz (S)`dIc1c2/dVb2c1.

39 `gpi (S)`Conductance floor base-emitter junction: dIb1/dVb2e1 + dIb2/dVb2e1.

40 `sgpi (S)`Conductance sidewall base-emitter junction: dIb1S/dVb1e1.

41 `gmux (S)`Dependence avalanche multiplication on internal b-e junction: -dIavl/dVb2e1.

42 `gmu (S)`Dependence avalanche multiplication on internal b-c junction: -dIavl/dVb2c2.

43 `gmuz (S)`Dependence avalanche multiplication on external b-c junction:-dIavl/dVb2c1.

44 `grbv (S)`dIb1b2/dVb1b2.

45 `grbvx (S)`Emitter Early-effect on Ib1b2: dIb1b2/dVb2e1.

46 `grbvy (S)`Internal collector Early-effect on Ib1b2: dIb1b2/dVb2c2.

47 `grbvz (S)`External collector Early effect on Ib1b2: dIb1b2/dVb2c1.

48 `gmuex (S)`Conductance floor extrinsic b-c junction: dIex/dVb1c1 + dIsub/dVb1c1 + dIb3/dVb1c1.

49 `xgmuex (S)`Conductance sidewall extrinsic b-c junction: dXIex/dVbc1 + dXIsub/dVbc1.

50 `gsub (S)`Conductance s-c junction: dIsf/dVsc1.

51 `gpnp (S)`Transconductance floor extrinsic PNP transistor: dIsub/dVb1c1.

52 `xgpnp (S)`Transconductance sidewall extrinsic PNP transistor: dXIsub/dVbc1.

53 `cbex (F)`Capacitance floor b-e junction: dQte/dVb2e1 + dQbe/dVb2e1 + dQn/dVb2e1.

54 `cbey (F)` Internal collector Early-effect on Qbe: dQbe/dVb2c2.

55 `cbez (F)` External collector Early-effect on Qbe: dQbe/dVb2c1.

56 `scte (F)` Dependence of QteS on internal b-e junction: dQteS/dVb2e1.

57 `cbcx (F)` Emitter Early-effect on Qbc: dQbc/dVb2e1.

58 `cbcy (F)` Capacitance intrinsic b-c junction: dQtc/dVb2c2 + dQbc/dVb2c2 + dQepi/dVb2c2.

59 `cbcz (F)` Collector Early-effect on Qtc: dQtc/dVb2c1 + dQbc/dVb2c1 + dQepi/dVb2c1.

60 `cb1b2 (F)` Capacitance AC current crowding: dQb1b2/dVb1b2 = Cb.

61 `cb1b2x (F)` Dependence of Qb1b2 on internal b-e junction voltage: dQb1b2/dVb2e1.

62 `cbcex (F)` Capacitance floor extrinsic b-c junction: dQtex/dVb1c1 + dQex/dVb1c1.

63 `xcbcex (F)` Capacitance sidewall extrinsic b-c junction: dXQtex/dVbc1 + dXQex/dVbc1.

64 `cts (F)` Capacitance s-c junction: dQtex/dVb1c1 + dQex/dVb1c1.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Vertical NPN/PNP Transistor (bjt504)

Description

The bjt504 model provides a detailed description of a vertical integrated NPN and PNP transistor. It is described in the Philips Bipolar Modelbook (Dec.95) as TN/TNS and TP/TPS level 504.

The NPN is also described in Nat.Lab. Unclassified Report Nr. 006/94 as Mextram Bipolar Transistor Model. Information on how to obtain this document can be found on Source Link by searching for Philips.

In addition to the model description a `level` parameter is added. Via the `level` parameter the user can switch between Philips Bipolar Modelbook (Dec.95) and Philips Bipolar Modelbook (Dec.94).

The `imax` parameter is used to aid convergence and to prevent numerical overflow. The junction characteristics of the transistor are accurately modeled for currents up to `imax`. For currents above `imax`, the junction is modeled as a linear resistor and a warning is printed.

The descriptions of the operating point derivatives are given for the NPN type. For the PNP type the terminal voltage in the descriptions has to be exchanged. E.g.:

NPN: gx = dIn/dVb2e1

PNP: gx = dIn/dVe1b2

This device is supported within altergroups.

This device is dynamically loaded from the shared object /vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so

Instance Definition

Name C B E [S] ModelName parameter=value ...

Instance Parameters

1 `level`=XLEVEL.

2 `mult`=XMULT.

3 `tref`=XTREF.

4 `dta`=XDTA.

5 `exmod`=XEXMOD.

6 `exphi`=XEXPHI.

7 `exavl`=XEXAVL.

8 `is`=XIS.

9 `ik`=XIK.

10 ver=XVER.

11 vef=XVEF.

12 bf=XBF.

13 ibf=XIBF.

14 mlf=XMLF.

15 xibi=XXIBI.

16 bri=XBRI.

17 ibr=XIBR.

18 vlr=XVLR.

19 xext=XXEXT.

20 wavl=XWAVL.

21 vavl=XVAVL.

22 sfh=XSFH.

23 re=XRE.

24 rbc=XRBC.

25 rbv=XRBV.

26 rcc=XRCC.

27 rcv=XRCV.

28 scrcv=XSCRCV.

29 ihc=XIHC.

30 axi=XAXI.

31 cje=XCJE.

32 vde=XVDE.

33 pe=XPE.

34 xcje=XXCJE.

35 cbeo=XCBEO.

36 cjc=XCJC.

37 vdc=XVDC.

38 pc=XPC.

39 xp=XXP.

40 mc=XMC.

41 xcjc=XXCJC.

42 cbco=XCBCO.

43 mtau=XMTAU.

44 taue=XTAUE.

45 taub=XTAUB.

46 tepi=XTEPI.

47 taur=XTAUR.

48 deg=XDEG.

49 xrec=XXREC.

50 aqbo=XAQBO.

51 ae=XAE.

52 ab=XAB.

53 aepi=XAEPI.

54 aex=XAEX.

55 ac=XAC.

56 dvgbf=XDVGBF.

57 dvgbr=XDVGBR.

58 vgb=XVGB.

59 vgc=XVGC.

60 vgj=XVGJ.

61 dvgte=XDVGTE.

62 af=XAF.

63 kf=XKF.

64 kfn=XKFN.

65 iss=XISS.

66 iks=XIKS.

67 cjs=XCJS.

68 vds=XVDS.

69 ps=XPS.

70 vgs=XVGS.

71 as=XAS.

Model Definition

model modelName bjt504 parameter=value ...

Model Parameters

1 `type=npn`Transistor type.
Possible values are `npn` or `pnp`.

2 `level=`NULLLEVEL.

3 `mult=`NULLMULT.

4 `tref=`NULLTREF.

5 `dta=`NULLDTA.

6 `exmod=`NULLEXMOD.

7 `exphi=`NULLEXPHI.

8 `exavl=`NULLEXAVL.

9 `is=`NULLIS.

10 `ik=`NULLIK.

11 `ver=`NULLVER.

12 `vef=`NULLVEF.

13 `bf=`NULLBF.

14 `ibf=`NULLIBF.

15 `mlf=`NULLMLF.

16 `xibi=`NULLXIBI.

17 `bri=`NULLBRI.

18 `ibr=`NULLIBR.

19 `vlr=`NULLVLR.

20 `xext=`NULLXEXT.

21 `wavl`=NULLWAVL.

22 `vavl`=NULLVAVL.

23 `sfh`=NULLSFH.

24 `re`=NULLRE.

25 `rbc`=NULLRBC.

26 `rbv`=NULLRBV.

27 `rcc`=NULLRCC.

28 `rcv`=NULLRCV.

29 `scrcv`=NULLSCRCV.

30 `ihc`=NULLIHC.

31 `axi`=NULLAXI.

32 `cje`=NULLCJE.

33 `vde`=NULLVDE.

34 `pe`=NULLPE.

35 `xcje`=NULLXCJE.

36 `cbeo`=NULLCBEO.

37 `cjc`=NULLCJC.

38 `vdc`=NULLVDC.

39 `pc`=NULLPC.

40 `xp`=NULLXP.

41 `mc`=NULLMC.

42 `xcjc`=NULLXCJC.

43 `cbco`=NULLCBCO.

44 `mtau`=NULLMTAU.

45 `taue`=NULLTAUE.

46 `taub`=NULLTAUB.

47 `tepi`=NULLTEPI.

48 `taur`=NULLTAUR.

49 `deg`=NULLDEG.

50 `xrec`=NULLXREC.

51 `aqbo`=NULLAQBO.

52 `ae`=NULLAE.

53 `ab`=NULLAB.

54 `aepi`=NULLAEPI.

55 `aex`=NULLAEX.

56 `ac`=NULLAC.

57 `dvgbf`=NULLDVGBF.

58 `dvgbr`=NULLDVGBR.

59 `vgb`=NULLVGB.

60 `vgc`=NULLVGC.

61 `vgj`=NULLVGJ.

62 `dvgte`=NULLDVGTE.

63 `af`=NULLAF.

64 `kf`=NULLKF.

65 `kfn=NULLKFN`.

66 `iss=NULLISS`.

67 `iks=NULLIKS`.

68 `cjs=NULLCJS`.

69 `vds=NULLVDS`.

70 `ps=NULLPS`.

71 `vgs=NULLVGS`.

72 `as=NULLAS`.

Operating-Point Parameters

1 `level=XLEVEL`.

2 `mult=XMULT`.

3 `tref=XTREF`.

4 `dta=XDTA`.

5 `exmod=XEXMOD`.

6 `exphi=XEXPHI`.

7 `exavl=XEXAVL`.

8 `is=XIS`.

9 `ik=XIK`.

10 `ver=XVER`.

11 `vef=XVEF`.

12 `bf=XBF`.

13 `ibf=XIBF`.

14 `mlf`=XMLF.

15 `xibi`=XXIBI.

16 `bri`=XBRI.

17 `ibr`=XIBR.

18 `vlr`=XVLR.

19 `xext`=XXEXT.

20 `wavl`=XWAVL.

21 `vavl`=XVAVL.

22 `sfh`=XSFH.

23 `re`=XRE.

24 `rbc`=XRBC.

25 `rbv`=XRBV.

26 `rcc`=XRCC.

27 `rcv`=XRCV.

28 `scrcv`=XSCRCV.

29 `ihc`=XIHC.

30 `axi`=XAXI.

31 `cje`=XCJE.

32 `vde`=XVDE.

33 `pe`=XPE.

34 `xcje`=XXCJE.

35 `cbeo`=XCBEO.

36 `cjc`=XCJC.

37 `vdc`=XVDC.

38 `pc`=XPC.

39 `xp`=XXP.

40 `mc`=XMC.

41 `xcjc`=XXCJC.

42 `cbco`=XCBCO.

43 `mtau`=XMTAU.

44 `taue`=XTAUE.

45 `taub`=XTAUB.

46 `tepi`=XTEPI.

47 `taur`=XTAUR.

48 `deg`=XDEG.

49 `xrec`=XXREC.

50 `aqbo`=XAQBO.

51 `ae`=XAE.

52 `ab`=XAB.

53 `aepi`=XAEPI.

54 `aex`=XAEX.

55 `ac`=XAC.

56 `dvgbf`=XDVGBF.

57 `dvgbr`=XDVGBR.

58 `vgb=XVGB`.

59 `vgc=XVGC`.

60 `vgj=XVGJ`.

61 `dvgte=XDVGTE`.

62 `af=XAF`.

63 `kf=XKF`.

64 `kfn=XKFN`.

65 `iss=XISS`.

66 `iks=XIKS`.

67 `cjs=XCJS`.

68 `vds=XVDS`.

69 `ps=XPS`.

70 `vgs=XVGS`.

71 `as=XAS`.

72 `Vb2e1Vb2e1`.

73 `Vb2c2Vb2c2`.

74 `Vb2c1Vb2c1`.

75 `Vb1c1Vb1c1`.

76 `Ve1eVe1e`.

77 `InIn`.

78 `Ic1c2Ic1c2`.

79 `Ib1b2Ib1b2`.

80 `Ib1`Ib1.

81 `SIb1`SIb1.

82 `Ib2`Ib2.

83 `Ib3`Ib3.

84 `Iex`Iex.

85 `XIex`XIex.

86 `Iavl`Iavl.

87 `IRE`IRE.

88 `IRBC`IRBC.

89 `IRCC`IRCC.

90 `Qe`Qe.

91 `Qte`Qte.

92 `SQte`SQte.

93 `Qbe`Qbe.

94 `Qbc`Qbc.

95 `Qtc`Qtc.

96 `Qepi`Qepi.

97 `Qb1b2`Qb1b2.

98 `Qtex`Qtex.

99 `XQtex`XQtex.

100 `Qex`Qex.

101 `XQex`XQex.

102 `gx`gx.

103 `gy`gy.

104 `gz`gz.

105 `Sgpi`Sgpi.

106 `gpix`gpix.

107 `gpiy`gpiy.

108 `gpiz`gpiz.

109 `gmux`gmux.

110 `gmuy`gmuy.

111 `gmuz`gmuz.

112 `gmuex`gmuex.

113 `Xgmuex`Xgmuex.

114 `grcvy`grcvy.

115 `grcvz`grcvz.

116 `rbv`rbv.

117 `grbvx`grbvx.

118 `grbvy`grbvy.

119 `grbvz`grbvz.

120 `RE`RE.

121 `RBC`RBC.

122 `RCC`RCC.

123 `SCbe`SCbe.

124 CbexCbex.

125 CbeyCbey.

126 CbezCbez.

127 CbcxCbcx.

128 CbcyCbcy.

129 CbczCbcz.

130 CbcexCbcex.

131 XCbcexXCbcex.

132 Cb1b2Cb1b2.

133 Cb1b2xCb1b2x.

134 Cb1b2yCb1b2y.

135 Cb1b2zCb1b2z.

136 gmgm.

137 betabeta.

138 goutgout.

139 gmugmu.

140 RBRB.

141 CbeCbe.

142 CbcCbc.

143 fTfT.

144 IqsIqs.

145 XiWepiXiWepi.

146 Vb2c2starVb2c2star.

147 PdissPdiss.

148 TKTK.

149 IsubIsub.

150 XIsubXIsub.

151 IsfIsf.

152 QtsQts.

153 gSgS.

154 XgSXgS.

155 gSfgSf.

156 CtsCts.

Parameter Index

In the following index, I refers to instance parameters, M refers to the model parameters section, O refers to the output parameters section, and OP refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of M-35 means the 35th model parameter.

BSIM1 Field Effect Transistor (bsim1)

Description

BSIM1 is a semiempirical MOSFET model developed at the University of California, Berkeley. All the model parameters are extracted directly from physical devices. Three charge models are available. In SPICE mode, you can refer to BSIM1 as MOS level 4 or BSIM level 1. BSIM1 transistors require that you use a model statement.

This device is supported within altergroups.

To examine the equations used for this component, consult the *Spectre Circuit Simulator Device Model Equations* manual.

Sample Instance Statement

```
m1 (1 2 0 0) nchmod l=5u w=10u as=40u ad=40u pd=28u ps=28u m=1
```

Sample Model Statement

```
model nchmod bsim1 vfb0=-0.5 lvfb=0.5 wvfb=0.3 phi0=0.8 eta0=0.056 k1=0.5 muz=454
eg=0.99 gap1=5.5e-04 trs=1e-3 trd=1e-3 xpart=0.5 rs=10 rd=10
```

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as (m`^{2}`)`Area of source diffusion.

4 `ad (m`^{2}`)`Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `ld (m)`Drain diffusion length.

10 `ls (m)`Source diffusion length.

11 `m=1`Multiplicity factor (number of MOSFETs in parallel).

12 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

13 `trise (C)`Temperature rise from ambient.

14 `degradation=no`Hot-electron degradation flag.
Possible values are no or `yes`.

Model Definition

```
model modelName bsim1 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

Threshold Voltage parameters

2 vfb0=-0.8 VFlat-band voltage.

3 lvfb=0 V µmLength dependence of vfb.

4 wvfb=0 V µmWidth dependence of vfb.

5 pvfb=0 V µmWidth-length dependence of vfb.

6 phi0=0.75 VSurface potential.

7 lphi=0 V µmLength dependence of phi.

8 wphi=0 V µmWidth dependence of phi.

9 pphi=0 V µmWidth-length dependence of phi.

10 k1=0.7 √V Body-effect coefficient.

11 lk1=0 √V µmLength dependence of k1.

12 wk1=0 √V µmWidth dependence of k1.

13 pk1=0 √V µmWidth-length dependence of k1.

14 k2=0Charge-sharing parameter.

15 lk2=0 µmLength dependence of k2.

16 wk2=0 µmWidth dependence of k2.

17 pk2=0 µmWidth-length dependence of k2.

18 eta0=0Drain-induced barrier-lowering coefficient.

19 leta=0 µmLength dependence of eta.

20 weta=0 µmWidth dependence of eta.

21 peta=0 µmWidth-length dependence of eta.

22 x2e=0 1/VBody-bias dependence of eta.

23 `lx2e=0 µm/V`Length dependence of `x2e`.

24 `wx2e=0 µm/V`Width dependence of `x2e`.

25 `px2e=0 µm/V`Width-length dependence of `x2e`.

26 `x3e=0 1/V`Drain-bias dependence of `eta`.

27 `lx3e=0 µm/V`Length dependence of `x3e`.

28 `wx3e=0 µm/V`Width dependence of `x3e`.

29 `px3e=0 µm/V`Width-length dependence of `x3e`.

Mobility Parameters

30 `muz=400` $cm^2/V\ s$Low-field mobility.

31 `lmuz=0` $cm^2/V\ s$Length dependence of muz.

32 `wmuz=0` $cm^2/V\ s$Width dependence of muz.

33 `pmuz=0` $cm^2/V\ s$Width-length dependence of muz.

34 `x2mz=0` $cm^2/V^2\ s$Body-bias dependence of `muz`.

35 `lx2mz=0` $cm^2\ \mu m/V^2\ s$
Length dependence of `x2mz`.

36 `wx2mz=0` $cm^2\ \mu m/V^2\ s$
Width dependence of `x2mz`.

37 `px2mz=0` $cm^2\ \mu m/V^2\ s$
Width-length dependence of `x2mz`.

38 `mus=450` $cm^2/V\ s$Mobility in the saturation region.

39 `lmus=0` $cm^2\ \mu m/V\ s$
Length dependence of `mus`.

40 `wmus=0` $cm^2\ \mu m/V\ s$
Width dependence of `mus`.

41 `pmus=0` cm^2 µm/V s
Width-length dependence of `mus`.

42 `x2ms=0` cm^2/V^2 s Body-bias dependence of `mus`.

43 `lx2ms=0` cm^2 µm/V^2 s
Length dependence of `x2ms`.

44 `wx2ms=0` cm^2 µm/V^2 s
Width dependence of `x2ms`.

45 `px2ms=0` cm^2 µm/V^2 s
Width-length dependence of `x2ms`.

46 `x3ms=0` cm^2/V^2 s Drain-bias dependence of `mus`.

47 `lx3ms=0` cm^2 µm/V^2 s
Length dependence of `x3ms`.

48 `wx3ms=0` cm^2 µm/V^2 s
Width dependence of `x3ms`.

49 `px3ms=0` cm^2 µm/V^2 s
Width-length dependence of `x3ms`.

Mobility Modulation Parameters

50 `u00=0` 1/V Gate voltage dependence of mobility.

51 `lu0=0` µm/V Length dependence of `u0`.

52 `wu0=0` µm/V Width dependence of `u0`.

53 `pu0=0` µm/V Width-length dependence of `u0`.

54 `x2u0=0` 1/V^2 Body-bias dependence of `u0`.

55 `lx2u0=0` µm/V^2 Length dependence of `x2u0`.

56 `wx2u0=0` µm/V^2 Width dependence of `x2u0`.

57 `px2u0=0` µm/V^2 Width-length dependence of `x2u0`.

Velocity Saturation Parameters

58 u10=0 1/VVelocity saturation coefficient.

59 lu1=0 µm/VLength dependence of u1.

60 wu1=0 µm/VWidth dependence of u1.

61 pu1=0 µm/VWidth-length dependence of u1.

62 x2u1=0 $1/V^2$Body-bias dependence of u1.

63 lx2u1=0 $\mu m/V^2$Length dependence of x2u1.

64 wx2u1=0 $\mu m/V^2$Width dependence of x2u1.

65 px2u1=0 $\mu m/V^2$Width-length dependence of x2u1.

66 x3u1=0 $1/V^2$Drain-bias dependence of u1.

67 lx3u1=0 $\mu m/V^2$Length dependence of x3u1.

68 wx3u1=0 $\mu m/V^2$Width dependence of x3u1.

69 px3u1=0 $\mu m/V^2$Width-length dependence of x3u1.

Subthreshold Parameters

70 n0=0Subthreshold swing parameter.

71 ln0=0 µmLength dependence of subthreshold swing parameter.

72 wn0=0 µmWidth dependence of subthreshold swing parameter.

73 pn0=0 µmWidth-length dependence of subthreshold swing parameter.

74 nb=0 $\sqrt{V}$ Body-bias dependence of n0.

75 lnb=0 $\sqrt{V}$ µmLength dependence of nb.

76 wnb=0 $\sqrt{V}$ µmWidth dependence of nb.

77 pnb=0 $\sqrt{V}$ µmWidth-length dependence of nb.

78 nd=0 1/VDrain-bias dependence of n0.

79 lnd=0 µm/VLength dependence of nd.

80 wnd=0 µm/VWidth dependence of nd.

81 pnd=0 µm/VWidth-length dependence of nd.

82 subthmod=2Subthreshold model selector.

Impact Ionization Parameters

83 ai0=0 1/VHot-electron effect on Rout parameter.

84 lai0=0 µm/VLength dependence of ai0.

85 wai0=0 µm/VWidth dependence of ai0.

86 pai0=0 µm/VWidth-length dependence of ai0.

87 bi0=0 VHot-electron effect on Rout exponent.

88 lbi0=0 V µmLength dependence of bi0.

89 wbi0=0 V µmWidth dependence of bi0.

90 pbi0=0 V µmWidth-length dependence of bi0.

Length and Width Modulation Parameters

91 dl0=0 µmLateral diffusion.

92 dw0=0 µmField oxide encroachment.

93 lref=∞ mReference channel length.

94 wref=∞ mReference channel width.

95 xw=0 mWidth variation due to masking and etching.

96 xl=0 mLength variation due to masking and etching.

Temperature Effects Parameters

97 `temp (C)`Parameters measurement temperature. Default set by options.

98 `trise=0 C`Temperature rise from ambient.

99 `tempmod=432`Temperature model selector.

100 `version=432`Version selector.

101 `uto=0 C`Mobility temperature offset.

102 `ute=-1.5`Mobility temperature exponent.

103 `tlev=0`DC temperature selector.

104 `tlevc=0`AC temperature selector.

105 `eg=1.12452 V`Energy band gap.

106 `gap1=7.02e-4 V/C`^{2}
Band gap temperature coefficient.

107 `gap2=1108 K`Band gap temperature offset.

108 `trs=0 1/C`Temperature coefficient for source resistance.

109 `trd=0 1/C`Temperature coefficient for drain resistance.

110 `xti=3`Saturation current temperature exponent.

Overlap Capacitance Parameters

111 `cgso=0 F/m`Gate-source overlap capacitance.

112 `cgdo=0 F/m`Gate-drain overlap capacitance.

113 `cgbo=0 F/m`Gate-bulk overlap capacitance.

114 `meto=0 m`Metal overlap in fringing field.

Charge Model Selection Parameters

115 `capmod=bsim` Intrinsic charge model.
Possible values are `none`, `meyer`, `yang`, or `bsim`.

116 `xpart=1` Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

117 `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.

Parasitic Resistance Parameters

118 `rs=0` Ω Source resistance.

119 `rd=0` Ω Drain resistance.

120 `rsh=0` Ω/sqr Source/drain diffusion sheet resistance.

121 `rsc=0` Ω Source contact resistance.

122 `rdc=0` Ω Drain contact resistance.

123 `rss=0` Ω m Scalable source resistance.

124 `rdd=0` Ω m Scalable drain resistance.

125 `minr=0.1` Ω Minimum source/drain resistance.

126 `hdif=0` m Length of heavily doped diffusion.

127 `ldif=0` m Lateral diffusion beyond the gate.

128 `lgcs=0` m Gate-to-contact length of source side.

129 `lgcd=0` m Gate-to-contact length of drain side.

130 `sc=∞` m Spacing between contacts.

Junction Diode Parameters

131 `js` (A/m^2) Bulk junction reverse saturation current density.

132 `is=1e-14 A`Bulk junction reverse saturation current.

133 `n=1`Junction emission coefficient.

134 `imelt=`imaxA'`Explosion current, diode is linearized beyond this current to aid convergence.

135 `jmelt=`jmaxA/m'`^{2}
Explosion current density, diode is linearized beyond this current to aid convergence.

136 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`. Possible values are `no` or `yes`.

Junction Capacitance Model Parameters

137 `cbs=0 F`Bulk-source zero-bias junction capacitance.

138 `cbd=0 F`Bulk-drain zero-bias junction capacitance.

139 `cj=0 F/m`^{2}Zero-bias junction bottom capacitance density.

140 `mj=1/2`Bulk junction bottom grading coefficient.

141 `pb=0.8 V`Bulk junction potential.

142 `fc=0.5`Forward-bias depletion capacitance threshold.

143 `cjsw=0 F/m`Zero-bias junction sidewall capacitance density.

144 `mjsw=1/3`Bulk junction sidewall grading coefficient.

145 `pbsw=0.8 V`Side-wall junction potential.

146 `fcsw=0.5`Side-wall forward-bias depletion capacitance threshold.

Operating Region Warning Control Parameters

147 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

148 `imax=1 A`Maximum current, currents above this limit generate a warning.

149 `jmax=1e8 A/m`^{2}Maximum current density, currents above this limit generate a warning.

150 `bvj=∞ V`Junction reverse breakdown voltage.

151 `vbox=1e9 tox V`Oxide breakdown voltage.

Process and Power Supply Parameters

152 `tox=4e-8 m`Gate oxide thickness.

153 `vdd=5 V`Drain voltage at which parameters are extracted.

Default Device Parameters

154 `w=3e-6 m`Channel width.

155 `l=3e-6 m`Channel length.

156 `as=0 m`^{2}Area of source diffusion.

157 `ad=0 m`^{2}Area of drain diffusion.

158 `ps=0 m`Perimeter of source diffusion.

159 `pd=0 m`Perimeter of drain diffusion.

160 `nrd=0 m/m`Number of squares of drain diffusion.

161 `nrs=0 m/m`Number of squares of source diffusion.

162 `ldd=0 m`Drain diffusion length.

163 `lds=0 m`Source diffusion length.

Noise Model Parameters

164 `noisemod=1`Noise model selector.

165 `kf=0`Flicker (1/f) noise coefficient.

166 `af=1`Flicker (1/f) noise exponent.

167 `ef=1`Flicker (1/f) noise frequency exponent.

168 `wnoi=1e-5 m`Channel width at which noise parameters were extracted.

Auto Model Selector Parameters

169 `wmax=1.0 m`Maximum channel width for which the model is valid.

170 `wmin=0.0 m`Minimum channel width for which the model is valid.

171 `lmax=1.0 m`Maximum channel length for which the model is valid.

172 `lmin=0.0 m`Minimum channel length for which the model is valid.

Degradation Parameters

173 `degramod=spectre`Degradation model selector.
Possible values are `spectre` or `bert`.

174 `degradation=no`Hot-electron degradation flag.
Possible values are `no` or `yes`.

175 `dvthc=1 V`Degradation coefficient for threshold voltage.

176 `dvthe=1`Degradation exponent for threshold voltage.

177 `duoc=1 S`Degradation coefficient for transconductance.

178 `duoe=1`Degradation exponent for transconductance.

179 `crivth=0.1 V`Maximum allowable threshold voltage shift.

180 `criuo=10%`Maximum allowable normalized mobility change.

181 `crigm=10%`Maximum allowable normalized transconductance change.

182 `criids=10%`Maximum allowable normalized drain current change.

183 `wnom=5e-6 m`Nominal device width in degradation calculation.

184 `lnom=1e-6 m`Nominal device length in degradation calculation.

185 `vbsn=0 V`Substrate voltage in degradation calculation.

186 `vdsni=0.1 V`Drain voltage in `Ids` degradation calculation.

187 `vgsni=5 V`Gate voltage in `Ids` degradation calculation.

188 `vdsng=0.1 V`Drain voltage in `Gm` degradation calculation.

189 `vgsng=5 V`Gate voltage in `Gm` degradation calculation.

Spectre Stress Parameters

190 `esat=1.1e7 V/m`Critical field in `Vdsat` calculation.

191 `esatg=2.5e6 1/m`Gate voltage dependence of `esat`.

192 `vpg=-0.25`Gate voltage modifier.

193 `vpb=-0.13`Gate voltage modifier.

194 `subc1=2.24e-5`Substrate current coefficient.

195 `subc2=-0.1e-5 1/V`Substrate current coefficient.

196 `sube=6.4`Substrate current exponent.

197 `strc=1`Stress coefficient.

198 `stre=1`Stress exponent.

BERT Stress Parameters

199 `h0=1`Aging coefficient.

200 `hgd=0 1/V`Bias dependence of `h0`.

201 `m0=1`Aging exponent.

202 `mgd=0 1/V`Bias dependence of `m0`.

203 `ecrit0=1.1e5 V/cm`Critical electric field.

204 `lecrit0=0 µm V/cm`Length dependence of `ecrit0`.

205 `wecrit0=0 µm V/cm`Width dependence of `ecrit0`.

206 `ecritg=0 1/cm`Gate voltage dependence of `ecrit0`.

207 `lecritg=0 µm/cm`Length dependence of `ecritg`.

208 `wecritg=0 µm/cm`Width dependence of `ecritg`.

209 `ecritb=0 1/cm`Substrate voltage dependence of `ecrit0`.

210 `lecritb=0 µm/cm`Length dependence of `ecritb`.

211 `wecritb=0 µm/cm`Width dependence of `ecritb`.

212 `lc0=1`Substrate current coefficient.

213 `llc0=0 µm`Length dependence of `lc0`.

214 `wlc0=0 µm`Width dependence of `lc0`.

215 `lc1=1`Substrate current coefficient.

216 `llc1=0 µm`Length dependence of `lc1`.

217 `wlc1=0 µm`Width dependence of `lc1`.

218 `lc2=1`Substrate current coefficient.

219 `llc2=0 µm`Length dependence of `lc2`.

220 `wlc2=0 µm`Width dependence of `lc2`.

221 `lc3=1`Substrate current coefficient.

222 `llc3=0 µm`Length dependence of `lc3`.

223 `wlc3=0 µm`Width dependence of `lc3`.

224 `lc4=1`Substrate current coefficient.

225 `llc4=0 µm`Length dependence of `lc4`.

226 wlc4=0 µmWidth dependence of lc4.

227 lc5=1Substrate current coefficient.

228 llc5=0 µmLength dependence of lc5.

229 wlc5=0 µmWidth dependence of lc5.

230 lc6=1Substrate current coefficient.

231 llc6=0 µmLength dependence of lc6.

232 wlc6=0 µmWidth dependence of lc6.

233 lc7=1Substrate current coefficient.

234 llc7=0 µmLength dependence of lc7.

235 wlc7=0 µmWidth dependence of lc7.

Imax and Imelt

The imax parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to imax. If imax is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding imax, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, imelt, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds imelt, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of imelt to prevent arithmetic exception, with the exponential term replaced by a linear equation at imelt.

Both of these parameters have current density counterparts, jmax and jmelt, that you can specify if you want the absolute current values to depend on the device area.

Auto Model Selection

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 <= inst_length < lmax  and   wmin <= inst_width  < wmax
```

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)` Effective channel width.

2 `leff (m)` Effective channel length.

3 `rseff (Ω)` Effective source resistance.

4 `rdeff (Ω)` Effective drain resistance.

5 `aseff` (m^2) Effective area of source diffusion.

6 `adeff` (m^2) Effective area of drain diffusion.

7 `pseff (m)` Effective perimeter of source diffusion.

8 pdeff (m)Effective perimeter of source diffusion.

9 isseff (A)Effective source-bulk junction reverse saturation current.

10 isdeff (A)Effective drain-bulk junction reverse saturation current.

11 cbseff (F)Effective zero-bias source-bulk junction capacitance.

12 cbdeff (F)Effective zero-bias drain-bulk junction capacitance.

13 vto (V)Effective zero-bias threshold voltage.

14 vfb (V)Effective flat-band voltage.

15 phi (V)Effective surface potential.

16 k1 ($\sqrt{V}$)Effective body-effect coefficient.

17 k2Effective charge-sharing parameter.

18 etaEffective DIBL coefficient.

Operating-Point Parameters

1 type=nTransistor type.
Possible values are n or p.

2 region=triodeEstimated operating region.
Possible values are off, triode, sat, subth, or breakdown.

3 degradation=noHot-electron degradation flag.
Possible values are no or yes.

4 reversedReverse mode indicator.
Possible values are no or yes.

5 ids (A)Resistive drain-to-source current.

6 vgs (V)Gate-source voltage.

7 vds (V)Drain-source voltage.

8 vbs (V)Bulk-source voltage.

9 `vth (V)`Threshold voltage.

10 `vdsat (V)`Drain-source saturation voltage.

11 `betaeff` (A/V^2)Effective beta.

12 `gm (S)`Common-source transconductance.

13 `gds (S)`Common-source output conductance.

14 `gmbs (S)`Body-transconductance.

15 `cbd (F)`Drain-bulk junction capacitance.

16 `cbs (F)`Source-bulk junction capacitance.

17 `cgs (F)`Gate-source capacitance.

18 `cgd (F)`Gate-drain capacitance.

19 `cgb (F)`Gate-bulk capacitance.

20 `ron` (Ω)ON-resistance.

21 `id (A)`Resistive drain current.

22 `ibulk (A)`Resistive bulk current.

23 `pwr (W)`Power at op point.

24 `gmoverid (1/V)`Gm/Ids.

25 `isub (A)`Substrate current.

26 `stress`Hot-electron stress.

27 `age (s)`Device age.

28 `he_vdsat (V)`hot electron vdsat.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

BSIM2 Field Effect Transistor (bsim2)

Description

BSIM2 is a semiempirical MOSFET model developed at the University of California, Berkeley. All the model parameters are extracted directly from physical devices. Both the drain current and output resistance are accurately modeled. Three charge models are available. In SPICE mode, you can refer to BSIM2 as MOS level 5 or BSIM level 2. BSIM2 transistors require that you use a model statement.

This device is supported within altergroups.

To examine the equations used for this component, consult the *Spectre Circuit Simulator Device Model Equations* manual.

Sample Instance Statement

```
m2 (0 2 1 1) pchmod l=5u w=10u as=40u ad=40u pd=28u ps=28u m=1
```

Sample Model Statement

```
model pchmod bsim2 type=p vfb0=-0.5 lvfb=0.5 wvfb=0.3 phi0=0.8 eta0=0.056 k1=0.5
eg=0.99 gap1=5.5e-04 trs=1e-3 trd=1e-3 xpart=0.5 rs=10 rd=10
```

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as` (m^2)Area of source diffusion.

4 `ad` (m^2)Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `ld (m)`Drain diffusion length.

10 `ls (m)`Source diffusion length.

11 `m=1`Multiplicity factor (number of MOSFETs in parallel).

12 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

13 `trise (C)`Temperature rise from ambient.

14 `degradation=no`Hot-electron degradation flag.
Possible values are `no` or `yes`.

Model Definition

```
model modelName bsim2 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are `n` or `p`.

Threshold Voltage Parameters

2 `vfb0=-0.8 V`Flat-band voltage.

3 `lvfb=0 V µm`Length dependence of `vfb`.

4 `wvfb=0 V µm`Width dependence of `vfb`.

5 `pvfb=0 V µm`Width-length dependence of `vfb`.

6 `phi0=0.75 V`Surface potential.

7 `lphi=0 V µm`Length dependence of `phi`.

8 `wphi=0 V µm`Width dependence of `phi`.

9 `pphi=0 V µm`Width-length dependence of `phi`.

10 `k1=0.7 √V` Body-effect coefficient.

11 `lk1=0 √V µm`Length dependence of `k1`.

12 wk1=0 √V µmWidth dependence of k1.

13 pk1=0 √V µmWidth-length dependence of k1.

14 k2=0Charge-sharing parameter.

15 lk2=0 µmLength dependence of k2.

16 wk2=0 µmWidth dependence of k2.

17 pk2=0 µmWidth-length dependence of k2.

18 eta0=0Drain-induced barrier-lowering coefficient.

19 leta0=0 µmLength dependence of eta0.

20 weta0=0 µmWidth dependence of eta0.

21 peta0=0 µmWidth-length dependence of eta0.

22 etab=0 1/VBody-bias dependence of eta0.

23 letab=0 µm/VLength dependence of etab.

24 wetab=0 µm/VWidth dependence of etab.

25 petab=0 µm/VWidth-length dependence of etab.

Mobility Parameters

26 mu0=400 cm^2/V sLow-field mobility.

27 lmu0=0 cm^2/V sLength dependence of mu0.

28 wmu0=0 cm^2/V sWidth dependence of mu0.

29 pmu0=0 cm^2/V sWidth-length dependence of mu0.

30 mu0b=0 cm^2/V^2 sBody-bias dependence of muz.

31 lmu0b=0 cm^2 µm/V^2 s
Length dependence of x2mz.

32 wmu0b=0 cm^2 µm/V^2 s
Width dependence of x2mz.

33 pmu0b=0 cm^2 µm/V^2 s
Width-length dependence of x2mz.

34 mus0=450 cm^2/V sMobility in the saturation region.

35 lmus0=0 cm^2 µm/V s
Length dependence of mus0.

36 wmus0=0 cm^2 µm/V s
Width dependence of mus0.

37 pmus0=0 cm^2 µm/V s
Width-length dependence of mus0.

38 musb=0 cm^2/V^2 sBody-bias dependence of mus0.

39 lmusb=0 cm^2 µm/V^2 s
Length dependence of mus0.

40 wmusb=0 cm^2 µm/V^2 s
Length dependence of mus0.

41 pmusb=0 cm^2 µm/V^2 s
Length dependence of mus0.

42 mu20=1Empirical channel length modulation parameter.

43 lmu20=0 µmLength dependence of mu20.

44 wmu20=0 µmWidth dependence of mu20.

45 pmu20=0 µmWidth-length dependence of mu20.

46 mu2b=0 1/VBody-bias dependence of mu20.

47 lmu2b=0 µm/VLength dependence of mu2b.

48 wmu2b=0 µm/VWidth dependence of mu2b.

49 pmu2b=0 µm/VWidth-length dependence of mu2b.

50 mu2g=0 1/VGate-bias dependence of mu20.

51 lmu2g=0 µm/VLength dependence of mu2g.

52 wmu2g=0 µm/VWidth dependence of mu2g.

53 pmu2g=0 µm/VWidth-length dependence of mu2g.

54 mu30=5 cm^2/V^2 sEmpirical output resistance parameter.

55 lmu30=0 cm^2 $\mu m/V^2$ s
Length dependence of mu30.

56 wmu30=0 cm^2 $\mu m/V^2$ s
Width dependence of mu30.

57 pmu30=0 cm^2 $\mu m/V^2$ s
Width-length dependence of mu30.

58 mu3b=0 cm^2/V^3 sBody-bias dependence of mu30.

59 lmu3b=0 cm^2 $\mu m/V^3$ s
Length dependence of mu3b.

60 wmu3b=0 cm^2 $\mu m/V^3$ s
Width dependence of mu3b.

61 pmu3b=0 cm^2 $\mu m/V^3$ s
Width-length dependence of mu3b.

62 mu3g=0 cm^2/V^3 sGate-bias dependence of mu30.

63 lmu3g=0 cm^2 $\mu m/V^3$ s
Length dependence of mu3g.

64 wmu3g=0 cm^2 $\mu m/V^3$ s
Width dependence of mu3g.

65 pmu3g=0 cm^2 $\mu m/V^3$ s
Width-length dependence of mu3g.

66 mu40=0 cm^2/V^3 sEmpirical output resistance parameter.

67 lmu40=0 cm^2 µm/V^3 s
Length dependence of mu40.

68 wmu40=0 cm^2 µm/V^3 s
Width dependence of mu40.

69 pmu40=0 cm^2 µm/V^3 s
Width-length dependence of mu40.

70 mu4b=0 cm^2/V^3 sEmpirical output resistance parameter.

71 lmu4b=0 cm^2 µm/V^3 s
Length dependence of mu4b.

72 wmu4b=0 cm^2 µm/V^3 s
Width dependence of mu4b.

73 pmu4b=0 cm^2 µm/V^3 s
Width-length dependence of mu4b.

74 mu4g=0 cm^2/V^3 sGate-bias dependence of mu4g.

75 lmu4g=0 cm^2 µm/V^3 s
Length dependence of mu4g.

76 wmu4g=0 cm^2 µm/V^3 s
Width dependence of mu4g.

77 pmu4g=0 cm^2 µm/V^3 s
Width-length dependence of mu4g.

Mobility Modulation Parameters

78 ua0=0 1/VGate voltage dependence of mobility.

79 lua0=0 µm/VLength dependence of ua0.

80 wua0=0 µm/VWidth dependence of ua0.

81 pua0=0 µm/VWidth-length dependence of ua0.

82 uab=0 1/V^2Body-bias dependence of ua.

83 luab=0 µm/V^{2}Length dependence of uab.

84 wuab=0 µm/V^{2}Width dependence of uab.

85 puab=0 µm/V^{2}Width-length dependence of uab.

86 ub0=0 1/V^{2}Second-order effect of gate voltage dependence of mobility.

87 lub0=0 µm/V^{2}Length dependence of ub0.

88 wub0=0 µm/V^{2}Width dependence of ub0.

89 pub0=0 µm/V^{2}Width-length dependence of ub0.

90 ubb=0 1/V^{3}Body-bias dependence of ub.

91 lubb=0 µm/V^{3}Length dependence of ubb.

92 wubb=0 µm/V^{3}Width dependence of ubb.

93 pubb=0 µm/V^{3}Width-length dependence of ubb.

Velocity Saturation Parameters

94 u10=0 1/VVelocity saturation coefficient.

95 lu10=0 µm/VLength dependence of u1.

96 wu10=0 µm/VWidth dependence of u1.

97 pu10=0 µm/VWidth-length dependence of u1.

98 u1b=0 1/V^{2}Body-bias dependence of u1.

99 lu1b=0 µm/V^{2}Length dependence of u1b.

100 wu1b=0 µm/V^{2}Width dependence of u1b.

101 pu1b=0 µm/V^{2}Width-length dependence of u1b.

102 u1d=0 1/V^{2}Drain-bias dependence of u1.

103 lu1d=0 µm/V^{2}Length dependence of u1d.

104 wu1d=0 μm/V²Width dependence of u1d.

105 pu1d=0 μm/V²Width-length dependence of u1d.

Subthreshold Parameters

106 n0=0Subthreshold swing parameter.

107 ln0=0 μmLength dependence of subthreshold swing parameter.

108 wn0=0 μmWidth dependence of subthreshold swing parameter.

109 pn0=0 μmWidth-length dependence of subthreshold swing parameter.

110 nb=0 √V Body-bias dependence of n0.

111 lnb=0 √V μmLength dependence of nb.

112 wnb=0 √V μmWidth dependence of nb.

113 pnb=0 √V μmWidth-length dependence of nb.

114 nd=0 1/VDrain-bias dependence of n0.

115 lnd=0 μm/VLength dependence of nd.

116 wnd=0 μm/VWidth dependence of nd.

117 pnd=0 μm/VWidth-length dependence of nd.

118 vof0=1 VThreshold voltage offset in the subthreshold region.

119 lvof0=0 V μmLength dependence of vof.

120 wvof0=0 V μmWidth dependence of vof.

121 pvof0=0 V μmWidth-length dependence of vof.

122 vofb=0Body-bias dependence of vof0.

123 lvofb=0 μmLength dependence of vofb.

124 wvofb=0 μmWidth dependence of vofb.

125 pvofb=0 µmWidth-length dependence of vofb.

126 vofd=0Drain-bias dependence of vof0.

127 lvofd=0 µmLength dependence of vofd.

128 wvofd=0 µmWidth dependence of vofd.

129 pvofd=0 µmWidth-length dependence of vofd.

130 subthmod=2Subthreshold model selector.

Impact Ionization Parameters

131 ai0=0 1/VHot-electron effect on Rout parameter.

132 lai0=0 µm/VLength dependence of ai0.

133 wai0=0 µm/VWidth dependence of ai0.

134 pai0=0 µm/VWidth-length dependence of ai0.

135 aib=0 1/V^2Body-bias dependence of ai0.

136 laib=0 µm/V^2Length dependence of aib.

137 waib=0 µm/V^2Width dependence of aib.

138 paib=0 µm/V^2Width-length dependence of aib.

139 bi0=0 VHot-electron effect on Rout exponent.

140 lbi0=0 V µmLength dependence of bi0.

141 wbi0=0 V µmWidth dependence of bi0.

142 pbi0=0 V µmWidth-length dependence of bi0.

143 bib=0Body-bias dependence of bi0.

144 lbib=0 µmLength dependence of bib.

145 wbib=0 µmWidth dependence of bib.

146 pbib=0 µmWidth-length dependence of bib.

Transition Region Bound Parameters

147 vghigh=0.2 VUpper bound of the transition region.

148 lvghigh=0 V µmLength dependence of Vghigh.

149 wvghigh=0 V µmWidth dependence of Vghigh.

150 pvghigh=0 V µmWidth-length dependence of Vghigh.

151 vglow=-0.15 VLower bound of the transition region.

152 lvglow=0 V µmLength dependence of Vglow.

153 wvglow=0 V µmWidth dependence of Vglow.

154 pvglow=0 V µmWidth-length dependence of Vglow.

Length and Width Modulation Parameters

155 dl0=0 µmLateral diffusion.

156 dw0=0 µmField oxide encroachment.

157 lref=∞ mReference channel length.

158 wref=∞ mReference channel width.

159 xw=0 mWidth variation due to masking and etching.

160 xl=0 mLength variation due to masking and etching.

Temperature Effects Parameters

161 temp (C)Parameters measurement temperature. Default set by options.

162 trise=0 CTemperature rise from ambient.

163 tempmod=432Temperature model selector.

164 `version=432`Version selector.

165 `uto=0 C`Mobility temperature offset.

166 `ute=-1.5`Mobility temperature exponent.

167 `tlev=0`DC temperature selector.

168 `tlevc=0`AC temperature selector.

169 `ptc=0 V/C`Surface potential temperature coefficient.

170 `eg=1.12452 V`Energy band gap.

171 `gap1=7.02e-4 V/C`^{2}
Band gap temperature coefficient.

172 `gap2=1108 K`Band gap temperature offset.

173 `trs=0 1/C`Temperature coefficient for source resistance.

174 `trd=0 1/C`Temperature coefficient for drain resistance.

175 `xti=3`Saturation current temperature exponent.

Overlap Capacitance Parameters

176 `cgso=0 F/m`Gate-source overlap capacitance.

177 `cgdo=0 F/m`Gate-drain overlap capacitance.

178 `cgbo=0 F/m`Gate-bulk overlap capacitance.

179 `meto=0 m`Metal overlap in fringing field.

Charge Model Selection Parameters

180 `capmod=bsim`Intrinsic charge model.
Possible values are `none`, `meyer`, `yang`, or `bsim`.

181 `xpart=1`Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

182 `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.

Parasitic Resistance Parameters

183 `rs=0 Ω`Source resistance.

184 `rd=0 Ω`Drain resistance.

185 `rsh=0 Ω/sqr`Source/drain diffusion sheet resistance.

186 `rsc=0 Ω`Source contact resistance.

187 `rdc=0 Ω`Drain contact resistance.

188 `rss=0 Ω m`Scalable source resistance.

189 `rdd=0 Ω m`Scalable drain resistance.

190 `minr=0.1 Ω`Minimum source/drain resistance.

191 `hdif=0 m`Length of heavily doped diffusion.

192 `ldif=0 m`Lateral diffusion beyond the gate.

193 `lgcs=0 m`Gate-to-contact length of source side.

194 `lgcd=0 m`Gate-to-contact length of drain side.

195 `sc=∞ m`Spacing between contacts.

Junction Diode Parameters

196 `js` (A/m^2)Bulk junction reverse saturation current density.

197 `is=1e-14 A`Bulk junction reverse saturation current.

198 n=1Junction emission coefficient.

199 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`. Possible values are `no` or `yes`.

200 `imelt=`imaxA'`Explosion current, diode is linearized beyond this current to aid convergence.

201 `jmelt=`jmaxA/m'`^{2}
Explosion current density, diode is linearized beyond this current to aid convergence.

Operating Region Warning Control Parameters

202 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

203 `imax=1 A`Maximum current, currents above this limit generate a warning.

204 `jmax=1e8 A/m`^{2}Maximum current density, currents above this limit generate a warning.

205 `bvj=∞ V`Junction reverse breakdown voltage.

206 `vbox=1e9 tox V`Oxide breakdown voltage.

Junction Capacitance Model Parameters

207 `cbs=0 F`Bulk-source zero-bias junction capacitance.

208 `cbd=0 F`Bulk-drain zero-bias junction capacitance.

209 `cj=0 F/m`^{2}Zero-bias junction bottom capacitance density.

210 `mj=1/2`Bulk junction bottom grading coefficient.

211 `pb=0.8 V`Bulk junction potential.

212 `fc=0.5`Forward-bias depletion capacitance threshold.

213 `cjsw=0 F/m`Zero-bias junction sidewall capacitance density.

214 `mjsw=1/3`Bulk junction sidewall grading coefficient.

215 `pbsw=0.8 V`Side-wall junction potential.

216 `fcsw=0.5`Side-wall forward-bias depletion capacitance threshold.

Process and Power Supply Parameters

217 `tox=4e-8 m`Gate oxide thickness.

218 `vdd=5 V`Drain voltage at which parameters are extracted.

219 `vgg=5 V`Gate voltage at which parameters are extracted.

220 `vbb=-5 V`Body voltage at which parameters are extracted.

Default Device Parameters

221 `w=3e-6 m`Channel width.

222 `l=3e-6 m`Channel length.

223 `as=0 m`^{2}Area of source diffusion.

224 `ad=0 m`^{2}Area of drain diffusion.

225 `ps=0 m`Perimeter of source diffusion.

226 `pd=0 m`Perimeter of drain diffusion.

227 `nrd=0 m/m`Number of squares of drain diffusion.

228 `nrs=0 m/m`Number of squares of source diffusion.

229 `ldd=0 m`Drain diffusion length.

230 `lds=0 m`Source diffusion length.

Noise Model Parameters

231 `noisemod=1`Noise model selector.

232 `kf=0`Flicker (1/f) noise coefficient.

233 `af=1`Flicker (1/f) noise exponent.

234 `ef=1`Flicker (1/f) noise frequency exponent.

235 `wnoi=1e-5 m`Channel width at which noise parameters were extracted.

Auto Model Selector Parameters

236 `wmax=1.0 m`Maximum channel width for which the model is valid.

237 `wmin=0.0 m`Minimum channel width for which the model is valid.

238 `lmax=1.0 m`Maximum channel length for which the model is valid.

239 `lmin=0.0 m`Minimum channel length for which the model is valid.

Degradation Parameters

240 `degramod=spectre`Degradation model selector.
Possible values are `spectre` or `bert`.

241 `degradation=no`Hot-electron degradation flag.
Possible values are `no` or `yes`.

242 `dvthc=1 V`Degradation coefficient for threshold voltage.

243 `dvthe=1`Degradation exponent for threshold voltage.

244 `duoc=1 S`Degradation coefficient for transconductance.

245 `duoe=1`Degradation exponent for transconductance.

246 `crivth=0.1 V`Maximum allowable threshold voltage shift.

247 `criuo=10%`Maximum allowable normalized mobility change.

248 `crigm=10%`Maximum allowable normalized transconductance change.

249 `criids=10%`Maximum allowable normalized drain current change.

250 `wnom=5e-6 m`Nominal device width in degradation calculation.

251 `lnom=1e-6 m`Nominal device length in degradation calculation.

252 `vbsn=0 V`Substrate voltage in degradation calculation.

253 `vdsni=0.1 V`Drain voltage in `Ids` degradation calculation.

254 `vgsni=5 V`Gate voltage in `Ids` degradation calculation.

255 `vdsng=0.1 V`Drain voltage in `Gm` degradation calculation.

256 `vgsng=5 V`Gate voltage in `Gm` degradation calculation.

Spectre Stress Parameters

257 `esat=1.1e7 V/m`Critical field in `Vdsat` calculation.

258 `esatg=2.5e6 1/m`Gate voltage dependence of `esat`.

259 `vpg=-0.25`Gate voltage modifier.

260 `vpb=-0.13`Gate voltage modifier.

261 `subc1=2.24e-5`Substrate current coefficient.

262 `subc2=-0.1e-5 1/V`Substrate current coefficient.

263 `sube=6.4`Substrate current exponent.

264 `strc=1`Stress coefficient.

265 `stre=1`Stress exponent.

BERT Stress Parameters

266 `h0=1`Aging coefficient.

267 `hgd=0 1/V`Bias dependence of `h0`.

268 `m0=1`Aging exponent.

269 `mgd=0 1/V`Bias dependence of `m0`.

270 `ecrit0=1.1e5 V/cm`Critical electric field.

271 `lecrit0=0 µm V/cm`Length dependence of `ecrit0`.

272 `wecrit0=0 µm V/cm`Width dependence of `ecrit0`.

273 `ecritg=0 1/cm`Gate voltage dependence of `ecrit0`.

274 `lecritg=0 µm/cm`Length dependence of `ecritg`.

275 `wecritg=0 µm/cm`Width dependence of `ecritg`.

276 `ecritb=0 1/cm`Substrate voltage dependence of `ecrit0`.

277 `lecritb=0 µm/cm`Length dependence of `ecritb`.

278 `wecritb=0 µm/cm`Width dependence of `ecritb`.

279 `lc0=1`Substrate current coefficient.

280 `llc0=0 µm`Length dependence of `lc0`.

281 `wlc0=0 µm`Width dependence of `lc0`.

282 `lc1=1`Substrate current coefficient.

283 `llc1=0 µm`Length dependence of `lc1`.

284 `wlc1=0 µm`Width dependence of `lc1`.

285 `lc2=1`Substrate current coefficient.

286 `llc2=0 µm`Length dependence of `lc2`.

287 `wlc2=0 µm`Width dependence of `lc2`.

288 `lc3=1`Substrate current coefficient.

289 `llc3=0 µm`Length dependence of `lc3`.

290 `wlc3=0 µm`Width dependence of `lc3`.

291 `lc4=1`Substrate current coefficient.

292 `llc4=0 µm`Length dependence of `lc4`.

293 `wlc4=0 µm`Width dependence of `lc4`.

294 `lc5=1`Substrate current coefficient.

295 `llc5=0` μmLength dependence of `lc5`.

296 `wlc5=0` μmWidth dependence of `lc5`.

297 `lc6=1`Substrate current coefficient.

298 `llc6=0` μmLength dependence of `lc6`.

299 `wlc6=0` μmWidth dependence of `lc6`.

300 `lc7=1`Substrate current coefficient.

301 `llc7=0` μmLength dependence of `lc7`.

302 `wlc7=0` μmWidth dependence of `lc7`.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

Both of these parameters have current density counterparts, `jmax` and `jmelt`, that you can specify if you want the absolute current values to depend on the device area.

Auto Model Selection

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 <= inst_length < lmax  and   wmin <= inst_width  < wmax
```

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)`Effective channel width.

2 `leff (m)`Effective channel length.

3 `rseff (Ω)`Effective source resistance.

4 `rdeff (Ω)`Effective drain resistance.

5 `aseff` (m^2)Effective area of source diffusion.

6 `adeff` (m^2)Effective area of drain diffusion.

7 `pseff (m)`Effective perimeter of source diffusion.

8 `pdeff (m)`Effective perimeter of source diffusion.

9 `isseff (A)`Effective source-bulk junction reverse saturation current.

10 `isdeff (A)`Effective drain-bulk junction reverse saturation current.

11 `cbseff (F)`Effective zero-bias source-bulk junction capacitance.

12 `cbdeff (F)`Effective zero-bias drain-bulk junction capacitance.

13 `vto (V)`Effective zero-bias threshold voltage.

14 `vfb (V)`Effective flat-band voltage.

15 `phi (V)`Effective surface potential.

16 `k1 (√V )`Effective body-effect coefficient.

17 `k2`Effective charge-sharing parameter.

18 `eta`Effective DIBL coefficient.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are `n` or `p`.

2 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

3 `degradation=no`Hot-electron degradation flag.
Possible values are `no` or `yes`.

4 `reversed`Reverse mode indicator.
Possible values are `no` or `yes`.

5 `ids (A)`Resistive drain-to-source current.

6 `vgs (V)`Gate-source voltage.

7 `vds (V)`Drain-source voltage.

8 `vbs (V)`Bulk-source voltage.

9 `vth (V)`Threshold voltage.

10 `vdsat (V)`Drain-source saturation voltage.

11 `betaeff (A/V`^{2}`)`Effective beta.

12 `gm (S)`Common-source transconductance.

13 `gds (S)`Common-source output conductance.

14 `gmbs (S)`Body-transconductance.

15 `cbd (F)`Drain-bulk junction capacitance.

16 `cbs (F)`Source-bulk junction capacitance.

17 `cgs (F)`Gate-source capacitance.

18 `cgd (F)`Gate-drain capacitance.

19 `cgb (F)`Gate-bulk capacitance.

20 `ron (Ω)`ON-resistance.

21 `id (A)`Resistive drain current.

22 `ibulk (A)`Resistive bulk current.

23 `pwr (W)`Power at op point.

24 `gmoverid (1/V)`Gm/Ids.

25 `isub (A)`Substrate current.

26 `stress`Hot-electron stress.

27 `age (s)`Device age.

28 `he_vdsat (V)`hot electron vdsat.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the

description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

```
ad   M-224
ad   I-4
adeff   O-6
af   M-233
age   OP-27
ai0   M-131
aib   M-135
alarm   M-202
as   I-3
as   M-223
aseff   O-5
betaeff   OP-11
bi0   M-139
bib   M-143
bvj   M-205
capmod   M-180
cbd   OP-15
cbd   M-208
cbdeff   O-12
cbs   OP-16
cbs   M-207
cbseff   O-11
cgb   OP-19
cgbo   M-178
cgd   OP-18
cgdo   M-177
cgs   OP-17
cgso   M-176
cj   M-209
cjsw   M-213
crigm   M-248
criids   M-249
criuo   M-247
crivth   M-246
degradation   OP-3
degradation   M-241
degradation   I-14
degramod   M-240
dl0   M-155
dskip   M-199
duoc   M-244
duoe   M-245
lc1   M-282
lc2   M-285
lc3   M-288
lc4   M-291
lc5   M-294
lc6   M-297
lc7   M-300
ld   I-9
ldd   M-229
ldif   M-192
lds   M-230
lecrit0   M-271
lecritb   M-277
lecritg   M-274
leff   O-2
leta0   M-19
letab   M-23
lgcd   M-194
lgcs   M-193
lk1   M-11
lk2   M-15
llc0   M-280
llc1   M-283
llc2   M-286
llc3   M-289
llc4   M-292
llc5   M-295
llc6   M-298
llc7   M-301
lmax   M-238
lmin   M-239
lmu0   M-27
lmu0b   M-31
lmu20   M-43
lmu2b   M-47
lmu2g   M-51
lmu30   M-55
lmu3b   M-59
lmu3g   M-63
lmu40   M-67
lmu4b   M-71
lmu4g   M-75
nrs   I-8
nrs   M-228
pai0   M-134
paib   M-138
pb   M-211
pbi0   M-142
pbib   M-146
pbsw   M-215
pd   I-6
pd   M-226
pdeff   O-8
peta0   M-21
petab   M-25
phi   O-15
phi0   M-6
pk1   M-13
pk2   M-17
pmu0   M-29
pmu0b   M-33
pmu20   M-45
pmu2b   M-49
pmu2g   M-53
pmu30   M-57
pmu3b   M-61
pmu3g   M-65
pmu40   M-69
pmu4b   M-73
pmu4g   M-77
pmus0   M-37
pmusb   M-41
pn0   M-109
pnb   M-113
pnd   M-117
pphi   M-9
ps   I-5
ps   M-225
pseff   O-7
ptc   M-169
pu10   M-97
pu1b   M-101
pu1d   M-105
pua0   M-81
ute   M-166
uto   M-165
vbb   M-220
vbox   M-206
vbs   OP-8
vbsn   M-252
vdd   M-218
vds   OP-7
vdsat   OP-10
vdsng   M-255
vdsni   M-253
version   M-164
vfb   O-14
vfb0   M-2
vgg   M-219
vghigh   M-147
vglow   M-151
vgs   OP-6
vgsng   M-256
vgsni   M-254
vof0   M-118
vofb   M-122
vofd   M-126
vpb   M-260
vpg   M-259
vth   OP-9
vto   O-13
w   M-221
w   I-1
wai0   M-133
waib   M-137
wbi0   M-141
wbib   M-145
wecrit0   M-272
wecritb   M-278
wecritg   M-275
weff   O-1
weta0   M-20
wetab   M-24
wk1   M-12
wk2   M-16
wlc0   M-281
```

BSIM3 MOS Transistor (bsim3)

Description

This is the BSIM3 version-2 (BSIM3v2) model. The BSIM3v2 model is a physically-based, predictive, and computationally efficient model developed at the University of California, Berkeley. It is suitable for both digital and analog applications. In SPICE mode, refer to BSIM3 as MOS level 10. BSIM3 transistors require that you use a model statement.

This device is supported within altergroups.

To examine the equations used for this component, consult the *Spectre Circuit Simulator Device Model Equations* manual.

Sample Instance Statement

```
m3 (1 2 0 0) nchmod l=1.5u w=100u as=450p ad=450p pd=209u ps=209u nrd=207m nrs=207m
m=1
```

Sample Model Statement

```
model nchmod bsim3 vtho=5.94e-01 phi=0.69 k1=0.72 k2=0 w0=1.3e-07 tox=5.9e-09
rdsw=80 uo=499 xj=2e-07 vsat=600e+04 at=3.4e+04 a0=0.8 cdsc=1.4e-03 nfactor=1.03
```

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as` (m^2)Area of source diffusion.

4 `ad` (m^2)Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `m=1`Multiplicity factor (number of MOSFETs in parallel).

10 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

11 `trise`Temperature rise from ambient.

Model Definition

```
model modelName bsim3 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

Threshold Voltage Parameters

2 `vtho=0 V`Threshold voltage at zero body bias.

3 `phi=0.7 V`Surface potential at strong inversion.

4 `k1=0.53 √V` Body-effect coefficient.

5 `k2=-0.0186`Charge-sharing parameter.

6 `k3=80`Narrow width coefficient.

7 `k3b=0 1/V`Narrow width coefficient.

8 `w0=2.5e-6 m`Narrow width coefficient.

9 `nlx=1.74e-7 m`Lateral nonuniform doping coefficient.

10 `gamma1=0` √V Body-effect coefficient near the surface.

11 `gamma2=0` √V Body-effect coefficient in the bulk.

12 `theta=0.02 1/V`Drain-induced barrier lowering coefficient.

13 `eta=0.3 1/V`Effective drain voltage coefficient.

14 `litl (m)`Depth of current path.

15 `vfb (V)`Flat-band voltage.

16 `vbx (V)`Threshold voltage transition body voltage.

17 `vbi (V)`Substrate junction built-in potential.

18 `vbm=-5 V`Maximum applied body voltage.

19 `dvt0=2.2`First coefficient of short-channel effects.

20 `dvt1=0.53`Second coefficient of short-channel effects.

21 `dvt2=-0.032 1/V`Body-bias coefficient of short-channel effects.

22 `a0=1 for nmos and 4.4 for pmos`
Nonuniform depletion width effect coefficient.

23 `a1=0 for nmos, 0.23 for pmos`
No-saturation coefficient.

24 `a2=1 for nmos, 0.08 for pmos`
No-saturation coefficient.

25 `keta=-0.047 1/V`Body-bias coefficient for non-uniform depletion width effect.

Process Parameters

26 `nsub=2e15 cm`$^{-3}$Substrate doping concentration.

27 `npeak=1.7e17 cm`$^{-3}$
Peak channel doping concentration.

28 `ngate (cm`$^{-3}$`)`Poly-gate doping concentration.

29 `xj=0.15e-6 m`Source/drain junction depth.

30 `dl=0 m`Lateral diffusion for one side.

31 `dw=0 m`Width reduction for one side.

32 `tox=1.5e-8 m`Gate oxide thickness.

33 `vdd=5 V`Maximum drain voltage.

34 `xt=1.55e-7 m`Doping depth.

35 `ldd=0 m`Total length of lightly doped drain region.

36 `rds0=0` Ω Total drain-source resistance.

37 `rdsw=0` Ω µmWidth dependence of drain-source resistance.

Mobility Parameters

38 `uo=670` $cm^2/V\ s$ Low-field surface mobility at `tnom`. Default is 250 for PMOS.

39 `vsat=9.58e4 m/s`Carrier saturation velocity at `tnom`.

40 `ua=2.25e-9 m/v`First-order mobility reduction coefficient.

41 `ub=5.87e-19` m^2/v^2
Second-order mobility reduction coefficient.

42 `uc=0.0465 1/V`Body-bias dependence of mobility.

43 `uc0=0`Mobility coefficient.

Output Resistance Parameters

44 `satmod=2`Saturation model selector.

45 `bulkmod=1`Bulk-charge effect model selector.

46 `drout=0.56`DIBL effect on output resistance coefficient.

47 `alpha=1.9`Reference voltage multiplication factor.

48 `em=4.1e7 V/m`Maximum electric field.

49 `pclm=1.3`Channel length modulation coefficient.

50 `pdibl1=0.39`First coefficient of drain-induced barrier lowering.

51 `pdibl2=8.6e-3`Second coefficient of drain-induced barrier lowering.

52 `pscbe1=4.24e8 V/m`First coefficient of substrate current body effect.

53 `pscbe2=1e-5 m/v`Second coefficient of substrate current body effect.

54 `pvag=0`Gate dependence of Early voltage.

Subthreshold Parameters

55 `subthmod=2`Subthreshold model selector.

56 `vghigh=0.12 V`Upper bound of transition region.

57 `vglow=-0.12 V`Lower bound of transition region.

58 `cdsc=2.4e-4 F/m`^{2}Source/drain and channel coupling capacitance.

59 `cdscb=0 F/m`^{2} VBody-bias dependence of `cdsc`.

60 `nfactor=1`Subthreshold swing coefficient.

61 `cit=0 F`Interface trap parameter for subthreshold swing.

62 `voff=-0.11 V`Threshold voltage offset.

63 `dsub=drout`DIBL effect in subthreshold region.

64 `eta0=0.08`DIBL coefficient subthreshold region.

65 `etab=-0.07 1/V`Body-bias dependence of `et0`.

Parasitic Resistance Parameters

66 `rsh=0 Ω/sqr`Source/drain diffusion sheet resistance.

67 `rs=0` ΩSource resistance.

68 `rd=0` ΩDrain resistance.

69 `lgcs=0` mGate-to-contact length of source side.

70 `lgcd=0` mGate-to-contact length of drain side.

71 `rsc=0` ΩSource contact resistance.

72 `rdc=0` ΩDrain contact resistance.

73 `rss=0` Ω mScalable source resistance.

74 `rdd=0` Ω mScalable drain resistance.

75 `sc=∞` mSpacing between contacts.

76 `ldif=0` mLateral diffusion beyond the gate.

77 `hdif=0` mLength of heavily doped diffusion.

78 `minr=0.1` ΩMinimum source/drain resistance.

Junction Diode Model Parameters

79 `js` (A/m^2)Bulk junction reverse saturation current density.

80 `is=1e-14` ABulk junction reverse saturation current.

81 n=1Junction emission coefficient.

82 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`.
Possible values are no or `yes`.

83 `imelt=`imaxA'`Explosion current.

84 `jmelt=`jmaxA/m'`^{2}
Explosion current density.

Overlap Capacitance Parameters

85 `cgso (F/m)`Gate-source overlap capacitance.

86 `cgdo (F/m)`Gate-drain overlap capacitance.

87 `cgbo (F/m)`Gate-bulk overlap capacitance.

88 `meto=0 m`Metal overlap in fringing field.

Junction Capacitance Model Parameters

89 `cbs=0 F`Bulk-source zero-bias junction capacitance.

90 `cbd=0 F`Bulk-drain zero-bias junction capacitance.

91 `cj=5e-4 F/m`^{2}Zero-bias junction bottom capacitance density.

92 `mj=1/2`Bulk junction bottom grading coefficient.

93 `pb=0.8 V`Bulk junction built-in potential.

94 `fc=0.5`Forward-bias depletion capacitance threshold.

95 `cjsw=5e-10 F/m`Zero-bias junction sidewall capacitance density.

96 `mjsw=1/3`Bulk junction sidewall grading coefficient.

97 `pbsw=0.8 V`Side-wall junction built-in potential.

98 `fcsw=0.5`Side-wall forward-bias depletion capacitance threshold.

Charge Model Selection Parameters

99 `capmod=yang`Intrinsic charge model.
Possible values are `none`, `meyer`, `yang`, or `bsim`.

100 `xpart=1`Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

101 `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, 0 for 0/100.

Default Instance Parameters

102 `w=5e-6 m`Default channel width.

103 `l=5e-6 m`Default channel length.

104 `as=0 m`^{2}Default area of source diffusion.

105 `ad=0 m`^{2}Default area of drain diffusion.

106 `ps=0 m`Default perimeter of source diffusion.

107 `pd=0 m`Default perimeter of drain diffusion.

108 `nrd=0 m/m`Default number of squares of drain diffusion.

109 `nrs=0 m/m`Default number of squares of source diffusion.

Temperature Effects Parameters

110 `tnom (C)`Parameters measurement temperature. Default set by `options`.

111 `trise=0 C`Temperature rise from ambient.

112 `tlev=0`DC temperature selector.

113 `tlevc=0`AC temperature selector.

114 `eg=1.12452 V`Energy band gap.

115 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

116 `gap2=1108 C`Band gap temperature offset.

117 `kt1=-0.11 V`Temperature coefficient for threshold voltage.

118 `kt1l=-1.86e-7 v m`Temperature coefficient for threshold voltage.

119 `kt2=0.022`Temperature coefficient for threshold voltage.

120 `at=3.3e4 m/s`Temperature coefficient for `vsat`.

121 `ua1=4.31e-9 m/v`Temperature coefficient for ua.

122 ub1=-7.61e-18 m^2/v^2
Temperature coefficient for ub.

123 uc1=-0.056 1/VTemperature coefficient for uc.

124 trs=0 1/CTemperature parameter for source resistance.

125 trd=0 1/CTemperature parameter for drain resistance.

126 ute=-1.5Mobility temperature exponent.

127 xti=3Saturation current temperature exponent.

128 ptc=0 V/CSurface potential temperature coefficient.

129 tcv=0 V/CThreshold voltage temperature coefficient.

130 pta=0 V/CJunction potential temperature coefficient.

131 ptp=0 V/CSidewall junction potential temperature coefficient.

132 cta=0 1/CJunction capacitance temperature coefficient.

133 ctp=0 1/CSidewall junction capacitance temperature coefficient.

Noise Model Parameters

134 noisemod=1Noise model selector.

135 kf=0Flicker (1/f) noise coefficient.

136 af=1Flicker (1/f) noise exponent.

137 ef=1Flicker (1/f) noise frequency exponent.

138 wnoi=1e-5 mChannel width at which noise parameters were extracted.

139 a=1e16 for nmos and 9.9e14 for pmos
Oxide trap density coefficient.

140 b=5e4 for nmos and 2.4e3 for pmos
Oxide trap density coefficient.

141 `c=-1.4e-8 for nmos and 1.4e-8 for pmos`
Oxide trap density coefficient.

Operating Region Warning Control Parameters

142 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

143 `imax=1 A`Maximum allowable current.

144 `jmax=1e8 A/m`^{2}Maximum allowable current density.

145 `bvj=∞ V`Junction reverse breakdown voltage.

146 `vbox=1e9 tox V`Oxide breakdown voltage.

147 `maxvp=1.12 V`Maximum allowable voltage across the gate poly layer.

Auto Model Selector Parameters

148 `wmax=1.0 m`Maximum channel width for which the model is valid.

149 `wmin=0.0 m`Minimum channel width for which the model is valid.

150 `lmax=1.0 m`Maximum channel length for which the model is valid.

151 `lmin=0.0 m`Minimum channel length for which the model is valid.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

Both of these parameters have current density counterparts, `jmax` and `jmelt`, that you can specify if you want the absolute current values to depend on the device area.

Auto Model Selection

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 <= inst_length < lmax  and   wmin <= inst_width  < wmax
```

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)`Effective channel width.

2 `leff (m)`Effective channel length.

3 `rseff (Ω)`Effective source resistance.

4 `rdeff` (Ω)Effective drain resistance.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are n or p.

2 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

3 `reversed`Reverse mode indicator.
Possible values are no or `yes`.

4 `ids (A)`Resistive drain-to-source current.

5 `vgs (V)`Gate-source voltage.

6 `vds (V)`Drain-source voltage.

7 `vbs (V)`Bulk-source voltage.

8 `vth (V)`Threshold voltage.

9 `vdsat (V)`Drain-source saturation voltage.

10 `gm (S)`Common-source transconductance.

11 `gds (S)`Common-source output conductance.

12 `gmbs (S)`Body-transconductance.

13 `betaeff` (A/V^2)Effective `beta`.

14 `cbd (F)`Drain-bulk junction capacitance.

15 `cbs (F)`Source-bulk junction capacitance.

16 `cgs (F)`Gate-source capacitance.

17 `cgd (F)`Gate-drain capacitance.

18 `cgb (F)`Gate-bulk capacitance.

19 `ron (Ω)`On-resistance.

20 `id (A)`Resistive drain current.

21 `ibulk (A)`Resistive bulk current.

22 `pwr (W)`Power at op point.

23 `gmoverid (1/V)`Gm/Ids.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

BSIM3v3 MOS Transistor (bsim3v3)

Description

BSIM3v3 is the version-3 of bsim3 model.The versions supported are 3.1, 3.2, 3.21, 3.22, 3.23 and 3.24. It uses single-piece equations for all regions to improve the smoothness of the model characteristics. BSIM3v3 also allows the binning option like the approach used in bsim1 and bsim2. This option is provided for people who want to achieve the highest accuracy of the model. The binning equation is given by

P = P0 + Pl / Leff + Pw / Weff + Pp / (Leff * 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.

For more information on this model, please consult the University of California at Berkeley BSIM3 home page at

`http://www-device.eecs.berkeley.edu/~bsim3/index.html`

This device is supported within altergroups.

To examine the equations used for this component, consult the *Spectre Circuit Simulator Device Model Equations* manual.

Sample Instance Statement

```
m4 (0 2 1 1) pchmod w=2u l=0.8u as=250p ad=250p pd=168p ps=168p m=1
```

Sample Model Statement

```
model pchmod bsim3v3 type=p mobmod=1 capmod=2 version=3.1 tox=9e-5 cdsc=1e-3
cdscb=-4.36889e-4 cdscd=0 cit=0 nfactor=1.79 xj=1.5e-7 vsat=1.5737e5 at=1e5
a0=1.2522809 ags=0.2912413 a1=1.01222e-4 a2=0.996841 keta=0 nch=4.06263e17
ngate=7.6e19 k1=0.823562
```

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as (m`^{2}`)`Area of source diffusion.

4 `ad (m`^{2}`)`Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `m=1`Multiplicity factor (number of MOSFETs in parallel).

10 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

11 `nqsmod`NQS flag.

12 `trise`Temperature rise from ambient.

Model Definition

```
model modelName bsim3v3 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

Threshold Voltage Parameters

2 `vtho (V)`Threshold voltage at zero body bias for long-channel devices. For enhancement-mode devices, vtho > 0 for n-channel and vth < 0 for p-channel. Default value is calculated from other model parameters.

3 `vfb=-1`Flat-band voltage.

4 `k1=0.5 √V` Body-effect coefficient.

5 `k2=-0.0186`Charge-sharing parameter.

6 `k3=80`Narrow width coefficient.

7 `k3b=0 1/V`Narrow width coefficient.

8 `w0=2.5e-6 m`Narrow width coefficient.

9 `nlx=1.74e-7 m`Lateral nonuniform doping coefficient.

10 `gamma1 (√V )`Body-effect coefficient near the surface.

11 `gamma2 (√V )`Body-effect coefficient in the bulk.

12 `vbx (V)`Threshold voltage transition body voltage.

13 `vbm=-3 V`Maximum applied body voltage.

14 `dvt0=2.2`First coefficient of short-channel effects.

15 `dvt1=0.53`Second coefficient of short-channel effects.

16 `dvt2=-0.032 1/V`Body-bias coefficient of short-channel effects.

17 `dvt0w=0`First coefficient of narrow-width effects.

18 `dvt1w=5.3e6`Second coefficient of narrow-width effects.

19 `dvt2w=-0.032 1/V`Body-bias coefficient of narrow-width effects.

20 `a0=1`Nonuniform depletion width effect coefficient.

21 `b0=0 m`Bulk charge coefficient due to narrow width effect.

22 `b1=0 m`Bulk charge coefficient due to narrow width effect.

23 `a1=0`No-saturation coefficient.

24 `a2=1`No-saturation coefficient.

25 `ags=0 F/m`^{2} `V`Gate-bias dependence of Abulk.

26 `keta=-0.047 1/V`Body-bias coefficient for non-uniform depletion width effect.

Process Parameters

27 `nsub=6e16 cm`$^{-3}$Substrate doping concentration.

28 `nch=1.7e17 cm`$^{-3}$Peak channel doping concentration.

29 `ngate=∞ cm`$^{-3}$
Poly-gate doping concentration.

30 `xj=0.15e-6 m`Source/drain junction depth.

31 `lint=0 m`Lateral diffusion for one side.

32 `wint=0 m`Width reduction for one side.

33 `ll=0 m`Length dependence of delta L.

34 `lln=1`Length exponent of delta L.

35 `lw=0 m`Width dependence of delta L.

36 `lwn=1`Width exponent of delta L.

37 `lwl=0` m^2Area dependence of delta L.

38 `wl=0 m`Length dependence of delta W.

39 `wln=1`Length exponent of delta W.

40 `ww=0 m`Width dependence of delta W.

41 `wwn=1`Width exponent of delta W.

42 `wwl=0` m^2Area dependence of delta W.

43 `dwg=0 m/v`Gate-bias dependence of channel width.

44 `dwb=0` $m/\sqrt{v}$ Body-bias dependence of channel width.

45 `tox=1.5e-8 m`Gate oxide thickness.

46 `toxm=tox m`Tox at which parameters were extracted.

47 `xt=1.55e-7 m`Doping depth.

48 `rdsw=0` Ω μmWidth dependence of drain-source resistance.

49 `prwb=0` $1/\sqrt{v}$ Body-effect coefficient for Rds.

50 `prwg=0 1/V`Gate-effect coefficient for Rds.

51 `wr=1`Width offset for parasitic resistance.

52 `binunit=1`Bin parameter unit selector. 1 for microns and 2 for meters.

Mobility Parameters

53 `mobmod=1`Mobility model selector.

54 u0=670 cm^2/V sLow-field surface mobility at tnom. Default is 250 for PMOS Mobility can also be specified in M2/Vs..

55 vsat=8e4 m/sCarrier saturation velocity at tnom.

56 ua=2.25e-9 m/vFirst-order mobility reduction coefficient.

57 ub=5.87e-19 m^2/v^2
Second-order mobility reduction coefficient.

58 uc=-4.65e-11 m/v^2
Body-bias dependence of mobility. Default is -0.046 and unit is 1/V for mobmod=3.

Output Resistance Parameters

59 drout=0.56DIBL effect on output resistance coefficient.

60 pclm=1.3Channel length modulation coefficient.

61 pdiblc1=0.39First coefficient of drain-induced barrier lowering.

62 pdiblc2=8.6e-3Second coefficient of drain-induced barrier lowering.

63 pdiblcb=0 1/VBody-effect coefficient for DIBL.

64 pscbe1=4.24e8 V/mFirst coefficient of substrate current body effect.

65 pscbe2=1e-5 m/vSecond coefficient of substrate current body effect.

66 pvag=0Gate dependence of Early voltage.

67 delta=0.01 VEffective drain voltage smoothing parameter.

Subthreshold Parameters

68 cdsc=2.4e-4 F/m^2Source/drain and channel coupling capacitance.

69 cdscb=0 F/m^2 VBody-bias dependence of cdsc.

70 cdscd=0 F/m^2 VDrain-bias dependence of cdsc.

71 `nfactor=1`Subthreshold swing coefficient.

72 `cit=0 F`Interface trap parameter for subthreshold swing.

73 `voff=-0.08 V`Threshold voltage offset.

74 `dsub=drout`DIBL effect in subthreshold region.

75 `eta0=0.08`DIBL coefficient subthreshold region.

76 `etab=-0.07 1/V`Body-bias dependence of `et0`.

Substrate Current Parameters

77 `alpha0=0 m/v`Substrate current impact ionization coefficient.

78 `alpha1=0 1/V`Substrate current impact ionization coefficient.

79 `beta0=30 1/V`Substrate current impact ionization exponent.

Parasitic Resistance Parameters

80 `rsh=0 Ω/sqr`Source/drain diffusion sheet resistance.

81 `rs=0 Ω`Source resistance.

82 `rd=0 Ω`Drain resistance.

83 `lgcs=0 m`Gate-to-contact length of source side.

84 `lgcd=0 m`Gate-to-contact length of drain side.

85 `rsc=0 Ω`Source contact resistance.

86 `rdc=0 Ω`Drain contact resistance.

87 `rss=0 Ω m`Scalable source resistance.

88 `rdd=0 Ω m`Scalable drain resistance.

89 `sc=∞ m`Spacing between contacts.

90 `ldif=0 m`Lateral diffusion beyond the gate.

91 `hdif=0 m`Length of heavily doped diffusion.

92 `minr=0.1 Ω`Minimum source/drain resistance.

Junction Diode Model Parameters

93 `js (A/m`^{2}`)`Bulk junction reverse saturation current density.

94 `jsw=0 A/m`Sidewall junction reverse saturation current density.

95 `is=1e-14 A`Bulk junction reverse saturation current.

96 `n=1`Junction emission coefficient.

97 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`.
Possible values are `no` or `yes`.

98 `imelt=`imaxA'`Explosion current.

99 `jmelt=`jmaxA/m'`^{2}
Explosion current density.

Overlap Capacitance Parameters

100 `cgso (F/m)`Gate-source overlap capacitance.

101 `cgdo (F/m)`Gate-drain overlap capacitance.

102 `cgbo=2 Dwc Cox F/m`
Gate-bulk overlap capacitance. The default value is 0 if version=3.0.

103 `meto=0 m`Metal overlap in fringing field.

104 `cgsl=0 F/m`Gate-source overlap capacitance in LDD region.

105 `cgdl=0 F/m`Gate-drain overlap capacitance in LDD region.

106 `ckappa=0.6`Overlap capacitance fitting parameter.

Junction Capacitance Model Parameters

107 `cbs=0 F`Bulk-source zero-bias junction capacitance.

108 `cbd=0 F`Bulk-drain zero-bias junction capacitance.

109 `cj=5e-4 F/m`^{2}Zero-bias junction bottom capacitance density.

110 `mj=1/2`Bulk junction bottom grading coefficient.

111 `pb=1 V`Bulk junction built-in potential.

112 `fc=0.5`Forward-bias depletion capacitance threshold.

113 `cjsw=5e-10 F/m`Zero-bias junction sidewall capacitance density.

114 `mjsw=0.33`Bulk junction sidewall grading coefficient.

115 `pbsw=1 V`Side-wall junction built-in potential.

116 `cjswg=cjsw F/m`Zero-bias gate-side junction capacitance density.

117 `mjswg=mjsw`Gate-side junction grading coefficient.

118 `pbswg=pbsw V`Gate-side junction built-in potential.

119 `fcsw=0.5`Side-wall forward-bias depletion capacitance threshold.

Charge Model Selection Parameters

120 `capmod=2`Intrinsic charge model.

121 `nqsmod=0`Non-quasi static model selector. Set to 1 to turn on nqs.

122 `dwc=wint m`Delta W for capacitance model.

123 `dlc=lint m`Delta L for capacitance model.

124 `clc=1e-7 m`Intrinsic capacitance fitting parameter.

125 `cle=0.6`Intrinsic capacitance fitting parameter.

126 `cf (F/m)`Fringe capacitance parameter.

127 `elm=5`Elmore constant of the channel.

128 `vfbcv=-1`Flat-band voltage for capmod=0.

129 `acde=1 1/V`CV parameter.

130 `moin=15 1/V`CV parameter.

131 `noff=1`Transition parameter.

132 `voffcv=0`Transition parameter.

133 `xpart=0`Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

134 `llc=ll m`Length dependence of delta L for CV.

135 `lwc=lw m`Width dependence of delta L for CV.

136 `lwlc=lwl m`^{2}Area dependence of delta L for CV.

137 `wlc=wl m`Length dependence of delta W for CV.

138 `wwc=ww m`Width dependence of delta W for CV.

139 `wwlc=wwl m`^{2}Area dependence of delta W for CV.

Default for Instance Parameters

140 `w=5e-6 m`Default channel width.

141 `l=5e-6 m`Default channel length.

142 `as=0 m`^{2}Default area of source diffusion.

143 `ad=0 m`^{2}Default area of drain diffusion.

144 `ps=0 m`Default perimeter of source diffusion.

145 `pd=0 m`Default perimeter of drain diffusion.

146 `nrd=0 m/m`Default number of squares of drain diffusion.

147 `nrs=0 m/m`Default number of squares of source diffusion.

148 `version=3.1`Model version selector. The available versions are 3.1, 3.2, 3.21, 3.22, 3.23 and 3.24.

149 `paramchk=1`Model parameter checking selector.

150 `fullreinit=0`Model parameter full reinit selector.

Temperature Effects Parameters

151 `tnom (C)`Parameters measurement temperature. Default set by `options`.

152 `trise=0 C`Temperature rise from ambient.

153 `tlev=0`DC temperature selector.

154 `tlevc=0`AC temperature selector.

155 `eg=1.12452 V`Energy band gap.

156 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

157 `gap2=1108 C`Band gap temperature offset.

158 `diomod=1`Backward compatibility diode flag.

159 `kt1=-0.11 V`Temperature coefficient for threshold voltage.

160 `kt1l=0 v m`Temperature coefficient for threshold voltage.

161 `kt2=0.022`Temperature coefficient for threshold voltage.

162 `at=3.3e4 m/s`Temperature coefficient for `vsat`.

163 `ua1=4.31e-9 m/v`Temperature coefficient for ua.

164 `ub1=-7.61e-18` m^2/v^2
Temperature coefficient for ub.

165 `uc1=-5.5e-11` m/v^2
Temperature coefficient for uc. Default is -0.056 for mobmod=3.

166 `prt=0 Ω`Temperature coefficient for Rds.

167 `trs=0 1/C`Temperature parameter for source resistance.

168 `trd=0 1/C`Temperature parameter for drain resistance.

169 `ute=-1.5`Mobility temperature exponent.

170 `xti=3`Saturation current temperature exponent.

171 `pta=0 V/C`Junction potential temperature coefficient.

172 `tpb=0 V/C`Temperature coefficient for `pb`.

173 `ptp=0 V/C`Sidewall junction potential temperature coefficient.

174 `tpbsw=0 V/C`Temperature coefficient for `pbsw`.

175 `tpbswg=0 V/C`Temperature coefficient for `pbswg`.

176 `cta=0 1/C`Junction capacitance temperature coefficient.

177 `tcj=0 1/C`Temperature coefficient for `cj`.

178 `ctp=0 1/C`Sidewall junction capacitance temperature coefficient.

179 `tcjsw=0 1/C`Temperature coefficient for `cjsw`.

180 `tcjswg=0 1/C`Temperature coefficient for `cjswg`.

Noise Model Parameters

181 `noimod=1`Noise model selector.

182 `kf=0`Flicker (1/f) noise coefficient.

183 `af=1`Flicker (1/f) noise exponent.

184 `ef=1`Flicker (1/f) noise frequency exponent.

185 `noia=1e20`Oxide trap density coefficient. Default is 9.9e18 for pmos.

186 `noib=5e4`Oxide trap density coefficient. Default is 2.4e3 for pmos.

187 `noic=-1.4e-12`Oxide trap density coefficient. Default is 1.4e-8 for pmos.

188 `noid=2e14`flicker noise subthreshold-above threshold transition coefficient.

189 `wnoi=1e-5 m`Channel width at which noise parameters were extracted.

190 `em=4.1e7 V/m`Maximum electric field.

191 `flkmod=0`Flicker noise model (0 for Ids based model, 1 for gm based model).

192 `gamma=2.0/3.0`Thermal noise coefficient.

Auto Model Selector Parameters

193 `wmax=1 m`Maximum channel width for which the model is valid.

194 `wmin=0 m`Minimum channel width for which the model is valid.

195 `lmax=1 m`Maximum channel length for which the model is valid.

196 `lmin=0 m`Minimum channel length for which the model is valid.

Operating Region Warning Control Parameters

197 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

198 `imax=1 A`Maximum allowable current.

199 `jmax=1e8` A/m^2Maximum allowable current density.

200 `bvj=∞ V`Junction reverse breakdown voltage.

201 `vbox=1e9 tox V`Oxide breakdown voltage.

202 `warn=on`Parameter to turn warnings on and off.
Possible values are `off` or `on`.

Length Dependent Parameters

203 `xl=0 m`Length variation due to masking and etching.

Width dependent parameters

204 `xw=0 m`Width variation due to masking and etching.

DC-Mismatch Dependent Parameters

205 `mvtwl=0.0 v m`Threshold mismatch area dependence.

206 `mvtwl2=0.0 v m^1.5`
Threshold mismatch area square dependence.

207 `mvt0=0.0 V`Threshold mismatch intercept.

208 `mbewl=0.0 m`Beta mismatch area dependence.

209 `mbe0=0.0`Beta mismatch intercept.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

Both of these parameters have current density counterparts, `jmax` and `jmelt`, that you can specify if you want the absolute current values to depend on the device area.

Auto Model Selection

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 <= inst_length < lmax  and   wmin <= inst_width  < wmax
```

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)`Effective channel width.

2 `leff (m)`Effective channel length.

3 `rseff (Ω)`Effective source resistance.

4 `rdeff (Ω)`Effective drain resistance.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are n or p.

2 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

3 `reversed` Reverse mode indicator.
Possible values are `no` or `yes`.

4 `ids (A)` Resistive drain-to-source current.

5 `vgs (V)` Gate-source voltage.

6 `vds (V)` Drain-source voltage.

7 `vbs (V)` Bulk-source voltage.

8 `vth (V)` Threshold voltage.

9 `vdsat (V)` Drain-source saturation voltage.

10 `gm (S)` Common-source transconductance.

11 `gds (S)` Common-source output conductance.

12 `gmbs (S)` Body-transconductance.

13 `betaeff` (A/V^2) Effective `beta`.

14 `cjd (F)` Drain-bulk junction capacitance.

15 `cjs (F)` Source-bulk junction capacitance.

16 `cgg (F)` dQg_dVg.

17 `cgd (F)` dQg_dVd.

18 `cgs (F)` dQg_dVs.

19 `cgb (F)` dQg_dVbk.

20 `cdg (F)` dQd_dVg.

21 `cdd (F)` dQd_dVd.

22 `cds (F)` dQd_dVs.

23 `cdb (F)` dQd_dVb.

24 csg (F)dQs_dVg.

25 csd (F)dQs_dVd.

26 css (F)dQs_dVs.

27 csb (F)dQs_dVb.

28 cbg (F)dQb_dVg.

29 cbd (F)dQb_dVd.

30 cbs (F)dQb_dVs.

31 cbb (F)dQb_dVb.

32 ron (Ω)On-resistance.

33 id (A)Resistive drain current.

34 ibulk (A)Resistive bulk current.

35 pwr (W)Power at op point.

36 gmoverid (1/V)Gm/Ids.

Parameter Index

In the following index, I refers to instance parameters, M refers to the model parameters section, O refers to the output parameters section, and OP refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of M-35 means the 35th model parameter.

BSIM4 MOS Transistor (bsim4)

Description

BSIM4 is the version-4.20 of bsim model. It uses single-piece equations for all regions to improve the smoothness of the model characteristics. BSIM4 also allows the binning option like the approach used in bsim3 This option is provided for people who want to achieve the highest accuracy of the model. The binning equation is given by

P = P0 + Pl / Leff + Pw / Weff + Pp / (Leff * 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. BSIM4 transistors require that you use a model statement.

For more information on this model, please consult the University of California at Berkeley BSIM4 home page at

http://www-device.eecs.berkeley.edu/~bsim3/bsim4.html

This device is supported within altergroups.

Sample Instance Statement

```
m4 (0 2 1 1) pchmod w=2u l=0.8u as=250p ad=250p pd=168p ps=168p m=1
```

Sample Model Statement

```
model pchmod bsim4 type=p mobmod=0 capmod=2 version=4.20 toxe=3e-9 cdsc=2.58e-4
cdscb=0 cdscd=6.1e-8 cit=0 nfactor=1.1 xj=9e-8 vfb=0.76vsat=9.2e4 at=3.3e4 a0=1.1
ags=1.0e-20 a1=0 ngate=9e19 vth0=-0.42a1=0 a2=1 delta=0.014 pvag=1e-20 pclm=6.28e-
4 pdits=0.2 pditsl=2.3e6pditsd=0.23 fprout=0.2 pdiblcb=3.4e-8 pdiblc1=0.81
```

```
drout=0.56pdiblc2=9.84e-6 pscbe1=8.14e8 pscbe2=9.58e-07 lint=5e-9 wint=5e-
9dmcg=5e-6 dmci=5e-6 dmdg=5e-6 dmcgt=6e-7 dwj=4.5e-8 rsh=6cgso=7.43e-10
cgdo=7.43e-10 cgbo=2.56e-11 cgsl=1e-14 cgdl=1e-14ckappas=0.5 ckappad=0.5 noff=0.9
voffcv=0.02 acde=1 moin=15 xpart=0kt1l=0 kt2=2.2e-2 lpe0=5.75e-8 lpeb=2.3e-10
dvt0=2.89 dvt1=0.53dvt2=-3.2e-2 dvt0w=0 dvt1w=0 dvt2w=0 dvtp0=7.32e-7
dvtp1=0.12dsub=0.058 eta0=0.001 u0=4.19e-2 ua=8.7e-16 ub=3.06e-18 k1=0.33uc=4.6e-
13 ute=-1.5 ua1=4.31e-9 ub1=7.61e-18 uc1=-5.6e-11 k2=-1.87e-2rdsw=369.4 rdw=184.7
rsw=184.7 prwg=3.22e-8 prwb=6.8e-11 wr=1rdswmin=0 rdwmin=0 rswmin=0 prt=0 b0=-1e-
20 k3=80 k3b=0 w0=2.5e-6b1=0 keta=-0.047 alpha0=7.4e-2 alpha1=0.005 beta0=30
Instance Definition
```

Name d g s b ModelName parameter=value ...

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as` (m^2)Area of source diffusion.

4 `ad` (m^2)Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `m=1`Multiplicity factor (number of MOSFETs in parallel).

10 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

11 `trnqsmod`Transient NQS flag.

12 `acnqsmod`AC NQS flag.

13 `trise`Temperature rise from ambient.

14 `rgatemod`Rgate flag.

15 `rbodymod`Rbody flag.

16 `geomod`Geometry flag.

17 `rgeomod`Diffusion resistance and contact model flag.

18 `rbpb (Ω)`Resistance connected between bNode and bNode .

19 `rbpd (Ω)`Resistance connected between bNode and dbNode.

20 `rbps (Ω)`Resistance connected between bNode and sbNode.

21 `rbdb (Ω)`Resistance connected between dbNode and bNode.

22 `rbsb (Ω)`Resistance connected between sbNode and bNode.

23 `nf`Number of device fingers.

24 `min`Minimum number of device fingers.

Model Definition

```
model modelName bsim4 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

Threshold Voltage Parameters

2 `vtho (V)`Threshold voltage at zero body bias for long-channel devices. For enhancement-mode devices, vtho > 0 for n-channel and vth < 0 for p-channel. Default value is calculated from other model parameters.

3 `vfb=-1 V`Flat-band voltage.

4 `phin=0 V`Non-uniform vertical doping effect on surface potential.

5 `k1=0.53` $\sqrt{V}$ Body-effect coefficient.

6 `k2=-0.0186`Charge-sharing parameter.

7 `k3=80`Narrow width coefficient.

8 `k3b=0 1/V`Narrow width coefficient.

9 `w0=2.5e-6 m`Narrow width coefficient.

10 `lpe0=1.74e-7 m`Lateral non-uniform doping at Vbs=0.

11 `lpeb=0`Lateral non-uniform doping effect on K1.

12 `gamma1` ($\sqrt{V}$)Body-effect coefficient near the surface.

13 `gamma2` ($\sqrt{V}$)Body-effect coefficient in the bulk.

14 `vbx (V)`Threshold voltage transition body voltage.

15 `vbm=-3 V`Maximum applied body voltage.

16 `dvt0=2.2`First coefficient of short-channel effects.

17 `dvt1=0.53`Second coefficient of short-channel effects.

18 `dvt2=-0.032 1/V`Body-bias coefficient of short-channel effects.

19 `dvtp0=0 m`First Coef. of drain-induced Vth shift for long-channel pocket devices.

20 `dvtp1=0 1/V`Second Coef. of drain-induced Vth shift for long-channel pocket devices.

21 `dvt0w=0`First coefficient of narrow-width effects.

22 `dvt1w=5.3e6 1/m`Second coefficient of narrow-width effects.

23 `dvt2w=-0.032 1/V`Body-bias coefficient of narrow-width effects.

24 `a0=1`Nonuniform depletion width effect coefficient.

25 `b0=0 m`Bulk charge coefficient due to narrow width effect.

26 `b1=0 m`Bulk charge coefficient due to narrow width effect.

27 a1=0No-saturation coefficient.

28 a2=1No-saturation coefficient.

29 ags=0 F/m^2 VGate-bias dependence of Abulk.

30 keta=-0.047 1/VBody-bias coefficient for non-uniform depletion width effect.

Process Parameters

31 epsrox=3.9Gate dielectric constant.

32 toxe=3.0e-9 mElectrical gate oxide thickness.

33 toxp=toxe mElectrical gate oxide thickness.

34 dtox=0.0 mDifference between electrical and physical gate oxide thickness.

35 ndep=1.7e17 cm^{-3}Channel doping concentration.

36 nsd=1.0e20 cm^{-3}Source-drain doping concentration.

37 nsub=6e16 cm^{-3}Substrate doping concentration.

38 ngate=0 cm^{-3}Poly-gate doping concentration.

39 xj=0.15e-6 mSource/drain junction depth.

40 lint=0 mLateral diffusion for one side.

41 wint=0 mWidth reduction for one side.

42 ll=0Length dependence of delta L.

43 lln=1Length exponent of delta L.

44 lw=0Width dependence of delta L.

45 lwn=1Width exponent of delta L.

46 lwl=0Area dependence of delta L.

47 wl=0Length dependence of delta W.

48 `wln=1`Length exponent of delta W.

49 `ww=0`Width dependence of delta W.

50 `wwn=1`Width exponent of delta W.

51 `wwl=0`Area dependence of delta W.

52 `dwg=0 m/v`Gate-bias dependence of channel width.

53 `dwb=0 m/√v` Body-bias dependence of channel width.

54 `toxm=toxe m`Toxe at which parameters were extracted.

55 `xt=1.55e-7 m`Doping depth.

56 `binunit=1`Bin parameter unit selector. 1 for microns and 2 for meters.

Bias-Dependent RDS Parameters

57 `rdsmod=1`bias-dependent D/S model selector.

58 `rdsw=200`Zero bias LDD resistance per unit width for RDSMOD=0..

59 `rdswmin=0`LDD resistance per unit width at high Vgs and zero Vbs for RDSMOD=0.

60 `rdw=100`Zero bias LDD resistance per unit width for RDSMOD=1..

61 `rdwmin=0`LDD resistance per unit width at high Vgs and zero Vbs for RDSMOD=1.

62 `rsw=100`Zero bias LDD resistance per unit width for RDSMOD=1..

63 `rswmin=0`LDD resistance per unit width at high Vgs and zero Vbs for RDSMOD=1.

64 `prwb=0 1/√v` Body-effect coefficient for Rds.

65 `prwg=1 1/V`Gate-effect coefficient for Rds.

66 `wr=1`Width offset for parasitic resistance.

Mobility Parameters

67 `mobmod=0`Mobility model selector.

68 `u0=670 cm`^{2}`/V s`Low-field surface mobility at `tnom`. Default is 250 for PMOS.

69 `vsat=8e4 m/s`Carrier saturation velocity at `tnom`.

70 `ua=1.0e-9 m/v`First-order mobility reduction coefficient. Default is 1.0e-15 if Mobmod = 2.

71 `ub=1.0e-19 m`^{2}`/v`^{2}
Second-order mobility reduction coefficient.

72 `uc=-4.65e-11 m/v`^{2}
Body-bias dependence of mobility. Default is -0.0465 and unit is 1/V for mobmod=1.

73 `eu=1.67`Exponent for mobility degradation of mobmod=2. Default is 1.0 for Pmos.

Output Resistance Parameters

74 `drout=0.56`DIBL effect on output resistance coefficient.

75 `fprout=0.0 V/√m`
Effect of pocket implant on Rout degradation.

76 `pclm=1.3`Channel length modulation coefficient.

77 `pdiblc1=0.39`First coefficient of drain-induced barrier lowering.

78 `pdiblc2=8.6e-3`Second coefficient of drain-induced barrier lowering.

79 `pdiblcb=0 1/V`Body-effect coefficient for DIBL.

80 `pscbe1=4.24e8 V/m`First coefficient of substrate current body effect.

81 `pscbe2=1e-5 m/v`Second coefficient of substrate current body effect.

82 `pvag=0`Gate dependence of Early voltage.

83 `delta=0.01 V`Effective drain voltage smoothing parameter.

84 pdits=0.0 1/VEffect of pocket implant on Rout degradation.

85 pditsl=0.0 1/mChannel-length of drain-induced Vth shift on Rout.

86 pditsd=0.0 1/VChannel-length of drain-induced Vth shift on Rout.

Subthreshold Parameters

87 cdsc=2.4e-4 F/m^2Source/drain and channel coupling capacitance.

88 cdscb=0 F/m^2 VBody-bias dependence of cdsc.

89 cdscd=0 F/m^2 VDrain-bias dependence of cdsc.

90 nfactor=1Subthreshold swing coefficient.

91 cit=0 F/m^2Interface trap parameter for subthreshold swing.

92 voff=-0.08 VThreshold voltage offset.

93 voffl=0.0 VChannel-length dependence of Voff..

94 minv=0Vgsteff fitting parameter for moderate inversion condition..

95 dsub=droutDIBL effect in subthreshold region.

96 eta0=0.08DIBL coefficient subthreshold region.

97 etab=-0.07 1/VBody-bias dependence of et0.

Substrate Current Parameters

98 alpha0=0 m/vSubstrate current impact ionization coefficient.

99 alpha1=0 1/VSubstrate current impact ionization coefficient.

100 beta0=30 1/VSubstrate current impact ionization exponent.

Parasitic Resistance Parameters

101 rgatemodRgate flag.

102 `rsh=0 Ω/sqr`Source/drain diffusion sheet resistance.

103 `rshg=0.1 Ω/sqr`Gate electrode diffusion sheet resistance.

104 `dmcg=0 m`Distance from S/D contact center to the gate edge.

105 `dmci=dmcg m`Distance from S/D contact center to the isolation edge in the channel-length direction.

106 `dmdg=0 m`Distance from S/D contact center to the gate edge.

107 `dmcgt=0 m`DMCG of test structures.

108 `dwj=Dwc`Offset of the S/D junction width.

109 `xgw=0 m`Distance from the gate contact to the channel edge.

110 `xgl=0 m`Offset of the gate length due to variations in patterning.

111 `ngcon=1`Number of gate contacts.

112 `nf=1`Number of device fingers.

113 `min`Minimum number of device fingers.

114 `permod=1`Perimeter model selector.

115 `geomod`Geometry flag.

116 `rgeomod`Diffusion resistance and contact model flag.

117 `xw=0 m`Width variation due to masking and etching.

118 `xl=0 m`Length variation due to masking and etching.

119 `minr=0.001 Ω`Minimum source/drain resistance.

Gate-Induced Drain Leakage Parameters

120 `agidl=0 1/Ω`Pre-exponential coefficient for GIDL.

121 `bgidl=2.3e9 V`Exponential coefficient for GIDL.

122 `cgidl=0.5 V`^{3}Exponential coefficient for GIDL.

123 `egidl=0.8 V`Fitting parameter for band bending for GIDL.

Gate Tunneling Parameters

124 `igcmod=0`Gate-to-channel tunneling model selector.

125 `igbmod=1`Gate-to-substrate tunneling model selector.

126 `aigbacc=0.43` $\sqrt{F/g}$ `s/m`
Parameter for Igb in accumulation.

127 `bigbacc=0.054` $\sqrt{F/g}$ `/sm`
Parameter for Igb in accumulation.

128 `cigbacc=0.075 1/V`Parameter for Igb in accumulation.

129 `nigbacc=1`Parameter for Igb in accumulation.

130 `aigbinv=0.35` $\sqrt{F/g}$ `s/m`
Parameter for Igb in inversion.

131 `bigbinv=0.03` $\sqrt{F/g}$ `/sm`
Parameter for Igb in inversion.

132 `cigbinv=0.006 1/V`Parameter for Igb in inversion.

133 `eigbinv=1.1 V`Parameter for Igb in inversion.

134 `nigbinv=3`Parameter for Igb in inversion.

135 `aigc=0.054` $\sqrt{F/g}$ `s/m`
Parameter for Igcs and Igcd. Default value for Pmos is 0.31.

136 `bigc=0.054` $\sqrt{F/g}$ `/sm`
Parameter for Igcs and Igcd. Default value for Pmos is 0.024.

137 `cigc=0.075 1/V`Parameter for Igcs and Igcd. Default value for Pmos is 0.03.

138 `aigsd=0.43` $\sqrt{F/g}$ `s/m`
Parameter for Igs and Igd. Default value for Pmos is 0.31.

139 `bigsd=0.054 √F/g /sm`
Parameter for Igs and Igd. Default value for Pmos is 0.024.

140 `cigsd=0.075 1/V`Parameter for Igs and Igd. Default value for Pmos is 0.03.

141 `dlcig=Lint m`Source/drain overlap length for Igs and Igd.

142 `nigc=1.0`Source/drain overlap length for Igs and Igd.

143 `poxedge=1.0`Factor for the gate oxide thickness in source/drain overlap regions.

144 `pigcd=1.0`Vds dependence of Igcs and Igcd.

145 `ntox=1.0`Exponent for the gate oxide ratio.

146 `toxref=3.0e-9 m`Nominal gate oxide thickness for gate dielectric tunneling current model only.

Junction Diode Model Parameters

147 `diomod=1`Diode model selector.

148 `js=1.0e-4 A/m`^{2}Bottom junction reverse saturation current density..

149 `jss=1.0e-4 A/m`^{2}Bottom junction reverse saturation current density..

150 `jsd=Jss A/m`^{2}Bottom junction reverse saturation current density..

151 `jsws=0 A/m`Isolation-edge sidewall reverse saturation current density..

152 `jswd=Jsws A/m`Isolation-edge sidewall reverse saturation current density..

153 `jswgs=0 A/m`Gate-edge sidewall reverse saturation current density..

154 `jswgd=Jswgs A/m`Gate-edge sidewall reverse saturation current density.

155 `is=1e-14 A`Bulk junction reverse saturation current.

156 `n=1`Junction emission coefficient.

157 `njs=1`Bulk-Source junction emission coefficient.

158 `njd=Njs`Bulk-Source junction emission coefficient.

159 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`. Possible values are `no` or `yes`.

160 `imelt=`imaxA'`Explosion current.

161 `jmelt=`jmaxA/m'`^{2}
Explosion current density.

162 `ijthsrev=0.1 A`Limiting current in reverse bias region.

163 `ijthdrev=Ijthsrev A`
Limiting current in reverse bias region.

164 `ijthsfwd=0.1 A`Limiting current in forward bias region.

165 `ijthdfwd=Ijthsfwd A`
Limiting current in forward bias region.

166 `xjbvs=1.0`Limiting current in forward bias region.

167 `xjbvd=xjbvs`Limiting current in forward bias region.

Overlap Capacitance Parameters

168 `cgso (F/m)`Non LDD region source-gate overlap capacitance per unit channel width.

169 `cgdo (F/m)`Non LDD region drain-gate overlap capacitance per unit channel width.

170 `cgbo=2 Dwc Coxe F/m`
Non LDD region drain-gate overlap capacitance per unit channel width..

171 `meto=0 m`Metal overlap in fringing field.

172 `cgsl=0 F/m`Overlap capacitance between gate and lightly-doped source region.

173 `cgdl=Cgdl F/m`Overlap capacitance between gate and lightly-doped drain region.

174 `ckappas=0.6 V`Coefficient of bias-dependent overlap capacitance for the source side.

175 `ckappad=Ckappas V`Coefficient of bias-dependent overlap capacitance for the source side.

Junction Capacitance Model Parameters

176 `cj=5e-4 F/m`^{2}Zero-bias junction bottom capacitance density.

177 `cjs=5e-4 F/m`^{2}Zero bias bottom junction capacitance per unit area..

178 `cjd=5e-4 F/m`^{2}Zero bias bottom junction capacitance per unit area..

179 `mj=1/2`Bulk junction bottom grading coefficient.

180 `mjs=1/2`Bulk junction bottom grading coefficient.

181 `mjd=Mjs`Bulk junction bottom grading coefficient.

182 `pb=1 V`Bottom junction built-in potential.

183 `pbs=1 V`Bottom junction built-in potential.

184 `pbd=pbs V`Bottom junction built-in potential.

185 `fc=0.5`Forward-bias depletion capacitance threshold.

186 `cjsw=5e-10 F/m`Zero-bias junction sidewall capacitance density.

187 `cjsws=5e-10 F/m`Zero-bias junction sidewall capacitance density.

188 `cjswd=Cjsws F/m`Zero-bias junction sidewall capacitance density.

189 `mjsw=0.33`Isolation-edge sidewall junction capacitance grading coefficient..

190 `mjsws=0.33`Isolation-edge sidewall junction capacitance grading coefficient..

191 `mjswd=Mjsws`Isolation-edge sidewall junction capacitance grading coefficient..

192 `pbsw=1 V`Isolation-edge sidewall junction built-in potential.

193 `pbsws=1 V`Isolation-edge sidewall junction built-in potential.

194 `pbswd=pbsws V`Isolation-edge sidewall junction built-in potential.

195 `cjswg=cjswg F/m`Zero-bias gate-side junction capacitance density.

196 `cjswgs=cjsws F/m`Zero-bias gate-side junction capacitance density.

197 `cjswgd=cjswgs F/m`Zero-bias gate-side junction capacitance density.

198 `mjswg=mjsw`Gate-edge sidewall junction grading coefficient.

199 `mjswgs=mjsws`Gate-edge sidewall junction grading coefficient.

200 `mjswgd=mjsws`Gate-edge sidewall junction grading coefficient.

201 `pbswg=pbsw V`Gate-edge sidewall junction built-in potential.

202 `pbswgs=pbsws V`Gate-edge sidewall junction built-in potential.

203 `pbswgd=pbsws V`Gate-edge sidewall junction built-in potential.

204 `fcsw=0.5`Side-wall forward-bias depletion capacitance threshold.

205 `bvs=10.0 V`Breakdown voltage.

206 `bvd=Bvs V`Breakdown voltage.

Charge Model Selection Parameters

207 `capmod=2`Intrinsic charge model.

208 `trnqsmod=0`Transient Non-quasi static model selector. Set to 1 to turn on nqs.

209 `acnqsmod=0`Ac Non-quasi static model selector. Set to 1 to turn on nqs.

210 `dwc=wint m`Delta W for capacitance model.

211 `dlc=Lint m`Delta L for capacitance model.

212 `clc=1e-7 m`Constant term for the short channel model..

213 `cle=0.6`Intrinsic capacitance fitting parameter.

214 `cf (F/m)`Coefficient of bias-dependent overlap capacitance for the source side.

215 `vfbcv=-1`Flat-band voltage for capmod=0.

216 `acde=1 m/v`Exponential coefficient for charge thickness in CAPMOD=2 for accumulation and depletion regions.

217 `moin=15 1/V` Exponential coefficient for charge thickness for accumulation and depletion regions.

218 `noff=1` Transition parameter.

219 `voffcv=0 V` CV parameter in VgsteffCV for weak to strong inversion..

220 `xpart=0` Charge partition number. Use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

221 `llc=ll` Length dependence of delta L for CV.

222 `lwc=lw` Width dependence of delta L for CV.

223 `lwlc=lwl` Area dependence of delta L for CV.

224 `wlc=wl` Length dependence of delta W for CV.

225 `wwc=ww` Width dependence of delta W for CV.

226 `wwlc=wwl` Area dependence of delta W for CV.

Default for Instance Parameters

227 `w=5e-6 m` Default channel width.

228 `l=5e-6 m` Default channel length.

229 `as=0 m`^{2} Default area of source diffusion.

230 `ad=0 m`^{2} Default area of drain diffusion.

231 `ps=0 m` Default perimeter of source diffusion.

232 `pd=0 m` Default perimeter of drain diffusion.

233 `nrd=0 m/m` Default number of squares of drain diffusion.

234 `nrs=0 m/m` Default number of squares of source diffusion.

235 `version=4.20` Model version selector.

236 `level=14` Model level selector for spice compatibility.

237 `paramchk=1`Model parameter checking selector.

238 `fullreinit=0`Model parameter full reinit selector.

Temperature Effects Parameters

239 `tnom (C)`Parameters measurement temperature. Default set by `options`.

240 `trise=0 C`Temperature rise from ambient.

241 `tlev=0`DC temperature selector.

242 `tlevc=0`AC temperature selector.

243 `eg=1.12452 V`Energy band gap.

244 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

245 `gap2=1108 C`Band gap temperature offset.

246 `kt1=-0.11 V`Temperature coefficient for threshold voltage.

247 `kt1l=0 v m`Temperature coefficient for threshold voltage.

248 `kt2=0.022`Temperature coefficient for threshold voltage.

249 `at=3.3e4 m/s`Temperature coefficient for `vsat`.

250 `ua1=4.31e-9 m/v`Temperature coefficient for `ua`.

251 `ub1=-7.61e-18` m^2/v^2
Temperature coefficient for `ub`.

252 `uc1=-5.5e-11` m/v^2
Temperature coefficient for `uc`. Default is -0.056 for mobmod=3.

253 `prt=0 Ω m`Temperature coefficient for Rds.

254 `ute=-1.5`Mobility temperature exponent.

255 `xti=3`Saturation current temperature exponent.

256 `xtis=3`Bulk-Source junction saturation current temperature exponent.

257 xtid=3Bulk-Source junction saturation current temperature exponent.

258 pta=0 V/CTemperature coefficient for pb.

259 tpb=0 V/CTemperature coefficient for pb.

260 ptp=0 V/CTemperature coefficient for pbsw.

261 tpbsw=0 V/CTemperature coefficient for pbsw.

262 tpbswg=0 V/CTemperature coefficient for pbswg.

263 cta=0 1/CTemperature coefficient for cj.

264 tcj=0 1/CTemperature coefficient for cj.

265 ctp=0 1/CTemperature coefficient for cjsw.

266 tcjsw=0 1/CTemperature coefficient for cjsw.

267 tcjswg=0 1/CTemperature coefficient for cjswg.

Noise Model Parameters

268 fnoimod=1Flicker noise model selector.

269 tnoimod=0Thermal noise model selector.

270 kf=0Flicker noise exponent.

271 af=1Flicker noise exponent.

272 ef=1Flicker noise frequency exponent.

273 noia=6.25e41Flicker noise parameter B. Default is 6.188e40 for pmos.

274 noib=3.125e26Flicker noise parameter C. Default is 1.5e25 for pmos.

275 noic=8.75e9Flicker noise parameter C.

276 wnoi=1e-5 mChannel width at which noise parameters were extracted.

277 em=4.1e7 V/mSaturation field.

278 `flkmod`Flicker Noise Model.

279 `ntnoi=1`Noise factor for short-channel devices for TNOIMOD=0 only.

280 `tnoia=1.5`Coefficient of channel-length dependence of total channel thermal noise.

281 `tnoib=3.5`Coefficient of channel-length dependence of total channel thermal noise.

Substrate Network Parameters

282 `rbodymod`Rbody flag.

283 `xrcrg1=12`Parameter for distributed channel-resistance effect for both intrinsic-input resistance and charge-deficit NQS models.

284 `xrcrg2=1`Parameter to account for the excess channel diffusion resistance for both intrinsic-input resistance and charge-deficit NQS models.

285 `rbpb (Ω)`Resistance connected between bNode and bNode .

286 `rbpd (Ω)`Resistance connected between bNode and dbNode.

287 `rbps (Ω)`Resistance connected between bNode and sbNode.

288 `rbdb (Ω)`Resistance connected between dbNode and bNode.

289 `rbsb (Ω)`Resistance connected between sbNode and bNode.

290 `gbmin=1.0e-12 1/Ω`
Conductance in parallel with each of the five substrate resistances to avoid potential numerical instability due to unreasonably too large a substrate resistance.

Auto Model Selector Parameters

291 `wmax=1 m`Maximum channel width for which the model is valid.

292 `wmin=0 m`Minimum channel width for which the model is valid.

293 `lmax=1 m`Maximum channel length for which the model is valid.

294 `lmin=0 m`Minimum channel length for which the model is valid.

Operating Region Warning Control Parameters

295 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

296 `imax=1 A`Maximum allowable current.

297 `jmax=1e8 A/m`^{2}Maximum allowable current density.

298 `bvj=∞ V`Junction reverse breakdown voltage.

299 `vbox=1e9 toxe V`Oxide breakdown voltage.

300 `warn=on`Parameter to turn warnings on and off.
Possible values are `off` or `on`.

DC-Mismatch Dependent Parameters

301 `mvtwl=0.0 v m`Threshold mismatch area dependence.

302 `mvtwl2=0.0 v m^1.5`
Threshold mismatch area square dependence.

303 `mvt0=0.0 V`Threshold mismatch intercept.

304 `mbewl=0.0 m`Beta mismatch area dependence.

305 `mbe0=0.0`Beta mismatch intercept.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

Both of these parameters have current density counterparts, `jmax` and `jmelt`, that you can specify if you want the absolute current values to depend on the device area.

Auto Model Selection

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 <= inst_length < lmax  and   wmin <= inst_width  < wmax
```

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)`Effective channel width.

2 `leff (m)`Effective channel length.

3 `rgbi (Ω)`Gate bias-independent resistance.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are n or p.

2 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

3 `reversed`Reverse mode indicator.
Possible values are `no` or `yes`.

4 `ids (A)`Resistive drain-to-source current.

5 `vgs (V)`Gate-source voltage.

6 `vds (V)`Drain-source voltage.

7 `vbs (V)`Bulk-source voltage.

8 `vth (V)`Threshold voltage.

9 `vdsat (V)`Drain-source saturation voltage.

10 `gm (S)`Common-source transconductance.

11 `gds (S)`Common-source output conductance.

12 `gmbs (S)`Body-transconductance.

13 `betaeff` (A/V^2)Effective `beta`.

14 `cjd (F)`Drain-bulk junction capacitance.

15 `cjs (F)`Source-bulk junction capacitance.

16 `cgg (F)`dQg_dVg.

17 `cgd (F)`dQg_dVd.

18 `cgs (F)`dQg_dVs.

19 `cgb (F)`dQg_dVbk.

20 `cdg (F)`dQd_dVg.

21 `cdd (F)`dQd_dVd.

22 `cds (F)`dQd_dVs.

23 `cdb (F)`dQd_dVb.

24 `csg (F)`dQs_dVg.

25 `csd (F)`dQs_dVd.

26 `css (F)`dQs_dVs.

27 `csb (F)`dQs_dVb.

28 `cbg (F)`dQb_dVg.

29 `cbd (F)`dQb_dVd.

30 `cbs (F)`dQb_dVs.

31 `cbb (F)`dQb_dVb.

32 `ron (Ω)`On-resistance.

33 `id (A)`Resistive drain current.

34 `ibulk (A)`Resistive bulk current.

35 `pwr (W)`Power at op point.

36 `gmoverid (1/V)`Gm/Ids.

37 `rdeff (Ω)`Effective drain resistance.

38 `rseff (Ω)`Effective source resistance.

39 `rgbd (Ω)`Gate bias-dependent resistance.

40 `igidl (A)`Gate-to-body tunneling current.

41 `igd (A)`Gate-to-drain tunneling current.

42 `igs (A)`Gate-to-source tunneling current.

43 `igb (A)`Gate-to-bulk tunneling current.

44 `igcs (A)`Gate-to-channel (source side) tunneling current.

45 `igcd (A)`Gate-to-channel (drain side) tunneling current.

46 `gbs (S)`bulk-source diode conductance.

47 `gbd (S)`bulk-drain diode conductance.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

BTA SOI Transistor (btasoi)

Description

BTASOI is an SOI model developed by BTA Technology based on bsim3v3. It is a new, simple and compact SOI model that can accommodate both the fully-depleted, FD and partially-depleted, PD modes, adopting the transition voltage, Vtr, for the definition of body condition. It can also simulate the special characteristics of SOI devices such as kink effect and reduction of saturation current due to self-heating. Simulation results with this model are in excellent agreement with the experimental data for 0.25um SIMOX technology. BTASOI devices require that you use a model statement.

If you want to get more information about this model, please contact BTA Technology at http://www.btat.com

This device is supported within altergroups.

Sample Instance Statement

```
m5 (1 2 0 0) nchmod l=1.5u w=100u as=450p ad=450p pd=209u ps=209u  m=1
```

Sample Model Statement

```
model nchmod btasoi type=n b3v3mod=no version=3.1 vtho=0.62 k1=0.672 k2=0.038
nlx=1.14e-7 dvt0=4.1 a0=1.08 nch=2.65e17 u0=4.01e-2 a1=0 a2=1 ags=9.8e-4
vsat=1.77e5
```

Instance Definition

```
Name  d  g  s  [bg]  [b] ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as` (m^2)Area of source diffusion.

4 `ad` (m^2)Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 m=1Multiplicity factor (number of MOSFETs in parallel).

10 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, or `subth`.

11 `trise`Temperature rise from ambient.

12 `rbody` (Ω)Body resistance.

Model Definition

```
model modelName btasoi parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

2 `b3v3mod=no`B3v3 compatible flag.
Possible values are no or `yes`.

3 `version=3.1`Model version selector.

4 `btasoiver=1.0`BTASOI Model version selector.

Threshold Voltage Parameters

5 `vtho (V)`Threshold voltage at zero body bias for long-channel devices. For enhancement-mode devices, vtho > 0 for n-channel and vth < 0 for p-channel. Default value is calculated from other model parameters.

6 `k1=0.5 √V` Body-effect coefficient.

7 `k2=-0.0186`Charge-sharing parameter.

8 `k3=80`Narrow width coefficient.

9 `k3b=0 1/V`Narrow width coefficient.

10 `w0=2.5e-6 m`Narrow width coefficient.

11 `nlx=1.74e-7 m`Lateral nonuniform doping coefficient.

12 `gamma1 (√V )`Body-effect coefficient near the surface.

13 `gamma2 (√V )`Body-effect coefficient in the bulk.

14 `vbx (V)`Threshold voltage transition body voltage.

15 `vbm=-3 V`Maximum applied body voltage.

16 `dvt0=2.2`First coefficient of short-channel effects.

17 `dvt1=0.53`Second coefficient of short-channel effects.

18 `dvt2=-0.032 1/V`Body-bias coefficient of short-channel effects.

19 `dvt0w=0`First coefficient of narrow-width effects.

20 `dvt1w=5.3e6`Second coefficient of narrow-width effects.

21 `dvt2w=-0.032 1/V`Body-bias coefficient of narrow-width effects.

22 `a0=1`Nonuniform depletion width effect coefficient.

23 `b0=0 m`Bulk charge coefficient due to narrow width effect.

24 `b1=0 m`Bulk charge coefficient due to narrow width effect.

25 `a1=0`No-saturation coefficient.

26 `a2=1`No-saturation coefficient.

27 `ags=0` F/m^2 `V`Gate-bias dependence of Abulk.

28 `keta=-0.047 1/V`Body-bias coefficient for non-uniform depletion width effect.

Process Parameters

29 `nsub=6e16` cm^{-3}Substrate doping concentration.

30 `nch=1.7e17` cm^{-3}Peak channel doping concentration.

31 `ngate` (cm^{-3})Poly-gate doping concentration.

32 `xj=0.15e-6 m`Source/drain junction depth.

33 `lint=0 m`Lateral diffusion for one side.

34 `wint=0 m`Width reduction for one side.

35 `ll=0 m`Length dependence of delta L.

36 `lln=1`Length exponent of delta L.

37 `lw=0 m`Width dependence of delta L.

38 `lwn=1`Width exponent of delta L.

39 `lwl=0 m`^{2}Area dependence of delta L.

40 `wl=0 m`Length dependence of delta W.

41 `wln=1`Length exponent of delta W.

42 `ww=0 m`Width dependence of delta W.

43 `wwn=1`Width exponent of delta W.

44 `wwl=0 m`^{2}Area dependence of delta W.

45 `dwg=0 m/v`Gate-bias dependence of channel width.

46 `dwb=0 m/√v` Body-bias dependence of channel width.

47 `tox=1.5e-8 m`Gate oxide thickness.

48 `tbox=4e-7 m`Buried oxide thickness.

49 `tsi=8e-8 m`Silicon film thickness.

50 `xt=1.55e-7 m`Doping depth.

51 `rdsw=0 Ω µm`Width dependence of drain-source resistance.

52 `prwb=0 1/√v` Body-effect coefficient for Rds.

53 `prwg=0 1/V`Gate-effect coefficient for Rds.

54 `wr=1`Width offset for parasitic resistance.

55 `xl=0 m`Length variation due to masking and etching.

56 `xw=0 m`Width variation due to masking and etching.

57 `binunit=1`Bin parameter unit selector. 1 for microns and 2 for meters.

Mobility Parameters

58 `mobmod=1`Mobility model selector.

59 u0=670 cm^2/V sLow-field surface mobility at tnom. Default is 250 for PMOS.

60 vsat=8e4 m/sCarrier saturation velocity at tnom.

61 ua=2.25e-9 m/vFirst-order mobility reduction coefficient.

62 ub=5.87e-19 m^2/v^2
Second-order mobility reduction coefficient.

63 uc=-4.65e-11 m/v^2
Body-bias dependence of mobility. Default is -0.046 and unit is 1/V for mobmod=3.

Output Resistance Parameters

64 drout=0.56DIBL effect on output resistance coefficient.

65 pclm=1.3Channel length modulation coefficient.

66 pdiblc1=0.39First coefficient of drain-induced barrier lowering.

67 pdiblc2=8.6e-3Second coefficient of drain-induced barrier lowering.

68 pdiblcb=0 1/VBody-effect coefficient for DIBL.

69 pscbe1=4.24e8 V/mFirst coefficient of substrate current body effect.

70 pscbe2=1e-5 m/vSecond coefficient of substrate current body effect.

71 pvag=0Gate dependence of Early voltage.

72 delta=0.01 VEffective drain voltage smoothing parameter.

Subthreshold Parameters

73 cdsc=2.4e-4 F/m^{2}Source/drain and channel coupling capacitance.

74 cdscb=0 F/m^2 VBody-bias dependence of cdsc.

75 cdscd=0 F/m^2 VDrain-bias dependence of cdsc.

76 nfactor=1Subthreshold swing coefficient.

77 `cit=0 F`Interface trap parameter for subthreshold swing.

78 `voff=-0.08 V`Threshold voltage offset.

79 `dsub=drout`DIBL effect in subthreshold region.

80 `eta0=0.08`DIBL coefficient subthreshold region.

81 `etab=-0.07 1/V`Body-bias dependence of `et0`.

Substrate Current Parameters

82 `alpha0=0 m/v`Substrate current impact ionization coefficient.

83 `beta0=30 1/V`Substrate current impact ionization exponent.

Parasitic Resistance Parameters

84 `rsh=0 Ω/sqr`Source/drain diffusion sheet resistance.

85 `rs=0 Ω`Source resistance.

86 `rd=0 Ω`Drain resistance.

87 `rsc=0 Ω`Source contact resistance.

88 `rdc=0 Ω`Drain contact resistance.

89 `rss=0 Ω m`Scalable source resistance.

90 `rdd=0 Ω m`Scalable drain resistance.

91 `hdif=0 m`Length of heavily doped diffusion.

92 `ldif=0 m`Lateral diffusion beyond the gate.

93 `minr=0.1 Ω`Minimum source/drain resistance.

Junction Diode Model Parameters

94 `js` (A/m^2)Bulk junction reverse saturation current density.

95 `jsw=0 A/m`Sidewall junction reverse saturation current density.

96 `is=1e-14 A`Bulk junction reverse saturation current.

97 `n=1`Junction emission coefficient.

98 `dskip=yes`Use simple piecewise linear model for diode currents below 0.1*`iabstol`. Possible values are `no` or `yes`.

99 `imelt=`imaxA'`Explosion current.

100 `imelt1=imax A/m`Explosion current density for `is1`.

101 `imelt2=imax A/m`Explosion current density for `is2`.

102 `imelt3=imax A/m`Explosion current density for `is3`.

Overlap Capacitance Parameters

103 `cgso (F/m)`Gate-source overlap capacitance.

104 `cgdo (F/m)`Gate-drain overlap capacitance.

105 `cgbo=2 Dwc Cox F/m`
Gate-bulk overlap capacitance. The default value is 0 if version=3.0.

106 `meto=0 m`Metal overlap in fringing field.

107 `cgsl=0 F/m`Gate-source overlap capacitance in LDD region.

108 `cgdl=0 F/m`Gate-drain overlap capacitance in LDD region.

109 `ckappa=0.6`Overlap capacitance fitting parameter.

Junction Capacitance Model Parameters

110 `cbs=0 F`Bulk-source zero-bias junction capacitance.

111 `cbd=0 F`Bulk-drain zero-bias junction capacitance.

112 `cj=5e-4` F/m^2Zero-bias junction bottom capacitance density.

113 `mj=1/2`Bulk junction bottom grading coefficient.

114 `pb=1 V`Bulk junction built-in potential.

115 `fc=0.5`Forward-bias depletion capacitance threshold.

116 `cjsw=5e-10 F/m`Zero-bias junction sidewall capacitance density.

117 `mjsw=0.33`Bulk junction sidewall grading coefficient.

118 `pbsw=1 V`Side-wall junction built-in potential.

119 `cjswg=cjsw F/m`Zero-bias gate-side junction capacitance density.

120 `mjswg=mjsw`Gate-side junction grading coefficient.

121 `pbswg=pbsw V`Gate-side junction built-in potential.

122 `fcsw=fc`Side-wall forward-bias depletion capacitance threshold.

123 `tau=0 s`Transit time.

Charge Model Selection Parameters

124 `capmod=2`Intrinsic charge model.

125 `dwc=wint m`Delta W for capacitance model.

126 `dlc=lint m`Delta L for capacitance model.

127 `clc=1e-7 m`Intrinsic capacitance fitting parameter.

128 `cle=0.6`Intrinsic capacitance fitting parameter.

129 `cf (F/m)`Fringe capacitance parameter.

130 `vfbcv=-1`Flat-band voltage for capmod=0.

131 `xpart=0`Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

Default Instance Parameters

132 `w=5e-6 m`Default channel width.

133 `l=5e-6 m`Default channel length.

134 `as=0 m`^{2}Default area of source diffusion.

135 `ad=0 m`^{2}Default area of drain diffusion.

136 `ps=0 m`Default perimeter of source diffusion.

137 `pd=0 m`Default perimeter of drain diffusion.

138 `nrd=0 m/m`Default number of squares of drain diffusion.

139 `nrs=0 m/m`Default number of squares of source diffusion.

Temperature Effects Parameters

140 `tnom (C)`Parameters measurement temperature. Default set by `options`.

141 `tmax=500 C`Maximum device temperature above ambient.

142 `trise=0 C`Temperature rise from ambient.

143 `selft=0`Self heating option.

144 `tlev=0`DC temperature selector.

145 `tlevc=0`AC temperature selector.

146 `eg=1.12452 V`Energy band gap.

147 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

148 `gap2=1108 C`Band gap temperature offset.

149 `kt1=-0.11 V`Temperature coefficient for threshold voltage.

150 `kt1l=0 v m`Temperature coefficient for threshold voltage.

151 `kt2=0.022`Temperature coefficient for threshold voltage.

152 `at=3.3e4 m/s`Temperature coefficient for `vsat`.

153 `ua1=4.31e-9 m/v`Temperature coefficient for `ua`.

154 `ub1=-7.61e-18` m^2/v^2
Temperature coefficient for `ub`.

155 `uc1=-5.5e-11` m/v^2
Temperature coefficient for `uc`. Default is -0.056 for mobmod=3.

156 `prt=0 Ω`Temperature coefficient for Rds.

157 `trs=0 1/C`Temperature parameter for source resistance.

158 `trd=0 1/C`Temperature parameter for drain resistance.

159 `ute=-1.5`Mobility temperature exponent.

160 `dt1=0`First temperature coefficient for tau.

161 `dt2=0`Second temperature coefficient for tau.

162 `xti=3`Saturation current temperature exponent.

163 `xti1=3`Saturation current temperature exponent.

164 `xti2=xti1`Saturation current temperature exponent.

165 `xti3=xti1`Saturation current temperature exponent.

166 `ptc=0 V/C`Surface potential temperature coefficient.

167 `pta=0 V/C`Junction potential temperature coefficient.

168 `ptp=0 V/C`Sidewall junction potential temperature coefficient.

169 `cta=0 1/C`Junction capacitance temperature coefficient.

170 `ctp=0 1/C`Sidewall junction capacitance temperature coefficient.

Noise Model Parameters

171 `noimod=1`Noise model selector.

172 `kf=0`Flicker (1/f) noise coefficient.

173 `af=1`Flicker (1/f) noise exponent.

174 `ef=1`Flicker (1/f) noise frequency exponent.

175 `noia=1e20`Oxide trap density coefficient. Default is 9.9e18 for pmos.

176 `noib=5e4`Oxide trap density coefficient. Default is 2.4e3 for pmos.

177 `noic=-1.4e-12`Oxide trap density coefficient. Default is 1.4e-8 for pmos.

178 `em=4.1e7 V/m`Maximum electric field.

Auto Model Selector Parameters

179 `wmax=1 m`Maximum channel width for which the model is valid.

180 `wmin=0 m`Minimum channel width for which the model is valid.

181 `lmax=1 m`Maximum channel length for which the model is valid.

182 `lmin=0 m`Minimum channel length for which the model is valid.

Operating Region Warning Control Parameters

183 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

184 `imax=1 A`Maximum allowable current.

185 `bvj=∞ V`Junction reverse breakdown voltage.

186 `vbox=1e9 tox V`Oxide breakdown voltage.

187 `warn=on`Parameter to turn warnings on and off.
Possible values are `off` or `on`.

SOI Specific Parameters

188 `vbtho=10 V`Back-gate threshold voltage.

189 `vtr0=0.3 V`Long-channel transition body voltage at Vds=0.

190 `knk=0.01`Vtr smoothing factor.

191 `dice=0`Drain-induced charge-sharing parameter.

192 `dvtrd=0 V`Vtr dependence on Vds.

193 `dvtrg=1 V`Vtr dependence on Vgs.

194 `dvtrbg=1.0`Smoothing factor for back-gate bias.

195 `a0bg=0`Back-gate saturation region coefficient.

196 `dbg=1`Diode fully depletion adjustment factor.

197 `dvtr=1.0`Diode back-gate dependence factor.

198 `vbgf=0.0`Flat-band voltage for back-gate.

199 `rth0=0 Ω`Self-heating thermal resistance.

200 `cth0=1 F`Self-heating thermal capacitance.

201 `l1=0`Vgs dependence of characteristic length.

202 `aii=0`First parameter for critical field.

203 `bii=0`Second parameter for critical field.

204 `cii=0`Gate dependence of critical field.

205 `dii=0`Body dependence of critical field.

206 `ndiode=1`Diode non-ideality factor.

207 `nt=1`Reverse tunneling non-ideality factor.

208 `is1=1e-16 A`First diode parameter.

209 `is2=0 A`Second diode parameter.

210 `is3=0 A`Tunneling diode parameter.

211 `edl=2e-6 m`Electron diffusion length.

212 `kb=0 m`Parasitic bipolor base width.

213 `delacc=0.02 V`Capacitance smoothing parameter in accumulation region.

214 `delr=0.01 V`Vbs smoothing parameter for C-V.

215 `dqsq=8e-3 V`Vtr smoothing parameter for C-V.

216 `a0cv=0.1`A0 for C-V calculation.

217 `qgvd0=1`Cgd fitting parameter.

LThe `jmelt` parameter is used to aid convergence and prevent numerical overflow. The junction characteristics of the FET are accurately modeled for current (density) up to `jmelt`. For current density above `jmelt`, the junction is modeled as a linear resistor and a warning is printed.

Auto Model Selection

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 <= inst_length < lmax  and   wmin <= inst_width  < wmax
```

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)`Effective channel width.

2 `leff (m)`Effective channel length.

3 `rtheff (Ω)`Effective thermal resistance.

4 `Ctheff (F)`Effective thermal capacitance.

5 `rseff (Ω)`Effective source resistance.

6 `rdeff (Ω)`Effective drain resistance.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are n or p.

2 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, or `subth`.

3 `reversed`Reverse mode indicator.
Possible values are no or `yes`.

4 `vgs (V)`Gate-source voltage.

5 `vds (V)`Drain-source voltage.

6 `vbs (V)`Bulk-source voltage.

7 `ids (A)`Resistive drain-to-source current.

8 `isub (A)`Resistive substrate current.

9 `ibd (A)`Resistive bulk-to-drain junction current.

10 `ibs (A)`Resistive bulk-to-source junction current.

11 `vth (V)`Threshold voltage.

12 `vdsat (V)`Drain-source saturation voltage.

13 `gm (S)`Common-source transconductance.

14 `gds (S)`Common-source output conductance.

15 `gmbs (S)`Body-transconductance.

16 `ueff` (cm^2/V s)Effective mobility.

17 `betaeff` (A/V^2)Effective `beta`.

18 `cjd (F)`Drain-bulk junction capacitance.

19 `cjs (F)`Source-bulk junction capacitance.

20 `cgg (F)`dQg_dVg.

21 `cgd (F)`dQg_dVd.

22 `cgs (F)`dQg_dVs.

23 `cgb (F)`dQg_dVbk .

24 `cdg (F)`dQd_dVg.

25 `cdd (F)`dQd_dVd.

26 `cds (F)`dQd_dVs.

27 `cdb (F)`dQd_dVb.

28 `csg (F)`dQs_dVg.

29 `csd (F)`dQs_dVd.

30 `css (F)`dQs_dVs.

31 `csb (F)`dQs_dVb.

32 `cbg (F)`dQb_dVg.

33 `cbd (F)`dQb_dVd.

34 `cbs (F)`dQb_dVs.

35 `cbb (F)`dQb_dVb.

36 `ron (Ω)`On-resistance.

37 `id (A)`Total resistive drain current.

38 `is (A)`Total resistive source current.

39 `ib (A)`Total resistive bulk current.

40 `pwr (W)`Power at op point.

41 `gmoverid (1/V)`Gm/Ids.

42 `tdev (C)`Temperature rise from ambient.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Two Terminal Capacitor (capacitor)

Description

You can assign the capacitance or let Spectre 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

C(inst) = Cj*(L - 2*etch)*(W - 2*etch) + 2*Cjsw*(W + L - 4*etch)

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)*(1 + c1*V + c2*V^2 + ...)

or

Q(V) = C(inst)*V*(1 + 1/2*c1*V + 1/3*c2*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)*[1 + tc1*(T - tnom) + tc2*(T - tnom)^2].

where

T = trise(inst) + temp

if trise(inst) is given, and

T = trise(model) + temp

if trise(inst) is not given.

Sample Instance Statement

Without model:

c2 (1 0) capacitor c=2.5u w=2u l=2.5u tc1=1e-8

With model:

c2 (1 0) proc_cap c=2.5u w=2u l=2.5u tc1=1e-8

Sample Model Statement

```
model proc_cap capacitor c=2u tc1=1.2e-8 tnom=25 w=4u l=4u cjsw=2.4e-10
```

Instance Definition

```
Name  1  2 ModelName parameter=value ...
Name  1  2 capacitor parameter=value ...
```

Instance Parameters

1 `c (F)`Capacitance.

2 `w (m)`Capacitor width.

3 `l (m)`Capacitor length.

4 `m=1`Multiplicity factor.

5 `scale=1`Scale factor.

6 `trise (C)`Temperature rise from ambient.

7 `tc1 (1/C)`Linear temperature coefficient.

8 `tc2` (C^{-2})Quadratic temperature coefficient.

9 `ic (V)`Initial condition.

The instance parameter `scale`, if specified, overrides the value given by the `option` parameter `scale`. The `w` and `l` parameters are scaled by the resulting `scale`, and the `option` parameter `scalem`. The values of `w` and `l` printed by Spectre are those given in the input file, and these values might not have the correct units if the scaling factors are not unity. The actual capacitor dimensions are stored in the output parameters. You can obtain these dimensions with the info statement.

Model Definition

```
model modelName capacitor parameter=value ...
```

Model Parameters

1 `c=0 F`Default capacitance.

2 `tc1=0 1/C`Linear temperature coefficient.

3 `tc2=0` C^{-2}Quadratic temperature coefficient.

4 `trise=0 C`Default `trise` value for instance.

5 `tnom (C)`Parameters measurement temperature. Default set by `options`.

6 `w=0 m`Default capacitor width.

7 `l=0 m`Default capacitor length.

8 `etch=0 m`Narrowing due to side etching.

9 `cj=0` F/m^2Bottom capacitance density.

10 `cjsw=0 F/m`Sidewall capacitance.

11 `scalec=1`Capacitance scaling factor.

12 `coeffs=[...]`Vector of polynomial capacitance coefficients.

13 `rforce=1 Ω`Resistance used when forcing initial conditions.

Output Parameters

1 `leff (m)`Effective capacitor length.

2 `weff (m)`Effective capacitor width.

3 `ceff (F)`Effective capacitance.

Operating-Point Parameters

1 `cap (F)`Capacitance at operating point.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Linear Current Controlled Current Source (cccs)

Description

A current-controlled source senses the current with a probe device. A valid probe is a component instance in the circuit that naturally computes current. For example, probes can be voltage sources (independent or controlled), inductors, transmission lines, microstrip lines, N-ports, and transformers. If the probe device computes more than one current (such as transmission lines, microstrip lines, and N-ports), the index of the probe port through which the controlling current flows needs to be specified. Positive current exits the source node and enters the sink node of the controlled source.

This device is supported within altergroups.

Sample Instance Statement

```
vcs (pos gnd) cccs gain=2.5 probe=v1 m=1 //Note that v1 is an instance of a voltage
source
```

Instance Definition

```
Name  sink  src cccs parameter=value ...
```

Instance Parameters

1 `gain=0 A/A`Current gain.

2 `m=1`Multiplicity factor.

3 `probe`Device through which the controlling current flows.

4 `port=0`Index of the probe port through which the controlling current flows.

Operating-Point Parameters

1 `i (A)`Input current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Linear Current Controlled Voltage Source (ccvs)

Description

A current-controlled source senses the current with a probe device. A valid probe is a component instance in the circuit that naturally computes current. For example, probes can be voltage sources (independent or controlled), inductors, transmission lines, microstrip lines, N-ports, and transformers. If the probe device computes more than one current (such as transmission lines, microstrip lines, and N-ports), the index of the probe port through which the controlling current flows needs to be specified. Current through the controlled voltage source is calculated and is defined to be positive if it flows from the positive terminal, through the source, to the negative terminal.

This device is supported within altergroups.

Sample Instance Statement

```
vvs (pos gnd) ccvs rm=1 probe=v1 m=1   //Note that v1 is an instance of a voltage
source
```

Instance Definition

```
Name  p  n ccvs parameter=value ...
```

Instance Parameters

1 `rm=0 Ω`Transresistance.

2 `probe`Device through which the controlling current flows.

3 `port=0`Index of the probe port through which the controlling current flows.

4 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Circuit Reduced Order Model (cktrom)

Description

The circuit reduced order model is described by a set of partial differential equations in the form of:

$$x = Ax + Bu \qquad (1)$$

$$y = Cx + Du \qquad (2)$$

where Eqn.(1) is the state equation, Eqn.(2) is the output equation, A is nxn matrix, B is nxm, C is mxn, and D is an mxm matrix. x is a vector of state variables. Input u is a vector of voltages at all the ports and output y is a vector of electric current at all the ports. The number of inputs is always equal to the number of outputs. The order of the terminals in the input must be consistent with the matrix equations. In the input file, the matrices A, B, C and D are in the form of long vectors with row order.

This device is not supported within altergroup.

Sample Instance Statement

```
rom3 (net11 0) cktrom a=[ -2.022852e+14  2.583012e+13  9.553125e+13  9.627727e+13
1.533971e+13 9.987851e+13 4.592012e+13 -1.671024e+14  2.296589e+13 -2.719915e+14
7.668472e+12 -1.564519e+14  8.543123e+13  3.395689e+13 -3.863150e+14 -1.101618e+14
-5.415116e+14 -2.303841e+14  9.627728e+13 -1.711915e+14 -1.001818e+14 -
8.123120e+14 -2.272715e+14 -9.965181e+14 1.514961e+14  7.668372e+12 -6.415116e+14
-3.272715e+14 -2.852751e+15 -3.564466e+14 9.999851e+13 -1.761619e+14 -1.312841e+14
-8.967181e+14 -4.563456e+14 -4.068747e+15 ]

b=[3.366776e+06  5.932470e+05 -1.508475e+06  4.349182e+06 -3.128869e+06 -
2.995677e+06 -2.831481e+06  2.708942e+06 -4.968876e+06 -3.338945e+06 -3.278564e+06
3.925648e+06 ]

c=[-3.111296e+06  1.593292e+06  3.324594e+06  3.083731e+06  5.887179e+06
3.766094e+06 -5.049263e+05 -4.275158e+06  3.035578e+06 -3.666385e+06  3.424639e+06
-3.832285e+06]

d=[1.254627e-01  0.000000e+00 0.000000e+00  1.236790e-01 ]
```

Instance Definition

```
Name  sink0  src0  [sink1]  [src1]  [sink2]  [src2]  [sink3]  [src3] ... cktrom
parameter=value ...
```

Instance Parameters

1 `m=1`Multiplicity factor.

2 `a=[...]`Coefficient matrix A of state equations.

3 `b=[...]`Coefficient Matrix B of state equations.

4 `c=[...]`Coefficient matrix C of output equations.

5 `d=[...]`Coefficient matrix D of output equations.

Operating-Point Parameters

1 `i=[...] A`Port currents.

2 `v=[...] V`Port voltages.

3 `pwr (W)`Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

a	I-2	c	I-4	i	OP-1	pwr	OP-3
b	I-3	d	I-5	m	I-1	v	OP-2

Magnetic Core with Hysteresis (core)

Description

This component models the magnetic hysteresis, with air gap, frequency, and temperature effects. The model is based on the AWB model for magnetic cores and windings. The user has to specify the cores material and geometric parameters to model the hysteresis.

The material parameters to specify are the `Br`, `Bm` and `Hc` of the core. The geometric parameters are the area, magnetic path length and the air gap of the core.

You can specify the magnetic path length in one of the following ways:

- Give the length directly in cm.
- Or give the outer and inner diameter of the core.

Cores without terminals represent complete magnetic loops. Cores with terminals are fragments that you can use as building blocks to build models of complicated core structures. For example, you can use the following set of core fragments to model an `E` core:

```
W1  e1p  e1m  winding turns=80 core=C1
W3a e2p  e2c  winding turns=80 core=C3
W3b e2c  e2m  winding turns=80 core=C3
C1  m1   0    permalloy area=1 len=2
C2  m1   0    permalloy area=2 len=2
C3  m1   0    permalloy area=1 len=2
model permalloy core ...
```

There are three parallel core fragments representing each of the three fingers on the `E`. One 80 turn winding is connected to core fragment `C1`. A center-tapped 160 turn winding (implemented as a pair of windings) are wrapped around core fragment `C3`. Node `m1` is a `magnetic` node whose value is in magnetomotive force and flow is flux.

You can calculate the frequency and temperature dependency of the core model by specifying the frequency loss parameters and the temperature effects parameters. You can make all the core parameters vary in temperature, including the permeability, saturation flux, and core loss. For frequency losses, a static model refers to a value that you type in for frequency. This model does not adjust the shape of the B-H loop in response to power dissipation or rate of rise of the applied currents and voltages during transient analysis.

This device is not supported within altergroup.

The hysteresis is modeled by different regions whose equations are:

phi = phir + (phis - phir) F/(F + Ha) for region number 1

phi = phis * (F - Fc)/(F - Hb) for region number 2

where

phi = flux density

F = magnetomotive force

phir = residual flux density

phis = Saturated flux density

Fc = Coercive magnetic force

Ha and Hb = shape parameters.

Sample Instance Statement

```
c1 (1 0) core_mod area=1.2 len=8.1 id=0.55 gap=0.25
```

Sample Model Statement

```
model core_mod core len=7.7 area=0.85 br=1e3 bm=5e3 hc_t1=0.2 p1_f1=2.08 f1=10e3
p2_f2=50 f2=100K bflux=1e3 density=4.75
```

Instance Definition

```
Name ... ModelName parameter=value ...
```

Instance Parameters

1 `area (cm`^{2}`)`Effective magnetic cross-sectional area of core.

2 `len (cm)`Effective length of magnetic path.

3 `id (cm)`Inner diameter of toroidal core.

4 `od (cm)`Outer diameter of toroidal core.

5 `gap (cm)`Gap length.

6 `m=1`Multiplicity factor.

Model Definition

```
model modelName core parameter=value ...
```

Model Parameters

1 `br=1 gauss`Residual flux density.

2 `bm=1 gauss`Saturation flux density.

3 `hc=1 oersteds`Coercive magnetizing force (value of H where B equals 0).

4 `area=1 cm`^{2}Effective magnetic cross-sectional area of core.

5 `len=1 cm`Effective length of magnetic path.

6 `id (cm)`Inner diameter of toroidal core.

7 `od (cm)`Outer diameter of toroidal core.

8 `gap=0.0 cm`Gap length.

Initial Conditions

9 `b0 (gauss)`Initial condition for core.

Frequency Loss Parameters

10 `freq (Hz)`Core operating frequency.

11 `p1_f1 (W/Kg)`Core power loss at frequency `f1`.

12 `f1 (Hz)`Reference frequency for power loss.

13 `p2_f2 (W/Kg)`Core power loss at frequency `f2`.

14 `f2 (Hz)`Reference frequency for power loss.

15 `bflux (gauss)`Reference flux density.

16 `density (g/cm`^{3}`)`Core density.

Temperature Effects Parameters

17 `temp (C)`Core operating temperature.

18 `bm_t1 (gauss)`Saturated flux density `Bm` at `T1`.

19 `br_t1 (gauss)`Residual flux density `Br` at `T1`.

20 `hc_t1 (oersteds)`Coercive force `Hc` at `T1`.

21 `t1 (C)`Reference temperature.

Operating-Point Parameters

1 `b (gauss)`Flux density of the core.

2 `h (oersteds)`Magnetic field strength.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Logic-to-Analog Converter (d2a)

Description

The logic-to-analog converter converts a binary signal from a logic simulator to an analog waveform.

This device is not supported within altergroup.

Sample Instance Statement

```
d2a_1 (net1 net2) d2a src=”99991” val0=0 val1=2.5 valx=1.25 rise=200p fall=200p m=2
//99991 is an analog net
```

Instance Definition

```
Name  p  n d2a parameter=value ...
```

Instance Parameters

1 `src`The foreign simulators name for the source of the analog signal.

2 `nestlev=0`Number of nesting levels to ignore in the hierarchical name. This should be used skip over extra levels that do not exist in the co-simulator.

3 `val0=0 V`Final value for logical 0.

4 `val1=5 V`Final value for logical 1.

5 `valx (V)`Final value for logical X.

6 `valz (V)`Final value for logical Z.

7 `rise=1ns s`Time for transition from `val0` to `val1`.

8 `fall=1ns s`Time for transition from `val1` to `val0`.

9 `ron=100 Ω`Output resistance when in active state.

10 `m=1`Multiplicity factor.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Delay Line (delay)

Description

The delay line model is a four terminal device with zero output impedance and infinite input impedance. The output between nodes p and n is the input voltage between nodes `ps` and `ns` delayed by the time delay `td` and scaled by `gain`.

This device is not supported within altergroup.

Sample Instance Statement

```
dl1(outp outn cntrlp cntrln) delay td=10n gain=1.5
```

Instance Definition

```
Name  p  n  ps  ns delay parameter=value ...
```

Instance Parameters

1 `td=0.0 s`Time delay.

2 `gain=1`Gain parameter.

3 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `v (V)`Output voltage.

Diode Level 500 (dio500)

Description

The `dio500` model provides a detailed description of the diode currents in forward and reverse biased Si-diodes. It is described in the Philips Bipolar Modelbook (Dec.93) as `Diode level 500`. Information on how to obtain this document can be found on Source Link by searching for Philips.

In extension to the modelbook description a minimum conductance `gmin` is inserted between the diode nodes to aid convergence. The value of `gmin` is set by an options statement, default is gmin = 1.0e-12 S .

The `imax` parameter is used to aid convergence and to prevent numerical overflow. The junction characteristics of the diode are accurately modeled for currents up to `imax`. For currents above `imax`, the junction is modeled as a linear resistor, and a warning is printed.

This device is supported within altergroups.

This device is dynamically loaded from the shared object /vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so

Sample Instance Statement

d1 (pnode 0) phdiode area=2

Sample Model Statement

model phdiode dio500 is=3.5e-12 rs=26.3 n=2.7 imax=1e20 vlc=1.8 vbr=9.63 cj=2.65e-11 dta=12.88 tau=7.5e-10 tnom=25

Instance Definition

Name `a` `k` ModelName parameter=value ...

Instance Parameters

1 `area=1.0`Multiplication factor.

2 `mult`Alias of area factor.

3 `m=1.0`Multiplicity factor.

4 `region=fwd`Estimated DC operating region, used as a convergence aid. Possible values are `fwd`, `rev` or `brk`.

Model Definition

model modelName dio500 parameter=value ...

Model Parameters

1 `is=7.13e-13 A`Saturation current.

2 `n=1.044`Junction emission coefficient.

3 `vlc=0.0 V`Voltage dependence at low forward currents.

4 `vbr=7.459 V`Breakdown voltage.

5 `emvbr=1.36e+06 V/cm`
Electric field at breakdown.

6 `csrh=7.44e-07 A/cm`
Shockley-Read-Hall generation.

7 `cbbt=3.255 A/V`Band to band tunneling.

8 `ctat=3.31e-06 A/cm`
Trap assisted tunneling.

9 `rs=0.0 Ω`Series resistance.

10 `tau=500.0e-12 s`Transit time.

11 `cj=7.0e-12 F`Zero-bias depletion capacitance.

12 `vd=0.9 V`Diffusion voltage.

13 `p=0.4`Grading coefficient.

14 `tref (C)`Reference temperature. Default set by option `tnom`.

15 `tnom (C)`Alias of `tref`.

16 `tr (C)`Alias of `tref`.

17 `vg=1.206 V`Bandgap voltage.

18 `ptrs=0.0`Power for temperature dependence of `rs`.

19 `kf=0.0`Flickernoise coefficient.

20 `af=1.0`Flickernoise exponent.

21 `dta=0.0 K`Difference between device temperature and ambient temperature.

22 `trise (K)`Alias of `dta`.

23 `imax=1.0 A`Explosion current.

Operating-Point Parameters

1 `vak (V)`Diode voltage, measured from anode to cathode (including `rs`).

2 `id (A)`Total resistive diode current.

3 `qd (Coul)`Diffusion charge.

4 `qt (Coul)`Depletion charge.

5 `rst (Ω)`Series resistance (temperature updated).

6 `rl (Ω)`AC linearized resistance.

7 `cl (F)`AC linearized capacitance.

8 `pwr (W)`Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Junction Diode (diode)

Description

The junction diode model includes nonlinear junction capacitance and reverse breakdown.

This device is supported within altergroups.

To examine the equations used for this component, consult the *Spectre Circuit Simulator Device Model Equations* manual.

Sample Instance Statement

```
d0 (dp dn) pdiode l=3e-4 w=2.5e-4 area=1
```

Sample Model Statement

```
model pdiode diode is=1.8e-5 rs=1.43 n=1.22 nz=2.31 gleak=6.2e-5 rsw=10 isw=6.1e-
10 ibv=0.95e-3 tgs=2 ik=1.2e7 fc=0.5 cj=1.43e-3 pb=0.967 mj=0.337 cjsw=2.76e-9
vjsw=0.94 jmax=1e20
```

Instance Definition

```
Name  a  c ModelName parameter=value ...
```

In forward operation, the voltage on the anode (a) is more positive than the voltage on the cathode (c).

Instance Parameters

1 `area`Junction area factor.

2 `perim`Junction perimeter factor.

3 `l=1e-6 m`Drawn length of junction.

4 w=1e-6 mDrawn width of junction.

5 m=1Multiplicity factor.

6 scale=1Scale factor.

7 region=onEstimated operating region.
Possible values are off, on or breakdown.

8 trise (C)Temperature rise from ambient.

The instance parameter scale, if specified, overrides the value given by the option parameter scale. If the model parameter allow_scaling is set to yes then, the area, perim, l and w parameters are scaled by scale. By default allow_scaling is set to no and no scaling of geometry parameters will occur. The values of area, perim, l and w printed out by spectre are those given in the input, and these values might not have the correct units if the scaling factors are not unity.

Model Definition

```
model modelName diode parameter=value ...
```

Model Parameters

Model selector parameters

1 level=1Model selector. 1 = junction and 2 = Fowler-Nordheim.

Process parameters

2 etch=0 mNarrowing due to etching per side.

3 etchl=etch mLength reduction due to etching per side.

Junction diode parameters

4 js=1e-14 ASaturation current (*area).

5 jsw=0 ASidewall saturation current (*perim).

6 n=1Emission coefficient.

7 ns=1Sidewall emission coefficient.

8 ik=∞ AHigh-level injection knee current (*area).

9 ikp=ik AHigh-level injection knee current for sidewall (*area).

10 area=1Junction area factor.

11 perim=0Junction perimeter factor.

12 allow_scaling=noAllow scale option and instance scale parameter to affect diode instance geometry parameters.
Possible values are no or yes.

Capacitive parameters

13 tt=0 sTransit time.

14 cd=0 FLinear capacitance (*area).

15 cjo=0 FZero-bias junction capacitance (*area).

16 vj=1 VJunction potential.

17 m=0.5Grading coefficient.

18 cjsw=0 FZero-bias sidewall junction capacitance (*perim).

19 vjsw=1 VSidewall junction potential.

20 mjsw=0.33Sidewall grading coefficient.

21 fc=0.5Forward-bias depletion capacitance threshold.

Breakdown parameters

22 bv=∞ VReverse breakdown voltage. Note: bv=0 is not the same as bv=infinity.

23 ibv=0.001 ACurrent at breakdown voltage (*area).

24 nz=1Emission coefficient for Zener diode.

25 `bvj=∞ V`Voltage at which junction breakdown warning is issued.

Parasitic resistance parameters

26 `rs=0 Ω`Series resistance (/area).

27 `rsw=0 Ω`Sidewall series resistance (/perim).

28 `gleak=0 S`Bottom junction leakage conductance (*area).

29 `gleaksw=0 S`Sidewall junction leakage conductance (*perim).

30 `minr=0.1 Ω`Minimum series resistance.

Temperature effects parameters

31 `tlev=0`DC temperature selector.

32 `tlevc=0`AC temperature selector.

33 `eg=1.124481 V`Band gap. Note: when not specified, the default value is temperature dependent. It is 1.124481 at temp=27C.

34 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

35 `gap2=1108 C`Band gap temperature offset.

36 `xti=3`Saturation current temperature exponent.

37 `tbv1=0 1/C`Linear temperature coefficient for `bv`.

38 `tbv2=0` C^{-2}Quadratic temperature coefficient for `bv`.

39 `tnom (C)`Parameters measurement temperature. Default set by `options`.

40 `trise=0 C`Temperature rise from ambient.

41 `trs=0 1/C`Linear temperature coefficient for parasitic resistance.

42 `trs2=0` C^{-2}Quadratic temperature coefficient for parasitic resistance.

43 `tgs=0 1/C`Linear temperature coefficient for leakage conductance.

44 tgs2=0 C^{-2}Quadratic temperature coefficient for leakage conductance.

45 cta=0 1/CJunction capacitance temperature coefficient.

46 ctp=0 1/CSidewall junction capacitance temperature coefficient.

47 pta=0 V/CJunction potential temperature coefficient.

48 ptp=0 V/CSidewall junction potential temperature coefficient.

Junction diode model control parameters

49 jmelt=jmax AExplosion current (*area).

50 jmax=1 AMaximum allowable current (*area).

51 dskip=yesUse simple piece-wise linear model for diode currents below 0.1*iabstol.
Possible values are no or yes.

Fowler-Nordheim diode parameters

52 if=1e-10 A/V^nfForward Fowler-Nordheim current coefficient (*area).

53 ir=if A/V^nrReverse Fowler-Nordheim current coefficient (*area).

54 ecrf=2.55e10 V/mForward critical field.

55 ecrr=ecrf V/mReverse critical field.

56 nf=2Forward voltage power.

57 nr=nfReverse voltage power.

58 tox=1e-8 mThickness of insulating layer.

Noise model parameters

59 kf=0Flicker noise (1/f) coefficient.

60 af=1Flicker noise (1/f) exponent.

Both of these parameters have current density counterparts, `jmax` and `jmelt`, that you can specify if you want the absolute current values to depend on the device area.

Operating-Point Parameters

1 `region=on` Estimated operating region.
Possible values are `off`, `on` or `breakdown`.

2 `v (V)` Extrinsic diode voltage.

3 `i (A)` Resistive diode current.

4 `pwr (W)` Power dissipation.

5 `res (Ω)` Resistance of intrinsic diode.

6 `cap (F)` Junction capacitance.

7 `resp (Ω)` Resistance of intrinsic sidewall diode.

8 `capp (F)` Sidewall junction capacitance.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

EKV MOSFET Transistor (ekv)

Description

The EPFL-EKV mosfet model was developed by Electronics Laboratories, Swiss Federal Institute of Technology (EPFL), Switzerland. The detailed description of the model and equations can be found in the *Spectre Circuit Simulator Device Model Equations* manual. EKV transistors require that you use a model statement.

This device is supported within altergroups.

Sample Instance Statement

```
mn1 (dn gn sn 0) ekvnmos w=1.5u l=1u ad=2.6p as=2.6p pd=6.6p ps=6.6p nrd=1.54
nrs=1.54
```

Sample Model Statement

```
model ekvnmos ekv type=n update=2.6 xqc=0.4 cox=3.4e-3 xj=0.145e-6 vto=0.6
gamma=0.71 phi=0.967 kp=155e-6 e0=88e6 iba=200e6 ibb=350e6 tnom=25 tcv=1.55e-3
bex=-1.45 kf=1e-27 af=1 hdif=0.94e-6 rsh=512 jsw=1.5e-10
```

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as` (m^2)Area of source diffusion.

4 `ad` (m^2)Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `rdc (Ω)`Drain contact resistance.

10 `rsc (Ω)`Source contact resistance.

11 `m=1`Multiplicity factor (number of MOSFETs in parallel).

12 `ns=1`Series Multiplicity factor (number of MOSFETs in series).

13 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, or `subth`.

14 `trise`Temperature rise from ambient.

Model Definition

```
model modelName ekv parameter=value ...
```

Model Parameters

Device type parameters

1 `type=n`Transistor type.
Possible values are n or p.

Process parameters

2 `tox=2e-8 m`Gate oxide thickness.

3 `cox=7e-4 F/m`^{2}Gate oxide capacitance. (Overrides Tox).

4 `xj=1.0e-7 m`Metallurgical junction depth.

5 `dw=0 m`Channel Width Correction.

6 `dl=0 m`Channel Length Correction.

7 `nfs=0 cm`$^{-2}$Fast surface state density.

8 `nsub=1.13e16 cm`$^{-3}$
Channel doping concentration.

Drain current model parameters

9 `vto=0.5 V`Threshold voltage at zero body bias.

10 `gamma=1.0 √V` Body-effect parameter.

11 `phi=0.7 V`Surface potential at strong inversion.

12 `kp=5.0e-5 A/V`^{2}Transconductance parameter.

13 `e0=1.0e12 V/m`Vertical Critical Field.

14 `ucrit=2.0e6 V/cm`Longitudinal Critical field for mobility degradation.

15 `theta=0.0 1/V`Mobility reduction coefficient.

16 `uo (cm`^{2}`/V s)`Carrier surface mobility.

17 `vmax (m/s)`Carrier saturation velocity.

18 `vfb (V)`Flat-band voltage.

19 `lambda=0.5`Channel length modulation parameter.

20 `weta=0.25`Narrow Channel Effect Coefficient.

21 `leta=0.1`Short Channel Effect Coefficient.

22 `xw=0 m`Width variation due to masking and etching.

23 `xl=0 m`Length variation due to masking and etching.

24 `meto=0 m`Metal overlap in fringing field.

Impact ionization parameters

25 `iba=0 1/m`First Impact Ionization Coefficient.

26 `ibb=3.0e8 V/m`Second Impact Ionization Coefficient.

27 `ibc=0`Third Impact Ionization Coefficient.

28 `ibn=1.0`Saturation velocity factor for impact ionization.

Reverse Short Channel parameters

29 `q0=0 A s/m`^{2}Reverse short channel peak charge density.

30 `lk=2.9e-7 m`Reverse short channel characteristic length.

Charge model selection parameters

31 `xqc=0.0`Drain/source channel charge partition in saturation for charge models, e.g. use 0.4 for 40/60, 0.5 for 50/50, 0 for 0/100.

Junction diode model parameters

32 `is=1e-14 A`Bulk junction reverse saturation current.

33 `js (A/m`^{2}`)`Bulk junction reverse saturation current density.

34 `jsw=0 A/m`Bulk junction reverse saturation sidewall current density.

35 `n=1`Junction emission coefficient.

36 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`. Possible values are `no` or `yes`.

37 `imelt=`imaxA'`Explosion current, diode is linearized beyond this current to aid convergence.

Junction capacitance model parameters

38 `cbd=0 F`Bulk-drain zero-bias p-n capacitance.

39 `cbs=0 F`Bulk-source zero-bias p-n capacitance.

40 `cj=0 F/m`^{2}Zero-bias junction bottom capacitance density.

41 `cjsw=0 F/m`Zero-bias junction sidewall capacitance density.

42 `mj=0.5`Bulk junction bottom grading coefficient.

43 `mjsw=0.33`Bulk junction sidewall grading coefficient.

44 `cjswg=0 F/m`Gate-side zero-bias junction sidewall capacitance density.

45 `mjswg=0.33`Gate-side bulk junction sidewall grading coefficient.

46 `pbswg=0.8 V`Gate-side junction built-in potential.

47 `fc=0.5`Forward-bias capacitance coefficient.

48 `pb=0.8 V`Bulk p-n bottom contact potential.

49 `pbsw=0.8 V`Side-wall contact potential.

50 `tt=0.0 V`Bulk p-n transit time.

51 `fcsw=0.5`Side-wall forward-bias depletion capacitance threshold.

Overlap capacitance parameters

52 `cgso=0 F/m`Gate-source overlap capacitance.

53 `cgdo=0 F/m`Gate-drain overlap capacitance.

54 `cgbo=0 F/m`Gate-bulk overlap capacitance.

Parasitic resistance parameters

55 `rs=0 Ω`Source resistance.

56 `rd=0 Ω`Drain resistance.

57 `rsh=0 Ω/sqr`Source/drain diffusion sheet resistance.

58 `rss=0 Ω m`Scalable source resistance.

59 `rdd=0 Ω m`Scalable drain resistance.

60 `rsc=0 Ω`Source contact resistance.

61 `rdc=0 Ω`Drain contact resistance.

62 `minr=0.1 Ω`Minimum source/drain resistance.

63 `ldif=0 m`Lateral diffusion beyond the gate.

64 `hdif=0 m`Length of heavily doped diffusion.

Short distance matching parameters

65 `avto=0 V m`Area related threshold voltage mismatch parameter.

66 `akp=0 m`Area related gain mismatch parameter.

67 `agamma=0 √V m`
Area related body effect mismatch parameter.

Operating region warning control parameters

68 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

69 `imax=1 A`Maximum current, currents above this limit generate a warning.

70 `jmax=1e8` A/m^2Maximum current density, currents above this limit generate a warning.

71 `vbox=1e9 tox V`Oxide breakdown voltage.

72 `bvj=∞ V`Junction reverse breakdown voltage.

Temperature effects parameters

73 `tnom (C)`Parameters measurement temperature. Default set by `options`.

74 `trise=0 C`Temperature rise from ambient.

75 `tcv=1.0e-3 V/C`Threshold voltage temperature coefficient.

76 `bex=-1.5`Mobility temperature exponent.

77 `ucex=0.8`Longitudinal critical field temp. exponent.

78 `ibbt=9.0e-4 1/C`Temperature coefficient for IBB.

79 `xti=3`Saturation current temperature exponent.

80 `tlev=0`DC temperature selector.

81 `tlevc=0`AC temperature selector.

82 `eg=1.12452 V`Energy band gap.

83 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

84 `gap2=1108 C`Band gap temperature offset.

85 `tr1=0.6`First source-drain resistance temperature coefficient.

86 `tr2=0.6`Second source-drain resistance temperature coefficient.

87 `ptc=0 V/C`Surface potential temperature coefficient.

88 `pta=0 V/C`Junction potential temperature coefficient.

89 `ptp=0 V/C`Sidewall junction potential temperature coefficient.

90 `cta=0 1/C`Junction capacitance temperature coefficient.

91 `ctp=0 1/C`Sidewall junction capacitance temperature coefficient.

Default instance parameters

92 `w=3e-6 m`Default channel width.

93 `l=3e-6 m`Default channel length.

94 `as=0` m^2Default area of source diffusion.

95 `ad=0` m^2Default area of drain diffusion.

96 `ps=0 m`Default perimeter of source diffusion.

97 `pd=0 m`Default perimeter of drain diffusion.

98 `nrd=0 m/m`Default number of squares of drain diffusion.

99 `nrs=0 m/m`Default number of squares of source diffusion.

Noise model parameters

100 `noisemod=1`Noise model selector.

101 `kf=0`Flicker (1/f) noise coefficient.

102 `af=1`Flicker (1/f) noise exponent.

103 `ef=1`Flicker (1/f) noise frequency exponent.

Model selection parameters

104 `nqs=0`Nonquasi-static flag.

105 `satlim=exp(4)`Ratio defining saturation limit.

106 `ekvint=0.0`Interpolation function selector.

107 `scalem=1.0`Model scaling factor.

108 `update=2.6`Model version selector.

Auto Model Selector parameters

109 `wmax=1.0 m`Maximum channel width for which the model is valid.

110 `wmin=0.0 m`Minimum channel width for which the model is valid.

111 `lmax=1.0 m`Maximum channel length for which the model is valid.

112 `lmin=0.0 m`Minimum channel length for which the model is valid.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

Both of these parameters have current density counterparts, `jmax` and `jmelt`, that you can specify if you want the absolute current values to depend on the device area.

Auto Model Selection

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 <= inst_length < lmax  and   wmin <= inst_width  < wmax
```

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)`Effective channel width.

2 `leff (m)`Effective channel length.

3 `rseff (Ω)`Effective source resistance.

4 `rdeff (Ω)`Effective drain resistance.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are `n` or `p`.

2 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, or `subth`.

3 `reversed`Reverse mode indicator.
Possible values are `no` or `yes`.

4 `ids (A)`Resistive drain-to-source current.

5 `vgs (V)`Gate-source voltage.

6 `vds (V)`Drain-source voltage.

7 `vbs (V)`Bulk-source voltage.

8 `vp (V)`Pinchoff voltage.

9 `vth (V)`Threshold voltage.

10 `vdss (V)`Drain-source saturation voltage.

11 `gm (S)`Common-source transconductance.

12 `gds (S)`Common-source output conductance.

13 `gmbs (S)`Body-transconductance.

14 `nfac`Slope factor.

15 `if (A)`Forward current.

16 `ir (A)`Reverse current.

17 `irprime (A)`Reverse current.

18 `isub (A)`Substrate Current.

19 `ibd (A)`Bulk-drain junction current.

20 `ibs (A)`Bulk-source junction current.

21 `pwr (W)`Power at op point.

22 `gmoverid (1/V)Gm/Ids`.

23 `gamma (`$\sqrt{V}$` )`Body-effect parameter.

24 `cjd (F)`Drain-bulk junction capacitance.

25 `cjs (F)`Source-bulk junction capacitance.

26 `cgg (F)`Gate-gate capacitance.

27 `cgd (F)`Gate-drain capacitance.

28 `cgs (F)`Gate-source capacitance.

29 `cgb (F)`Gate-bulk capacitance.

30 `cdg (F)`Drain-gate capacitance.

31 `cdd (F)`Drain-drain capacitance.

32 `cds (F)`Drain-source capacitance.

33 `cdb (F)`Drain-bulk capacitance.

34 `csg (F)`Source-gate capacitance.

35 `csd (F)`Source-drain capacitance.

36 `css (F)`Source-source capacitance.

37 `csb (F)`Source-bulk capacitance.

38 `cbg (F)`Bulk-gate capacitance.

39 `cbd (F)`Bulk-drain capacitance.

40 `cbs (F)`Bulk-source capacitance.

41 `cbb (F)`Bulk-bulk capacitance.

42 `vm (V)`Early voltage.

43 `vovrdr (V)`Overdrive voltage.

44 `tau (s)`NQS time constant.

45 `tau0 (s)`Intrinsic time constant.

46 `ron (Ω)`On-resistance.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Ratiometric Fourier Analyzer (fourier)

Description

The ratiometric Fourier analyzer measures the Fourier coefficients of two different signals at a specified fundamental frequency without loading the circuit. The algorithm used is based on the Fourier integral rather than the discrete Fourier transform and therefore is not subject to aliasing. Even on broad-band signals, it computes a small number of Fourier coefficients accurately and efficiently. Therefore, this Fourier analyzer is suitable on clocked sinusoids generated by sigma-delta converters, pulse-width modulators, digital-to-analog converters, sample-and-holds, and switched-capacitor filters as well as on the traditional low-distortion sinusoids produced by amplifiers or filters.

The analyzer is active only during a transient analysis. For each signal, the analyzer prints the magnitude and phase of the harmonics along with the total harmonic distortion at the end of the transient analysis. The total harmonic distortion is found by summing the power in all of the computed harmonics except DC and the fundamental. Consequently, the distortion is not accurate if you request an insufficient number of harmonics The Fourier analyzer also prints the ratio the spectrum of the first signal to the fundamental of the second, so you can use the analyzer to compute large signal gains and immittances directly.

If you are concerned about accuracy, perform an additional Fourier transform on a pure sinusoid generated by an independent source. Because both transforms use the same time points, the relative errors measured with the known pure sinusoid are representative of the errors in the other transforms. In practice, this second Fourier transform is performed on the reference signal. To increase the accuracy of the Fourier transform, use the `points` parameter to increase the number of points. Tightening `reltol` and setting `errpreset=conservative` are two other measures to consider.

The accuracy of the magnitude and phase for each harmonic is independent of the number of harmonics computed. Thus, increasing the number of harmonics (while keeping `points` constant) does not change the magnitude and phase of the low order harmonics, but it does improve the accuracy of the total harmonic distortion computation. However, if you do not specify `points`, you can increase accuracy by requesting more harmonics, which creates more points.

The large number of points required for accurate results is not a result of aliasing. Many points are needed because a quadratic polynomial interpolates the waveform between the time-points. If you use too few time-points the polynomials deviate slightly from the true waveform between time-points and all of the computed Fourier coefficients are slightly in error. The algorithm that computes the Fourier integral does accept unevenly spaced time-points, but because it uses quadratic interpolation, it is usually more accurate using time-steps that are small and nearly evenly spaced.

This device is not supported within altergroup.

Sample Instance Statement

```
four1 (1 0) fourmod harms=50
```

Sample Model Statement

```
model fourmod fourier fund=900M points=2500 order=2
```

Instance Definition

```
Name  [p]  [n]  [pr]  [nr] ModelName parameter=value ...
```

Name [p] [n] [pr] [nr] fourier parameter=value ...

The signal between terminals p and n is the test or numerator signal. The signal between terminals `pr` and `nr` is the reference or denominator signal. Fourier analysis is performed on terminal currents by specifying the `term` or `refterm` parameters. If both `term` and p or n are specified, then the terminal current becomes the numerator and the node voltages become the denominator. By mixing voltages and currents, it is possible to compute large signal immittances.

Instance Parameters

1 `fund (Hz)`Fundamental frequency.

2 `points=20 maxharm`
Minimum number of time points.

3 `active=yes`Whether Fourier analysis should be performed or skipped.
Possible values are `no` or `yes`.

4 `order=2`Order of interpolation.

5 `term`Terminal used to measure current for test (numerator) channel.

6 `refterm`Terminal used to measure current for reference (denominator) channel.

7 `harmsvec=[...]`Array of desired harmonics for test (numerator) channel.

8 `harms=9`Number of harmonics for test (numerator) channel, if an array is not given. The harmonics start from `firstharm` and go up to `firstharm` + `harms` - `1`.

9 `refharmsvec=[...]`Array of desired harmonics for reference (denominator) channel.

10 `refharms=9`Number of harmonics for reference (denominator) channel, if an array is not given. The harmonics start from `reffirstharm` and go up to `reffirstharm` + `harms` - `1`.

11 `scale=1`Scale factor for ratioed results.

12 `firstharm=1`First harmonic computed for test (numerator) channel.

13 `reffirstharm=1`First harmonic computed for reference (denominator) channel.

14 `normharm=1`Normalizing harmonic for test (numerator) channel.

15 `refnormharm=1`Normalizing harmonic for reference (denominator) channel.

16 `where=logfile`Where Fourier results should be printed.
Possible values are `screen`, `logfile` or `both`.

Model Definition

```
model modelName fourier parameter=value ...
```

Model Parameters

1 `fund (Hz)`Fundamental frequency.

2 `points=20 maxharm`
Minimum number of time points.

3 `harms=9`Desired number of harmonics.

4 `active=yes`Whether Fourier analysis should be performed or skipped.
Possible values are no or `yes`.

5 `order=2`Order of interpolation.

6 `firstharm=1`First harmonic computed for test (numerator) channel.

7 `reffirstharm=1`First harmonic computed for reference (denominator) channel.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

active I-3	harms I-8	points I-2	refnormharm I-15
active M-4	harms M-3	points M-2	refterm I-6
firstharm M-6	harmsvec I-7	reffirstharm M-7	scale I-11
firstharm I-12	normharm I-14	reffirstharm I-13	term I-5
fund I-1	order M-5	refharms I-10	where I-16
fund M-1	order I-4	refharmsvec I-9	

GaAs MESFET (gaas)

Description

The GaAs MESFET model was derived from the model by H. Statz and others at Raytheon. This model is completely symmetric and is modified slightly to make it charge conserving. GaAs MESFET instances require that you use a model statement.

This device is supported within altergroups.

There are some convergence problems with this model because of `Cgs` going to zero beyond pinchoff. The problems occur when the gate is driven from an inductive source, and there is no other capacitance at the gate. To prevent these problems, avoid setting `Cgd` to zero and add sidewall capacitance to the gate-source and gate-drain junctions. A good estimate for these capacitors is C = pi*epsilon*w/2 where w is the gate width in microns and epsilon = 0.116 fF/micron.

To examine the equations used for this component, consult the *Spectre Circuit Simulator Device Model Equations* manual.

Sample Instance Statement

```
m1 (1 2 0) nmes area=1 m=2
```

Sample Model Statement

```
model nmes gaas type=n vto=-2 beta=0.06 lambda=0 b=0.25 rs=3.65 alpha=1.9 rd=1.98
is=1.1e-9 n=1.28 fc=0.5 cgs=0.365e-12
```

Instance Definition

```
Name  d  g  s ModelName parameter=value ...
```

Instance Parameters

1 `area=1`Junction area factor.

2 `m=1`Multiplicity factor.

3 `region=fwd`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

Model Definition

```
model modelName gaas parameter=value ...
```

Model Parameters

Device type parameters

1 `type=n`Transistor type.
Possible values are n or p.

Drain current parameters

2 `vto=-2 V`Pinch-off voltage.

3 `beta=0.0001 A/V`^{2}Transconductance parameter.

4 `lambda=0 1/V`Channel length modulation parameter.

5 `b=0.3 1/V`Doping tail extending parameter.

6 `alpha=2 1/V`Saturation voltage parameter.

Parasitic resistance parameters

7 `rd=0` ΩDrain resistance (/area).

8 `rs=0` ΩSource resistance (/area).

9 `rg=0` ΩGate resistance (/area).

10 `minr=0.1` ΩMinimum source/drain/gate resistance.

Junction diode model parameters

11 `is=1e-14` AGate saturation current (*area).

12 `n=1`Emission coefficient for the gate junction.

13 `imelt=`imaxA'`Explosion current (*area).

14 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`.
Possible values are `no` or `yes`.

Junction capacitance model parameters

15 `capmod=2`Charge model selector.

16 `cgs=0` FGate-source zero-bias junction capacitance (*area).

17 `cgd=0` FGate-drain zero-bias junction capacitance (*area).

18 `pb=1` VGate junction potential.

19 `fc=0.5`Junction capacitor forward-bias threshold.

20 `delta=0.2` VGate capacitance pinch-off transition width.

Temperature effects parameters

21 `tnom (C)`Parameters measurement temperature. Default set by options.

22 `trise=0` CTemperature rise from ambient.

23 `xti=3`Temperature exponent for effect on `is`.

Operating region warning control parameters

24 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

25 `imax=1 A`Maximum allowable current (*area).

26 `bvj=∞ V`Junction reverse breakdown voltage.

Noise model parameters

27 `kf=0`Flicker noise (1/f) coefficient.

28 `af=1`Flicker noise (1/f) exponent.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

The bv parameter detects the junction breakdown only. The breakdown currents of the junctions are not modeled.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are n or p.

2 `region=fwd`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

3 `ids (A)`Resistive drain current.

4 `vth (V)`Threshold voltage.

5 `vgs (V)`Gate-source voltage.

6 `vds (V)`Drain-source voltage.

7 `vdsat (V)`Drain saturation voltage.

8 `gm (S)`Common-source transconductance.

9 `gds (S)`Common-source output conductance.

10 `cgs (F)`Gate-source capacitance.

11 `cgd (F)`Gate-drain capacitance.

12 `ig (A)`Resistive gate current.

13 `pwr (W)`Power at operating point.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

4

Component Statements Part 2

This chapter discusses the following topics:

Hetero-Junction Bipolar Transistor (hbt)

Description

The HBT (Hetero-junction Bipolar Transistor) model was developed by UCSD as part of the ARPA High Speed Circuit Design Program. The model has four external electrical nodes, one thermal node, and up to nine internals depending on the complexity of the model users specified. Detailed description of the model and equations and be found in the Spectre Circuit Simulator Device Model Equations manual.

This device is supported within altergroups.

Sample Instance Statement

```
q7 (net5 net2 0) hbtmod m=1 top=25
```

Sample Model Statement

```
model hbtmod hbt type=npn bf=500 br=1000 xtb=-2.4 xti=0 xcjc=0.83 mje=0.34 fc=0.5
eg=1.2 ise=5.5e-15 vjc=0.84 vaf=40 cjc=5.1e-15
```

Instance Definition

```
Name  c  b  e  [t]  [s] ModelName parameter=value ...
```

It is not necessary to specify the substrate and thermal terminal. If left unspecified, the substrate node is connected to ground while the thermal node is fixed to the ambient temperature. However, you must specify the thermal node if you specify the substrate node.

Instance Parameters

1 `area=1`Transistor area factor.

2 `m=1`Multiplicity factor.

3 `top (C)`Average device operating temperature.

4 `region=fwd`Estimated operating region.
Possible values are `off`, `fwd`, `rev`, or `sat`.

Model Definition

```
model modelName hbt parameter=value ...
```

Model Parameters

Structural Parameters

1 `type=npn`Transistor type.
Possible values are `npn` or `pnp`.

Saturation Current Parameters

2 `is=1e-25 A`Saturation value for forward collector current (*area).

3 `ise=1e-25 A`Saturation value for nonideal base current. (*area).

4 `isex=1e-25 A`Saturation current for emitter leakage diode (*area).

5 `isc=1e-20 A`Saturation value for intrinsic BC junction current. (*area).

6 `iscx=1e-20 A`Saturation value for extrinsic B-C junction current (*area).

7 `ics=1e-30 A`Saturation value for C-S junction current (*area).

Emission Coefficient Parameters

8 `nf=1`Forward collector current ideality factor.

9 `nr=1`Reverse ideality factor.

10 `ne=2`Nonideal base forward current ideality factor.

11 `nex=2`Ideality factor for emitter leakage diode.

12 `nc=2`Intrinsic B-C junction ideality factor.

13 `ncx=2`Ideality factor for extrinsic B-C junction.

14 `ncs=2`Ideality factor for C-S junction.

Current Gain Parameters

15 `bf=1000 A/A`Forward ideal current gain (beta).

16 `br=1000 A/A`Reverse ideal current gain.

17 `isa=1e10 A`Collector E-B barrier limiting current (*area).

18 `na=2`Collector E-B barrier ideality factor.

19 `isb=1e10 A`Collector B-C barrier limiting current (*area).

20 `nb=2`Collector B-C barrier ideality factor.

21 `ik=1e10 A`Knee current for dc high-level injection effect (*area).

Early Voltage Parameters

22 `vaf=500 V`Forward Early voltage.

23 `var=500 V`Reverse Early voltage.

Breakdown Voltage Parameters

24 `bkdn=no`Flag denoting B-C breakdown should be included.
Possible values are `no` or `yes`.

25 `bvc=50 A`Collector-base breakdown voltage BVcbo.

26 `nbc=8`Exponent for B-C multiplication factor versus voltage.

27 `fa=0.9`Factor for specification of avalanche voltage.

28 `imax=1 A`Maximum allowable base current (*area).

29 `imelt=10 A`Explosion current (*area).

Parasitic Resistance Parameters

30 `rbi=0 Ω`Intrinsic base resistance (/area).

31 `rbx=0 Ω`Extrinsic base resistance (/area).

32 `rci=0` ΩIntrinsic collector resistance (/area).

33 `rcx=0` ΩExtrinsic collector resistance (/area).

34 `re=0` ΩEmitter resistance (/area).

35 `rex=0` ΩExtrinsic emitter leakage diode series resistance (/area).

Junction Capacitance Parameters

36 `cje=0` FB-E depletion capacitance at zero bias (*area).

37 `vje=1.6` VB-E built-in potential for Cj.

38 `mje=0.5`Exponent for voltage variation of B-E Cj.

39 `cemin=0` FMinimum B-E capacitance (*area).

40 `fce=0.8`Factor for start of high bias B-E Cj approximation.

41 `cjc=0` FIntrinsic B-C depletion capacitance at zero bias (*area).

42 `vjc=1.4` VIntrinsic B-C built-in potential for Cj.

43 `mjc=0.33`Exponent for voltage variation of Intrinsic B-C Cj.

44 `ccmin=0` FMinimum B-C capacitance (*area).

45 `fc=0.8`Factor for start of high bias B-C Cj approximation.

46 `cjcx=0` FExtrinsic B-C depletion capacitance at zero bias (*area).

47 `vjcx=1.4` VExtrinsic B-C built-in potential for Cj.

48 `mjcx=0.33`B-C junction exponent.

49 `cxmin=0` FMinimum extrinsic B-C capacitance (*area).

50 `xcjc=0`Fraction of B-C capacitance tied to external base node.

51 `cjs=0` FB-S depletion capacitance at zero bias (*area).

52 `vjs=1.4` VB-S built-in potential for Cj.

53 `mjs=0.5`Exponent for voltage variation of C-S Cj.

Transit Time and Excess Phase Parameters

54 `tfb=0 s`Base transit time.

55 `tbexs=0`Excess B-E hererojunction transit time.

56 `tbcxs=0`Excess B-C hererojunction transit time.

57 `tfc0=0 s`Collector forward transit time.

58 `icrit0=1e3 A`Critical current for intrinsic Cj variation.

59 `itc=0 A`Characteristic current for Tfc.

60 `itc2=0 A`Characteristic current for Tfc.

61 `vtc=1e3 V`Characteristic voltage for Tfc.

62 `tkrk=0 s`Forward transit time for Kirk effect.

63 `vkrk=1e3 V`Characteristic voltage for Kirk effect.

64 `ikrk=1e3 A`Characteristic voltage for Kirk effect.

65 `tr=0 s`Reverse charge storage time for intrinsic B-C junction.

66 `trx=0 s`Reverse charge storage time for extrinsic B-C junction.

67 `fex=0 s`Factor to determine excess phase.

Temperature Effects Parameters

68 `selft=no`Flag denoting self-heating.
Possible values are `no` or `yes`.

69 `tnom (C)`Parameters measurement temperature. Default set by options.

70 `top=27 C`Average device operating temperature.

71 `rth=0 Ω`Thermal resistance of device.

72 `cth=0 F`Thermal capacitance of device.

73 `xti=2`Exponent for `is` temperature dependence.

74 `xtb=2`Exponent for beta temperature dependence.

75 `tne=0`Coefficient for `ne` temperature dependence.

76 `tnc=0`Coefficient for `nc` temperature dependence.

77 `tnex=0`Coefficient for `nex` temperature dependence.

78 `eae=0 V`Activation energy for `isa` temperature dependence.

79 `eac=0 V`Activation energy for `isb` temperature dependence.

80 `eaa=0 V`Activation energy for `ise` temperature dependence.

81 `eab=0 V`Activation energy for `isc` temperature dependence.

82 `eax=0 V`Activation energy for `isex` temperature dependence.

83 `xre=0`Exponent for `re` temperature dependence.

84 `xrex=0`Exponent for `rex` temperature dependence.

85 `xrb=0`Exponent for `rb` temperature dependence.

86 `xrc=0`Exponent for `rc` temperature dependence.

87 `tvje=0 V/C`Coefficient for `vje` temperature dependence.

88 `tvjc=0 V/C`Coefficient for `vjc` temperature dependence.

89 `tvjcx=0 V/C`Coefficient for `vjcx` temperature dependence.

90 `tvjs=0 V/C`Coefficient for `vjs` temperature dependence.

91 `xtitc=0`Exponent for `itc` temperature dependence.

92 `xtitc2=0`Exponent for `itc2` temperature dependence.

93 `xttf=0`Exponent for `tf` temperature dependence.

94 `xttkrk=0`Exponent for `tkrk` temperature exponent.

95 `xtvkrk=0`Exponent for `vkrk` temperature dependence.

96 `xtikrk=0`Exponent for `ikrk` temperature dependence.

97 `xrt=0`Exponent for `rth` temperature dependence.

98 `eg=1.5 V`Activation energy for `is` temperature dependence.

99 `dtmax=1000 C`Maximum expected temperature rise above heat sink.

Noise Model Parameters

100 `kfn=0`Flicker (1/f) noise coefficient.

101 `afn=1`Flicker (1/f) noise exponent.

102 `bfn=1`Flicker noise frequency exponent.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

Operating-Point Parameters

1 `type=npn`Transistor type.
Possible values are `npn` or `pnp`.

2 `region=fwd`Estimated operating region.
Possible values are `off`, `fwd`, `rev`, or `sat`.

3 `vbe (V)`Base-emitter voltage.

4 `vbc (V)`Base-collector voltage.

5 `vce (V)`Collector-emitter voltage.

6 `vcs (V)`XC-substrate voltage.

7 `temp (C)`Device temperature.

8 `ith (A)`Thermal source.

9 `ice (A)`Intrinsic B-C current.

10 `ibe (A)`Intrinsic B-E current.

11 `ics (A)`C-S junction current.

12 `ibei (A)`B-E junction current.

13 `ibci (A)`B-C junction current.

14 `ibex (A)`XB-E junction current.

15 `ibcx (A)`XB-C junction current.

16 `ibk (A)`Breakdown current.

17 `dice_dvbe (S)`Intrinsic dIce/dVbe.

18 `dice_dvbc (S)`Intrinsic dIce_dVbc.

19 `dibe_dvbe (S)`Intrinsic dIbe_dVbe.

20 `dibe_dvbc (S)`Intrinsic dIbe_dVbc.

21 `dqbe_dvbe (F)`Intrinsic dQbe_dVbe.

22 `dqbe_dvbc (F)`Intrinsic dQbe_dVbc.

23 `dqbc_dvbe (F)`Intrinsic dQbc_dVbe.

24 `dqbc_dvbc (F)`Intrinsic dQbc_dVbc.

25 `cbcx (F)`XB-C junction capacitance.

26 `cbcxx (F)`EXTB-C junction capacitance.

27 `ccs (F)`Substrate junction capacitance.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

HV MOS Transistor (hvmos)

Description

HV (High-Voltage) MOS transistor model is a deep submicron, high-voltage MOSFET model. It is based on the BSIM3v3 version 3.1. Major enhancements include current-crowding effect at high gate bias, asymmetric source-drain structure, mobility reduction, transconductance reduction under high Vgs at saturation region, forward and reverse mode, self-heating, and more flexible gate-dependent output characteristics. HVMOS can be used for high voltage IC design applications such as Flash memory with asymmetric LDD structures, LCD drivers, CCD, E2PROM and LDMOS applications.

Like BSIM3v3, the HVMOS transistor model also allows the binning option to achieve even higher accuracy. The binning equation is given by

```
P = P0 + Pl / Leff + Pw / Weff + Pp / (Leff * Weff)
```

Only the P0 parameters are listed. Pl, Pw, and Pp 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. HVMOS transistors require that you use a model statement.

This device is supported within altergroups.

Sample Instance Statement

```
m1 (1 2 0 0) hvnmos w=1.5u l=1u ad=2.6p as=2.6p pd=6.6p ps=6.6p nrd=1.54 nrs=1.54
```

Sample Model Statement

```
model hvnmos hvmos vtho=0.53 w0=2.14e-6 nlx=1.8e-7 nch=2.3e18 xj=0.22e-6 k1=0.48
k2=-0.02 drout=1.1 rsh=10 cgso=2.4e-10 cgdo=2.4e-10
```

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as (m`^{2}`)`Area of source diffusion.

4 `ad (m`^{2}`)`Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `ld (m)`Length of drain diffusion region.

10 `ls (m)`Length of source diffusion region.

11 `m=1`Multiplicity factor (number of MOSFETs in parallel).

12 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, or `subth`.

13 `trise`Temperature rise from ambient.

Model Definition

```
model modelName hvmos parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

Threshold Voltage Parameters

2 `vtho (V)`Threshold voltage at zero body bias for long-channel devices. For enhancement-mode devices, vtho > 0 for n-channel and vtho < 0 for p-channel.

3 `k1=0.5` $\sqrt{V}$ Body-effect coefficient.

4 `k2=-0.0186`Charge-sharing parameter.

5 `k3=80`Narrow width coefficient.

6 `k3b=0 1/V`Narrow width coefficient.

7 `w0=2.5e-6 m`Narrow width coefficient.

8 `nlx=1.74e-7 m`Lateral nonuniform doping coefficient.

9 `dvt0=2.2`First coefficient of short-channel effects.

10 `dvt1=0.53`Second coefficient of short-channel effects.

11 `dvt2=-0.032 1/V`Body-bias coefficient of short-channel effects.

12 `a0f=1`Forward nonuniform depletion width effect coefficient.

13 `a0r=a0f`Reverse nonuniform depletion width effect coefficient.

14 `b0=0 m`Bulk charge coefficient due to narrow width effect.

15 `b1=0 m`Bulk charge coefficient due to narrow width effect.

16 `a1=0`No-saturation coefficient.

17 `a2=1`No-saturation coefficient.

18 `ags=0` F/m^2 VGate-bias dependence of Abulk.

19 `ketaf=-0.047 1/V`Body-bias coefficient for non-uniform depletion width effect.

20 `ketar=ketaf 1/V`Reverse body-bias coefficient for non-uniform depletion width effect.

Process Parameters

21 `nch=1.7e17 cm`$^{-3}$Peak channel doping concentration.

22 `xj=0.15e-6 m`Source/drain junction depth.

23 `lint=0 m`Lateral diffusion for one side.

24 `wint=0 m`Width reduction for one side.

25 `ll=0 m`Length dependence of delta L.

26 `lln=1`Length exponent of delta L.

27 `lw=0 m`Width dependence of delta L.

28 `lwn=1`Width exponent of delta L.

29 `lwl=0 m`^{2}Area dependence of delta L.

30 `lmin=0 m`The minimum channel length for which the model is still valid.

31 `lmax=1 m`The maximum channel length for which the model is still valid.

32 `wl=0 m`Length dependence of delta W.

33 `wln=1`Length exponent of delta W.

34 `ww=0 m`Width dependence of delta W.

35 `wwn=1`Width exponent of delta W.

36 `wwl=0 m`^{2}Area dependence of delta W.

37 `wmin=0 m`The minimum channel width for which the model is still valid.

38 `wmax=1 m`The maximum channel width for which the model is still valid.

39 `dwg=0 m/v`Gate-bias dependence of channel width.

40 `dwb=0 m/√v` Body-bias dependence of channel width.

41 `tox=1.5e-8 m`Gate oxide thickness.

42 `rd0=0` ΩFixed drain resistance.

43 `rs0=0` ΩFixed source resistance.

44 `rdw=0` Ω µmWidth dependence of drain resistance.

45 `rsw=0` Ω µmWidth dependence of source resistance.

46 `prwb=0 1/√v` Body-effect coefficient for Rds.

47 `prwg=0 1/√v` Gate-effect coefficient for Rds.

48 `wr=1`Width offset for parasitic resistance.

49 `binunit=2`Bin parameter unit selector. 1 for microns and 2 for meters.

Mobility Parameters

50 `mobmod=1`Mobility model selector.

51 `u0f=670` cm^2/V sForward low-field surface mobility at `tnom`. Default is 250 for PMOS.

52 `u0r=u0f` cm^2/V sReverse low-field surface mobility at `tnom`.

53 `vsatf=8e4 m/s`Forward carrier saturation velocity at `tnom`.

54 `dvsatf=0 m/s`Forward gate-bias dependence of saturation velocity.

55 `dvsatbf=0 m/s`Forward body-bias dependence of saturation velocity.

56 `vsatr=vsatf m/s`Reverse carrier saturation velocity at `tnom`.

57 `dvsatr=dvsatf m/s`Reverse gate-bias dependence of saturation velocity.

58 `dvsatbr=dvsatbf m/s`
Reverse body-bias dependence of saturation velocity.

59 `uaf=2.25e-9 m/v`Forward first-order mobility reduction coefficient.

60 `ubf=5.87e-19` m^2/v^2
Forward second-order mobility reduction coefficient.

61 `ucf=-4.65e-11 m/v`^{2}
Forward body-bias dependence of mobility. Default is -0.046 and unit is 1/V for mobmod=3.

62 `udf=0 m/v`^{2}Forward source-resistance dependence of mobility.

63 `uar=uaf m/v`Reverse first-order mobility reduction coefficient.

64 `ubr=ubf m`^{2}`/v`^{2}Reverse second-order mobility reduction coefficient.

65 `ucr=ucf m/v`^{2}Reverse body-bias dependence of mobility.

66 `udr=udf m/v`^{2}Reverse source-resistance dependence of mobility.

Output Resistance Parameters

67 `drout=0.56`DIBL effect on output resistance coefficient.

68 `pclmf=1.3`Forward channel length modulation coefficient.

69 `pclmr=pclmf`Reverse channel length modulation coefficient.

70 `pdiblc1f=0.39`Forward first coefficient of drain-induced barrier lowering.

71 `pdiblc1r=pdiblc1f`
Reverse first coefficient of drain-induced barrier lowering.

72 `pdiblc2f=8.6e-3`Forward second coefficient of drain-induced barrier lowering.

73 `pdiblc2r=pdiblc2f`
Reverse second coefficient of drain-induced barrier lowering.

74 `pdiblcbf=0 1/V`Body-effect coefficient for DIBL.

75 `pdiblcbr=pdiblcbf 1/V`
Reverse body-effect coefficient for DIBL.

76 `pscbe1f=4.24e8 V/m`
First coefficient of substrate current body effect.

77 `pscbe2f=1e-5 m/v`Second coefficient of substrate current body effect.

78 `pscbeg=0 V/m`Third coefficient of substrate current body effect.

79 `pscbe1r=pscbe1f V/m`
Reverse first coefficient of substrate current body effect.

80 `pscbe2r=pscbe2f m/v`
Reverse second coefficient of substrate current body effect.

81 `pclmgf=0`Forward gate dependence of Vaclm.

82 `pclmgr=pclmgf`Reverse gate dependence of Vaclm.

83 `pclmbf=0`Forward body dependence of Vaclm.

84 `pclmbr=pclmbf`Reverse body dependence of Vaclm.

85 `pdiblgf=0`Forward gate dependence of Vadibl.

86 `pdiblgr=pdiblgf`Reverse gate dependence of Vadibl.

87 `delta=0.01 V`Effective drain voltage smoothing parameter.

Subthreshold Parameters

88 `cdsc=2.4e-4 F/m`^{2}Source/drain and channel coupling capacitance.

89 `cdscb=0 F/m`^{2} VBody-bias dependence of `cdsc`.

90 `cdscd=0 F/m`^{2} VDrain-bias dependence of `cdsc`.

91 `nfactor=1`Subthreshold swing coefficient.

92 `cit=0 F`Interface trap parameter for subthreshold swing.

93 `voff=-0.08 V`Threshold voltage offset.

94 `dsub=drout`DIBL effect in subthreshold region.

95 `eta0f=0.08`DIBL coefficient subthreshold region.

96 `etabf=-0.07 1/V`Body-bias dependence of `et0`.

97 `eta0r=eta0f`Reverse DIBL coefficient subthreshold region.

98 `etabr=etabf 1/V`Body-bias dependence of `eta0r`.

Substrate current parameters

99 `alpha0=0 m/V`Substrate current impact ionization coefficient.

100 `beta0=30 1/V`Substrate current impact ionization exponent.

Parasitic Resistance Parameters

101 `rsh=0 Ω/sqr`Source/drain diffusion sheet resistance.

102 `rs=0 Ω`Source resistance.

103 `rd=0 Ω`Drain resistance.

104 `lgcs=0 m`Gate-to-contact length of source side.

105 `lgcd=0 m`Gate-to-contact length of drain side.

106 `rsc=0 Ω`Source contact resistance.

107 `rdc=0 Ω`Drain contact resistance.

108 `rss=0 Ω m`Scalable source resistance.

109 `rdd=0 Ω m`Scalable drain resistance.

110 `sc=∞ m`Spacing between contacts.

111 `ldif=0 m`Lateral diffusion beyond the gate.

112 `hdif=0 m`Length of heavily doped diffusion .

113 `minr=0.1 Ω`Minimum source/drain resistance.

Junction Diode Model Parameters

114 `js` (A/m^2)Bulk junction reverse saturation current density.

115 `is=1e-14 A`Bulk junction reverse saturation current.

116 n=1Junction emission coefficient.

117 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`. Possible values are `no` or `yes`.

118 `imelt=`imaxA'`Explosion current.

119 `jmelt=`jmaxA/m'`^{2}
Explosion current density.

Overlap Capacitance Parameters

120 `cgso (F/m)`Gate-source overlap capacitance.

121 `cgdo (F/m)`Gate-drain overlap capacitance.

122 `cgbo (F/m)`Gate-bulk overlap capacitance.

123 `meto=0 m`Metal overlap in fringing field.

124 `cgsl=0 F/m`Gate-source overlap capacitance in LDD region.

125 `cgdl=0 F/m`Gate-drain overlap capacitance in LDD region.

126 `ckappa=0.6`Overlap capacitance fitting parameter.

127 `deltaacc=0.1 V`Capacitance smoothing parameter.

Junction Capacitance Model Parameters

128 `cbs=0 F`Bulk-source zero-bias junction capacitance.

129 `cbd=0 F`Bulk-drain zero-bias junction capacitance.

130 `cj=5e-4 F/m`^{2}Zero-bias junction bottom capacitance density.

131 `mj=1/2`Bulk junction bottom grading coefficient.

132 `pb=0.8 V`Bulk junction built-in potential.

133 `fc=0.5`Forward-bias depletion capacitance threshold.

134 `cjsw=5e-10 F/m`Zero-bias junction sidewall capacitance density.

135 `mjsw=1/3`Bulk junction sidewall grading coefficient.

136 `pbsw=0.8 V`Side-wall junction built-in potential.

137 `fcsw=0.5`Side-wall forward-bias depletion capacitance threshold.

Charge Model Selection Parameters

138 `capmod=2`Intrinsic charge model.

139 `nqsmod=0`Non-quasi static model selector. Set to 1 to turn on nqs.

140 `dwc=wint m`Delta W for capacitance model.

141 `dlc=lint m`Delta L for capacitance model.

142 `clc=1e-7 m`Intrinsic capacitance fitting parameter.

143 `cle=0.6`Intrinsic capacitance fitting parameter.

144 `cf (F/m)`Fringe capacitance parameter.

145 `a0cvf=a0f`A0 for C-V calculation.

146 `a0cvr=a0r`Reverse A0 for C-V calculation.

147 `qgvd0f=1`Cgd fitting parameter.

148 `qgvd0r=qgvd0f`Reverse Cgd fitting parameter.

149 `elm=5`Elmore constant of the channel.

150 `xpart=0`Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

Default Instance Parameters

151 `w=5e-6 m`Default channel width.

152 `l=5e-6 m`Default channel length.

153 `as=0 m`^{2}Default area of source diffusion.

154 `ad=0 m`^{2}Default area of drain diffusion.

155 `ps=0 m`Default perimeter of source diffusion.

156 `pd=0 m`Default perimeter of drain diffusion.

157 `nrd=0 m/m`Default number of squares of drain diffusion.

158 `nrs=0 m/m`Default number of squares of source diffusion.

159 `xw=0 m`Width variation due to masking and etching.

160 `xl=0 m`Length variation due to masking and etching.

Temperature Effects Parameters

161 `tnom (C)`Parameters measurement temperature. Default set by `options`.

162 `trise=0 C`Temperature rise from ambient.

163 `tlev=0`DC temperature selector.

164 `tlevc=0`AC temperature selector.

165 `eg=1.12452 V`Energy band gap.

166 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

167 `gap2=1108 C`Band gap temperature offset.

168 `kt1=-0.11 V`Temperature coefficient for threshold voltage.

169 `kt1l=0 v m`Temperature coefficient for threshold voltage.

170 `kt2=0.022`Temperature coefficient for threshold voltage.

171 `atf=3.3e4 m/s`Temperature coefficient for `vsatf`.

172 `atr=atf m/s`Temperature coefficient for `vsatr`.

173 `at1f=0 m/s`Temperature coefficient for `dvsatf`.

174 `at1r=at1f m/s`Temperature coefficient for `dvsatr`.

175 ua1f=4.31e-9 m/vTemperature coefficient for uaf.

176 ub1f=-7.61e-18 m^2/v^2
Temperature coefficient for ubf.

177 uc1f=-5.5e-11 m/v^2
Temperature coefficient for ucf. Default is -0.056 for mobmod=3.

178 ud1f=0 m/v^2Temperature coefficient for udf.

179 ua1r=ua1f m/vTemperature coefficient for uar.

180 ub1r=ub1f m^2/v^2Temperature coefficient for ubr.

181 uc1r=uc1f m/v^2Temperature coefficient for ucr.

182 ud1r=0 m/v^2Temperature coefficient for udr.

183 rth=0 ΩSelf-heating thermal resistance.

184 rthg=0 1/VGate-effect coefficient for Rth.

185 rthb=0 $1/\sqrt{v}$ Body-effect coefficient for Rth.

186 prt=0 ΩTemperature coefficient for Rds.

187 trs=0 1/CTemperature parameter for source resistance.

188 trd=0 1/CTemperature parameter for drain resistance.

189 ute=-1.5Mobility temperature exponent.

190 xti=3Saturation current temperature exponent.

191 ptc=0 V/CSurface potential temperature coefficient.

192 tcv=0 V/CThreshold voltage temperature coefficient.

193 pta=0 V/CJunction potential temperature coefficient.

194 ptp=0 V/CSidewall junction potential temperature coefficient.

195 cta=0 1/CJunction capacitance temperature coefficient.

196 `ctp=0 1/C`Sidewall junction capacitance temperature coefficient.

Noise Model Parameters

197 `noimod=1`Noise model selector.

198 `kf=0`Flicker (1/f) noise coefficient.

199 `af=1`Flicker (1/f) noise exponent.

200 `ef=1`Flicker (1/f) noise frequency exponent.

201 `noia=1e20`Oxide trap density coefficient. Default is 9.9e18 for pmos.

202 `noib=5e4`Oxide trap density coefficient. Default is 2.4e3 for pmos.

203 `noic=-1.4e-8`Oxide trap density coefficient. Default is 1.4e-8 for pmos.

Operating Region Warning Control Parameters

204 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

205 `imax=1 A`Maximum allowable current.

206 `jmax=1e8 A/m`^{2}Maximum allowable current density.

207 `bvj=∞ V`Junction reverse breakdown voltage.

208 `vbox=1e9 tox V`Oxide breakdown voltage.

Cross-Term Dependent Parameters

The `imax` (`jmax`) parameter is used to aid convergence and prevent numerical overflow. The junction characteristics of the FET are accurately modeled for current (density) up to `imax` (`jmax`). For currents (density) above `imax` (`jmax`), the junction is modeled as a linear resistor and a warning is printed.

Output Parameters

1 `weff (m)`Effective channel width.

2 leff (m)Effective channel length.

3 rseff (Ω)Effective source resistance.

4 rdeff (Ω)Effective drain resistance.

Operating-Point Parameters

1 type=nTransistor type.
Possible values are n or p.

2 region=triodeEstimated operating region.
Possible values are off, triode, sat, or subth.

3 reversedReverse mode indicator.
Possible values are no or yes.

4 ids (A)Resistive drain-to-source current.

5 vgs (V)Gate-source voltage.

6 vds (V)Drain-source voltage.

7 vbs (V)Bulk-source voltage.

8 vth (V)Threshold voltage.

9 vdsat (V)Drain-source saturation voltage.

10 gm (S)Common-source transconductance.

11 gds (S)Common-source output conductance.

12 gmbs (S)Body-transconductance.

13 betaeff (A/V^2)Effective beta.

14 cjd (F)Drain-bulk junction capacitance.

15 cjs (F)Source-bulk junction capacitance.

16 cgg (F)Cgg.

17 `cgd` `(F)`Cgd.

18 `cgs` `(F)`Cgs.

19 `cgb` `(F)`Cgb.

20 `cdg` `(F)`Cdg.

21 `cdd` `(F)`Cdd.

22 `cds` `(F)`Cds.

23 `cdb` `(F)`Cdb.

24 `csg` `(F)`Csg.

25 `csd` `(F)`Csd.

26 `css` `(F)`Css.

27 `csb` `(F)`Csb.

28 `cbg` `(F)`Cbg.

29 `cbd` `(F)`Cbd.

30 `cbs` `(F)`Cbs.

31 `cbb` `(F)`Cbb.

32 `ron` `(Ω)`On-resistance.

33 `id` `(A)`Resistive drain current.

34 `ibulk` `(A)`Resistive bulk current.

35 `pwr` `(W)`Power at op point.

36 `gmoverid` `(1/V)`Gm/Ids.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Two Terminal Inductor (inductor)

Description

The inductance of this component can be a function of temperature or branch current. If you do not specify the inductance in the instance statement, it is taken from the model.

This device is supported within altergroups.

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

L(I) = L(inst) * (1 + c1 * I + c2 * I^2 + ...).

The branch flux as a function of current is

Flux(I) = L(inst) * I * (1 + 1/2 * c1 * I + 1/3 * c2 * 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) * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2].

whereT = trise(inst) + temp

if trise(inst) is given,

otherwise

T = trise(model) + temp

Sample Instance Statement

Without model:

```
l33 (0 net29) inductor l=10e-9 r=1 m=1
```

With model:

```
l33 (0 net29) ind l=10e-9 r=1 m=1
```

Sample Model Statement

```
model ind inductor l=6e-9 r=1 tc1=1e-12 tc2=1e-12 tnom=25
```

Instance Definition

```
Name  1  2 ModelName parameter=value ...
Name  1  2 inductor parameter=value ...
```

Instance Parameters

1 `l (H)`Inductance.

2 `r (Ω)`Resistance.

3 `m=1`Multiplicity factor.

4 `trise`Temperature rise from ambient.

5 `ic (A)`Initial condition.

6 `isnoisy=yes`Should inductor resistance generate noise.
Possible values are `no` or `yes`.

Model Definition

```
model modelName inductor parameter=value ...
```

Model Parameters

1 `l=0 H`Default inductance.

2 `r=0 Ω`Default resistance.

3 `tc1=0 1/C`Linear temperature coefficient.

4 `tc2=0 C`$^{-2}$Quadratic temperature coefficient.

5 `trise=0 C`Default `trise` value for instance.

6 `tnom (C)`Parameters measurement temperature. Default set by `options`.

7 `rforce=1e9 Ω`^{2}Resistance used when forcing nodesets and initial conditions.

8 `coeffs=[...]`Vector of polynomial inductance coefficients.

9 `scalei=1`Inductance scaling factor.

Noise Model Parameters

10 `kf=0`Flicker (1/f) noise coefficient.

11 `af=2`Flicker (1/f) noise exponent.

Output Parameters

1 `indeff (H)`Effective inductance.

Operating-Point Parameters

1 `ind (H)` Inductance at operating point.

2 `i (A)` Current at operating point.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

af M-11	indeff O-1	m I-3	tc1 M-3
coeffs M-8	isnoisy I-6	r M-2	tc2 M-4
i OP-2	kf M-10	r I-2	tnom M-6
ic I-5	l M-1	rforce M-7	trise I-4
ind OP-1	l I-1	scalei M-9	trise M-5

Interconnect Capacitance (intcap)

Description

Intcap is a model for the calculation of the interconnect capacitance, which takes into account the local layer composition and the tracks spacing width. It is described in the Philips MOST Modelbook (Dec.96) as INTCAP model.

(c) Philips Electronics N.V. 1993,1996

The model is extended by the device parameters `lxbelps`, `lxbelin` and `lxbelins`, according to a specification by H.Okel (I&A Hamburg).

This device is supported within altergroups.

This device is dynamically loaded from the shared object /vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so

Sample Instance Statement

intc (net9 net12) intconcap m=1 ael=2.5e-15 ain=2e-15 aps=1.8e-15

Sample Model Statement

model intconcap intcap cbps=1.5e-13 cebpsm=0.9e-15 cebpsi=0.83e-15 cbin=1.45e-13 cbins=1.4e-13

Instance Definition

Name `n1` n2 ModelName parameter=value ...

Instance Parameters

1 `m=1`Multiplicity factor.

2 `ael=0.0 m`^{2}The common area of `EL` track of the reference electrode.

3 `ain=0.0 m`^{2}The common area of `IN` track of the reference electrode.

4 `ains=0.0 m`^{2}The common area of `INS` track of the reference electrode.

5 `aps=0.0 m`^{2}The common area of `PS` track of the reference electrode.

6 `lbel=0.0 m`The sum of periphery length of `EL`-segments common to node n2 downwards.

7 `lbin=0.0 m`The sum of periphery length of `IN`-segments to node n2 downwards.

8 `lbins=0.0 m`The sum of periphery length of `INS`-segments common to node n2 downwards.

9 `lbps=0.0 m`The sum of periphery length of `PS`-segments common to node n2 downwards.

10 `lfbel=0.0 m`The sum of periphery length-factor products `EL` downwards.

11 `lfbin=0.0 m`The sum of periphery length-factor products `IN` downwards.

12 `lfbins=0.0 m`The sum of periphery length-factor products `INS` downwards.

13 `lfbps=0.0 m`The sum of periphery length-factor products `PS` downwards.

14 `lftel=0.0 m`The sum of periphery length-factor products `EL` upwards.

15 `lftin=0.0 m`The sum of periphery length-factor products `IN` upwards.

16 `lftins=0.0 m`The sum of periphery length-factor products `INS` upwards.

17 `lftps=0.0 m`The sum of periphery length-factor products `PS` upwards.

18 `ltel=0.0 m`The sum of periphery length of `EL`-segments common to node n2 upwards.

19 `ltin=0.0 m`The sum of periphery length of `IN`-segments common to node n2 upwards.

20 `ltins=0.0 m`The sum of periphery length of `INS`-segments common to node n2 upwards.

21 `ltps=0.0 m`The sum of periphery length of `PS`-segments common to node n2 upwards.

22 `ldsel=0.0 m`The sum of Li/Si quotients for `EL` tracks.

23 `ldsin=0.0 m`The sum of Li/Si quotients for `IN` tracks.

24 `ldsins=0.0 m`The sum of Li/Si quotients for `INS` tracks.

25 `ldsps=0.0 m`The sum of Li/Si quotients for `PS` tracks.

26 `lxbinps=0.0 m`The sum of Li/Si quotients for an `IN` track in parallel with an `PS` track.

27 `lxbinsin=0.0 m`The sum of Li/Si quotients for an `INS` track in parallel with an `IN` track.

28 `lxbinsps=0.0 m`The sum of Li/Si quotients for an `INS` track in parallel with an `PS` track.

29 `lxbelps=0.0 m`The sum of Li/Si quotients for an `EL` track in parallel with an `PS` track.

30 `lxbelin=0.0 m`The sum of Li/Si quotients for an `EL` track in parallel with an `IN` track.

31 `lxbelins=0.0 m`The sum of Li/Si quotients for an `EL` track in parallel with an `INS` track.

The Spectre option `scale`, default value is 1.0, scales the geometric

parameters. The actual areas (parameters starting with letter a) are equal

axxx * scale ^ 2

The actual lengths (parameters starting with letter `l`) are equal

lxxx * scale

Model Definition

model modelName intcap parameter=value ...

Model Parameters

1 `cbps=0.0 F/m`^{2}Bottom capacitance, `PS` to node n2.

2 `cebpsm=0.0 F/m`Edge to bottom capacitance (`PS`), 1.0um spacing.

3 `cebpsi=0.0 F/m`Edge to bottom capacitance (`PS`), single track.

4 `cetpsm=0.0 F/m`Edge to top capacitance (`PS`), 1.0um spacing.

5 `cetpsi=0.0 F/m`Edge to top capacitance (`PS`), single track.

6 `cbin=0.0 F/m`^{2}Bottom capacitance, `IN` to node n2.

7 `cebinm=0.0 F/m`Edge to bottom capacitance (`IN`), 1.0um spacing.

8 `cebini=0.0 F/m`Edge to bottom capacitance (`IN`), single track.

9 `cetinm=0.0 F/m`Edge to top capacitance (`IN`), 1.0um spacing.

10 `cetini=0.0 F/m`Edge to top capacitance (`IN`), single track.

11 `cbins=0.0 F/m`^{2}Bottom capacitance, `INS` to node n2.

12 `cebinsm=0.0 F/m`Edge to bottom capacitance (`INS`), 1.0um spacing.

13 `cebinsi=0.0 F/m`Edge to bottom capacitance (`INS`), single track.

14 `cetinsm=0.0 F/m`Edge to top capacitance (`INS`), 1.0um spacing.

15 `cetinsi=0.0 F/m`Edge to top capacitance (`INS`), single track.

16 `cbel=0.0 F/m`^{2}Bottom capacitance, `EL` to node n2.

17 `cebelm=0.0 F/m`Edge to bottom capacitance (`EL`), 1.0um spacing.

18 `cebeli=0.0 F/m`Edge to bottom capacitance (`EL`), single track.

19 `cetelm=0.0 F/m`Edge to top capacitance (`EL`), 1.0um spacing.

20 `ceteli=0.0 F/m`Edge to top capacitance (`EL`), single track.

21 `cecps=0.0 F/m`Lateral capacitance (`PS`), 1.0um spacing.

22 `cecin=0.0 F/m`Lateral capacitance (`IN`), 1.0um spacing.

23 `cecins=0.0 F/m`Lateral capacitance (`INS`), 1.0um spacing.

24 `cecel=0.0 F/m`Lateral capacitance (`EL`), 1.0um spacing.

Output Parameters

1 `cap (F)`Total Capacitance.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Current Probe (iprobe)

Description

Current through the probe is computed and is defined to be positive if it flows from the input node, through the probe, to the output node. The current variable is given the name of the `iprobe` instance, so you cannot create an `iprobe` with the same name as a circuit node.

This device is not supported within altergroup.

Sample Instance Statement

```
ip (1  0) iprobe
```

Instance Definition

```
Name  in  out iprobe
```

Independent Current Source (isource)

Description

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

I(T) = I(tnom) * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2].

Sample Instance Statement

```
i1 (in 0) isource dc=0 type=pulse delay=10n val0=0 val1=500u period=500n rise=1n
fall=1n width=250n
```

Instance Definition

```
Name  sink  src isource parameter=value ...
```

Positive current exits the source node and enters the sink node.

Instance Parameters

1 `dc=0` ADC value.

General Waveform Parameters

2 `type=dc`Waveform type.
Possible values are `dc`, `pulse`, `pwl`, `sine`, or `exp`.

3 `fundname`Name of the fundamental frequency. Must be specified if the source is active during a pdisto analysis or it is the active clock during an envlp analysis.

4 `delay=0 s`Waveform delay time.

Pulse Waveform Parameters

5 `val0=0 A`Zero value used in pulse and exponential waveforms.

6 `val1=1 A`One value used in pulse and exponential waveforms.

7 `period=∞ s`Period of waveform.

8 `rise (s)`Rise time for pulse waveform (time for transition from `val0` to `val1`).

9 `fall (s)`Fall time for pulse waveform (time for transition from `val1` to `val0`).

10 `width=∞ s`Pulse width (duration of `val1`).

PWL Waveform Parameters

11 `file`Name of file containing waveform.

12 `wave=[...]`Vector of time/value pairs that defines waveform.

13 `offset=0 A`DC offset for the PWL waveform.

14 `scale=1`Scale factor for the PWL waveform.

15 `stretch=1`Scale factor for time given for the PWL waveform.

16 `allbrkpts`All the points in the PWL waveform are breakpoints if set to yes. Default is yes if the number of points is less than 20.
Possible values are `no` or `yes`.

17 `pwlperiod (s)`Period of the periodic PWL waveform.

18 `twidth=pwlperiod/1000 s`
Transition width used when making PWL waveforms periodic.

Sinusoidal Waveform Parameters

19 `sinedc=dc A`DC level for sinusoidal waveforms.

20 `ampl=1 A`Peak amplitude of sinusoidal waveform.

21 `freq=0 Hz`Frequency of sinusoidal waveform.

22 `sinephase=0 °`Phase of sinusoid when `t=delay`.

23 `ampl2=1 A`Peak amplitude of second sinusoidal waveform.

24 `freq2=0 Hz`Frequency of second sinusoidal waveform.

25 `sinephase2=0 °`Phase of second sinusoid when `t=delay`.

26 `fundname2`Name of the fundamental frequency associated with freq2. Must be specified if freq2 is used in a pdisto analysis.

27 `fmmodindex=0`FM index of modulation for sinusoidal waveform.

28 `fmmodfreq=0 Hz`FM modulation frequency for sinusoidal waveform.

29 `ammodindex=0`AM index of modulation for sinusoidal waveform.

30 `ammodfreq=0 Hz`AM modulation frequency for sinusoidal waveform.

31 `ammodphase=0 °`AM phase of modulation for sinusoidal waveform.

32 `damp=0 1/s`Damping factor for sinusoidal waveform.

Exponential Waveform Parameters

33 `td1=0 s`Rise start time for exponential wave.

34 `tau1 (s)`Rise time constant for exponential wave.

35 `td2 (s)`Fall start time for exponential wave.

36 `tau2 (s)`Fall time constant for exponential wave.

Noise Parameters

37 `noisefile`Name of file containing excess spot noise data in the form of frequency-noise pairs.

38 `noisevec=[...] A`^{2}`/Hz`Excess spot noise as a function of frequency in the form of frequency-noise pairs.

Small Signal Parameters

39 `mag=0 A`Small signal current.

40 `phase=0 °`Small signal phase.

41 `xfmag=1 A/A`Transfer function analysis magnitude.

42 `pacmag=0 A`Periodic AC analysis magnitude.

43 `pacphase=0 °`Periodic AC analysis phase.

Multiplication Factor Parameters

44 `m=1`Multiplicity factor.

Temperature Effects Parameters

45 `tc1=0 1/C`First order temperature coefficient.

46 `tc2=0 C`$^{-2}$Second order temperature coefficient.

47 `tnom=27 C`Parameter measurement temperature. Default set by options.

If you do not specify the DC value, it is assumed to be the `time=0` value of the waveform.

In DC analyses, the only active parameters are `dc`, `m`, and the temperature coefficient parameters. In AC analyses, the only active parameters are `m`, `mag` and `phase`. In transient analyses, all parameters are active except the small signal parameters and the noise parameters. The `type` parameter selects which type of waveform is generated. You may specify parameters for more than one waveform type, and use the alter statement to change the waveform type between analyses.

A vector of time-value pairs describes the piecewise linear waveform. As an alternative, you can read the waveform from a file. In this case, you give time-value pairs one pair per line with a space or tab between the time and the value.

If you set `allbrkpts` to `yes`, you force the simulator to place time points at each point specified in a PWL waveform during a transient analysis. This can be very expensive for waveforms with many points. If you set `allbrkpts` to `no`, Spectre inspects the waveform, looking for abrupt changes, and forces time points only at those changes.

The PWL waveform is periodic if you specify `pwlperiod`. If the value of the waveform specified is not exactly the same at both its beginning its end, then you must provide a nonzero value `twidth`. Before repeating, the waveform changes linearly in an interval of `twidth` from its value at (`period` - `twidth`) to its value at the beginning of the waveform. Thus `twidth` must always be less than `period`.

You can give the excess noise of the source as a file or specify it with a vector of frequency-noise pairs. For a file, give the frequency-noise pairs one pair per line with a space or tab between the frequency and noise values.

Operating-Point Parameters

1 `i (A)`Current through the source.

2 `v (V)`Voltage across the source.

3 `pwr (W)`Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Junction Field Effect Transistor (jfet)

Description

The JFET model is derived from the FET model of Shichman and Hodges. JFETs require that you use a model statement.

This device is supported within altergroups.

Sample Instance Statement

```
jf1 (net1 net2 0) jmod area=1
```

Sample Model Statement

```
model jmod jfet beta=9e-5 lambda=0 type=n vt0=-18.7 rd=10 rs=10 cgs=1.3e-13 pb=0.65
```

Instance Definition

```
Name  d  g  s  [b] ModelName parameter=value ...
```

You do not have to specify the back gate terminal when you use the four-terminal model. If left unspecified, the substrate is connected to ground.

Instance Parameters

1 `area=1`Junction area factor.

2 `m=1`Multiplicity factor.

3 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

Model Definition

`model modelName jfet parameter=value ...`

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

Drain Current Model Parameters

2 `level=1`Drain current model level selector.

3 `vto=-2 V`Pinchoff voltage.

4 `beta=0.0001 A/V`^{2}Transconductance parameter.

5 `lambda=0 1/V`Channel length modulation parameter.

6 `lambda1=0 1/V`Gate dependence of channel length modulation parameter.

7 `np=2`Power-law exponent.

8 `alpha=2`Triode-to-saturation transition parameter.

9 `io=0 A`Subthreshold current parameter.

10 `ns=1`Subthreshold swing parameter.

11 `ai=0 1/V`Impact ionization current coefficient.

12 `bi=0 V`Impact ionization current exponent.

Four Terminal Threshold Voltage Parameters

13 `vtop=0.6 V`Back gate to channel junction potential.

14 `vtos=1.2 V`Threshold voltage slope.

15 `vtoe=0.33`Threshold voltage exponent.

16 `vtoc=-3.3 V`Threshold voltage constant.

Parasitic Resistance Parameters

17 `rd=0 Ω`Drain resistance (/area).

18 `rs=0 Ω`Source resistance (/area).

19 `rg=0 Ω`Gate resistance (/area).

20 `rb=0 Ω`Back gate resistance (/area).

21 `minr=0.1 Ω`Minimum source/drain/gate resistance.

Junction Diode Model Parameters

22 `is=1e-14 A`Gate saturation current (*area).

23 `n=1`Emission coefficient for G-D and G-S junctions.

24 `imelt=`imaxA'`Explosion current (*area).

25 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`.
Possible values are `no` or `yes`.

Junction Capacitance Model Parameters

26 `tt=0 s`Transit time.

27 `cgs=0 F`Gate-source zero-bias junction capacitance (*area).

28 `cgd=0 F`Gate-drain zero-bias junction capacitance (*area).

29 `mj=1/2`Junction grading coefficient.

30 `pb=1 V`Gate-junction potential.

31 `fc=0.5`Junction capacitor forward-bias threshold.

Four Terminal Junction Parameters

32 `isb=1e-14 A`Back gate-saturation current (*area).

33 `nb=1`Emission coefficient for back gate-junctions.

34 `cgbs=0 F`Back gate-source zero-bias junction capacitance (*area).

35 `cgbd=0 F`Back gate-drain zero-bias junction capacitance (*area).

36 `mjb=1/2`Back gate-junction grading coefficient.

37 `pbb=1 V`Back gate-junction potential.

Temperature Effect Parameters

38 `tnom (C)`Parameters measurement temperature. Default set by `options`.

39 `trise=0 C`Temperature rise from ambient.

40 `xti=3`Temperature exponent for effect on `is`.

41 `tlev=0`DC temperature selector.

42 `tlevc=0`AC temperature selector.

43 `eg=1.12452 V`Energy band gap.

44 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

45 `gap2=1108 C`Band gap temperature offset.

46 `tcv=0 1/C`Threshold voltage temperature coefficient.

47 `bto=0 C`Transconductance parameter temperature offset.

48 `bte=0`Transconductance parameter temperature exponent.

49 `lto=0 C`Channel length modulation parameters temperature offset.

50 `lte=0`Channel length modulation parameters temperature exponent.

51 `tc1=0 1/C`Linear temperature coefficient for parasitic resistors.

52 `tc2=0 C`$^{-2}$Quadratic temperature coefficient for parasitic resistors.

Operating Region Warning Control Parameters

53 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

54 `imax=1 A`Maximum allowable current (*area).

55 `bvj=∞ V`Junction reverse breakdown voltage.

Noise Parameters

56 `kf=0`Flicker noise (1/f) coefficient.

57 `af=1`Flicker noise (1/f) exponent.

58 `kfd=0`Flicker noise (1/f) coefficient for gate diodes.

59 `afg=1`Flicker noise (1/f) exponent for gate diodes.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

The `bv` parameter is used to detect the junction breakdown only. The breakdown currents of the junctions are not modeled.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are n or p.

2 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

3 `ids (A)`Resistive drain current.

4 `vgs (V)`Gate-source voltage.

5 `vds (V)`Drain-source voltage.

6 `vth (V)`Threshold at op point.

7 `vdsat (V)`Drain saturation voltage.

8 `gm (S)`Common-source transconductance.

9 `gds (S)`Common-source output conductance.

10 `cgs (F)`Gate-source capacitance.

11 `cgd (F)`Gate-drain capacitance.

12 `ig (A)`Resistive gate current.

13 `pwr (W)`Power at op point.

14 `qd (V)`Threshold at op point.

15 `qg (V)`Threshold at op point.

16 `qs (V)`Threshold at op point.

17 `qb (V)`Threshold at op point.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the

description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

af M-57	gap1 M-44	mj M-29	tc2 M-52
afg M-59	gap2 M-45	mjb M-36	tcv M-46
ai M-11	gds OP-9	n M-23	tlev M-41
alarm M-53	gm OP-8	nb M-33	tlevc M-42
alpha M-8	ids OP-3	np M-7	tnom M-38
area I-1	ig OP-12	ns M-10	trise M-39
beta M-4	imax M-54	pb M-30	tt M-26
bi M-12	imelt M-24	pbb M-37	type M-1
bte M-48	io M-9	pwr OP-13	type OP-1
bto M-47	is M-22	qb OP-17	vds OP-5
bvj M-55	isb M-32	qd OP-14	vdsat OP-7
cgbd M-35	kf M-56	qg OP-15	vgs OP-4
cgbs M-34	kfd M-58	qs OP-16	vth OP-6
cgd OP-11	lambda M-5	rb M-20	vto M-3
cgd M-28	lambda1 M-6	rd M-17	vtoc M-16
cgs M-27	level M-2	region OP-2	vtoe M-15
cgs OP-10	lte M-50	region I-3	vtop M-13
dskip M-25	lto M-49	rg M-19	vtos M-14
eg M-43	m I-2	rs M-18	xti M-40
fc M-31	minr M-21	tc1 M-51	

Junction Capacitor (juncap)

Description

The juncap model is intended to describe the behavior of the diodes that are formed by the source, drain or well-to-bulk junctions in MOS devices. It is described in the Philips MOST Modelbook (Dec.93) as JUNCAP model. Information on how to obtain this document can be found on Source Link by searching for Philips.

In extension to the modelbook description a minimum conductance `gmin` is inserted between the juncap nodes, to aid convergence. The value of `gmin` is set by an options statement, default = 1e-12 S.

The `imax` parameter is used to aid convergence and to prevent numerical overflow. The junction characteristics of the junction capacitor are accurately modeled for currents up to `imax`. For currents above `imax`, the junction is modeled as a linear resistor and a warning is printed.

This device is supported within altergroups.

This device is dynamically loaded from the shared object /vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so

Sample Instance Statement

c2 (1 2) capmod ab=7e-12 lg=5e-6 region=rev

Sample Model Statement

model capmod juncap type=n cjbr=0.2 cjgr=0.2 cjsr=0.2 tref=25 jsgbr=2e-3 jsdbr=0.28e-3 jsggr=1e-5 jsdgr=0.33e-6 vdsr=0.8 vdgr=0.8 vdbr=0.8

Instance Definition

Name `n` [b] ModelName parameter=value ...

Instance Parameters

1 `ab=1.0 scale`^{2} `m`^{2}
Diffusion area. Scale set by option scale.

2 `ls=1.0 scale m`Length of the sidewall of the diffusion area ab which is not under the gate. Scale set by option scale.

3 `lg=1.0 scale m`Length of the sidewall of the diffusion area ab which is under the gate. Scale set by option scale.

4 `m=1.0`Multiplicity factor.

5 `region=rev`Estimated DC operating region, used as a convergence aid. Possible values are `fwd` or `rev`.

Model Definition

model modelName juncap parameter=value ...

Model Parameters

Structural parameters

1 `type=n` Type of the juncap device.
Possible values are `n` or `p`.

2 `vb (V)` Not used for juncap model.

3 `bv (V)` Alias of `vb`.

4 `level` Not used for juncap model.

Current parameters

5 `jsgbr=1.0e-3 A/m`^{2}
Bottom saturation-current density due to electron-hole generation at reference voltage.

6 `jsdbr=1.0e-3 A/m`^{2}
Bottom saturation-current density due to diffusion from back contact.

7 `jsgsr=1.0e-3 A/m` Sidewall saturation-current density due to electron-hole generation at reference voltage.

8 `jsdsr=1.0e-3 A/m` Sidewall saturation-current density due to diffusion from back contact.

9 `jsggr=1.0e-3 A/m` Gate edge saturation-current density due to electron-hole generation at reference voltage.

10 `jsdgr=1.0e-3 A/m` Gate edge saturation-current density due to diffusion from back contact.

11 `imax=1.0 A` Explosion current.

Temperature effects parameters

12 `dta=0.0 K` Temperature offset of the juncap element with respect to ambient temperature.

13 `trise=0.0 K`Alias of `dta`.

14 `tr (C)`Temperature at which the parameters have been determined. Default set by option `tnom`.

15 `tref (C)`Alias of tr. Default set by option `tnom`.

16 `tnom (C)`Alias of tr. Default set by option `tnom`.

Junction capacitance parameters

17 `cjbr=1.0e-12 F/m`^{2}
Bottom junction capacitance at reference voltage.

18 `cjsr=1.0e-12 F/m`Sidewall junction capacitance at reference voltage.

19 `cjgr=1.0e-12 F/m`Gate edge junction capacitance at reference voltage.

Emission coefficient parameters

20 `nb=1.0`Emission coefficient of the bottom forward current.

21 `ns=1.0`Emission coefficient of the sidewall forward current.

22 `ng=1.0`Emission coefficient of the gate-edge forward current.

Voltage parameters

23 `vr=0.0 V`Voltage at which parameters have been determined.

24 `vdbr=1.0 V`Diffusion voltage of the bottom junction at reference temperature.

25 `vdsr=1.0 V`Diffusion voltage of the sidewall junction at reference temperature.

26 `vdgr=1.0 V`Diffusion voltage of the gate edge junction at reference temperature.

Grading coefficient parameters

27 `pb=0.4`Bottom-junction grading coefficient.

28 `ps=0.4`Sidewall-junction grading coefficient.

29 `pg=0.4`Gate edge-junction grading coefficient.

Output Parameters

1 `cjb (F)`Capacitance of bottom area `ab`.

2 `cjs (F)`Capacitance of locos-edge `ls`.

3 `cjg (F)`Capacitance of gate-edge `lg`.

4 `isdb (A)`Diffusion saturation-current of bottom area `ab`.

5 `isds (A)`Diffusion saturation-current of locos-edge `ls`.

6 `isdg (A)`Diffusion saturation-current of gate-edge `lg`.

7 `isgb (A)`Generation saturation-current of bottom area `ab`.

8 `isgs (A)`Generation saturation-current of locos-edge `ls`.

9 `isgg (A)`Generation saturation-current of gate-edge `lg`.

10 `vdb (V)`Diffusion voltage of bottom area `ab`.

11 `vds (V)`Diffusion voltage of locos-edge `ls`.

12 `vdg (V)`Diffusion voltage of gate-edge `lg`.

Operating-Point Parameters

1 `v (V)`Diode bias voltage (v = va - vk).

2 `i (A)`Total resistive current from anode to cathode (i = ia = -ik).

3 `gm (S)`Total differential conductance.

4 `q (Coul)`Total junction charge (q = qa = -qk).

5 `c (F)`Total capacitance.

6 `pwr (W)`Power.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

MISN Field Effect Transistor (misnan)

Description

The MISN model is formulated in terms of solutions for the boundary surface potentials of the channel and has the inherent property of continuous modeling. It is an inhouse MOSFET model of NORTEL. The MISN model requires a model statement.

This device is not supported within altergroup.

This device is dynamically loaded from the shared object /vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libnortel.so

Sample Instance Statement

mn1 (1 2 0 0) nch w=1.5u l=1u ad=2.6p as=2.6p pd=6.6p ps=6.6p

Sample Model Statement

model nch misnan type=n cox=4.4e-6 dop=2e17 phi=-0.43 xj=0.23 scrat=1.4 mu=400 rws=250 is=0.98e-13 cjgo=2e-13 noimdl=1

Instance Definition

Name `d g s b` ModelName parameter=value ...

Instance Parameters

1 `w=1e-5 m`Channel width.

2 `l=3e-6 m`Channel length.

3 `as=3e-11` m^2Area of source diffusion.

4 `ad=3e-11` m^2Area of drain diffusion.

5 `ps=2.6e-5 m`Perimeter of source diffusion.

6 `pd=2.6e-5 m`Perimeter of drain diffusion.

7 `m=1`Multiplicity factor (number of MOSFETs in parallel).

8 `region=triode`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `triode`, `sat`, or `subthresh`.

Model Definition

model modelName misnan parameter=value ...

Model Parameters

Intrinsic MOS parameters

1 `type=n`Transistor gender.
Possible values are n or p.

2 `cox=4.309e-7 F/cm`^{2}
Gate oxide cap per unit area.

3 `dop=1.665e17 cm`$^{-3}$
Substrate doping. Default = 2.58e17 for pmos.

4 `phi=-0.55 V`Gate Fermi potential.

5 `qss=-5.078e-8 Coul/cm`^{2}
Effective gate oxide charge per unit area. Default = 1.05e-8 for pmos.

6 `dopldd=3.2e17 cm`$^{-3}$
LDD region doping concentration. Default = 3.2e19 for pmos.

Geometry parameters

7 `lvar=0` μmGate length correction.

8 `wvar=0` μmGate width correction.

9 `dls=0.0273` μmSideway diffusion length of source region. Default = 0.037 for pmos.

10 `dld=0.0273` μmSideway diffusion length of drain region. Default = 0.037 for pmos.

11 `dl=0.07` μmSideways diffusion length of S/D regions. Default = 0.04 for pmos.

12 `dw=0.032` μmElectrical channel width correction. Default = 0.018 for pmos.

Threshold voltage parameters

13 `xj=0.24` μmSource/drain-to-substrate junction depth. Default = 0.31 for pmos.

14 `scrat=1.5`Short channel threshold voltage ratio. Default = 0.7 for pmos.

15 `scind=1.45`Short channel threshold voltage index. Default = 1.42 for pmos.

16 `ncrat=0.17`Narrow channel threshold voltage ratio. Default = 0.095 for pmos.

17 `athp=7.5`Factor controlling peak magnitude effect. Default = 3.5 for pmos.

18 `athl=2e4 1/cm`Factor controlling channel length dependence effect. Default = 4e4 for pmos.

19 `athb=-1.7e-3`Factor controlling substrate bias dependence effect. Default = -6e-3.

Mobility parameters

20 `mu=577 cm`^{2}`/V s`Low-field carrier mobility. Default = 120 for pmos.

21 `mutxp=1.72`Temperature coefficient for the carrier mobility. Default = 1.01 for pmos.

22 `kg=1.4e-7 cm/V`Gate field factor. Default = 1.685e-7.

23 `v0=3.21e7 cm/s`Scattering limited velocity. Default = 2.45e7.

24 `v0txp=-6.3`Temp coefficient for scattering limited velocity. Default = -5 for pmos.

25 `find=1.25`Field mobility index factor. Default = 1.9 for pmos.

26 `gfc=9.1e-10`Gate voltage dependence of enhanced gate-field scattering. Default = 1.05e-10 for pmos.

27 `gfcm=3e-5`Drain voltage dependence of enhanced gate-field scattering. Default = 2.3e-3 for pmos.

28 `gfmb=1.45e-3`Factor controlling substrate bias dependence of enhanced gate-field scattering. Default 3.3e-3 for pmos.

29 `csf=1.06e-11`Drain voltage dependence of coulombic scattering. Default = 1.35e-12 for pmos.

30 `csfb=1.61e-3`Body voltage dependence of coulombic scattering. Default = 8.5e-3 for pmos.

Saturation parameters

31 `dprat=15`Drain region/channel doping ration. Default = 2 for pmos.

32 `satpr=0.2`Saturation region shaping factor. Default = 1.0 for pmos.

33 `sbdr=0.3535534`Primary parameter controlling the onset of saturation.

34 `sadr=5`Secondary parameter controlling the onset of saturation.

Capacitance parameters

35 `sccf=0.25`Inner fringing factor for the N+ S/D.

Extrinsic parameters

36 `rws=480` Ω µmSource series resistance. Default = 1180 for pmos.

37 `rwd=480` Ω µmDrain series resistance. Default = 1180 for pmos.

38 `rsd=-1` Ω µmDrain/source series resistance. Negative value for asymmetrical devices.

39 `rgsh=0` Ω µmGate series resistance.

40 `wtgf=0.28` µmWidth of transition from gate to field oxide under poly.

41 `cpts=5.7e-9 F/cm`^{2}
Poly-to-substrate capacitance per unit area.

42 `cgfrs=1e-12 F/cm`Gate-source overlap fringing field capacitance.

43 `cgfrd=1e-12 F/cm`Gate-drain overlap fringing field capacitance.

44 `cgfr=1.36e-12 F/cm`
Gate overlap fringing field capacitance.

Junction parameters

45 `is=1.02e-12 A/cm`^{2}
Sat current per unit area of S/D region-injection component. Default = 9.21e-13 for pmos.

46 `isg=1e-20 A/cm`Sat current per unit length of gate oxide periphery-injection component. Default = 1.17e-20.

47 `isf=1e-20 A/cm`Sat current per unit length of field oxide periphery-injection component. Default = 1.17e-20.

48 `ig=1.31e-10 A/cm`^{2}
Sat current per unit area of S/D region-generation component. Default = 8.27e-10.

49 `igg=6.99e-14 A/cm`Sat current per unit length of gate oxide periphery-generation/ recombination component. Default = 6.47e-14.

50 `igf=6.99e-14 A/cm`Sat current per unit length of field oxide periphery-generation/ recombination component. Default = 6.47e-14.

51 `cjo=9.39e-8 F/cm`^{2}
Zero bias junction capacitance per unit area. Default = 1.273e-7 for pmos.

52 `ena=0.387`Junction capacitance coefficient for the area component. Default = 0.472 for pmos.

53 `cjgo=2.085e-12 F/cm`
Zero bias junction cap per unit length of gate oxide periphery. Default = 1.864e-12 for pmos.

54 `eng=0.322`Junction cap coefficient for gate oxide periphery component. Default = 0.334 for pmos.

55 `cjfo=3.037e-12 F/cm`
Zero bias junction cap per unit length of field oxide periphery. Default = 3.077e-12 for pmos.

56 `enf=0.322`Junction cap coefficient for field oxide periphery component. Default = 0.334 for pmos.

Noise parameters

57 `noimdl=1`Noise model selector.

58 `nt=1.6e10 cm`$^{-2}$Surface trap density. Default = 4e9 for pmos.

59 `nttx=-4`Surface trap density temperature coefficient.

60 `fidx=0.85`Flicker noise frequency coefficient.

61 `beta=1`Thermal noise proportional constant.

62 `sgma=3e-16`Capture cross section. Default = 3e-15 for pmos.

63 `xtau=1e-8`1/E depth.

64 `wbar=1`Barrier height for tunneling. Default = 4 for pmos.

65 `dept=3e-7`Depth of trap distribution.

Operating-Point Parameters

1 `vgs (V)`Gate-source voltage.

2 `vds (V)`Drain-source voltage.

3 `vbs (V)`Bulk-source voltage.

4 `id (A)`Drain current.

5 `vth (V)`Threshold voltage.

6 `vdsat (V)`Drain-source saturation voltage.

7 `gm (S)`Common-source transconductance.

8 `gd (S)`Common-source output conductance.

9 `gs (S)`Body-transconductance.

10 `gmb (S)`Body transconductance.

11 `gjs (S)`Drain-bulk junction conductance.

12 `ibs (A)`Drain-bulk junction current.

13 `gjd (S)`Source-bulk junction conductance.

14 `ibd (A)`Source-bulk junction current.

15 `qgg (Coul)`Gate charge.

16 `qss (Coul)`Source charge.

17 `qdd (Coul)`Drain charge.

18 `qbb (Coul)`Bulk charge.

19 `cgg (F)`Cgg.

20 `cgs (F)`Cgs.

21 `cgd (F)`Cgd.

22 `cgb (F)`Cgb.

23 `csg (F)`Csg.

24 `css (F)`Css.

25 `csd (F)`Csd.

26 `csb (F)`Csb.

27 `cdg (F)`Cdg.

28 `cds (F)`Cds.

29 `cdd (F)`Cdd.

30 `cdb (F)`Cdb.

31 `cbg (F)`Cbg.

32 `cbs (F)`Cbs.

33 `cbd (F)`Cbd.

34 `cbb (F)`Cbb.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the

description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

ad	I-4	csd	OP-25	ibd	OP-14	region	I-8
as	I-3	csf	M-29	ibs	OP-12	rgsh	M-39
athb	M-19	csfb	M-30	id	OP-4	rsd	M-38
athl	M-18	csg	OP-23	ig	M-48	rwd	M-37
athp	M-17	css	OP-24	igf	M-50	rws	M-36
beta	M-61	dept	M-65	igg	M-49	sadr	M-34
cbb	OP-34	dl	M-11	is	M-45	satpr	M-32
cbd	OP-33	dld	M-10	isf	M-47	sbdr	M-33
cbg	OP-31	dls	M-9	isg	M-46	sccf	M-35
cbs	OP-32	dop	M-3	kg	M-22	scind	M-15
cdb	OP-30	dopldd	M-6	l	I-2	scrat	M-14
cdd	OP-29	dprat	M-31	lvar	M-7	sgma	M-62
cdg	OP-27	dw	M-12	m	I-7	type	M-1
cds	OP-28	ena	M-52	mu	M-20	v0	M-23
cgb	OP-22	enf	M-56	mutxp	M-21	v0txp	M-24
cgd	OP-21	eng	M-54	ncrat	M-16	vbs	OP-3
cgfr	M-44	fidx	M-60	noimdl	M-57	vds	OP-2
cgfrd	M-43	find	M-25	nt	M-58	vdsat	OP-6
cgfrs	M-42	gd	OP-8	nttx	M-59	vgs	OP-1
cgg	OP-19	gfc	M-26	pd	I-6	vth	OP-5
cgs	OP-20	gfcm	M-27	phi	M-4	w	I-1
cjfo	M-55	gfmb	M-28	ps	I-5	wbar	M-64
cjgo	M-53	gjd	OP-13	qbb	OP-18	wtgf	M-40
cjo	M-51	gjs	OP-11	qdd	OP-17	wvar	M-8
cox	M-2	gm	OP-7	qgg	OP-15	xj	M-13
cpts	M-41	gmb	OP-10	qss	M-5	xtau	M-63
csb	OP-26	gs	OP-9	qss	OP-16		

MOS Level-0 Transistor (mos0)

Description

The MOS0 model is a simplified MOS level-1 model. The MOS0 DC drain current model is different from the Shichman and Hodges model because body effects are not modeled. The intrinsic MOS gate capacitances are replaced by the following linear overlap capacitances:

Gate to source/drain (`capmod` = `overlap`)

Gate to bulk (`capmod` = `bulk`)

Gate, source, and drain to ground (`capmod` = `gnd`)

MOS0 is usually used as a MOS switch. This model recognizes all the MOS and BSIM instance parameters but only uses `l` and `w`, ignoring all other parameters. MOS0 transistors require that you use a model statement.

This device is not supported within altergroup.

Sample Instance Statement

```
mp1 (0 1 2 2) pchmod0 l=2u w=30u ad=120p as=75p pd=36u ps=6u
```

Sample Model Statement

```
model pchmod0 mos0 type=p vto=-0.683 tox=0.21e-7 ld=0.45e-6 tnom=27
```

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `m=1`Multiplicity factor (number of MOSFETs in parallel).

Model Definition

```
model modelName mos0 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are `n` or `p`.

Drain Current Model Parameters

2 `vto=0 V`Threshold voltage at zero body bias.

3 `kp=2.0718e-5 A/V`^{2}
Transconductance parameter.

4 `lambda=0.02 1/V`Channel length modulation parameter.

5 `tox=1e-7 m`Gate oxide thickness.

6 `ld=0 m`Lateral diffusion.

7 `wd=0 m`Field-oxide encroachment.

Charge Model Selection Parameters

8 `capmod=gnd`Intrinsic charge model.
Possible values are `none`, `overlap`, `bulk`, or `gnd`.

Temperature Parameters

9 `tnom (C)`Parameters measurement temperature. Default set by `options`.

10 `trise=0 C`Temperature rise from ambient.

Default Device Parameters

11 `w=3e-6 m`Default channel width.

12 `l=3e-6 m`Default channel length.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are `n` or `p`.

2 `id (A)`Resistive drain current.

3 `vgs (V)`Gate-source voltage.

4 `vds (V)`Drain-source voltage.

5 `vbs (V)`Bulk-source voltage.

6 `vth (V)`Threshold voltage.

7 `vdsat (V)`Drain-source saturation voltage.

8 `gm (S)`Common-source transconductance.

9 `gds (S)`Common-source output conductance.

10 `cgs (F)`Gate-source capacitance.

11 `cgd (F)`Gate-drain capacitance.

12 `cgate (F)`Gate-Ground capacitance.

13 `ron (Ω)`On-resistance.

14 `pwr (W)`Power at op point.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

MOS Level-1 Transistor (mos1)

Description

The MOS1 model is derived from the FET model of Shichman and Hodges. The velocity saturation and the mobility variation effects can also be incorporated into MOS1. Three charge models are available. MOS1 transistors require that you use a model statement.

This device is supported within altergroups.

Sample Instance Statement

```
nch1 (1 2 0 0) nchmod1 l=2u w=15u ad=60p as=37.5p pd=23u ps=6u
```

Sample Model Statement

```
model nchmod1 mos1 vto=0.78 gamma=0.56 kp=0.8675e-4 tox=0.21e-7 nsub=0.21e17
ld=0.55e-6 capmod=yang vmax=4e5 theta=0.19 cbs=11e-15 cbd=10e-15 lambda=0.1
```

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as` (m^2)Area of source diffusion.

4 `ad` (m^2)Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `ld (m)`Length of drain diffusion region.

10 `ls (m)`Length of source diffusion region.

11 `m=1`Multiplicity factor (number of MOSFETs in parallel).

12 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

13 `trise`Temperature rise from ambient.

14 `degradation=no`Hot-electron degradation flag.
Possible values are `no` or `yes`.

Model Definition

```
model modelName mos1 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are `n` or `p`.

Drain Current Model Parameters

2 `vto=0 V`Threshold voltage at zero body bias.

3 `kp=2.0718e-5 A/V`^{2}
Transconductance parameter.

4 `lambda=0 1/V`Channel length modulation parameter.

5 `phi=0.7 V`Surface potential at strong inversion.

6 `gamma=0` $\sqrt{V}$ Body-effect parameter.

7 `uo=600 cm`^{2}`/V s`Carrier surface mobility.

8 `vmax=`∞ `m/s`Carrier saturation velocity.

9 `theta=0 1/V`Mobility modulation coefficient.

Process Parameters

10 `nsub=1.13e16 cm`$^{-3}$
Channel doping concentration.

11 `nss=0 cm`$^{-2}$Surface state density.

12 `nfs=0 cm`$^{-2}$Fast surface state density.

13 `tpg=+1`Type of gate (+1 = opposite of substate, -1 = same as substate, 0 = aluminum).

14 `ld=0 m`Lateral diffusion.

15 `wd=0 m`Field-oxide encroachment.

16 `xw=0 m`Width variation due to masking and etching.

17 `xl=0 m`Length variation due to masking and etching.

18 `tox=1e-7 m`Gate oxide thickness.

Impact Ionization Parameters

19 `ai0=0 1/V`Impact ionization current coefficient.

20 `lai0=0 µm/V`Length sensitivity of `ai0`.

21 `wai0=0 µm/V`Width sensitivity of `ai0`.

22 `bi0=0 V`Impact ionization current exponent.

23 `lbi0=0 µm V`Length sensitivity of `bi0`.

24 `wbi0=0 µm V`Width sensitivity of `bi0`.

Overlap Capacitance Parameters

25 `cgso=0 F/m`Gate-source overlap capacitance.

26 `cgdo=0 F/m`Gate-drain overlap capacitance.

27 `cgbo=0 F/m`Gate-bulk overlap capacitance.

28 `meto=0 m`Metal overlap in fringing field.

Charge Model Selection Parameters

29 `capmod=bsim`Intrinsic charge model.
Possible values are `none`, `meyer`, `yang`, or `bsim`.

30 `xpart=1`Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

31 `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, 0 for 0/100.

Parasitic Resistance Parameters

32 `rs=0 Ω`Source resistance.

33 `rd=0 Ω`Drain resistance.

34 `rss=0 Ω m`Scalable source resistance.

35 `rdd=0 Ω m`Scalable drain resistance.

36 `rsh=0 Ω/sqr`Source/drain diffusion sheet resistance.

37 `rsc=0 Ω`Source contact resistance.

38 `rdc=0 Ω`Drain contact resistance.

39 `minr=0.1 Ω`Minimum source/drain resistance.

40 `ldif=0 m`Lateral diffusion beyond the gate.

41 `hdif=0 m`Length of heavily doped diffusion.

42 `lgcs=0 m`Gate-to-contact length of source side.

43 `lgcd=0 m`Gate-to-contact length of drain side.

44 `sc=∞ m`Spacing between contacts.

Junction Diode Model Parameters

45 `js (A/m`^{2}`)`Bulk junction reverse saturation current density.

46 `is=1e-14 A`Bulk junction reverse saturation current.

47 `n=1`Junction emission coefficient.

48 `dskip=yes`Use simple piece-wise linear model for diode currents below `0.1*iabstol`. Possible values are `no` or `yes`.

49 `imelt=`imaxA'`Explosion current, diode is linearized beyond this current to aid convergence.

50 `jmelt=`jmaxA/m'`^{2}
Explosion current density, diode is linearized beyond this current to aid convergence.

Junction Capacitance Model Parameters

51 `cbs=0 F`Bulk-source zero-bias junction capacitance.

52 `cbd=0 F`Bulk-drain zero-bias junction capacitance.

53 `cj=0 F/m`^{2}Zero-bias junction bottom capacitance density.

54 `mj=1/2`Bulk junction bottom grading coefficient.

55 `pb=0.8 V`Bulk junction built-in potential.

56 `fc=0.5`Forward-bias depletion capacitance threshold.

57 `cjsw=0 F/m`Zero-bias junction sidewall capacitance density.

58 `mjsw=1/3`Bulk junction sidewall grading coefficient.

59 `pbsw=0.8 V`Side-wall junction built-in potential.

60 `fcsw=0.5`Side-wall forward-bias depletion capacitance threshold.

Operating Region Warning Control Parameters

61 alarm=noneForbidden operating region.
Possible values are none, off, triode, sat, subth, or rev.

62 imax=1 AMaximum current, currents above this limit generate a warning.

63 jmax=1e8 A/m^{2}Maximum current density, currents above this limit generate a warning.

64 bvj=∞ VJunction reverse breakdown voltage.

65 vbox=1e9 tox VOxide breakdown voltage.

Temperature Effects Parameters

66 tnom (C)Parameters measurement temperature. Default set by options.

67 trise=0 CTemperature rise from ambient.

68 uto=0 CMobility temperature offset.

69 ute=-1.5Mobility temperature exponent.

70 tlev=0DC temperature selector.

71 tlevc=0AC temperature selector.

72 eg=1.12452 VEnergy band gap.

73 gap1=7.02e-4 V/CBand gap temperature coefficient.

74 gap2=1108 CBand gap temperature offset.

75 f1ex=0Temperature exponent for ucrit.

76 lamex=0 1/CTemperature parameter for lambda and kappa.

77 trs=0 1/CTemperature parameter for source resistance.

78 trd=0 1/CTemperature parameter for drain resistance.

79 xti=3Saturation current temperature exponent.

80 `ptc=0 V/C`Surface potential temperature coefficient.

81 `tcv=0 V/C`Threshold voltage temperature coefficient.

82 `pta=0 V/C`Junction potential temperature coefficient.

83 `ptp=0 V/C`Sidewall junction potential temperature coefficient.

84 `cta=0 1/C`Junction capacitance temperature coefficient.

85 `ctp=0 1/C`Sidewall junction capacitance temperature coefficient.

Default Instance Parameters

86 `w=3e-6 m`Default channel width.

87 `l=3e-6 m`Default channel length.

88 `as=0 m`^{2}Default area of source diffusion.

89 `ad=0 m`^{2}Default area of drain diffusion.

90 `ps=0 m`Default perimeter of source diffusion.

91 `pd=0 m`Default perimeter of drain diffusion.

92 `nrd=0 m/m`Default number of squares of drain diffusion.

93 `nrs=0 m/m`Default number of squares of source diffusion.

94 `ldd=0 m`Default length of drain diffusion region.

95 `lds=0 m`Default length of source diffusion region.

Noise Model Parameters

96 `noisemod=1`Noise model selector.

97 `kf=0`Flicker (1/f) noise coefficient.

98 `af=1`Flicker (1/f) noise exponent.

99 `ef=1`Flicker (1/f) noise frequency exponent.

100 `wnoi=1e-5 m`Channel width at which noise parameters were extracted.

Auto Model Selector Parameters

101 `wmax=1.0 m`Maximum channel width for which the model is valid.

102 `wmin=0.0 m`Minimum channel width for which the model is valid.

103 `lmax=1.0 m`Maximum channel length for which the model is valid.

104 `lmin=0.0 m`Minimum channel length for which the model is valid.

Degradation Parameters

105 `degramod=spectre`Degradation model selector.
Possible values are `spectre` or `bert`.

106 `degradation=no`Hot-electron degradation flag.
Possible values are `no` or `yes`.

107 `dvthc=1 V`Degradation coefficient for threshold voltage.

108 `dvthe=1`Degradation exponent for threshold voltage.

109 `duoc=1 S`Degradation coefficient for transconductance.

110 `duoe=1`Degradation exponent for transconductance.

111 `crivth=0.1 V`Maximum allowable threshold voltage shift.

112 `criuo=10%`Maximum allowable normalized mobility change.

113 `crigm=10%`Maximum allowable normalized transconductance change.

114 `criids=10%`Maximum allowable normalized drain current change.

115 `wnom=5e-6 m`Nominal device width in degradation calculation.

116 `lnom=1e-6 m`Nominal device length in degradation calculation.

117 `vbsn=0 V`Substrate voltage in degradation calculation.

118 `vdsni=0.1 V`Drain voltage in `Ids` degradation calculation.

119 `vgsni=5 V`Gate voltage in `Ids` degradation calculation.

120 `vdsng=0.1 V`Drain voltage in `Gm` degradation calculation.

121 `vgsng=5 V`Gate voltage in `Gm` degradation calculation.

Spectre Stress Parameters

122 `esat=1.1e7 V/m`Critical field in Vdsat calculation.

123 `esatg=2.5e6 1/m`Gate voltage dependence of `esat`.

124 `vpg=-0.25`Gate voltage modifier.

125 `vpb=-0.13`Gate voltage modifier.

126 `subc1=2.24e-5`Substrate current coefficient.

127 `subc2=-0.1e-5 1/V`Substrate current coefficient.

128 `sube=6.4`Substrate current exponent.

129 `strc=1`Stress coefficient.

130 `stre=1`Stress exponent.

BERT Stress Parameters

131 `h0=1`Aging coefficient.

132 `hgd=0 1/V`Bias dependence of `h0`.

133 `m0=1`Aging exponent.

134 `mgd=0 1/V`Bias dependence of `m0`.

135 `ecrit0=1.1e5 V/cm`Critical electric field.

136 lecrit0=0 µm V/cmLength dependence of ecrit0.

137 wecrit0=0 µm V/cmWidth dependence of ecrit0.

138 ecritg=0 1/cmGate voltage dependence of ecrit0.

139 lecritg=0 µm/cmLength dependence of ecritg.

140 wecritg=0 µm/cmWidth dependence of ecritg.

141 ecritb=0 1/cmSubstrate voltage dependence of ecrit0.

142 lecritb=0 µm/cmLength dependence of ecritb.

143 wecritb=0 µm/cmWidth dependence of ecritb.

144 lc0=1Substrate current coefficient.

145 llc0=0 µmLength dependence of lc0.

146 wlc0=0 µmWidth dependence of lc0.

147 lc1=1Substrate current coefficient.

148 llc1=0 µmLength dependence of lc1.

149 wlc1=0 µmWidth dependence of lc1.

150 lc2=1Substrate current coefficient.

151 llc2=0 µmLength dependence of lc2.

152 wlc2=0 µmWidth dependence of lc2.

153 lc3=1Substrate current coefficient.

154 llc3=0 µmLength dependence of lc3.

155 wlc3=0 µmWidth dependence of lc3.

156 lc4=1Substrate current coefficient.

157 llc4=0 µmLength dependence of lc4.

158 `wlc4=0` µmWidth dependence of `lc4`.

159 `lc5=1`Substrate current coefficient.

160 `llc5=0` µmLength dependence of `lc5`.

161 `wlc5=0` µmWidth dependence of `lc5`.

162 `lc6=1`Substrate current coefficient.

163 `llc6=0` µmLength dependence of `lc6`.

164 `wlc6=0` µmWidth dependence of `lc6`.

165 `lc7=1`Substrate current coefficient.

166 `llc7=0` µmLength dependence of `lc7`.

167 `wlc7=0` µmWidth dependence of `lc7`.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

Both of these parameters have current density counterparts, `jmax` and `jmelt`, that you can specify if you want the absolute current values to depend on the device area.

Auto Model Selection

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 <= inst_length < lmax  and   wmin <= inst_width  < wmax
```

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)`Effective channel width.

2 `leff (m)`Effective channel length.

3 `rseff (Ω)`Effective source resistance.

4 `rdeff (Ω)`Effective drain resistance.

5 `aseff` (m^2)Effective area of source diffusion.

6 `adeff` (m^2)Effective area of drain diffusion.

7 `pseff (m)`Effective perimeter of source diffusion.

8 pdeff (m)Effective perimeter of source diffusion.

9 isseff (A)Effective source-bulk junction reverse saturation current.

10 isdeff (A)Effective drain-bulk junction reverse saturation current.

11 cbseff (F)Effective zero-bias source-bulk junction capacitance.

12 cbdeff (F)Effective zero-bias drain-bulk junction capacitance.

Operating-Point Parameters

1 type=nTransistor type.
Possible values are n or p.

2 region=triodeEstimated operating region.
Possible values are off, triode, sat, subth, or breakdown.

3 degradation=noHot-electron degradation flag.
Possible values are no or yes.

4 reversedReverse mode indicator.
Possible values are no or yes.

5 ids (A)Resistive drain-to-source current.

6 vgs (V)Gate-source voltage.

7 vds (V)Drain-source voltage.

8 vbs (V)Bulk-source voltage.

9 vth (V)Threshold voltage.

10 vdsat (V)Drain-source saturation voltage.

11 gm (S)Common-source transconductance.

12 gds (S)Common-source output conductance.

13 gmbs (S)Body-transconductance.

14 gameff ($\sqrt{V}$)Effective body effect coefficient.

15 `betaeff` `(A/V`^{2}`)`Effective `beta`.

16 `cbd (F)`Drain-bulk junction capacitance.

17 `cbs (F)`Source-bulk junction capacitance.

18 `cgs (F)`Gate-source capacitance.

19 `cgd (F)`Gate-drain capacitance.

20 `cgb (F)`Gate-bulk capacitance.

21 `ron (`Ω`)`On-resistance.

22 `id (A)`Resistive drain current.

23 `ibulk (A)`Resistive bulk current.

24 `pwr (W)`Power at op point.

25 `gmoverid (1/V)Gm/Ids`.

26 `isub (A)`Substrate current.

27 `stress`Hot-electron stress.

28 `age (s)`Device age.

29 `he_vdsat (V)`Hot Electron `Vdsat`.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Compact MOS-Transistor Distortion Model (mos1000)

Description

The mos10.00 model is an experimental model based on the thesis of Ronald van Langevelde: "A compact MOSFET Model for Distortion Analysis in Analog Circuit Design", Technische Universiteit Eindhoven, 1998.

Note: In noise analysis, mos10.00 instances will not generate any contribution, since there are no noise sources included (yet) in the mos10.00 model.

(c) Philips Electronics N.V. 1999

In extension to the description a minimum conductance `gmin` is inserted between the drain and source node, to aid convergence. The value of `gmin` is set by an options statement, default = 1e-12 S.

This device is supported within altergroups.

This device is dynamically loaded from the shared object
`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Instance Definition

```
Name  d  g  s  [b] ModelName parameter=value ...
```

Instance Parameters

1 `w=1.0 scale m`Drawn channel width in the lay-out. Scale set by option scale.

2 `l=1.0 scale m`Drawn channel length in the lay-out. Scale set by option scale.

3 `mult=1`Number of devices in parallel.

4 `area=1`Alias of `mult`.

5 `region=triode`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `triode`, `sat`, or `subth`.

6 `m=1`Multiplicity factor.

Model Definition

model modelName mos1000 parameter=value ...

Model Parameters

Device type parameters

1 `type=n`Transistor gender.
Possible values are n or p.

Geometry Parameters

2 `ler=1.0e-6 m`Effective channel length of the reference transistor.

3 `wer=1e-6 m`Effective channel width of the reference transistor.

4 `lvar=0.0 m`Difference between the actual and the programmed poly-silicon gate length.

5 `lap=45.0e-9 m`Effective channel length reduction per side.

6 `wvar=-5.0e-9 m`Difference between the actual and the programmed field-oxide opening.

7 `wot=50.0e-9 m`Effective channel width reduction per side.

Threshold-Voltage Parameters

8 `vfbr=-518.9e-03 V`Flat-band voltage for reference transistor.

9 `stvfb=-1.2e-03 V/K`
Coefficient of temperature dependence of `vfb`.

10 `slvfb=24.0e-09 V m`
Coefficient of length dependence of `vfb`.

11 `sl2vfb=-1.1e-15 V m`^{2}
Second coefficient of length dependence of `vfb`.

12 `swvfb=4.400e-09 V m`
Coefficient of the width dependency of `vfb`.

13 `kor=368.0e-03 √V`
Body effect coefficient for the reference transistor.

14 `slko=-8.240e-09 √V  m`
Coefficient of the length dependence of `ko`.

15 `sl2ko=-2.260e-15 √V  m`^{2}
Second coefficient of the length dependence of `ko`.

16 `swko=5.86e-09 √V  m`
Coefficient of the width dependence of `ko`.

17 `phibr=0.6 V`Surface potential at strong inversion.

Channel-Current Parameters

18 `betsq=370.9e-06 A/V`^{2}
Gain factor for an infinite square transistor.

19 `etabet=1.6`Exponent of the temperature dependence of the gain factor.

20 `thesrr=16.10e-3 1/V`^{2}
Mobility degradation parameter due to surface roughness scattering.

21 `stthesr=0.0 1/(V`^{2} `K)`
Coefficient of the temperature dependence of `thesr`.

22 `swthesr=0.0 1/(V`^{2} `m)`
Coefficient of the width dependence of `thesr`.

23 `thephr=0.055 1/V`Mobility degradation parameter due to phonon scattering.

24 `sttheph=0.0 1/(V K)`
Coefficient of the temperature dependence of `theph`.

25 `swtheph=0.0 1/(V m)`
Coefficient of the width dependence of `theph`.

26 `etamobr=1.6` Effective field parameter for dependence on depletion charge.

27 `swetamob=0.0 1/m` Coefficient of the width dependence of `etamobr`.

28 `thersq=0.155 1/V` Coefficient of gate voltage independent part of series resistance.

29 `swther=0.0 1/(V m)`
Coefficient of the width dependence of `ther`.

30 `ther1=0.0 V` Numerator of gate voltage independent part of series resistance.

31 `ther2=1.0 V` Denominator of gate voltage independent part of series resistance.

32 `thenr=0.480 1/V` Velocity saturation parameter due to optical phonon scattering.

33 `stthen=0.0 1/(V K)`
Coefficient of the temperature dependence of `then`.

34 `swthen=0.0 1/(V m)`
Coefficient of the width dependence of `then`.

35 `thepr=0.0 1/V` Velocity saturation parameter due to acoustic phonon scattering.

36 `stthep=0.0 1/(V K)`
Coefficient of the temperature dependence of `thep`.

37 `swthep=0.0 1/(V m)`
Coefficient of the width dependence of `thep`.

38 `gthep=1.0` Velocity saturation factor due to acoustic phonon scattering.

39 `thethr=3.227e-3` $1/V^3$
Coefficient of self-heating.

40 `sltheth=2.460e-9` $1/(V^3\ m)$
Coefficient of the length dependence of `theth`.

41 `swtheth=0.0` $1/(V^3\ m)$
Coefficient of the width dependence of `theth`.

Sub-threshold parameters

42 `sdiblo=2.030e-03 1/√V`
Drain-induced barrier lowering parameter.

43 `sdiblexp=1.340`Exponent of the length dependence of `sdibl`.

44 `dphi=0.800 V`Parameter for short-channel subtreshold behaviour.

Saturation Parameters

45 `ssfsq=6.250e-03 1/√V`
Static feedback parameter.

46 `swssf=0.0 1/(√V m)`
Coefficient of the width dependence of `ssf`.

47 `alpsq=0.010 m`Characteristic length parameter for channel length modulation.

48 `swalp=0.0 m`Coefficient of the width dependence of `alp`.

49 `vp=0.075 V`Characteristic voltage of channel-length modulation.

Smoothing Parameters

50 `mexpo=0.093`Smoothing factor.

51 `mexpl=0.065`Coefficient of the length dependence of `mexp`.

Weak-Avalanche Parameters

52 `a1r=6`Factor of the weak-avalanche current.

53 `sta1=0.0 1/K`Coefficient of the temperature dependence of `a1`.

54 `sla1=1.30e-6 m`Coefficient of the length dependence of `a1`.

55 `swa1=3.0e-06 m`Coefficient of the width dependence of `a1`.

56 `a2r=38.0 V`Exponent of the weak-avalanche current.

57 `sla2=1.00e-06 V m`Coefficient of the length dependence of `a2`.

58 swa2=2.00e-06 V mCoefficient of the width dependence of a2.

59 a3r=0.650Factor of the drain-source voltage above which weak-avalanche occurs.

60 sla3=-550.0e-06 mCoefficient of the length dependence of a3.

61 swa3=0.0 mCoefficient of the width dependence of a3.

Charge Parameters

62 tox=4.5e-09 mThickness of the oxide layer.

63 col=320e-12 F/mGate overlap capacitance per unit channel width.

Temperature Parameters

64 tr (C)Reference temperature. Default set by option tnom.

65 tref (C)Alias of tr. Default set by option tnom.

66 tnom (C)Alias of tr. Default set by option tnom.

67 dta=0.0 KTemperature offset of the device.

68 trise=0.0 KAlias of dta.

Output Parameters

1 le (m)Effective channel length.

2 we (m)Effective channel width.

3 vfb (V)Flat-band voltage.

4 ko ($\sqrt{V}$)Body effect coefficient.

5 phib (V)Surface potential at strong inversion.

6 bet (A/V^2)Gain factor.

7 thesr ($1/V^2$)Mobility degradation parameter due to surface roughness scattering.

8 `theph (1/V)`Mobility degradation parameter due to phonon scattering.

9 `etamob`Effective field parameter for dependence on depletion charge.

10 `ther (1/V)`Coefficient of gate voltage independent part of series resistance.

11 `ther1 (V)`Numerator of gate voltage independent part of series resistance.

12 `ther2 (V)`Denominator of gate voltage independent part of series resistance.

13 `then (1/V)`Velocity saturation parameter due to optical phonon scattering.

14 `thep (1/V)`Velocity saturation parameter due to acoustic phonon scattering.

15 `gthep`Velocity saturation factor due to acoustic phonon scattering.

16 `theth` $(1/V^3)$Coefficient of self-heating.

17 `sdibl` $(1/\sqrt{V})$Drain-induced barrier lowering parameter.

18 `dphi (V)`Parameter for short-channel subtreshold behaviour.

19 `ssf` $(1/\sqrt{V})$Static feedback parameter.

20 `alp (m)`Characteristic length parameter for channel length modulation.

21 `vp (V)`Characteristic voltage of channel-length modulation.

22 `mexp`Smoothing factor.

23 `phit (V)`Thermal voltage.

24 `a1`Factor of the weak-avalanche current.

25 `a2 (V)`Exponent of the weak-avalanche current.

26 `a3`Factor of the drain-source voltage above which weak-avalanche occurs.

27 `cox (F)`Gate-to-channel capacitance (* mult).

28 `cgdo (F)`Gate-drain overlap capacitance (* mult).

29 `cgso (F)`Gate-source overlap capacitance (* mult).

Operating-Point Parameters

1 `ide (A)`Resistive drain current.

2 `ige (A)`Resistive gate current.

3 `ise (A)`Resistive source current.

4 `ibe (A)`Resistive bulk current.

5 `vds (V)`Drain-source voltage.

6 `vgs (V)`Gate-source voltage.

7 `vsb (V)`Source-bulk voltage.

8 `ids (A)`Resistive drain current.

9 `idb (A)`Resistive drain-bulk current.

10 `isb (A)`Resistive source-bulk current.

11 `iavl (A)`Substrate current.

12 `pwr (W)`Power.

13 `vto (V)`Threshold voltage at zero back-bias.

14 `vts (V)Vts`.

15 `vgt (V)`Effective gate drive including backbias and drain effects.

16 `vdss (V)`Saturation voltage at actual bias.

17 `vsat (V)`Saturation limit.

18 `gm (S)`Transconductance (d ids / d vgs).

19 `gmb (S)`Bulk transconductance (d ids / d vbs).

20 `gds (S)`Output conductance (d ids / d vds).

21 `cdd (F)`Capacitance (d qd / d vd).

22 `cdg (F)`Capacitance (- d qd / d vg).

23 `cds (F)`Capacitance (- d qd / d vs).

24 `cdb (F)`Capacitance (- d qd / d vb).

25 `cgd (F)`Capacitance (- d qg / d vd).

26 `cgg (F)`Capacitance (d qg / d vg).

27 `cgs (F)`Capacitance (- d qg / d vs).

28 `cgb (F)`Capacitance (- d qg / d vb).

29 `csd (F)`Capacitance (- d qs / d vd).

30 `csg (F)`Capacitance (- d qs / d vg).

31 `css (F)`Capacitance (d qs / d vs).

32 `csb (F)`Capacitance (- d qs / d vb).

33 `cbd (F)`Capacitance (- d qb / d vd).

34 `cbg (F)`Capacitance (- d qb / d vg).

35 `cbs (F)`Capacitance (- d qb / d vs).

36 `cbb (F)`Capacitance (d qb / d vb).

37 uTransistor gain (gm/gds).

38 `rout (Ω)`Small signal output resistance (1/gds).

39 `vearly (V)`Equivalent Early voltage (|id|/gds).

40 `keff (√V )`Describes body effect at actual bias.

41 `beff (S/V)`Effective beta at actual bias in the simple MOS model (2*|ids|/vgt2^2).

42 `fug (Hz)`Unity gain frequency at actual bias (gm/(2*pi*cin)).

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Compact MOS-Transistor Distortion Model (mos1100)

Description

The mos1100 model is based on the thesis of Ronald van Langevelde: "A compact MOSFET Model for Distortion Analysis in Analog Circuit Design", Technische Universiteit Eindhoven, 1998.

(c) Philips Electronics N.V. 2001

In extension to the description a minimum conductance `gmin` is inserted between the drain and source node, to aid convergence. The value of `gmin` is set by an options statement, default = 1e-12 S.

This device is supported within altergroups.

This device is dynamically loaded from the shared object

`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Instance Definition

```
Name  d  g  s  [b] ModelName parameter=value ...
```

Instance Parameters

1 `w=1.0 scale m`Drawn channel width in the layout. Scale set by option scale..

2 `l=1.0 scale m`Drawn channel length in the layout. Scale set by option scale..

3 `mult=1`Number of devices in parallel.

4 `area=1`Alias of `mult`.

5 `region=triode`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `triode`, `sat`, or `subth`.

6 `m=1`Multiplicity factor.

Model Definition

```
model modelName mos1100 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor gender.
Possible values are `n` or `p`.

Geometry Parameters

2 `ler=1.0e-6 m`Effective channel length of the reference transistor.

3 `wer=1e-5 m`Effective channel width of the reference transistor.

4 `lvar=0.0 m`Difference between the actual and the programmed poly-silicon gate length.

5 `lap=4.0e-8 m`Effective channel length reduction per side.

6 `wvar=0.0 m`Difference between the actual and the programmed field-oxide opening.

7 `wot=0.0 m`Effective channel width reduction per side.

Threshold-Voltage Parameters

8 `vfbr=-1.050 V`Flat-band voltage for reference transistor.

9 `stvfb=0.5e-03 V/K`Coefficient of temperature dependence of `vfb`.

10 `kor=0.5` $\sqrt{V}$ Body effect coefficient for the reference transistor.

11 `slko=0.0` $\sqrt{V}$ m
Coefficient of the length dependence of `ko`.

12 `sl2ko=0.0` $\sqrt{V}$ m^2
Second coefficient of the length dependence of `ko`.

13 `swko=0.0` $\sqrt{V}$ m
Coefficient of the width dependence of `ko`.

14 `phibr=0.95 V`Surface potential at strong inversion.

15 `slphib=0.0 Vm`Coefficient of the length dependence of phib.

16 `sl2phib=0.0 Vm`^{2}Second coefficient of the length dependence of phib.

17 `swphib=0.0 Vm`^{2}Coefficient of the width dependence of phib.

Channel-Current Parameters

18 `fbet1=0.0`Relative mobility decrease due to first lateral profile.

19 `lp1=0.8e-6`Characteristic length of first lateral profile.

20 `lp2=0.8e-6`Characteristic length of second lateral profile.

21 `fbet2=0.0`Relative mobility decrease due to second lateral profile.

22 `betsq=370.9e-06 A/V`^{2}
Gain factor for an infinite square transistor.

23 `etabet=1.3`Exponent of the temperature dependence of the gain factor.

24 `thesrr=0.4 1/V`Coefficient of the mobility reduction due to surface roughness scattering.

25 `swthesr=0.0 m`Coefficient of the width dependence of thesr.

26 `thephr=1.29e-2 1/V`
Coefficient of the mobility reduction due to phonon scattering.

27 `swtheph=0.0 m`Coefficient of the width dependence of theph.

28 `etaph=1.75`Exponent of the temperature dependence of theph.

29 `etamobr=1.4`Effective field parameter for dependence on depletion/inversion charge.

30 `stetamob=0.0 1/K`Coefficient of the temperature dependence of etamob.

31 `swetamob=0.0 m`Coefficient of the width dependence of etamob.

32 `nur=1.0`Exponent of the field dependence of the mobility model minus 1.

33 `nuexp=5.25`Exponent of the temperature dependence of parameter nu.

34 `therr=0.155 1/V`Coefficient of the series resistance.

35 `swther=0.0 m`Coefficient of the width dependence of ther.

36 `etar=0.95`Exponent of the temperature dependence of ther.

37 `thesatr=0.5 1/V`Velocity saturation parameter due to optical/acoustic phonon scattering.

38 `etasat=1.04 1/V`Exponent of the temperature dependence of thesat.

39 `swthesat=0.0 m`Coefficient of the width dependence of thesat.

40 `slthesat=1.0`Coefficient of length dependence of thesat.

41 `thesatexp=1.0`Exponent of length dependence of thesat.

Drain-Feedback Parameters

42 `thethr=1e-3 1/V`^{3}Coefficient of self-heating.

43 `thethexp=1.0`Exponent of the length dependence of theth.

44 `ssfr=6.25e-3 1/√V`
Static feedback parameter.

45 `swssf=0.0 m`Coefficient of the width dependence of ssf.

46 `slssf=1.0e-6 m`Coefficient of the length dependence of ssf.

47 `alpr=0.010`Factor of the channel length modulation.

48 `swalp=0.0 m`Coefficient of the width dependence of `alp`.

49 `slalp=1.0 m`Coefficient of the length dependence of alp.

50 `alpexp=1.0`Exponent of the length dependence of alp.

51 `vp=5.0e-2 V`Characteristic voltage of channel-length modulation.

Sub-Threshold Parameters

52 `sdiblo=2e-03 1/√V`
Drain-induced barrier lowering parameter.

53 `sdiblexp=1.35`Exponent of the length dependence of sdibl.

54 `mor=0.0`Parameter for short-channel subtreshold slope.

55 `moexp=1.34`Exponent of the length dependence of mo.

Smoothing Parameter

56 `lmin=1.5e-7 m`Minimum effective channel length in technology, used for calculation of smoothing factor m.

Weak-Avalanche Parameters

57 `a1r=6`Factor of the weak-avalanche current.

58 `sla1=0.0 m`Coefficient of the length dependence of a1.

59 `swa1=0.0 m`Coefficient of the width dependence of a1.

60 `sta1=0.0 1/K`Coefficient of the temperature dependence of a1.

61 `a2r=38.0 V`Exponent of the weak-avalanche current.

62 `sla2=0.0 V m`Coefficient of the length dependence of a2.

63 `swa2=0.0 V m`Coefficient of the width dependence of a2.

64 `a3r=1.0`Factor of the drain-source voltage above which weak-avalanche occurs.

65 `sla3=0.0 m`Coefficient of the length dependence of a3.

66 `swa3=0.0 m`Coefficient of the width dependence of a3.

Gate Current Parameters

67 `iginvr=0.0 A/V`^{2}Gain factor for intrinsic gate tunnelling current in inversion.

68 `igaccr=0.0 A/V`^{2}Gain factor for intrinsic gate tunnelling current in accumulation.

69 `igovr=0.0 A/V`^{2}Gain factor for Source/Drain overlap gate tunnelling current.

Charge parameters

70 `tox=3.2e-09 m`Thickness of the oxide layer.

71 `col=3.2e-10 F/m`Gate overlap capacitance per unit channel width.

Noise Parameters

72 `ntr=1.656e-20 J`Coefficient of the thermal noise.

73 `nfar=1.573e22 1/(Vm`^{4}`)`
First coefficient of the flicker noise.

74 `nfbr=4.752e8 1/(Vm`^{2}`)`
Second coefficient of the flicker noise.

75 `nfcr=0.0 1/V`Third coefficient of the flicker noise.

Temperature Parameters

76 `tr (C)`Reference temperature. Default set by option `tnom`.

77 `tref (C)`Alias of `tr`. Default set by option `tnom`.

78 `tnom (C)`Alias of `tr`. Default set by option `tnom`.

79 `dta=0.0 K`Temperature offset of the device.

80 `trise=0.0 K`Alias of `dta`.

Other Parameters

81 `kpinv=0.0 1/√V`
Inverse of body-effect factor of the poly-silicon gate.

82 `ther1=0.0 V`Numerator of gate voltage dependent part of series resistance.

83 `ther2=1.0 V`Denominator of gate voltage dependent part of series resistance.

84 `vp=5.0e-2 V`Characteristic voltage of channel-length modulation.

85 `binv=48.0 V`Probability factor for intrinsic gate tunnelling current in inversion.

86 `bacc=48.0 V`Probability factor for intrinsic gate tunnelling current in accumulation.

87 `vfbov=0.0 V`Flat-band voltage for the Source/Drain overlap extensions.

88 `kov=2.5 √V` Body-effect factor for the Source/Drain overlap extensions.

89 `gatenoise=0.0`Flag for in/exclusion of induced gate thermal noise.

Output Parameters

1 `vto (V)`Zero-bias threshold voltage.

2 `le (m)`Effective channel length.

3 `we (m)`Effective channel width.

4 `ko (√V )`Body-effect factor.

5 `phib (V)`Surface potential at the onset of strong inversion.

6 `bet (A/V`^{2}`)`Gain factor.

7 `thesr (1/V)`Mobility degradation parameter due to surface roughness scattering.

8 `theph (1/V)`Mobility degradation parameter due to phonon scattering.

9 `etamob`Effective field parameter for dependence on depletion charge.

10 `nu`Exponent of field dependence of mobility model.

11 `ther (1/V)`Coefficient of series resistance.

12 `thesat (1/V)`Velocity saturation parameter due to optical/acoustic phonon scattering.

13 `theth (1/V`^{3}`)`Coefficient of self-heating.

14 `sdibl (1/√V )`Drain-induced barrier lowering parameter.

15 `mo`Parameter for (short-channel) subtreshold slope.

16 `ssf (1/√V )`Static-feedback parameter.

17 `alp`Factor of channel length modulation.

18 mexpSmoothing factor.

19 a1Factor of the weak-avalanche current.

20 a2 (V)Exponent of the weak-avalanche current.

21 a3Factor of the drain-source voltage above which weak-avalanche occurs.

22 iginv (A/V^2)Gain factor for intrinsic gate tunnelling current in inversion.

23 igacc (A/V^2)Gain factor for intrinsic gate tunnelling current in accumulation.

24 igov (A/V^2)Gain factor for Source/Drain overlap tunnelling current.

25 cox (F)Oxide capacitance for the intrinsic channel (* mult).

26 cgdo (F)Oxide capacitance for the gate-drain overlap (* mult).

27 cgso (F)Oxide capacitance for the gate-source overlap (* mult).

28 nt (J)Thermal noise coefficient.

29 nfa (1/(Vm^4))First coefficient of the flicker noise.

30 nfb (1/(Vm^2))Second coefficient of the flicker noise.

31 nfc (1/V)Third coefficient of the flicker noise.

32 tox (m)Thickness of gate oxide layer.

Operating-Point Parameters

1 nfcr=0.0 1/VThird coefficient of the flicker noise.

2 ide (A)Resistive drain current.

3 ige (A)Resistive gate current.

4 ise (A)Resistive source current.

5 ibe (A)Resistive bulk current.

6 isb (A)Resistive source-bulk current.

7 `idb (A)`Resistive drain-bulk current.

8 `pwr (W)`Power.

9 `ids (A)`Drain current, excl. avalanche and tunnel currents.

10 `iavl (A)`Substrate current due to weak-avalanche.

11 `igs (A)`Gate-to-source current due to direct tunnelling.

12 `igd (A)`Gate-to-drain current due to direct tunnelling.

13 `igb (A)`Gate-to-bulk current due to direct tunnelling.

14 `vds (V)`Drain-source voltage.

15 `vgs (V)`Gate-source voltage.

16 `vsb (V)`Source-bulk voltage.

17 `vts (V)`Threshold voltage including back-bias effects.

18 `vth (V)`Threshold voltage including back-bias and drain-bias effects.

19 `vgt (V)`Effective gate drive voltage including back-bias and drain voltage effects.

20 `vdss (V)`Drain saturation voltage at actual bias.

21 `vsat (V)`Saturation limit.

22 `gm (S)`Transconductance (d ids / d vgs).

23 `gmb (S)`Substrate-transconductance (d ids / d vbs).

24 `gds (S)`Output conductance (d ids / d vds).

25 `cdd (F)`Capacitance (d qd / d vd).

26 `cdg (F)`Capacitance (- d qd / d vg).

27 `cds (F)`Capacitance (- d qd / d vs).

28 `cdb (F)`Capacitance (- d qd / d vb).

29 `cgd (F)`Capacitance (- d qg / d vd).

30 `cgg (F)`Capacitance (d qg / d vg).

31 `cgs (F)`Capacitance (- d qg / d vs).

32 `cgb (F)`Capacitance (- d qg / d vb).

33 `csd (F)`Capacitance (- d qs / d vd).

34 `csg (F)`Capacitance (- d qs / d vg).

35 `css (F)`Capacitance (d qs / d vs).

36 `csb (F)`Capacitance (- d qs / d vb).

37 `cbd (F)`Capacitance (- d qb / d vd).

38 `cbg (F)`Capacitance (- d qb / d vg).

39 `cbs (F)`Capacitance (- d qb / d vs).

40 `cbb (F)`Capacitance (d qb / d vb).

41 uTransistor gain (gm/gds).

42 `rout (Ω)`Small-signal output resistance (1/gds).

43 `vearly (V)`Equivalent Early voltage (|id|/gds).

44 `keff (√V )`Body effect parameter.

45 `beff` (A/V^2)Gain factor.

46 `fug (Hz)`Unity gain frequency at actual bias (gm/(2*pi*cin)).

47 `sqrtsfw (V/√Hz )`
Input-referred RMS white noise voltage density.

48 `sqrtsff (V/√Hz )`
Input-referred RMS white noise voltage density at 1 kHz.

49 `fknee (Hz)`Cross-over frequency above which white noise is dominant.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

MOS Level-15 Transistor (mos15)

Description

The MOS15 model is the AMS level 15 model which is the modified Berkeley SPICE level-2 model with the DC model replaced by that of AMS. It is an analytical one-dimensional model that incorporates most of the second-order small-size effects. A smoother version of the level-15 model (with continuous Gds at Vdsat) was also developed. Three charge models are available. MOS15 transistors require the use of a model statement.

This device is not supported within altergroup.

This device is dynamically loaded from the shared object

`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libstmodels.so`

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as` (m^2)Area of source diffusion.

4 `ad` (m^2)Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `ld (m)`Length of drain diffusion region.

10 `ls (m)`Length of source diffusion region.

11 `m=1`Multiplicity factor (number of MOSFETs in parallel).

12 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, or `subth`.

13 `trise`Temperature rise from ambient.

Model Definition

```
model modelName mos15 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

Drain Current Model Parameters

2 `vto=0 V`Threshold voltage at zero body bias.

3 `kp=2.0718e-5 A/V`^{2}
Transconductance parameter.

4 `lambda=0 1/V`Channel length modulation parameter.

5 `phi=0.7 V`Surface potential at strong inversion.

6 `gamma=0` $\sqrt{V}$ Body-effect parameter.

7 `uo=600 cm`^{2}`/V s`Carrier surface mobility.

8 `vmax=`∞ `m/s`Carrier saturation velocity.

9 `ucrit=0 V/cm`Critical field for mobility degradation.

10 `uexp=0`Critical field exponent for mobility degradation.

11 `utra=0 1/V`Transverse field for mobility.

12 `neff=1`Total channel charge coefficient.

13 `delta=0`Width effect on threshold voltage.

Process Parameters

14 `nsub=1.13e16 cm`$^{-3}$
Channel doping concentration.

15 `nss=0 cm`$^{-2}$Surface state density.

16 `nfs=0 cm`$^{-2}$Fast surface state density.

17 `tpg=+1`Type of gate (+1 = opposite of substate, -1 = same as substate, 0 = aluminum).

18 `tox=1e-7 m`Gate oxide thickness.

19 `ld=0 m`Lateral diffusion.

20 `wd=0 m`Field-oxide encroachment.

21 `xw=0 m`Width variation due to masking and etching.

22 `xl=0 m`Length variation due to masking and etching.

23 `xj=0 m`Source/drain junction depth.

Impact Ionization Parameters

24 `ai0=0 1/V`Impact ionization current coefficient.

25 `lai0=0 µm/V`Length sensitivity of `ai0`.

26 `wai0=0 µm/V`Width sensitivity of `ai0`.

27 `bi0=0 V`Impact ionization current exponent.

28 `lbi0=0` μm VLength sensitivity of `bi0`.

29 `wbi0=0` μm VWidth sensitivity of `bi0`.

Overlap Capacitance Parameters

30 `cgso=0` F/mGate-source overlap capacitance.

31 `cgdo=0` F/mGate-drain overlap capacitance.

32 `cgbo=0` F/mGate-bulk overlap capacitance.

33 `meto=0` mMetal overlap in fringing field.

Charge Model Selection Parameters

34 `capmod=bsim`Intrinsic charge model.
Possible values are `none`, `meyer`, `yang`, or `bsim`.

35 `xpart=1`Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

36 `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, 0 for 0/100.

Parasitic Resistance Parameters

37 `rs=0` ΩSource resistance.

38 `rd=0` ΩDrain resistance.

39 `rsh=0` Ω/sqrSource/drain diffusion sheet resistance.

40 `rss=0` Ω mScalable source resistance.

41 `rdd=0` Ω mScalable drain resistance.

42 `rsc=0` ΩSource contact resistance.

43 `rdc=0` ΩDrain contact resistance.

44 `minr=0.1` ΩMinimum source/drain resistance.

45 `ldif=0 m`Lateral diffusion beyond the gate.

46 `hdif=0 m`Length of heavily doped diffusion.

47 `lgcs=0 m`Gate-to-contact length of source side.

48 `lgcd=0 m`Gate-to-contact length of drain side.

49 `sc=∞ m`Spacing between contacts.

Junction Diode Model Parameters

50 `js` (A/m^2)Bulk junction reverse saturation current density.

51 `is=1e-14 A`Bulk junction reverse saturation current.

52 `n=1`Junction emission coefficient.

53 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*iabstol.
Possible values are `no` or `yes`.

54 `imax=1 A`Explosion current.

55 `jmax=1e8` A/m^2Explosion current density.

Junction Capacitance Model Parameters

56 `cbs=0 F`Bulk-source zero-bias junction capacitance.

57 `cbd=0 F`Bulk-drain zero-bias junction capacitance.

58 `cj=0` F/m^2Zero-bias junction bottom capacitance density.

59 `mj=1/2`Bulk junction bottom grading coefficient.

60 `pb=0.8 V`Bulk junction built-in potential.

61 `fc=0.5`Forward-bias depletion capacitance threshold.

62 `cjsw=0 F/m`Zero-bias junction sidewall capacitance density.

63 `mjsw=1/3`Bulk junction sidewall grading coefficient.

64 `pbsw=0.8 V`Side-wall junction built-in potential.

65 `fcsw=0.5`Side-wall forward-bias depletion capacitance threshold.

Operating Region Warning Control Parameters

66 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

67 `bvj=∞ V`Junction reverse breakdown voltage.

Temperature Effects Parameters

68 `tnom (C)`Parameters measurement temperature. Default set by options.

69 `trise=0 C`Temperature rise from ambient.

70 `uto=0 C`Mobility temperature offset.

71 `ute=-1.5`Mobility temperature exponent.

72 `tlev=0`DC temperature selector.

73 `tlevc=0`AC temperature selector.

74 `eg=1.12452 V`Energy band gap.

75 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

76 `gap2=1108 C`Band gap temperature offset.

77 `f1ex=0`Temperature exponent for `ucrit`.

78 `lamex=0 1/C`Temperature parameter for `lambda` and `kappa`.

79 `trs=0 1/C`Temperature parameter for source resistance.

80 `trd=0 1/C`Temperature parameter for drain resistance.

81 `xti=3`Saturation current temperature exponent.

82 `ptc=0 V/C`Surface potential temperature coefficient.

83 `tcv=0 V/C`Threshold voltage temperature coefficient.

84 `pta=0 V/C`Junction potential temperature coefficient.

85 `ptp=0 V/C`Sidewall junction potential temperature coefficient.

86 `cta=0 1/C`Junction capacitance temperature coefficient.

87 `ctp=0 1/C`Sidewall junction capacitance temperature coefficient.

Default Instance Parameters

88 `w=3e-6 m`Default channel width.

89 `l=3e-6 m`Default channel length.

90 `as=0` m^2Default area of source diffusion.

91 `ad=0` m^2Default area of drain diffusion.

92 `ps=0 m`Default perimeter of source diffusion.

93 `pd=0 m`Default perimeter of drain diffusion.

94 `nrd=0 m/m`Default number of squares of drain diffusion.

95 `nrs=0 m/m`Default number of squares of source diffusion.

96 `ldd=0 m`Default length of drain diffusion region.

97 `lds=0 m`Default length of source diffusion region.

Noise Model Parameters

98 `kf=0`Flicker (1/f) noise coefficient.

99 `af=1`Flicker (1/f) noise exponent.

100 `ef=1`Flicker (1/f) noise frequency exponent.

101 `noisemod=1`Noise model selector.

The imax (jmax) parameter is used to aid convergence and prevent numerical overflow. The junction characteristics of the FET are accurately modeled for current (density) up to imax (jmax). For currents (density) above imax (jmax), the junction is modeled as a linear resistor and a warning is printed.

Output Parameters

1 weff (m)Effective channel width.

2 leff (m)Effective channel length.

3 rseff (Ω)Effective source resistance.

4 rdeff (Ω)Effective drain resistance.

Operating-Point Parameters

1 type=nTransistor type.
Possible values are n or p.

2 region=triodeEstimated operating region.
Possible values are off, triode, sat, or subth.

3 reversedReverse mode indicator.
Possible values are no or yes.

4 id (A)Resistive drain current.

5 vgs (V)Gate-source voltage.

6 vds (V)Drain-source voltage.

7 vbs (V)Bulk-source voltage.

8 vth (V)Threshold voltage.

9 vdsat (V)Drain-source saturation voltage.

10 gm (S)Common-source transconductance.

11 gds (S)Common-source output conductance.

12 `gmbs (S)`Body-transconductance.

13 `gameff (√V )`Effective body effect coefficient.

14 `betaeff (A/V`^{2}`)`Effective beta.

15 `cbd (F)`Drain-bulk junction capacitance.

16 `cbs (F)`Source-bulk junction capacitance.

17 `cgs (F)`Gate-source capacitance.

18 `cgd (F)`Gate-drain capacitance.

19 `cgb (F)`Gate-bulk capacitance.

20 `ron (Ω)`On-resistance.

21 `ib (A)`Resistive bulk current.

22 `pwr (W)`Power at op point.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

MOS Level-2 Transistor (mos2)

Description

The MOS2 model is the level-2 model from Berkeley SPICE. The MOS2 model is an analytical, one-dimensional model that incorporates most of the second-order small-size effects. A smoother version of the level-2 model (with continuous `Gds` at `Vdsat`) is also available. Three charge models are available. MOS2 transistors require that you use a model statement.

This device is supported within altergroups.

Sample Instance Statement

```
mn2 (1 2 0 0) nch2 w=10u ad=20p as=20p ps=24u pd=24u
```

Sample Model Statement

```
model nch2 mos2 type=n vto=0.66 lambda=0.018 gamma=0.6 nsub=0.213e16 kp=0.978e-4
tpg=-1 vmax=6e4 ucrit=1e7 utra=0.1 uexp=0.2 is=0
```

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as (m`^{2}`)`Area of source diffusion.

4 `ad (m`^{2}`)`Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `ld (m)`Length of drain diffusion region.

10 `ls (m)`Length of source diffusion region.

11 `m=1`Multiplicity factor (number of MOSFETs in parallel).

12 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

13 `trise`Temperature rise from ambient.

14 `degradation=no`Hot-electron degradation flag.
Possible values are `no` or `yes`.

Model Definition

```
model modelName mos2 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are `n` or `p`.

Drain Current Model Parameters

2 `vto=0 V`Threshold voltage at zero body bias.

3 `kp=2.0718e-5 A/V`^{2}
Transconductance parameter.

4 `lambda=0 1/V`Channel length modulation parameter.

5 `phi=0.7 V`Surface potential at strong inversion.

6 `gamma=0` $\sqrt{V}$ Body-effect parameter.

7 `uo=600 cm`^{2}`/V s`Carrier surface mobility.

8 `vmax=`∞ `m/s`Carrier saturation velocity.

9 `ucrit=0 V/cm`Critical field for mobility degradation.

10 `uexp=0`Critical field exponent for mobility degradation.

11 `utra=0 1/V`Transverse field for mobility.

12 `neff=1`Total channel charge coefficient.

13 `delta=0`Width effect on threshold voltage.

14 `smooth=yes`Drain current smoothing flag.
Possible values are `no` or `yes`.

Process Parameters

15 `nsub=1.13e16 cm`$^{-3}$
Channel doping concentration.

16 nss=0 cm^{-2}Surface state density.

17 nfs=0 cm^{-2}Fast surface state density.

18 tpg=+1Type of gate (+1 = opposite of substate, -1 = same as substate, 0 = aluminum).

19 tox=1e-7 mGate oxide thickness.

20 ld=0 mLateral diffusion.

21 wd=0 mField-oxide encroachment.

22 xw=0 mWidth variation due to masking and etching.

23 xl=0 mLength variation due to masking and etching.

24 xj=0 mSource/drain junction depth.

Impact Ionization Parameters

25 ai0=0 1/VImpact ionization current coefficient.

26 lai0=0 µm/VLength sensitivity of ai0.

27 wai0=0 µm/VWidth sensitivity of ai0.

28 bi0=0 VImpact ionization current exponent.

29 lbi0=0 µm VLength sensitivity of bi0.

30 wbi0=0 µm VWidth sensitivity of bi0.

Overlap Capacitance Parameters

31 cgso=0 F/mGate-source overlap capacitance.

32 cgdo=0 F/mGate-drain overlap capacitance.

33 cgbo=0 F/mGate-bulk overlap capacitance.

34 meto=0 mMetal overlap in fringing field.

Charge Model Selection Parameters

35 `capmod=bsim` Intrinsic charge model.
Possible values are `none`, `meyer`, `yang`, or `bsim`.

36 `xpart=1` Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

37 `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, 0 for 0/100.

Parasitic Resistance Parameters

38 `rs=0` Ω Source resistance.

39 `rd=0` Ω Drain resistance.

40 `rsh=0` Ω/sqr Source/drain diffusion sheet resistance.

41 `rss=0` Ω m Scalable source resistance.

42 `rdd=0` Ω m Scalable drain resistance.

43 `rsc=0` Ω Source contact resistance.

44 `rdc=0` Ω Drain contact resistance.

45 `minr=0.1` Ω Minimum source/drain resistance.

46 `ldif=0` m Lateral diffusion beyond the gate.

47 `hdif=0` m Length of heavily doped diffusion.

48 `lgcs=0` m Gate-to-contact length of source side.

49 `lgcd=0` m Gate-to-contact length of drain side.

50 `sc=∞` m Spacing between contacts.

Junction Diode Model Parameters

51 `js` (A/m^2) Bulk junction reverse saturation current density.

52 is=1e-14 ABulk junction reverse saturation current.

53 n=1Junction emission coefficient.

54 dskip=yesUse simple piece-wise linear model for diode currents below 0.1*iabstol.
Possible values are no or yes.

55 imelt=`imaxA'Explosion current, diode is linearized beyond this current to aid convergence.

56 jmelt=`jmaxA/m'^{2}
Explosion current density, diode is linearized beyond this current to aid convergence.

Junction Capacitance Model Parameters

57 cbs=0 FBulk-source zero-bias junction capacitance.

58 cbd=0 FBulk-drain zero-bias junction capacitance.

59 cj=0 F/m^{2}Zero-bias junction bottom capacitance density.

60 mj=1/2Bulk junction bottom grading coefficient.

61 pb=0.8 VBulk junction built-in potential.

62 fc=0.5Forward-bias depletion capacitance threshold.

63 cjsw=0 F/mZero-bias junction sidewall capacitance density.

64 mjsw=1/3Bulk junction sidewall grading coefficient.

65 pbsw=0.8 VSide-wall junction built-in potential.

66 fcsw=0.5Side-wall forward-bias depletion capacitance threshold.

Operating Region Warning Control Parameters

67 alarm=noneForbidden operating region.
Possible values are none, off, triode, sat, subth, or rev.

68 imax=1 AMaximum current, currents above this limit generate a warning.

69 `jmax=1e8 A/m`^{2}Maximum current density, currents above this limit generate a warning.

70 `bvj=∞ V`Junction reverse breakdown voltage.

71 `vbox=1e9 tox V`Oxide breakdown voltage.

Temperature Effects Parameters

72 `tnom (C)`Parameters measurement temperature. Default set by `options`.

73 `trise=0 C`Temperature rise from ambient.

74 `uto=0 C`Mobility temperature offset.

75 `ute=-1.5`Mobility temperature exponent.

76 `tlev=0`DC temperature selector.

77 `tlevc=0`AC temperature selector.

78 `eg=1.12452 V`Energy band gap.

79 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

80 `gap2=1108 C`Band gap temperature offset.

81 `f1ex=0`Temperature exponent for `ucrit`.

82 `lamex=0 1/C`Temperature parameter for `lambda` and `kappa`.

83 `trs=0 1/C`Temperature parameter for source resistance.

84 `trd=0 1/C`Temperature parameter for drain resistance.

85 `xti=3`Saturation current temperature exponent.

86 `ptc=0 V/C`Surface potential temperature coefficient.

87 `tcv=0 V/C`Threshold voltage temperature coefficient.

88 `pta=0 V/C`Junction potential temperature coefficient.

89 `ptp=0 V/C`Sidewall junction potential temperature coefficient.

90 `cta=0 1/C`Junction capacitance temperature coefficient.

91 `ctp=0 1/C`Sidewall junction capacitance temperature coefficient.

Default Instance Parameters

92 `w=3e-6 m`Default channel width.

93 `l=3e-6 m`Default channel length.

94 `as=0` m^2Default area of source diffusion.

95 `ad=0` m^2Default area of drain diffusion.

96 `ps=0 m`Default perimeter of source diffusion.

97 `pd=0 m`Default perimeter of drain diffusion.

98 `nrd=0 m/m`Default number of squares of drain diffusion.

99 `nrs=0 m/m`Default number of squares of source diffusion.

100 `ldd=0 m`Default length of drain diffusion region.

101 `lds=0 m`Default length of source diffusion region.

Noise Model Parameters

102 `noisemod=1`Noise model selector.

103 `kf=0`Flicker (1/f) noise coefficient.

104 `af=1`Flicker (1/f) noise exponent.

105 `ef=1`Flicker (1/f) noise frequency exponent.

106 `wnoi=1e-5 m`Channel width at which noise parameters were extracted.

Auto Model Selector Parameters

107 `wmax=1.0 m`Maximum channel width for which the model is valid.

108 `wmin=0.0 m`Minimum channel width for which the model is valid.

109 `lmax=1.0 m`Maximum channel length for which the model is valid.

110 `lmin=0.0 m`Minimum channel length for which the model is valid.

Degradation Parameters

111 `degramod=spectre`Degradation model selector.
Possible values are `spectre` or `bert`.

112 `degradation=no`Hot-electron degradation flag.
Possible values are `no` or `yes`.

113 `dvthc=1 V`Degradation coefficient for threshold voltage.

114 `dvthe=1`Degradation exponent for threshold voltage.

115 `duoc=1 S`Degradation coefficient for transconductance.

116 `duoe=1`Degradation exponent for transconductance.

117 `crivth=0.1 V`Maximum allowable threshold voltage shift.

118 `criuo=10%`Maximum allowable normalized mobility change.

119 `crigm=10%`Maximum allowable normalized transconductance change.

120 `criids=10%`Maximum allowable normalized drain current change.

121 `wnom=5e-6 m`Nominal device width in degradation calculation.

122 `lnom=1e-6 m`Nominal device length in degradation calculation.

123 `vbsn=0 V`Substrate voltage in degradation calculation.

124 `vdsni=0.1 V`Drain voltage in `Ids` degradation calculation.

125 `vgsni=5 V`Gate voltage in `Ids` degradation calculation.

126 `vdsng=0.1 V`Drain voltage in `Gm` degradation calculation.

127 `vgsng=5 V`Gate voltage in `Gm` degradation calculation.

Spectre Stress Parameters

128 `esat=1.1e7 V/m`Critical field in Vdsat calculation.

129 `esatg=2.5e6 1/m`Gate voltage dependence of `esat`.

130 `vpg=-0.25`Gate voltage modifier.

131 `vpb=-0.13`Gate voltage modifier.

132 `subc1=2.24e-5`Substrate current coefficient.

133 `subc2=-0.1e-5 1/V`Substrate current coefficient.

134 `sube=6.4`Substrate current exponent.

135 `strc=1`Stress coefficient.

136 `stre=1`Stress exponent.

BERT Stress Parameters

137 `h0=1`Aging coefficient.

138 `hgd=0 1/V`Bias dependence of `h0`.

139 `m0=1`Aging exponent.

140 `mgd=0 1/V`Bias dependence of `m0`.

141 `ecrit0=1.1e5 V/cm`Critical electric field.

142 `lecrit0=0 µm V/cm`Length dependence of `ecrit0`.

143 `wecrit0=0 µm V/cm`Width dependence of `ecrit0`.

144 `ecritg=0 1/cm`Gate voltage dependence of `ecrit0`.

145 `lecritg=0 µm/cm`Length dependence of `ecritg`.

146 `wecritg=0 µm/cm`Width dependence of `ecritg`.

147 `ecritb=0 1/cm`Substrate voltage dependence of `ecrit0`.

148 lecritb=0 µm/cmLength dependence of ecritb.

149 wecritb=0 µm/cmWidth dependence of ecritb.

150 lc0=1Substrate current coefficient.

151 llc0=0 µmLength dependence of lc0.

152 wlc0=0 µmWidth dependence of lc0.

153 lc1=1Substrate current coefficient.

154 llc1=0 µmLength dependence of lc1.

155 wlc1=0 µmWidth dependence of lc1.

156 lc2=1Substrate current coefficient.

157 llc2=0 µmLength dependence of lc2.

158 wlc2=0 µmWidth dependence of lc2.

159 lc3=1Substrate current coefficient.

160 llc3=0 µmLength dependence of lc3.

161 wlc3=0 µmWidth dependence of lc3.

162 lc4=1Substrate current coefficient.

163 llc4=0 µmLength dependence of lc4.

164 wlc4=0 µmWidth dependence of lc4.

165 lc5=1Substrate current coefficient.

166 llc5=0 µmLength dependence of lc5.

167 wlc5=0 µmWidth dependence of lc5.

168 lc6=1Substrate current coefficient.

169 llc6=0 µmLength dependence of lc6.

170 `wlc6=0` μmWidth dependence of `lc6`.

171 `lc7=1`Substrate current coefficient.

172 `llc7=0` μmLength dependence of `lc7`.

173 `wlc7=0` μmWidth dependence of `lc7`.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

Both of these parameters have current density counterparts, `jmax` and `jmelt`, that you can specify if you want the absolute current values to depend on the device area.

Auto Model Selection

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 <= inst_length < lmax and wmin <= inst_width < wmax

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)`Effective channel width.

2 `leff (m)`Effective channel length.

3 `rseff (Ω)`Effective source resistance.

4 `rdeff (Ω)`Effective drain resistance.

5 `aseff` (m^2)Effective area of source diffusion.

6 `adeff` (m^2)Effective area of drain diffusion.

7 `pseff (m)`Effective perimeter of source diffusion.

8 `pdeff (m)`Effective perimeter of source diffusion.

9 `isseff (A)`Effective source-bulk junction reverse saturation current.

10 `isdeff (A)`Effective drain-bulk junction reverse saturation current.

11 `cbseff (F)`Effective zero-bias source-bulk junction capacitance.

12 `cbdeff (F)`Effective zero-bias drain-bulk junction capacitance.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are `n` or `p`.

2 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

3 `degradation=no`Hot-electron degradation flag.
Possible values are `no` or `yes`.

4 `reversed`Reverse mode indicator.
Possible values are `no` or `yes`.

5 `ids (A)`Resistive drain-to-source current.

6 `vgs (V)`Gate-source voltage.

7 `vds (V)`Drain-source voltage.

8 `vbs (V)`Bulk-source voltage.

9 `vth (V)`Threshold voltage.

10 `vdsat (V)`Drain-source saturation voltage.

11 `gm (S)`Common-source transconductance.

12 `gds (S)`Common-source output conductance.

13 `gmbs (S)`Body-transconductance.

14 `gameff` ($\sqrt{V}$)Effective body effect coefficient.

15 `betaeff` (A/V^2)Effective `beta`.

16 `cbd (F)`Drain-bulk junction capacitance.

17 `cbs (F)`Source-bulk junction capacitance.

18 `cgs (F)`Gate-source capacitance.

19 `cgd (F)`Gate-drain capacitance.

20 `cgb (F)`Gate-bulk capacitance.

21 `ron (Ω)`On-resistance.

22 `id (A)`Resistive drain current.

23 `ibulk (A)`Resistive bulk current.

24 `pwr (W)`Power at op point.

25 `gmoverid (1/V)Gm/Ids`.

26 `isub (A)`Substrate current.

27 `stress`Hot-electron stress.

28 `age (s)`Device age.

29 `he_vdsat (V)`Hot Electron `Vdsat`.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

MOS Level-3 Transistor (mos3)

Description

The MOS3 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.

This device is supported within altergroups.

Sample Instance Statement

```
mp3 (0 1 2 2) pchmos3 l=2u w=30u ad=120p as=75p pd=36u ps=6u
```

Sample Model Statement

```
model pchmos3 mos3 type=p vto=-0.83 gamma=0.4511 kappa=5 ld=0.45e-6 kp=0.334e-4
tox=0.3e-7 nsub=0.2e17 capmod=yang vmax=4.5e5 theta=0.25 cbs=10e-15 cbd=10e-15
```

Instance Definition

```
Name  d  g  s  b ModelName parameter=value ...
```

Instance Parameters

1 `w (m)`Channel width.

2 `l (m)`Channel length.

3 `as (m`^{2}`)`Area of source diffusion.

4 `ad (m`^{2}`)`Area of drain diffusion.

5 `ps (m)`Perimeter of source diffusion.

6 `pd (m)`Perimeter of drain diffusion.

7 `nrd (m/m)`Number of squares of drain diffusion.

8 `nrs (m/m)`Number of squares of source diffusion.

9 `ld (m)`Length of drain diffusion region.

10 `ls (m)`Length of source diffusion region.

11 `m=1`Multiplicity factor (number of MOSFETs in parallel).

12 `region=triode`Estimated operating region.
Possible values are `off`, `triode`, `sat`, `subth`, or `breakdown`.

13 `trise`Temperature rise from ambient.

14 `degradation=no`Hot-electron degradation flag.
Possible values are no or `yes`.

Model Definition

```
model modelName mos3 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

Drain Current Model Parameters

2 `vto=0 V`Threshold voltage at zero body bias.

3 `kp=2.0718e-5 A/V`^{2}
Transconductance parameter.

4 `theta=0 1/V`Mobility modulation coefficient.

5 `phi=0.7 V`Surface potential at strong inversion.

6 `gamma=0` $\sqrt{V}$ Body-effect parameter.

7 `uo=600 cm`^{2}`/V s`Carrier surface mobility.

8 `vmax=∞ m/s`Carrier saturation velocity.

9 `eta=0 1/V`Static feedback coefficient.

10 `kappa=0.2`Saturation field factor.

11 `delta=0`Width effect on threshold voltage.

Process Parameters

12 `nsub=1.13e16 cm`$^{-3}$
Channel doping concentration.

13 `nss=0 cm`$^{-2}$Surface state density.

14 `nfs=0 cm`$^{-2}$Fast surface state density.

15 `tpg=+1`Type of gate (+1 = opposite of substate, -1 = same as substate, 0 = aluminum).

16 `tox=1e-7 m`Gate oxide thickness.

17 `ld=0 m`Lateral diffusion.

18 `wd=0 m`Field-oxide encroachment.

19 `xw=0 m`Width variation due to masking and etching.

20 `xl=0 m`Length variation due to masking and etching.

21 `xj=0 m`Source/drain junction depth.

Impact Ionization Parameters

22 `ai0=0 1/V`Impact ionization current coefficient.

23 `lai0=0 µm/V`Length sensitivity of `ai0`.

24 `wai0=0 µm/V`Width sensitivity of `ai0`.

25 `bi0=0 V`Impact ionization current exponent.

26 `lbi0=0 µm V`Length sensitivity of `bi0`.

27 `wbi0=0 µm V`Width sensitivity of `bi0`.

Overlap Capacitance Parameters

28 `cgso=0 F/m`Gate-source overlap capacitance.

29 `cgdo=0 F/m`Gate-drain overlap capacitance.

30 `cgbo=0 F/m`Gate-bulk overlap capacitance.

31 `meto=0 m`Metal overlap in fringing field.

Charge Model Selection Parameters

32 `capmod=bsim`Intrinsic charge model.
Possible values are `none`, `meyer`, `yang`, or `bsim`.

33 `xpart=1`Drain/source channel charge partition in saturation for BSIM charge model, use 0.0 for 40/60, 0.5 for 50/50, or 1.0 for 0/100.

34 `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, 0 for 0/100.

Parasitic Resistance Parameters

35 `rs=0 Ω`Source resistance.

36 `rd=0 Ω`Drain resistance.

37 `rsh=0 Ω/sqr`Source/drain diffusion sheet resistance.

38 `rss=0 Ω m`Scalable source resistance.

39 `rdd=0 Ω m`Scalable drain resistance.

40 `rsc=0 Ω`Source contact resistance.

41 `rdc=0 Ω`Drain contact resistance.

42 `minr=0.1 Ω`Minimum source/drain resistance.

43 `ldif=0 m`Lateral diffusion beyond the gate.

44 `hdif=0 m`Length of heavily doped diffusion.

45 `lgcs=0 m`Gate-to-contact length of source side.

46 `lgcd=0 m`Gate-to-contact length of drain side.

47 `sc=∞ m`Spacing between contacts.

Junction Diode Model Parameters

48 `js (A/m`^{2}`)`Bulk junction reverse saturation current density.

49 `is=1e-14 A`Bulk junction reverse saturation current.

50 `n=1`Junction emission coefficient.

51 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`. Possible values are `no` or `yes`.

52 `imelt=`imaxA'`Explosion current, diode is linearized beyond this current to aid convergence.

53 `jmelt=`jmaxA/m'`^{2}

Explosion current density, diode is linearized beyond this current to aid convergence.

Junction Capacitance Model Parameters

54 `cbs=0 F`Bulk-source zero-bias junction capacitance.

55 `cbd=0 F`Bulk-drain zero-bias junction capacitance.

56 `cj=0 F/m`^{2}Zero-bias junction bottom capacitance density.

57 `mj=1/2`Bulk junction bottom grading coefficient.

58 `pb=0.8 V`Bulk junction built-in potential.

59 `fc=0.5`Forward-bias depletion capacitance threshold.

60 `cjsw=0 F/m`Zero-bias junction sidewall capacitance density.

61 `mjsw=1/3`Bulk junction sidewall grading coefficient.

62 `pbsw=0.8 V`Side-wall junction built-in potential.

63 `fcsw=0.5`Side-wall forward-bias depletion capacitance threshold.

Operating Region Warning Control Parameters

64 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `triode`, `sat`, `subth`, or `rev`.

65 `imax=1 A`Maximum current, currents above this limit generate a warning.

66 `jmax=1e8` A/m^2Maximum current density, currents above this limit generate a warning.

67 `bvj=∞ V`Junction reverse breakdown voltage.

68 `vbox=1e9 tox V`Oxide breakdown voltage.

Temperature Effects Parameters

69 `tnom (C)`Parameters measurement temperature. Default set by `options`.

70 `trise=0 C`Temperature rise from ambient.

71 `uto=0 C`Mobility temperature offset.

72 `ute=-1.5`Mobility temperature exponent.

73 `tlev=0`DC temperature selector.

74 `tlevc=0`AC temperature selector.

75 `eg=1.12452 V`Energy band gap.

76 `gap1=7.02e-4 V/C`Band gap temperature coefficient.

77 `gap2=1108 C`Band gap temperature offset.

78 `f1ex=0`Temperature exponent for `ucrit`.

79 `lamex=0 1/C`Temperature parameter for `lambda` and `kappa`.

80 `trs=0 1/C`Temperature parameter for source resistance.

81 `trd=0 1/C`Temperature parameter for drain resistance.

82 `xti=3`Saturation current temperature exponent.

83 `ptc=0 V/C`Surface potential temperature coefficient.

84 `tcv=0 V/C`Threshold voltage temperature coefficient.

85 `pta=0 V/C`Junction potential temperature coefficient.

86 `ptp=0 V/C`Sidewall junction potential temperature coefficient.

87 `cta=0 1/C`Junction capacitance temperature coefficient.

88 `ctp=0 1/C`Sidewall junction capacitance temperature coefficient.

Default Instance Parameters

89 `w=3e-6 m`Default channel width.

90 `l=3e-6 m`Default channel length.

91 `as=0` m^2Default area of source diffusion.

92 `ad=0` m^2Default area of drain diffusion.

93 `ps=0 m`Default perimeter of source diffusion.

94 `pd=0 m`Default perimeter of drain diffusion.

95 `nrd=0 m/m`Default number of squares of drain diffusion.

96 `nrs=0 m/m`Default number of squares of source diffusion.

97 `ldd=0 m`Default length of drain diffusion region.

98 `lds=0 m`Default length of source diffusion region.

Noise Model Parameters

99 `noisemod=1`Noise model selector.

100 `kf=0`Flicker (1/f) noise coefficient.

101 `af=1`Flicker (1/f) noise exponent.

102 `ef=1`Flicker (1/f) noise frequency exponent.

103 `wnoi=1e-5 m`Channel width at which noise parameters were extracted.

Auto Model Selector Parameters

104 `wmax=1.0 m`Maximum channel width for which the model is valid.

105 `wmin=0.0 m`Minimum channel width for which the model is valid.

106 `lmax=1.0 m`Maximum channel length for which the model is valid.

107 `lmin=0.0 m`Minimum channel length for which the model is valid.

Degradation Parameters

108 `degramod=spectre`Degradation model selector.
Possible values are `spectre` or `bert`.

109 `degradation=no`Hot-electron degradation flag.
Possible values are `no` or `yes`.

110 `dvthc=1 V`Degradation coefficient for threshold voltage.

111 `dvthe=1`Degradation exponent for threshold voltage.

112 `duoc=1 S`Degradation coefficient for transconductance.

113 `duoe=1`Degradation exponent for transconductance.

114 `crivth=0.1 V`Maximum allowable threshold voltage shift.

115 `criuo=10%`Maximum allowable normalized mobility change.

116 `crigm=10%`Maximum allowable normalized transconductance change.

117 `criids=10%`Maximum allowable normalized drain current change.

118 `wnom=5e-6 m`Nominal device width in degradation calculation.

119 `lnom=1e-6 m`Nominal device length in degradation calculation.

120 `vbsn=0 V`Substrate voltage in degradation calculation.

121 `vdsni=0.1 V`Drain voltage in `Ids` degradation calculation.

122 `vgsni=5 V`Gate voltage in `Ids` degradation calculation.

123 `vdsng=0.1 V`Drain voltage in `Gm` degradation calculation.

124 `vgsng=5 V`Gate voltage in `Gm` degradation calculation.

Spectre Stress Parameters

125 `esat=1.1e7 V/m`Critical field in Vdsat calculation.

126 `esatg=2.5e6 1/m`Gate voltage dependence of `esat`.

127 `vpg=-0.25`Gate voltage modifier.

128 `vpb=-0.13`Gate voltage modifier.

129 `subc1=2.24e-5`Substrate current coefficient.

130 `subc2=-0.1e-5 1/V`Substrate current coefficient.

131 `sube=6.4`Substrate current exponent.

132 `strc=1`Stress coefficient.

133 `stre=1`Stress exponent.

BERT Stress Parameters

134 `h0=1`Aging coefficient.

135 `hgd=0 1/V`Bias dependence of `h0`.

136 `m0=1`Aging exponent.

137 mgd=0 1/VBias dependence of m0.

138 ecrit0=1.1e5 V/cmCritical electric field.

139 lecrit0=0 µm V/cmLength dependence of ecrit0.

140 wecrit0=0 µm V/cmWidth dependence of ecrit0.

141 ecritg=0 1/cmGate voltage dependence of ecrit0.

142 lecritg=0 µm/cmLength dependence of ecritg.

143 wecritg=0 µm/cmWidth dependence of ecritg.

144 ecritb=0 1/cmSubstrate voltage dependence of ecrit0.

145 lecritb=0 µm/cmLength dependence of ecritb.

146 wecritb=0 µm/cmWidth dependence of ecritb.

147 lc0=1Substrate current coefficient.

148 llc0=0 µmLength dependence of lc0.

149 wlc0=0 µmWidth dependence of lc0.

150 lc1=1Substrate current coefficient.

151 llc1=0 µmLength dependence of lc1.

152 wlc1=0 µmWidth dependence of lc1.

153 lc2=1Substrate current coefficient.

154 llc2=0 µmLength dependence of lc2.

155 wlc2=0 µmWidth dependence of lc2.

156 lc3=1Substrate current coefficient.

157 llc3=0 µmLength dependence of lc3.

158 wlc3=0 µmWidth dependence of lc3.

159 lc4=1Substrate current coefficient.

160 llc4=0 µmLength dependence of lc4.

161 wlc4=0 µmWidth dependence of lc4.

162 lc5=1Substrate current coefficient.

163 llc5=0 µmLength dependence of lc5.

164 wlc5=0 µmWidth dependence of lc5.

165 lc6=1Substrate current coefficient.

166 llc6=0 µmLength dependence of lc6.

167 wlc6=0 µmWidth dependence of lc6.

168 lc7=1Substrate current coefficient.

169 llc7=0 µmLength dependence of lc7.

170 wlc7=0 µmWidth dependence of lc7.

Imax and Imelt

The imax parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to imax. If imax is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding imax, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, imelt, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds imelt, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of imelt to prevent arithmetic exception, with the exponential term replaced by a linear equation at imelt.

Both of these parameters have current density counterparts, jmax and jmelt, that you can specify if you want the absolute current values to depend on the device area.

Auto Model Selection

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 <= inst_length < lmax and   wmin <= inst_width < wmax

Example

```
model ModelName ModelType {
   1:     <model parameters> lmin=2 lmax=4 wmin=1 wmax=2
   2:     <model parameters> lmin=1 lmax=2 wmin=2 wmax=4
   3:     <model parameters> lmin=2 lmax=4 wmin=4 wmax=6
}
```

Then for a given instance

```
M1 1 2 3 4 ModelName w=3 l=1.5
```

the program would search all the models in the model group with the name `ModelName` and then pick the first model whose geometric range satisfies the selection criteria. In the preceding example, the auto model selector program would choose ModelName.2.

The user must specify both length (l) and width (w) on the device instance line to enable automatic model selection.

Output Parameters

1 `weff (m)`Effective channel width.

2 `leff (m)`Effective channel length.

3 `rseff (Ω)`Effective source resistance.

4 `rdeff (Ω)`Effective drain resistance.

5 aseff (m^2)Effective area of source diffusion.

6 adeff (m^2)Effective area of drain diffusion.

7 pseff (m)Effective perimeter of source diffusion.

8 pdeff (m)Effective perimeter of source diffusion.

9 isseff (A)Effective source-bulk junction reverse saturation current.

10 isdeff (A)Effective drain-bulk junction reverse saturation current.

11 cbseff (F)Effective zero-bias source-bulk junction capacitance.

12 cbdeff (F)Effective zero-bias drain-bulk junction capacitance.

Operating-Point Parameters

1 type=nTransistor type.
Possible values are n or p.

2 region=triodeEstimated operating region.
Possible values are off, triode, sat, subth, or breakdown.

3 degradation=noHot-electron degradation flag.
Possible values are no or yes.

4 reversedReverse mode indicator.
Possible values are no or yes.

5 ids (A)Resistive drain-to-source current.

6 vgs (V)Gate-source voltage.

7 vds (V)Drain-source voltage.

8 vbs (V)Bulk-source voltage.

9 vth (V)Threshold voltage.

10 vdsat (V)Drain-source saturation voltage.

11 gm (S)Common-source transconductance.

12 `gds (S)`Common-source output conductance.

13 `gmbs (S)`Body-transconductance.

14 `gameff` ($\sqrt{V}$)Effective body effect coefficient.

15 `betaeff` (A/V^2)Effective `beta`.

16 `cbd (F)`Drain-bulk junction capacitance.

17 `cbs (F)`Source-bulk junction capacitance.

18 `cgs (F)`Gate-source capacitance.

19 `cgd (F)`Gate-drain capacitance.

20 `cgb (F)`Gate-bulk capacitance.

21 `ron` (Ω)On-resistance.

22 `id (A)`Resistive drain current.

23 `ibulk (A)`Resistive bulk current.

24 `pwr (W)`Power at op point.

25 `gmoverid (1/V)Gm/Ids`.

26 `isub (A)`Substrate current.

27 `stress`Hot-electron stress.

28 `age (s)`Device age.

29 `he_vdsat (V)`Hot Electron `Vdsat`.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the

description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Long Channel JFET/MOSFET Model (mos30)

Description

This long channel JFET/MOSFET model is specially developed to describe the drift region of LDMOS, EPMOS and VDMOS devices. It is described in the Philips MOST Modelbook (Dec.95) as MOS model, level 30 (Used for DMOS). Information on how to obtain this document can be found on Source Link by searching for Philips.

Note: In noise analysis, mos30 instances will not generate any contribution, since there are no noise sources included in the mos30 model.

Warning: Dont use this model. It is obsolete.

Mos30 will be removed from spectre in the next release.

This device is supported within altergroups.

This device is dynamically loaded from the shared object
`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Sample Instance Statement

```
mn30 (1 2 0 0) nchmod area=2 mult=1
```

Sample Model Statement

```
model nchmod mos30 type=n tox=1.1e-5 ron=150 rsat=500 psat=2 vsat=1 vsub=0.59
cgate=1.65e-12 csub=1.1e-9 tref=25
```

Instance Definition

```
Name  d  g  s  [b] ModelName parameter=value ...
```

Instance Parameters

1 `mult=1`Number of devices in parallel.

2 `area=1`Alias of mult.

3 `region=triode`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `triode`, `sat`, or `subth`.

4 `m=1`Multiplicity factor.

Model Definition

```
model modelName mos30 parameter=value ...
```

Model Parameters

1 `type=n`Transistor gender.
Possible values are n or p.

2 `ron=1.0 Ω`Ohmic resistance at zero bias.

3 `rsat=1.0 Ω`Space charge resistance at zero bias.

4 `vsat=10.0 V`Critical drain-source voltage for hot carriers.

5 `psat=1.0`Velocity saturation coefficient.

6 `vp=-1.0 V`Pinch off voltage at zero gate and substrate voltages.

7 `tox=-1.0 cm`Gate oxide thickness.

8 `dch=1.0e15` cm^{-3}Doping level channel.

9 dsub=1.0e15 cm^{-3}Doping level substrate.

10 vsub=0.6 VSubstrate diffusion voltage.

11 vgap=1.2 VBandgap voltage channel.

12 cgate=0.0 FGate capacitance at zero bias.

13 csub=0.0 FSubstrate capacitance at zero bias.

14 tausc=0.0 sSpace charge transit time of the channel.

15 ach=0.0Temperature coefficient resistivity of the channel.

16 kf=0.0Flickernoise coefficient.

17 af=1.0Flickernoise exponent.

18 tr (C)Reference temperature. Default set by option tnom.

19 tref (C)Alias of tr. Default set by option tnom.

20 tnom (C)Alias of tr. Default set by option tnom.

21 dta=0.0 KTemperature offset of the device.

22 trise=0.0 KAlias of dta.

Output Parameters

1 ront (Ω)Ohmic resistance at zero bias.

2 rsat (Ω)Space charge resistance at zero bias.

3 vsatt (V)Critical drain-source voltage for hot carriers.

4 vsubt (V)Substrate diffusion voltage.

5 cgate (F)Gate capacitance at zero bias.

6 csubt (F)Substrate capacitance at zero bias.

Operating-Point Parameters

1 `pwr (W)`Power.

2 `ids (A)`Total current including velocity saturation.

3 `qb (Coul)`Substrate charge.

4 `qg (Coul)`Gate charge.

5 `qds (Coul)`Space charge in the channel.

6 `gdsd (S)`Conductance (d ids / d vd).

7 `gdsg (S)`Conductance (d ids / d vg).

8 `gdss (S)`Conductance (d ids / d vs).

9 `gdsb (S)`Conductance (d ids / d vb).

10 `cbd (F)`Capacitance (d qb / d vd).

11 `cbg (F)`Capacitance (d qb / d vg).

12 `cbs (F)`Capacitance (d qb / d vs).

13 `cbb (F)`Capacitance (d qb / d vb).

14 `cgd (F)`Capacitance (d qg / d vd).

15 `cgg (F)`Capacitance (d qg / d vg).

16 `cgs (F)`Capacitance (d qg / d vs).

17 `cgb (F)`Capacitance (d qg / d vb).

18 `cdsd (F)`Capacitance (d qds / d vd).

19 `cdsg (F)`Capacitance (d qds / d vg).

20 `cdss (F)`Capacitance (d qds / d vs).

21 `cdsb (F)`Capacitance (d qds / d vb).

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

ach M-15	cgd OP-14	m I-4	tox M-7
af M-17	cgg OP-15	mult I-1	tr M-18
area I-2	cgs OP-16	psat M-5	tref M-19
cbb OP-13	csub M-13	pwr OP-1	trise M-22
cbd OP-10	csubt O-6	qb OP-3	type M-1
cbg OP-11	dch M-8	qds OP-5	vgap M-11
cbs OP-12	dsub M-9	qg OP-4	vp M-6
cdsb OP-21	dta M-21	region I-3	vsat M-4
cdsd OP-18	gdsb OP-9	ron M-2	vsatt O-3
cdsg OP-19	gdsd OP-6	ront O-1	vsub M-10
cdss OP-20	gdsg OP-7	rsat O-2	vsubt O-4
cgate O-5	gdss OP-8	rsat M-3	
cgate M-12	ids OP-2	tausc M-14	
cgb OP-17	kf M-16	tnom M-20	

Long Channel JFET/MOSFET Model (mos3002)

Description

This long channel JFET/MOSFET model is specially developed to describe the drift region of LDMOS, EPMOS and VDMOS devices. It is described in the Philips MOST Modelbook (Dec.98) as MOS model, level 3002 (Used for DMOS). Information on how to obtain this document can be found on Source Link by searching for Philips.

Note: In noise analysis, mos3002 instances will not generate any contribution, since there are no noise sources included in the mos3002 model.

This device is supported within altergroups.

This device is dynamically loaded from the shared object

`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Sample Instance Statement

```
mn3 (1 2 0 0) nch3002 area=1 m=2
```

Sample Model Statement

```
model nch3002 mos3002 ron=20 rsat=150 vsat=1 tox=1.23e-5 dch=1.1e16 vsub=0.58
csub=5.43e-13 tausc=1.2e-12 kf=1 tref=27 psat=1 dta=0
```

Instance Definition

```
Name  d  g  s  [b] ModelName parameter=value ...
```

Instance Parameters

1 `mult=1`Number of devices in parallel.

2 `area=1`Alias of mult.

3 `region=triode`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `triode`, `sat`, or `subth`.

4 `m=1`Multiplicity factor.

Model Definition

```
model modelName mos3002 parameter=value ...
```

Model Parameters

1 `type=n`Transistor gender.
Possible values are n or p.

2 `ron=1.0 Ω`Ohmic resistance at zero bias.

3 `rsat=1.0 Ω`Space charge resistance at zero bias.

4 `vsat=10.0 V`Critical drain-source voltage for hot carriers.

5 `psat=1.0`Velocity saturation coefficient.

6 `vp=-1.0 V`Pinch off voltage at zero gate and substrate voltages.

7 `tox=-1.0 cm`Gate oxide thickness.

8 `dch=1.0e15 cm`$^{-3}$Doping level channel.

9 `dsub=1.0e15 cm`$^{-3}$Doping level substrate.

10 `vsub=0.6 V`Substrate diffusion voltage.

11 `vgap=1.2 V`Bandgap voltage channel.

12 `cgate=0.0 F`Gate capacitance at zero bias.

13 `csub=0.0 F`Substrate capacitance at zero bias.

14 `tausc=0.0 s`Space charge transit time of the channel.

15 `ach=0.0`Temperature coefficient resistivity of the channel.

16 `kf=0.0`Flickernoise coefficient.

17 `af=1.0`Flickernoise exponent.

18 `tr (C)`Reference temperature. Default set by option `tnom`.

19 `tref (C)`Alias of tr. Default set by option `tnom`.

20 `tnom (C)`Alias of tr. Default set by option `tnom`.

21 `dta=0.0 K`Temperature offset of the device.

22 `trise=0.0 K`Alias of `dta`.

Output Parameters

1 `ront (Ω)`Ohmic resistance at zero bias.

2 `rsat (Ω)`Space charge resistance at zero bias.

3 `vsatt (V)`Critical drain-source voltage for hot carriers.

4 `vsubt (V)`Substrate diffusion voltage.

5 `cgate (F)`Gate capacitance at zero bias.

6 `csubt (F)`Substrate capacitance at zero bias.

Operating-Point Parameters

1 `pwr (W)`Power.

2 `ids (A)`Total current including velocity saturation.

3 `qb (Coul)`Substrate charge.

4 `qg (Coul)`Gate charge.

5 `qds (Coul)`Space charge in the channel.

6 `gdsd (S)`Conductance (d ids / d vd).

7 `gdsg (S)`Conductance (d ids / d vg).

8 `gdss (S)`Conductance (d ids / d vs).

9 `gdsb (S)`Conductance (d ids / d vb).

10 `cbd (F)`Capacitance (d qb / d vd).

11 `cbg (F)`Capacitance (d qb / d vg).

12 `cbs (F)`Capacitance (d qb / d vs).

13 `cbb (F)`Capacitance (d qb / d vb).

14 `cgd (F)`Capacitance (d qg / d vd).

15 `cgg (F)`Capacitance (d qg / d vg).

16 `cgs (F)`Capacitance (d qg / d vs).

17 `cgb (F)`Capacitance (d qg / d vb).

18 `cdsd (F)`Capacitance (d qds / d vd).

19 `cdsg (F)`Capacitance (d qds / d vg).

20 `cdss (F)`Capacitance (d qds / d vs).

21 `cdsb (F)`Capacitance (d qds / d vb).

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Long Channel JFET/MOSFET Model (mos3100)

Description

This long channel JFET/MOSFET model is special developed to describe the drift region of LDMOS, EPMOS and VDMOS devices. It is described in the Philips MOST Modelbook (Dec.98) as MOS model, level 3002 (Used for DMOS). Information on how to obtain this document can be found on Source Link by searching for Philips.

Note: In noise analysis, mos3100 instances will not generate any contribution, since there are no noise sources included in the mos3100 model.

(c) Philips Electronics N.V. 1993, 1994, 1996, 1998

This device is supported within altergroups.

This device is dynamically loaded from the shared object
`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Instance Definition

```
Name  d  g  s  [b] ModelName parameter=value ...
```

Instance Parameters

1 `mult=1`Number of devices in parallel.

2 `area=1`Alias of mult.

3 `region=triode`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `triode`, `sat`, or `subth`.

4 `m=1`Multiplicity factor.

Model Definition

```
model modelName mos3100 parameter=value ...
```

Model Parameters

1 `type=n`Transistor gender.
Possible values are n or p.

2 `ron=1.0 Ω`Ohmic resistance at zero bias.

3 `rsat=1.0 Ω`Space charge resistance at zero bias.

4 `vsat=10.0 V`Critical drain-source voltage for hot carriers.

5 `psat=1.0`Velocity saturation coefficient.

6 `vp=-1.0 V`Pinch off voltage at zero gate and substrate voltages.

7 `tox=-1.0 m`Gate oxide thickness.

8 `dch=1.0e21` m^{-3}Doping level channel.

9 `dsub=1.0e21` m^{-3}Doping level substrate.

10 `vsub=0.6 V`Substrate diffusion voltage.

11 `vgap=1.2 V`Bandgap voltage channel.

12 `cgate=0.0 F`Gate capacitance at zero bias.

13 `csub=0.0 F`Substrate capacitance at zero bias.

14 `tausc=0.0 s`Space charge transit time of the channel.

15 `ach=0.0`Temperature coefficient resistivity of the channel.

16 `tr (C)`Reference temperature. Default set by option `tnom`.

17 `tref (C)`Alias of tr. Default set by option `tnom`.

18 `tnom (C)`Alias of tr. Default set by option `tnom`.

19 `dta=0.0 K`Temperature offset of the device.

20 `trise=0.0 K`Alias of `dta`.

Output Parameters

1 `ront (Ω)`Ohmic resistance at zero bias.

2 `rsat (Ω)`Space charge resistance at zero bias.

3 `vsatt (V)`Critical drain-source voltage for hot carriers.

4 `vsubt (V)`Substrate diffusion voltage.

5 `cgate (F)`Gate capacitance at zero bias.

6 `csubt (F)`Substrate capacitance at zero bias.

Operating-Point Parameters

1 `pwr (W)`Power.

2 `ids (A)`Total current including velocity saturation.

3 `qb (Coul)`Substrate charge.

4 `qg (Coul)`Gate charge.

5 `qds (Coul)`Space charge in the channel.

6 `gdsd (S)`Conductance (d ids / d vd).

7 `gdsg (S)`Conductance (d ids / d vg).

8 `gdss (S)`Conductance (d ids / d vs).

9 `gdsb (S)`Conductance (d ids / d vb).

10 `cbd (F)`Capacitance (d qb / d vd).

11 `cbg (F)`Capacitance (d qb / d vg).

12 `cbs (F)`Capacitance (d qb / d vs).

13 `cbb (F)`Capacitance (d qb / d vb).

14 `cgd (F)`Capacitance (d qg / d vd).

15 `cgg (F)`Capacitance (d qg / d vg).

16 `cgs (F)`Capacitance (d qg / d vs).

17 `cgb (F)`Capacitance (d qg / d vb).

18 `cdsd (F)`Capacitance (d qds / d vd).

19 `cdsg (F)`Capacitance (d qds / d vg).

20 `cdss (F)`Capacitance (d qds / d vs).

21 `cdsb (F)`Capacitance (d qds / d vb).

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the

description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

ach	M-15	cgd	OP-14	m	I-4	tnom	M-18
area	I-2	cgg	OP-15	mult	I-1	tox	M-7
cbb	OP-13	cgs	OP-16	psat	M-5	tr	M-16
cbd	OP-10	csub	M-13	pwr	OP-1	tref	M-17
cbg	OP-11	csubt	O-6	qb	OP-3	trise	M-20
cbs	OP-12	dch	M-8	qds	OP-5	type	M-1
cdsb	OP-21	dsub	M-9	qg	OP-4	vgap	M-11
cdsd	OP-18	dta	M-19	region	I-3	vp	M-6
cdsg	OP-19	gdsb	OP-9	ron	M-2	vsat	M-4
cdss	OP-20	gdsd	OP-6	ront	O-1	vsatt	O-3
cgate	O-5	gdsg	OP-7	rsat	O-2	vsub	M-10
cgate	M-12	gdss	OP-8	rsat	M-3	vsubt	O-4
cgb	OP-17	ids	OP-2	tausc	M-14		

Silicon On Isolator JFET Model (mos40)

Description

This long channel JFET/MOSFET model is special developed to describe the drift region of LDMOS, EPMOS and VDMOS devices. It is described in the Philips MOST Modelbook (Dec.98) as MOS model, level 3002 (Used for DMOS). Information on how to obtain this document can be found on Source Link by searching for Philips.

Note: In noise analysis, mos40 instances will not generate any contribution, since there are no noise sources included in the mos40 model.

This device is supported within altergroups.

This device is dynamically loaded from the shared object

`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Instance Definition

Name d g s [b] ModelName parameter=value ...

Instance Parameters

1 `mult=1`Number of devices in parallel.

2 `area=1`Alias of mult.

3 `region=triode`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `triode`, `sat`, or `subth`.

4 `m=1`Multiplicity factor.

Model Definition

```
model modelName mos40 parameter=value ...
```

Model Parameters

1 `type=n`Transistor gender.
Possible values are n or p.

2 `ron=1.0 Ω`Ohmic resistance at zero bias.

3 `rsat=1.0 Ω`Space charge resistance at zero bias.

4 `vsat=10.0 V`Critical drain-source voltage for hot carriers.

5 `psat=1.0`Velocity saturation coefficient.

6 `vp=-1.0 V`Pinch off voltage at zero gate and substrate voltages.

7 `tox=-1.0 m`Gate oxide thickness.

8 `dch=1.0e21` m^{-3}Doping level channel.

9 `tbox=-1.0` m^{-3}Box thicknes.

10 `cbox=0.0` m^{-3}Wafer capacitance.

11 `cgate=0.0 F`Gate capacitance at zero bias.

12 `tausc=0.0 s`Space charge transit time of the channel.

13 `ach=0.0`Temperature coefficient resistivity of the channel.

14 `tr (C)`Reference temperature. Default set by option `tnom`.

15 `tref (C)`Alias of tr. Default set by option `tnom`.

16 `tnom (C)`Alias of tr. Default set by option `tnom`.

17 `dta=0.0 K`Temperature offset of the device.

18 `trise=0.0 K`Alias of `dta`.

Output Parameters

1 `ront (Ω)`Ohmic resistance at zero bias.

2 `rsat (Ω)`Space charge resistance at zero bias.

3 `vsatt (V)`Critical drain-source voltage for hot carriers.

4 `vbox (V)`Box voltage.

5 `cgate (F)`Gate capacitance at zero bias.

Operating-Point Parameters

1 `pwr (W)`Power.

2 `ids (A)`Total current including velocity saturation.

3 `qb (Coul)`Substrate charge.

4 `qg (Coul)`Gate charge.

5 `qds (Coul)`Space charge in the channel.

6 `gdsd (S)`Conductance (d ids / d vd).

7 `gdsg (S)`Conductance (d ids / d vg).

8 `gdss (S)`Conductance (d ids / d vs).

9 `gdsb (S)`Conductance (d ids / d vb).

10 `cbd` `(F)` Capacitance (d qb / d vd).

11 `cbg` `(F)` Capacitance (d qb / d vg).

12 `cbs` `(F)` Capacitance (d qb / d vs).

13 `cbb` `(F)` Capacitance (d qb / d vb).

14 `cgd` `(F)` Capacitance (d qg / d vd).

15 `cgg` `(F)` Capacitance (d qg / d vg).

16 `cgs` `(F)` Capacitance (d qg / d vs).

17 `cgb` `(F)` Capacitance (d qg / d vb).

18 `cdsd` `(F)` Capacitance (d qds / d vd).

19 `cdsg` `(F)` Capacitance (d qds / d vg).

20 `cdss` `(F)` Capacitance (d qds / d vs).

21 `cdsb` `(F)` Capacitance (d qds / d vb).

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

cdss	OP-20	gdss	OP-8	rsat	O-2	vsat	M-4
cgate	O-5	ids	OP-2	rsat	M-3	vsatt	O-3

Compact MOS-Transistor Model (mos705)

Description

The mos705 model is a compact MOS-transistor model, intended for the simulation of circuit behavior with emphasis on analog applications. It is described in the Philips MOST Modelbook (Dec.93) as MOS model, level 705.

In extension to the modelbook description a minimum conductance `gmin` is inserted between the drain and source node, to aid convergence. The value of `gmin` is set by an options statement, default = 1e-12 S.

This device is supported within altergroups.

This device is dynamically loaded from the shared object
`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Sample Instance Statement

```
mn1 (1 2 0 0) mna7 ln=120e-6 wn=12e-6
```

Sample Model Statement

```
model mna7 mos705 type=n vtn=0.853 betan=77e-6 tox=15e-9 vfb=-850e-3 tref=25
subthn=3 phi=0.645 lap=100e-9 gkn=-350e-9 th1n=0.15 th2n=0.046 th3n=0.1 fnoise=1e-
10
```

Instance Definition

Name `d g s` [b] ModelName parameter=value ...

Instance Parameters

1 `wn=1.0 scale m` Drawn channel width in the lay-out of the actual transistor. Scale set by option scale.

2 `ln=1.0 scale m`Drawn channel length in the lay-out of the actual transistor. Scale set by option scale.

3 `w=1.0 scale m`Alias for `wn`.

4 `l=1.0 scale m`Alias for `ln`.

5 `mult=1`Number of devices in parallel.

6 `area=1`Alias of `mult`.

7 `region=triode`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `triode`, `sat`, or `subth`.

8 `m=1`Multiplicity factor.

Model Definition

```
model modelName mos705 parameter=value ...
```

Model Parameters

1 `type=n`Transistor gender.
Possible values are n or p.

2 `vtn=0 V`Threshold voltage of the reference transistor at the reference temperature.

3 `kon=0 √V Ko` of the reference transistor.

4 `kn=100m √V K` of the reference transistor.

5 `vsbxn=0 VVsbx` of the reference transistor.

6 `delvx=0 VDvsbx` of the reference transistor.

7 `th1n=0 1/VThe1` of the reference transistor.

8 `th2n=0 1/√V The2` of the reference transistor.

9 `th3n=0 1/VThe3` of the reference transistor at the reference temperature.

10 `gamman=0Gam` of the reference transistor.

11 shiftn=0 V^(1-n)Sh of the reference transistor.

12 nn=0N of the reference transistor.

13 pn=0 1/VP of the reference transistor.

14 ava=0A of the reference transistor.

15 avb=1 VB of the reference transistor.

16 avc=0C of the reference transistor.

17 wref=100u mEffective width of the reference transistor.

18 wtol=0 mDifference between drawn and effective gate width.

19 dvtn=0 V mNarrow-width factor of the threshold voltage at vsbref.

20 dkon=0 √v mNarrow-width factor of ko.

21 dkn=0 √v mNarrow-width factor of k.

22 dvsbxn=0 V mNarrow-width factor of vsbx.

23 ddelvx=0 VmNarrow-width factor of dvsbx.

24 betan=20u A/V^{2}Gain factor of a infinite-square transistor at the reference temperature.

25 dth1n=0 m/VNarrow-width factor of the1.

26 dth2n=0 m/√V Narrow-width factor of the2.

27 dth3n=0 m/VNarrow-width factor of the3.

28 dgamn=0 mNarrow-width factor of gam.

29 dava=0 mNarrow-width factor of a.

30 davb=0 V mNarrow-width factor of b.

31 davc=0 mNarrow-width factor of c.

32 lref=100u mEffective length of the reference transistor.

33 ltol=0 mDifference between drawn and actual gate polysilicon length.

34 gvtn=0 V mShort-channel factor of the threshold voltage at vsbref.

35 gkon=0 √V mShort-channel factor of ko.

36 gkn=0 √V mShort-channel factor of k.

37 gvsbxn=0 V mShort-channel factor of vsbx.

38 gdelvx=0 V mShort-channel factor of dvsbx.

39 gth1n=0 m/VShort-channel factor of the1.

40 gth2n=0 m/√V Short-channel factor of the2.

41 gth3n=0 m/VShort-channel factor of the3.

42 ggamn=0 mShort-channel factor of gam.

43 gshift=0 V^(1-n) m^2
Short-channel factor of sh.

44 gnn=0 mShort-channel factor of n.

45 gpn=0 m/VShort-channel factor of p.

46 gava=0 mShort-channel factor of a.

47 gavb=0 V mShort-channel factor of b.

48 gavc=0 mShort-channel factor of c.

49 lap=0 mHalf of the effective channel-length reduction due to lateral diffusion.

50 vsbref=0 VSource to bulk reference voltage for parameter determination.

51 phi=600m VDiffusion potential at the reference temperature.

52 tcvt=-1m V/KTemperature coefficient of vto.

53 tbetan=1.5Power temperature coefficient of bet.

54 `tth3n=0 1/(V K)`Temperature coefficient of `the3`.

55 `tgth3n=0 m/(V K)`Temperature coefficient of the length dependence of `the3`.

56 `m=1.0`Subthreshold-slope factor at reference back bias and at the reference temperature.

57 `subthn=0`Weak-inversion factor.

58 `vtr=0 V`Depletion-MOS-transistor-transition voltage.

59 `ratio=0`Depletion-MOS-transistor-gain ratio.

60 `vfb=0 V`Flat-band voltage.

61 `tox=100n m`Gate-oxide thickness.

62 `col=0 F/m`Gate/drain or gate/source overlap capacitance per unit length.

63 `fnoise=0` m^2 V^2Flicker-noise factor.

64 `tnoise=0`Thermal-noise factor.

Temperature parameters

65 `tr (C)`Reference temperature. Default set by option `tnom`.

66 `tref (C)`Alias of `tr`. Default set by option `tnom`.

67 `tnom (C)`Alias of `tr`. Default set by option `tnom`.

68 `dta=0 K`Deviation between the temperature of the transistor and the temperature of the circuit.

69 `trise=0 K`Alias of `dta`.

Output Parameters

1 `weff (V)`Effective channel width of the actual transistor.

2 `leff (V)`Effective channel length of the actual transistor.

3 `twophif (V)`Diffusion potential.

4 `bet (A/V`^{2}`)`Gain factor of the transistor.

5 `k (`$\sqrt{V}$`)`Body-effect factor.

6 `ko (`$\sqrt{V}$`)`Initial body-effect factor for dual `k` approach.

7 `vsbx (V)`Transition voltage for dual `k` approach.

8 `dvsbx (V)`Transition-voltage range for dual `K` approach.

9 `vto (V)`Threshold voltage.

10 `von (V)`Onset voltage of the superthreshold region.

11 `the1 (1/V)`Gate-bias-controlled transverse-field mobility reduction factor.

12 `the2 (1/`$\sqrt{V}$`)`Back-bias-controlled transverse-field mobility reduction factor.

13 `the3 (1/V)`Lateral-field mobility reduction factor (velocity saturation).

14 `gam`Static-drain-feedback factor.

15 `sh (V^(1-n))`Threshold-voltage-shift factor.

16 `n`Threshold-voltage-shift exponent.

17 `p (1/V)`Back-bias-shift factor.

18 `me (`$\sqrt{V}$`)`Auxiliary parameter for subthreshold-slope factor.

19 `a`Weak-avalanche multiplier.

20 `b (V)`Weak-avalanche exponent factor.

21 `c`Saturation-voltage reduction factor.

22 `cox (F)`Gate capacitance.

23 `cgso (F)`Gate/source-overlap capacitance.

24 `cgdo (F)`Gate/drain-overlap capacitance.

25 `vtre (V)`Depletion MOS transistor transition voltage.

26 `ratio`Depletion MOS transistor gain ratio.

27 `vfbe (V)`Flat band voltage.

28 `vtemp (V)`kT/q at actual device temperature.

29 `gnoise (V`^{2}`)`Coefficient of the flicker noise for the actual transistor.

30 `unoise (J)`Coefficient of the thermal noise for the actual transistor.

Operating-Point Parameters

1 `ide (A)`Drain current.

2 `ige (A)`Gate current.

3 `ise (A)`Source current.

4 `ibe (A)`Bulk current.

5 `vds (V)`Drain-source voltage.

6 `vgs (V)`Gate-source voltage.

7 `vsb (V)`Source-bulk voltage.

8 `ids (A)`Drain-source current.

9 `idb (A)`Drain-bulk current.

10 `isb (A)`Source-bulk current.

11 `pwr (W)`Power.

12 `vts (V)`Vto including back-bias effects.

13 `vgt (V)`Effective gate drive including back-bias and drain effects.

14 `vdss (V)`Saturation voltage at actual bias.

15 `gm (S)`Transconductance (d ids / d vgs).

16 `gmb (S)`Bulk transconductance (d ids / d vbs).

17 `gds (S)`Output conductance (d ids / d vds).

18 `cdd (F)`Capacitance (d qd / d vd).

19 `cdg (F)`Capacitance (- d qd / d vg).

20 `cds (F)`Capacitance (- d qd / d vs).

21 `cdb (F)`Capacitance (- d qd / d vb).

22 `cgd (F)`Capacitance (- d qg / d vd).

23 `cgg (F)`Capacitance (d qg / d vg).

24 `cgs (F)`Capacitance (- d qg / d vs).

25 `cgb (F)`Capacitance (- d qg / d vb).

26 `csd (F)`Capacitance (- d qs / d vd).

27 `csg (F)`Capacitance (- d qs / d vg).

28 `css (F)`Capacitance (d qs / d vs).

29 `csb (F)`Capacitance (- d qs / d vb).

30 `cbd (F)`Capacitance (- d qb / d vd).

31 `cbg (F)`Capacitance (- d qb / d vg).

32 `cbs (F)`Capacitance (- d qb / d vs).

33 `cbb (F)`Capacitance (d qb / d vb).

34 uTransistor gain (gm/gds).

35 `rout (Ω)`Small signal output resistance (1/gds).

36 `vearly (V)`Equivalent Early voltage (|Id|/gds).

37 `keff (√V )`Describes body effect at actual bias.

38 `beff (S/V)`Effective beta at actual bias in the simple MOS model.

39 `fug (Hz)` Unity gain frequency at actual bias (gm/(2*pi*cox)).

40 `sqrtsfw (V/√Hz )`
Input-referred RMS white noise voltage (sqrt(sth)/gm).

41 `sqrtsff (V/√Hz )`
Input-referred RMS 1/f noise voltage at 1kHz (sqrt(gnoise/1000)).

42 `fknee (Hz)` Cross-over frequency above which white noise is dominant.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

a O-19
area I-6
ava M-14
avb M-15
avc M-16
b O-20
beff OP-38
bet O-4
betan M-24
c O-21
cbb OP-33
cbd OP-30
cbg OP-31
cbs OP-32
cdb OP-21
cdd OP-18
cdg OP-19
cds OP-20
cgb OP-25
cgd OP-22
cgdo O-24
cgg OP-23
cgs OP-24
cgso O-23
dta M-68
dth1n M-25
dth2n M-26
dth3n M-27
dvsbx O-8
dvsbxn M-22
dvtn M-19
fknee OP-42
fnoise M-63
fug OP-39
gam O-14
gamman M-10
gava M-46
gavb M-47
gavc M-48
gdelvx M-38
gds OP-17
ggamn M-42
gkn M-36
gkon M-35
gm OP-15
gmb OP-16
gnn M-44
gnoise O-29
k O-5
keff OP-37
kn M-4
ko O-6
kon M-3
l I-4
lap M-49
leff O-2
ln I-2
lref M-32
ltol M-33
m M-56
m I-8
me O-18
mult I-5
n O-16
nn M-12
p O-17
phi M-51
pn M-13
pwr OP-11
ratio M-59
ratio O-26
region I-7
the3 O-13
tnoise M-64
tnom M-67
tox M-61
tr M-65
tref M-66
trise M-69
tth3n M-54
twophif O-3
type M-1
u OP-34
unoise O-30
vds OP-5
vdss OP-14
vearly OP-36
vfb M-60
vfbe O-27
vgs OP-6
vgt OP-13
von O-10
vsb OP-7
vsbref M-50
vsbx O-7
vsbxn M-5

Compact MOS-Transistor Model (mos902)

Description

The mos902 model is a compact MOS-transistor model, intended for the simulation of circuit behavior with emphasis on analog applications. It is described in the Philips MOST Modelbook (Feb.98) as MOS model, level 902. Information on how to obtain this document can be found on Source Link by searching for Philips.

In extension to the modelbook description a minimum conductance `gmin` is inserted between the drain and source node, to aid convergence. The value of `gmin` is set by an options statement, default = 1e-12 S.

This device is supported within altergroups.

This device is dynamically loaded from the shared object

`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Sample Instance Statement

```
mp1 (0 1 2 2) mos9pch w=10u l=2u area=1.5
```

Sample Model Statement

```
model mos9pch mos902 ler=0.93e-6 wer=20e-6 tref=27 vtor=1.11 kr=0.54 phibr=0.66
vsbxr=100 the1r=0.19 slk=-0.215e-6 swk=98e-9 swthe3=7.8e-9
```

Instance Definition

Name `d g s` [b] ModelName parameter=value ...

Instance Parameters

1 `w=1.0 scale m`Drawn channel width in the lay-out. Scale set by option scale.

2 `l=1.0 scale m`Drawn channel length in the lay-out. Scale set by option scale.

3 `mult=1`Number of devices in parallel.

4 `area=1`Alias of `mult`.

5 `region=triode`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `triode`, `sat`, or `subth`.

6 `m=1`Multiplicity factor.

Model Definition

```
model modelName mos902 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor gender.
Possible values are n or p.

Geometry Parameters

2 `ler=2.5e-6 m`Effective channel length of the reference transistor.

3 `wer=25e-6 m`Effective channel width of the reference transistor.

4 `lvar=0.3e-6 m`Difference between the actual and the programmed poly-silicon gate length.

5 `lap=0.1e-6 m`Effective channel length reduction per side.

6 `wvar=3e-6 m`Difference between the actual and the programmed field-oxide opening.

7 `wot=1e-6 m`Effective channel width reduction per side.

8 `wdog=0 m`Characteristic drawn gate width, below which dogboning appears.

Threshold-Voltage Parameters

9 `vtor=0.8 V`Threshold voltage at zero back-bias.

10 `stvto=0.01 V/K`Coefficient of the temperature dependence of `vto`.

11 `slvto=0.5e-6 V m`Coefficient of the length dependence of `vto`.

12 `sl2vto=0 V m`^{2}Second coefficient of the length dependence of `vto`.

13 `swvto=5e-6 V m`Coefficient of the width dependence of `vto`.

14 `kor=0.5` $\sqrt{V}$ Low-backbias body factor.

15 `slko=1e-6` $\sqrt{V}$ `m`
Coefficient of the length dependence of `ko`.

16 `swko=10e-6` $\sqrt{V}$ `m`
Coefficient of the width dependence of `ko`.

17 `kr=0.1` $\sqrt{V}$ High-backbias body factor.

18 `slk=0.5e-6` $\sqrt{V}$ `m`
Coefficient of the length dependence of `k`.

19 `swk=5e-6` $\sqrt{V}$ `m`
Coefficient of the width dependence of `k`.

20 `phibr=0.65 V`Surface potential at strong inversion.

21 `vsbxr=0.9 V`Transition voltage for the dual-`k`-factor model.

22 `slvsbx=0.5e-6 V m`Coefficient of the length dependence of `vsbx`.

23 `swvsbx=5e-6 V m`Coefficient of the width dependence of `vsbx`.

Channel-Current Parameters

24 `betsq=0.1e-3 A/V`^{2}
Gain factor for an infinite square transistor.

25 `etabet=0.5`Exponent of the temperature dependence of the gain factor.

26 `the1r=0.05 1/V`Coefficient of the mobility reduction due to the gate-induced field.

27 `stthe1r=3e-3 1/(V K)`
Coefficient of the temperature dependence of `the1`.

28 `slthe1r=50e-9 m/V`Coefficient of the length dependence of `the1`.

29 `stlthe1=5e-9 m/(V K)`
Coefficient of the temperature dependence of `slthe1`.

30 `swthe1=1e-6 m/V`Coefficient of the width dependence of `the1`.

31 `fthe1=0`Coefficient describing the width dependence of `the1` for `w` < `wdog`.

32 `the2r=17e-3 1/√V`
Coefficient of the mobility reduction due to the back-bias.

33 `stthe2r=0.1e-3 1/(√V  K)`
Coefficient of the temperature dependence of `the2`.

34 `slthe2r=5e-9 m/√V`
Coefficient of the length dependence of `the2`.

35 `stlthe2=0.5e-9 m/(√V  K)`
Coefficient of the temperature dependence of `slthe2`.

36 `swthe2=0.1e-6 m/√V`
Coefficient of the width dependence of `the2`.

37 `the3r=37e-3 1/V`Coefficient of the mobility reduction due to the lateral field.

38 `stthe3r=0.1e-3 1/(V K)`
Coefficient of the temperature dependence of `the3`.

39 `slthe3r=5e-9 m/V`Coefficient of the length dependence of `the3`.

40 `stlthe3=0.5e-9 m/(V K)`
Coefficient of the temperature dependence of `slthe3`.

41 `swthe3=0.1e-6 m/V`Coefficient of the width dependence of `the3`.

Drain-Feedback Parameters

42 `gam1r=40e-3 V^(1-etads)`
Coefficient for the drain induced threshold shift for large gate drive.

43 `slgam1=0.1e-6 V^(1-etads) m`
Coefficient of the length dependence of `gam1`.

44 `swgam1=1e-6 V^(1-etads) m`
Coefficient of the width dependence of `gam1`.

45 `etadsr=0.6`Exponent of the vds dependence of `gam1`.

46 `alpr=4e-3`Factor of the channel-length modulation.

47 `etaalp=0.5`Exponent of the length dependence of `alp`.

48 `slalp=0.14e-3 m^etaalp`
Coefficient of the length dependence of `alp`.

49 `swalp=0.1e-6 m`Coefficient of the width dependence of `alp`.

50 `vpr=0.25 V`Characteristic voltage of the channel-length modulation.

Sub-threshold Parameters

51 `gamoor=1.1e-3`Coefficient for the drain induced threshold shift at zero gate drive.

52 `slgamoo=10e-15` m^2
Coefficient of the length dependence of `gamoo`.

53 `etagamr=2`Exponent of the back-bias dependence of `gamo`.

54 `mor=0.3`Factor for the subthreshold slope.

55 `stmo=0.01 1/K`Coefficient of the temperature dependence of `mo`.

56 `slmo=1.4e-3 √m`
Coefficient of the length dependence of `mo`.

57 `etamr=2`Exponent of the back-bias dependence of `m`.

58 `zet1r=0.7`Weak-inversion correction factor.

59 `etazet=0.5`Exponent of the length dependence of `zet1`.

60 `slzet1=0.14e-6 m^etazet`
Coefficient of the length dependence of `zet1`.

61 `vsbtr=99 V`Limiting voltage of the vsb dependence of `m` and `gamo`.

62 `slvsbt=10e-6 V m`Coefficient of the length dependence of `vsbt`.

Weak-Avalanche Parameters

63 `a1r=22`Factor of the weak-avalanche current.

64 `sta1=0.1 1/K`Coefficient of the temperature dependence of `a1`.

65 `sla1=10e-6 m`Coefficient of the length dependence of `a1`.

66 `swa1=0.1e-3 m`Coefficient of the width dependence of `a1`.

67 `a2r=33 V`Exponent of the weak-avalanche current.

68 `sla2=10e-6 V m`Coefficient of the length dependence of `a2`.

69 `swa2=0.1e-3 V m`Coefficient of the width dependence of `a2`.

70 `a3r=0.6`Factor of the drain-source voltage above which weak-avalanche occurs.

71 `sla3=1e-6 m`Coefficient of the length dependence of `a3`.

72 `swa3=10e-6 m`Coefficient of the width dependence of `a3`.

Charge Parameters

73 `tox=20e-9 m`Thickness of the oxide layer.

74 `col=50e-12 F/m`Gate overlap capacitance per unit channel width.

Noise Parameters

75 `ntr=21e-21 J`Coefficient of the thermal noise.

76 `nfr=16e-12` V^2Coefficient of the flicker noise.

Temperature Parameters

77 `tr (C)`Reference temperature. Default set by option `tnom`.

78 `tref (C)`Alias of `tr`. Default set by option `tnom`.

79 `tnom (C)`Alias of `tr`. Default set by option `tnom`.

80 `dta=0 K`Temperature offset of the device.

81 `trise=0 K`Alias of `dta`.

Output Parameters

1 `le (m)`Effective channel length.

2 `we (m)`Effective channel width.

3 `vto (V)`Threshold voltage at zero back-bias.

4 `ko` ($\sqrt{V}$)Low-backbias body factor.

5 `k` ($\sqrt{V}$)High-backbias body factor.

6 `phib (V)`Surface potential at strong inversion.

7 `vsbx (V)`Transition voltage for the dual-k-factor model.

8 `bet` (A/V^2)Gain factor (* mult).

9 `the1 (1/V)`Coefficient of the mobility reduction due to the gate-induced field.

10 `the2` ($1/\sqrt{V}$)Coefficient of the mobility reduction due to the back-bias.

11 `the3 (1/V)`Coefficient of the mobility reduction due to the lateral field.

12 `gam1 (V^(1-etads))`Coefficient for the drain induced threshold shift for large gate drive.

13 `etads`Exponent of the vds dependence of `gam1`.

14 `alp`Factor of the channel-length modulation.

15 `vp (V)`Characteristic voltage of the channel-length modulation.

16 `gamoo`Coefficient for the drain induced threshold shift at zero gate drive.

17 `etagam`Exponent of the back-bias dependence of `gamo`.

18 `mo`Factor for the subthreshold slope.

19 `etam`Exponent of the back-bias dependence of `m`.

20 `phit (V)`Thermal voltage.

21 `zet1`Weak-inversion correction factor.

22 `vsbt (V)`Limiting voltage of the vsb dependence of `m` and `gamo`.

23 `a1`Factor of the weak-avalanche current.

24 `a2 (V)`Exponent of the weak-avalanche current.

25 `a3`Factor of the drain-source voltage above which weak-avalanche occurs.

26 `cox (F)`Gate-to-channel capacitance (* mult).

27 `cgdo (F)`Gate-drain overlap capacitance (* mult).

28 `cgso (F)`Gate-source overlap capacitance (* mult).

29 `nt (J)`Coefficient of the thermal noise.

30 `nf` (V^2)Coefficient of the flicker noise (/ mult).

Operating-Point Parameters

1 `ide (A)`Resistive drain current.

2 `ige (A)`Resistive gate current.

3 `ise (A)`Resistive source current.

4 `ibe (A)`Resistive bulk current.

5 `vds (V)`Drain-source voltage.

6 `vgs (V)`Gate-source voltage.

7 `vsb (V)`Source-bulk voltage.

8 `ids (A)`Resistive drain-source current.

9 `idb (A)`Resistive drain-bulk current.

10 `isb (A)`Resistive source-bulk current.

11 `iavl (A)`Substrate current.

12 `pwr (W)`Power.

13 `vt1 (V)Vto` including backbias effects.

14 `vgt2 (V)`Effective gate drive including backbias and drain effects.

15 `vdss1 (V)`Saturation voltage at actual bias.

16 `vsat (V)`Saturation limit.

17 `gm (S)`Transconductance (d ids / d vgs).

18 `gmb (S)`Bulk transconductance (d ids / d vbs).

19 `gds (S)`Output conductance (d ids / d vds).

20 `cdd (F)`Capacitance (d qd / d vd).

21 `cdg (F)`Capacitance (- d qd / d vg).

22 `cds (F)`Capacitance (- d qd / d vs).

23 `cdb (F)`Capacitance (- d qd / d vb).

24 `cgd (F)`Capacitance (- d qg / d vd).

25 `cgg (F)`Capacitance (d qg / d vg).

26 `cgs (F)`Capacitance (- d qg / d vs).

27 `cgb (F)`Capacitance (- d qg / d vb).

28 `csd (F)`Capacitance (- d qs / d vd).

29 `csg (F)`Capacitance (- d qs / d vg).

30 `css (F)`Capacitance (d qs / d vs).

31 `csb (F)`Capacitance (- d qs / d vb).

32 `cbd (F)`Capacitance (- d qb / d vd).

33 `cbg (F)`Capacitance (- d qb / d vg).

34 `cbs (F)`Capacitance (- d qb / d vs).

35 `cbb (F)`Capacitance (d qb / d vb).

36 uTransistor gain (gm/gds).

37 `rout (Ω)`Small signal output resistance (1/gds).

38 `vearly (V)`Equivalent Early voltage (|id|/gds).

39 `keff (√V )`Describes body effect at actual bias.

40 `beff (S/V)`Effective beta at actual bias in the simple MOS model (2*|ids|/vgt2^2).

41 `fug (Hz)`Unity gain frequency at actual bias (gm/(2*pi*cin)).

42 `sqrtsfw (V/√Hz )`
Input-referred RMS white noise voltage (sqrt(sth)/gm).

43 `sqrtsff (V/√Hz )`
Input-referred RMS 1/f noise voltage at 1kHz (sqrt(nf/1000)).

44 `fknee (Hz)`Cross-over frequency above which white noise is dominant.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Compact MOS-Transistor Model (mos903)

Description

The mos903 model is a compact MOS-transistor model, intended for the simulation of circuit behavior with emphasis on analog applications. It is described in the Philips MOST Modelbook (Jun.98) as MOS model, level 903. Information on how to obtain this document can be found on Source Link by searching for Philips.

In extension to the modelbook description a minimum conductance `gmin` is inserted between the drain and source node, to aid convergence. The value of `gmin` is set by an options statement, default = 1e-12 S.

This device is supported within altergroups.

This device is dynamically loaded from the shared object

`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Sample Instance Statement

```
m_1 (1 2 0 0) mos9nch w=0.35e-6 l=0.35e-6
```

Sample Model Statement

```
model mos9nch mos903 ler=3.5e-7 wer=1e-5 lvar=0 lap=2.2e-8 wvar=0 wot=3e-8
vtor=0.76 the1r=0.67 stthe1r=-1.76e-3 etaalp=0 slalp=0 alpr=0.01
```

Instance Definition

```
Name  d  g  s  [b] ModelName parameter=value ...
```

Instance Parameters

1 `w=1.0 scale m`Drawn channel width in the lay-out. Scale set by option scale.

2 `l=1.0 scale m`Drawn channel length in the lay-out. Scale set by option scale.

3 `mult=1`Number of devices in parallel.

4 `area=1`Alias of `mult`.

5 `region=triode`Estimated DC operating region, used as a convergence aid.
Possible values are `off`, `triode`, `sat`, or `subth`.

6 `m=1`Multiplicity factor.

Model Definition

`model modelName mos903 parameter=value ...`

Model Parameters

Device Type Parameters

1 `type=n`Transistor gender.
Possible values are n or p.

Geometry Parameters

2 `ler=2.5e-6 m`Effective channel length of the reference transistor.

3 `wer=25e-6 m`Effective channel width of the reference transistor.

4 `lvar=0.3e-6 m`Difference between the actual and the programmed poly-silicon gate length.

5 `lap=0.1e-6 m`Effective channel length reduction per side.

6 `wvar=3e-6 m`Difference between the actual and the programmed field-oxide opening.

7 `wot=1e-6 m`Effective channel width reduction per side.

Threshold-Voltage Parameters

8 `vtor=0.8 V`Threshold voltage at zero back-bias.

9 `stvto=0.01 V/K`Coefficient of the temperature dependence of `vto`.

10 `slvto=0.5e-6 V m`Coefficient of the length dependence of `vto`.

11 `sl2vto=0 V` m^2Second coefficient of the length dependence of `vto`.

12 `swvto=5e-6 V m`Coefficient of the width dependence of `vto`.

13 `kor=0.5 √V` Low-backbias body factor.

14 `slko=1e-6 √V  m`
Coefficient of the length dependence of `ko`.

15 `swko=10e-6 √V  m`
Coefficient of the width dependence of `ko`.

16 `kr=0.1 √V` High-backbias body factor.

17 `slk=0.5e-6 √V  m`
Coefficient of the length dependence of `k`.

18 `swk=5e-6 √V  m`
Coefficient of the width dependence of `k`.

19 `phibr=0.65 V`Surface potential at strong inversion.

20 `vsbxr=0.9 V`Transition voltage for the dual-`k`-factor model.

21 `slvsbx=0.5e-6 V m`Coefficient of the length dependence of `vsbx`.

22 `swvsbx=5e-6 V m`Coefficient of the width dependence of `vsbx`.

Channel-Current Parameters

23 `betsq=0.1e-3 A/V`^{2}
Gain factor for an infinite square transistor.

24 `etabet=0.5`Exponent of the temperature dependence of the gain factor.

25 `the1r=0.05 1/V`Coefficient of the mobility reduction due to the gate-induced field.

26 `stthe1r=3e-3 1/(V K)`
Coefficient of the temperature dependence of `the1`.

27 `slthe1r=50e-9 m/V`Coefficient of the length dependence of `the1`.

28 `stlthe1=5e-9 m/(V K)`
Coefficient of the temperature dependence of `slthe1`.

29 `swthe1=1e-6 m/V`Coefficient of the width dependence of `the1`.

30 `wdog=0 m`Characteristic drawn gate width, below which dogboning appears.

31 `fthe1=0`Coefficient describing the width dependence of `the1` for `w` < `wdog`.

32 `the2r=17e-3 1/√V`
Coefficient of the mobility reduction due to the back-bias.

33 `stthe2r=0.1e-3 1/(√V K)`
Coefficient of the temperature dependence of `the2`.

34 `slthe2r=5e-9 m/√V`
Coefficient of the length dependence of `the2`.

35 `stlthe2=0.5e-9 m/(√V K)`
Coefficient of the temperature dependence of `slthe2`.

36 `swthe2=0.1e-6 m/√V`
Coefficient of the width dependence of `the2`.

37 `the3r=37e-3 1/V`Coefficient of the mobility reduction due to the lateral field.

38 `stthe3r=0.1e-3 1/(V K)`
Coefficient of the temperature dependence of `the3`.

39 `slthe3r=5e-9 m/V`Coefficient of the length dependence of `the3`.

40 `stlthe3=0.5e-9 m/(V K)`
Coefficient of the temperature dependence of `slthe3`.

41 `swthe3=0.1e-6 m/V`Coefficient of the width dependence of `the3`.

Drain-Feedback Parameters

42 gam1r=40e-3 V^(1-etads)
Coefficient for the drain induced threshold shift for large gate drive.

43 slgam1=0.1e-6 V^(1-etads) m
Coefficient of the length dependence of gam1.

44 swgam1=1e-6 V^(1-etads) m
Coefficient of the width dependence of gam1.

45 etadsr=0.6Exponent of the vds dependence of gam1.

46 alpr=4e-3Factor of the channel-length modulation.

47 etaalp=0.5Exponent of the length dependence of alp.

48 slalp=0.14e-3 m^etaalp
Coefficient of the length dependence of alp.

49 swalp=0.1e-6 mCoefficient of the width dependence of alp.

50 vpr=0.25 VCharacteristic voltage of the channel-length modulation.

Sub-Threshold Parameters

51 gamoor=1.1e-3Coefficient for the drain induced threshold shift at zero gate drive.

52 slgamoo=10e-15 m^2
Coefficient of the length dependence of gamoo.

53 etagamr=2Exponent of the back-bias dependence of gamo.

54 mor=0.3Factor for the subthreshold slope.

55 stmo=0.01 1/KCoefficient of the temperature dependence of mo.

56 slmo=1.4e-3 √m
Coefficient of the length dependence of mo.

57 etamr=2Exponent of the back-bias dependence of m.

58 `zet1r=0.7`Weak-inversion correction factor.

59 `etazet=0.5`Exponent of the length dependence of `zet1`.

60 `slzet1=0.14e-6 m^etazet`
Coefficient of the length dependence of `zet1`.

61 `vsbtr=99 V`Limiting voltage of the vsb dependence of `m` and `gamo`.

62 `slvsbt=10e-6 V m`Coefficient of the length dependence of `vsbt`.

Weak-Avalanche Parameters

63 `a1r=22`Factor of the weak-avalanche current.

64 `sta1=0.1 1/K`Coefficient of the temperature dependence of `a1`.

65 `sla1=10e-6 m`Coefficient of the length dependence of `a1`.

66 `swa1=0.1e-3 m`Coefficient of the width dependence of `a1`.

67 `a2r=33 V`Exponent of the weak-avalanche current.

68 `sla2=10e-6 V m`Coefficient of the length dependence of `a2`.

69 `swa2=0.1e-3 V m`Coefficient of the width dependence of `a2`.

70 `a3r=0.6`Factor of the drain-source voltage above which weak-avalanche occurs.

71 `sla3=1e-6 m`Coefficient of the length dependence of `a3`.

72 `swa3=10e-6 m`Coefficient of the width dependence of `a3`.

Charge Parameters

73 `tox=20e-9 m`Thickness of the oxide layer.

74 `col=50e-12 F/m`Gate overlap capacitance per unit channel width.

Noise Parameters

75 `ntr=21e-21 J`Coefficient of the thermal noise.

76 `nfmod=0.0`Switch that selects either old or new flicker noise model.

77 `nfr=16e-12 V`^{2}Flicker noise coefficient of the reference transistor (for nfmod =1).

78 `nfar=7.15e+22 1/(V m`^{4}`)`
First coefficient of the flicker noise coefficient of the reference transistor (for nfmod=1).

79 `nfbr=2.16e+7 1/(V m`^{2}`)`
Second coefficient of the flicker noise coefficient of the reference transistor (for nfmod=1).

80 `nfcr=0.0 1/V`Third coefficient of the flicker noise coefficient of the reference transistor (for nfmod=1).

Temperature Parameters

81 `tr (C)`Reference temperature. Default set by option `tnom`.

82 `tref (C)`Alias of `tr`. Default set by option `tnom`.

83 `tnom (C)`Alias of `tr`. Default set by option `tnom`.

84 `dta=0 K`Temperature offset of the device.

85 `trise=0 K`Alias of `dta`.

Other Parameters

86 `th3mod=1`Flag for theta3 clipping.

Output Parameters

1 `le (m)`Effective channel length.

2 `we (m)`Effective channel width.

3 `vto (V)`Threshold voltage at zero back-bias.

4 `ko (`$\sqrt{V}$` )`Low-backbias body factor.

5 `k (√V )`High-backbias body factor.

6 `phib (V)`Surface potential at strong inversion.

7 `vsbx (V)`Transition voltage for the dual-`k`-factor model.

8 `bet (A/V`^{2}`)`Gain factor (* mult).

9 `the1 (1/V)`Coefficient of the mobility reduction due to the gate-induced field.

10 `the2 (1/√V )`Coefficient of the mobility reduction due to the back-bias.

11 `the3 (1/V)`Coefficient of the mobility reduction due to the lateral field.

12 `gam1 (V^(1-etads))`Coefficient for the drain induced threshold shift for large gate drive.

13 `etads`Exponent of the vds dependence of `gam1`.

14 `alp`Factor of the channel-length modulation.

15 `vp (V)`Characteristic voltage of the channel-length modulation.

16 `gamoo`Coefficient for the drain induced threshold shift at zero gate drive.

17 `etagam`Exponent of the back-bias dependence of `gamo`.

18 `mo`Factor for the subthreshold slope.

19 `etam`Exponent of the back-bias dependence of `m`.

20 `phit (V)`Thermal voltage.

21 `zet1`Weak-inversion correction factor.

22 `vsbt (V)`Limiting voltage of the vsb dependence of `m` and `gamo`.

23 `a1`Factor of the weak-avalanche current.

24 `a2 (V)`Exponent of the weak-avalanche current.

25 `a3`Factor of the drain-source voltage above which weak-avalanche occurs.

26 `cox (F)`Gate-to-channel capacitance (* mult).

27 `cgdo (F)`Gate-drain overlap capacitance (* mult).

28 `cgso (F)`Gate-source overlap capacitance (* mult).

29 `nt (J)`Coefficient of the thermal noise.

30 `nf` (V^2)Coefficient of the flicker noise (/ mult) (nfmod = 0).

31 `nfa` (1/(V m^4))First coefficient of the flickernoise of the actual transistor (nfmod = 1).

32 `nfb` (1/(V m^2))Second coefficient of the flickernoise of the actual transistor (nfmod = 1).

33 `nfc (1/V)`Second coefficient of the flickernoise of the actual transistor (nfmod = 1).

34 `tox (m)`Thickness of gate oxide layer.

Operating-Point Parameters

1 `ide (A)`Resistive drain current.

2 `ige (A)`Resistive gate current.

3 `ise (A)`Resistive source current.

4 `ibe (A)`Resistive bulk current.

5 `vds (V)`Drain-source voltage.

6 `vgs (V)`Gate-source voltage.

7 `vsb (V)`Source-bulk voltage.

8 `ids (A)`Resistive drain-source current.

9 `idb (A)`Resistive drain-bulk current.

10 `isb (A)`Resistive source-bulk current.

11 `iavl (A)`Substrate current.

12 `pwr (W)`Power.

13 `vt1 (V)Vto` including backbias effects.

14 `vgt2 (V)`Effective gate drive including backbias and drain effects.

15 `vdss1 (V)`Saturation voltage at actual bias.

16 `vsat (V)`Saturation limit.

17 `gm (S)`Transconductance (d ids / d vgs).

18 `gmb (S)`Bulk transconductance (d ids / d vbs).

19 `gds (S)`Output conductance (d ids / d vds).

20 `cdd (F)`Capacitance (d qd / d vd).

21 `cdg (F)`Capacitance (- d qd / d vg).

22 `cds (F)`Capacitance (- d qd / d vs).

23 `cdb (F)`Capacitance (- d qd / d vb).

24 `cgd (F)`Capacitance (- d qg / d vd).

25 `cgg (F)`Capacitance (d qg / d vg).

26 `cgs (F)`Capacitance (- d qg / d vs).

27 `cgb (F)`Capacitance (- d qg / d vb).

28 `csd (F)`Capacitance (- d qs / d vd).

29 `csg (F)`Capacitance (- d qs / d vg).

30 `css (F)`Capacitance (d qs / d vs).

31 `csb (F)`Capacitance (- d qs / d vb).

32 `cbd (F)`Capacitance (- d qb / d vd).

33 `cbg (F)`Capacitance (- d qb / d vg).

34 `cbs (F)`Capacitance (- d qb / d vs).

35 `cbb (F)`Capacitance (d qb / d vb).

36 uTransistor gain (gm/gds).

37 `rout (Ω)`Small signal output resistance (1/gds).

38 `vearly (V)`Equivalent Early voltage (|id|/gds).

39 `keff (√V )`Describes body effect at actual bias.

40 `beff (S/V)`Effective beta at actual bias in the simple MOS model (2*|ids|/vgt2^2).

41 `fug (Hz)`Unity gain frequency at actual bias (gm/(2*pi*cin)).

42 `sqrtsfw (V/√Hz )`
Input-referred RMS white noise voltage (sqrt(sth)/gm).

43 `sqrtsff (V/√Hz )`
Input-referred RMS 1/f noise voltage at 1kHz (sqrt(nf/1000)).

44 `fknee (Hz)`Cross-over frequency above which white noise is dominant.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Microstrip Line (msline)

Description

This is a microstrip line based on the equations of Hammerstad and Jensen. The model contains a thickness correction to the width and frequency dependent permittivity and characteristic impedance. The dispersion equations are those of Kirschning and Jansen.

This device is supported within altergroups.

Sample Instance Statement

```
tl1 (in 0 out 0) msline l=0.15 w=0.01 h=0.01
```

Instance Definition

```
Name  t1  b1  t2  b2 msline parameter=value ...
```

Instance Parameters

1 `l=0 m`Length.

2 `w (m)`Width.

3 `h (m)`Substrate height.

4 `t=0 m`Conductor thickness.

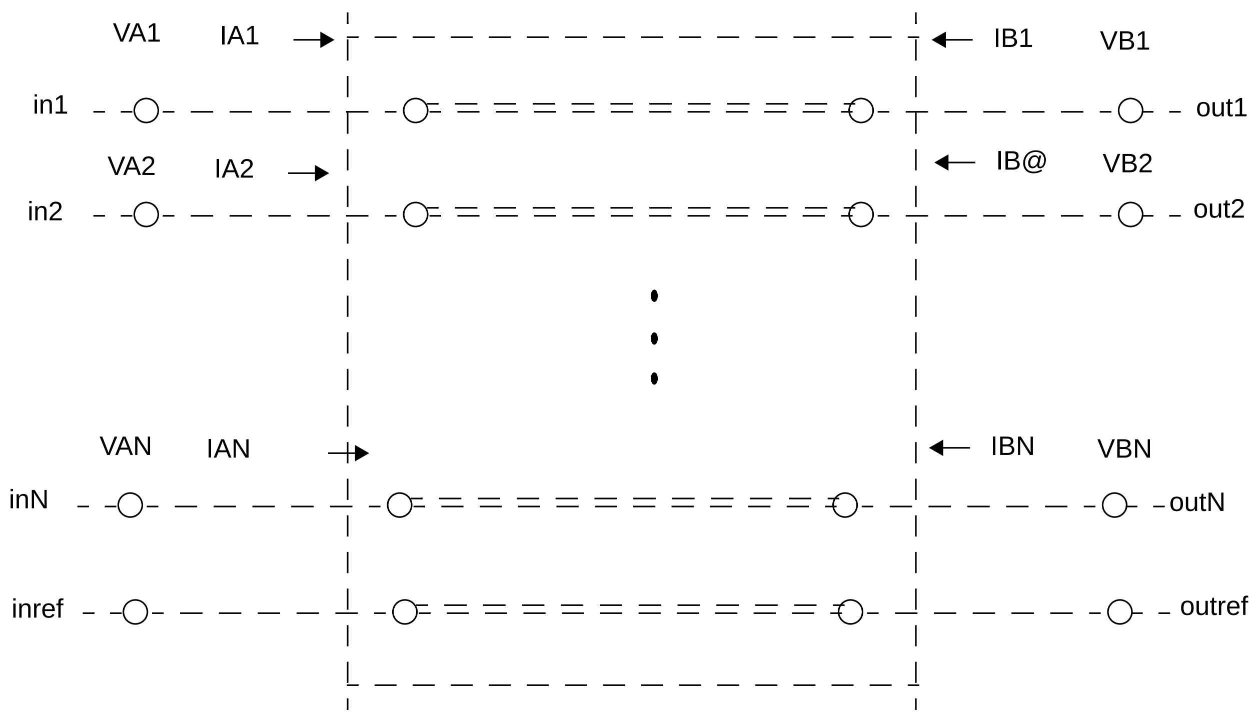
5 `eps=1`Substrate permittivity relative to a vacuum.

6 `m=1`Multiplicity factor.

7 `fmax=10e9 Hz`Maximum signal frequency.

Multi-Conductor Transmission Line (mtline)

Description



A multi-conductor transmission line (mtline) is characterized by constant RLGC matrices or frequency dependent RLGC data. An mtline can have as many conductors as there are as described in the input, however, there must be at least two conductors with one conductor used as reference to define terminal voltages. The reference conductor can be ground. The order of the conductors is the same as the order of the data in the input.

All of the conductors are assumed to have the same length, the input to mtline are conductor length, per-unit-length resistance (R), inductance (L), conductance (G), and capacitance (C) matrices. Because these matrices are generally symmetric, either full matrix description or lower half matrix description can be used. For example, to describe the resistance matrix of a four conductor mtline:

[50 10 1]

```
  R = [ 10 50 10 ] Ohm/meter
      [ 1  10 50 ]
```

The following two model descriptions are equivalent:

```
model line mtline
+ r=[ 50 10 1
+    10 50 10
+    1  10 50 ]
+ ...
model line mtline
+ r=[ 50
+    10 50
+    1  10 50 ]
+ ...
```

Frequency dependent RLGC matrices are described in a data file through model parameter `file`. The frequency axis can be scaled with the `scale` parameter. The frequencies in the data file are then multiplied by `scale` before the simulator uses them. The default scale factor is unity. An example data file is listed below:

```
; Comments: rl.dat
FORMAT FREQ:  R1:1 R2:1 R2:2
              L1:1 L2:1 L2:2
0.001e+9:     4.444 0.000383 4.444
              4.565 0.3545   4.565
0.010e+9:     4.447 0.003834 4.447
              4.565 0.3545   4.565
0.100e+9      4.476 0.03834  4.476
```

	4.565	0.3545	4.565
1.000e+9	4.762	0.3834	4.762
	3.103	0.2357	3.103
10.00e+9	13.96	1.082	13.96
	2.718	0.2058	2.718
100.0e+9	56.88	3.294	56.88
	2.531	0.1866	2.531

Constant matrix is the first choice of input if both the constant matrix and tabular data are provided. If only one frequency point is provided in the `file`, the RLGC matrices are treated as constant matrices.

Rational fitting is used to build a stable model for each mtline instance. The fitting procedure can take a long time for complicated data, the reduced order model (ROM) file option `romdatfile` is used to store and re-use the model in subsequent simulations.

Maximum signal frequency fmax is used to determine the relevant range of rational fitting. The inverse of trise in input signal can be used as an estimation of fmax.

This device is not supported within altergroup.

Sample Instance Statement

```
x1 (a1 b1 a2 b2 0 0) mtline len=0.01
+ r=[ 0.3
+     0.0 0.3 ]
+ c=[ 0.35p
+    -0.03p 0.35p ]
```

Sample Model Statement

```
model mtmodel mtline
+ r=[ 0.3
+     0.0 0.3 ]
+ c=[ 0.35p
+    -0.03p 0.35p ]
model mtmodel mtline
+ r=[ 0.3 0.0
```

```
+     0.0 0.3 ]
+ c=[ 0.35p -0.03p
+    -0.03p  0.35p ]
model mtmodel mtline
+ c=[ 0.35p
+    -0.03p 0.35p ]
+ file="rl.data" scale=1
```

Instance Definition

```
Name  in1  out1  in2  out2 ... ModelName parameter=value ...
Name  in1  out1  in2  out2 ... mtline parameter=value ...
```

The last two terminals will be used as refin and refout respectively.

Instance Parameters

1 `len=0.01 m`Physical length of line.

2 `m=1`Multiplicity factor.

3 `r=[...] Ω/m`Resistance matrix per unit length.

4 `l=[...] H/m`Inductance matrix per unit length.

5 `g=[...] S/m`Conductor matrix per unit length.

6 `c=[...] F/m`Capacitance matrix per unit length.

7 `rskin=[...] Ω/m`Skin effect resistance matrix per unit length.

8 `gdloss=[...] S/m`Dielectric loss conductance matrix per unit length.

9 `file`RLGC data file that contains the frequency dependent RLGC data.

10 `freqscale=1`Frequency scale factor for frequency dependent RLGC data.

11 `romdatfile="rom.dat"`
File that contains the time-domain reduced order model (ROM).

Model Definition

```
model modelName mtline parameter=value ...
```

Model Parameters

RLGC Data Parameters

1 `r=[...] Ω/m`Resistance matrix per unit length.

2 `l=[...] H/m`Inductance matrix per unit length.

3 `g=[...] S/m`Conductor matrix per unit length.

4 `c=[...] F/m`Capacitance matrix per unit length.

5 `rskin=[...] Ω/m`Skin effect resistance matrix per unit length.

6 `gdloss=[...] S/m`Dielectric loss conductance matrix per unit length.

7 `file`RLGC data file that contains the frequency dependent RLGC data.

8 `freqscale=1`Frequency scale factor for frequency dependent RLGC data.

Rational Fitting Parameters

9 `fmax=1e9 Hz`Maximum signal frequency used to determine the relevant range of rational fitting.

10 `resolution=moderate`
Frequency sampling resolution.
Possible values are `conservative`, `moderate` or `liberal`.

Note: Spectre uses rational fitting algorithm to build a stable model that approximates the desired transmission line characteristics. The accuracy of the mtline model is solely dependent on how well the rational approximation is over frequency range [0, `fmax`].

Model parameter `resolution` controls the accuracy of the rational approximation. Tighter resolution control uses higher order rational fitting, which generally results in a more accurate model. However, a higher order model also increases simulation time and tends to be unstable. For most transmission line applications, `moderate` resolution is sufficient to guarantee accuracy and stability.

Modeling Frequency Dependent Effects

One can model the frequency dependent RLGC matrices by providing the data file using model parameter `file`. One should always try to provide accurate and sufficient data to describe the frequency dependent RLGC matrices.

In addition, the following simplified equation can be used to model skin effect with the constant RLGC matrices

R(f) = r + sqrt(f) * (1 + j) * `rskin`,

and the following equation can be used to model dielectric loss with the constant RLGC matrices

G(f) = g + f / sqrt(1 + 4 * (f / fmax)^2) * `gdloss`,

where f stands for frequency.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

`c  M-4`	`freqscale  I-10`	`l  I-4`	`romdatfile  I-11`
`c  I-6`	`g  I-5`	`len  I-1`	`rskin  M-5`
`file  I-9`	`g  M-3`	`m  I-2`	`rskin  I-7`
`file  M-7`	`gdloss  I-8`	`r  I-3`	
`fmax  M-9`	`gdloss  M-6`	`r  M-1`	
`freqscale  M-8`	`l  M-2`	`resolution  M-10`	

Mutual Inductor (mutual_inductor)

Description

The mutual inductor couples two previously specified inductors. There is no limit to the number of inductors that you can couple or to the number of couplings to a particular inductor, but you must specify separate mutual inductor statements for each coupling. Using the `dot` convention, place a `dot` on the first terminal of each inductor.

This device is not supported within altergroup.

The mutual inductor modifies the constitutive equations of two isolated inductors to

v1 = L11*di1/dt + M*di2/dt

v2 = M*di1/dt + L22*di2/dt

where the mutual inductance, M, is computed from the coupling coefficient, k, using k = |M|/sqrt(L11*L22).

Sample Instance Statement with Two Inductors

```
l1 (1 0) inductor
l2 (2 0) inductor
ml1 mutual_inductor coupling=1 ind1=l1 ind2=l2
```

Instance Definition

```
Name mutual_inductor parameter=value ...
```

Instance Parameters

1 `coupling=0`Coupling coefficient.

2 `ind1`Inductor to be coupled.

3 `ind2`Inductor to be coupled.

Node Capacitance (nodcap)

Description

The nodcap model is generally used to model voltage dependent capacitances and currents of the source and drain diodes of MOS transistors and the capacitances of related interconnection areas and sidewalls (IN and PS regions). It is described in the Philips MOST Modelbook (Dec.93) as NODCAP model.

In extension to the modelbook description a minimum conductance `gmin` is inserted between the nodcap nodes, to aid convergence. The value of `gmin` is set by an options statement, default = 1e-12 S.

The `imax` parameter is used to aid convergence and to prevent numerical overflow. The junction characteristics of the node capacitor are accurately modeled for currents up to `imax`. For currents above `imax`, the junction is modeled as a linear resistor and a warning is printed.

This device is supported within altergroups.

This device is dynamically loaded from the shared object

`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Sample Instance Statement

```
n14 (outc vcc) pcapmod ad=191.13e-6 pdg=70e-6 pdcs=75
```

Sample Model Statement

```
model pcapmod nodcap type=p js=10e-6 pt=2 cjr=0.6e-3 vdr=750e-3 tref=27 pmr=0.4
vr=0 kpers=4.9e-6 cox=1.77e-3
```

Instance Definition

```
Name  nn  [ns] ModelName parameter=value ...
```

Instance Parameters

1 `m=1.0` Multiplicity factor.

2 `region=rev` Estimated DC operating region, used as a convergence aid. Possible values are `fwd`, `rev` or `brk`.

3 `cf=0.0 F` Fixed capacitance.

4 `ad=0.0 scale`^{2} `m`^{2}
Diffusion area (source or drain). Scale set by option scale.

5 `pdcs=0.0 scale m` Length of the side-wall of the diffusion area AD which is not under the gate. Scale set by option scale.

6 `pdg=0.0 scale m` Length of the side-wall of the diffusion area AD which is under the gate. Scale set by option scale.

7 `ain=0.0 scale`^{2} `m`^{2}
Area of metal interconnection over thick oxide. Scale set by option scale.

8 `ag=0.0 scale`^{2} `m`^{2}
Area metal or poly-Si over thin oxide. Scale set by option scale.

9 `aps=0.0 scale`^{2} `m`^{2}
Area metal of poly-Si over thick oxide. Scale set by option scale.

10 `pin=0.0 scale m`Length of the side-wall of AIN. Scale set by option scale.

11 `pps=0.0 scale m`Length of the side-wall of APS. Scale set by option scale.

Model Definition

```
model modelName nodcap parameter=value ...
```

Model Parameters

1 `type=n`Type of the nodcap device.
Possible values are n or p.

2 `kpjs=0.0 m`Conversion factor of the sidewall length and the gate edge length to its equivalent area for the saturation current.

3 `n=1.0`Emission coefficient.

4 `bv=`∞ `V`Reverse break-down voltage.

5 `imax=1.0 A`Explosion current.

6 `pt=2.0`Temperature coefficient of the saturation current.

7 `vr=0.0 V`Voltage at which CJR is specified.

8 `fc=0.5`Forward-bias non-ideal junction capacitance coefficient.

9 `pmr=0.5`Grading coefficient.

10 `kpers=0.0 m`Conversion factor of the sidewall length to its equivalent area for the junction capacitance.

11 `kperg=0.0 m`Conversion factor of the gate edge length to its equivalent area for the junction capacitance.

12 `cin=0.0 F/m`^{2}Specific capacitance of the (interconnection) metal to the substrate.

13 `cox=0.0 F/m`^{2}Specific capacitance of poly-silicon or aluminum to the substrate over thin oxide.

14 `cps=0.0 F/m`^{2}Specific capacitance of poly-silicon to the substrate over thick oxide.

15 `cins=0.0 F/m`IN-side-wall capacitance.

16 `cpss=0.0 F/m`PS-side-wall capacitance.

17 `dta=0.0 K`Temperature offset of the device with respect to TEMP.

18 `trise (K)`Alias of `dta`.

19 `tref=27.0 C`Temperature at which the parameters are specified. Default set by option `tnom`.

20 `tnom (C)`Alias of `tref`.

21 `tr (C)`Alias of `tref`.

22 `js=100u A/m`^{2}Saturation current density.

23 `cjr=0.0 F/m`^{2}Specific junction capacitance at Vd = VR.

24 `vdr=0.8 V`Diffusion voltage.

Output Parameters

1 `jst (A/m`^{2}`)`Saturation current density (temperature updated).

2 `cjrt (F/m`^{2}`)`Specific junction capacitance at Vd = VR (temperature updated).

3 `vdrt (V)`Diffusion voltage (temperature updated).

1 `i0 (A)`Saturation current (temperature updated).

Operating-Point Parameters

1 `region=rev` Estimated DC operating region, used as a convergence aid. Possible values are `fwd`, `rev` or `brk`.

2 `vsub (V)` Voltage across the node-capacitance, which is measured from NS (substrate or N-well) to NN (source or drain). Vsub is usually negative.

3 `isub (A)` Resistive leakage current.

4 `cap (F)` Junction capacitance.

5 `q (Coul)` Junction charge.

6 `gm (S)` Total differential conductance.

7 `c0 (F)` Constant part of the junction capacitance.

8 `pwr (W)` Power.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Set Node Quantities (node)

Description

Quantities are used to hold information about particular types of signals, such as their units, absolute tolerances, and maximum allowed change per Newton iteration. Use the `quantity` statement to create new quantities or to redefine properties of an existing quantity. Use this statement to set the quantities for a particular node.

For example, to indicate that the node `net1` is used for thermal signals, the following node statement could be used.

i17 (net1) node value=Temp flow=Pwr

`Temp` and `Pwr` are predefined quantities.

This device is not supported within altergroup.

Sample Instance Statement

```
node1 (1 2 3) node value="T" flow="W" strength=override //Must define T and W with
quantity statement.
```

Instance Definition

```
Name  1  [2] ... node parameter=value ...
```

Instance Parameters

1 `value`Value quantity.

2 `flow`Flow quantity.

3 `strength=override`

Quantity strength.
Possible values are `indifferent`, `suggest`, `insist`, or `override`.

Linear N Port (nport)

Description

An N-port takes its characteristics from an S-parameter data file. An N-port can have as many ports as there are in the N-port described in the S-parameter data file. Each pair of terminals in the `nport` instance statement represents one port. Because there is no limit to the number of ports, there is no limit to the number of terminals. However, the terminals must be given in pairs and there must be at least one pair. The order of the pairs is the same as the order of the ports in the data file. Any missing ports should be skipped.

The S-parameter data file specifies the characteristics of the N-port. You can scale the frequency axis with the `scale` parameter. The frequencies in the data file are then multiplied by scale before the simulator uses them. The default scale factor is unity. S-parameters can be in one of the following formats: real-imag, mag-deg, mag-rad, db-deg, and db-rad.

If interp=spline is specified, the data is interpolated and extrapolated by using cubic splines on the data in polar form. A simple algorithm removes 2 pi jumps in the phase data. Frequency points where the data is measured must therefore be close enough to avoid an excessive number of jumps. Unfortunately, noisy phase data can cause unnecessary warning messages.

If interp=rational is specified, the data is interpolated and extrapolated using a rational function fit to the data. The degree of rational interpolation is automatically selected based on the values of abserr and relerr, unless ratorder is given, in which case relerr and abserr are ignored in selecting the order of the rational function interpolation. It is usually better to allow the simulator to automatically select the rational interpolation order.

If the S-parameter data contains noise, abserr and relerr should be set so that the fitting procedure can ignore the noise, for example, by setting abserr above the noise floor and/or relaxing relerr as necessary.

Because the fitting procedure can take a long time for complicated data, the reduced order model (ROM) file option is available to store and re-use the rational interpolation function in subsequent simulations.

It is not practical to rely on extrapolated data.

This device is not supported within altergroup.

Sample Instance Statement

```
x1 (a1 0 b1 0 b3 0) ndata file="sparam2.data"
```

Sample Model Statement

```
model ndata nport file="sparam.data" scale=1
```

Instance Definition

```
Name  t1  b1  [t2]  [b2] ... ModelName parameter=value ...
Name  t1  b1  [t2]  [b2] ... nport parameter=value ...
```

Terminals must be given in pairs.

Instance Parameters

1 `m=1`Multiplicity factor.

2 `file`S-parameter data file name.

3 `scale=1`Frequency scale factor.

4 `interp=spline`Method to interpolate s-parameter data.
Possible values are `spline` or `rational`.

Spline Interpolation Parameters

5 `usewindow=yes`Use smooth data windowing function. The use of window improves time-domain amplitude resolution, set this parameter to no for better high-frequency resolution.
Possible values are `no` or `yes`.

Rational Interpolation Parameters

6 `relerr=0.01`Maximum relative allowed tolerance for rational interpolation errors. Deviations of the nport model from supplied s-parameter data of relative magnitude less than relerr are generally ignored.

7 `abserr=1e-4`Maximum absolute allowed tolerance for rational interpolation errors. Deviations of the nport model from supplied s-parameter data of absolute magnitude less than abserr are generally ignored.

8 `romdatfile`File used for storing time-domain reduced order model (ROM).

9 `ratorder`Order of rational function to use in fitting the s-parameter data. If this argument is given, relerr and abserr are ignored in selecting the order of

the rational function interpolation. If ratorder is not specified then the program will attempt to select an order of rational interpolation that satisfies the criteria implied by abserr and relerr.

Noise Parameters

10 `trise (C)`Temperature rise from ambient.

11 `thermalnoise=yes`Thermal noise.
Possible values are `no` or `yes`.

Model Definition

```
model modelName nport parameter=value ...
```

Model Parameters

1 `file`S-parameter data file name.

2 `scale=1`Frequency scale factor.

3 `usewindow=yes`Use smooth data windowing function. The use of window improves time-domain amplitude resolution, set this parameter to no for better high-frequency resolution.
Possible values are `no` or `yes`.

4 `trise=0 C`Default temperature rise from ambient.

5 `thermalnoise=yes`Thermal noise.
Possible values are `no` or `yes`.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the

description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

abserr I-7	m I-1	scale I-3	trise I-10
file I-2	ratorder I-9	scale M-2	trise M-4
file M-1	relerr I-6	thermalnoise I-11	usewindow I-5
interp I-4	romdatfile I-8	thermalnoise M-5	usewindow M-3

Parameter Value Tester (paramtest)

Description

The parameter value tester tests the values of its parameters and prints a message if they satisfy the testers criteria. The tester therefore allows you to check the ranges of subcircuit parameters. If you specify more than one test, the message is printed if any test passes. The message is also printed if no tests are specified.

This device is not supported within altergroup.

Sample Instance Statement

```
tooShort paramtest errorif=(l < 0.2um) message="W of device is less than 0.2um"
```

Instance Definition

```
Name paramtest parameter=value ...
```

Instance Parameters

1 `printif`Message is printed if this value is nonzero.

2 `warnif`Message is printed as a warning if this value is nonzero.

3 `errorif`Message is printed as an error and program quits if this value is nonzero.

4 `message`Text of message.

5 `severity=status`Message severity (use if printing message without test).
Possible values are `debug`, `status`, `warning`, `error`, or `fatal`.

Polynomial Current Controlled Current Source (pcccs)

Description

A vector of coefficients specifies the polynomial function that defines the relationship between the output current and the controlling currents. You must specify at least one coefficient.

This device is not supported within altergroup.

For a polynomial in M variables a1, a2, ... am, the polynomial function F(a0,a1,...,am) is given by

F = c0 + c1 * a1 + c2 * a2 + ...

+ c(m+1) * a1^2 + c(m+2) * a1 * a2 +...

+ c(2m+1) * a2^2 + c(2m+2) * a2 * a3 +...

where the cs are coefficients of the polynomial terms.

Sample Instance Statement

```
vpc (net1 0) pcccs probes=[vb vc ve vlp vpn] coeffs=[0 8.8e6 -8.8e6 9e6 8e6 -9e6]
```

Instance Definition

```
Name  sink  src pcccs parameter=value ...
```

Instance Parameters

1 `coeffs=[...]`Polynomial coefficients. At least one must be given.

2 `probes=[...]`Devices through which the controlling currents flow.

3 `ports=[...]`Indice of the probe ports through which the controlling currents flow.

4 `gain=1`Gain Parameter.

5 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)` Output voltage.

3 `pwr (W)` Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

coeffs	I-1	i	OP-1	ports	I-3	pwr	OP-3
gain	I-4	m	I-5	probes	I-2	v	OP-2

Polynomial Current Controlled Voltage Source (pccvs)

Description

The polynomial function defining the relationship between the output voltage and the controlling currents is specified by a vector of coefficients. At least one coefficient must always be specified. Current through the voltage source is calculated and is defined to be positive if it flows from the positive terminal, through the source, to the negative terminal.

This device is not supported within altergroup.

For a polynomial in M variables a1, a2, ... am, the polynomial function F(a0,a1,...,am) is given by

F = c0 + c1 * a1 + c2 * a2 + ...

+ c(m+1) * a1^2 + c(m+2) * a1 * a2 +...

+ c(2m+1) * a2^2 + c(2m+2) * a2 * a3 +...

where the cs are coefficients of the polynomial terms.

Sample Instance Statement

```
ixy (net1 0) pccvs coeffs=[0 1 0 1] probes=[vin1 vin2] gain=2
```

Instance Definition

```
Name  p  n pccvs parameter=value ...
```

Instance Parameters

1 `coeffs=[...]`Polynomial coefficients. At least one must be given.

2 `probes=[...]`Devices through which the controlling currents flow.

3 `ports=[...]`Indice of the probe ports through which the controlling currents flow.

4 `gain=1`Gain Parameter.

5 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

coeffs I-1	i OP-1	ports I-3	pwr OP-3
gain I-4	m I-5	probes I-2	v OP-2

Physical Resistor (phy_res)

Description

A physical resistor consists of a linear resistor (tied between `t1` and `t2`) and two diodes (tied between `t1`-`t0` and `t2`-`t0`). The diodes are junction diodes. Under normal operation, the two diodes are reverse biased, but the parameter `subtype` can reverse the direction of the diodes. If you do not specify `t0`, ground is assumed. The instance parameters always override model parameters. If you do not specify the instance resistance value, it is calculated from the model parameters.

This device is supported within altergroups.

If R(inst) is not given and R(model) is given,

R(inst) = R(model).

Otherwise,

R(inst) = Rsh * (L - 2 * etchl) / (W - 2 * etch).

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

R(V) = dV / dI

= R(inst) / (1 + c1 * V + c2 * V^2 + ...)

where

V = V(t1) - V(t2)

Here V is the controlling voltage across the resistor. It is also the controlling voltage when the model parameter `polyarg` is set to `diff`. In this form, the physical resistor is symmetric with respect to V(t1) and V(t2). The branch current as a function of the applied voltage is given by

I(V) = (V / R(inst)) * (1 + 1/2 * c1 * V + 1/3 * c2 * V^2 + ...)

where ck is the kth entry in the coefficient vector.

If the model parameter `polyarg` is set to `sum`, then the controlling voltage is defined as

Vsum = ((V(t1) - V(t0)) + (V(t2) - V(t0)))/2

Here, Vsum is the controlling voltage between the resistor and the substrate, `t0`. In this case, the device becomes asymmetric with respect to V(t1) and V(t2). The branch current as a function of the applied voltage for this case is given by

I(V) = (V / R(inst)) * (1 + c1 * Vsum + c2 * Vsum^2 + ...)

The large-signal conductance is given by

G(V) = I/V = (1 + c1 * Vsum + c2 * Vsum^2 + ...) / R(inst)

Since the device is asymmetrical, the small-signal conductance is not very meaningful.

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(inst) + temp

if trise(inst) is given, and

T = trise(model) + 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 `t1`-`t0` and `t2`-`t0`. Otherwise, nonlinear junction capacitors are used and the zero-bias capacitance values are calculated from the model parameters.

If C(inst) is not given and C(model) is given,

C(inst) = C(model).

Otherwise,

C(inst) = 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].

Sample Instance Statement

```
res1 (net9 vcc) resphy l=1e-3 w=2e-6
```

Sample Model Statement

```
model resphy phy_res rsh=85 tc1=1.53e-3 tc2=4.67e-7 etch=0 cj=1.33e-3 cjsw=3.15e-
10 tc1c=9.26e-4
```

Instance Definition

```
Name  1  2  [0] ModelName parameter=value ...
```

Instance Parameters

1 `r (Ω)`Resistance.

2 `c (F)`Linear capacitance.

3 `l (m)`Line length.

4 `w (m)`Line width.

5 `region=normal`Estimated operating region.
Possible values are `normal` or `breakdown`.

6 `tc1=0 1/C`Linear temperature coefficient of resistor.

7 `tc2=0` C^{-2}Quadratic temperature coefficient of resistor.

8 `tc1c=0 1/C`Linear temperature coefficient of linear capacitor.

9 `tc2c=0` C^{-2}Quadratic temperature coefficient of linear capacitor.

10 `trise (C)`Temperature rise from ambient.

11 `m=1`Multiplicity factor.

The `w` and `l` parameters are scaled by the option parameters `scale` and `scalem`. The values of `w` and `l` printed by Spectre are those given in the input file. These values may not have the correct units if the scaling factors are not unity. The actual effective resistor dimensions are stored in the output parameters. You can obtain these dimensions with the info statement. You can delete the diodes from the device by either setting `is=0` or `subtype=poly`. You can also set both `mj` and `mjsw` to zero to make the capacitance linear but still calculated from the device geometry. If `subtype=poly`, the linear capacitors will always be used irrespective of the values of `mj` and `mjsw`.

Model Definition

```
model modelName phy_res parameter=value ...
```

Model Parameters

Substrate Type Parameters

1 `subtype=p`Substrate type.
Possible values are `n`, `p` or `poly`.

Resistance Parameters

2 `r=∞ Ω`Default resistance.

3 `rsh=∞ Ω/sqr`
Sheet resistance.

4 `minr=0.1 Ω`Minimum resistance.

5 `coeffs=[...]`Vector of polynomial conductance coefficients.

6 `polyarg=diff`Polynomial model argument type.
Possible values are `sum` or `diff`.

Temperature Effects Parameters

7 `tc1=0 1/C`Linear temperature coefficient of resistor.

8 `tc2=0` C^{-2}Quadratic temperature coefficient of resistor.

9 `tc1c=0` C^{-2}Linear temperature coefficient of linear capacitor.

10 `tc2c=0` C^{-2}Quadratic temperature coefficient of linear capacitor.

11 `tnom (C)`Parameters measurement temperature. Default set by `options`.

12 `trise=0 C`Temperature rise from ambient.

Junction Diode Model Parameters

13 is (A)Saturation current.

14 js=0 A/m^{2}Saturation current density.

15 n=1Emission coefficient.

16 eg=1.11 VBand gap.

17 xti=3Saturation current temperature exponent.

18 imelt=`imaxA'Explosion current, diode is linearized beyond this current to aid convergence.

19 jmelt=`jmeltA/m'^{2}
Explosion current density, diode is linearized beyond this current to aid convergence.

20 imax=1 AMaximum current, currents above this limit generate a warning.

21 jmax=1e8 A/m^{2}Maximum current density, currents above this limit generate a warning.

22 dskip=yesUse simple piece-wise linear model for diode currents below 0.1*iabstol. Possible values are no or yes.

23 bvj=∞ VJunction reverse breakdown voltage.

Junction Capacitance Model Parameters

24 c=0 FDefault linear capacitance.

25 cj=0 F/m^{2}Zero-bias junction bottom capacitance density.

26 cjsw=0 F/mZero-bias junction sidewall capacitance density.

27 mj=1/2Junction bottom grading coefficient.

28 mjsw=1/3Junction sidewall grading coefficient.

29 pb=0.8 VJunction bottom built-in potential.

30 pbsw=0.8 VJunction sidewall built-in potential.

31 `fc=0.5`Junction bottom capacitor forward-bias threshold.

32 `fcsw=0.5`Junction sidewall capacitor forward-bias threshold.

33 `tt=0 s`Transit time.

Device Size Parameters

34 `l=∞ m`Default line length.

35 `w=1e-6 m`Default line width.

36 `etch=0 m`Narrowing due to etching.

37 `etchl=0 m`Length reduction due to etching.

38 `etchc=etch m`Narrowing due to etching for capacitances.

39 `etchlc=etchl m`Length reduction due to etching for capacitances.

40 `scaler=1`Resistance scaling factor.

41 `scalec=1`Capacitance scaling factor.

Noise Model Parameters

42 `kf=0`Flicker (1/f) noise coefficient.

43 `af=1`Flicker (1/f) noise exponent.

44 `wdexp=1`Flicker (1/f) noise W exponent.

45 `ldexp=1`Flicker (1/f) noise L exponent.

46 `weexp=0`Flicker (1/f) noise W effective exponent.

47 `leexp=0`Flicker (1/f) noise L effective exponent.

48 `fexp=1`Flicker (1/f) noise frequency exponent.

Output Parameters

1 `leff (m)`Effective line length.

2 `weff (m)`Effective line width.

3 `iseff (A)`Effective saturation current.

4 `reff (Ω)`Effective resistance.

5 `ceff (F)`Effective zero-bias capacitance.

Operating-Point Parameters

1 `subtype=p`Substrate type.
Possible values are `n`, `p` or `poly`.

2 `region=normal`Estimated operating region.
Possible values are `normal` or `breakdown`.

3 `i (A)`Current through the resistor.

4 `capd1 (F)`Capacitance at the positive node.

5 `capd2 (F)`Capacitance at the negative node.

6 `id1 (A)`Current between nodes `t1` and `t0`.

7 `id2 (A)`Current between nodes `t2` and `t0`.

8 `res (Ω)`Resistance between nodes `t1` and `t2`.

9 `resd1 (Ω)`Resistance between nodes `t1` and `t0`.

10 `resd2 (Ω)`Resistance between nodes `t2` and `t0`.

11 `pwr (W)`Power at op point.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

af M-43	i OP-3	mjsw M-28	tc1 I-6
bvj M-23	id1 OP-6	n M-15	tc1 M-7
c I-2	id2 OP-7	pb M-29	tc1c I-8
c M-24	imax M-20	pbsw M-30	tc1c M-9
capd1 OP-4	imelt M-18	polyarg M-6	tc2 I-7
capd2 OP-5	is M-13	pwr OP-11	tc2 M-8
ceff O-5	iseff O-3	r I-1	tc2c I-9
cj M-25	jmax M-21	r M-2	tc2c M-10
cjsw M-26	jmelt M-19	reff O-4	tnom M-11
coeffs M-5	js M-14	region I-5	trise I-10
dskip M-22	kf M-42	region OP-2	trise M-12
eg M-16	l M-34	res OP-8	tt M-33
etch M-36	l I-3	resd1 OP-9	w I-4
etchc M-38	ldexp M-45	resd2 OP-10	w M-35
etchl M-37	leexp M-47	rsh M-3	wdexp M-44
etchlc M-39	leff O-1	scalec M-41	weexp M-46
fc M-31	m I-11	scaler M-40	weff O-2
fcsw M-32	minr M-4	subtype M-1	xti M-17
fexp M-48	mj M-27	subtype OP-1	

Independent Resistive Source (port)

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, AC and PAC signals of the port. However, the amplitude of the sine wave in the transient and PAC analyses can alternatively be specified as the power in dBm delivered by the port when terminated with the reference resistance.

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.

This device is not supported within altergroup.

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

V(T) = V * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2]

Sample Instance Statement

```
p20 (2 0) port num=2 r=50 type=pulse period=1e-9 rise=1e-10 fall=1e-10 val1=1
width=0.5n mag=1
```

Instance Definition

```
Name  p  n port parameter=value ...
```

Instance Parameters

1 `dc=0` VDC value.

General Waveform parameters

2 `type=dc`Waveform type.
Possible values are `dc`, `pulse`, `pwl`, `sine`, or `exp`.

3 `fundname`Name of the fundamental frequency. Must be specified if the source is active during a pdisto analysis or it is the active clock during an envlp analysis.

4 `delay=0 s`Waveform delay time.

Pulse Waveform Parameters

5 `val0=0 V`Zero value used in pulse and exponential waveforms.

6 `val1=1 V`One value used in pulse and exponential waveforms.

7 `period=∞ s`Period of waveform.

8 `rise (s)`Rise time for pulse waveform (time for transition from `val0` to `val1`).

9 `fall (s)`Fall time for pulse waveform (time for transition from `val1` to `val0`).

10 `width=∞ s`Pulse width (duration of `val1`).

PWL Waveform Parameters

11 `file`Name of file containing waveform.

12 `wave=[...]`Vector of time/value pairs that defines waveform.

13 `offset=0 V`DC offset for the PWL waveform.

14 `scale=1`Scale factor for the PWL waveform.

15 `stretch=1`Scale factor for time given for the PWL waveform.

16 `allbrkpts`All the points in the PWL waveform are breakpoints if set to yes. Default is yes if the number of points is less than 20.
Possible values are `no` or `yes`.

17 `pwlperiod (s)`Period of the periodic PWL waveform.

18 `twidth=pwlperiod/1000 s`
Transition width used when making PWL waveforms periodic.

Sinusoidal Waveform Parameters

19 `sinedc=dc V`DC level for sinusoidal waveforms.

20 `ampl=1 V`Peak amplitude of sinusoidal waveform.

21 `dbm (dBm)`Amplitude of sinusoidal waveform in dBm (alternative to `ampl`).

22 `freq=0 Hz`Frequency of sinusoidal waveform.

23 `sinephase=0 °`Phase of sinusoid when `t=delay`.

24 `ampl2=1 V`Peak amplitude of second sinusoidal waveform.

25 `dbm2 (dBm)`Amplitude of second sinusoidal waveform in dBm (alternative to `ampl2`).

26 `freq2=0 Hz`Frequency of second sinusoidal waveform.

27 `sinephase2=0 °`Phase of second sinusoid when `t=delay`.

28 `fundname2`Name of the fundamental frequency associated with freq2. Must be specified if freq2 is used in a pdisto analysis.

29 `fmmodindex=0`FM index of modulation for sinusoidal waveform.

30 `fmmodfreq=0 Hz`FM modulation frequency for sinusoidal waveform.

31 `ammodindex=0`AM index of modulation for sinusoidal waveform.

32 `ammodfreq=0 Hz`AM modulation frequency for sinusoidal waveform.

33 `ammodphase=0 °`AM phase of modulation for sinusoidal waveform.

34 `damp=0 1/s`Damping factor for sinusoidal waveform.

Exponential Waveform Parameters

35 `td1=0 s`Rise start time for exponential wave.

36 `tau1 (s)`Rise time constant for exponential wave.

37 `td2 (s)`Fall start time for exponential wave.

38 `tau2 (s)`Fall time constant for exponential wave.

Noise Parameters

39 `noisefile`Name of file containing excess spot noise data in the form of frequency-noise pairs.

40 `noisevec=[...]` V^2`/Hz`Excess spot noise as a function of frequency in the form of frequency-noise pairs.

41 `noisetemp (C)`Noise temperature of port. If not specified, the noise temperature is taken to be the actual temperature of the port.

Port Parameters

42 `r=50 Ω`Reference resistance.

43 `num`Port number.

44 `m=1`Multiplicity factor.

Small signal parameters

45 `mag=0 V`Small signal voltage.

46 `phase=0 °`Small signal phase.

47 `xfmag=1 V/V`Transfer function analysis magnitude.

48 `pacmag=0 V`Periodic AC analysis magnitude.

49 `pacdbm (dBm)`Periodic AC analysis magnitude in dBm (alternative to `pacmag`).

50 `pacphase=0 °`Periodic AC analysis phase.

Temperature Effects Parameters

51 `tc1=0 1/C`First order temperature coefficient.

52 `tc2=0 C`$^{-2}$Second order temperature coefficient.

53 `tnom (C)`Parameters measurement temperature. Default set by options.

If you do not specify the DC value, it is assumed to be the `time=0` value of the waveform.

In DC analyses, the only active parameters are `dc`, `m`, and the temperature coefficient parameters. In AC analyses, the only active parameters are `m`, `mag` and `phase`. In transient analyses, all parameters are active except the small signal parameters and the noise parameters. The `type` parameter selects which type of waveform is generated. You may specify parameters for more than one waveform type, and use the alter statement to change the waveform type between analyses.

A vector of time-value pairs describes the piecewise linear waveform. As an alternative, you can read the waveform from a file. In this case, you give time-value pairs one pair per line with a space or tab between the time and the value.

If you set `allbrkpts` to `yes`, you force the simulator to place time points at each point specified in a PWL waveform during a transient analysis. This can be very expensive for waveforms with many points. If you set `allbrkpts` to no, Spectre inspects the waveform, looking for abrupt changes, and forces time points only at those changes.

The PWL waveform is periodic if you specify `pwlperiod`. If the value of the waveform specified is not exactly the same at both its beginning its end, then you must provide a nonzero value `twidth`. Before repeating, the waveform changes linearly in an interval of `twidth` from its value at (`period` - `twidth`) to its value at the beginning of the waveform. Thus `twidth` must always be less than `period`.

You can give the excess noise of the source as a file or specify it with a vector of frequency-noise pairs. For a file, give the frequency-noise pairs one pair per line with a space or tab between the frequency and noise values.

When computing the noise figure of a circuit driven at its input by a port, the noise temperature (noisetemp) of the port should be set to 16.85C (290K) in order to match the standard IEEE definition of noise figure. In addition, all other sources of noise in the port (noisefile and noisevec) should be disabled. If a noiseless port is desired, set the noise temperature to absolute zero or below, and do not specify a noise file or noise vector.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Polynomial Voltage Controlled Current Source (pvccs)

Description

A vector of coefficients specifies the polynomial function that defines the relationship between the output current and the controlling voltages. You must specify at least one coefficient. Current exits the source node and enters the sink node.

This device is not supported within altergroup.

For a polynomial in M variables a1, a2, ... am, the polynomial function F(a0,a1,...,am) is given by

F = c0 + c1 * a1 + c2 * a2 + ...

+ c(m+1) * a1^2 + c(m+2) * a1 * a2 +...

+ c(2m+1) * a2^2 + c(2m+2) * a2 * a3 +...

where the cs are coefficients of the polynomial terms.

Sample Instance Statement

```
v2 (net1 0 net2 0) pvccs coeffs=[0 -2e-3 - 10e-3] gain=2 m=1
```

Instance Definition

```
Name  sink  src  ps1  ns1 ... pvccs parameter=value ...
```

Instance Parameters

1 `coeffs=[...]`Polynomial coefficients. At least one must be given.

2 `gain=1`Gain Parameter.

3 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Polynomial Voltage Controlled Voltage Source (pvcvs)

Description

A vector of coefficients specifies the polynomial function that defines the relationship between the output voltage and the controlling voltages. You must specify at least one coefficient. Current through the voltage source is calculated and is defined to be positive if it flows from the positive terminal, through the source, to the negative terminal.

This device is not supported within altergroup.

For a polynomial in M variables a1, a2, ... am, the polynomial function F(a0,a1,...,am) is given by

F = c0 + c1 * a1 + c2 * a2 + ...

+ c(m+1) * a1^2 + c(m+2) * a1 * a2 +...

+ c(2m+1) * a2^2 + c(2m+2) * a2 * a3 +...

where the `cs` are coefficients of the polynomial terms.

Sample Instance Statement

```
v1 (p 0 c1 0) pvcvs coeffs=[0 0 0 0.1 1 1] gain=1
```

Instance Definition

```
Name  p  n  ps1  ns1 ... pvcvs parameter=value ...
```

Instance Parameters

1 `coeffs=[...]`Polynomial coefficients. At least one must be given.

2 `gain=1`Gain Parameter.

3 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Quantity Information (quantity)

Description

Quantities are used to hold information about particular types of signals, such as their units, absolute tolerances, and maximum allowed change per Newton iteration. Two predefined quantities are voltage and current. A node indicates the type of its signals by keeping pointers to quantities, thus an electrical node points to the voltage quantity for its value, and to the current for its residue. Since many electrical nodes point to the same quantities, changing an attribute on the quantity, such as the absolute tolerance, changes it for many nodes. Use this statement to create new quantities or to redefine properties of an existing quantity. Use the `node` statement to set the quantities for a particular node.

The predefined quantities are as follows:

I: Electrical current in Amperes.

Units = A

Absolute tolerance = 1 pA

Huge value = 4..036 MA

Blowup value = 1 GA

MMF: Magnetomotive force in Ampere-Turns.

Units = A*turn

Absolute tolerance = 1 pA*turn

Huge value = 4..036 MA*turn

Blowup value = 1 GA*turn

Pwr: Power in Watts.

Units = W

Absolute tolerance = 1 nW

Huge value = 4..036 MW

Blowup value = 1 GW

Temp: Temperatures in Celsius.

Units = C

Absolute tolerance = 100 uC

Huge value = 4..036 MC

Blowup value = 1 GC

U: Unitless signals scaled to unity.

Absolute tolerance = 1e-06

Huge value = 4..036e+06

Blowup value = 1e+09

V: Electomotive force in Volts.

Units = V

Absolute tolerance = 1 uV

Maximum change = 300 mV if limit=delta

Huge value = 1 kV

Blowup value = 1 GV

Wb: Magnetic flux in Webers.

Units = Wb

Absolute tolerance = 1 nWb

Huge value = 4..036 MWb

Blowup value = 1 GWb

This device is not supported within altergroup.

Sample Instance Statement

```
voltageQ quantity name="V" abstol=3u maxdelta=500m huge=10K blowup=1G
```

Instance Definition

```
Name quantity parameter=value ...
```

Instance Parameters

1 `name`Name.

2 `description`Description of quantity.

3 `units`Units.

Newton Parameters

4 `abstol`Absolute tolerance.

5 `maxdelta=∞`
Maximum change allowed on a Newton iteration when `limit=delta`.

6 `huge=4..036e+06`Maximum change allowed on a Newton iteration otherwise.

7 `blowup=1e+09`If a signal exceeds this value, the simulation will terminate with an error. It is assumed that the circuit is unstable and is blowing up.

Diffusion Resistor Model (rdiff)

Description

The rdiff model is a diffusion resistor model, which accurately models the temperature, applied bias and back-bias dependencies of NWell, N+, and P+ resistors. It is described in the paper MODEL FOR DIFFUSION RESISTORS (NWell, N+, P+) USED IN CMOS IC DESIGNS by M.J.B.Bolt, FASELEC Process Development Group, PDG-93029, Modified 3rd May 1995.

Some extensions to that description are applied:

Appropriate model and instance parameter default values are used.

No clipping of parameters is performed. Parameter values are checked for validity. If invalid parameter values occur, the job is aborted with an error message.

For exact inverse behavior of the model in case of Vh less than Vl, the setting of Vbl=abs(Vb-Vl) is replaced by Vbl=min(abs(Vb-Vh),abs(Vb-Vl)). Additionally, the direction of Ih is inverted in this case.

Note: In noise analysis, rdiff instances will not generate any contribution, since there are no noise sources included in the rdiff model.

This device is supported within altergroups.

This device is dynamically loaded from the shared object

`/vobs/spectre_dev/tools.sun4v/spectre/lib/cmi/2.0.doc/libphilips.so`

Sample Instance Statement

```
r2 (1 2 0) rdsn l=9u w=2u nb=0 m=1
```

Sample Model Statement

```
model rdsn rdiff level=1 tr=27 dta=0 rshr=2.5e3 wtol=0.22u rint=3.5u swvp=13.4u
power=2 tcr1=1.5e-3 tcr2=1e-5 vpr=40
```

Instance Definition

```
Name  h  l  [b] ModelName parameter=value ...
```

Instance Parameters

1 `l=1.0 scale m`Drawn length of resistor. Must be greater than zero. Scale set by option scale.

2 `w=1.0 scale m`Drawn width of resistor. Must be greater than zero. Scale set by option scale.

3 `nb=0.0`Number of bends in the resistor. Must be greater or equal zero.

4 `m=1.0`Multiplicity factor. Must be greater than zero.

Model Definition

```
model modelName rdiff parameter=value ...
```

Model Parameters

1 `level=1.0` Level of this model. Must be 1.

2 `tr (C)` Reference temperature. Default set by option `tnom`.

3 `tref (C)` Alias of `tr`. Default set by option `tnom`.

4 `tnom (C)` Alias of `tr`. Default set by option `tnom`.

5 `dta=0 K` Temperature offset of the device.

6 `trise=0 K` Alias of dta.

7 `rshr=1.0e+3 Ω/sqr`
Sheet resistance at reference temperature. Must be greater than zero.

8 `wtol=0.0 m` Offset between the drawn and effective resistor width.

9 `tcr1=0.0 1/K` Linear temperature coefficient of the resistor.

10 `tcr2=0.0` $1/K^2$ Quadratic temperature coefficient of the resistor.

11 `vpr=100.0 V` Reference Pinch-off voltage.

12 `swvp=0.0 V/m` Coefficient of the width dependence of `vp`.

13 `power=1.5` Voltage exponent. Must be greater than zero.

14 `vdr=1.0 V` Diffusion voltage at reference temperature.

15 `rint=0.0 Ω m` Interface resistance at reference temperature.

16 `tcrint1=0.0 1/K` Linear temperature coefficient of the interface resistor.

Output Parameters

1 `vd (V)` Diffusion voltage. Must be greater than zero.

2 `rsh (Ω/sqr)` Sheet resistance. Must be greater than zero.

3 `vp (V)` Pinch-off voltage. Must be greater than zero.

4 `r0 (Ω)` Zero bias resistance. Must be greater than zero.

Operating-Point Parameters

1 `vhl (V)` Absolute value of the applied bias across the resistor.

2 `vbl (V)` Absolute value of the back-bias across the resistor.

3 `ih (A)` DC current into the resistor.

4 `r (Ω)` Actual resistance value.

5 `pwr (W)` Power.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Four Terminal Relay (relay)

Description

The four-terminal relay is a voltage controlled relay tied between terminals `t1` and `t2`. The voltage between terminals `ps` and `ns` controls the relay resistance. The relay resistance varies nonlinearly between `ropen` and `rclosed`, the open relay resistance and closed relay resistance, respectively. These resistance values correspond to control voltages of `vt1` and `vt2` respectively. The four parameters, `vt1`, `vt2`, `ropen`, and `rclosed`, can be instance or model parameters.

As an alternative, you can specify the threshold voltage `vth` and a transition width `trans` rather than specifying `vt1` and `vt2`. These two parameters are then calculated from `vth` and `trans`. If all four parameters are specified, `vth` and `trans` override `vt1` and `vt2`. However, `vt1` and `vt2` values you specify on the instance override any model parameter specifications.

The final model parameter, `hysteresis`, designates a hysteresis with the on voltage shifted from `vth` by an amount `hysteresis` and the off voltage shifted by the same amount in the opposite direction. The direction of shift depends on the sign of `trans` (or the relative magnitudes of `vt1` and `vt2`): if `trans` is positive, the on voltage shifts by `+hysteresis`; if trans is negative (implying that the relay is normally on), the on-voltage shifts by `-hysteresis`.

This device is not supported within altergroup.

Operating conductance is calculated from the instance parameters as follows:

When Vc lies between vt1 and vt2,

G = Gmin + (Gmin - Gmax) * [2 * (Vc - vt1)^3

 - 3 * (vt2 - vt1) * (Vc - vt1)^2] / (vt2 - vt1)^3

Otherwise, if vt1 < vt2, then

G = Gmin   for Vc < vt1 and

G = Gmax   for Vc > vt2.

If vt1 > vt2,

G = Gmin   for Vc > vt1 and

G = Gmax   for Vc < vt2.

where Gmin = 1 / ropen, Gmax = 1 / rclosed, and Vc = V(ps) - V(ns).

Sample Instance Statement

```
rel1 (1 2 ps ns) my_relay ropen=1G rclosed=2
```

Sample Model Statement

```
model my_relay relay vt1=2.5 vt2=5 ropen=100M rclosed=0.1
```

Instance Definition

```
Name  1  2  ps  ns ModelName parameter=value ...
Name  1  2  ps  ns relay parameter=value ...
```

Instance Parameters

1 `vt1 (V)`Relay resistance is `ropen` at this voltage.

2 `vt2=vt1+1.0 V`Relay resistance is `rclosed` at this voltage.

3 `ropen=∞ Ω`
Resistance of a fully open relay.

4 `rclosed=1.0 Ω`Resistance of a fully closed relay.

5 `m=1.0`Multiplicity factor.

6 `region=off`Estimated operating region.
Possible values are `off` or on.

Model Definition

```
model modelName relay parameter=value ...
```

Model Parameters

1 `vt1 (V)`Relay resistance is `ropen` at this voltage.

2 `vt2=vt1+1.0 V`Relay resistance is `rclosed` at this voltage.

3 `ropen=∞ Ω`
Resistance of a fully open relay.

4 `rclosed=1.0 Ω`Resistance of a fully closed relay.

5 `hysteresis=0.0 V`Switching Hysteresis.

6 `vth=0.0 V`Threshold Voltage.

7 `trans=0.0 V`Switch Transition Region Width.

Operating-Point Parameters

1 `region=off`Estimated operating region.
Possible values are `off` or `on`.

2 `res (Ω)`Relay resistance.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

hysteresis M-5	region I-6	ropen M-3	vt2 I-2
m I-5	region OP-1	trans M-7	vt2 M-2
rclosed I-4	res OP-2	vt1 M-1	vth M-6
rclosed M-4	ropen I-3	vt1 I-1	

Two Terminal Resistor (resistor)

Description

You can give the resistance explicitly or allow it to be computed from the physical length and width of the resistor. In either case, the resistance can be a function of temperature or applied voltage.

This device is supported within altergroups.

If R(inst) is not given,

R(inst) = R(model)

if R(model) is given, and

R(inst) = Rsh * (L - 2 * etchl) / (W - 2 * etch)

otherwise.

If the polynomial coefficients vector (`coeffs=[c1 c2 ...]`) is specified, the resistor is nonlinear. When `nonlinform` is set to `g`, the resistance is

R(V) = dV / dI

= R(inst) / (1 + c1 * V + c2 * V^2 + ...).

The branch current as a function of applied voltage is

I(V) = (V / R(inst)) * (1 + 1/2 * c1 * V + 1/3 * c2 * V^2 + ...)

When `nonlinform` is set to `r`, the resistance is

R(V) = dV / dI

= R(inst) * (1 + c1 * V + c2 * V^2 + ...).

where ck is the kth entry in the coefficient vector.

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

R(T) = R(tnom) * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2]

where

T = trise(inst) + temp

if trise(inst) is given, and

T = trise(model) + temp

otherwise.

Sample Instance Statement

Without model:

r1 (1 2) resistor r=1.2K m=2

With model:

r1 (1 2) resmod l=8u w=1u

Sample Model Statement

```
model resmod resistor rsh=150 l=2u w=2u etch=0.05u tc1=0.1 tnom=27 kf=1
```

Instance Definition

```
Name  1  2 ModelName parameter=value ...
Name  1  2 resistor parameter=value ...
```

Instance Parameters

1 `r (Ω)`Resistance.

2 `l (m)`Resistor length.

3 `w (m)`Resistor width.

4 `m=1`Multiplicity factor.

5 `scale=1`Scale factor.

6 `resform`Use the resistance form for this instance. Default is `yes` if `r < thresh`.
Possible values are `no` or `yes`.

7 `tc1=0 1/C`Linear temperature coefficient.

8 `tc2=0` C^{-2}Quadratic temperature coefficient.

9 `trise (C)`Temperature rise from ambient.

10 `isnoisy=yes`Should resistor generate noise.
Possible values are `no` or `yes`.

The instance parameter `scale`, if specified, overrides the value given by the `option` parameter `scale`. The `w` and `l` parameters are scaled by the resulting `scale`, and the `option` parameter `scalem`. The values of `w` and `l` printed out by spectre are those given in the input, and these values might not have the correct units if the scaling factors are not unity. The actual effective resistor dimensions are stored in the output parameters. You can obtain these dimensions by using the info statement.

Model Definition

```
model modelName resistor parameter=value ...
```

Model Parameters

Resistance Parameters

1 `r=∞ Ω`Default resistance.

2 `rsh=∞ Ω/sqr`
Sheet resistance.

3 `thresh=1.0e-3 Ω`Resistances smaller than this will use the resistance form, as opposed to the standard conductance form.

Resistor Size Parameters

4 `l=∞ m`Default resistor length.

5 `w=1e-6 m`Default resistor width.

6 `etch=0 m`Width narrowing due to etching per side.

7 `etchl=0 m`Length narrowing due to etching per side.

8 `scaler=1`Resistance scaling factor.

Temperature Effects Parameters

9 `tc1=0 1/C`Linear temperature coefficient.

10 `tc2=0` C^{-2}Quadratic temperature coefficient.

11 `tnom (C)`Parameters measurement temperature. Default set by `options`.

12 `trise=0 C`Default temperature rise from ambient.

Nonlinearity Coefficients

13 `coeffs=[...]`Vector of polynomial conductance coefficients.

14 `nonlinform=g`The form of the nonlinear resistance.
Possible values are `g` or `r`.

Noise Model Parameters

15 `kf=0`Flicker (1/f) noise coefficient.

16 `af=2`Flicker (1/f) noise exponent.

17 `wdexp=1`Flicker (1/f) noise W exponent.

18 `ldexp=1`Flicker (1/f) noise L exponent.

19 `weexp=0`Flicker (1/f) noise W effective exponent.

20 `leexp=0`Flicker (1/f) noise L effective exponent.

21 `fexp=1`Flicker (1/f) noise frequency exponent.

DC-Mismatch Model Parameters

22 `mr=0.0`Resistor mismatch dependence.

23 `mrl=0.0`Resistor mismatch length dependence.

24 `mrlp=0.0`Resistor mismatch length power dependence.

25 `mrw=0.0`Resistor mismatch width dependence.

26 `mrwp=0.0`Resistor mismatch width power dependence.

27 `mrlw1=0.0`Resistor mismatch area 1 dependence.

28 `mrlw1p=0.0`Resistor mismatch area 1 power dependence.

29 `mrlw2=0.0`Resistor mismatch area 2 dependence.

30 `mrlw2p=0.0`Resistor mismatch area 2 power dependence.

The instance parameter `resform` and the model parameter `thresh` control whether a resistor is formulated in the standard conductance form, or in the resistance form. If the value of the resistor is smaller than `thresh`, Spectre uses the resistance form; otherwise it uses the conductance form. If `resform` is set on an instance, it overrides the `thresh` parameter. The resistance form is appropriate for very small resistances and the conductance form is intended for larger resistances. Using the conductance form for very small resistances or the resistance form for very large resistances can cause convergence problems.

With the resistance form, the resistance can be zero; with the conductance form, the resistance can be infinite. The resistance form is less efficient than the conductance form. You cannot change the formulation of a resistor once it has been determined. Spectre makes this choice by comparing the initial value of the resistance to `thresh`.

Modeling AC Resistance

In certain situations, a part of a circuit that is required to calculate the DC operating point needs to be removed during a subsequent AC analysis or visa versa. An example of a situation in which this occurs is when measuring the loop gain of a feedback amplifier. In this case the feedback loop must be removed when computing the AC response of the amplifier. In Spectre, the most accurate method of doing this is to use an ideal switch component (see `spectre -h switch`), e.g.

```
Vin (pin 0) vsource mag=1
    OA1 (pin nin out) opamp
    Sw1 (nin out 0) switch position=1 ac_position=2
    LoopGain ac start=1 stop=1MHz
```

Another possibility is that the resistance of an instance changes from one analysis to another. The following subcircuit models a resistance whose value is given by the parameter `rac` during AC analyses, and `rdc` for all other analyses.

```
subckt ac_res (a b)
    parameters rdc=1 rac=2
    R1 (a i) resistor r=rdc
    Rac (i b) resistor r=rac-rdc
    Sw (i b) switch position=1 ac_position=0
ends ac_res
```

Output Parameters

1 `leff (m)`Effective resistor length.

2 `weff (m)`Effective resistor width.

3 `reff (Ω)`Effective resistance.

Operating-Point Parameters

1 `v (V)`Voltage at operating point.

2 `i (A)`Current through the resistor.

3 `res (Ω)` Resistance at op point.

4 `pwr (W)` Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

af M-16	leff O-1	pwr OP-4	tc2 I-8
coeffs M-13	m I-4	r M-1	thresh M-3
etch M-6	mr M-22	r I-1	tnom M-11
etchl M-7	mrl M-23	reff O-3	trise I-9
fexp M-21	mrlp M-24	res OP-3	trise M-12
i OP-2	mrlw1 M-27	resform I-6	v OP-1
isnoisy I-10	mrlw1p M-28	rsh M-2	w M-5
kf M-15	mrlw2 M-29	scale I-5	w I-3
l I-2	mrlw2p M-30	scaler M-8	wdexp M-17
l M-4	mrw M-25	tc1 M-9	weexp M-19
ldexp M-18	mrwp M-26	tc1 I-7	weff O-2
leexp M-20	nonlinform M-14	tc2 M-10	

s-Domain Linear Current Controlled Current Source (scccs)

Description

The device output is defined through a transfer function given as a ratio of two polynomials in the complex variable s. Polynomials can be specified in terms of either coefficients or roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

To specify the polynomial in terms of the coefficients, you enter them as a vector in ascending order of the power of the variable s, starting from the constant term. For example, to specify a denominator of 3s^2 + 4s + 1, use `denom=[1 4 3]`.

To specify a polynomial in terms of its roots, you give the roots as a vector of complex frequencies (frequencies should be in radians/second). You must give both the real and imaginary parts of the root, even when the root is real. For the transfer function to be stable, all poles must have negative real values. When specifying a complex root, you should also specify its complex conjugate. However, if you omit the conjugate root, Spectre will supply the missing root and print a warning that a missing root was supplied. The order of the roots is not important. For example, to specify poles of s = -1, s = 4j, s = -4j, s = -2 + 2j, and s = -2 - 2j; use `poles=[-1 0 0 4 0 -4 -2 2 -2 -2]`.

Either the numerator or the denominator specification can be omitted. An omitted denominator or numerator is taken to be 1.

The parameter `gain` is interpreted either as the DC gain or, if the function has zeros or poles at the origin, as a constant factor.

This device is not supported within altergroup.

Sample Instance Statement

```
l1 (2 1) inductor l=15
sc1 (1 0) scccs probe=l1 zeros=[0 6 0 -6 2 -8 2 8] poles=[-1 0 0 64 0 -64 -2 8 -2 -8]
```

Instance Definition

```
Name  sink  src scccs parameter=value ...
```

Instance Parameters

1 `probe`Device through which the controlling current flows.

2 `port=0`Index of the probe port through which the controlling current flows.

3 `gain=1`Transfer function gain.

4 `numer=[...]`Vector of numerator coefficients.

5 `denom=[...]`Vector of denominator coefficients.

6 `zeros=[...]`Vector of complex zeros.

7 `poles=[...]`Vector of complex poles.

8 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)` Input current.

2 `v (V)` Output voltage.

3 `pwr (W)` Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

denom I-5	m I-8	port I-2	v OP-2
gain I-3	numer I-4	probe I-1	zeros I-6
i OP-1	poles I-7	pwr OP-3	

s-Domain Current Controlled Voltage Source (sccvs)

Description

The device output is defined through a transfer function given as a ratio of two polynomials in the complex variable s. Polynomials can be specified in terms of either coefficients or roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

To specify the polynomial in terms of the coefficients, you enter them as a vector in ascending order of the power of the variable s, starting from the constant term. For example, to specify a denominator of 3s^2 + 4s + 1, use `denom=[1 4 3]`.

To specify a polynomial in terms of its roots, you give the roots as a vector of complex frequencies (frequencies should be in radians/second). You must give both the real and imaginary parts of the root, even when the root is real. For the transfer function to be stable, all poles must have negative real values. When specifying a complex root, you should also specify its complex conjugate. However, if you omit the conjugate root, Spectre will supply the missing root and print a warning that a missing root was supplied. The order of the roots is not important. For example, to specify poles of s = -1, s = 4j, s = -4j, s = -2 + 2j, and s = -2 - 2j; use `poles=[-1 0 0 4 0 -4 -2 2 -2 -2]`.

Either the numerator or the denominator specification can be omitted. An omitted denominator or numerator is taken to be 1.

The parameter `gain` is interpreted either as the DC gain or, if the function has zeros or poles at the origin, as a constant factor.

This device is not supported within altergroup.

Sample Instance Statement

```
myv (1 0) vsource type=sine freq=10K
scc1 (2 0) sccvs probe=myv gain=0.5 numer=[2] denom=[5]
```

Instance Definition

```
Name  p  n sccvs parameter=value ...
```

Instance Parameters

1 `probe`Device through which the controlling current flows.

2 `port=0`Index of the probe port through which the controlling current flows.

3 `gain=1`Transfer function gain.

4 `numer=[...]`Vector of numerator coefficients.

5 `denom=[...]`Vector of denominator coefficients.

6 `zeros=[...]`Vector of complex zeros.

7 `poles=[...]`Vector of complex poles.

8 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

denom I-5	m I-8	port I-2	v OP-2
gain I-3	numer I-4	probe I-1	zeros I-6
i OP-1	poles I-7	pwr OP-3	

s-Domain Linear Voltage Controlled Current Source (svccs)

Description

The device output is defined through a transfer function given as a ratio of two polynomials in the complex variable s. Polynomials can be specified in terms of either coefficients or roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

To specify the polynomial in terms of the coefficients, you enter them as a vector in ascending order of the power of the variable s, starting from the constant term. For example, to specify a denominator of 3s^2 + 4s + 1, use `denom=[1 4 3]`.

To specify a polynomial in terms of its roots, you give the roots as a vector of complex frequencies (frequencies should be in radians/second). You must give both the real and imaginary parts of the root, even when the root is real. For the transfer function to be stable, all poles must have negative real values. When specifying a complex root, you should also specify its complex conjugate. However, if you omit the conjugate root, Spectre will supply the missing root and print a warning that a missing root was supplied. The order of the roots is not important. For example, to specify poles of s = -1, s = 4j, s = -4j, s = -2 + 2j, and s = -2 - 2j; use `poles=[-1 0 0 4 0 -4 -2 2 -2 -2]`.

Either the numerator or the denominator specification can be omitted. An omitted denominator or numerator is taken to be 1.

The parameter `gain` is interpreted either as the DC gain or, if the function has zeros or poles at the origin, as a constant factor.

This device is not supported within altergroup.

Sample Instance Statement

```
s2 (1 0 control 0) svccs gain=0.4 numer=[2 3] denom=[4 5 1]
```

Instance Definition

```
Name  sink  src  ps  ns svccs parameter=value ...
```

Instance Parameters

1 `gain=1`Transfer function gain.

2 `numer=[...]`Vector of numerator coefficients.

3 `denom=[...]`Vector of denominator coefficients.

4 `zeros=[...]`Vector of complex zeros.

5 `poles=[...]`Vector of complex poles.

6 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

denom I-3	m I-6	pwr OP-3
gain I-1	numer I-2	v OP-2
i OP-1	poles I-5	zeros I-4

s-Domain Voltage Controlled Voltage Source (svcvs)

Description

The device output is defined through a transfer function given as a ratio of two polynomials in the complex variable s. Polynomials can be specified in terms of either coefficients or roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

To specify the polynomial in terms of the coefficients, you enter them as a vector in ascending order of the power of the variable s, starting from the constant term. For example, to specify a denominator of 3s^2 + 4s + 1, use `denom=[1 4 3]`.

To specify a polynomial in terms of its roots, you give the roots as a vector of complex frequencies (frequencies should be in radians/second). You must give both the real and imaginary parts of the root, even when the root is real. For the transfer function to be stable, all poles must have negative real values. When specifying a complex root, you should also specify its complex conjugate. However, if you omit the conjugate root, Spectre will supply the missing root and print a warning that a missing root was supplied. The order of the roots is not important. For example, to specify poles of s = -1, s = 4j, s = -4j, s = -2 + 2j, and s = -2 - 2j; use `poles=[-1 0 0 4 0 -4 -2 2 -2 -2]`.

Either the numerator or the denominator specification can be omitted. An omitted denominator or numerator is taken to be 1.

The parameter `gain` is interpreted either as the DC gain or, if the function has zeros or poles at the origin, as a constant factor.

This device is not supported within altergroup.

Sample Instance Statement

```
e1 (1 0 control 0) svccs gain=5 poles=[-1 0 1 0] zero=[0 0 1 0]
```

Instance Definition

```
Name  p  n  ps  ns svcvs parameter=value ...
```

Instance Parameters

1 `gain=1`Transfer function gain.

2 `numer=[...]`Vector of numerator coefficients.

3 `denom=[...]`Vector of denominator coefficients.

4 `zeros=[...]`Vector of complex zeros.

5 `poles=[...]`Vector of complex poles.

6 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Ideal Switch (switch)

Description

Ideal switch is a single-pole multiple-throw switch with infinite `off` resistance and zero on resistance. The switch is provided to allow you to reconfigure your circuit between analyses. You can only change the switch state between analyses (using the alter statement), not during an analysis.

When the switch is set to position 0 it is open. In other words, no terminal is connected to any other. When the switch is set to position 1, terminal 1 is connected to terminal 0, and all others are unconnected. When the position is set to 2, terminal 2 is connected to terminal 0, etc.

An offset voltage is supported. It is placed in series with the common terminal. The negative side of the source is connected to the common terminal.

The switch can change its position based on which analysis type is being performed using the `xxx_position` parameters. This feature should be used carefully. Careless use can generate discontinuities that result in convergence problems. Once an analysis specific position has been specified using `xxx_position`, it will always dominate over a position given with the `position` parameter. To disable an analysis specific position, alter it to its default value of unspecified.

This device is not supported within altergroup.

Sample Instance Statement

```
sw1 (t1 t2 t3) switch dc_position=0 ac_position=1 tran_position=2
```

Instance Definition

```
Name  t0  t1 ... switch parameter=value ...
```

Instance Parameters

1 `position=0`Switch position (0, 1, 2, ...).

2 `dc_position`Position to which switch is set at start of DC analysis.

3 `ac_position`Position to which switch is set at start of AC analysis.

4 `tran_position`Position to which switch is set at start of transient analysis.

5 `ic_position` Position to which switch is set at start of IC analysis (precedes transient analysis).

6 `offset=0` Offset voltage in series with common terminal.

7 `m=1.0` Multiplicity factor.

Output Parameters

1 `present_position`
Current switch position.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

ac_position I-3	ic_position I-5	offset I-6	present_position O-1
dc_position I-2	m I-7	position I-1	tran_position I-4

Transmission Line (tline)

Description

Lossy or lossless transmission line.

```
                |-----------------|
t1 -o----------|                 |-----------o- t2
    +   I1 ->  |                 |  <- I2    +
    V1         |      tline      |           V2
    -   I1 <-  |                 |  -> I2    -
b1 -o----------|                 |-----------o- b2
                |-----------------|
```

The lossy transmission line model includes dielectric and conductor loss effects. The conductor loss includes skin effect assuming finite or infinite conductor thickness.

Only the odd mode is modeled, so only the voltage difference across each port is important. (The absolute voltage of each terminal is not significant.) Also, the current into one node of a port exactly equals the current leaving the other node of the port.

This device is supported within altergroups.

Sample Instance Statement

```
t1 (1 0 2 0) lmodel z0=100
```

Sample Model Statement

```
model lmodel tline f=10M z0=50 alphac=8501 fc=10M dcr=88
```

Instance Definition

```
Name  t1  b1  t2  b2 ModelName parameter=value ...
Name  t1  b1  t2  b2 tline parameter=value ...
```

Instance Parameters

1 `z0=50 Ω`Characteristic impedance of lossless line.

2 `td (s)`Time delay of a lossless line in seconds, a measure of the electrical length.

3 `f (Hz)`Reference frequency (used in conjunction to the normalized length to specify electrical length of line).

4 `nl=0.25`Normalized electrical length in wavelengths at `f` of a lossless line.

5 `vel=1`Propagation velocity of the line given as a multiple of c, the speed of light in free space. (vel <= 1).

6 `len=0 m`Physical length (used with `vel` to specify electrical length of line).

7 `m=1`Multiplicity factor.

Model Definition

```
model modelName tline parameter=value ...
```

Model Parameters

1 `z0=50 Ω`Characteristic impedance of lossless line.

2 `f (Hz)`Reference frequency (used in conjunction to the normalized length to specify electrical length of line).

3 `vel=1`Propagation velocity of the line given as a multiple of `c`, the speed of light in free space. (vel <= 1).

Conductor Loss Parameters

4 `corner=0 Hz`Corner frequency for skin effect, frequency where skin depth equals the conductors wall thickness.

5 `dcr=0 Ω/m`DC series resistance per unit length.

6 `fc (Hz)`Conductor loss measurement frequency (use with `r`, `qc`, or `alphac`).

7 `r=0 Ω/m`Conductor (series) resistance per unit length at `fc`.

8 `alphac=0 dB/m`Conductor loss at `fc` (low loss approximation).

9 `qc=∞`Conductor loss quality factor at `fc` (low loss approximation).

Dielectric Loss Parameters

10 `fd (Hz)`Dielectric loss measurement frequency (use with `qd`).

11 `g=0 S/m`Dielectric (shunt) conductance per unit length.

12 `alphad=0 dB/m`Dielectric loss (low loss approximation).

13 `qd=∞`Dielectric loss quality factor at `fd` (low loss approximation).

Lossless Case

The lossless transmission line is specified with parameters `z0` and `td`. The device behavior is then:

V1(t) - z0*I1(t) = V2(t-td) + z0*I2(t-td) and

V2(t) - zO*I2(t) = V1(t-td) + z0*I1(t-td).

where t is time and `td` is the delay. Note, if the device is terminated by a matched impedance of `z0` (across t2 and b2), then it becomes an ideal delay. i.e V2(t) = V1(t-td).

To Model Both Even and Odd Modes

Use two lines as shown below:

```
            tline_inner
                   |---------------|
     i1------------|    z0_inner   |--------------i2
                   |    td_inner   |
               ----|---------------|-------
               |                          |
     o1--------|      tline_outer         |-------o2
               |   |---------------|      |
               ----|    z0_outer   |-------
                   |    td_outer   |
         ----------|---------------|-------------
         |                                      |
        gnd                                    gnd
```

This model is suitable for a coax where tline_inner models the inner/outer conductor line (or the odd mode) while tline_outer models the outer/ground line (or the even mode). Note that this model is non-symmetric.

Lossy Case

In the frequency-domain the device is modeled by

V1(jw) - Z(jw)*I1(jw) = S12(jw)* [V2(jw) + Z(jw)I2(jw)]

and

V2(jw) - Z(jw)*I2(jw) = S21(jw)* [V1(jw) + Z(jw)I1(jw)]

where j=sqrt(-1) and w is the angular frequency in radians/s.

The loss coefficient is computed from

S21(jw) = S12(jw) = exp(-Gamma(jw)*len)

where

Gamma(jw) = sqrt (Zc(jw) * Yd(jw))

where Zc represents the per-unit-length series impedance and Yd represents the per-unit-length shunt admittance loss (as described below).

The characteristic impedance (Z) is computed from

Z(jw) = sqrt(Zc(jw) / Yd(jw)).

The time-domain behavior of the lossy transmission line is computed through a recursive convolution algorithm.

The dielectric loss (Yd) is computed from

Yd(jw) = G + j* w/(z0*c*vel)

where G is the per-unit-length shunt conductance and can be specified in three ways.

- G = g { when g is given }
- G = 2/z0 * alphad { when `alphad` is given }
- G = 2/z0 * fd/(2*qd*c*vel) { when `fd` and `qd` are given }
  - where `c` is the speed of light.

The series impedance (Zc) is computed from

Zc(jw) = Zi + j*w*z0/(c*vel).

where Zi represents the internal loss. When skin effect is not present then

Zi = dcr

where `dcr` is the DC series per-unit-length resistance.

Skin Effect Assuming Finite Thickness

In this case the internal impedance (Zi) is computed from

Zi = Ri + j*w*Li

where Ri and Li exhibit the following behavior

```
Ri                                    Li
 |                         **          |
 |                     ****          L0|********
 |                *****                |        **
 |          ******                     |          *
```

```
dcr|*********                          |           **
   |                                   |             **********
   ----------|-----------------       ----------|-------------****
         W_corner          (freq)           W_corner        (freq)
```

The expressions for Ri and Li are

when w << W_corner: Ri ~ dcr   and   Li ~ dcr/(1.5*W_corner)

when w >> W_corner: Ri ~ dcr*sqrt(w/W_corner)

and

Li ~ dcr/(sqrt(w*W_corner))

Otherwise: Ri = dcr * nt * (sinh(2*nt)+sin(2*nt))/(cosh(2*nt)-cos(2*nt))

and

Li = dcr * nt * (sinh(2*nt)-sin(2*nt))/(cosh(2*nt)-cos(2*nt)) / w

where nt=sqrt(w/W_corner) is the normalized thickness

The equations can be found in:

Ramo, Whinnery, Van Duzer. Fields and waves in communication electronics. 1965. See Section on "Impedance of thin-walled conductors". Pg 301.

The corner frequency (W_corner) results from skin effect on conductors of finite thickness. As frequency decreases, skin depth increases resulting in more conductor to pass the current, which results in lower loss. However, at the corner frequency, the skin depth equals the radius of the conductor. Decreasing the frequency below that point does not further reduce the loss.

The corner frequency (W_corner) can be specified in two ways.

- When `dcr` and `corner` are given, then

  W_corner= 2*pi*corner

- When `dcr`, `r`, and `fc` are given, then

  W_corner = 2*pi*fc* (dcr/r)^2

In addition, there are two alternative ways to specify `r`.

- r = 2*z0*alphac   { when alphac is given }
- r = 2*z0*fc/(2*qc*c*vel)   { when qc is given }

where c is the speed of light are defined below.

Skin Effect Assuming Infinite Thickness

In this case there is no corner frequency (and no `dcr`), and the internal loss (Zi) is computed from

Zi= Ri + j*w*Li

where Ri = r*sqrt(w/(2*pi*fc)) and Li = r/sqrt(w*2*pi*fc).

Again, `r` can be specified directly, or using `alphac` or `qc` as described above in the case of finite thickness.

Three Ways to Specify `vel`, `td`, and `len`

- When `vel` and `len` are given

  td = len/(vel*c)

- When `td` and `vel` are given

  len = td*vel*c

- When `f`, `nl` and `vel` are given

  td = nl/f

  len = (nl/f)*vel*c

The parameter `len` is the physical length, `c` is the speed of light and `vel` is the propagation velocity as a multiple of `c`. (Recall that velocity = c/sqrt(relative dielectric constant).) The parameter `f` is a reference frequency and `nl` is the normalized electrical length in wavelengths at `f`.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

GaAs MESFET (tom2)

Description

TOM2 stands for Triquint Own Model version-2. It is an improved GaAs MESFET developed by David H. Smith. The charge model in TOM2 is similar to that of the Statz model and does not conserve charge. Therefore, this model should not be used to simulate circuits that requires charge conservation such as charge-pump circuits. TOM2 GaAs MESFET instances require that you use a model statement.

This device is supported within altergroups.

There are some convergence problems with this model because of `Cgs` going to zero beyond pinchoff. The problems occur when the gate is driven from an inductive source, and there is no other capacitance at the gate. To prevent these problems, avoid setting `Cgd` to zero and add side wall capacitance to the gate-source and gate-drain junctions. A good estimate for these capacitors is C = pi*epsilon*w/2 where w is the gate width in microns and epsilon = 0.116 fF/micron.

Sample Instance Statement

```
mt1 (2 1 0) tom2mos area=1 region=fwd
```

Sample Model Statement

```
model tom2mos tom2 vto=-0.55 alpha=3.9 beta=0.001 gamma=0.075 delta=100 ng=1 rd=550
rs=550 rg=1 is=0.295e-14 n=1.2 cgs=1.4e-15 cgd=2e-16 cds=3e-16
```

Instance Definition

```
Name  d  g  s ModelName parameter=value ...
```

Instance Parameters

1 `area=1`Junction area factor.

2 `m=1`Multiplicity factor.

3 `region=fwd`Estimated operating region.
Possible values are `off`, `triode`, `sat`, or `subth`.

Model Definition

```
model modelName tom2 parameter=value ...
```

Model Parameters

Device Type Parameters

1 `type=n`Transistor type.
Possible values are n or p.

Drain Current Parameters

2 `vto=-2.5 V`Threshold voltage.

3 `alpha=2 1/V`Knee-voltage parameter.

4 `beta=0.1 A/V`^{2}Transconductance parameter.

5 `gamma=0 1/V`Threshold shifting parameter.

6 `delta=0.2 V`Output feedback parameter.

7 q=2Power-law parameter.

Subthreshold Parameters

8 `ng=0`Subthreshold slope gate parameter.

9 `nd=0 1/V`Subthreshold slope drain pull parameter.

Parasitic Resistance Parameters

10 `rd=0 Ω`Drain resistance (/area).

11 `rs=0 Ω`Source resistance (/area).

12 `rg=0 Ω`Gate resistance (/area).

13 `minr=0.1 Ω`Minimum source/drain/gate resistance.

Junction Diode Model Parameters

14 `is=1e-14 A`Gate diode saturation current (*area).

15 `n=1`Emission coefficient for the gate junction.

16 `imelt=`imaxA'`Explosion current (*area).

17 `dskip=yes`Use simple piece-wise linear model for diode currents below 0.1*`iabstol`. Possible values are `no` or `yes`.

Junction Capacitance Model Parameters

18 `capmod=2`Charge model selector.

19 `cgs=0 F`Gate-source zero-bias junction capacitance (*area).

20 `cgd=0 F`Gate-drain zero-bias junction capacitance (*area).

21 `cds=0 F`Drain-to-source capacitance.

22 `vbi=1 V`Gate diode built-in potential.

23 `vmax=0.95`Gate diode capacitance limiting voltage.

24 `vdelta=0.2 V`Capacitance transition voltage.

25 `tau=0 s`Conduction current delay time.

Temperature Effects Parameters

26 `tnom (C)`Parameters measurement temperature. Default set by options.

27 `xti=0`Temperature exponent for effect on `is`.

28 `eg=1.11 V`Energy band gap.

29 `vtotc=0 V/C`Temperature coefficient for `vto`.

30 `vbitc=0 V/C`Temperature coefficient for `vbi`.

31 `alphatce=0 1/C`Temperature coefficient for `alpha`.

32 `betatce=0 1/C`Temperature coefficient for `beta`.

33 `gammatc=0 1/C`Temperature coefficient for `gamma`.

34 `trs1=0 1/C`Temperature parameter for source resistance.

35 `trd1=0 1/C`Temperature parameter for drain resistance.

36 `trg1=0 1/C`Temperature parameter for gate resistance.

37 `cgdtce=0 1/C`Drain junction capacitance temperature coefficient.

38 `cgstce=0 1/C`Source junction capacitance temperature coefficient.

Operating Region Warning Control Parameters

39 `imax=1 A`Maximum allowable current (*area).

40 `bvj=`∞ `V`Junction reverse breakdown voltage.

Noise Model Parameters

41 `kf=0`Flicker (1/f) noise coefficient.

42 `af=1`Flicker (1/f) noise exponent.

43 `kfd=0`Flicker noise (1/f) coefficient for gate diodes.

44 `afg=1`Flicker noise (1/f) exponent for gate diodes.

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the FET are accurately modeled for currents up to `imax`. For currents above `imax`, the junction is modeled as a linear resistor and a warning is printed. The `bv` parameter detects the junction breakdown only. The breakdown currents of the junctions are not modeled.

Operating-Point Parameters

1 `type=n`Transistor type.
Possible values are `n` or `p`.

2 `region=fwd`Estimated operating region.
Possible values are `off`, `triode`, `sat`, or `subth`.

3 `vgs (V)`Gate-source voltage.

4 `vds (V)`Drain-source voltage.

5 `id (A)`Drain current.

6 `ig (A)`Gate current.

7 `ids (A)`Drain-to-source current.

8 `gm (S)`Common-source transconductance.

9 `gds (S)`Common-source output conductance.

10 `vth (V)`Threshold voltage.

11 `cgs (F)`Gate-source capacitance.

12 `cgd (F)`Gate-drain capacitance.

13 `cds (F)`Drain-source capacitance.

14 `pwr (W)`Power at op point.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Linear Two Winding Ideal Transformer (transformer)

Description

Winding 1 connects terminals `t1` and `b1`, and winding 2 connects `t2` and `b2`. The number of turns on windings 1 and 2 are given by `n1` and `n2`, respectively, and `n2` must not be zero. The absolute number of turns of each winding is not important, only the ratio of `n1` to `n2`. Current through winding 1 is computed.

This device is not supported within altergroup.

An ideal transformer is modeled, so it acts as a transformer at DC. Thus

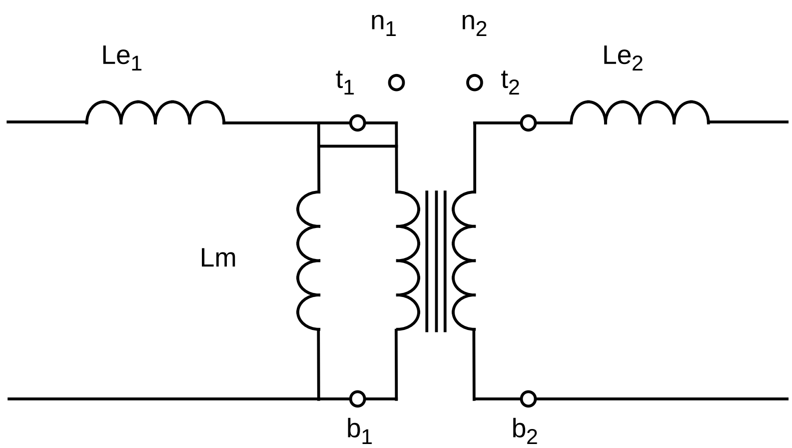
$$\frac{v_1}{v_2} = \frac{t_1}{t_2} = \frac{i_2}{i_1}$$

To model a physical transformer with L_1 and L_2 as the inductance of the windings and k as the coupling coefficient, add an inductor $L_m = k\,L_1$ in parallel with winding 1 and inductors $L_{e1} = L_1\,(1 - k)$ and $L_{e2} = L_2\,(1 - k)$ in series with windings 1 and 2, respectively. The turns ratio can be computed with

$$\frac{n_1}{n_2} = \sqrt{\frac{L_1}{L_2}}$$

k can be calculated from the L_1 (the inductance of winding 1 with winding 2 open) and L_s (the inductance of winding 1 with winding 2 shorted) with

$$k = \sqrt{1 - \frac{L_s}{L_1}}$$



Linear Two-Winding Ideal Transformer

Instance Definition

```
Name  t1  b1  t2  b2 transformer parameter=value ...
```

Instance Parameters

1 n1=1Number of turns on winding 1.

2 n2=1Number of turns on winding 2.

3 m=1Multiplicity factor.

VBIC Bipolar Transistor (vbic)

Description

The VBIC model was developed as a replacement for the SPICE G-P model. The model has four electrical terminals, two thermal terminals, and up to nine internal nodes, depending on the model parameters that the user specifies. Detailed description of the model and equations are given in the Spectre Circuit Simulator Device Model Equations manual.

This device is supported within altergroups.

Sample Instance Statement

```
q1 (1 2 0 0 0) vbjt area=1
```

Sample Model Statement

```
model vbjt vbic type=npn is=2e-16 iben=4.5e-15 isp=1e-15 gamm=1.55e-11 ikf=0.0021
ikr=0.0021 vef=15 ver=7 rbi=35 rbx=7 re=3 rs=15 cje=1.5e-14 tf=15e-12 selft=yes
rth=1K
```

Instance Definition

```
Name  c  b  e  [s]  [dt]  [tl] ModelName parameter=value ...
```

`tl` node is the local temperature and the `dt` node is the rise above the local temperature caused by the thermal power dissipated by the device being modeled by VBIC. Consequently, the `tl` node can be connected to a thermal network that models heat flow through the substrate and/or between devices. It is not necessary to specify the substrate and the two thermal terminals. If left unspecified, the substrate and the `tl` thermal terminal are connected to ground. But if the self-heating flag is turned on and `dt` is not given, an internal node is created for self-heating. You must specify the substrate terminal if you specify `dt` and both substrate and `dt` must be given if `tl` needs to be specified.

Instance Parameters

1 `area=1`Transistor area factor.

2 `m=1`Multiplicity factor.

3 `region=fwd`Estimated operating region.
Possible values are `off`, `fwd`, `rev`, `sat`, or `breakdown`.

4 `trise`Temperature rise from ambient.

Model Definition

```
model modelName vbic parameter=value ...
```

Model Parameters

Structural Parameters

1 `type=npn`Transistor type.
Possible values are `npn` or `pnp`.

Saturation Current Parameters

2 `is=1e-16` ATransport saturation current (*area).

3 `ibei=1e-18` AIdeal B-E saturation current. (*area).

4 `iben=0` ANonideal B-E saturation current (*area).

5 `ibci=1e-16` AIdeal B-C saturation current. (*area).

6 `ibcn=0` ANonideal B-C saturation current (*area).

7 `isp=0` AParasitic transport saturation current. (*area).

8 `ibeip=0` AIdeal parasitic B-E saturation current (*area).

9 `ibenp=0` ANonideal parasitic B-E saturation current (*area).

10 `ibcip=0` AIdeal parasitic B-C saturation current (*area).

11 `ibcnp=0` ANonideal parasitic B-C saturation current (*area).

12 `vo=0` VEpi drift saturation voltage.

13 `gamm=0` VEpi doping parameter.

14 `hrcf=1`High current RC factor.

15 `wbe=1`Portion of Ibei from Vbei.

16 `wsp=1`Portion of Iccp from Vbep.

Emission Coefficient Parameters

17 `nf=1`Forward emission coefficient.

18 `nr=1`Reverse emission coefficient.

19 `nei=1`Ideal B-E emission coefficient.

20 `nen=2`Nonideal B-E emission coefficient.

21 `nci=1`Ideal B-C emission coefficient.

22 `ncn=2`Nonideal B-C emission coefficient.

23 `nfp=1`Parasitic forward emission coefficient.

24 `ncip=1`Ideal parasitic B-C emission coefficient.

25 `ncnp=2`Nonideal parasitic B-C emission coefficient.

Current Gain Parameters

26 `ikf=`∞ AForward knee current (*area).

27 `ikr=`∞ AReverse knee current (*area).

28 `ikp=`∞ AParasitic knee current (*area).

Early Voltage Parameters

29 `vef=`∞ VForward Early voltage.

30 `ver=`∞ VReverse Early voltage.

Breakdown Voltage Parameters

31 `avc1=0`B-C weak avalanche parameter.

32 `avc2=0`B-C weak avalanche parameter.

Parasitic Resistance Parameters

33 `rbi=0 Ω`Intrinsic base resistance (/area).

34 `rbx=0 Ω`Extrinsic base resistance (/area).

35 `re=0 Ω`Emitter resistance (/area).

36 `rs=0 Ω`Substrate resistance (/area).

37 `rbp=0 Ω`Parasitic base resistance (/area).

38 `rcx=0 Ω`Extrinsic collector resistance (/area).

39 `rci=0 Ω`Intrinsic collector resistance (/area).

Junction Capacitance Parameters

40 `cje=0 F`B-E zero-bias capacitance (*area).

41 `pe=0.75 V`B-E built-in potential.

42 `me=0.33`B-E grading coefficient.

43 `aje=-0.5`B-E capacitance smoothing factor.

44 `fc=0.9`Forward-bias depletion capacitance limit.

45 `cbeo=0 F`Extrinsic B-E overlap capacitance (*area).

46 `cjc=0 F`B-C zero-bias capacitance (*area).

47 `cjep=0 F`B-C extrinsic zero-bias capacitance (*area).

48 `pc=0.75 V`B-C built-in potential.

49 `mc=0.33`B-C grading coefficient.

50 `ajc=-0.5`B-C capacitance smoothing factor.

51 `cbco=0 F`Extrinsic B-C overlap capacitance (*area).

52 `qco=0 Coul`Epi charge parameter.

53 `cjcp=0 F`S-C zero-bias capacitance (*area).

54 `ps=0.75 V`S-C built-in potential.

55 `ms=0.33`S-C grading coefficient.

56 `ajs=-0.5`S-C capacitance smoothing factor.

Transit Time and Excess Phase Parameters

57 `tf=0 s`Forward transit time.

58 `tr=0 s`Reverse transit time.

59 `td=0 s`Forward excess-phase delay time.

60 `qtf=0`Variation of `tf` with base width modulation.

61 `xtf=0`Coefficient of `tf` with bias dependence.

62 `vtf=0`Coefficient of `tf` dependence on Vbc.

63 `itf=0`Coefficient of `tf` dependence on Ic.

Temperature Effects Parameters

64 `selft=0`Flag denoting self-heating.
Possible values are `no` or `yes`.

65 `tnom (C)`Parameters measurement temperature. Default set by options.

66 `trise=0 C`Temperature rise from ambient.

67 `rth=0 Ω`Thermal resistance, must be given for self-heating.

68 `cth=0 F`Thermal capacitance.

69 `xis=3 V`Temperature exponent of Is.

70 `xii=3 V`Temperature exponent of Ibei, Ibci, Ibeip, and Ibcip.

71 `xin=3 V`Temperature exponent of Iben, Ibcn, Ibenp, and Ibcnp.

72 `tnf=0 V`Temperature coefficient of Nf.

73 `tavc=0 V`Temperature coefficient of Avc2.

74 `ea=1.12 V`Activation energy for `is`.

75 `eaie=1.12 V`Activation energy for Ibei.

76 `eaic=1.12 V`Activation energy for Ibci/Ibeip.

77 `eais=1.12 V`Activation energy for Ibcip.

78 `eane=1.12 V`Activation energy for Iben.

79 `eanc=1.12 V`Activation energy for Ibcn/Ibenp.

80 `eans=1.12 V`Activation energy for Ibcnp.

81 `xre=0`Temperature exponent of `re`.

82 `xrb=0`Temperature exponent of `rb`.

83 `xrc=0`Temperature exponent of `rc`.

84 `xrs=0`Temperature exponent of `rs`.

85 `xvo=0`Temperature exponent of `vo`.

86 `dtmax=226.85 C`Maximum expected device temperature. (500 K).

Noise Model Parameters

87 `kfn=0`B-E flicker (1/f) noise coefficient.

88 `afn=1`B-E flicker (1/f) noise exponent.

89 `bfn=1`B-E flicker (1/f) noise dependence.

Junction Diode Model Control Parameters

90 `dskip=yes`Skip junction calculations if they are reverse-saturated.
Possible values are `no` or `yes`.

91 `imelt=10` AExplosion current (*area).

Operating Region Warning Control Parameters

92 `bvbe=∞` VB-E breakdown voltage.

93 `bvbc=∞` VB-C breakdown voltage.

94 `bvce=∞` VC-E breakdown voltage.

95 `bvsub=∞` VSubstrate junction breakdown voltage.

96 `vbefwd=0.2` VB-E forward voltage.

97 `vbcfwd=0.2` VB-C forward voltage.

98 `vsubfwd=0.2` VSubstrate junction forward voltage.

99 `imax=1` AMaximum allowable base current (*area).

100 `imax1=imax` AMaximum allowable collector current (*area).

101 `alarm=none`Forbidden operating region.
Possible values are `none`, `off`, `fwd`, `rev`, or `sat`.

DC-Mismatch Model Parameters

102 `mvt0=0.0` VThreshold mismatch intercept.

Imax and Imelt

The `imax` parameter aids convergence and prevents numerical overflow. The junction characteristics of the device are accurately modeled for current up to `imax`. If `imax` is exceeded during iterations, the linear model is substituted until the current drops below imax or until convergence is achieved. If convergence is achieved with the current exceeding `imax`, the results are inaccurate, and Spectre prints a warning.

A separate model parameter, `imelt`, is used as a limit warning for the junction current. This parameter can be set to the maximum current rating of the device. When any component of the junction current exceeds `imelt`, note that base and collector currents are composed of many exponential terms, Spectre issues a warning and the results become inaccurate. The junction current is linearized above the value of `imelt` to prevent arithmetic exception, with the exponential term replaced by a linear equation at `imelt`.

Operating-Point Parameters

1 `type=npn` Transistor type.
Possible values are `npn` or `pnp`.

2 `region=fwd` Estimated operating region.
Possible values are `off`, `fwd`, `rev`, `sat`, or `breakdown`.

3 `vbe (V)` Base-emitter voltage.

4 `vbc (V)` Base-collector voltage.

5 `vce (V)` Collector-emitter voltage.

6 `vcs (V)` Collector-substrate voltage.

7 `temp (C)` Device temperature.

8 `ith (A)` Thermal source.

9 `ic (A)` Intrinsic DC collector current. (Icc - Ibc + Igc).

10 `ib (A)` Intrinsic DC base current. (Ibe + Ibc - Igc).

11 `icc (A)` C-E current.

12 `ibe (A)` Intrinsic B-E junction current.

13 `ibc (A)` Intrinsic B-C junction current.

14 `ibex (A)` BX-E junction current.

15 `igc (A)` Breakdown current.

16 `iccp (A)` Parasitic transistor C-E current.

17 `ibep (A)` Parasitic transistor B-E current.

18 `ibcp (A)` Parasitic transistor B-C current.

19 `betadc (A/A)` Ratio of external collector current to external base current. (ic_ext/ib_ext).

20 `gm (S)` Intrinsic small-signal transconductance. (gm = dIcc_dVbe + dIcc_dVbc).

21 `gpi (S)` Intrinsic small-signal input conductance. (gpi = dIbe_dVbe).

22 `go (S)` Intrinsic small-signal output conductance. (go = -dIcc_dVbc).

23 `gmu (S)` Intrinsic small-signal Collector-Base conductance. (gmu = dIbc_dVbc).

24 `cpi (F)` Intrinsic small-signal B-E capacitance. Same as cje.

25 `cmu (F)` Intrinsic small signal B-C capacitance. Same as cjc.

26 `betaac (A/A)` Small-signal common-emitter current gain. (gm/gpi).

27 `ft (Hz)` Unity small-signal current-gain frequency.

28 `dic_dvbe (S)` Intrinsic dIc/dVbe.

29 `dic_dvbc (S)` Intrinsic dIc_dVbc.

30 `dib_dvbe (S)` Intrinsic dIb_dVbe.

31 `dib_dvbc (S)` Intrinsic dIb_dVbc.

32 `rbi (Ω)` Intrinsic base resistance.

33 `rci (Ω)` Intrinsic collector resistance.

34 `rbp (Ω)` Parasitic transistor base resistance.

35 `cje (F)` Intrinsic B-E capacitance.

36 `cjc (F)` Intrinsic B-C capacitance.

37 `cbex (F)` BX-E junction capacitance.

38 `cbcx (F)` B-CX junction capacitance.

39 `cbep (F)` Parasitic B-E junction capacitance.

40 `cbcp (F)` Parasitic B-C junction capacitance.

41 `pwr (W)` Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Linear Voltage Controlled Current Source (vccs)

Description

Positive current exits the source node and enters the sink node.

This device is supported within altergroups.

Sample Instance Statement

```
v1 (1 0 2 3) gm=-1 m=2
```

Instance Definition

```
Name  sink  src  ps  ns vccs parameter=value ...
```

Instance Parameters

1 `gm=0` STransconductance.

2 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Linear Voltage Controlled Voltage Source (vcvs)

Description

Current through the voltage source is calculated and is defined to be positive if it flows from the positive terminal, through the source, to the negative terminal.

This device is supported within altergroups.

Sample Instance Statement

```
e1 (out1 0 pos neg) vcvs gain=10
```

Instance Definition

```
Name  p  n  ps  ns vcvs parameter=value ...
```

Instance Parameters

1 `gain=0 V/V`Voltage gain.

2 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Independent Voltage Source (vsource)

Description

Current through the source is computed and is defined to be positive if it flows from the positive node, through the source, to the negative node.

This device is supported within altergroups.

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

V(T) = V(tnom) * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2]

Sample Instance Statement

```
vpulse1 (1 0) vsource type=pulse val0=0 val1=5 period=100n rise=10n fall=10n
width=40n
vpwl1 (1 0) vsource type=pwl wave=[1n 0 1.1n 2 1.5n 0.5 2n 3 5n 5] pwlperiod=5n
```

Instance Definition

```
Name  p  n vsource parameter=value ...
```

Instance Parameters

1 `dc=0` VDC value.

General Waveform Parameters

2 `type=dc`Waveform type.
Possible values are `dc`, `pulse`, `pwl`, `sine`, or `exp`.

3 `fundname`Name of the fundamental frequency. Must be specified if the source is active during a pdisto analysis or it is the active clock during an envlp analysis.

4 `delay=0 s`Waveform delay time.

Pulse Waveform Parameters

5 `val0=0 V`Zero value used in pulse and exponential waveforms.

6 `val1=1 V`One value used in pulse and exponential waveforms.

7 `period=∞ s`Period of waveform.

8 `rise (s)`Rise time for pulse waveform (time for transition from `val0` to `val1`).

9 `fall (s)`Fall time for pulse waveform (time for transition from `val1` to `val0`).

10 `width=∞ s`Pulse width (duration of `val1`).

PWL Waveform Parameters

11 `file`Name of file containing waveform.

12 `wave=[...]`Vector of time/value pairs that defines waveform.

13 `offset=0 V`DC offset for the PWL waveform.

14 `scale=1`Scale factor for the PWL waveform.

15 `stretch=1`Scale factor for time given for the PWL waveform.

16 `allbrkpts`All the points in the PWL waveform are breakpoints if set to yes. Default is yes if the number of points is less than 20.
Possible values are `no` or `yes`.

17 `pwlperiod (s)`Period of the periodic PWL waveform.

18 `twidth=pwlperiod/1000 s`
Transition width used when making PWL waveforms periodic.

Sinusoidal Waveform Parameters

19 `sinedc=dc V`DC level for sinusoidal waveforms.

20 `ampl=1 V`Peak amplitude of sinusoidal waveform.

21 `freq=0 Hz`Frequency of sinusoidal waveform.

22 `sinephase=0 °`Phase of sinusoid when `t=delay`.

23 `ampl2=1 V`Peak amplitude of second sinusoidal waveform.

24 `freq2=0 Hz`Frequency of second sinusoidal waveform.

25 `sinephase2=0 °`Phase of second sinusoid when `t=delay`.

26 `fundname2`Name of the fundamental frequency associated with freq2. Must be specified if freq2 is used in a pdisto analysis.

27 `fmmodindex=0`FM index of modulation for sinusoidal waveform.

28 `fmmodfreq=0 Hz`FM modulation frequency for sinusoidal waveform.

29 `ammodindex=0`AM index of modulation for sinusoidal waveform.

30 `ammodfreq=0 Hz`AM modulation frequency for sinusoidal waveform.

31 `ammodphase=0 °`AM phase of modulation for sinusoidal waveform.

32 `damp=0 1/s`Damping factor for sinusoidal waveform.

Exponential Waveform Parameters

33 `td1=0 s`Rise start time for exponential wave.

34 `tau1 (s)`Rise time constant for exponential wave.

35 `td2 (s)`Fall start time for exponential wave.

36 `tau2 (s)`Fall time constant for exponential wave.

Noise Parameters

37 `noisefile`Name of file containing excess spot noise data in the form of frequency-noise pairs.

38 `noisevec=[...]` V^2`/Hz`Excess spot noise as a function of frequency in the form of frequency-noise pairs.

Small Signal Parameters

39 `mag=0 V`Small signal voltage.

40 `phase=0 °`Small signal phase.

41 `xfmag=1 V/V`Transfer function analysis magnitude.

42 `pacmag=0 V`Periodic AC analysis magnitude.

43 `pacphase=0 °`Periodic AC analysis phase.

Multiplication Factor Parameters

44 `m=1`Multiplicity factor.

Temperature Effects Parameters

45 `tc1=0 1/C`First order temperature coefficient.

46 `tc2=0 C`$^{-2}$Second order temperature coefficient.

47 `tnom=27 C`Parameter measurement temperature. Default set by options.

If you do not specify the DC value, it is assumed to be the `time=0` value of the waveform.

In DC analyses, the only active parameters are `dc`, `m`, and the temperature coefficient parameters. In AC analyses, the only active parameters are `m`, `mag` and `phase`. In transient analyses, all parameters are active except the small signal parameters and the noise parameters. The `type` parameter selects which type of waveform is generated. You may specify parameters for more than one waveform type, and use the alter statement to change the waveform type between analyses.

A vector of time-value pairs describes the piecewise linear waveform. As an alternative, you can read the waveform from a file. In this case, you give time-value pairs one pair per line with a space or tab between the time and the value.

If you set `allbrkpts` to `yes`, you force the simulator to place time points at each point specified in a PWL waveform during a transient analysis. This can be very expensive for waveforms with many points. If you set `allbrkpts` to `no`, Spectre inspects the waveform, looking for abrupt changes, and forces time points only at those changes.

The PWL waveform is periodic if you specify `pwlperiod`. If the value of the waveform specified is not exactly the same at both its beginning its end, then you must provide a nonzero value `twidth`. Before repeating, the waveform changes linearly in an interval of `twidth` from its value at (`period` - `twidth`) to its value at the beginning of the waveform. Thus `twidth` must always be less than `period`.

You can give the excess noise of the source as a file or specify it with a vector of frequency-noise pairs. For a file, give the frequency-noise pairs one pair per line with a space or tab between the frequency and noise values.

Operating-Point Parameters

1 `v (V)`Voltage across the source.

2 `i (A)`Current through the source.

3 `pwr (W)`Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

Winding for Magnetic Core (winding)

Description

This winding is used in conjunction with magnetic cores to model coils and transformers with hysteresis. Each winding must be associated with a single core, though a core may have any number of windings.

Winding connects terminals `t1` and `b1`. Current through the winding is computed.

This device is not supported within altergroup.

Sample Instance Statement

```
c1 (1 0) core_mod area=1.2 len=8.1 id=0.45 id=0.55 gap=0.25
y1 (2 0) winding turn=5 core=c1 resis=1m
```

Instance Definition

```
Name  t  b winding parameter=value ...
```

Instance Parameters

1 `turn=1`Number of turns on winding.

2 `resis (Ω)`Resistance of the winding.

3 `m=1`Multiplicity factor.

4 `core`Name of core around which winding is wrapped.

Initial Conditions

5 `ic=0.0 A`Initial condition on the winding.

z-Domain Linear Current Controlled Current Source (zcccs)

Description

The output is defined with a transfer function given as the ratio of two polynomials in the complex variable z. Each polynomial can be specified using either its coefficients or its roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

You may specify polynomials either in the complex variable `z` or 1/z by setting optional parameter `polyarg` to `z` or `inversez` respectively. By default, it is set to `inversez`. If you choose to provide the coefficients of a polynomial, enter them as a vector in ascending order of the power of the variable z or 1/z, starting from the constant term. For example, to specify a denominator of 3z^-2 + 4z^-1 + 1, use `denom=[1 4 3]`. Or to specify a denominator of 4z^2 + 3z - 2, use `polyarg=z denom=[2 3 4]`.

To specify transfer function in terms of its zeros and poles in z-plane, give them as vectors of complex numbers. You must always give the real and imaginary portions of the root, even when the root is real. You may give either both roots of a complex-conjugate pair or only one. In the latter case the conjugate complex root will be generated automatically. The order of the roots is not important. For example, to specify poles of z = 1, z = 4j, z = -4j, z = 2 + 2j, and z = 2 - 2j, use `poles=[1 0 0 4 0 -4 2 2 2 -2]` or, omitting conjugate poles, `poles=[1 0 0 4 2 2 ]`.

Either the numerator or the denominator specification can be omitted. An omitted denominator or numerator is taken to be 1.

The parameter `gain` is interpreted either as the DC gain or, if the function has zeros or poles on the unit circle, as a constant factor.

Transition time (`tt`) is an optional parameter that at each sampling point forces linear transition of the output to a new value within the specified time range. By default, it is set to one percent of the sampling period.

The sampling delay (`td`) is another optional parameter, with the default value of 0, that lets you set asynchronous sampling rates.

To use the s to z transformation, set the optional `sxz` parameter to one of the transformation methods - forward differences, backward differences, or bilinear. When the `sxz` parameter is specified, the transfer function specification is assumed to be given in the complex variable s and it will be transformed to the complex variable z using the indicated method.

This device is not supported within altergroup.

Sample Instance Statement

```
va (1 0) vsource type=sine freq=10K
z2 (2 0) zcccs probe=va gain=1 ts=4.9e-5 tt=1e-5 polyarg=inservez
numer=[1 -1] denom=[1 0]
```

Instance Definition

```
Name  sink  src zcccs parameter=value ...
```

Instance Parameters

1 `probe`Device through which the controlling current flows.

2 `port=0`Index of the probe port through which the controlling current flows.

3 `ts=1 s`Sampling period.

4 `td=0 s`Sampling delay.

5 `tt=0.01 ts s`Transition time.

6 `gain=1`Transfer function gain.

7 `polyarg=inversez`Polynomial argument.
Possible values are `z` or `inversez`.

8 `sxz=none`s to z transformation.
Possible values are `none`, `backward`, `forward`, or `bilinear`.

9 `numer=[...]`Vector of numerator coefficients.

10 `denom=[...]`Vector of denominator coefficients.

11 `zeros=[...]`Vector of complex zeros.

12 `poles=[...]`Vector of complex poles.

13 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Input current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

denom I-10	numer I-9	probe I-1	ts I-3
gain I-6	poles I-12	pwr OP-3	tt I-5
i OP-1	polyarg I-7	sxz I-8	v OP-2
m I-13	port I-2	td I-4	zeros I-11

z-Domain Current Controlled Voltage Source (zccvs)

Description

The output is defined with a transfer function given as the ratio of two polynomials in the complex variable z. Each polynomial can be specified using either its coefficients or its roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

You may specify polynomials either in the complex variable z or 1/z by setting optional parameter `polyarg` to `z` or `inversez` respectively. By default, it is set to `inversez`. If you choose to provide the coefficients of a polynomial, enter them as a vector in ascending order of the power of the variable z or 1/z, starting from the constant term. For example, to specify a denominator of 3z^-2 + 4z^-1 + 1, use `denom=[1 4 3]`. Or to specify a denominator of 4z^2 + 3z - 2, use `polyarg=z denom=[2 3 4]`.

To specify transfer function in terms of its zeros and poles in z-plane, give them as vectors of complex numbers. You must always give the real and imaginary portions of the root, even when the root is real. You may give either both roots of a complex-conjugate pair or only one. In the latter case the conjugate complex root will be generated automatically. The order of the roots is not important. For example, to specify poles of z = 1, z = 4j, z = -4j, z = 2 + 2j, and z = 2 - 2j, use `poles=[1 0 0 4 0 -4 2 2 2 -2]` or, omitting conjugate poles, `poles=[1 0 0 4 2 2 ]`.

Either the numerator or the denominator specification can be omitted. An omitted denominator or numerator is taken to be 1.

The parameter `gain` is interpreted either as the DC gain or, if the function has zeros or poles on the unit circle, as a constant factor.

Transition time (`tt`) is an optional parameter that at each sampling point forces linear transition of the output to a new value within the specified time range. By default, it is set to one percent of the sampling period.

The sampling delay (`td`) is another optional parameter, with the default value of 0, that lets you set asynchronous sampling rates.

To use the s to z transformation, set the optional `sxz` parameter to one of the transformation methods - forward differences, backward differences, or bilinear. When the `sxz` parameter is specified, the transfer function specification is assumed to be given in the complex variable s and it will be transformed to the complex variable z using the indicated method.

This device is not supported within altergroup.

Sample Instance Statement

```
va (1 0) vsource type=sine freq=10K
z2 2 0 zccvs probe=va gain=-2 ts=5e-5 tt=1.1e-5 numer=[1 -1]
```

Instance Definition

```
Name  p  n zccvs parameter=value ...
```

Instance Parameters

1 `probe`Device through which the controlling current flows.

2 `port=0`Index of the probe port through which the controlling current flows.

3 `ts=1 s`Sampling period.

4 `td=0 s`Sampling delay.

5 `tt=0.01 ts s`Transition time.

6 `gain=1`Transfer function gain.

7 `polyarg=inversez`Polynomial argument.
Possible values are `z` or `inversez`.

8 `sxz=none`s to z transformation.
Possible values are `none`, `backward`, `forward`, or `bilinear`.

9 `numer=[...]`Vector of numerator coefficients.

10 `denom=[...]`Vector of denominator coefficients.

11 `zeros=[...]`Vector of complex zeros.

12 `poles=[...]`Vector of complex poles.

13 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)` Output voltage.

3 `pwr (W)` Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

denom I-10	numer I-9	probe I-1	ts I-3
gain I-6	poles I-12	pwr OP-3	tt I-5
i OP-1	polyarg I-7	sxz I-8	v OP-2
m I-13	port I-2	td I-4	zeros I-11

z-Domain Linear Voltage Controlled Current Source (zvccs)

Description

The output is defined with a transfer function given as the ratio of two polynomials in the complex variable z. Each polynomial can be specified using either its coefficients or its roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

You may specify polynomials either in the complex variable z or 1/z by setting optional parameter `polyarg` to `z` or `inversez` respectively. By default, it is set to `inversez`. If you choose to provide the coefficients of a polynomial, enter them as a vector in ascending order of the power of the variable z or 1/z, starting from the constant term. For example, to specify a denominator of 3z^-2 + 4z^-1 + 1, use `denom=[1 4 3]`. Or to specify a denominator of 4z^2 + 3z - 2, use `polyarg=z denom=[2 3 4]`.

To specify transfer function in terms of its zeros and poles in z-plane, give them as vectors of complex numbers. You must always give the real and imaginary portions of the root, even when the root is real. You may give either both roots of a complex-conjugate pair or only one. In the latter case the conjugate complex root will be generated automatically. The order of the roots is not important. For example, to specify poles of z = 1, z = 4j, z = -4j, z = 2 + 2j, and z

= 2 - 2j, use `poles=[1 0 0 4 0 -4 2 2 2 -2]` or, omitting conjugate poles, `poles=[1 0 0 4 2 2 ]`.

Either the numerator or the denominator specification can be omitted. An omitted denominator or numerator is taken to be 1.

The parameter `gain` is interpreted either as the DC gain or, if the function has zeros or poles on the unit circle, as a constant factor.

Transition time (`tt`) is an optional parameter that at each sampling point forces linear transition of the output to a new value within the specified time range. By default, it is set to one percent of the sampling period.

The sampling delay (`td`) is another optional parameter, with the default value of 0, that lets you set asynchronous sampling rates.

To use the s to z transformation, set the optional `sxz` parameter to one of the transformation methods - forward differences, backward differences, or bilinear. When the `sxz` parameter is specified, the transfer function specification is assumed to be given in the complex variable s and it will be transformed to the complex variable z using the indicated method.

This device is not supported within altergroup.

Sample Instance Statement

```
va (1 0) vsource type=sine freq=10K
z1 (2 0 1 0) zvccs gain=2 ts=4.5e-5 tt=1e-5 zeros=[-1 0] poles=[0 0]
```

Instance Definition

```
Name  sink  src  ps  ns zvccs parameter=value ...
```

Instance Parameters

1 `ts=1 s`Sampling period.

2 `td=0 s`Sampling delay.

3 `tt=0.01 ts s`Transition time.

4 `gain=1`Transfer function gain.

5 `polyarg=inversez`Polynomial argument.
Possible values are `z` or `inversez`.

6 sxz=nones to z transformation.
Possible values are none, backward, forward, or bilinear.

7 numer=[...]Vector of numerator coefficients.

8 denom=[...]Vector of denominator coefficients.

9 zeros=[...]Vector of complex zeros.

10 poles=[...]Vector of complex poles.

11 m=1Multiplicity factor.

Operating-Point Parameters

1 i (A)Output current.

2 v (V)Output voltage.

3 pwr (W)Power dissipation.

Parameter Index

In the following index, I refers to instance parameters, M refers to the model parameters section, O refers to the output parameters section, and OP refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of M-35 means the 35th model parameter.

denom I-8	numer I-7	sxz I-6	v OP-2
gain I-4	poles I-10	td I-2	zeros I-9
i OP-1	polyarg I-5	ts I-1	
m I-11	pwr OP-3	tt I-3	

z-Domain Voltage Controlled Voltage Source (zvcvs)

Description

The output is defined with a transfer function given as the ratio of two polynomials in the complex variable z. Each polynomial can be specified using either its coefficients or its roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

You may specify polynomials either in the complex variable z or 1/z by setting optional parameter `polyarg` to `z` or `inversez` respectively. By default, it is set to `inversez`. If you choose to provide the coefficients of a polynomial, enter them as a vector in ascending order of the power of the variable z or 1/z, starting from the constant term. For example, to specify a denominator of 3z^-2 + 4z^-1 + 1, use `denom=[1 4 3]`. Or to specify a denominator of 4z^2 + 3z - 2, use `polyarg=z denom=[2 3 4]`.

To specify transfer function in terms of its zeros and poles in z-plane, give them as vectors of complex numbers. You must always give the real and imaginary portions of the root, even when the root is real. You may give either both roots of a complex-conjugate pair or only one. In the latter case the conjugate complex root will be generated automatically. The order of the roots is not important. For example, to specify poles of z = 1, z = 4j, z = -4j, z = 2 + 2j, and z = 2 - 2j, use `poles=[1 0 0 4 0 -4 2 2 2 -2]` or, omitting conjugate poles, `poles=[1 0 0 4 2 2 ]`.

Either the numerator or the denominator specification can be omitted. An omitted denominator or numerator is taken to be 1.

The parameter `gain` is interpreted either as the DC gain or, if the function has zeros or poles on the unit circle, as a constant factor.

Transition time (`tt`) is an optional parameter that at each sampling point forces linear transition of the output to a new value within the specified time range. By default, it is set to one percent of the sampling period.

The sampling delay (`td`) is another optional parameter, with the default value of 0, that lets you set asynchronous sampling rates.

To use the `s` to `z` transformation, set the optional `sxz` parameter to one of the transformation methods - forward differences, backward differences, or bilinear. When the `sxz` parameter is specified, the transfer function specification is assumed to be given in the complex variable s and it will be transformed to the complex variable z using the indicated method.

This device is not supported within altergroup.

Sample Instance Statement

```
va (1 0) vsource type=sine freq=10K
z3 (3 0 1 0) zvcvs gain=-1 ts=4e-5 tt=1e-5 numer=[-1 -1]
```

Instance Definition

```
Name  p  n  ps  ns zvcvs parameter=value ...
```

Instance Parameters

1 `ts=1 s`Sampling period.

2 `td=0 s`Sampling delay.

3 `tt=0.01 ts s`Transition time.

4 `gain=1`Transfer function gain.

5 `polyarg=inversez`Polynomial argument.
Possible values are `z` or `inversez`.

6 `sxz=none`s to z transformation.
Possible values are `none`, `backward`, `forward`, or `bilinear`.

7 `numer=[...]`Vector of numerator coefficients.

8 `denom=[...]`Vector of denominator coefficients.

9 `zeros=[...]`Vector of complex zeros.

10 `poles=[...]`Vector of complex poles.

11 `m=1`Multiplicity factor.

Operating-Point Parameters

1 `i (A)`Output current.

2 `v (V)`Output voltage.

3 `pwr (W)`Power dissipation.

Parameter Index

In the following index, `I` refers to instance parameters, `M` refers to the model parameters section, `O` refers to the output parameters section, and `OP` refers to the operating point parameters section. The number indicates where to look in the appropriate section to find the description for that parameter. For example, a reference of `M-35` means the 35th model parameter.

5

Analysis Statements

This chapter discusses the following topics:

AC Analysis (ac)

Description

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

Spectre can perform the analysis while sweeping a parameter. The parameter can be frequency, temperature, component instance parameter, component model parameter, or netlist parameter. If changing a parameter affects the DC operating point, 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.

Definition

Name ac parameter=value ...

Parameters

1	`prevoppoint=no`	Use operating point computed on the previous analysis. Possible values are `no` or `yes`.

Sweep interval parameters

2	`start=0`	Start sweep limit.
3	`stop`	Stop sweep limit.
4	`center`	Center of sweep.
5	`span=0`	Sweep limit span.
6	`step`	Step size, linear sweep.
7	`lin=50`	Number of steps, linear sweep.
8	`dec`	Points per decade.

9 `log=50`	Number of steps, log sweep.
10 `values=[...]`	Array of sweep values.

Sweep variable parameters

11 `dev`	Device instance whose parameter value is to be swept.
12 `mod`	Model whose parameter value is to be swept.
13 `param`	Name of parameter to sweep.
14 `freq (Hz)`	Frequency when parameter other than frequency is being swept.

State-file parameters

15 `readns`	File that contains estimate of DC solution (nodeset).

Output parameters

16 `save`	Signals to output. Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.
17 `nestlvl`	Levels of subcircuits to output.
18 `oppoint=no`	Should operating point information be computed, and if so, where should it be sent. Possible values are no, `screen`, `logfile`, or `rawfile`.

Convergence parameters

19 `restart=yes`	Do not use previous DC solution as initial guess. Possible values are no or `yes`.

Annotation parameters

20 `annotate=sweep`	Degree of annotation. Possible values are no, `title`, `sweep`, `status`, or `steps`.

21 `stats=no` Analysis statistics.
Possible values are `no` or `yes`.

22 `title` Analysis title.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. However, if a previous analysis computed an operating point, you can set `prevoppoint=yes` to avoid recomputing it. For example, if you use this option when the previous analysis was a transient analysis, the operating point is the state of the circuit on the final time point.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

annotate 20	log 9	readns 15	step 6
center 4	mod 12	restart 19	stop 3
dec 8	nestlvl 17	save 16	title 22
dev 11	oppoint 18	span 5	values 10
freq 14	param 13	start 2	
lin 7	prevoppoint 1	stats 21	

Alter a Circuit, Component, or Netlist Parameter (alter)

Description

The `alter` statement changes the value of any modifiable component or netlist parameter for any analyses that follow. The parameter to be altered can be circuit temperature, a device instance parameter, a device model parameter, a netlist parameter, or a subcircuit parameter for a particular subcircuit instance. You can alter the circuit temperature by giving the parameter name as `param=temp` with no `dev`, `mod` or `sub` parameter. You can alter a top-level netlist parameter by giving the parameter name with no `dev`, `mod` or `sub` parameter. You can alter a subcircuit parameter for a particular subcircuit instance by specifying the subcircuit instance name with the `sub` parameter, and the subcircuit parameter name with the `param` parameter. Each `alter` statement can change only one parameter.

Definition

Name alter parameter=value ...

Parameters

1	`mod`	Device model.
2	`dev`	Device instance.
3	`sub`	Subcircuit instance.
4	`param`	Name of parameter to be altered.
5	`value`	New value for parameter.
6	`annotate`	Degree of annotation. Possible values are `no` or `title`.

Alter Group (altergroup)

Description

The `altergroup` statement changes the values of any modifiable model, instance or netlist parameter for any analyses that follow. Within an alter group, you can specify model statements, instance statements and parameter statements. These statements should be bound within braces. The opening brace is required at the end of the line defining the alter group. Alter groups cannot be nested or specified within subcircuits. The following statements are not allowed within altergroups (analyses, `export`, `ic`, `nodeset`, `paramset`, `save`, and `sens`).

Within an alter group, each device (instance or model) is first defaulted and then the device parameters are updated. For netlist parameters, the expressions are updated and evaluated.

For subckt within altergroup, all instances of the subckts are modified during the altergroup. There are strict checks that do not allow changes to topology.

You can include files into the alter group and can use the `simulator lang=spice` directive. See `spectre -h include` for more details. A model defined in the netlist, has to have the same model name and primitive type (`bsim2`, `bsim3`, `bjt`) in the alter group. An instance defined in the netlist, has to have the same instance name, terminal connections and primitive type. For model groups you can change the number of models in the group. There is a restriction that you cannot change from a model to a model group and vice versa. See `spectre -h bsim3v3` for details on model groups.

Definition

Name altergroup parameter=value ...

Parameters

1 `annotate`	Degree of annotation. Possible values are no or `title`.

Example:

```
FastCorner altergroup {
        parameters p2=1 p3=p1+2
        model myres resistor r1=1e3 af=1
        model mybsim bsim3v3 lmax=p1 lmin=3.5e-7
        m1 (n1 n2 n3 n4) mybsim w=0.3u l=1.2u
}
```

The list of public devices supported by `altergroup`:

tline	vsource	vcvs	vccs
vbic	tom2	resistor	phy_res
msline	mos3	mos2	mos1
jfet	isource	inductor	hvmos
hbt	gaas	ekv	diode
ccvs	cccs	capacitor	btasoi
bsim4	bsim3v3	bsim3	bsim2
bsim1	bjt		

The list of public devices not supported by `altergroup`:

cktrom	zvcvs	zvccs	zccvs
zcccs	winding	transformer	switch
svcvs	svccs	sccvs	scccs
relay	pvcvs	pvccs	port
pccvs	pcccs	nport	mtline
mos0	mutual_inductor	iprobe	delay
d2a	core	b3soipd	a2d

Check Parameter Values (check)

Description

The `check` analysis checks the values of component parameters to assure they are reasonable. This analysis reduces the cost of data entry errors. Various filters specify which parameters are checked. You can perform checks on input, output, or operating-point parameters. Use this analysis in conjunction with the `+param` command line argument, which specifies a file that contains component parameter soft limits.

Definition

Name check parameter=value ...

Parameters

1 `what=all`	What parameters should be checked. Possible values are `none`, `inst`, `models`, `input`, `output`, `all`, or `oppoint`.

DC Analysis (dc)

Description

The DC analysis finds the DC operating-point or DC transfer curves of the circuit. To generate transfer curves, specify a parameter and a sweep range. The swept parameter can be circuit temperature, a device instance parameter, a device model parameter, a netlist parameter, or a subcircuit parameter for a particular subcircuit instance. You can sweep the circuit temperature by giving the parameter name as `param=temp` with no `dev`, `mod` or `sub` parameter. You can sweep a top-level netlist parameter by giving the parameter name with no `dev`, `mod` or `sub` parameter. You can sweep a subcircuit parameter for a particular subcircuit instance by specifying the subcircuit instance name with the `sub` parameter, and the subcircuit parameter name with the `param` parameter. After the analysis has completed, the modified parameter returns to its original value.

Definition

Name dc parameter=value ...

Parameters

Sweep interval parameters

1 `start=0`	Start sweep limit.
2 `stop`	Stop sweep limit.
3 `center`	Center of sweep.
4 `span=0`	Sweep limit span.
5 `step`	Step size, linear sweep.
6 `lin=50`	Number of steps, linear sweep.
7 `dec`	Points per decade.
8 `log=50`	Number of steps, log sweep.
9 `values=[...]`	Array of sweep values.

Sweep variable parameters

10 `dev`	Device instance whose parameter value is to be swept.
11 `mod`	Model whose parameter value is to be swept.

12 `param` — Name of parameter to sweep.

State-file parameters

13 `force=none` — What should be used to force values for DC. Uses the values from the device and node ICs.
Possible values are `none`, `node`, `dev`, or `all`.

14 `readns` — File that contains estimate of DC solution (nodeset).

15 `readforce` — File that contains force values.

16 `write` — File to which solution at first step in sweep is written.

17 `writefinal` — File to which solution at last step in sweep is written.

Output parameters

18 `save` — Signals to output.
Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.

19 `nestlvl` — Levels of subcircuits to output.

20 `print=no` — Print node voltages.
Possible values are `no` or `yes`.

21 `oppoint=no` — Should operating point information be computed, and if so, where should it be sent.
Possible values are `no`, `screen`, `logfile`, or `rawfile`.

22 `check=yes` — Check operating point parameters against soft limits.
Possible values are `no` or `yes`.

Convergence parameters

23 `homotopy=all` — Method used when no convergence on initial attempt of DC analysis.
Possible values are `none`, `gmin`, `source`, `dptran`, `ptran`, `arclength`, or `all`.

24 `restart=yes` Do not use previous solution as initial guess.
Possible values are `no` or `yes`.

25 `maxiters=150` Maximum number of iterations.

26 `maxsteps=10000` Maximum number of steps used in homotopy method.

Annotation parameters

27 `annotate=sweep` Degree of annotation.
Possible values are `no`, `title`, `sweep`, `status`, or `steps`.

28 `title` Analysis title.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) and determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. If you specify the `oppoint` parameter, Spectre computes and outputs the linearized model for each nonlinear component.

Nodesets help find the DC or initial transient solution. You can supply them in the circuit description file with nodeset statements, or in a separate file using the `readns` parameter. When nodesets are given, Spectre computes an initial guess of the solution by performing a DC analysis while forcing the specified values onto nodes by using a voltage source in series with a resistor whose resistance is `rforce`. Spectre then removes these voltage sources and resistors and computes the true solution from this initial guess.

Nodesets have two important uses. First, if a circuit has two or more solutions, nodesets can bias the simulator towards computing the desired one. Second, they are a convergence aid. By estimating the solution of the largest possible number of nodes, you might be able to eliminate a convergence problem or dramatically speed convergence.

When you simulate the same circuit many times, we suggest that you use both the `write` and `readns` parameters and give the same file name to both parameters. The DC analysis then converges quickly even if the circuit has changed somewhat since the last simulation, and the nodeset file is automatically updated.

You may specify values to force for the DC analysis by setting the parameter `force`. The values used to force signals are specified by using the `force` file, the `ic` statement, or the `ic` parameter on the capacitors and inductors. The force parameter controls the interaction of various methods of setting the force values. The effects of individual settings are:

`force=none`: Any initial condition specifiers are ignored.

`force=node`: The `ic` statements are used, and the `ic` parameter on the capacitors and inductors are ignored.

`force=dev`: The `ic` parameters on the capacitors and inductors are used, and the `ic` statements are ignored.

`force=all`: Both the `ic` statements and the `ic` parameters are used, with the `ic` parameters overriding the `ic` statements.

If you specify a `force` file with the `readforce` parameter, force values read from the file are used, and any `ic` statements are ignored.

Once you specify the force conditions, Spectre computes the DC analysis with the specified nodes forced to the given value by using a voltage source in series with a resistor whose resistance is `rforce` (see `options`).

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

annotate 27	lin 6	param 12	start 1
center 3	log 8	print 20	step 5
check 22	maxiters 25	readforce 15	stop 2
dec 7	maxsteps 26	readns 14	title 28
dev 10	mod 11	restart 24	values 9
force 13	nestlvl 19	save 18	write 16
homotopy 23	oppoint 21	span 4	writefinal 17

DC Device Matching Analysis (dcmatch)

Description

The `dcmatch` analysis performs DC device mis-matching analysis for a given output. It computes the deviation in the DC operating point of the circuit caused by mismatch in the devices. Users need to specify mismatch parameters in their model cards for each device contributing to the deviation. The analysis uses the device mismatch models to construct equivalent mismatch current sources to all the devices that have mismatch modeled. These current sources will have zero mean and some variance. The variance of the current sources are computed according to mismatch models. It then computes the 3-sigma variance of dc voltages or curents at user specified outputs due to the mismatch current sources. The simulation results displays the devices rank ordered by their contributions to the outputs. In addition, for mosfet devices, it displays threshold voltage mismatch, current factor mismatch, gate voltage mismatch, and drain current mismatch. For bipolar devices, it displays base-emitter junction voltage mismatch. For resistors, it displays resistor mismatches.

The analysis replaces multiple simulation runs by circuit designers for accuracy versus size analysis. It automatically identifies the set of critical matched components during circuit design. For example, when there are matched pairs in the circuit, the contribution of two matched transistors will be equal in magnitude and opposite in sign. Typical usage are to simulate the output offset voltage of operational amplifiers, estimate the variation in bandgap voltages, and predict the accuracy of current steering DACS.

Definition

Name ... dcmatch parameter=value ...

Parameters

1 `mth`	Relative mismatch contribution threshold value.
2 `where=screen`	Where DC-Mismatch analysis results should be printed. Possible values are `screen`, `logfile`, `file`, or `rawfile`.
3 `file`	File name for results to be printed if where=file is used.

Probe parameters

4 `oprobe`	Compute mismatch at the output defined by this component.

Port parameters

5 `portv`	Voltage across this probe port is output of the analysis.
6 `porti`	Current through this probe port is output of the analysis.

Sweep interval parameters

7 `start=0`	Start sweep limit.
8 `stop`	Stop sweep limit.
9 `center`	Center of sweep.
10 `span=0`	Sweep limit span.
11 `step`	Step size, linear sweep.
12 `lin=50`	Number of steps, linear sweep.
13 `dec`	Points per decade.
14 `log=50`	Number of steps, log sweep.
15 `values=[...]`	Array of sweep values.

Sweep variable parameters

16 `dev`	Device instance whose parameter value is to be swept.
17 `mod`	Model whose parameter value is to be swept.
18 `param`	Name of parameter to sweep.

State-file parameters

19 `readns`	File that contains estimate of DC solution (nodeset).

Output parameters

20 `save`	Signals to output. Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.
21 `nestlvl`	Levels of subcircuits to output.

22 `oppoint=no` — Should operating point information be computed, and if so, where should it be sent.
Possible values are `no`, `screen`, `logfile`, or `rawfile`.

Convergence parameters

23 `prevoppoint=no` — Use operating point computed on the previous analysis.
Possible values are `no` or `yes`.

24 `restart=yes` — Do not use previous DC solution as initial guess.
Possible values are `no` or `yes`.

Annotation parameters

25 `annotate=sweep` — Degree of annotation.
Possible values are `no`, `title`, `sweep`, `status`, or `steps`.

26 `stats=no` — Analysis statistics.
Possible values are `no` or `yes`.

27 `title` — Analysis title.

The `dcmatch` analysis will find a dc operating point first. If the dc analysis fails, then the dcmatch analysis will fail also. The parameter `mth` is a threshold value relative to maximum contribution. Any device contribution less than (*`mth * maximum`*) will not be reported where *`maximum`* is the maximum contribution among all the devices of a given type.

Example

```
dcmm1 dcmatch mth=1e-3 oprobe=vd porti=1
dcmm2 dcmatch mth=1e-3 oprobe=r3 portv=1
dcmm3 n1 n2 dcmatch mth=1e-3 where=rawfile stats=yes
dcmm4 n3 0 dcmatch mth=1e-3 where=file file="%C:r.info.what"
sweep1 sweep dev=mp6 param=w start=80e-6 stop=90e-6 step=2e-6 {
dcmm5 dcmatch oprobe=vd mth=1e-3 where=rawfile }
dcmm6 n3 0 dcmatch mth=0.01 dev=x1.mp2 param=w start=15e-6 stop=20e-6 step=1e-6
dcmm7 n3 0 dcmatch mth=0.01 param=temp start=25 stop=100 step=25
```

Note: `porti` allows users to select a current associated with a specific device given in oprobe as an output. This device, however, has to have its terminal currents as network variables, i.e. the device has to be an inductor, a vsource, a switch, a tline, a controlled voltage source, an iprobe, or other type of device which has current solution. Further, for inductor, vsource, switch, controlled voltage source and iprobe, `porti` can only be set to one, since these devices are two terminal devices (one port); and for tline `porti` can be set to one or two, since it is a four terminal device (two ports).

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Envelope Following Analysis (envlp)

Description

This analysis computes the envelope response of a circuit. The user specifies the analysis `clockname`. The simulator automatically determines the clock period by looking through all the sources with the specified name. The envelope response is computed over the interval from `start` to `stop`. If the interval is not a multiple of the clock period, it is rounded off to the nearest multiple before the stop time. The initial condition is taken to be the DC steady-state solution if not otherwise given.

Envelope following analysis is most efficient for circuits where the modulation bandwidth is orders of magnitude lower than the clock frequency. This is typically the case, for example, in circuits where the clock is the only fast varying signal and other input signals have a spectrum whose frequency range is orders of magnitude lower than the clock frequency. For another example, the down conversion of two closely placed frequencies can also generate a slow-varying modulation envelope.

The analysis generates two types of output files, a voltage versus time (td) file, and an amplitude/phase versus time (fd) file for each of specified harmonic of the clock fundamental.

Definition

Name envlp parameter=value ...

Parameters

Envelope fundamental parameters

1 `clockname` Name of the clock fundamental.

2 `modulationbw (Hz)` Modulation bandwidth.

Simulation interval parameters

3 `stop (s)` Stop time.

4 `start=0 s` Start time.

5 `tstab=0 s` Initial stabilization time.

6 `outputstart=start s`
Output is saved only after this time is reached.

Time-step parameters

7 `maxstep (s)` Maximum time step for inner transient integration. Default derived from `errpreset`.

8 `envmaxstep (s)` Maximum outer envelope step size. Default derived from `errpreset`.

Initial-condition parameters

9 `ic=all` What should be used to set initial condition. Possible values are `dc`, `node`, `dev`, or `all`.

10 `skipdc=no` If yes, there will be no dc analysis for initial transient. Possible values are `no` or `yes`.

11 `readic` File that contains initial transient condition.

Convergence parameters

12 `readns`	File that contains estimate of initial DC solution.
13 `cmin=0 F`	Minimum capacitance from each node to ground.

State-file parameters

14 `write`	File to which initial transient solution is to be written.
15 `writefinal`	File to which final transient solution is to be written.
16 `swapfile`	Temporary file that holds the matrix information used by Newton's method. Tells Spectre to use a regular file rather than virtual memory to hold the matrix information. Use this option if Spectre complains about not having enough memory to complete this analysis.

Integration method parameters

17 `method`	Integration method. Default derived from `errpreset`. Possible values are `euler`, `trap`, `traponly`, `gear2`, or `gear2only`.

Accuracy parameters

18 `errpreset=moderate`	Selects a reasonable collection of parameter settings. Possible values are `conservative`, `moderate` or `liberal`.
19 `relref`	Reference used for the relative convergence criteria. Default derived from `errpreset`. Possible values are `pointlocal`, `alllocal`, `sigglobal`, or `allglobal`.
20 `lteratio`	Ratio used to compute LTE tolerances from Newton tolerance. Default derived from `errpreset`.
21 `steadyratio`	Ratio used to compute steady state tolerances from LTE tolerance. Default derived from `errpreset`.

22 `envlteratio`	Ratio used to compute envelope LTE tolerances. Default derived from `errpreset`.

Annotation parameters

23 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
24 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
25 `title`	Analysis title.

Output parameters

26 `harms=1`	Number of clock harmonics to output.
27 `harmsvec=[...]`	Array of desired clock harmonics. Alternate form of `harms` that allows selection of specific harmonics.
28 `outputtype=both`	Output type. Possible values are `both`, `envelope` or `spectrum`.
29 `save`	Signals to output. Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.
30 `nestlvl`	Levels of subcircuits to output.
31 `compression=no`	Do data compression on output. Possible values are `no` or `yes`.
32 `strobeperiod (s)`	The output strobe interval (in seconds of envelope following time). The actual strobe interval is rounded to an integer multiple of the clock period.

Newton parameters

33 `maxiters=5`	Maximum number of Newton iterations per transient integration time step.
34 `envmaxiters=3`	Maximum number of Newton iterations per envelope step.

35 `restart=yes` Do not use previous DC solution as initial guess. Possible values are `no` or `yes`.

Circuit age

36 `circuitage (Years)` Stress Time. Age of the circuit used to simulate hot-electron degradation of MOSFET and BSIM circuits.

The simulator examines all the sources whose name matches the clock name specified in the analysis line by the `clockname` parameter to determine the clock frequency. If more than one frequencies are found, the least common factor of these frequencies is used as the clock frequency.

The maximum envelope step size is affected by many parameters. It can be directly limited by `envmaxstep`. It is also limited by `modulationbw`. The user gives an estimate of the modulation bandwidth. The simulator will put at least eight points within the modulation period.

The `harms` and `harmsvec` parameters affect the simulation time in a significant way. The spectrum is calculated for all the specified harmonics for all sampled integration cycles as the envelope following analysis marches on. For each harmonic, a file is generated. The user should refrain from specifying unnecessary harmonics. Typically, `harms` is set to 1 or 2.

Most parameters of this analysis are inherited from either transient or PSS analysis and their meanings are consistent. However, a few of them need to be clarified. The effect of `errpreset` on some particular envelope following analysis parameters is shown in the following table.

Parameter defaults as a function of errpreset

errpreset	envmaxstep	steayratio	envlteratio
`liberal`	Interval/10	1.0	10.0
`moderate`	Interval/50	0.1	1.0
`conservative`	Interval/100	0.01	0.1

Its effect on parameters such as `reltol`, `relref`, `method`, `maxstep`, and `lteratio` is the same as defined for transient analysis, except for that the transient simulation interval is always a clock period.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Circuit Information (info)

Description

The circuit information analysis outputs several kinds of information about the circuit and its components. You can use various filters to specify what information is output. You can create a listing of model, instance, temperature-dependent, input, output, and operating point parameters. You can also generate a summary of the minimum and maximum parameter values (by using `extremes=yes` or `only`). Finally, you can request that Spectre provide a node-to-terminal map (by using `what=terminals`) or a terminal-to-node map (by using `what=nodes`).

The following are brief descriptions of the types of parameters you can request with the `info` statement:

Input parameters: Parameters that you specify in the netlist, such as the given length of a MOSFET or the saturation current of a bipolar transistor (use `what=inst`, `models`, `input`, or `all`)

Output parameters: Parameters that are computed by Spectre, such as temperature dependent parameters and the effective length of a MOSFET after scaling (use `what=output` or `all`)

Operating-point parameters: Parameters that depend on the actual solution computed (use `what=oppoint`)

Definition

Name info parameter=value ...

Parameters

1 `what=oppoint` What parameters should be printed.
Possible values are `none`, `inst`, `models`, `input`, `output`, `nodes`, `all`, `terminals`, `oppoint`, `captab`, or `parameters`.

2 `where=logfile` Where parameters should be printed.
Possible values are `nowhere`, `screen`, `file`, `logfile`, or `rawfile`.

3 `file="%C:r.info.what"`
File name when `where=file`.

4 `save` Signals to output.
Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.

5 `nestlvl` Levels of subcircuits to output.

6 `extremes=yes` Print minimum and maximum values.
Possible values are `no`, `yes` or `only`.

7 `title` Analysis title.

Captab parameters

8 `detail=node` How detailed should the capacitance table be.
Possible values are `node`, `nodetoground` or `nodetonode`.

9 `sort=name` How to sort the capacitance table.
Possible values are `name` or `value`.

10 `threshold=0 F` Threshold value for printing capacitances (ignore capacitances smaller than this value).

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Monte Carlo Analysis (montecarlo)

Description

The `montecarlo` analysis is a swept analysis with associated child analyses similar to the sweep analysis (see `spectre -h sweep`.) The Monte Carlo analysis refers to statistics blocks" where statistical distributions and correlations of netlist parameters are specified. (Detailed information on statistics blocks is given below.) For each iteration of the Monte Carlo analysis, new pseudo-random values are generated for the specified netlist parameters (according to their specified distributions) and the list of child analyses are then executed.

"export" statements are associated with the child analysis. These export statements allow scalar calculator expressions to be specified which can be used to measure circuit output or performance values (such as for example the slew-rate of an op-amp). For more details see "spectre -h export". During a Monte Carlo analysis, these export statement values will vary as the netlist parameters vary for each Monte Carlo iteration, and are stored in a scalar data file for post processing. By varying netlist parameters and evaluating export statements, the Monte Carlo analysis becomes a tool that allows you to examine and predict circuit performance variations, which affect yield.

The statistics blocks allow you to specify batch-to-batch (process) and per-instance (mismatch) variations for netlist parameters. These statistically-varying netlist parameters can be referenced by models or instances in the main netlist and may represent IC manufacturing process variation, or component variations for board-level designs for example. The following description gives a simplified example of the Monte Carlo analysis flow:

```
perform nominal run if requested
if any errors in nominal run then stop
foreach Monte Carlo iteration {
  if process variations specified then
    apply "process" variation to parameters
```

```
  if mismatch variations specified then
    foreach subcircuit instance {
       apply "mismatch" variation to parameters
    }
  foreach child analysis {
    run child analysis
      evaluate any export statements and
      store results in a scalar data file
  }
}
```

Definition

Name montecarlo parameter=value ...

Parameters

Analysis parameters

1 `numruns=100`	Number of Monte Carlo iterations to perform (not including nominal).
2 `seed`	Optional starting seed for random number generator.
3 `scalarfile`	Output file that will contain output scalar data.
4 `paramfile`	Output file that will contain output scalar data labels.

Saving Process Parameters

5 `saveprocessparams`

Whether or not to save scalar data for statistically varying process parameters which are subject to process variation. Possible values are no or `yes`.

6 `processscalarfile`

Output file that will contain process parameter scalar data.

7 `processparamfile`

Output file that will contain process parameter scalar data labels.

8 `saveprocessvec=[...]` Array of statistically varying process parameters (which are subject to process variation) to save as scalar data in `processscalarfile`.

9 `firstrun=1` Starting iteration number.

10 `variations=process`
Level of statistical variation to apply.
Possible values are `process`, `mismatch` or `all`.

Flags

11 `donominal=yes` Whether or not to perform nominal run.
Possible values are no or `yes`.

12 `appendsd=no` Whether or not to append scalar data.
Possible values are no or `yes`.

13 `savefamilyplots=no`
Whether or not to save data for family plots. If yes, this could require a lot of disk space.
Possible values are no or `yes`.

14 `saveprocessparams`
Whether or not to save scalar data for statistically varying process parameters which are subject to process variation.
Possible values are no or `yes`.

Annotation parameters

15 `annotate=sweep` Degree of annotation.
Possible values are no, `title`, `sweep`, or `status`.

16 `title` Analysis title.

Detailed Description and Examples

`numruns`:(default=100)

The number of Monte Carlo iterations to perform. The simulator will perform a loop, running the specified child analyses and evaluating any export statements numruns times.

`seed`:(no default)

seed for the random number generator. By always specifying the same seed, you can reproduce a previous experiment. If you do not specify a seed, then each time you run the analysis, you will get different results i.e. a different stream of pseudo-random numbers will be generated.

`scalarfile`="filename"

This parameter allows an ASCII file to be specified in which scalar data (results of export expressions that resolve to scalar values) will be written. The data from this file can be read and plotted in histograms by Artist. For each iteration of each Monte Carlo child analyses, Spectre (through Artil) will write a line to this ASCII file which contains scalar data (one scalar expression per column e.g. slewrate or bandwidth.) The default name for this file will be of the form name.mcdata, where name is the name of the Monte Carlo analysis instance. This file contains only the matrix of numeric values. Artist Monte Carlo users will be more familiar with the term "mcdata" file for the scalar file. Additionally, when the Analog Artist Monte Carlo tool is used to generate the spectre netlist file, Spectre will merge the values of the statistically varying process parameters into this file containing the scalar data (results of export expressions). This means that Analog Artist can later read the data, and create scatterplots of the statistically varying process parameters against each other, or against the results of the export expressions. In this way, the user can see correlations between process parameter variations and circuit performances variations. This data merging will take place whenever the scalarfile and processscalarfile (see below) are written in the same directory.

paramfile="filename"

This file contains the titles, sweep variable values and the full expression for each of the columns in the scalarfile. Artist Monte Carlo users will be more familiar with the term "mcparam" file for the paramfile.This file will be created in the psf directory by default, unless you specify some path information in the filename.

`processscalarfile`="filename"

If saveprocessparams is set to yes, then the process (batch-to-batch) values of all statistically varying parameters are saved to this scalar data file. You can use the saveprocessvec to filter out a subset of parameters in which case Spectre will save only the parameters specified in saveprocessvec to the processscalarfile.) The processscalarfile is equivalent to the scalarfile, except that the data in the scalarfile contains the values of the scalar export statements, whereas the data in the processscalarfile contains the corresponding process parameter values. The default name for this file will be of the form instname.process.mcdata, where instname is the name of the Monte Carlo analysis instance. This file will be created in the psf directory by default, unless you specify some path information in the filename. You can load the processscalar file and processparamfile into the Artist statistical postprocessing environment to plot/verify the process parameter distributions. If you later merge the processparamfile with the data in the scalarfile, you can then plot export scalar values against the corresponding process parameters by loading this merged file into the Artist statistical postprocessing environment.

`processparamfile`="filename"

This file contains the titles, sweep variable values for each of the columns in the processscalarfile. These titles will be the names of the process parameters.

The processparamfile is equivalent to the paramfile, except that the paramfile contains the name of the export expressions, whereas the processparamfile contains the names of the process parameters. The default name for this file will be of the form instname.process.mcparam, where instname is the name of the Monte Carlo analysis instance. This file will be created in the psf directory by default, unless you specify some path information in the filename.

`firstrun`:(default=1)

index of first iteration. If the first iteration is specified as some number n greater than one, then the beginning n-1 iterations are "skipped" i.e. the Monte Carlo analysis behaves as if the first n-1 iterations were run, but without actually performing the child analyses for these iterations. The subsequent stream of random numbers generated for the remaining iterations will be the same as if the first n-1 iterations were actually run. By specifying the first iteration number and the same value for seed, you can reproduce a particular run or sequence of runs from a previous experiment (for example to examine an outlier case in more detail.)

`variations`={`process`,`mismatch`,$_{\text{all}}$} (defaults to process).

Whether to apply process (batch-to-batch) variations only, or mismatch (per-instance) variations only, or both together. This assumes that you have specified appropriate statistical distributions in the statistics block. You cannot request that mismatch variations be applied unless you have specified mismatch statistics in the statistics block. You cannot request that process variations be applied unless you have specified process statistics in the statistics block. More details on statistics blocks are given below.

`saveprocessvec`=[rshsp TOX ...]

If saveprocessparams is specified as yes, then save the process (batch-to-batch) values of only those parameters listed in saveprocessvec in the processparamfile. This acts as a filter so that you do not save all process parameters to the file. If you do not want to filter the list of process parameters, then do not specify this parameter.

`donominal`={yes,no}(defaults to yes).

This parameter controls whether or not Spectre should perform a nominal run before starting the main Monte Carlo loop of iterations. If any errors are encountered during the nominal run (e.g. convergence problems, incorrect export expressions, etc.) then Spectre will issue an appropriate error message and immediately abandon the Monte Carlo analysis.

If donominal is set to "no", then Spectre will run the Monte Carlo iterations only, and will not perform a nominal analysis. If any errors are encountered during the Monte Carlo iterations, Spectre will issue a warning and continue with the next iteration of the Monte Carlo loop.

`appendsd`={yes,no}(defaults to no).

Specifies whether to append scalar data to an existing scalarfile, or to overwrite the existing scalarfile. This flag applies to both the scalar file and the processscalarfile.

`savefamilyplots`={yes,no}.

If "yes", a data file (e.g. psf) is saved for each analysis for each Monte Carlo iteration, in addition to the export scalar results which are saved to the ASCII scalar data file at the end of each iteration. Saving the full data files between runs enables the cloud plotting feature (overlaid waveforms) in Artist. It also enables the user to define/evaluate new calculator measurements after the simulation has been run using the Artist calculator. This feature could result in a huge amount of data being stored to disk, and it is advised that you use this feature with care. If you do decide to use this feature, it is advisable to keep the number of saved quantities to a minimum. If this parameter is set to "no", then data files are overwritten by each Monte Carlo iteration.

`annotate`={no,`title`,`sweep`,`status`}

Degree of annotation. Use the maximum value of "status" to print a summary of which runs did not converge or had problems evaluating "export" statements, etc.

Examples

```
// do a Monte Carlo analysis, with process variations only
// useful for looking at absolute performance spreads
mc1 montecarlo variations=process seed=1234 numruns=200 {
  dcop1 dc      // a child analysis
  tran1 tran start=0 stop=1u    // another child analysis
  // export calculations are sent to the scalardata file
  export slewrate=oceanEval("slewRate(v("vout"),10n,t,30n,t,10,90 )")
}
// do a Monte Carlo analysis, with mismatch variations only
// useful for detecting spreads in differential circuit
// applications, etc. Do not perform a nominal run.
```

```
mc2 montecarlo donominal=no variations=mismatch seed=1234 numruns=200 {
  dcop2 dc
  tran2 tran start=0 stop=1u
  export slewrate=oceanEval("slewRate(v("vout"),10n,t,30n,t,10,90 )")
}
// do both together...
mc3 montecarlo saveprocessparams=yes variations=all numruns=200 {
  dcop3 dc
  tran3 tran start=0 stop=1u
  export slewrate=oceanEval("slewRate(v("vout"),10n,t,30n,t,10,90 )")
}
```

Specifying Parameter Distributions using Statistics Blocks

The "statistics blocks" are used to specify the input statistical variations for a Monte Carlo analysis. A statistics block may contain one or more "process" blocks (which represents batch-to-batch type variations), and/or one or more "mismatch" blocks (which represents on-chip or device mismatch variations), in which the distributions for parameters are specified. Statistics blocks may also contain one or more correlation statements to specify the correlations between specified process parameters, and/or to specify correlated device instances (for example matched pairs). Statistics blocks may also contain a "truncate" statement which may be used for generating truncated distributions. The distributions specified in the process block will be sampled once per Monte Carlo iteration, and are typically used to represent batch-to-batch, or process variations, whereas the distributions specified in the mismatch block are sampled on a per subcircuit instance basis and are typically used to represent device-to-device mismatch for devices on the same chip. In the case where the same parameter is subject to both process and mismatch variations, then the sampled process value becomes the mean for the mismatch random number generator for that particular parameter.

Note: Multiple statistics blocks may exist, in which case they accumulate or overlay. Typically, process variations, mismatch variations and correlations between process parameters will be specified in one statistics block. A second statistics block would be specified where actual device instance correlations are specified (i.e. specification of matched pairs).

Statistics blocks can be specified using combinations of the Spectre keywords statistics, process, mismatch, vary, truncate and correlate. Braces {} are used to delimit blocks.

The following example shows some sample statistics blocks, which are discussed below along with syntax requirements.

```
// define some netlist parameters to represent process parameters
// such as sheet resistance and mismatch factors
parameters rshsp=200 rshpi=5k rshpi_std=0.4K xisn=1 xisp=1 xxx=20000 uuu=200
```

```
// define statistical variations, to be used
// with a MonteCarlo analysis.
statistics {
    process {   // process: generate random number once per MC run
        vary rshsp dist=gauss std=12 percent=yes
        vary rshpi dist=gauss std=rshpi_std // rshpi_std is a parameter
        vary xxx dist=lnorm std=12
        vary uuu dist=unif N=10 percent=yes
        ...
    }
    mismatch {  // mismatch: generate a random number per instance
        vary rshsp dist=gauss std=2
        vary xisn dist=gauss std=0.5
        vary xisp dist=gauss std=0.5
    }
    // some process parameters are correlated
    correlate param=[rshsp rshpi] cc=0.6
    // specify a global distribution truncation factor
    truncate tr=6.0     // +/- 6 sigma
}
//  a separate statistics block to specify correlated (i.e. matched) components
// where m1 and m2 are subckt instances.
statistics {
    correlate dev=[m1 m2] param=[xisn xisp] cc=0.8
}
```

Specifying Distributions

Parameter variations are specified using the following syntax:

`vary PAR_NAME dist=<type> {std=<value> | N=<value>} {percent=yes|no}`

Three types of parameter distributions are available: gaussian, lognormal and uniform, corresponding to the <*type*> keywords `gauss`, `lnorm` and`unif` respectively. For both the `gauss` and the `lnorm` distributions, you specify a standard deviation using the `std` keyword.

Gaussian Distribution

For the gaussian distribution, the mean value is taken as the current value of the parameter being varied, giving a distribution denoted by Normal(mean,std). Using the example above, parameter rshpi is varied with a distribution of Normal(5k,0.4k)

Lognormal Distribution

The lognormal distribution is denoted by

log(x) = Normal(log(mean), std)

where x is the parameter being specified as having a lognormal distribution.

(NOTE: log() is the natural logarithm function.) For parameter xxx in the example above, the process variation is according to

log(xxx) = Normal(log(20000), 12)

Uniform Distribution

The uniform distribution for parameter x is generated according to

x = unif(mean-N, mean+N)

such that the mean value is the nominal value of the parameter x, and the parameter is varied about the mean with a range of +/- N. The standard deviation is not specified for the uniform distribution, but its value can be calculated from the formula: std=N/sqrt(3).

Values as percentages

The "percent" flag indicates whether the standard deviation std or uniform range N are specified in absolute terms (percent=no) or as a percentage of the mean value (percent=yes). For parameter uuu in the example above, the mean value is 200, and the variation is 200 +/- 10%*(200) i.e. 200 +/- 20. For parameter rshsp, the process variation is given by Normal(200, 12%*(200)) i.e. Normal(200, 24). It is not advised that you use the percent=yes with the lognormal distribution.

Process and Mismatch Variations

The statistics specified in a process block are applied at global scope, and the distributions are sampled once per Monte Carlo iteration. The statistics specified in a mismatch block are applied on a per-subcircuit instance basis, and are sampled once per subcircuit instance. If you place model cards and/or device instances in subcircuits, and add a mismatch block to your statistics block you can effectively model device-to-device mismatch for these devices/ models.

Correlation Statements

There are two types of correlation statements that you can use: process parameter correlation statements, and instance correlation statements.

Process Parameter Correlation

The syntax of the process parameter correlation statement is:

correlate param=[list of parameters] cc=<value>

This allows you to specify a correlation coefficient between multiple process parameters. You can specify multiple process parameter correlation statements in a statistics block, to build a matrix of process parameter correlations. During a Monte Carlo analysis, process parameter values will be randomly generated according to the specified distributions and correlations.

Mismatch Correlation (Matched Devices)

The syntax of the instance or mismatch correlation statement is:

correlate dev=[list of subcircuit instances] {param=[list of parameters]} cc=<value>

where the device or subcircuit instances to be matched are listed in the list of subcircuit instances, and the list of parameters specifies exactly which parameters with mismatch variations are to be correlated.

The instance mismatch correlation statement is used to specify correlations for particular subcircuit instances. If a subcircuit contains a device, you can effectively use the instance correlation statements to specify that certain devices are correlated (i.e. matched) and give the correlation coefficient. You can optionally specify exactly which parameters are to be correlated by giving a list of parameters (each of which must have had distributions specified for it in a mismatch block), or specify no parameter list, in which case all parameters with mismatch statistics specified are correlated with the given correlation coefficient. The correlation coefficients are specified in the <value> field and must be between +/- 1.0, not including 1.0 or -1.0.

Note: correlation coefficients can be constants or expressions, as can "std" and "N" when specifying distributions.

Truncation Factor

The default truncation factor for gaussian distributions (and for the gaussian distribution underlying the lognormal distribution) is 4.0 sigma. Randomly generated values which are outside the range of mean +/- 4.0 sigma are automatically rejected and regenerated until they fall inside the range. You can change the truncation factor using the "truncate" statement. The syntax is:

```
truncate tr=<value>
```

Note: The value of the truncation factor can be a constant or an expression.

Note: Parameter correlations can be affected by using small truncation factors.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

annotate 15	numruns 1	savefamilyplots 13	scalarfile 3
appendsd 12	paramfile 4	saveprocessparams 5	seed 2
donominal 11	processparamfile 7	saveprocessparams 14	title 16
firstrun 9	processscalarfile 6	saveprocessvec 8	variations 10

Noise Analysis (noise)

Description

The `noise` analysis linearizes the circuit about the operating point and computes the noise spectral density at the output. If you identify an input source, the transfer function and the input-referred noise for an equivalent noise-free network is computed. In addition, if the input source is noisy, then the noise figure is computed.

The noise is computed at the output of the circuit. The output is specified with either a pair of nodes or a probe component. To specify the output of a circuit with a probe, specify it with the `oprobe` parameter. If the output is voltage (or potential), choose a `resistor` or a `port` as the output probe. If the output is current (or flow), choose a `vsource` or `iprobe` as the output probe.

If the input-referred noise is desired, specify the input source using the `iprobe` parameter. Currently, only a `vsource`, an `isource`, or a `port` may be used as an input probe. If the input source is noisy, as is a `port`, the noise analysis will compute the noise factor (F) and noise figure (NF). To match the IEEE definition of noise figure, the input probe must be a port with no excess noise and its `noisetemp` must be set to 16.85C (290K). In addition, the output load must be a `resistor` or `port` and must be identified as the `oprobe`.

The noise analysis always computes the total noise at the output, which includes contributions from the input source and the output load. The amount of the output noise that is attributable to each noise source in the circuit is also computed and output individually. If the input source is identified, the input-referred noise is computed, which includes the noise from the input source itself. Finally, if the input source is identified and is noisy, the noise factor and noise figure are computed. Thus if

No = total output noise

Ns = noise at the output due to the input probe (the source)

Nl = noise at the output due to the output probe (the load)

IRN = input referred noise

G = gain of the circuit

F = noise factor

NF = noise figure

then,

IRN = sqrt(No^2 / G^2)

F = (No^2 - Nl^2)/Ns^2

NF = 10*log10(F)

When the results are output, No is named `out`, IRN is named `in`, G is named `gain`, F is named F, and NF is named `NF`.

Spectre can perform the analysis while sweeping a parameter. The parameter can be frequency, temperature, component instance parameter, component model parameter, or netlist parameter. If changing a parameter affects the DC operating point, 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.

Definition

Name [p] [n] noise parameter=value ...

The optional terminals (p and n) specify the output of the circuit. If you do not give the terminals, then you must specify the output with a probe component.

Parameters

1 `prevoppoint=no`	Use operating point computed on the previous analysis. Possible values are `no` or `yes`.

Sweep interval parameters

2 `start=0`	Start sweep limit.
3 `stop`	Stop sweep limit.
4 `center`	Center of sweep.
5 `span=0`	Sweep limit span.
6 `step`	Step size, linear sweep.
7 `lin=50`	Number of steps, linear sweep.
8 `dec`	Points per decade.
9 `log=50`	Number of steps, log sweep.
10 `values=[...]`	Array of sweep values.

Sweep variable parameters

11 `dev`	Device instance whose parameter value is to be swept.
12 `mod`	Model whose parameter value is to be swept.
13 `param`	Name of parameter to sweep.
14 `freq (Hz)`	Frequency when parameter other than frequency is being swept.

Probe parameters

15 `oprobe`	Compute total noise at the output defined by this component.
16 `iprobe`	Input probe. Refer the output noise to this component.

State-file parameters

17 `readns`	File that contains estimate of DC solution (nodeset).

Output parameters

18 `save`	Signals to output. Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.
19 `nestlvl`	Levels of subcircuits to output.
20 `oppoint=no`	Should operating point information be computed, and if so, where should it be sent. Possible values are `no`, `screen`, `logfile`, or `rawfile`.

Convergence parameters

21 `restart=yes`	Do not use previous DC solution as initial guess. Possible values are `no` or `yes`.

Annotation parameters

22 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
23 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
24 `title`	Analysis title.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. However, if a previous analysis computed an operating point, you can set `prevoppoint=yes` to avoid recomputing it. For example, if you use this option when the previous analysis was a transient analysis, the operating point is the state of the circuit on the final time point.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

annotate 22	lin 7	param 13	start 2
center 4	log 9	prevoppoint 1	stats 23
dec 8	mod 12	readns 17	step 6
dev 11	nestlvl 19	restart 21	stop 3
freq 14	oppoint 20	save 18	title 24
iprobe 16	oprobe 15	span 5	values 10

Immediate Set Options (options)

Description

The immediate set options statement sets or changes various program control options. These options take effect immediately and are set while the circuit is read. For further options, see the individual analyses.

Note: Options that are dependent on netlist parameter values, do not maintain their dependencies on those netlist parameters.

Definition

Name options parameter=value ...

Parameters

Tolerance parameters

1 `reltol=0.001`	Relative convergence criterion.
2 `vabstol=1e-06 V`	Voltage absolute tolerance convergence criterion.
3 `iabstol=1e-12 A`	Current absolute tolerance convergence criterion.

Temperature parameters

4 `temp=27 C`	Temperature.
5 `tnom=27 C`	Default component parameter measurement temperature.
6 `tempeffects=all`	Temperature effect selector. Possible values are `vt`, `tc` or `all`.

Output parameters

7 `save=selected`	Signals to output. Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.
8 `nestlvl=∞`	Levels of subcircuits to output.
9 `subcktprobelvl=0`	Level up to which subcircuit terminal current probes are to be set up.
10 `currents=selected`	Terminal currents to output. (See important note below about saving currents by using probes). Possible values are `all`, `nonlinear` or `selected`.

11 `useprobes=no` Use current probes when measuring terminal currents. (See important note below about saving currents by using probes).
Possible values are no or `yes`.

12 `redundant_currents=no`
If yes, save both currents through two terminal devices.
Possible values are no or `yes`.

13 `pwr=none` Power signals to create.
Possible values are `all`, `subckts`, `devices`, `total`, or none.

14 `saveahdlvars=selected`
AHDL variables to output.
Possible values are `all` or `selected`.

15 `rawfmt=psfbin` Output raw data file format.
Possible values are `nutbin`, `nutascii`, `wsfbin`, `wsfascii`, `psfbin`, `psfascii`, awb, or `sst2`.

16 `rawfile="%C:r.raw"` Output raw data file name.

Convergence parameters

17 `homotopy=all` Method used when no convergence on initial attempt of DC analysis.
Possible values are none, `gmin`, `source`, `dptran`, `ptran`, `arclength`, or `all`.

18 `limit=dev` Limiting algorithms to aid DC convergence.
Possible values are `delta`, `log` or `dev`.

Component parameters

19 `scalem=1` Model scaling factor.

20 `scale=1` Device instance scaling factor.

21 `compatible=spectre`
Encourage device equations to be compatible with a foreign simulator. This option does not affect input syntax.
Possible values are `spectre`, `spice2`, `spice3`, `cdsspice`, `hspice`, or `spiceplus`.

22 `approx=no` Use approximate models. Difference between approximate and exact models is generally very small.
Possible values are no or `yes`.

23 `macromodels=no` Circuit contains macromodels; sometimes helps performance.
Possible values are no or `yes`.

Error-checking parameters

24 `topcheck=full` Check circuit topology for errors.
Possible values are no, `min`, `full`, or `fixall`.

25 `ignshorts=no` Silently ignore shorted components.
Possible values are no or `yes`.

26 `diagnose=no` Print additional information that might help diagnose accuracy and convergence problems.
Possible values are no or `yes`.

27 `opptcheck=yes` Check operating point parameters against soft limits.
Possible values are no or `yes`.

Resistance parameters

28 `gmin=1e-12 S` Minimum conductance across each nonlinear device.

29 `gmin_check=max_v_only`
Specifies that effect of `gmin` should be reported if significant.
Possible values are no, `max_v_only`, `max_only`, or `all`.

30 `rforce=1 Ω` Resistance used when forcing nodesets and node-based initial conditions.

Quantity parameters

31 `value="V"` Default value quantity.

32 `flow="I"` Default flow quantity.

33 `quantities=no` Print quantities.
Possible values are no, `min` or `full`.

Annotation parameters

34 `audit=detailed`	Print time required by various parts of the simulator. Possible values are `no`, `brief`, `detailed`, or `full`.
35 `inventory=detailed`	Print summary of components used. Possible values are `no`, `brief` or `detailed`.
36 `narrate=yes`	Narrate the simulation. Possible values are `no` or `yes`.
37 `debug=no`	Give debugging messages. Possible values are `no` or `yes`.
38 `info=yes`	Give informational messages. Possible values are `no` or `yes`.
39 `note=yes`	Give notice messages. Possible values are `no` or `yes`.
40 `maxnotes=5`	Maximum number of times any notice will be issued per analysis.
41 `warn=yes`	Give warning messages. Possible values are `no` or `yes`.
42 `maxwarns=5`	Maximum number of times any warning message will be issued per analysis.
43 `error=yes`	Give error messages. Possible values are `no` or `yes`.
44 `digits=5`	Number of digits used when printing numbers.
45 `notation=eng`	When printing real numbers to the screen, what notation should be used. Possible values are `eng`, `sci` or `float`.
46 `cols=80`	Width of screen in characters.
47 `title`	Circuit title.

Matrix parameters

48 `pivotdc=no` Use numeric pivoting on every iteration of DC analysis. Possible values are `no` or `yes`.

49 `pivrel=0.001` Relative pivot threshold.

50 `pivabs=0` Absolute pivot threshold.

Miscellaneous parameters

51 `ckptclock=1800 s` Clock time checkpoint period.

Sensitivity parameters

52 `sensfile` Output sensitivity data file name.

53 `sensformat=tabular`
Format of sensitivity data.
Possible values are `tabular` or `list`.

54 `senstype=partial` Type of sensitivity being calculated.
Possible values are `partial` or `normalized`.

Important

Important note about `currents` and `useprobes` options

Adding probes to circuits that are sensitive to numerical noise might affect the solution. In such cases accurate solution may be obtained by reducing `reltol`.

The following devices will always use probes to save currents (even with `useprobes=no`): `port`, `delay`, `switch`, `hbt`, `transformer`, `core`, `winding`, `fourier`, `d2a`, `a2d`, `a2ao`, `a2ai`.

Sensitivity Definitions

When `senstype` is set to `partial`, the sensitivity being calculated is the partial derivative of a differentiable output variable *F* with respect to a design parameter *p*

dF

$$D\,(F \text{ w.r.t. } p) = \frac{}{dp}$$

This definition is not scale free. When `senstype` is set to `normalized`, the sensitivity being calculated is the normalized sensitivity

$$S\,(F \text{ w.r.t. } p) = \frac{d \ln F}{d \ln p} = \frac{p}{F}\frac{dF}{dp} = \frac{p}{F}\,D\,(F \text{ w.r.t. } p)$$

When either F or p takes a zero value, the above normalized definition no longer provides a useful measure, the following two seminormalized sensitivities are used instead:

$$S\,(F \text{ w.r.t. } p) = \frac{dF}{d \ln p} = p\,\frac{dF}{dp} = p\,D\,(F \text{ w.r.t. } p) \quad \text{if } F = 0$$

and

$$S\,(F \text{ w.r.t. } p) = \frac{d \ln F}{dp} = \frac{1}{F}\frac{dF}{dp} = \frac{1}{F}\,D\,(F \text{ w.r.t. } p) \quad \text{if } p = 0$$

When both *F* and *p* are zero, the partial sensitivity is used.

Topcheck Parameter

When `topcheck=full`, a topology check is performed and gmin is inserted between isolated nodes and ground. A heuristic topology check is also performed to find nodes that may be isolated due to the numerical nature of the circuit. For example, nodes isolated by reverse biased diodes in MOSFETS.

Use `topcheck=fixall` to attach gmin to all types of isolated nodes. Including the ones found by the heuristic topology check.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Periodic AC Analysis (pac)

Description

The periodic AC (PAC) analysis is used to compute transfer functions for circuits that exhibit frequency translation. Such circuits include mixers, switched-capacitor filters, samplers, phase-locked loops, and the like. It is a small-signal analysis like AC analysis, except the circuit is first linearized about a periodically varying operating point as opposed to a simple DC operating point. Linearizing about a periodically time-varying operating point allows transfer-functions that include frequency translation, whereas simply linearizing about a DC operating point could not because linear time-invariant circuits do not exhibit frequency translation. Also, the frequency of the sinusoidal stimulus is not constrained by the period of the large periodic solution.

Computing the small-signal response of a periodically varying circuit is a two step process. First, the small stimulus is ignored and the periodic steady-state response of the circuit to possibly large periodic stimulus is computed using PSS analysis. As a normal part of the PSS analysis, the periodically time-varying representation of the circuit is computed and saved for later use. The second step is applying the small stimulus to the periodically varying linear representation to compute the small signal response. This is done using the PAC analysis. A

PAC analysis cannot be used alone, it must follow a PSS analysis. However, any number of periodic small-signal analyses such as PAC, PSP, PXF, PNoise, can follow a PSS analysis.

Unlike other analyses in Spectre, this analysis can only sweep frequency.

Definition

Name pac parameter=value ...

Parameters

Sweep interval parameters

1 `start=0`	Start sweep limit.
2 `stop`	Stop sweep limit.
3 `center`	Center of sweep.
4 `span=0`	Sweep limit span.
5 `step`	Step size, linear sweep.
6 `lin=50`	Number of steps, linear sweep.
7 `dec`	Points per decade.
8 `log=50`	Number of steps, log sweep.
9 `values=[...]`	Array of sweep values.
10 `sweeptype`	Specifies if the sweep frequency range is absolute frequency of input or if it is relative to the port harmonics. Possible values are `absolute` or `relative`.
11 `relharmnum=1`	Harmonic to which relative frequency sweep should be referenced.

Output parameters

12 `sidebands=[...]` Array of relevant sidebands for the analysis.

13 `maxsideband=0` An alternative to the `sidebands` array specification, which automatically generates the array: [-maxsideband ... 0 ... +maxsideband].

14 `freqaxis` Specifies whether the results should be output versus the input frequency, the output frequency, or the absolute value of the output frequency. Default is `in` for logarithmic frequency sweeps and `absout` otherwise.
Possible values are `absout`, `out` or `in`.

15 `save` Signals to output.
Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.

16 `nestlvl` Levels of subcircuits to output.

17 `outputperiod=0.0 (no output)`
Time-domain output period. The time-domain small-signal response is computed for the period specified, rounded to the nearest integer multiple of the `pss` period.

Convergence parameters

18 `tolerance=1e-9` Relative tolerance for linear solver.

19 `gear_order=2` Gear order used for small-signal integration.

20 `solver=turbo` Solver type.
Possible values are `std` or `turbo`.

21 `oscsolver=turbo` Oscillator solver type.
Possible values are `std` or `turbo`.

Annotation parameters

22 `annotate=sweep` Degree of annotation.
Possible values are `no`, `title`, `sweep`, `status`, or `steps`.

23 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
24 `title`	Analysis title.

You can select the set of periodic small-signal output frequencies of interest by setting either the `maxsideband` or the `sidebands` parameters. For a given set of n integer numbers representing the sidebands K1, K2, ... Kn, the output frequency at each sideband is computed as f(out)= f(in) + Ki * fund(pss), where f(in) represent the (possibly swept) input frequency, and fund(pss) represents the fundamental frequency used in the corresponding PSS analysis. Thus, when analyzing a down-converting mixer, while sweeping the RF input frequency, the most relevant sideband for IF output is Ki= -1. When simulating an up-converting mixer, while sweeping IF input frequency, the most relevant sideband for RF output is Ki= 1. By setting the `maxsideband` value to Kmax, all 2 * Kmax + 1 sidebands from -Kmax to +Kmax are generated.

The number of requested sidebands does not change substantially the simulation time. However, the `maxacfreq` of the corresponding PSS analysis should be set to guarantee that | max{f(out)} | is less than `maxacfreq`, otherwise the computed solution might be contaminated by aliasing effects. The PAC simulation is not executed for | f(in) | greater than `maxacfreq`. Diagnostic messages are printed for those extreme cases, indicating how `maxacfreq` should be set in the PSS analysis. In the majority of the simulations, however, this is not an issue, because `maxacfreq` is never allowed to be smaller than 40x the PSS fundamental.

With PAC the frequency of the stimulus and of the response are usually different (this is an important way in which PAC differs from AC). The `freqaxis` parameter is used to specify whether the results should be output versus the input frequency (`in`), the output frequency (`out`), or the absolute value of the output frequency (`absout`).

Unlike AC analysis, PAC analysis can output the time-domain simulation results, by specifying the `outputperiod` parameter.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. Alternatively, you may specify the particular values that the sweep parameter should take using the `values` parameter. If you give both a specific set of values and a set specified using a sweep range, the two sets are merged and collated before being used. All frequencies are in Hertz.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Periodic Distortion Analysis (pdisto)

Description

Similar to PAC analysis, the periodic distortion (PDISTO) analysis calculates responses of a circuit that exhibits frequency translations. However, instead of having small signal linear behavior, it models the response as having components of a few harmonics of input signal frequencies. This allows computing responses to moderately large input signals. An example is intermodulation distortion with two moderate input signals. PDISTO treats one particular input signal (usually the one that causes the most nonlinearity or the largest response) as the large signal, and the others as moderate signals.

An initial transient analysis is carried out by first suppressing all moderate input signals. Then we run a number of (at least 2) stable iterations with all signals activated, which is followed by shooting Newton method. PDISTO employs the Mixed Frequency Time (MFT) algorithm extended to multiple fundamental frequencies. For details of MFT algorithm, please see Steady-State Methods for Simulating Analog and Microwave Circuits, by K. S. Kundert, J. K. White, and A. Sangiovanni-Vincentelli, Kluwer, Boston, 1990.

Like PSS, PDISTO uses shooting Newton method as its backbone. However, instead of doing a single transient integration, each Newton iteration does a number of transient integrations of one large signal period.

Each of the integrations differs by a phase-shift in each moderate input signal. The number of integrations is determined by the numbers of harmonics of moderate fundamentals specified by the user. Given maxharms=[k1 k2 ... kn], the total number of integrations is (2*k2+1)*(2*k3+1)*...*(2*kn+1). As one consequence, the efficiency of the algorithm depends significantly on the number of harmonics required to model the responses of moderate fundamentals. As another consequence, the number of harmonics of the large fundamental does not significantly affect the efficiency of the shooting algorithm. The boundary conditions of a shooting interval are such that the time domain integrations are consistent with a frequency domain transformation with a shift of one large signal period.

PDISTO inherits a majority of PSS parameters. A few new parameters are added. The most important ones are `funds` and `maxharms`. They replace PSS parameters, `fund` (or `period`) and `harms`, respectively. The `funds` parameter accepts a list of names of fundamentals that are present in the sources. These names are specified in the sources by parameter `fundname`. The first fundamental is considered as the large signal. A few heuristics can be used for picking the large fundamental.

(1) Pick the one which is not a sinusoidal.

(2) Pick the one which causes the most nonlinearity.

(3) Pick the one which causes the largest response.

The `maxharms` parameter accepts a list of numbers of harmonics that are required to sufficiently model responses due to different fundamentals.

Definition

Name pdisto parameter=value ...

Parameters

Distortion fundamental parameters

1	`funds=[...]`	Array of fundamental frequency names for fundamentals to use in analysis.
2	`maxharms=[...]`	Array of number of harmonics of each fundamental to consider for each fundamental.

Simulation interval parameters

3	`tstab=0.0 s`	Extra stabilization time after the onset of periodicity for independent sources.
4	`tstart=0.0 s`	Initial transient analysis start time.

Time-step parameters

5 `maxstep (s)` Maximum time step. Default derived from `errpreset`.

6 `step=0.001 period s`
Minimum time step that would be used solely to maintain the aesthetics of the results.

Initial-condition parameters

7 `ic=all` What should be used to set initial condition.
Possible values are `dc`, `node`, `dev`, or `all`.

8 `skipdc=no` If yes, there will be no dc analysis for transient.
Possible values are `no` or `yes`.

9 `readic` File that contains initial condition.

Convergence parameters

10 `readns` File that contains estimate of initial transient solution.

11 `cmin=0 F` Minimum capacitance from each node to ground.

Output parameters

12 `outputtype=time for PSS, freq for PDISTO`
Output type.
Possible values are `all`, `time` or `freq`.

13 `save` Signals to output.
Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.

14 `nestlvl` Levels of subcircuits to output.

15 `oppoint=no` Should operating point information be computed for initial timestep, and if so, where should it be sent.
Possible values are `no`, `screen`, `logfile`, or `rawfile`.

16 `skipstart=starttime s`
The time to start skipping output data.

17 `skipstop=stoptime s`
The time to stop skipping output data.

18 `skipcount` Save only one of every skipcount points.

19 `strobeperiod (s)` The output strobe interval (in seconds of transient time).

20 `strobedelay=0 s` The delay (phase shift) between the skipstart time and the first strobe point.

21 `compression=no` Do data compression on output.
Possible values are no or `yes`.

22 `saveinit=no` If set, the waveforms for the initial transient before steady state are saved.
Possible values are no or `yes`.

State-file parameters

23 `write` File to which initial transient solution (before steady-state) is to be written.

24 `writefinal` File to which final transient solution in steady-state is to be written.

25 `swapfile` Temporary file that holds steady-state information. Tells Spectre to use a regular file rather than virtual memory to hold the periodic operating point. Use this option if Spectre complains about not having enough memory to complete this analysis.

Integration method parameters

26 `method` Integration method. Default derived from `errpreset`.
Possible values are `euler`, `trap`, `traponly`, `gear2`, or `gear2only`.

Accuracy parameters

27 `errpreset=moderate`
Selects a reasonable collection of parameter settings.
Possible values are `liberal`, `moderate` or `conservative`.

28 `relref`	Reference used for the relative convergence criteria. Default derived from `errpreset`. Possible values are `pointlocal`, `alllocal`, `sigglobal`, or `allglobal`.
29 `lteratio`	Ratio used to compute LTE tolerances from Newton tolerance. Default derived from `errpreset`.
30 `steadyratio`	Ratio used to compute steady state tolerances from LTE tolerance. Default derived from `errpreset`.

31 `maxperiods=20 for driven PSS, 50 for autonomous PSS`
Maximum number of simulated periods to reach steady-state.

32 `tolerance=1e-4`	Relative tolerance for linear solver.
33 `finitediff`	Options for finite difference method refinement after quasi-periodic shooting method. finitediff is changed from no to samegrid automatically when readqpss and writeqpss are used to re-use QPSS results. Possible values are no, `yes` or `refine`.

Annotation parameters

34 `stats=no`	Analysis statistics. Possible values are no or `yes`.
35 `annotate=sweep`	Degree of annotation. Possible values are no, `title`, `sweep`, `status`, or `steps`.
36 `title`	Analysis title.

Newton parameters

37 `maxiters=5`	Maximum number of iterations per time step.
38 `restart=yes`	Do not use previous DC solution as initial guess. Possible values are no or `yes`.

Circuit age

39 `circuitage (Years)`	Stress Time. Age of the circuit used to simulate hot-electron degradation of MOSFET and BSIM circuits.
40 `writeqpss`	File to which final quasi-periodic steady-state solution is to be written. Small signal analyses such as qpac, qpxf and qpnoise can read in the steady-state solution from this file directly instead of running the qpss analysis again.
41 `readqpss`	File from which final quasi-periodic steady-state solution is to be read. Small signal analyses such as qpac, qpxf and qpnoise can read in the steady-state solution from this file directly instead of running the qpss analysis again.

Most of PDISTO analysis parameters are inherited from PSS analysis, and their meanings remain essentially unchanged. Two new important parameters are `funds` and `maxharms`. They replace and extend the role of `fund` and `harms` parameters of PSS analysis. One important difference is that `funds` accepts a list of fundamental names instead of actual frequencies. The frequencies associated with fundamentals are figured out automatically by the simulator. An important feature is that each input signal can be a composition of more than one sources. However, these sources must have the same fundamental name. For each fundamental name, its fundamental frequency is the greatest common factor of all frequencies associated with the name. Missing or not listing all fundamental names using the parameter `funds` will result in an amputation of the current simulation. However if `maxharms` is not given, a warning message will be issued, and the number of harmonics is defaulted to 1 for each of the fundamentals.

The role of some PSS parameters is extended. The parameter `maxperiods` that controls the maximum number of shooting iterations for PSS analysis also controls the number of the maximum number of shooting iterations for PDISTO analysis.

The tstab parameter controls both the length of the initial transient integration with only the clock tone activated and the number of stable iterations with moderate tones activated. The stable iterations are run before Newton iterations.

The `errpreset` parameter lets you adjust the simulator parameters to fit your needs quickly. In most cases, it should also be the only parameter you need to adjust. If you want a fast simulation with reasonable accuracy, you may set errpreset to liberal. If have some concern for accuracy, you may set errpreset to moderate. If accuracy is your main interest, you may set errpreset to conservative.

If users dont specify steadyratio, it is always 1.0 - not affected by `errpreset`. The effect of `errpreset` on other parameters is shown in the following table.

Parameter defaults as a function of errpreset

errpreset	reltol	relref	method	lteratio	maxstep
liberal	1e-3	sigglobal	gear2only	3.5	clock period/80
moderate	1e-4	sigglobal	gear2only	3.5	clock period/100
conservative	1e-5	sigglobal	gear2only	*	clock period/200

* : If user specified `reltol` <= 1e-5*10.0/3.5, lteratio=3.5. Otherwise lteratio=10.0.

The value of `reltol` can only decrease from its value in the options statement, when reltol in options statement is larger. Spectre sets the value of `maxstep` so that it is no larger than the value given in the table. Except for `reltol` and `maxstep`, `errpreset` does not change the value of any parameters you have explicitly set. The actual values used for the QPSS analysis are given in the log file.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Periodic Noise Analysis (pnoise)

Description

The Periodic Noise, or PNoise analysis is similar to the conventional noise analysis, except that it includes frequency conversion effects. Hence is it useful for predicting the noise behavior of mixers, switched-capacitor filters, and other periodically driven circuits. It is particularly useful for predicting the phase noise of autonomous circuits, such as oscillators.

PNoise analysis linearizes the circuit about the periodic operating point computed in the prerequisite PSS analysis. It is the periodically time-varying nature of the linearized circuit that accounts for the frequency conversion. In addition, the affect of a periodically time-varying bias point on the noise generated by the various components in the circuit is also included.

The time-average of the noise at the output of the circuit is computed in the form of a spectral density versus frequency. The output of the circuit is specified with either a pair of nodes or a probe component. To specify the output of a circuit with a probe, specify it using the `oprobe` parameter. If the output is voltage (or potential), choose a `resistor` or a `port` as the output probe. If the output is current (or flow), choose a `vsource` or `iprobe` as the output probe.

If the input-referred noise or noise figure is desired, specify the input source using the `iprobe` parameter. For input-referred noise, use either a vsource or isource as the input probe; for noise figure, use a port as the probe. Currently, only a `vsource`, an `isource`, or a `port` may be used as an input probe. If the input source is noisy, as is a `port`, the noise analysis will compute the noise factor (F) and noise figure (NF). To match the IEEE definition of noise figure, the input probe must be a port with no excess noise and its `noisetemp` must be set to 16.85C (290K). In addition, the output load must be a `resistor` or `port` and must be identified as the `oprobe`.

The reference sideband (`refsideband`) specifies which conversion gain is used when computing input-referred noise, noise factor, and noise figure. The reference sideband specifies the input frequency relative tothe output frequency with:

|f(input)| = |f(out) + refsideband * fund(pss)|

Use `refsideband=0` when the input and output of the circuit are at the same frequency (such as with amplifiers and filters). When `refsideband` differs from 0, the single side-band noise figure is computed.

The noise analysis always computes the total noise at the output, which includes contributions from the input source and the output load. The amount of the output noise that is attributable to each noise source in the circuit is also computed and output individually. If the input source is identified (using iprobe) and is a vsource or isourec, the input-referred noise is computed, which includes the noise from the input source itself. Finally, if the input source is identified (using iprobe) and is noisy, as is the case with ports, the noise factor and noise figure are computed. Thus if

No = total output noise

Ns = noise at the output due to the input probe (the source)

Nsi = noise at the output due to the image harmonic at the source

Nso = noise at the output due to harmonics other than input at the source

Nl = noise at the output due to the output probe (the load)

IRN = input referred noise

G = gain of the circuit

F = noise factor

NF = noise figure

Fdsb = double sideband noise factor

NFdsb = double sideband noise figure

Fieee = IEEE single sideband noise factor

NFieee = IEEE single sideband noise figure

then,

IRN = sqrt(No^2/G^2)

F = (No^2 - Nl^2)/Ns^2

NF = 10*log10(F)

Fdsb = (No^2 - Nl^2)/(Ns^2+Nsi^2)

NFdsb = 10*log10(Fdsb)

Fieee = (No^2 - Nl^2 - Nso^2)/Ns^2

NFieee = 10*log10(Fieee).

When the results are output, No is named `out`, IRN is named `in`, G is named `gain`, F, NF, Fdsb, NFdsb, Fieee, and NFieee are named F, `NF`, `Fdsb`, `NFdsb`, `Fieee`, and `NFieee` respectively.

Unlike other analyses in Spectre, this analysis can only sweep frequency.

Definition

Name [p] [n] pnoise parameter=value ...

The optional terminals (p and n) specify the output of the circuit. If you do not give the terminals, then you must specify the output with a probe component.

Parameters

Sweep interval parameters

1 `start=0`	Start sweep limit.
2 `stop`	Stop sweep limit.
3 `center`	Center of sweep.
4 `span=0`	Sweep limit span.
5 `step`	Step size, linear sweep.
6 `lin=50`	Number of steps, linear sweep.
7 `dec`	Points per decade.
8 `log=50`	Number of steps, log sweep.
9 `values=[...]`	Array of sweep values.
10 `sweeptype`	Specifies if the sweep frequency range is absolute frequency of input or if it is relative to the port harmonics. Possible values are `absolute` or `relative`.
11 `relharmnum=1`	Harmonic to which relative frequency sweep should be referenced.

Probe parameters

12 `oprobe` Compute total noise at the output defined by this component.

13 `iprobe` Refer the output noise to this component.

14 `refsideband` Conversion gain associated with this sideband is used when computing input-referred noise or noise figure.

Output parameters

15 `noisetype=sources`
Specifies if the pnoise analysis should output cross-power densities or noise source information.
Possible values are `sources`, `correlations` or `timedomain`.

16 `maxsideband=7` Maximum sideband included when computing noise either up-converted or down-converted to the output by the periodic drive signal.

17 `sidebands=[...]` Array of relevant sidebands for the analysis.

18 `save` Signals to output.
Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.

19 `nestlvl` Levels of subcircuits to output.

20 `maxcycles=0` Maximum cycle correlation frequency included when computing noise either up-converted or down-converted to the output by the periodic drive signal.

21 `cycles=[...]` Array of relevant cycle frequencies. Valid only if noisetype=correlations.

22 `noiseskipcount=0` Calculate time-domain noise on only one of every noiseskipcount time points.

23 `noisetimepoints=[...]` Additional timepoints for time-domain noise analysis..

24 `saveallsidebands=no`
Save noise contributors by sideband.
Possible values are `no` or `yes`.

Convergence parameters

25 `tolerance=1e-9`	Relative tolerance for linear solver.
26 `gear_order=2`	Gear order used for small-signal integration.
27 `solver=turbo`	Solver type. Possible values are `std` or `turbo`.
28 `oscsolver=turbo`	Oscillator solver type. Possible values are `std` or `turbo`.

Annotation parameters

29 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
30 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
31 `title`	Analysis title.

In practice, noise can mix with each of the harmonics of the periodic drive signal applied in the PSS analysis and end up at the output frequency. However, the PNoise analysis only includes the noise that mixes with a finite set of harmonics that are typically specified using the `maxsideband` parameter, but in special circumstances may be specified with the `sidebands` parameter. If Ki represents sideband i, then

f(noise_source) = f(out) + Ki * fund(pss)

The `maxsideband` parameter specifies the maximum |Ki| included in the PNoise calculation. Thus, noise at frequencies less than f(out)-maxsideband*fund(pss) and greater than f(out)+maxsideband*fund(pss) are ignored. If selected sidebands are specified using the `sidebands` parameter, then only those are included in the calculation. Care should be taken when specifying the sidebands because the results will be in error if you do not include a sideband that contributes significant noise to the output.

The number of requested sidebands does not change substantially the simulation time. However, the `maxacfreq` of the corresponding PSS analysis should be set to guarantee that |max{f(noise_source)}| is less than `maxacfreq`, otherwise the computed solution might be contaminated by aliasing effects. The PNoise simulation is not executed for |f(out)| greater than `maxacfreq`. Diagnostic messages are printed for those extreme cases, indicating which `maxacfreq` should be set in the PSS analysis. In the majority of the simulations, however, this is not an issue, because `maxacfreq` is never allowed to be smaller than 40x the PSS fundamental.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. Alternatively, you may specify the particular values that the sweep parameter should take using the `values` parameter. If you give both a specific set of values and a set specified using a sweep range, the two sets are merged and collated before being used. All frequencies are in Hertz.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

annotate 29	maxcycles 20	refsideband 14	stats 30
center 3	maxsideband 16	relharmnum 11	step 5
cycles 21	nestlvl 19	save 18	stop 2
dec 7	noiseskipcount 22	saveallsidebands 24	sweeptype 10
gear_order 26	noisetimepoints 23	sidebands 17	title 31
iprobe 13	noisetype 15	solver 27	tolerance 25
lin 6	oprobe 12	span 4	values 9
log 8	oscsolver 28	start 1	

Periodic S-Parameter Analysis (psp)

Description

The periodic SP (`psp`) analysis is used to compute scattering and noise parameters for n-port circuits that exhibit frequency translation, such as mixers. It is a small-signal analysis like SP analysis, except, as in PAC and PXF, the circuit is first linearized about a periodically varying operating point as opposed to a simple DC operating point. Linearizing about a periodically time-varying operating point allows the computation of S-parameters between circuit ports that convert signals from one frequency band to another. PSP can also calculate noise parameters in frequency-converting circuits. PSP computes noise figure (both single-sideband and double-sideband), input referred noise, equivalent noise parameters, and noise correlation matrices. As in PNoise, but unlike SP, the noise features of the PSP analysis include noise folding effects due to the periodically time-varying nature of the circuit.

Computing the n-port S-parameters and noise parameters of a periodically varying circuit is a two step process. First, the small stimulus is ignored and the periodic steady-state response of the circuit to possibly large periodic stimulus is computed using PSS analysis. As a normal part of the PSS analysis, the periodically time-varying representation of the circuit is computed and saved for later use. The second step is applying small-signal excitations to compute the n-port S-parameters and noise parameters. This is done using the PSP analysis. A PSP analysis cannot be used alone, it must follow a PSS analysis. However, any number of periodic small-signal analyses such as PAC, PSP, PXF, PNoise, can follow a single PSS analysis.

Unlike other analyses in Spectre, this analysis can only sweep frequency.

Definition

Name psp parameter=value ...

Parameters

Sweep interval parameters

1	`start=0`	Start sweep limit.
2	`stop`	Stop sweep limit.
3	`center`	Center of sweep.
4	`span=0`	Sweep limit span.
5	`step`	Step size, linear sweep.
6	`lin=50`	Number of steps, linear sweep.
7	`dec`	Points per decade.
8	`log=50`	Number of steps, log sweep.
9	`values=[...]`	Array of sweep values.

10 `sweeptype`	Specifies if the sweep frequency range is absolute frequency of input or if it is relative to the port harmonics. Possible values are `absolute` or `relative`.

Port parameters

11 `ports=[...]`	List of active ports. Ports are numbered in the order given. For purposes of noise figure computation, the input is considered port 1 and the output is port 2.
12 `portharmsvec=[...]`	List of harmonics active on specified list of ports. Must have a one-to-one correspondence with the ports vector.
13 `harmsvec=[...]`	List of harmonics, in addition to ones associated with specific ports by portharmsvec, that are active.

Output parameters

14 `freqaxis`	Specifies whether the results should be output versus the input frequency, the output frequency, or the absolute value of the input frequency. Default is `in`. Possible values are `absin`, `in` or `out`.

Noise parameters

15 `donoise=yes`	Perform noise analysis. If oprobe is specified as a valid port, this is set to yes, and a detailed noise output is generated. Possible values are `no` or `yes`.

Probe parameters

16 `maxsideband=7`	Maximum sideband included when computing noise either up-converted or down-converted to the output by the periodic drive signal.

Convergence parameters

17 `tolerance=1e-9`	Relative tolerance for linear solver.
18 `gear_order=2`	Gear order used for small-signal integration.

19 `solver=turbo`	Solver type. Possible values are `std` or `turbo`.
20 `oscsolver=turbo`	Oscillator solver type. Possible values are `std` or `turbo`.

Annotation parameters

21 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
22 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
23 `title`	Analysis title.

To specify the PSP analysis the port and port harmonic relations must be specified. You can select the ports of interest by setting the port parameter and the set of periodic small-signal output frequencies of interest by setting the `portharmsvec` or the `harmsvec` parameters. For a given set of n integer numbers representing the harmonics K1, K2, ... Kn, the scattering parameters at each port are computed at the frequencies f(scattered)= f(rel) + Ki * fund(pss), where f(rel) represents the relative frequency of a signal incident on a port, f(scattered) represents the frequency to which the relevant scattering parameter represents the conversion, and fund(pss) represents the fundamental frequency used in the corresponding PSS analysis.

Thus, when analyzing a down-converting mixer, with signal in the upper sideband, and sweeping the RF input frequency, the most relevant harmonic for RF input is Ki= 1 and for IF output Ki= 0. Hence we can associate K2=1 with the IF port and K1=0 with the RF port. S21 will represent the transmission of signal from the RF to IF, and S11 the reflection of signal back to the RF port. If the signal was in the lower sideband, then a choice of K1=-1 would be more appropriate.

Either the portharmsvec or the harmsvec parameters can be used to specify the harmonics of interest. If the portharmsvec is given, the harmonics must be in one-to-one correspondence with the ports, with each harmonic associated with a single port. If harmonics are specified in the optional harmsvec parameter, then all possible frequency-translating scattering parameters associated with the specified harmonics are computed.

With PSP the frequency of the input and of the response are usually different (this is an important way in which PSP differs from SP). Because the PSP computation involves inputs and outputs at frequencies that are relative to multiple harmonics, the freqaxis and sweeptype parameters behave somewhat differently in PSP than in PAC and PXF.

The sweeptype parameter controls the way the frequencies in the PSP analysis are swept. Specifying a relative sweep indicates to sweep relative to the analysis harmonics (not the PSS fundamental) and an absolute sweep is a sweep of the absolute input source frequency. For example, with a PSS fundamental of 100MHz, the portharmsvec set to [9 1] to examine a downconverting mixer, sweeptype=relative, and a sweep range of f(rel)=0->50MHz, then S21 would represent the strength of signal transmitted from the input port in the range 900->950MHz to the output port at frequencies 100->150MHz. Using sweeptype=absolute and sweeping the frequency from 900->950MHz would calculate the same quantities, since f(abs)=900->950MHz, and f(rel) = f(abs) - K1 * fund(pss) = 0->50MHz, because K1=9 and fund(pss) = 100MHz.

Usually it is not necessary to sweep frequency in PSP over more than one fundamental PSS period.

The `freqaxis` parameter is used to specify whether the results should be output versus the scattered frequency at the input port (`in`), the scattered frequency at the output port (`out`), or the absolute value of the frequency swept at the input port (`absin`).

Unlike in PAC/PXF/PNoise, increasing the number of requested ports and harmonics will increase the simulation time substantially.

To insure accurate results in PSP, the `maxacfreq` of the corresponding PSS analysis should be set to guarantee that | max{f(scattered)} | is less than `maxacfreq`, otherwise the computed solution might be contaminated by aliasing effects.

PSP analysis also computes noise figures, equivalent noise sources, and noise parameters. The noise computation, which is skipped only when donoise is set to no, requires additional simulation time. If

No = total output noise at frequency f

Ns = noise at the output due to the input probe (the source)

Nsi = noise at the output due to the image harmonic at the source

Nso = noise at the output due to harmonics other than input at the source

Nl = noise at the output due to the output probe (the load)

IRN = input referred noise

G = gain of the circuit

F = noise factor (single side band)

NF = noise figure (single side band)

Fdsb = double sideband noise factor

NFdsb = double sideband noise figure

Fieee = IEEE single sideband noise factor

NFieee = IEEE single sideband noise figure

then,

IRN = sqrt(No^2/G^2)

F = (No^2 - Nl^2)/Ns^2

NF = 10*log10(F)

Fdsb = (No^2 - Nl^2)/(Ns^2+Nsi^2)

NFdsb = 10*log10(Fdsb)

Fieee = (No^2 - Nl^2 - Nso^2)/Ns^2

NFieee = 10*log10(Fieee).

When the results are output, IRN is named `in`, G is named `gain`, F, NF, Fdsb, NFdsb, Fieee, and NFieee are named F, NF, Fdsb, NFdsb, Fieee, and NFieee respectively. Note that the gain computed by PSP is the voltage gain from the actual circuit input to the circuit output, not the gain from the internal port voltage source to the output.

To insure accurate noise calculations, the maxsideband or sidebands parameters must be set to include the relevant noise folding effects. maxsideband is only relevant to the noise computation features of PSP.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. Alternatively, you may specify the particular values that the sweep parameter should take using the `values` parameter. If you give both a specific set of values and a set specified using a sweep range, the two sets are merged and collated before being used. All frequencies are in Hertz.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Periodic Steady-State Analysis (pss)

Description

This analysis computes the periodic steady-state (PSS) response of a circuit, with a simulation time independent of the time-constants of the circuit. Also, it sets the circuits periodic operating point, which can then be used during a periodic time-varying small-signal analysis, such as PAC, PXF, and PNOISE.

PSS analysis is capable of handling both autonomous (non-driven) and driven (non-autonomous) circuits. Autonomous circuits are time-invariant circuits that have time-varying responses. Thus, autonomous circuits generate non-constant waveforms even though they are not driven by a time-varying stimulus. Driven circuits require some time-varying stimulus to generate a time-varying response. The most common example of an autonomous circuit is an oscillator. Common driven circuits include amplifiers, filters, mixers, etc.

With driven circuits the user specifies the analysis `period`, or its corresponding fundamental frequency `fund`. The `period` must be an integer multiple of the period of the drive signal or signals. Autonomous circuits have no drive signal and the actual period of oscillation is not known precisely by the user in advance. Instead, the user specifies an estimate of the oscillation period and PSS analysis computes the precise period along with the periodic solution waveforms.

When applied to autonomous circuits, PSS analysis requires the user to specify a pair of nodes, p and n. In fact this is how PSS analysis determines whether it is being applied to an autonomous or a driven circuit. If the pair of nodes is supplied, PSS assumes the circuit is autonomous; if not, the circuit is assumed to be driven.

A PSS analysis consists of two phases, an initial transient phase, which allows the circuit to be initialized, and the shooting phase, which is where the periodic steady-state solution is computed. The transient phase consists of three intervals.

The first starts at `tstart`, which is normally 0, and continues through the onset of periodicity `tonset` for the independent sources. The onset of periodicity, which is automatically generated, is the minimum time for which all sources are periodic. The second is an optional user specified stabilization interval whose length is `tstab`. The final interval whose length is `period` for driven circuits, or 4x `period` for autonomous circuits has a special use for the autonomous PSS analysis, i.e., the PSS analysis monitors the waveforms in the circuit and develops a better estimate of the oscillation period. Once the initial transient phase is complete, the shooting interval begins. In this phase, the circuit is repeatedly simulated over one period while adjusting the initial condition (and the period when applied to autonomous circuits) to find the periodic steady-state solution.

Typically the process takes three to five such iterations to reach steady-state. Upon completion, if requested by the user, the frequency-domain response is computed. For driven circuits, one can use writepss and readpss to reuse the results in a previous simulation.

Definition

Name [p] [n] pss parameter=value ...

Parameters

Simulation interval parameters

1 `period (s)`	Steady state analysis period (or its estimate for autonomous circuits).
2 `fund (Hz)`	Alternative to period specification. Steady state analysis fundamental frequency (or its estimate for autonomous circuits).
3 `tstab=0.0 s`	Extra stabilization time after the onset of periodicity for independent sources.
4 `tstart=0.0 s`	Initial transient analysis start time.

Time-step parameters

5 `maxstep (s)` Maximum time step. Default derived from `errpreset`.

6 `maxacfreq` Maximum frequency requested in a subsequent periodic small-signal analysis. Default derived from `errpreset` and `harms`.

7 `step=0.001 period s`
Minimum time step that would be used solely to maintain the aesthetics of the results.

Initial-condition parameters

8 `ic=all` What should be used to set initial condition.
Possible values are `dc`, `node`, `dev`, or `all`.

9 `skipdc=no` If yes, there will be no dc analysis for transient.
Possible values are `no` or `yes`.

10 `readic` File that contains initial condition.

Convergence parameters

11 `readns` File that contains estimate of initial transient solution.

12 `cmin=0 F` Minimum capacitance from each node to ground.

Output parameters

13 `harms=9` Number of harmonics to output when outputtype=`freq` or `all`.

14 `harmsvec=[...]` Array of desired harmonics. Alternate form of `harms` that allows selection of specific harmonics.

15 `outputtype=time for PSS, freq for PDISTO`
Output type.
Possible values are `all`, `time` or `freq`.

16 `save` Signals to output.
Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.

17 `nestlvl` Levels of subcircuits to output.

18 `oppoint=no` Should operating point information be computed for initial timestep, and if so, where should it be sent.
Possible values are `no`, `screen`, `logfile`, or `rawfile`.

19 `skipstart=starttime s`
The time to start skipping output data.

20 `skipstop=stoptime s`
The time to stop skipping output data.

21 `skipcount` Save only one of every skipcount points.

22 `strobeperiod (s)` The output strobe interval (in seconds of transient time).

23 `strobedelay=0 s` The delay (phase shift) between the skipstart time and the first strobe point.

24 `compression=no` Do data compression on output.
Possible values are `no` or `yes`.

25 `saveinit=no` If set, the waveforms for the initial transient before steady state are saved.
Possible values are `no` or `yes`.

State-file parameters

26 `write` File to which initial transient solution (before steady-state) is to be written.

27 `writefinal` File to which final transient solution in steady-state is to be written.

28 `swapfile` Temporary file that holds steady-state information. Tells Spectre to use a regular file rather than virtual memory to hold the periodic operating point. Use this option if Spectre complains about not having enough memory to complete this analysis.

29 `writepss` File to which the converged steady-state solution is to be written. finitediff is set to yes automatically to improve PSS results.

30 `readpss` — File from which a previously converged steady-state solution is to be read. PSS loads the solution and checks the residue of the circuit equations only. The solution is re-used if the residue is satisfying. Otherwise, the solution is re-converged using the finite difference method.

Integration method parameters

31 `method` — Integration method. Default derived from `errpreset`.
Possible values are `euler`, `trap`, `traponly`, `gear2`, or `gear2only`.

Accuracy parameters

32 `errpreset=moderate`
Selects a reasonable collection of parameter settings.
Possible values are `liberal`, `moderate` or `conservative`.

33 `relref` — Reference used for the relative convergence criteria. Default derived from `errpreset`.
Possible values are `pointlocal`, `alllocal`, `sigglobal`, or `allglobal`.

34 `lteratio` — Ratio used to compute LTE tolerances from Newton tolerance. Default derived from `errpreset`.

35 `steadyratio` — Ratio used to compute steady state tolerances from LTE tolerance. Default derived from `errpreset`.

36 `maxperiods=20 for driven PSS, 50 for autonomous PSS`
Maximum number of simulated periods to reach steady-state.

37 `tolerance=1e-4` — Relative tolerance for linear solver.

38 `finitediff` — Options for finite difference method refinement after shooting method for driven circuits.
Possible values are no, `yes` or `refine`.

39 `highorder` — Perform a high-order refinement after low-order convergence. The Multi-Interval Chebyshev polynomial spectral algorithm will be used.
Possible values are `no` or `yes`.

40 `psaratio=1`	Ratio used to compute high-order polynomial spectral accuracy from Newton tolerance.
41 `maxorder=16`	The maximum order of the Chebyshev polynomials used in waveform approximation. Possible values are from 2 to 16.
42 `fullpssvec`	Use the full vector containing solutions at all PSS time steps in the linear solver. Default derived from the size of the equation and the property of the PSS time steps. Possible values are `no` or `yes`.

Annotation parameters

43 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
44 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
45 `title`	Analysis title.

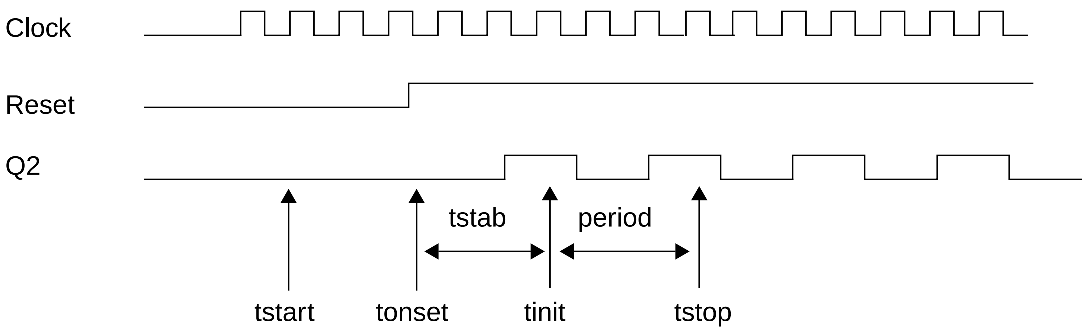
Newton parameters

46 `maxiters=5`	Maximum number of iterations per time step.
47 `restart=yes`	Do not use previous DC solution as initial guess. Possible values are `no` or `yes`.

Circuit age

48 `circuitage (Years)`	Stress Time. Age of the circuit used to simulate hot-electron degradation of MOSFET and BSIM circuits.

The initial transient analysis provides a flexible mechanism to direct the circuit to a particular steady-state solution of interest, and to avoid undesired solutions. Another usage of the initial transient simulation is helping convergence by eliminating large but fast decaying modes that are present in many circuits. For example, in case of driven circuits, consider the reset signal in the figure below.



In the figure above, the initial transient analysis is executed from `tstart` to `tstop`. If initial transient results are relevant, you can output them by setting `saveinit` to yes. The steady-state results are always computed for the specified `period`, from `tinit` to `tstop`. By default, `tstart` and `tstab` are set to zero, while `tinit`, `tonset` and `tstop` are always automatically generated.

It happens in some circuits that the linearity of the relationship between the initial and final state depends on when the shooting interval begins. Conceptually, when the shooting interval begins should not matter, as long as it is after the time when the stimuli have become periodic, because the periodic response repeats endlessly. However in practice, one can improve the convergence by starting at a good point, and degrade the convergence, which slows the analysis, by starting in a bad spot. In general, it is best to try to avoid starting the shooting interval at a point where the circuit is undergoing strong nonlinear behavior. For example, in switch-capacitor filters it is best if `tinit` falls at the beginning of a clock transition, preferably a transition that follows a relatively long period of settling. If instead `tinit` occurred during a clock transition or soon after one, then it is likely the opamps would be undergoing slew-rate limiting at the start of the shooting interval, which would act to slow convergence. Switching mixers follow similar rules.

When applying PSS analysis to oscillators, it is necessary to start the oscillator, just as you would if you were simulating the turn-on transient of the oscillator using transient analysis. The Designers Guide to Spice and Spectre [K. S. Kundert, Kluwer Academic Publishers, 1995] describes techniques for starting oscillators in some depth. In summary, there are two techniques for starting oscillators, using initial conditions, or using a brief impulsive stimulus. Initial conditions would be provided for the components of the oscillators resonator. If an impulsive stimulus is used, it should be applied so as to couple strongly into the oscillatory mode of the circuit, and poorly into any other long-lasting modes, such as those associated with bias circuitry. Either way, once the trigger is applied to start the oscillator, it is important to allow the oscillator to run for a while before the shooting methods are applied to compute the steady-state result. To do so, specify an additional stabilization interval using the `tstab` parameter. In practice, an additional stabilization interval often improves convergence.

By default, only the time-domain results are computed. If you specify either `harms` or `harmsvec` or set `outputtype` to `freq` or `all`, the frequency-domain results will also be computed. If frequency-domain results are requested, but the desired harmonics are not specified, its default value is 10. The time-domain output waveform generation can be inhibited by setting `outputtype` to `freq`.

The accuracy of the results does not depend on the number of harmonics that are requested, only on the accuracy parameters, which are set in the same fashion as in the transient analysis. Besides a few new parameters, like `steadyratio` and `maxacfreq`, all the others parameters work in PSS analysis in the exact same fashion as they work on transient analysis.

Several parameters determine the accuracy of the PSS analysis. `reltol` and `abstol` control the accuracy of the discretized equation solution. These parameters determine how well charge is conserved and how accurately steady-state or equilibrium points are computed. You can set the integration error, or the errors in the computation of the circuit dynamics (such as time constants), relative to `reltol` and `abstol` by setting the `lteratio` parameter.

The `steadyratio` parameter adjusts the maximum allowed mismatch in node voltages or current branches from the beginning to the end of the steady-state period. This value is multiplied by the `lteratio` and `reltol` to determine the convergence criterion. The relative convergence norm is printed out along with the actual mismatch value at the end of each iteration, thus indicating the progress of the steady-state iteration.

The finitediff parameter allows the use of finite difference (FD) after shooting. Usually this will eliminate the above mismatch in node voltages or current branches. It can also refine the grid of time steps. In some cases, numerical error of the linear solver still introduces a mismatch. One can set `steadyratio` to a smaller value to activate a tighter tolerance for the iterative linear solver. If finitediff is set to no, FD method is turned off. If it is set to yes, pss applies FD method and trying to improve the beginning small time steps if necessary. If it is set to refine, pss applies FD method and tries to refine the time steps. When the simulation uses 2nd-order method, uniform 2nd order gear is used. finitediff is changed from no to yes automatically when readpss and writepss are used to re-use PSS results

The `maxacfreq` parameter is used to automatically adjust the `maxstep` to reduce errors due to aliasing in frequency-domain results. By default, the `maxacfreq` is set to 4x the frequency of the largest requested harmonic, but is never set to less than 40x the fundamental.

The parameter `relref` determines how the relative error is treated. The `relref` options are:

`relref=pointlocal`: Compares the relative errors in quantities at each node to that node alone.

`relref=alllocal`: Compares the relative errors at each node to the largest values found for that node alone for all past time.

`relref=sigglobal`: Compares relative errors in each of the circuit signals to the maximum for all signals at any previous point in time.

`relref=allglobal`: Same as `relref=sigglobal` except that it also compares the residues (KCL error) for each node to the maximum of that nodes past history.

The `errpreset` parameter lets you adjust the simulator parameters to fit your needs quickly. In most cases, it should also be the only parameter you need to adjust.

Guidelines for using `errpreset` in driven circuits are as follows. If the circuit contains only one periodic tone and you are only interested in obtaining the periodic operating point, you may set `errpreset` to liberal, which gives reasonably accurate result and the simulation speed is the fastest. If the circuit contains more than one periodic tone and you are interested in intermodulation results, you may set `errpreset` to moderate, which gives very accurate result. If you want a very low noise floor in your simulation result and accuracy is your main interest, you may set `errpreset` to conservative. Multi-interval Chebsyehv (MIC) is activated automatically for moderate and conservative settings, unless you set `highorder=no` explicitly. MIC falls back to the original method if it encounters convergence difficulty. If you set `highorder=yes`, MIC will try harder to converge.

The effect of `errpreset` on other parameters for driven circuits is shown in the following table.

Parameter Defaults and Estimated Numerical Noise Floor in Simulation Result as a Function of errpreset

errpreset	reltol	relref	method	lteratio	steadyratio	noisefloor
liberal	1e-3	sigglobal	gear2only	3.5	0.001	-70dB
moderate	1e-3	alllocal	gear2only+mic	3.5	0.001	-120dB
conservative	1e-4	alllocal	gear2only+mic	*	0.01	-200dB

* : If user specified reltol <= 1e-4*10.0/3.5, lteratio=3.5. Otherwise lteratio=10.0.

Estimated numerical noise floor is for weakly nonlinear circuit with successful MIC simulation. For linear circuit, the noise floor is even lower and for very nonlinear circuit, you may need to tighten psaratio or increase maxacfreq to achieve this noise floor.

Guidelines for using errpreset in autonomous circuits are as follows. If you want a fast simulation with reasonable accuracy, you may set `errpreset` to liberal. If have some concern for accuracy, you may set `errpreset` to moderate. If accuracy is your main interest, you may set `errpreset` to conservative.

The effect of `errpreset` on other parameters for autonomous circuits is shown in the following table.

Parameter defaults as a function of errpreset

errpreset	reltol	relref	method	lteratio	steadyratio	maxstep
liberal	1e-3	sigglobal	traponly	3.5	0.001	period/80
moderate	1e-4	alllocal	gear2only	3.5	0.001	period/200
conservative	1e-5	alllocal	gear2only	*	0.001	period/400

* : If user specified `reltol` <= 1e-5*10.0/3.5, `lteratio`=3.5. Otherwise `lteratio`=10.0.

The value of `reltol` can only decrease from its value in the options statement, when reltol in options statement is larger. Spectre sets the value of `maxstep` so that it is no larger than the value given in the table. Except for `reltol` and `maxstep`, `errpreset` does not change the value of any parameters you have explicitly set. The actual values used for the PSS analysis are given in the log file.

If the circuit you are simulating can have infinitely fast transitions (for example, a circuit that contains nodes with no capacitance), Spectre might have convergence problems. To avoid this, you must prevent the circuit from responding instantaneously. You can accomplish this by setting `cmin`, the minimum capacitance to ground at each node, to a physically reasonable nonzero value. This often significantly improves Spectre convergence.

You may specify the initial condition for the transient analysis by using the `ic` statement or the `ic` parameter on the capacitors and inductors. If you do not specify the initial condition, the DC solution is used as the initial condition. The `ic` parameter on the transient analysis controls the interaction of various methods of setting the initial conditions. The effects of individual settings are:

`ic=dc`: Any initial condition specifiers are ignored, and the DC solution is used.

`ic=node`: The `ic` statements are used, and the `ic` parameter on the capacitors and inductors are ignored.

`ic=dev`: The `ic` parameters on the capacitors and inductors are used, and the `ic` statements are ignored.

`ic=all`: Both the `ic` statements and the `ic` parameters are used, and the `ic` parameters override the `ic` statements.

If you specify an initial condition file with the `readic` parameter, initial conditions from the file are used, and any `ic` statements are ignored.

Once you specify the initial conditions, Spectre computes the actual initial state of the circuit by performing a DC analysis. During this analysis, Spectre forces the initial conditions on nodes by using a voltage source in series with a resistor whose resistance is `rforce` (see `options`).

With the `ic` statement it is possible to specify an inconsistent initial condition (one that cannot be sustained by the reactive elements). Examples of inconsistent initial conditions include setting the voltage on a node with no path of capacitors to ground or setting the current through a branch that is not an inductor. If you initialize Spectre inconsistently, its solution jumps; that is, it changes instantly at the beginning of the simulation interval. You should avoid such changes if possible because Spectre can have convergence problems while trying to make the jump.

You can skip the DC analysis entirely by using the parameter `skipdc`. If the DC analysis is skipped, the initial solution will be either trivial, or given in the file you specified by the `readic` parameter, or, if the `readic` parameter is not given, the values specified on the ic statements. Device based initial conditions are not used for `skipdc`. Nodes that you do not specify with the `ic` file or `ic` statements will start at zero. You should not use this parameter unless you are generating a nodeset file for circuits that have trouble in the DC solution; it usually takes longer to follow the initial transient spikes that occur when the DC analysis is skipped than it takes to find the real DC solution. The `skipdc` parameter might also cause convergence problems in the transient analysis.

Nodesets help find the DC or initial transient solution. You can supply them in the circuit description file with nodeset statements, or in a separate file using the `readns` parameter. When nodesets are given, Spectre computes an initial guess of the solution by performing a DC analysis while forcing the specified values onto nodes by using a voltage source in series with a resistor whose resistance is `rforce`. Spectre then removes these voltage sources and resistors and computes the true solution from this initial guess.

Nodesets have two important uses. First, if a circuit has two or more solutions, nodesets can bias the simulator towards computing the desired one. Second, they are a convergence aid.

By estimating the solution of the largest possible number of nodes, you might be able to eliminate a convergence problem or dramatically speed convergence.

When you simulate the same circuit many times, we suggest that you use both the `write` and `readns` parameters and give the same file name to both parameters. The DC analysis then converges quickly even if the circuit has changed somewhat since the last simulation, and the nodeset file is automatically updated.

Nodesets and initial conditions have similar implementation but produce different effects. Initial conditions actually define the solution, whereas nodesets only influence it. When you simulate a circuit with a transient analysis, Spectre forms and solves a set of differential equations. However, differential equations have an infinite number of solutions, and a complete set of initial conditions must be specified in order to identify the desired solution. Any initial conditions you do not specify are computed by the simulator to be consistent. The transient waveforms then start from initial conditions. Nodesets are usually used as a convergence aid and do not affect the final results. However, in a circuit with more than one solution, such as a latch, nodesets bias the simulator towards finding the solution closest to the nodeset values.

The `method` parameter specifies the integration method. The possible settings and their meanings are:

`method=euler`: Backward-Euler is used exclusively.

`method=traponly`: Trapezoidal rule is used almost exclusively.

`method=trap`: Backward-Euler and the trapezoidal rule are used.

`method=gear2only`: Gears second-order backward-difference method is used almost exclusively.

`method=gear2`: Backward-Euler and second-order Gear are used.

The trapezoidal rule is usually the most efficient when you want high accuracy. This method can exhibit point-to-point ringing, but you can control this by tightening the error tolerances. For this reason, though, if you choose very loose tolerances to get a quick answer, either backward-Euler or second-order Gear will probably give better results than the trapezoidal rule. Second-order Gear and backward-Euler can make systems appear more stable than they really are. This effect is less pronounced with second-order Gear or when you request high accuracy.

Spectre provides two methods for reducing the number of output data points saved: `strobing`, based on the simulation time, and `skipping` time points, which saves only every Nth point.

The parameters `strobeperiod` and `strobedelay` control the strobing method.`strobeperiod` sets the interval between points that you want to save, and `strobedelay` sets the offset within the period relative to `skipstart`. The simulator forces a time step on each point to be saved, so the data is computed, not interpolated.

The skipping method is controlled by `skipcount`. If this is set to N, then only every Nth point is saved.

The parameters `skipstart` and `skipstop` apply to both data reduction methods. Before `skipstart` and after `skipstop`, Spectre saves all computed data.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Periodic Transfer Function Analysis (pxf)

Description

A conventional transfer function analysis computes the transfer function from every source in the circuit to a single output. It differs from a conventional AC analysis in that the AC analysis computes the response from a single stimulus to every node in the circuit. The difference between pac and pxf analysis are similar. The Periodic Transfer Function (pxf) analysis computes the transfer functions from any source at any frequency to a single output at a single frequency. Thus, like pac analysis, pxf analysis includes frequency conversion effects.

The pxf analysis directly computes such useful quantities as conversion efficiency (transfer function from input to output at desired frequency), image and sideband rejection (input to output at undesired frequency), and LO feed-through and power supply rejection (undesired input to output at all frequencies).

As with a pac, psp, and pnoise analyses, a pxf analysis must follow a PSS analysis.

Unlike other analyses in Spectre, this analysis can only sweep frequency.

Definition

Name [p] [n] pxf parameter=value ...

The optional terminals (p and n) specify the output of the circuit. If you do not give the terminals, then you must specify the output with a probe component.

Parameters

Sweep interval parameters

1 `start=0`	Start sweep limit.
2 `stop`	Stop sweep limit.
3 `center`	Center of sweep.
4 `span=0`	Sweep limit span.
5 `step`	Step size, linear sweep.
6 `lin=50`	Number of steps, linear sweep.
7 `dec`	Points per decade.
8 `log=50`	Number of steps, log sweep.

9 `values=[...]`	Array of sweep values.
10 `sweeptype`	Specifies if the sweep frequency range is absolute frequency of input or if it is relative to the port harmonics. Possible values are `absolute` or `relative`.
11 `relharmnum=1`	Harmonic to which relative frequency sweep should be referenced.

Probe parameters

12 `probe`	Compute every transfer function to this probe component.

Output parameters

13 `stimuli=sources`	Stimuli used for xf analysis. Possible values are `sources` or `nodes_and_terminals`.
14 `sidebands=[...]`	Array of relevant sidebands for the analysis.
15 `maxsideband=0`	An alternative to the `sidebands` array specification, which automatically generates the array: [-maxsideband ... 0 ... +maxsideband].
16 `freqaxis`	Specifies whether the results should be output versus the input frequency, the output frequency, or the absolute value of the input frequency. Default is `out` for logarithmic frequency sweeps and `absin` otherwise. Possible values are `absin`, `in` or `out`.
17 `save`	Signals to output. Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.
18 `nestlvl`	Levels of subcircuits to output.

Convergence parameters

19 `tolerance=1e-9`	Relative tolerance for linear solver.
20 `gear_order=2`	Gear order used for small-signal integration.

21 `solver=turbo`	Solver type. Possible values are `std` or `turbo`.
22 `oscsolver=turbo`	Oscillator solver type. Possible values are `std` or `turbo`.

Annotation parameters

23 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
24 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
25 `title`	Analysis title.

The variable of interest at the output can be voltage or current, and its frequency is not constrained by the period of the large periodic solution. While sweeping the selected output frequency, you can select the periodic small-signal input frequencies of interest by setting either the `maxsideband` or the `sidebands` parameters. For a given set of n integer numbers representing the sidebands K1, K2, ... Kn, the input signal frequency at each sideband is computed as f(in)= f(out) + Ki * fund(pss), where f(out) represent the (possibly swept) output signal frequency, and fund(pss) represents the fundamental frequency used in the corresponding `pss` analysis. Thus, when analyzing a down-converting mixer, and sweeping the IF output frequency, Ki= +1 for the RF input represents the first upper-sideband, while Ki= -1 for the RF input represents the first lower-sideband. By setting the `maxsideband` value to Kmax, all 2 * Kmax + 1 sidebands from -Kmax to +Kmax are be selected.

The number of requested sidebands does not change substantially the simulation time. However, the `maxacfreq` of the corresponding `pss` analysis should be set to guarantee that | max{f(in)} | is less than `maxacfreq`, otherwise the computed solution might be contaminated by aliasing effects. The `pxf` simulation is not executed for | f(out) | greater than `maxacfreq`. Diagnostic messages are printed for those extreme cases, indicating how `maxacfreq` should be set in the `pss` analysis. In the majority of the simulations, however, this is not an issue, because `maxacfreq` is never allowed to be smaller than 40x the `pss` fundamental.

With PXF the frequency of the stimulus and of the response are usually different (this is an important way in which `pxf` differs from XF). The `freqaxis` parameter is used to specify whether the results should be output versus the input frequency (`in`), the output frequency (`out`), or the absolute value of the input frequency (`absin`).

You can specify the output with a pair of nodes or a probe component. Any component with two or more terminals can be a voltage probe. When there are more than two terminals, they are grouped in pairs; and you use the `portv` parameter to select the appropriate pair of terminals. Alternatively, you can simply specify a voltage to be the output by giving a pair of nodes on the `pxf` analysis statement.

Any component that naturally computes current as an internal variable can be a current probe. If the probe component computes more than one current, you use the `porti` parameter to select the appropriate current. It is an error to specify both `portv` and `porti`. If neither is specified, the probe component provides a reasonable default.

The `stimuli` parameter specifies what is used for the inputs for the transfer functions. There are two choices. `stimuli=sources` indicates that the sources present in the circuit should be used. The `xfmag` parameters provided by the sources may be used to adjust the computed gain to compensate for gains or losses in a test fixture. One can limit the number of sources in hierarchical netlists by using the `save` and `nestlvl` parameters. `stimuli=nodes_and_terminals` indicates that all possible transfer functions should be computed.

This is useful when it is not known in advance which transfer functions are interesting. Transfer functions for nodes are computed assuming that a unit magnitude flow (current) source is connected from the node to ground. Transfer functions for terminals are computed assuming that a unit magnitude value (voltage) source is connected in series with the terminal. By default, the transfer functions from a small set of terminals are computed. If transfer functions from specific terminals are desired, specify the terminals in the save statement. You must use the `:probe` modifier (for example, `Rout:1:probe`) or specify `useprobes=yes` on the options statement. If transfer functions from all terminals are desired, specify `currents=all` and `useprobes=yes` on the options statement.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. Alternatively, you may specify the particular values that the sweep parameter should take using the `values` parameter. If you give both a specific set of values and a set specified using a sweep range, the two sets are merged and collated before being used. All frequencies are in Hertz.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Quasi-Periodic AC Analysis (qpac)

Description

The quasi periodic AC (QPAC) analysis is used to compute transfer functions for circuits that exhibit multitone frequency translation. Such circuits include mixers, switched-capacitor filters, samplers, phase-locked loops, and the like. It is a small-signal analysis like AC analysis, except the circuit is first linearized about a quasiperiodically varying operating point as opposed to a simple DC operating point. Linearizing about a quasiperiodically time-varying operating point allows transfer-functions that include frequency translation, whereas simply linearizing about a DC operating point could not because linear time-invariant circuits do not exhibit frequency translation. Also, the frequency of the sinusoidal stimulus is not constrained by the period of the large periodic solution.

Computing the small-signal response of a quasiperiodically varying circuit is a two step process. First, the small stimulus is ignored and the quasiperiodic steady-state response of the circuit to possibly large periodic stimuli is computed using QPSS analysis. As a normal part of the QPSS analysis, the quasiperiodically time-varying representation of the circuit is computed and saved for later use. The second step is to apply the small stimulus to the periodically varying linear representation to compute the small signal response. This is done using the QPAC analysis.

A QPAC analysis cannot be used alone, it must follow a QPSS analysis. However, any number of quasiperiodic small-signal analyses such as QPAC, QPSP, QPXF, QPNOISE, can follow a QPSS analysis.

Unlike other analyses in Spectre, this analysis can only sweep frequency.

Definition

Name qpac parameter=value ...

Parameters

Sweep interval parameters

1 `start=0`	Start sweep limit.
2 `stop`	Stop sweep limit.
3 `center`	Center of sweep.
4 `span=0`	Sweep limit span.
5 `step`	Step size, linear sweep.
6 `lin=50`	Number of steps, linear sweep.
7 `dec`	Points per decade.
8 `log=50`	Number of steps, log sweep.
9 `values=[...]`	Array of sweep values.
10 `sweeptype`	Specifies if the sweep frequency range is absolute frequency of input or if it is relative to the port harmonics. Possible values are `absolute` or `relative`.
11 `relharmnum=[...]`	Harmonic to which relative frequency sweep should be referenced.

Output parameters

12 `sidevec=[...]`	Array of relevant sidebands for the analysis.
13 `clockmaxharm=0`	An alternative to the `sidevec` array specification, which automatically generates the array: [-clockmaxharm ... 0 ... +clockmaxharms][-maxharms(QPSS)[2]...0...maxharms(QPSS)[2]][...].

14 `freqaxis`	Specifies whether the results should be output versus the input frequency, the output frequency, or the absolute value of the output frequency. Default is `in` for logarithmic frequency sweeps and `absout` otherwise. Possible values are `absout`, `out` or `in`.
15 `save`	Signals to output. Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.
16 `nestlvl`	Levels of subcircuits to output.

17 `outputperiod=0.0 (no output)`
Time-domain output period. The time-domain small-signal response is computed for the period specified, rounded to the nearest integer multiple of the `pss` period.

Convergence parameters

18 `tolerance=1e-9`	Relative tolerance for linear solver.
19 `gear_order=2`	Gear order used for small-signal integration, 1 or 2.
20 `solver=turbo`	Solver type. Possible values are `std` or `turbo`.

Annotation parameters

21 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
22 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
23 `title`	Analysis title.

User can select the set of periodic small-signal output frequencies of interest by setting either the `clockmaxharm` or the `sidevec` parameters. Sidebands are vectors in QPAC. Assume we have one large tone and one moderate tone in QPSS. A sideband K1 will be represented as [K1_1 K1_2]. Corresponding frequency is

K1_1 * fund(large tone of QPSS) + K1_2 * fund(moderate tone of QPSS)

We assume that there are L large and moderate tones in QPSS analysis and a given set of n integer vectors representing the sidebands

K1 = { K1_1,...K1_j..., K1_L} , K2, ... Kn. The output frequency at each sideband is computed as

f(out)= f(in) + SUM_j=1_to_L{Ki_j * fund_j(qpss)},

where f(in) represents the (possibly swept) input frequency, and fund_j(qpss) represents the fundamental frequency used in the corresponding QPSS analysis.Thus, when analyzing a down-converting mixer, while sweeping the RF input frequency, the most relevant sideband for IF output is { -1, 0} . When simulating an up-converting mixer, while sweeping IF input frequency, the most relevant sideband for RF output is { 1, 0}. User would enter `sidevec` as a sequence of integer numbers, separated by spaces. The set of vectors {1 1 0} {1 -1 0} {1 1 1} becomes sidevec=[1 1 0 1 -1 0 1 1 1]. For `clockmaxharm`, only the large tone - first fundamental will be affected by this entry, all the rest - moderate tones - will be limited by `maxharms`, specified for a QPSS analysis. Given maxharms=[k1max k2max ... knmax] in QPSS and clockmaxharm=Kmax all (2*Kmax + 1)*(2*k2max+1)*(2*k3max+1)*...*(2*knmax+1) sidebands are generated.

The number of requested sidebands changes substantially the simulation time. In addition, the `maxacfreq` of the corresponding QPSS analysis should be set to guarantee that | max{f(out)} | is less than `maxacfreq`, otherwise the computed solution might be contaminated by aliasing effects. The QPAC simulation is not executed for | f(in) | greater than `maxacfreq`. Diagnostic messages are printed for those extreme cases, indicating how `maxacfreq` should be set in the QPSS analysis. Usually, however, this is not an issue, because `maxacfreq` is never allowed to be smaller than 40x the QPSS fundamental.

With QPAC the frequency of the stimulus and of the response are usually different (this is an important way in which QPAC differs from AC). The `freqaxis` parameter is used to specify whether the results should be output versus the input frequency (`in`), the output frequency (`out`), or the absolute value of the output frequency (`absout`).

Unlike AC analysis, QPAC analysis can output the time-domain simulation results, by specifying the `outputperiod` parameter.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. Alternatively, you may specify the particular values that the sweep parameter should take using the `values` parameter. If you give both a specific set of values and a set specified using a sweep range, the two sets are merged and collated before being used. All frequencies are in Hertz.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Quasi-Periodic Noise Analysis (qpnoise)

Description

The Quasi-Periodic Noise, or QPNOISE analysis is similar to the conventional noise analysis, except that it includes frequency conversion and intermodulation effects. Hence is it useful for predicting the noise behavior of mixers, switched-capacitor filters, and other periodically or quasi-periodically driven circuits.

QPNOISE analysis linearizes the circuit about the quasi-periodic operating point computed in the prerequisite QPSS analysis. It is the quasiperiodically time-varying nature of the linearized circuit that accounts for the frequency conversion and intermodulation. In addition, the affect of a quasi-periodically time-varying bias point on the noise generated by the various components in the circuit is also included.

The time-average of the noise at the output of the circuit is computed in the form of a spectral density versus frequency. The output of the circuit is specified with either a pair of nodes or a probe component. To specify the output of a circuit with a probe, specify it using the `oprobe` parameter. If the output is voltage (or potential), choose a `resistor` or a `port` as the output probe. If the output is current (or flow), choose a `vsource` or `iprobe` as the output probe.

If the input-referred noise is desired, specify the input source using the `iprobe` parameter. Currently, only a `vsource`, an `isource`, or a `port` may be used as an input probe. If the input source is noisy, as is a `port`, the noise analysis will compute the noise factor (F) and noise figure (NF). To match the IEEE definition of noise figure, the input probe must be a port with no excess noise and its `noisetemp` must be set to 16.85C (290K). In addition, the output load must be a `resistor` or `port` and must be identified as the `oprobe`.

The reference sideband (`refsideband`) specifies which conversion gain is used when computing input-referred noise, noise factor, and noise figure. The reference sideband satisfies:

|f(input)| = |f(out) + refsideband frequency shift|.

The reference sideband option (refsidebandoption) specifies whether to consider the input at the frequency or the input at the individual quasi-periodic sideband specified. Note that Different sidebands can lead to the same frequency.

Sidebands are vectors in QPNOISE. Assume we have one large tone and one moderate tone in QPSS. A sideband Ki will be a vector [Ki_1 Ki_2]. It gives the frequency at

Ki_1 * fund(large tone of QPSS) + Ki_2 * fund(moderate tone of QPSS)

Use `refsideband=[0 0 ...]` when the input and output of the circuit are at the same frequency (such as with amplifiers and filters).

The noise analysis always computes the total noise at the output, which includes contributions from the input source and the output load. The amount of the output noise that is attributable to each noise source in the circuit is also computed and output individually. If the input source is identified (using iprobe) and is a vsource or isourec, the input-referred noise is computed, which includes the noise from the input source itself. Finally, if the input source is identified (using iprobe) and is noisy, as is the case with ports, the noise factor and noise figure are computed. Thus if

No = total output noise

Ns = noise at the output due to the input probe (the source)

Nsi = noise at the output due to the image harmonic at the source

Nso = noise at the output due to harmonics other than input at the source

Nl = noise at the output due to the output probe (the load)

IRN = input referred noise

G = gain of the circuit

F = noise factor

NF = noise figure

Fdsb = double sideband noise factor

NFdsb = double sideband noise figure

Fieee = IEEE single sideband noise factor

NFieee = IEEE single sideband noise figure

then,

IRN = sqrt(No^2 / G^2)

F = (No^2 - Nl^2)/Ns^2

NF = 10*log10(F)

Fdsb = (No^2 - Nl^2)/(Ns^2+Nsi^2)

NFdsb = 10*log10(Fdsb)

Fieee = (No^2 - Nl^2 - Nso)/Ns^2

NFieee = 10*log10(Fieee).

When the results are output, No is named `out`, IRN is named `in`, G is named `gain`, F, NF, Fdsb, NFdsb, Fieee, and NFieee are named F, `NF`, `Fdsb`, `NFdsb`, `Fieee`, and `NFieee` respectively.

The computation of gain and IRN in QPNOISE assumes that the circuit under test is impedance-matched to the input source. This can introduce inaccuracy into the gain and IRN computation.

Unlike other analyses in Spectre, this analysis can only sweep frequency.

Definition

Name [p] [n] qpnoise parameter=value ...

The optional terminals (p and n) specify the output of the circuit. If you do not give the terminals, then you must specify the output with a probe component.

Parameters

Sweep interval parameters

1 `start=0`	Start sweep limit.
2 `stop`	Stop sweep limit.
3 `center`	Center of sweep.
4 `span=0`	Sweep limit span.
5 `step`	Step size, linear sweep.
6 `lin=50`	Number of steps, linear sweep.
7 `dec`	Points per decade.
8 `log=50`	Number of steps, log sweep.
9 `values=[...]`	Array of sweep values.
10 `sweeptype`	Specifies if the sweep frequency range is absolute frequency of input or if it is relative to the port harmonics. Possible values are `absolute` or `relative`.
11 `relharmnum=[...]`	Harmonic to which relative frequency sweep should be referenced.

Probe parameters

12 `oprobe`	Compute total noise at the output defined by this component.
13 `iprobe`	Refer the output noise to this component.
14 `refsideband=[...]`	Conversion gain associated with this sideband is used when computing input-referred noise or noise figure.
15 `refsidebandoption=freq`	Whether to view the sideband as a specification of a frequency or a specification of an individual sideband. Possible values are `freq` or `individual`.

Output parameters

16 `clockmaxharm=7` Maximum clock harmonics range included when computing noise either up-converted or down-converted to the output by the periodic drive signal..

17 `sidevec=[...]` Array of relevant sidebands for the analysis.

18 `save` Signals to output.
Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.

19 `nestlvl` Levels of subcircuits to output.

20 `saveallsidebands=no`
Save noise contributors by sideband.
Possible values are no or `yes`.

Convergence parameters

21 `tolerance=1e-9` Relative tolerance for linear solver.

22 `gear_order=2` Gear order used for small-signal integration, 1 or 2.

23 `solver=turbo` Solver type.
Possible values are `std` or `turbo`.

Annotation parameters

24 `annotate=sweep` Degree of annotation.
Possible values are no, `title`, `sweep`, `status`, or `steps`.

25 `stats=no` Analysis statistics.
Possible values are no or `yes`.

26 `title` Analysis title.

In practice, noise can mix with each of the harmonics of the quasi-periodic drive signal applied in the QPSS analysis and end up at the output frequency. The QPNOISE analysis only includes the noise that mixes with a finite set of harmonics that are specified using the `clockmaxharm` and `sidevec` parameters. Sidebands are vectors in quasi-periodic analyses. For one large tone and one moderate tone in QPSS, a sideband K1 will be represented as [K1_1 K1_2]. Corresponding frequency shift is

K1_1 * fund(large tone of QPSS) + K1_2 * fund(moderate tone of QPSS)

We assume that there are L large and moderate tones in QPSS analysis and a given set of n integer vectors representing the sidebands

K1 = { K1_1,...K1_j..., K1_L} ,

K2, ... Kn.

If Ki represents sideband i, then

f(noise_source) = f(out) + SUM_j=1_to_L{ Ki_j * fund_j(qpss) },

The `clockmaxharm` parameter only affects clock frequency. It can be less or more than `maxharms[1]` in QPSS.Moderate tones are limited by `maxharms` specified in QPSS.If selected sidebands are specified using the `sidevec` parameter, then only those are included in the calculation. Care should be taken when specifying the `sidevec` or `clockmaxharm` QPNOISE and `maxharms` in QPSS. Noise results will be in error if you do not include a sideband that contributes significant noise to the output.

The number of requested sidebands will change substantially the simulation time.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. Alternatively, you may specify the particular values that the sweep parameter should take using the `values` parameter. If you give both a specific set of values and a set specified using a sweep range, the two sets are merged and collated before being used. All frequencies are in Hertz.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

iprobe 13	relharmnum 11	stats 25
lin 6	save 18	step 5

Quasi-Periodic S-Parameter Analysis (qpsp)

Description

The quasi periodic SP (QPSP) analysis is used to compute scattering and noise parameters for n-port circuits that exhibit frequency translation. Such circuits include mixers, switched-capacitor filters, samplers, phase-locked loops, and the like. It is a small-signal analysis like SP analysis, except, as in QPAC and QPXF, the circuit is first linearized about a quasiperiodically varying operating point as opposed to a simple DC operating point. Linearizing about a quasiperiodically time-varying operating point allows the computation of S-parameters between circuit ports that convert signals from one frequency band to another. QPSP can also calculate noise parameters in frequency-converting circuits. QPSP computes noise figure (both single-sideband and double-sideband), input referred noise, equivalent noise parameters, and noise correlation matrices. As in QPNOISE, but unlike SP, the noise features of the QPSP analysis include noise folding effects due to the periodically time-varying nature of the circuit.

Computing the n-port S-parameters and noise parameters of a quasiperiodically varying circuit is a two step process. First, the small stimulus is ignored and the quasiperiodic steady-state response of the circuit to possibly large periodic stimulus is computed using QPSS analysis. As a normal part of the QPSS analysis, the quasiperiodically time-varying representation of the circuit is computed and saved for later use. The second step is applying small-signal excitations to compute the n-port S-parameters and noise parameters. This is done using the QPSP analysis. A QPSP analysis cannot be used alone, it must follow a QPSS analysis. However, any number of periodic small-signal analyses such as QPAC, QPSP, QPXF, QPNOISE, can follow a single QPSS analysis.

Unlike other analyses in Spectre, this analysis can only sweep frequency.

Definition

Name qpsp parameter=value ...

Parameters

Sweep interval parameters

1 `start=0`	Start sweep limit.
2 `stop`	Stop sweep limit.
3 `center`	Center of sweep.
4 `span=0`	Sweep limit span.
5 `step`	Step size, linear sweep.
6 `lin=50`	Number of steps, linear sweep.
7 `dec`	Points per decade.
8 `log=50`	Number of steps, log sweep.
9 `values=[...]`	Array of sweep values.
10 `sweeptype`	Specifies if the sweep frequency range is absolute frequency of input or if it is relative to the port harmonics. Possible values are `absolute` or `relative`.

Port parameters

11 `ports=[...]`	List of active ports. Ports are numbered in the order given. For purposes of noise figure computation, the input is considered port 1 and the output is port 2.
12 `portharmsvec=[...]`	List of harmonics active on specified list of ports. Must have a one-to-one correspondence with the ports vector.

13 `harmsvec=[...]`	List of harmonics combinations, in addition to ones associated with specific ports by portharmsvec, that are active. Call them secondary..

Output parameters

14 `freqaxis`	Specifies whether the results should be output versus the input frequency, the output frequency, or the absolute value of the input frequency. Default is `in`. Possible values are `absin`, `in` or `out`.

Noise parameters

15 `donoise=yes`	Perform noise analysis. If oprobe is specified as a valid port, this is set to yes, and a detailed noise output is generated. Possible values are `no` or `yes`.

Probe parameters

16 `clockmaxharm=7`	Maximum clock harmonics range included when computing noise either up-converted or down-converted to the output by the periodic drive signal..

Convergence parameters

17 `tolerance=1e-9`	Relative tolerance for linear solver.
18 `gear_order=2`	Gear order used for small-signal integration, 1 or 2.
19 `solver=turbo`	Solver type. Possible values are `std` or `turbo`.

Annotation parameters

20 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
21 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
22 `title`	Analysis title.

To specify the QPSP analysis the port and port harmonics combinations must be specified. You can select the ports of interest by setting the port parameter and the set of periodic small-signal output frequencies of interest by setting the `portharmsvec` or the `harmsvec` parameters. Sidebands are vectors in QPSP. Assume we have one large tone and one moderate tone in QPSS. A sideband K1 will be represented as [K1_1 K1_2]. Corresponding frequency is

K1_1 * fund(large tone of QPSS) + K1_2 * fund(moderate tone of QPSS)

We assume that there are L (1 large plus L-1 moderate) tones in QPSS analysis and a given set of n integer vectors representing the sidebands

K1 = { K1_1,...K1_j..., K1_L } , K2, ... Kn.

integer numbers representing the harmonics K_1, K_2, ... K_n, the scattering parameters at each port are computed at the frequencies

f(scattered)= f(rel) + SUM_j=1_to_L{Ki_j * fund_j(qpss)},

where f(rel) represents the relative frequency of a signal incident on a port, f(scattered) represents the frequency to which the relevant scattering parameter represents the conversion, and fund_j(qpss) represents the fundamental frequency used in the corresponding QPSS analysis.

Thus, when analyzing a down-converting mixer, with signal in the upper sideband, and sweeping the RF input frequency, the most relevant harmonic for RF input is Ki= { 1 , 0 } and for IF output Ki= { 0 , 0 }. Hence we can associate K2={ 1 , 0 } with the IF port and K1={ 0 , 0 } with the RF port. S21 will represent the transmission of signal from the RF to IF, and S11 the reflection of signal back to the RF port. If the signal was in the lower sideband, then a choice of K1={ -1 , 0 } would be more appropriate.

Either the `portharmsvec` or the `harmsvec` parameters can be used to specify the harmonics of interest. If `portharmsvec` is given, the harmonics must be in one-to-one correspondence with the ports, with each harmonic associated with a single port. If harmonics are specified in the optional `harmsvec` parameter, then all possible frequency-translating scattering parameters associated with the specified harmonics are computed.

With QPSP the frequency of the input and of the response are usually different (this is an important way in which QPSP differs from SP). Because the QPSP computation involves inputs and outputs at frequencies that are relative to multiple harmonics, the `freqaxis` and `sweeptype` parameters behave somewhat differently in QPSP than in QPAC and QPXF.

The `sweeptype` parameter controls the way the frequencies in the QPSP analysis are swept. Specifying a relative sweep indicates to sweep relative to the analysis harmonics (not the QPSS fundamental) and an absolute sweep is a sweep of the absolute input source frequency. For example, with a QPSS fundamentals of 1000MHz and 900MHz, the portharmsvec set to [0 1 1 -1] to examine a downconverting mixer, sweeptype=relative, and a sweep range of f(rel)=0->50MHz, then S21 would represent the strength of signal transmitted from the input port in the range 900->950MHz to the output port at frequencies 100->150MHz. Using sweeptype=absolute and sweeping the frequency from 900->950MHz would calculate the same quantities, since f(abs)=900->950MHz, and f(rel) = f(abs) - (K1 * fund_1(qpss) + K2 * fund_2(qpss) = 0->50MHz, because K1=0, K2=1 and fund_1(qpss) = 1000MHz, fund_2(qpss) = 900MHz.

Usually it is not necessary to sweep frequency in QPSP over more than one period of large (clock) fundamental in QPSS.

The `freqaxis` parameter is used to specify whether the results should be output versus the scattered frequency at the input port (`in`), the scattered frequency at the output port (`out`), or the absolute value of the frequency swept at the input port (`absin`).

An increase in the number of requested ports and harmonics will increase the simulation time substantially.

To insure accurate results in QPSP, the `maxacfreq` of the corresponding PSS analysis should be set to guarantee that | max{f(scattered)} | is less than `maxacfreq`, otherwise the computed solution might be contaminated by aliasing effects.

QPSP analysis also computes noise figures, equivalent noise sources, and noise parameters. The noise computation, which is skipped only when donoise is set to no, requires additional simulation time. If

No = total output noise at frequency f

Ns = noise at the output due to the input probe (the source)

Nsi = noise at the output due to the image harmonic at the source

Nso = noise at the output due to harmonics other than input at the source

Nl = noise at the output due to the output probe (the load)

IRN = input referred noise

G = gain of the circuit

F = noise factor (single side band)

NF = noise figure (single side band)

Fdsb = double sideband noise factor

NFdsb = double sideband noise figure

Fieee = IEEE single sideband noise factor

NFieee = IEEE single sideband noise figure

then,

IRN = sqrt(No^2 / G^2)

F = (No^2 - Nl^2)/Ns^2

NF = 10*log10(F)

Fdsb = (No^2 - Nl^2)/(Ns^2+Nsi^2)

NFdsb = 10*log10(Fdsb)

Fieee = (No^2 - Nl^2 - Nso)/Ns^2

NFieee = 10*log10(Fieee).

When the results are output, IRN is named `in`, G is named `gain`, F, NF, Fdsb, NFdsb, Fieee, and NFieee are named F, NF, Fdsb, NFdsb, Fieee, and NFieee respectively. Note that the gain computed by QPSP is the voltage gain from the actual circuit input to the circuit output, not the gain from the internal port voltage source to the output.

To insure accurate noise calculations, the `clockmaxharm` parameters must be set to include the relevant noise folding effects. `clockmaxharm` is only relevant to the noise computation features of QPSP.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. Alternatively, you may specify the particular values that the sweep parameter should take using the `values` parameter. If you give both a specific set of values and a set specified using a sweep range, the two sets are merged and collated before being used. All frequencies are in Hertz.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Quasi-Periodic Steady State Analysis (qpss)

Description

Similar to PAC analysis, the periodic distortion (PDISTO) analysis calculates responses of a circuit that exhibits frequency translations. However, instead of having small signal linear behavior, it models the response as having components of a few harmonics of input signal frequencies. This allows computing responses to moderately large input signals. An example is intermodulation distortion with two moderate input signals. PDISTO treats one particular input signal (usually the one that causes the most nonlinearity or the largest response) as the large signal, and the others as moderate signals.

An initial transient analysis is carried out by first suppressing all moderate input signals. Then we run a number of (at least 2) stable iterations with all signals activated, which is followed by shooting Newton method. PDISTO employs the Mixed Frequency Time (MFT) algorithm extended to multiple fundamental frequencies. For details of MFT algorithm, please see Steady-State Methods for Simulating Analog and Microwave Circuits, by K. S. Kundert, J. K. White, and A. Sangiovanni-Vincentelli, Kluwer, Boston, 1990.

Like PSS, PDISTO uses shooting Newton method as its backbone. However, instead of doing a single transient integration, each Newton iteration does a number of transient integrations of one large signal period.

Each of the integrations differs by a phase-shift in each moderate input signal. The number of integrations is determined by the numbers of harmonics of moderate fundamentals specified by the user. Given maxharms=[k1 k2 ... kn], the total number of integrations is (2*k2+1)*(2*k3+1)*...*(2*kn+1). As one consequence, the efficiency of the algorithm depends significantly on the number of harmonics required to model the responses of moderate fundamentals. As another consequence, the number of harmonics of the large fundamental does not significantly affect the efficiency of the shooting algorithm. The boundary conditions of a shooting interval are such that the time domain integrations are consistent with a frequency domain transformation with a shift of one large signal period.

PDISTO inherits a majority of PSS parameters. A few new parameters are added. The most important ones are `funds` and `maxharms`. They replace PSS parameters, `fund`(or `period`) and `harms`, respectively. The `funds` parameter accepts a list of names of fundamentals that are present in the sources. These names are specified in the sources by parameter `fundname`. The first fundamental is considered as the large signal. A few heuristics can be used for picking the large fundamental.

(1) Pick the one which is not a sinusoidal.

(2) Pick the one which causes the most nonlinearity.

(3) Pick the one which causes the largest response.

The `maxharms` parameter accepts a list of numbers of harmonics that are required to sufficiently model responses due to different fundamentals.

Definition

Name qpss parameter=value ...

Parameters

Distortion fundamental parameters

1	`funds=[...]`	Array of fundamental frequency names for fundamentals to use in analysis.
2	`maxharms=[...]`	Array of number of harmonics of each fundamental to consider for each fundamental.

Simulation interval parameters

3	`tstab=0.0 s`	Extra stabilization time after the onset of periodicity for independent sources.
4	`tstart=0.0 s`	Initial transient analysis start time.

Time-step parameters

5 `maxstep (s)` — Maximum time step. Default derived from `errpreset`.

6 `step=0.001 period s`
Minimum time step that would be used solely to maintain the aesthetics of the results.

Initial-condition parameters

7 `ic=all` — What should be used to set initial condition.
Possible values are `dc`, `node`, `dev`, or `all`.

8 `skipdc=no` — If yes, there will be no dc analysis for transient.
Possible values are `no` or `yes`.

9 `readic` — File that contains initial condition.

Convergence parameters

10 `readns` — File that contains estimate of initial transient solution.

11 `cmin=0 F` — Minimum capacitance from each node to ground.

Output parameters

12 `outputtype=time for PSS, freq for PDISTO`
Output type.
Possible values are `all`, `time` or `freq`.

13 `save` — Signals to output.
Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.

14 `nestlvl` — Levels of subcircuits to output.

15 `oppoint=no` — Should operating point information be computed for initial timestep, and if so, where should it be sent.
Possible values are `no`, `screen`, `logfile`, or `rawfile`.

16 `skipstart=starttime s`
The time to start skipping output data.

17 `skipstop=stoptime s`
The time to stop skipping output data.

18 `skipcount` Save only one of every skipcount points.

19 `strobeperiod (s)` The output strobe interval (in seconds of transient time).

20 `strobedelay=0 s` The delay (phase shift) between the skipstart time and the first strobe point.

21 `compression=no` Do data compression on output.
Possible values are `no` or `yes`.

22 `saveinit=no` If set, the waveforms for the initial transient before steady state are saved.
Possible values are `no` or `yes`.

State-file parameters

23 `write` File to which initial transient solution (before steady-state) is to be written.

24 `writefinal` File to which final transient solution in steady-state is to be written.

25 `swapfile` Temporary file that holds steady-state information. Tells Spectre to use a regular file rather than virtual memory to hold the periodic operating point. Use this option if Spectre complains about not having enough memory to complete this analysis.

Integration method parameters

26 `method` Integration method. Default derived from `errpreset`.
Possible values are `euler`, `trap`, `traponly`, `gear2`, or `gear2only`.

Accuracy parameters

27 `errpreset=moderate`
Selects a reasonable collection of parameter settings.
Possible values are `liberal`, `moderate` or `conservative`.

28 relref — Reference used for the relative convergence criteria. Default derived from errpreset.
Possible values are pointlocal, alllocal, sigglobal, or allglobal.

29 lteratio — Ratio used to compute LTE tolerances from Newton tolerance. Default derived from errpreset.

30 steadyratio — Ratio used to compute steady state tolerances from LTE tolerance. Default derived from errpreset.

31 maxperiods=20 for driven PSS, 50 for autonomous PSS
Maximum number of simulated periods to reach steady-state.

32 tolerance=1e-4 — Relative tolerance for linear solver.

33 finitediff — Options for finite difference method refinement after quasi-periodic shooting method. finitediff is changed from no to samegrid automatically when readqpss and writeqpss are used to re-use QPSS results.
Possible values are no, yes or refine.

Annotation parameters

34 stats=no — Analysis statistics.
Possible values are no or yes.

35 annotate=sweep — Degree of annotation.
Possible values are no, title, sweep, status, or steps.

36 title — Analysis title.

Newton parameters

37 maxiters=5 — Maximum number of iterations per time step.

38 restart=yes — Do not use previous DC solution as initial guess.
Possible values are no or yes.

Circuit age

39 `circuitage (Years)`	Stress Time. Age of the circuit used to simulate hot-electron degradation of MOSFET and BSIM circuits.
40 `writeqpss`	File to which final quasi-periodic steady-state solution is to be written. Small signal analyses such as qpac, qpxf and qpnoise can read in the steady-state solution from this file directly instead of running the qpss analysis again.
41 `readqpss`	File from which final quasi-periodic steady-state solution is to be read. Small signal analyses such as qpac, qpxf and qpnoise can read in the steady-state solution from this file directly instead of running the qpss analysis again.

Most of PDISTO analysis parameters are inherited from PSS analysis, and their meanings remain essentially unchanged. Two new important parameters are `funds` and `maxharms`. They replace and extend the role of `fund` and `harms` parameters of PSS analysis. One important difference is that `funds` accepts a list of fundamental names instead of actual frequencies. The frequencies associated with fundamentals are figured out automatically by the simulator. An important feature is that each input signal can be a composition of more than one sources. However, these sources must have the same fundamental name. For each fundamental name, its fundamental frequency is the greatest common factor of all frequencies associated with the name. Missing or not listing all fundamental names using the parameter `funds` will result in an amputation of the current simulation. However if `maxharms` is not given, a warning message will be issued, and the number of harmonics is defaulted to 1 for each of the fundamentals.

The role of some PSS parameters is extended. The parameter `maxperiods` that controls the maximum number of shooting iterations for PSS analysis also controls the number of the maximum number of shooting iterations for PDISTO analysis.

The tstab parameter controls both the length of the initial transient integration with only the clock tone activated and the number of stable iterations with moderate tones activated. The stable iterations are run before Newton iterations.

The `errpreset` parameter lets you adjust the simulator parameters to fit your needs quickly. In most cases, it should also be the only parameter you need to adjust. If you want a fast simulation with reasonable accuracy, you may set errpreset to liberal. If have some concern for accuracy, you may set errpreset to moderate. If accuracy is your main interest, you may set errpreset to conservative.

If users dont specify steadyratio, it is always 1.0, not affected by errpreset. The effect of `errpreset` on other parameters is shown in the following table.

Parameter defaults as a function of errpreset
check siggloaal??

errpreset	reltol	relref	method	lteratio	maxstep
liberal	1e-3	sigglobal	gear2only	3.5	clock period/80
moderate	1e-4	siggloaal	gear2only	3.5	clock period/100
conservative	1e-5	sigglobal	gear2only	*	clock period/200

* : If user specified `reltol` <= 1e-5*10.0/3.5, `lteratio`=3.5. Otherwise `lteratio`=10.0.

The value of `reltol` can only decrease from its value in the options statement, when reltol in options statement is larger. Spectre sets the value of `maxstep` so that it is no larger than the value given in the table. Except for `reltol` and `maxstep`, `errpreset` does not change the value of any parameters you have explicitly set. The actual values used for the QPSS analysis are given in the log file.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Quasi-Periodic Transfer Function Analysis (qpxf)

Description

A conventional transfer function analysis computes the transfer function from every source in the circuit to a single output. It differs from a conventional AC analysis in that the AC analysis computes the response from a single stimulus to every node in the circuit. The difference between QPAC and QPXF analysis are similar. The Quasi Periodic Transfer Function or QPXF analysis computes the transfer functions from any source at any frequency to a single output at a single frequency. Thus, like QPAC analysis, QPXF analysis includes frequency conversion effects.

The QPXF analysis directly computes such useful quantities as conversion efficiency (transfer function from input to output at desired frequency), image and sideband rejection (input to output at undesired frequency), and LO feed-through and power supply rejection (undesired input to output at all frequencies).

As with a QPAC, QPSP, and QPNOISE analyses, a QPXF analysis must follow a QPSS analysis.

Unlike other analyses in Spectre, this analysis can only sweep frequency.

Definition

Name [p] [n] qpxf parameter=value ...

The optional terminals (p and n) specify the output of the circuit. If you do not give the terminals, then you must specify the output with a probe component.

Parameters

Sweep interval parameters

1	`start=0`	Start sweep limit.
2	`stop`	Stop sweep limit.
3	`center`	Center of sweep.
4	`span=0`	Sweep limit span.

5 `step`	Step size, linear sweep.
6 `lin=50`	Number of steps, linear sweep.
7 `dec`	Points per decade.
8 `log=50`	Number of steps, log sweep.
9 `values=[...]`	Array of sweep values.
10 `sweeptype`	Specifies if the sweep frequency range is absolute frequency of input or if it is relative to the port harmonics. Possible values are `absolute` or `relative`.
11 `relharmnum=[...]`	Harmonic to which relative frequency sweep should be referenced.

Probe parameters

12 `probe`	Compute every transfer function to this probe component.

Output parameters

13 `stimuli=sources`	Stimuli used for xf analysis. Possible values are `sources` or `nodes_and_terminals`.
14 `sidevec=[...]`	Array of relevant sidebands for the analysis.
15 `clockmaxharm=0`	An alternative to the `sidevec` array specification, which automatically generates the array: [-clockmaxharm ... 0 ... +clockmaxharms][-maxharms(QPSS)[2]...0...maxharms(QPSS)[2]][...].
16 `freqaxis`	Specifies whether the results should be output versus the input frequency, the output frequency, or the absolute value of the input frequency. Default is `out` for logarithmic frequency sweeps and `absin` otherwise. Possible values are `absin`, `in` or `out`.
17 `save`	Signals to output. Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.

18 `nestlvl` Levels of subcircuits to output.

Convergence parameters

19 `tolerance=1e-9` Relative tolerance for linear solver.

20 `gear_order=2` Gear order used for small-signal integration, 1 or 2.

21 `solver=turbo` Solver type.
Possible values are `std` or `turbo`.

Annotation parameters

22 `annotate=sweep` Degree of annotation.
Possible values are `no`, `title`, `sweep`, `status`, or `steps`.

23 `stats=no` Analysis statistics.
Possible values are `no` or `yes`.

24 `title` Analysis title.

The variable of interest at the output can be voltage or current, and its frequency is not constrained by the period of the large periodic solution. While sweeping the selected output frequency, you can select the periodic small-signal input frequencies of interest by setting either the `clockmaxharm` or the `sidevec` parameters. Sidebands are vectors in QPXF. Assume we have one large tone and one moderate tone in QPSS. A sideband K1 will be represented as [K1_1 K1_2]. Corresponding frequency is

K1_1 * fund(large tone of QPSS) + K1_2 * fund(moderate tone of QPSS)

We assume that there are L (1 large plus L-1 moderate) tones in QPSS analysis and a given set of n integer vectors representing the sidebands

K1 = { K1_1,...K1_j..., K1_L} , K2, ... Kn.

The input signal frequency at each sideband is computed as

f(in)= f(out) + SUM_j=1_to_L{Ki_j * fund_j(qpss)},

where f(out) represent the (possibly swept) output signal frequency, and fund_j(pss) represents the fundamental frequency used in the corresponding QPSS analysis. Thus, when analyzing a down-converting mixer, and sweeping the IF output frequency, Ki= {1, 0} for the RF input represents the first upper-sideband, while Ki= {-1, 0} for the RF input represents the first lower-sideband.

User would enter `sidevec` as a sequence of integer numbers, separated by spaces. The set of vectors {1 1 0} {1 -1 0} {1 1 1} becomes sidevec=[1 1 0 1 -1 0 1 1 1]. For `clockmaxharm`, only the large tone - first fundamental will be affected by this entry, all the rest - moderate tones - will be limited by `maxharms`, specified for a QPSS analysis. Given maxharms=[k1max k2max ... knmax] in QPSS and clockmaxharm=Kmax all (2*Kmax + 1)*(2*k2max+1)*(2*k3max+1)*...*(2*knmax+1) sidebands are generated.

The number of requested sidebands changes substantially the simulation time. In addition, the `maxacfreq` of the corresponding QPSS analysis should be set to guarantee that | max{f(in)} | is less than `maxacfreq`, otherwise the computed solution might be contaminated by aliasing effects. The QPXF simulation is not executed for | f(out) | greater than `maxacfreq`. Diagnostic messages are printed for those extreme cases, indicating how `maxacfreq` should be set in the QPSS analysis. In the majority of the simulations, however, this is not an issue, because `maxacfreq` is never allowed to be smaller than 40x the QPSS fundamental.

With QPXF the frequency of the stimulus and of the response are usually different (this is an important way in which QPXF differs from XF). The `freqaxis` parameter is used to specify whether the results should be output versus the input frequency (`in`), the output frequency (`out`), or the absolute value of the input frequency (`absin`).

You can specify the output with a pair of nodes or a probe component. Any component with two or more terminals can be a voltage probe. When there are more than two terminals, they are grouped in pairs; and you use the `portv` parameter to select the appropriate pair of terminals. Alternatively, you can simply specify a voltage to be the output by giving a pair of nodes on the QPXF analysis statement.

Any component that naturally computes current as an internal variable can be a current probe. If the probe component computes more than one current, you use the `porti` parameter to select the appropriate current. It is an error to specify both `portv` and `porti`. If neither is specified, the probe component provides a reasonable default.

The `stimuli` parameter specifies what is used for the inputs for the transfer functions. There are two choices. `stimuli=sources` indicates that the sources present in the circuit should be used. The `xfmag` parameters provided by the sources may be used to adjust the computed gain to compensate for gains or losses in a test fixture. One can limit the number of sources in hierarchical netlists by using the `save` and `nestlvl` parameters. `stimuli=nodes_and_terminals` indicates that all possible transfer functions should be computed.

This is useful when it is not known in advance which transfer functions are interesting. Transfer functions for nodes are computed assuming that a unit magnitude flow (current) source is connected from the node to ground. Transfer functions for terminals are computed assuming that a unit magnitude value (voltage) source is connected in series with the terminal. By default, the transfer functions from a small set of terminals are computed. If transfer functions from specific terminals are desired, specify the terminals in the save statement. You must use the `:probe` modifier (for example, `Rout:1:probe`) or specify `useprobes=yes` on the options statement. If transfer functions from all terminals are desired, specify `currents=all` and `useprobes=yes` on the options statement.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. Alternatively, you may specify the particular values that the sweep parameter should take using the `values` parameter. If you give both a specific set of values and a set specified using a sweep range, the two sets are merged and collated before being used. All frequencies are in Hertz.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

annotate 22	lin 6	sidevec 14	stimuli 13
center 3	log 8	solver 21	stop 2
clockmaxharm 15	nestlvl 18	span 4	sweeptype 10
dec 7	probe 12	start 1	title 24
freqaxis 16	relharmnum 11	stats 23	tolerance 19
gear_order 20	save 17	step 5	values 9

Deferred Set Options (set)

Description

The deferred set options statement sets or changes various program control options. You can set the options in any order and, once set, the options retain their value until reset. The set statement is queued with all analyses and is executed sequentially (The changes made to these options are deferred until the statement setting them is encountered). To set `temp`, `tnom`, `scalem`, or `scale`, use the alter statement. For further options, see individual analyses.

Definition

Name set parameter=value ...

Parameters

Tolerance parameters

1 `reltol=0.001` Relative convergence criterion.

2 `vabstol=1e-06 V` Voltage absolute tolerance convergence criterion.

3 `iabstol=1e-12 A` Current absolute tolerance convergence criterion.

Temperature parameters

4 `tempeffects=all` Temperature effect selector.
Possible values are `vt`, `tc` or `all`.

Convergence parameters

5 `homotopy=all` Method used when no convergence on initial attempt of DC analysis.
Possible values are `none`, `gmin`, `source`, `dptran`, `ptran`, `arclength`, or `all`.

6 `limit=dev` Limiting algorithms to aid DC convergence.
Possible values are `delta`, `log` or `dev`.

Component parameters

7 `compatible=spectre`
Encourage device equations to be compatible with a foreign simulator. This option does not affect input syntax.
Possible values are `spectre`, `spice2`, `spice3`, `cdsspice`, `hspice`, or `spiceplus`.

8 approx=no — Use approximate models. Difference between approximate and exact models is generally very small.
Possible values are no or yes.

Error-checking parameters

9 diagnose=no — Print additional information that might help diagnose accuracy and convergence problems.
Possible values are no or yes.

10 opptcheck=yes — Check operating point parameters against soft limits.
Possible values are no or yes.

Resistance parameters

11 gmin=1e-12 S — Minimum conductance across each nonlinear device.

12 gmin_check=max_v_only
Specifies that effect of gmin should be reported if significant.
Possible values are no, max_v_only, max_only, or all.

13 rforce=1 Ω — Resistance used when forcing nodesets and node-based initial conditions.

Quantity parameters

14 quantities=no — Print quantities.
Possible values are no, min or full.

Annotation parameters

15 narrate=yes — Narrate the simulation.
Possible values are no or yes.

16 debug=no — Give debugging messages.
Possible values are no or yes.

17 info=yes — Give informational messages.
Possible values are no or yes.

18 `note=yes`	Give notice messages. Possible values are no or `yes`.
19 `maxnotes=5`	Maximum number of times any notice will be issued per analysis.
20 `warn=yes`	Give warning messages. Possible values are no or `yes`.
21 `maxwarns=5`	Maximum number of times any warning message will be issued per analysis.
22 `error=yes`	Give error messages. Possible values are no or `yes`.
23 `digits=5`	Number of digits used when printing numbers.
24 `notation=eng`	When printing real numbers to the screen, what notation should be used. Possible values are `eng`, `sci` or `float`.
25 `annotate=no`	Degree of annotation. Possible values are no or `title`.

Matrix parameters

26 `pivotdc=no`	Use numeric pivoting on every iteration of DC analysis. Possible values are no or `yes`.
27 `pivrel=0.001`	Relative pivot threshold.
28 `pivabs=0`	Absolute pivot threshold.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Shell Command (shell)

Description

The shell analysis passes a command to the operating system command interpreter given in the SHELL environment variable. The command behaves as if it were typed into the command interpreter, except that any %X codes in the command are expanded first.

The default action of the shell analysis is to terminate the simulation.

Definition

Name shell parameter=value ...

Parameters

1 `cmd="kill %P"`	Shell command.
2 `iferror=quit`	What to do if command returns nonzero error status. Possible values are `quit` or `continue`.
3 `annotate`	Degree of annotation. Possible values are no, `title` or `yes`.

S-Parameter Analysis (sp)

Description

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 sequentially, and a linear small-signal analysis is performed. Spectre 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.

Spectre can perform the analysis while sweeping a parameter. The parameter can be frequency, temperature, component instance parameter, component model parameter, or netlist parameter. If changing a parameter affects the DC operating point, 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.

Definition

Name sp parameter=value ...

Parameters

1 `prevoppoint=no`	Use operating point computed on the previous analysis. Possible values are `no` or `yes`.

Sweep interval parameters

2 `start=0`	Start sweep limit.
3 `stop`	Stop sweep limit.
4 `center`	Center of sweep.
5 `span=0`	Sweep limit span.
6 `step`	Step size, linear sweep.
7 `lin=50`	Number of steps, linear sweep.
8 `dec`	Points per decade.
9 `log=50`	Number of steps, log sweep.
10 `values=[...]`	Array of sweep values.

Sweep variable parameters

11 `dev`	Device instance whose parameter value is to be swept.
12 `mod`	Model whose parameter value is to be swept.
13 `param`	Name of parameter to sweep.
14 `freq (Hz)`	Frequency when parameter other than frequency is being swept.

Port parameters

15 `ports=[...]`	List of active ports. Ports are numbered in the order given.

State-file parameters

16 `readns`	File that contains estimate of DC solution (nodeset).

Output parameters

17 `file`	S-parameters output file name.
18 `oppoint=no`	Should operating point information be computed, and if so, where should it be sent. Possible values are `no`, `screen`, `logfile`, or `rawfile`.

Noise parameters

19 `donoise=no`	Perform noise analysis. If oprobe is specified as a valid port, this is set to yes, and a detailed noise output is generated. Possible values are `no` or `yes`.
20 `oprobe`	Compute total noise at the output defined by this component.
21 `iprobe`	Input probe. Refer the output noise to this component.

Convergence parameters

22 `restart=yes`	Do not use previous DC solution as initial guess. Possible values are `no` or `yes`.

Annotation parameters

23 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
24 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
25 `title`	Analysis title.

If the list of active ports is specified with the `ports` parameter, then the ports are numbered sequentially from 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. If `donoise=yes` is specified, then the noise correlation matrix is computed. If in addition, the output is specified using `oprobe,` the amount that each noise source contributes to the output is computed. Finally, if an input is also specified (using `iprobe`), the two-port noise parameters are computed (F, Fmin, NF, NFmin, Gopt, Bopt, and Rn).

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. However, if a previous analysis computed an operating point, you can set `prevoppoint=yes` to avoid recomputing it. For example, if you use this option when the previous analysis was a transient analysis, the operating point is the state of the circuit on the final time point.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Stability Analysis (stb)

Description

The STB analysis linearizes the circuit about the DC operating point and computes the loop gain, gain and phase margins (if the sweep variable is frequency), for a feedback loop or a gain device.

Spectre can perform the analysis while sweeping a parameter. The parameter can be frequency, temperature, component instance parameter, component model parameter, or netlist parameter. If changing a parameter affects the DC operating point, 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.

Definition

Name stb parameter=value ...

Parameters

1	`prevoppoint=no`	Use operating point computed on the previous analysis. Possible values are `no` or `yes`.

Sweep interval parameters

2	`start=0`	Start sweep limit.
3	`stop`	Stop sweep limit.
4	`center`	Center of sweep.
5	`span=0`	Sweep limit span.
6	`step`	Step size, linear sweep.

7 `lin=50`	Number of steps, linear sweep.
8 `dec`	Points per decade.
9 `log=50`	Number of steps, log sweep.
10 `values=[...]`	Array of sweep values.

Sweep variable parameters

11 `dev`	Device instance whose parameter value is to be swept.
12 `mod`	Model whose parameter value is to be swept.
13 `param`	Name of parameter to sweep.
14 `freq (Hz)`	Frequency when parameter other than frequency is being swept.

Probe parameters

15 `probe`	Probe instance around which the loop gain is calculated.

State-file parameters

16 `readns`	File that contains estimate of DC solution (nodeset).

Output parameters

17 `save`	Signals to output. Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.
18 `nestlvl`	Levels of subcircuits to output.
19 `oppoint=no`	Should operating point information be computed, and if so, where should it be sent. Possible values are no, `screen`, `logfile`, or `rawfile`.

Convergence parameters

20 `restart=yes`	Do not use previous DC solution as initial guess. Possible values are `no` or `yes`.

Annotation parameters

21 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
22 `stats=no`	Analysis statistics. Possible values are `no` or `yes`.
23 `title`	Analysis title.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. However, if a previous analysis computed an operating point, you can set `prevoppoint=yes` to avoid recomputing it. For example, if you use this option when the previous analysis was a transient analysis, the operating point is the state of the circuit on the final time point.

Loop based and Device Based Algorithms

Two algorithms--the loop based and the device based, are available for small-signal stability analysis. Both algorithms are based on the calculation of Bodes return ratio. Loop gain waveform, gain margin, and phase margin are the analysis output.

The `probe` parameter must be specified to perform stability analysis. When it points to a current probe or voltage source instance, the loop based algorithm will be invoked; when it points to a supported active device instance, the device based algorithm will be invoked.

Loop Based Algorithm

The loop based algorithm calculates the true loop gain that consists of normal loop gain and reverse loop gain. The loop based algorithm requires the `probe` being placed on the feedback loop to identify and characterize the particular loop of interest. The introduction of the probe component should not change any of the circuit characteristics.

The loop based algorithm provides accurate stability information for single loop circuits, and multiloop circuits in which a `probe` component can be placed on a critical wire to break all loops. For a general multiloop circuit, such a critical wire may not be available. The loop based algorithm can only be performed on individual feedback loops to ensure they are stable. Although the stability of all feedback loops is only a necessary condition for the whole circuit to be stable, the multiloop circuit tends to be stable if all individual loops are associated with reasonable stability margins.

Device Based Algorithm

The device based algorithm calculates the loop gain around a particular active device. This algorithm is often applied to assess the stability of circuit design in which local feedback loops cannot be neglected; the loop based algorithm cannot be performed for these applications since the local feedback loops are inside the devices, they are not accessible from the schematic level or netlist level to insert the `probe` component.

With the `probe` parameter points to a particular active device, the dominant controlled source in the device will be nulled during the analysis. The dominant controlled source is defined as by nulling this source renders the active device to be passive. The device based algorithm produces accurate stability information for a circuit in which a critical active device can be identified such that nulling the dominant gain source of this device renders the whole network to be passive.

Stability Analysis of Differential Feedback Circuits

A balanced fully differential feedback circuit is illustrated below:

check???

```
|---------[ ZF ]---------|

|                  |

|                  X1
```

```
             | |-----------------| |
             | |                 | |
... ---[ ZS ]---|--|in+       out- |--|---[ ZL ]--- ...
               |     OPAMP       |
... ---[ ZS ]---|--|in-       out+ |--|---[ ZL ]--- ...
             | |                 | |
             | |-----------------| |
             |                   X2
             |                   |
             |---------[ ZF ]---------|
```

The feedback loops are broken at X1 and X2, with x1in and x2in being the input side nodes, x1out and x2out being the output side nodes. The following subcircuit connects these four nodes togather:

```
subckt diffprobe x1in x2in x1out x2out
    ibranch inout x1out iprobe
    vinj inout x1in iprobe
    evinj x2in x2out x1in x1out vcvs gain=0
    fiinj 0 x2out pcccs probes=[ibranch vinj] coeffs=[0 1 1] gain=0
ends diffprobe
```

Let `diffprobe_inst` be the instance of subcircuit `diffprobe`, the following analysis measures the differential-mode loop gain:

```
DMalterv alter dev=diffprobe_inst.evinj param=gain value=-1
DMalteri alter dev=diffprobe_inst.fiinj param=gain value=-1
DMloopgain stb probe=diffprobe_inst.vinj
```

and the following analysis measures the common-mode loop gain:

```
CMalterv alter dev=diffprobe_inst.evinj param=gain value=1
CMalteri alter dev=diffprobe_inst.fiinj param=gain value=1
CMloopgain stb probe=diffprobe_inst.vinj
```

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

Sweep Analysis (sweep)

Description

The `sweep` analysis sweeps a parameter executing the list of analyses (or multiple analyses) for each value of the parameter. The swept parameter can be circuit temperature, a device instance parameter, a device model parameter, a netlist parameter, or a subcircuit parameter for a particular subcircuit instance.

A set of parameters can be swept simultaneously, using the `paramset` parameter. The other sweep interval or variable parameters cannot be specified with the `paramset` parameter. Do `spectre -h paramset` for information on defining a paramset.

Within a sweep statement, 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.

You can sweep the circuit temperature by giving the parameter name as `param=temp` with no `dev`, `mod`, or `sub` parameter. You can sweep a top-level netlist parameter by giving the parameter name with no `dev`, `mod`, or `sub` parameter. You can sweep a subcircuit parameter for a particular subcircuit instance by specifying the subcircuit instance name with the `sub` parameter and the subcircuit parameter name with the `param` parameter. The same can be done using `dev` for the device instance name or `mod` for the device model name.

After the analysis has completed, the modified parameter returns to its original value.

Definition

Name sweep parameter=value ...

Parameters

Sweep interval parameters

1 `start=0`	Start sweep limit.
2 `stop`	Stop sweep limit.
3 `center`	Center of sweep.
4 `span=0`	Sweep limit span.
5 `step`	Step size, linear sweep.
6 `lin=50`	Number of steps, linear sweep.
7 `dec`	Points per decade.
8 `log=50`	Number of steps, log sweep.
9 `values=[...]`	Array of sweep values.

Sweep variable parameters

10 `dev`	Device instance whose parameter value is to be swept.
11 `sub`	Subcircuit instance whose parameter value is to be swept.
12 `mod`	Model whose parameter value is to be swept.
13 `param`	Name of parameter to sweep.
14 `paramset`	Name of parameter set to sweep.

Annotation parameters

`15 annotate=sweep`	Degree of annotation. Possible values are `no`, `title` or `sweep`.
`16 title`	Analysis title.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, or `dec`) and determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, the sweep is linear when the ratio of the stop-to-start values is less than 10 and logarithmic when this ratio is 10 or greater.

Example

```
swp sweep param=temp values=[-50 0 50 100 125] {
         oppoint dc oppoint=logfile
}
```

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

annotate 15	lin 6	paramset 14	stop 2
center 3	log 8	span 4	sub 11
dec 7	mod 12	start 1	title 16
dev 10	param 13	step 5	values 9

Time-Domain Reflectometer Analysis (tdr)

Description

The time-domain reflectometer analysis linearizes the circuit about the DC operating point and computes the reflection coefficients versus time, looking from the active ports into the circuit.

Definition

Name tdr parameter=value ...

Parameters

1 `stop`	Stop time.
2 `settling=stop`	Time required for circuit to settle.
3 `start=-0.1 stop`	Time output waveforms begin.
4 `smoothing=2`	Window smoothing parameter (useful range is 0 to 15).
5 `vel=1`	Propagation velocity of medium normalized to c.
6 `points=64`	Number of time points.
7 `ports=[...]`	List of active ports. If not given, all ports are used.
8 `readns`	File that contains estimate of DC solution (nodeset).
9 `restart=yes`	Do not use previous DC solution as initial guess. Possible values are `no` or `yes`.
10 `annotate=sweep`	Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, or `steps`.
11 `title`	Analysis title.
12 `oppoint=no`	Should operating point information be computed, and if so, where should it be sent. Possible values are `no`, `screen`, `logfile`, or `rawfile`.
13 `prevoppoint=yes`	Use operating point computed on the previous analysis. Possible values are `no` or `yes`.

Such a small-signal analysis begins by linearizing the circuit about an operating point. By default, this analysis computes the operating point, if it is not yet known, or recomputes it, if any significant component or circuit parameter has changed. However, if a previous analysis computed an operating point, you can set `prevoppoint=yes` to avoid recomputing it. For example, if you use this command when the previous analysis was a transient analysis, the operating point is the state of the circuit on the final time point.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

`annotate` 10	`prevoppoint` 13	`smoothing` 4	`vel` 5
`oppoint` 12	`readns` 8	`start` 3	
`points` 6	`restart` 9	`stop` 1	
`ports` 7	`settling` 2	`title` 11	

Transient Analysis (tran)

Description

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

Definition

Name tran parameter=value ...

Parameters

Simulation interval parameters

1 `stop (s)` Stop time.

2 `start=0 s` Start time.

3 `outputstart=start s`
Output is saved only after this time is reached.

4 `autostop=no` — Enable early termination of the analysis, when all measurement expressions have been evaluated.
Possible values are `no` or `yes`.

Time-step parameters

5 `maxstep (s)` — Maximum time step. Default derived from `errpreset`.

6 `step=0.001 (stop-start) s`
Minimum time step used by the simulator solely to maintain the aesthetics of the computed waveforms.

Initial-condition parameters

7 `ic=all` — What should be used to set initial condition.
Possible values are `dc`, `node`, `dev`, or `all`.

8 `skipdc=no` — If yes, there will be no dc analysis for transient.
Possible values are `no`, `yes`, `waveless`, `rampup`, `autodc`, or `sigrampup`.

9 `readic` — File that contains initial condition.

Convergence parameters

10 `readns` — File that contains estimate of initial transient solution.

11 `cmin=0 F` — Minimum capacitance from each node to ground.

State-file parameters

12 `write` — File to which initial transient solution is to be written.

13 `writefinal` — File to which final transient solution is to be written.

14 `ckptperiod` — Checkpoint the analysis periodically using the specified period.

Integration method parameters

15 `method` Integration method. Default derived from `errpreset`.
Possible values are `euler`, `trap`, `traponly`, `gear2`, `gear2only`, or `trapgear2`.

Accuracy parameters

16 `errpreset=moderate`
Selects a reasonable collection of parameter settings.
Possible values are `liberal`, `moderate` or `conservative`.

17 `relref` Reference used for the relative convergence criteria. Default derived from `errpreset`.
Possible values are `pointlocal`, `alllocal`, `sigglobal`, or `allglobal`.

18 `lteratio` Ratio used to compute LTE tolerances from Newton tolerance. Default derived from `errpreset`.

Annotation parameters

19 `stats=no` Analysis statistics.
Possible values are no or `yes`.

20 `annotate=sweep` Degree of annotation.
Possible values are no, `title`, `sweep`, `status`, or `steps`.

21 `title` Analysis title.

Output parameters

22 `save` Signals to output.
Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.

23 `nestlvl` Levels of subcircuits to output.

24 `oppoint=no` Should operating point information be computed for initial timestep, and if so, where should it be sent.
Possible values are no, `screen`, `logfile`, or `rawfile`.

25 `skipstart=starttime s`
The time to start skipping output data.

26 `skipstop=stoptime s`
The time to stop skipping output data.

27 `skipcount` Save only one of every skipcount points.

28 `strobeperiod (s)` The output strobe interval (in seconds of transient time).

29 `strobedelay=0 s` The delay (phase shift) between the skipstart time and the first strobe point.

30 `compression=no` Do data compression on output.
Possible values are `no` or `yes`.

31 `flushpoints` Flush outputs after number of calculated points.

32 `flushtime (s)` Flush outputs after real time has elapsed.

33 `flushofftime (s)` Time to stop flushing outputs.

34 `infoname` Name of info analysis to save operating point.

35 `infotimes=[...] s` Times when operating points should be saved.

Newton parameters

36 `maxiters=5` Maximum number of iterations per time step.

37 `restart=yes` Do not use previous DC solution as initial guess.
Possible values are `no` or `yes`.

Circuit age

38 `circuitage (Years)` Stress Time. Age of the circuit used to simulate hot-electron degradation of MOSFET and BSIM circuits.

You may specify the initial condition for the transient analysis by using the `ic` statement or the `ic` parameter on the capacitors and inductors. If you do not specify the initial condition, the DC solution is used as the initial condition. The `ic` parameter on the transient analysis controls the interaction of various methods of setting the initial conditions. The effects of individual settings are:

`ic=dc`: Any initial condition specifiers are ignored, and the DC solution is used.

`ic=node`: The `ic` statements are used, and the `ic` parameter on the capacitors and inductors are ignored.

`ic=dev`: The `ic` parameters on the capacitors and inductors are used, and the `ic` statements are ignored.

`ic=all`: Both the `ic` statements and the `ic` parameters are used, and the `ic` parameters override the `ic` statements.

If you specify an initial condition file with the `readic` parameter, initial conditions from the file are used, and any `ic` statements are ignored.

Once you specify the initial conditions, Spectre computes the actual initial state of the circuit by performing a DC analysis. During this analysis, Spectre forces the initial conditions on nodes by using a voltage source in series with a resistor whose resistance is `rforce` (see `options`).

With the `ic` statement it is possible to specify an inconsistent initial condition (one that cannot be sustained by the reactive elements). Examples of inconsistent initial conditions include setting the voltage on a node with no path of capacitors to ground or setting the current through a branch that is not an inductor. If you initialize Spectre inconsistently, its solution jumps; that is, it changes instantly at the beginning of the simulation interval. You should avoid such changes if possible because Spectre can have convergence problems while trying to make the jump.

You can skip the DC analysis entirely by using the parameter `skipdc`. If the DC analysis is skipped, the initial solution will be either trivial, or given in the file you specified by the `readic` parameter, or, if the `readic` parameter is not given, the values specified on the ic statements. Device based initial conditions are not used for `skipdc`. Nodes that you do not specify with the `ic` file or `ic` statements will start at zero. You should not use this parameter unless you are generating a nodeset file for circuits that have trouble in the DC solution; it usually takes longer to follow the initial transient spikes that occur when the DC analysis is skipped than it takes to find the real DC solution. The `skipdc` parameter might also cause convergence problems in the transient analysis.

The possible settings of parameter `skipdc` and their meanings are:

`skipdc=no`: Initial solution is calculated using the normal DC analysis (default).

`skipdc=yes`: Initial solution is given in the file specified by the `readic` parameter or the values specified on the ic statements.

`skipdc=wireless`: Same initial solution as skipdc=yes. But, the waveform production in the time-varying independent sources is disabled during the transient analysis. The independent source values are fixed to their initial values (not their DC values).

`skipdc=ramup`: The independent source values start at 0 and ramp up to their initial values in the first 10% of the analysis interval. After that their values remain constant. Zero initial solution is used.

`skipdc=autodc`: Same as `skipdc=wireless` if a nonzero initial condition is specified. Otherwise, same as `skipdc=ramup`.

`skipdc=sigrampup`: The independent source values start at 0 and ramp up to their initial values in the first phase of the simulation. Unlike `skipdc=rampup`, the waveform production in the time-varying independent source is enable after the rampup phase. The rampup simulation is from `start` to time=0 s, and the main simulation is from time=0 s to `stop`. If the start parameter is not specified, the default start time is set to -0.1*`stop`.

Nodesets help find the DC or initial transient solution. You can supply them in the circuit description file with nodeset statements, or in a separate file using the `readns` parameter. When nodesets are given, Spectre computes an initial guess of the solution by performing a DC analysis while forcing the specified values onto nodes by using a voltage source in series with a resistor whose resistance is `rforce`. Spectre then removes these voltage sources and resistors and computes the true solution from this initial guess.

Nodesets have two important uses. First, if a circuit has two or more solutions, nodesets can bias the simulator towards computing the desired one. Second, they are a convergence aid. By estimating the solution of the largest possible number of nodes, you might be able to eliminate a convergence problem or dramatically speed convergence.

When you simulate the same circuit many times, we suggest that you use both the `write` and `readns` parameters and give the same file name to both parameters. The DC analysis then converges quickly even if the circuit has changed somewhat since the last simulation, and the nodeset file is automatically updated.

Nodesets and initial conditions have similar implementation but produce different effects. Initial conditions actually define the solution, whereas nodesets only influence it. When you simulate a circuit with a transient analysis, Spectre forms and solves a set of differential equations. However, differential equations have an infinite number of solutions, and a complete set of initial conditions must be specified in order to identify the desired solution. Any initial conditions you do not specify are computed by the simulator to be consistent. The transient waveforms then start from initial conditions. Nodesets are usually used as a convergence aid and do not affect the final results. However, in a circuit with more than one solution, such as a latch, nodesets bias the simulator towards finding the solution closest to the nodeset values.

The `method` parameter specifies the integration method. The possible settings and their meanings are:

`method=euler`: Backward-Euler is used exclusively.

`method=traponly`: Trapezoidal rule is used almost exclusively.

`method=trap`: Backward-Euler and the trapezoidal rule are used.

`method=gear2only`: Gears second-order backward-difference method is used almost exclusively.

`method=gear2`: Backward-Euler and second-order Gear are used.

`method=trapgear2`: Allows all three integration methods to be used.

The trapezoidal rule is usually the most efficient when you want high accuracy. This method can exhibit point-to-point ringing, but you can control this by tightening the error tolerances. For this reason, though, if you choose very loose tolerances to get a quick answer, either backward-Euler or second-order Gear will probably give better results than the trapezoidal rule. Second-order Gear and backward-Euler can make systems appear more stable than they really are. This effect is less pronounced with second-order Gear or when you request high accuracy.

Several parameters determine the accuracy of the transient analysis. `reltol` and `abstol` control the accuracy of the discretized equation solution. These parameters determine how well charge is conserved and how accurately steady-state or equilibrium points are computed. You can set the integration error, or the errors in the computation of the circuit dynamics (such as time constants), relative to `reltol` and `abstol` by setting the `lteratio` parameter.

The parameter `relref` determines how the relative error is treated. The `relref` options are:

`relref=pointlocal`: Compares the relative errors in quantities at each node to that node alone.

`relref=alllocal`: Compares the relative errors at each node to the largest values found for that node alone for all past time.

`relref=sigglobal`: Compares relative errors in each of the circuit signals to the maximum for all signals at any previous point in time.

`relref=allglobal`: Same as `relref=sigglobal` except that it also compares the residues (KCL error) for each node to the maximum of that nodes past history.

The `errpreset` parameter lets you adjust the simulator parameters to fit your needs quickly. You can set `errpreset` to `conservative` if the circuit is very sensitive, or you can set it to `liberal` for a fast. but possibly inaccurate, simulation. The setting `errpreset=moderate` suits most needs.

The effect of `errpreset` on other parameters is shown in the following table. In this table, `T= stop - start`.
check???

errpreset	reltol	relref	method	maxstep	lteratio
liberal	10	allglobal	gear2	Interval/10	3.5
moderate		sigglobal	traponly	Interval/50	3.5
conservative	0.1	alllocal	gear2only	Interval/100	10.0

The value of `reltol` is increased or decreased from its value in the options statement, but it is not allowed to be larger than 0.01. Spectre sets the value of `maxstep` so that it is no larger than the value given in the table. Except for `reltol` and `maxstep`, `errpreset` does not change the value of any parameters you have explicitly set. The actual values used for the transient analysis are given in the log file.

If the circuit you are simulating can have infinitely fast transitions (for example, a circuit that contains nodes with no capacitance), Spectre might have convergence problems. To avoid this, you must prevent the circuit from responding instantaneously. You can accomplish this by setting `cmin`, the minimum capacitance to ground at each node, to a physically reasonable nonzero value. This often significantly improves Spectre convergence.

Spectre provides two methods for reducing the number of output data points saved: `strobing`, based on the simulation time, and `skipping` time points, which saves only every Nth point.

The parameters `strobeperiod` and `strobedelay` control the strobing method.`strobeperiod` sets the interval between points that you want to save, and `strobedelay` sets the offset within the period relative to `skipstart`. The simulator forces a time step on each point to be saved, so the data is computed, not interpolated.

The skipping method is controlled by `skipcount`. If this is set to N, then only every Nth point is saved.

The parameters `skipstart` and `skipstop` apply to both data reduction methods. Before `skipstart` and after `skipstop`, Spectre saves all computed data.

If you do not want any data saved before a given time, use `outputstart`. If you do not want any data saved after a given time, change the `stop` time.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

annotate 20	ic 7	readic 9	stats 19
autostop 4	infoname 34	readns 10	step 6
circuitage 38	infotimes 35	relref 17	stop 1
ckptperiod 14	lteratio 18	restart 37	strobedelay 29
cmin 11	maxiters 36	save 22	strobeperiod 28
compression 30	maxstep 5	skipcount 27	title 21
errpreset 16	method 15	skipdc 8	write 12
flushofftime 33	nestlvl 23	skipstart 25	writefinal 13
flushpoints 31	oppoint 24	skipstop 26	
flushtime 32	outputstart 3	start 2	

Transfer Function Analysis (xf)

Description

The transfer function analysis linearizes the circuit about the DC operating point and performs a small-signal analysis that calculates the transfer function from every independent source in the circuit to a designated output. The variable of interest at the output can be voltage or current.

You can specify the output with a pair of nodes or a probe component. Any component with two or more terminals can be a voltage probe. When there are more than two terminals, they are grouped in pairs; and you use the `portv` parameter to select the appropriate pair of terminals. Alternatively, you can simply specify a voltage to be the output by giving a pair of nodes on the xf analysis statement.

Any component that naturally computes current as an internal variable can be a current probe. If the probe component computes more than one current (as transmission lines, microstrip lines, and N-ports do), you use the `porti` parameter to select the appropriate current. It is an error to specify both `portv` and `porti`. If neither is specified, the probe component provides a reasonable default.

The `stimuli` parameter specifies what is used for the inputs for the transfer functions. There are two choices. `stimuli=sources` indicates that the sources present in the circuit should be used. The `xfmag` parameters provided by the sources may be used to adjust the computed gain to compensate for gains or losses in a test fixture. One can limit the number of sources in hierarchical netlists by using the `save` and `nestlvl` parameters.

The transfer functions computed versus output and source types are:

check??

Source	Output Type		Source
Type	voltage	current	Amplitude
vsource	V(out)/V(src)	I(out)/V(src)	V(src)=xfmag
isource	V(out)/I(src)	I(out)/I(src)	I(src)=xfmag
port	2*V(out)/V(src)	2*I(out)/V(src)	V(src)=2*xfmag

where xfmag defaults to 1 for each source type. For the port, V(src) is the internal source voltage.

Specifying `stimuli=nodes_and_terminals` indicates that all possible transfer functions should be computed. This is useful when it is not known in advance which transfer functions are interesting. Transfer functions for nodes are computed assuming that a unit magnitude flow (current) source is connected from the node to ground. Transfer functions for terminals are computed assuming that a unit magnitude potential (voltage) source is connected in series with the terminal. By default, the transfer functions from a small set of terminals are computed. If transfer functions from specific terminals are desired, specify the terminals in the save statement. You must use the `:probe` modifier (for example, `Rout:1:probe`) or specify `useprobes=yes` on the options statement. If transfer functions from all terminals are desired, specify `currents=all` and `useprobes=yes` on the options statement.

Spectre can perform the analysis while sweeping a parameter. The parameter can be frequency, temperature, component instance parameter, component model parameter, or netlist parameter. If changing a parameter affects the DC operating point, 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.

Definition

Name [p] [n] xf parameter=value ...

The optional terminals (p and n) specify the output of the circuit. If you do not give the terminals, then you must specify the output with a probe component.

Parameters

1 `prevoppoint=no`	Use operating point computed on the previous analysis. Possible values are `no` or `yes`.

Sweep interval parameters

2 `start=0`	Start sweep limit.
3 `stop`	Stop sweep limit.
4 `center`	Center of sweep.
5 `span=0`	Sweep limit span.
6 `step`	Step size, linear sweep.
7 `lin=50`	Number of steps, linear sweep.
8 `dec`	Points per decade.
9 `log=50`	Number of steps, log sweep.
10 `values=[...]`	Array of sweep values.

Sweep variable parameters

11 `dev`	Device instance whose parameter value is to be swept.
12 `mod`	Model whose parameter value is to be swept.

13 `param`	Name of parameter to sweep.
14 `freq (Hz)`	Frequency when parameter other than frequency is being swept.

Probe parameters

15 `probe`	Compute every transfer function to this probe component.

State-file parameters

16 `readns`	File that contains estimate of DC solution (nodeset).

Output parameters

17 `stimuli=sources`	Stimuli used for xf analysis. Possible values are `sources` or `nodes_and_terminals`.
18 `save`	Signals to output. Possible values are `all`, `lvl`, `allpub`, `lvlpub`, `selected`, or `none`.
19 `nestlvl`	Levels of subcircuits to output.
20 `oppoint=no`	Should operating point information be computed, and if so, where should it be sent. Possible values are no, `screen`, `logfile`, or `rawfile`.

Convergence parameters

21 `restart=yes`	Do not use previous DC solution as initial guess. Possible values are no or `yes`.

Annotation parameters

22 `annotate=sweep`	Degree of annotation. Possible values are no, `title`, `sweep`, `status`, or `steps`.
23 `stats=no`	Analysis statistics. Possible values are no or `yes`.
24 `title`	Analysis title.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep. Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can give a step size parameter (`step`, `lin`, `log`, `dec`) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, 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. However, if a previous analysis computed an operating point, you can set `prevoppoint=yes` to avoid recomputing it. For example, if you use this option when the previous analysis was a transient analysis, the operating point is the state of the circuit on the final time point.

Parameter Index

In the following index, the number following each parameter name indicates where to find the description of that parameter.

annotate 22	log 9	probe 15	stats 23
center 4	mod 12	readns 16	step 6
dec 8	nestlvl 19	restart 21	stimuli 17
dev 11	oppoint 20	save 18	stop 3
freq 14	param 13	span 5	title 24
lin 7	prevoppoint 1	start 2	values 10

6

Spectre Syntax

This chapter discusses the following topics:

Using Analogmodel for Model Passing (analogmodel)

`analogmodel` is a reserved word in Spectre that allows you to bind an instance to different masters based on the value of a special instance parameter called `modelname`. An instance of `analogmodel` must have a parameter named `modelname` whose string value will be the name of the master this instance will be bound to. The value of `modelname` can be passed into subcircuits.

The `analogmodel` keyword is used by the Cadence® Analog Design Environment to enable model name passing through the schematic hierarchy.

The syntax for an instance statement with a `modelname` parameter is given below:

```
name [(]node1 ... nodeN[)] analogmodel modelname=mastername [[param1=value1]
     ...[paramN=valueN]]
```

`name` Name of the statement or instance label.

[(]`node1...nodeN`[)] Names of the nodes that connect to the component.

`analogmodel` Special device name to indicate that this instance will have its master name specified by the value of the modelname parameter on the instance.

`modelname` Parameter to specify the master of this instance indicated by mastername. The mastername must either be a valid string identifier or a netlist parameter that must resolve to a valid master name – a primitive, model, subckt, or an AHDL module.

`param1` Parameter values for the component. Depending on the master type, these can either be device parameters or netlist parameters. This is an optional field.

An Example

```
//example spectre netlist to illustrate modelname parameter
simulator lang=spectre
parameters a="low" b="bottom" modelname=10
ahdl_include "VerilogAStuff.va"
topInst1 (out in) top
topInst2 analogmodel modelname="VAMaster" //VAMaster is defined in
"VerilogAStuff.va"
    topInst3 (out in) analogmodel modelname="resistor" //topInst3 binds to a
primitive
```

```
    topInst4 (out in) analogmodel modelname="myOwnRes" //topInst4 binds to a
modelcard "myOwnRes" defined below
    topinst5 (out in) weiredRes modelname=modelname //modelname is just another
netlist param
    model myOwnRes resistor r=100
    subckt (out in) top
      parameters a="mid"
      x1 (out in) analogmodel modelname=a //topInst1.x1 binds to "mid"
    ends top
    subckt (out in) mid
      parameters c="low"
      x1 (out in) analogmodel modelname=b //topInst1.x1.x1 binds to "bottom"
      x2 (out in) analogmodel modelname=c //topInst1.x1.x1.x2 binds to "low"
    ends mid
    subckt (out in) low
      x1 (out in) analogmodel modelname="bottom" //topInst1.x1.x1.x2.x1 binds to
"bottom"
    ends low
    subckt (out in) bottom
      x1 (out in) analogmodel modelname="resistor" //x1 binds to primitive
"resistor"
    ends bottom
    dc1 dc
```

Checkpoint - Restart (checkpoint)

Description

Spectre has the ability to save checkpoint files while the analyses are running, and to restart an analysis from its checkpoint file. Checkpoint files can be generated in several ways:

1) Periodically based on real time (wall clock time).

2) Asynchronous UNIX signals.

3) By other methods unique to the analyses.

To generate checkpoint files periodically based on real time, set the Spectre option `ckptclock` to the time interval in seconds that you want checkpoints. This option is turned on by default with a value of 1800 seconds (30 minutes). Spectre will delete the checkpoint file if the simulation completes normally. If the simulation terminates abnormally, the checkpoint file will not be deleted.

If Spectre receives the UNIX signal USR2, then Spectre will immediately write a checkpoint file. If Spectre receives interrupt signals like QUIT, TERM, INT, or HUP, Spectre will attempt to write a checkpoint file and then exit. After other fatal signals, it may not be possible for Spectre to write a checkpoint file.

The name of the checkpoint file is a combination of the circuit name, the analysis name, and the extension `.ckpt`. For example, if the circuit is named `test1` and the transient analysis is named `timeSweep`, then the checkpoint file will be named `test1.timeSweep.tran.ckpt`.

Spectre keeps only the latest checkpoint file. When a new checkpoint is created, it creates the file under a temporary name. After the file has been successfully written, it deletes the previous checkpoint file and renames the new file.

Currently only the transient analyses supports checkpoint and restart.

Checkpoint for Transient Analyses

The transient analysis can generate checkpoint files by using the above methods, or by generating a checkpoint file periodically based on the transient simulation time. This is accessed by a transient analysis parameter called `ckptperiod`, which is turned off by default.

Restart

To restart an analysis from a checkpoint file, use the +recover option on the Spectre command line. Spectre will look though the requested analyses to see if a checkpoint file exists for any of them. If a checkpoint file for a given analysis does exist, Spectre will skip over any analyses previous to that one, and start the analysis using the information from the file.

Configuring CMI Shared Objects (cmiconfig)

Description

Spectre supports the ability to install devices dynamically from shared objects at run time. CMI Configuration files are used to determine and locate the set of shared objects to be installed. Spectre first reads the default CMI configuration file which specifies the default shared objects provided by Cadence. The configuration file specified by the value of the CMI_CONFIG environment variable is then read. The third configuration file that Spectre reads is ~/.cmiconfig. Finally, the configuration file specified in Spectre's -cmiconfig command line argument is read. Each CMI configuration file modifies the existing configuration established by the configuration files read before.

The following commands can be used in a CMI configuration file.

`setpath`Specifies and resets the search path.
`setpath <path> or setpath ( <path1> <path2> ... <pathN> )`

`prepend`Adds a path before the current search path.
`prepend <path> or prepend ( <path1> <path2> ... <pathN> )`

`append`Adds a path after the current search path.
`append <path> or append ( <path1> <path2> ... <pathN> )`

`load`Add a shared object to the list of shared objects to load.
`loads [path/]<shared_object_name>`

`unload`Removes a shared object to the list of shared objects to load.
`unload <shared_object_name>`

For example, given the following CMI configuration file

```
append /hm/spectre_dev/tools.sun4v/spectrecmi/lib/cmi/1.0
load libbjtx+tfet.so
load libmosx.so
```

The shared objects `libbjtx+tfet.so` and `libmosx.so` are loaded from `/hm/spectre_dev/tools.sun4v/spectrecmi/lib/cmi/1.0` in addition to the default shared objects provided by Cadence.

Built-in Mathematical and Physical Constants (constants)

Description

Spectre supports the following list of built-in mathematical and physical constants:

M_ is a mathematical constant

M_E	2.7182818284590452354	1 / pi
M_LOG2E	1.4426950408889634074	log2(e)
M_LOG10E	0.43429448190325182765	log10(e)
M_LN2	0.69314718055994530942	ln(2)
M_LN10	2.30258509299404568402	ln(10)
M_PI	3.14159265358979323846	pi
M_TWO_PI	6.28318530717958647652	2 * pi
M_PI_2	1.57079632679489661923	pi/2
M_PI_4	0.78539816339744830962	pi/4
M_1_PI	0.31830988618379067154	1/pi
M_2_PI	0.63661977236758134308	2/pi
M_2_SQRTPI	1.12837916709551257390	2/sqrt(pi)
M_SQRT2	1.41421356237309504880	sqrt(2)
M_SQRT1_2	0.70710678118654752440	sqrt(1/2)
M_DEGPERRAD	57.2957795130823208772	number of degrees per radian

P_ is a physical constant

P_Q	1.6021918e-19	charge of electron in coulombs
P_C	2.997924562e8	speed of light in vacuum in meters/sec
P_K	1.3806226e-23	Boltzman's constant in joules/kelvin
P_H	6.6260755e-34	Planck's constant in joules*sec
P_EPS0	8.85418792394420013968e-12	permittivity of vacuum in farads/meter
P_U0	(4.0e-7 * M_PI)	permeability of vacuum in henrys/meter
P_CELSIUS0	273.15	zero celsius in kelvin

These constants can be used in expressions, or anywhere that a numeric value of expression is expected.

Convergence Difficulties (convergence)

Description

If you are having convergence difficulties, try the following suggestions:

1. Carefully evaluate and resolve any notice, warning or error messages.
2. Assure topology checker is used (set topcheck=full on options statement) and heed any warnings it gives.
3. Perform sanity checking on the parameter values using the parameter range checker (use `+param param-limits-file` as a command line argument) and heed any warnings. Print the minimum and maximum parameter value using the `info` analysis. Assure that the bounds given for instance, model, output, temperature-dependent, and operating-point (if possible) parameters are reasonable.
4. Small floating resistors connected to high impedance nodes can cause convergence difficulties. Avoid very small floating resistors, particularly small parasitic resistors in semiconductors. Use voltage sources or iprobes to measure current rather than small resistors.
5. Use realistic device models. Check all component parameters, particularly nonlinear device model parameters, to assure that they are reasonable.
6. Increase the value of gmin (on options statement).
7. Loosen tolerances, particularly absolute tolerances like `iabstol` (on options statement). If tolerances are set too tight, they might preclude convergence.
8. Try to simplify the nonlinear component models in order to avoid regions in the model that may contribute to convergence problems.

DC Convergence Suggestions

Once you have a solution, write it to a nodeset file using the `write` parameter, and read it back in on subsequent simulations using the `readns` parameter.

1. If you have an estimate of what the solution should be, use nodeset statements or a nodeset file and set as many nodes as possible.

2. If convergence difficulties occur when using nodesets or initial conditions, try increasing `rforce` (on options statement).
3. If this is not the first analysis, perhaps the solution from the previous analysis is far from the solution for this analysis. If so, set restart=yes.
4. If simulating a bipolar analog circuit, assure the region parameter on all transistors and diodes is set correctly.
5. If analysis fails at an extreme temperature, but succeeds at room temperature, then try adding a DC analysis that sweeps temperature. Start at room temperature, sweep to the extreme temperature, and write the final solution to a nodeset file.
6. Use numeric pivoting in the sparse matrix factorization by setting `pivotdc=yes` (on options statement). Sometimes it is also necessary to increase the pivot threshold to somewhere in the range of 0.1 to 0.5 using `pivrel` (on options statement).
7. Divide the circuit into smaller pieces and simulate them individually, but be careful to assure that the results will be close to what they would be if the rest of the circuit was present. Use the results to generate nodesets for the whole circuit.
8. If all else fails, replace the DC analysis with a transient analysis and modify all the independent sources to start at zero and ramp to their DC values. Run the transient analysis well beyond the time when all the sources have reached their final value (remember that transient analysis is very cheap when all of the signals in the circuit are not changing) and write the final point to a nodeset file. To make the transient analysis more efficient set the integration method to backward Euler (`method=euler`) and loosen the local truncation error criteria by increasing `lteratio`, say to 50. Occasionally, this approach will fail or be very slow because the circuit contains an oscillator. Often times the oscillation can be eliminated for the sake of finding the dc solution by setting the minimum capacitance from each node to ground (`cmin`) to a large value.

Transient Convergence Suggestions

1. Assure that a complete set of parasitic capacitors is used on nonlinear devices to avoid jumps in the solution waveforms. On MOS models, specify nonzero source and drain areas.
2. Use the `cmin` parameter to install a small capacitor from every node in the circuit to ground. This usually eliminates any jumps in the solution.

Export a Measurement to Be Evaluated (export)

The export feature is not supported. It is designated for internal use only.

Expressions (expressions)

Description

An expression is a construct that combines operands with operators to produce a result that is a function of the values of the operands and the semantic meaning of the operators. Any legal operand is also an expression in itself. Legal operands include numeric constants and references to top-level netlist parameters or subcircuit parameters. Calls to algebraic and trigonometric functions are also supported. The complete lists of operators, algebraic, and trigonometric functions are given after some examples.

Examples

```
simulator lang=spectre
parameters p1=1 p2=2            // declare some top-level parameters
r1 (1 0) resistor r=p1          // the simplest type of expression
r2 (1 0) resistor r=p1+p2       // a binary (+) expression
r3 (1 0) resistor r=5+6/2       // expression of constants, = 8
x1 s1 p4=8      // instantiate a subcircuit, defined in the following lines
subckt s1
parameters p1=4 p3=5 p4=6       // subcircuit parameters
    r1 (1 0) resistor r=p1              // another simple expression
    r2 (1 0) resistor r=p2*p2           // a binary multiply expression
    r3 (1 0) resistor r=(p1+p2)/p3      // a more complex expression
    r4 (1 0) resistor r=sqrt(p1+p2)     // an algebraic function call
    r5 (1 0) resistor r=3+atan(p1/p2)   // a trigonometric function call
    r6 (1 0) RESMOD r=(p1 ? p4+1 : p3)  // the ternary operator
ends
//  a model card, containing expressions
model RESMOD resistor tc1=p1+p2 tc2=sqrt(p1*p2)
//  some expressions used with analysis parameters
time_sweep tran start=0 stop=(p1+p2)*50e-6 // use 5*50e-6 = 150 us
//  a vector of expressions (see notes on vectors below)
dc_sweep dc param=p1 values=[0.5 1 +p2 (sqrt(p2*p2)) ]  // sweep p1
```

Where Expressions Can Be Used

The Spectre native netlist language allows expressions to be used where numeric values are expected 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. Expressions can be used when specifying device or analysis instance parameter values (for example specifying the resistance of a resistor or the stop time of a transient analysis, as outlined in the preceding example), when specifying model parameter values in model cards (for example specifying "bf=p1*0.8" for a bipolar model parameter, bf), or when specifying initial conditions and nodesets for individual circuit nodes.

Operators

The following operators are supported, listed in order of decreasing precedence. Parentheses can be used to change the order of evaluation. For a binary expression like "a+b", "a" is the first operand and "b" is the second operand. All operators are left associative, with the exceptions of the "to the power of" operator (**) and the ternary operator (? :), which are right associative. For logical operands, any nonzero value is considered true. The relational and equality operators return a value of 1 to indicate true or 0 to indicate false. There is no short circuiting of logical expressions involving && and ||.

Operator	Symbol(s)	Value
Unary +, Unary -	`+, -`	Value of operand, negative of operand.
To the power of	`**`	First operand to raised to power of second operand.
Multiply, Divide	`*, /`	Product, Quotient of operands.
Binary Plus/Minus	`+, -`	Sum, Difference of operands.
Shift	`<<, >>`	First operand shifted left (, right) by number of bits specified by second operand.
Relational	`<, <=, >, >=`	greater than, greater than or equal.
Equality	`==, !=`	True if operands are equal, not equal.
Bitwise AND	&	Bitwise AND (of integer operands).
Bitwise Exclusive NOR	`~^ (or ^~)`	Bitwise Exclusive NOR (of integer operands).
Bitwise OR	`|`	Bitwise OR (of integer operands).
Logical AND	&&	True only if both operands true.

Operator	Symbol(s)	Value
Logical OR	`||`	True if either operand is true.
Ternary Operator	`(`*cond*`) ? x : y`	Returns x if *cond* is true, y if not where x and y are expressions.

Algebraic and Trigonometric Functions:

The trigonometric and hyperbolic functions expect their operands to be specified in radians. The atan2() and hypot() functions are useful for converting from Cartesian to polar form.

Function	Description	Domain
`log(`*x*`)`	Natural logarithm	*x* > 0
`log10(`*x*`)`	Decimal logarithm	*x* > 0
`exp(`*x*`)`	Exponential	*x* < 80
`sqrt(`*x*`)`	Square Root	*x* > 0
`min(`*x*`,`*y*`)`	Minimum value	All *x*, all *y*
`max(`*x*`,`*y*`)`	Maximum value	All *x*, all *y*
`abs(`*x*`)`	Absolute value	All *x*
`pow(`*x*`,`*y*`)`	*x* to the power of *y*	All *x*, all *y*
`int(`*x*`)`	integer value of *x*	All *x*
`floor(`*x*`)`	largest integer <= *x*	All *x*
`ceil(`*x*`)`	smallest integer >= *x*	
`fmod(`*x*`,`*y*`)`	floating-point modulus	All *x*, all *y*, except *y*=0
`sin(`*x*`)`	Sine	All *x*
`cos(`*x*`)`	Cosine	All *x*
`tan(`*x*`)`	Tangent	All *x*, except *x* = n*(pi/2), where n odd
`asin(`*x*`)`	Arc-sine	-1 <= *x* <= 1
`acos(`*x*`)`	Arc-cosine	-1 <= *x* <= 1
`atan(`*x*`)`	Arc-tangent	All *x*
`atan2(`*x*`,`*y*`)`	Arc-tangent of *x*/*y*	All *x*, all *y*

Function	Description	Domain
`hypot(`*x*`,`*y*`)`	`sqrt(`*x*`*`*x*` + `*y*`*`*y*`)`	All *x*, all *y*
`sinh(`*x*`)`	Hyperbolic sine	All *x*
`cosh(`*x*`)`	Hyperbolic cosine	All *x*
`tanh(`*x*`)`	Hyperbolic tangent	All *x*
`asinh(`*x*`)`	Arc-hyberbolic sine	All *x*
`acosh(`*x*`)`	Arc-hyperbolic cosine	*x*
`atanh(`*x*`)`	Arc-hyperbolic tangent	-1 <= *x* <= 1

User-defined functions are also supported. See `spectre -h functions` for a description of user-defined functions.

A large number of built-in mathematical and physical constants are available for use in expressions. See `spectre -h constants` for the list of these constants

Using Expressions in Vectors

Expressions can be used as vector elements, as in the following example:

```
dc_sweep dc param=p1 values=[0.5 1 +p2 (sqrt(p2*p2)) ]  // sweep p1
```

Note that when expressions are used within vectors, anything other than constants, parameters, or unary expressions (unary +, unary -) must be surrounded by parentheses. Vector elements should be space separated for clarity, though this is not mandatory. The preceding `dc_sweep` example shows a vector of four elements, namely 0.5, 1, +p2, and sqrt(p2*p2). Note that the square root expression is surrounded by parentheses.

User Defined Functions (functions)

Description

Spectre's user-defined function capability allows you to build upon the provided set of built-in mathematical and trigonometric functions. You can write your own functions, and call these functions from within any expression. The syntax for calling a user-defined function is the same as the syntax for calling a built-in algebraic or trigonometric function. Note that user-defined functions must be defined before they are referenced (called). Arguments to user-defined functions will be taken as real values, and the functions will return real values. A user-defined function may contain only a single statement in braces and this statement must return an expression (which will typically be an expression involving the function arguments). The return expression may reference the built in parameters "temp" and "tnom". User-defined functions must be declared at the top level only, and must not be declared within subcircuits. User-defined functions may be called from anywhere that an expressions can be currently used in Spectre. User-defined functions may call other functions (both user-defined and built-in), however any user-defined function will need to be declared before it can be called. User-defined functions can override built-in mathematical and trigonometric functions.

Note: Only real values for arguments and return values are supported in this release.

See `spectre -h expressions` for a list of built-in algebraic and trigonometric functions.

Definition

```
real myfunc( [real arg1, ...real argn] ) {
```

Examples

```
real myfunc( real a, real b ) {
    return a+b*2+sqrt(a*sin(b));
}
```

An example of a function calling a previously defined function is given below.

```
real yourfunc( real a, real b ) {
    return a+b*myfunc(a,b);     // call "myfunc"
}
```

The final example shows how a user-defined function may be called from an expression in the Spectre netlist.

```
r1 (1 0) resistor r=myfunc(2.0, 4.5)
```

Global Nodes (global)

Description

The global statement allows a set of nodes to be designated as common to the main circuit and all subcircuits. Thus, components inside subcircuits can be attached to global nodes even though the subcircuits terminals are not attached to these nodes.

Any number of global nodes may be specified using the global statement. To do this, follow the keyword global with a list of the node names that you wish to declare as global. The first node name that appears in this list is taken to be the name of the ground node. Ground is also known as the datum or reference node. If a global statement is not used, 0 is taken to be the name of the ground node.

At most one global statement is allowed and, if present, it must be the first statement in the file (however, you can have `simulator lang=spectre` statement, before the global statement so that you can use mixed case names for the node names). Ground is always treated as global even if a global statement is not used.

Definition

```
global <ground> <node> ...
```

Initial Conditions (ic)

Description

The `ic` statement is used to provide initial conditions for nodes in the transient analysis. It can occur multiple times in the input and the information provided in all the occurrences is collected. Initial conditions will only be accepted for inductor currents and node voltages where the nodes have a path of capacitors to ground. For more information, read the description of transient analysis. It should be noted that specifying `cmin` for a transient analysis, will not satisfy the condition that a node has a capacitive path to ground.

Definition

```
ic <node=value> ...
```

This statement takes a list of signals as an argument. The concept of nodes has been generalized to signals where a signal is a value associated with a topological node of the circuit or some other unknown that is solved by the simulator, such as the current through a inductor or the voltage of the internal node in a diode. Topological nodes can be either at the top-level or in a subcircuit.

For example,

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

where 7=0 implies that 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.

The Structural if-statement (if)

Description

The structural if-statement can be used to conditionally instantiate other instance statements.

Definition

```
if <condition> <statement1> [ else <statement2> ]
```

The condition is a boolean expression based on the comparisons of various arithmetic expressions which are evaluated during circuit hierarchy flattening. The statement1 and statement2 fields can be ordinary instance statements, if-statements, or a list of these within braces ({}). (Note that ordinary instance statements need a newline to terminate them.) The `else` part is optional. When if-statements are nested without braces, an `else` matches the closest previous unmatched `if` at the same level.

It is possible to have duplicate instance names within the if statement under strict topological conditions: These are:

- references to instance with duplicate names is only possible within a structural if statement which has both an "if" part and an "else" part.
- both the "if" part and the "else" part must be either a simple one-statement block, or another structural if statement to which these same rules apply.
- The duplicate instances must have the same number of terminals and be bound to the same list of nodes.
- The duplicate instances must refer to the same primitive or model.
- Where duplicate instances refer to a model, the underlying primitive must be the same.

This feature allows automatic model selection based on any netlist or subcircuit parameter. As an example, consider using Spectre's inline subcircuits and structural if statement to implement automatic model selection based on bipolar device area. Here, the duplicate instances are the inline components.

```
model npn_default bjt is=3.2e-16 va=59.8
model npn10x10 bjt is=3.5e-16 va=61.5
model npn20x20 bjt is=3.77e-16 va=60.5
// npn_mod choses scaled models binned on area!
// if ( area < 100e-12 ) use model npn10x10
// else if ( area <  400e-12 ) use model npn20x20
// else use model npn_default
```

```
inline subckt npn_mod (c b e s)
  parameters area=5e-12
  if ( area < 100e-12 ) {
      npn_mod (c b e s) npn10x10  // 10u * 10u, inline device
  } else if ( area < 400e-12 ) {
      npn_mod (c b e s) npn20x20  // 20u * 20u, inline device
  } else {
      npn_mod (c b e s) npn_default       // 5u * 5u, inline device
  }
ends npn_mod
q1 (1 2 0 0) npn_mod area=350e-12         // gets 20x20 model
q2 (1 3 0 0) npn_mod area=25e-12          // gets 10x10 model
q3 (1 3 0 0) npn_mod area=1000e-12        // gets default model
```

Include File (include)

Description

File inclusion allows the circuit description to be spread over several files. The `include` statement itself is replaced by the contents of the file named. An included file may also contain include statements. If the name given is not an absolute path specification, then it is taken relative to the directory of the file currently being read.

In order to read existing SPICE library and model files, Spectre automatically switches to SPICE input mode when it opens an include file. Thus, all files that use the Spectre native language must begin with a `simulator lang=spectre` statement. The one exception is files that end with a ".scs" file extension which are treated specially and are read in Spectre input mode. This language mode treatment applies to files included by both Spectre's `include` statement, and CPP's `#include` statement.

After reading the include file, Spectre restores the language processing mode to what it was before the file was included, and continues reading the original file starting at the line after the include statement. Lines cannot be continued across file boundaries.

The CPP `#include` statement differs from Spectre's `include` statement in that CPP macro processing will not be performed on files included by Spectre, but will be performed on files included by CPP. If your netlist contains a `#include` statement, you must run CPP to perform this inclusion, otherwise an error will occur.

If the file to be included cannot be found in the same directory as the including file, both Spectre's `include` and CPP's `#include` will search for the file to be included along the search path specified by the `-I` command line arguments.

Spectre's `include` statement allows you to embed special characters in the name of the file to be included. Spectre's `include` statement will automatically expand the "~" character to the users home directory, and will expand environment variables and % codes, such as

```
include "~/models/${SIMULATOR}_pd/npn.scs"
```

which will look in the directory given by the environment variable SIMULATOR followed by `_pd`, which is under the `models` directory, in the users home directory. Note: These special character features are not available using CPP's `#include` statement.

Definition

```
include "filename"
```

Spectre Netlist Keywords (keywords)

Description

The following lists the reserved Spectre keywords, including netlist keywords, built-in algebraic and trigonometric functions, and built-in mathematical and physical constants. When creating a netlist, you should avoid using any of the keywords from this list in any context other than that in which it was intended. Creating node names, parameter names, instance names, or model names from any of these keywords will result in an error.

Keyword	Keyword Type
`M_1_PI`	Mathematical Constant
`M_2_PI`	Mathematical Constant
`M_2_SQRTPI`	Mathematical Constant
`M_DEGPERRAD`	Mathematical Constant
`M_E`	Mathematical Constant
`M_LN10`	Mathematical Constant
`M_LN2`	Mathematical Constant

Keyword	Keyword Type
M_LOG10E	Mathematical Constant
M_LOG2E	Mathematical Constant
M_PI	Mathematical Constant
M_PI_2	Mathematical Constant
M_PI_4	Mathematical Constant
M_SQRT1_2	Mathematical Constant
M_SQRT2	Mathematical Constant
M_TWO_PI	Mathematical Constant
P_C	Mathematical Constant
P_CELSIUS0	Mathematical Constant
P_EPS0	Mathematical Constant
P_H	Mathematical Constant
P_K	Mathematical Constant
P_Q	Mathematical Constant
P_U0	Mathematical Constant
abs	Algebraic Function
acos	Trigonometric Function
acosh	Trigonometric Function
altergroup	Netlist Keyword
asin	Trigonometric Function
asinh	Trigonometric Function
atan	Trigonometric Function
atan2	Trigonometric Function
atanh	Trigonometric Function
ceil	Algebraic Function
correlate	Netlist Keyword
cos	Trigonometric Function

Keyword	Keyword Type
`cosh`	Trigonometric Function
`else`	Netlist Keyword
`end`	Netlist Keyword
`ends`	Netlist Keyword
`exp`	Algebraic Function
`export`	Netlist Keyword
`floor`	Algebraic Function
`fmod`	Algebraic Function
`for`	Netlist Keyword
`function`	Netlist Keyword
`global`	Netlist Keyword
`hypot`	Algebraic Function
`ic`	Netlist Keyword
`if`	Netlist Keyword
`inline`	Netlist Keyword
`int`	Algebraic Function
`library`	Netlist Keyword
`local`	Netlist Keyword
`log`	Algebraic Function
`log10`	Algebraic Function
`march`	Netlist Keyword
`max`	Algebraic Function
`min`	Algebraic Function
`model`	Netlist Keyword
`nodeset`	Netlist Keyword
`parameters`	Netlist Keyword
`paramset`	Netlist Keyword

Keyword	Keyword Type
`plot`	Netlist Keyword
`pow`	Algebraic Function
`print`	Netlist Keyword
`pwr`	Netlist Keyword
`real`	Netlist Keyword
`return`	Netlist Keyword
`save`	Netlist Keyword
`sens`	Netlist Keyword
`sin`	Trigonometric Function
`sinh`	Trigonometric Function
`sqrt`	Algebraic Function
`statistics`	Netlist Keyword
`subckt`	Netlist Keyword
`tan`	Trigonometric Function
`tanh`	Trigonometric Function
`to`	Netlist Keyword
`truncate`	Netlist Keyword
`vary`	Netlist Keyword

Library - Sectional Include (library)

Description

Library inclusion allows the circuit description to be spread over several files. The library statement itself is replaced by the contents of the section of the library file. A library section may also contain library reference statements. If the file name given is not an absolute path specification, then it is taken relative to the directory of the file currently being read.

There are two kinds of library statements. One that references a library section, and one that defines a library section. The definition of a library section is prohibited in the netlist.

In order to read existing SPICE library and model files, Spectre automatically switches to SPICE input mode when it opens a library file. Thus, all files that use the Spectre native language must contain a `simulator lang=spectre` statement within each section of the library or the file can have a `scs` filename extension. After reading the library section, Spectre restores the language processing mode and continues reading the original file starting at the line after the library statement. Lines cannot be continued across file boundaries.

Spectre allows only one library per file, but a library may contain multiple sections (typically one section per process corner for example.)

Definition

```
Inside netlist (reference library section)
```

Sample Library

```
library corner_lib
section tt
   model nch bsim3v3 type=n mobmod=1 capmod=2 version=3.1
   + xj=1.7e-7 vsat=7.99e4 at=3.6e4 a0=0.799 ags=0.4
   + a1=0 a2=1 keta=-0.05 nch=2.8e17 ngate=1.31e20 k1=0.74
   model pch bsim3v3 type=p mobmod=1 capmod=2 version=3.1
   + xj=1.7e-7 vsat=1.38e5 at=1e5 a0=1.3 ags=0.3
   + a1=1.1e-4 a2=1 keta=0 nch=4.1e17 ngate=7.6e19 k1=0.88
   model knpn bjt is=10e-13 bf=170 va=58.7 ik=5.63e-3 rb=rbn rbm=86
   + re=3.2 cje=0.25e-12 pe=0.76 me=0.34 tf=249e-12 cjc=0.34e-12 pc=0.55
   + mc=0.35 ccs=2.4e-12 ms=0.35 ps=0.53 rc=169
   model kpnp bjt type=pnp is=10e-13 bf=60 va=43.1 ik=0.206e-3 rb=rbp rbm=64.3
   + re=33.8 cje=0.16e-12 pe=0.5 me=0.26 tf=36e-9 cjc=0.72e-12 pc=0.58
   + mc=0.34 ccs=2.5e-12 ps=0.53 ms=0.35 rc=276
endsection
section ss
   model nch bsim3v3 type=n mobmod=1 capmod=2 version=3.1
   + xj=1.7e-7 vsat=7.99e4 at=3.6e4 a0=0.799 ags=0.4
   + a1=0 a2=1 keta=-0.05 nch=2.8e17 ngate=1.31e20 k1=0.74
   model pch bsim3v3 type=p mobmod=1 capmod=2 version=3.1
   + xj=1.7e-7 vsat=1.38e5 at=1e5 a0=1.3 ags=0.3
   + a1=1.1e-4 a2=1 keta=0 nch=4.1e17 ngate=7.6e19 k1=0.88
   model knpn bjt is=10e-13 bf=70 va=58.7 ik=5.63e-3 rb=rbn rbm=86
   + re=3.2 cje=0.25e-12 pe=0.76 me=0.34 tf=249e-12 cjc=0.34e-12 pc=0.55
   + mc=0.35 ccs=2.4e-12 ms=0.35 ps=0.53 rc=169
   model kpnp bjt type=pnp is=10e-13 bf=30 va=43.1 ik=0.206e-3 rb=rbp rbm=64.3
```

```
   + re=33.8 cje=0.16e-12 pe=0.5 me=0.26 tf=36e-9 cjc=0.72e-12 pc=0.58
   + mc=0.34 ccs=2.5e-12 ps=0.53 ms=0.35 rc=276
endsection
section ff
   model nch bsim3v3 type=n mobmod=1 capmod=2 version=3.1
   + xj=1.7e-7 vsat=7.99e4 at=3.6e4 a0=0.799 ags=0.4
   + a1=0 a2=1 keta=-0.05 nch=2.8e17 ngate=1.31e20 k1=0.74
   model pch bsim3v3 type=p mobmod=1 capmod=2 version=3.1
   + xj=1.7e-7 vsat=1.38e5 at=1e5 a0=1.3 ags=0.3
   + a1=1.1e-4 a2=1 keta=0 nch=4.1e17 ngate=7.6e19 k1=0.88
   model knpn bjt is=10e-13 bf=220 va=58.7 ik=5.63e-3 rb=rbn rbm=86
   + re=3.2 cje=0.25e-12 pe=0.76 me=0.34 tf=249e-12 cjc=0.34e-12 pc=0.55
   + mc=0.35 ccs=2.4e-12 ms=0.35 ps=0.53 rc=169
   model kpnp bjt type=pnp is=10e-13 bf=90 va=43.1 ik=0.206e-3 rb=rbp rbm=64.3
   + re=33.8 cje=0.16e-12 pe=0.5 me=0.26 tf=36e-9 cjc=0.72e-12 pc=0.58
   + mc=0.34 ccs=2.5e-12 ps=0.53 ms=0.35 rc=276
endsection
endlibrary
```

Node Sets (nodeset)

Description

The `nodeset` statement is used to provide an initial guess for nodes in any DC analysis or the initial condition calculation for the transient analysis. It can occur multiple times in the input, the information provided in all the occurrences is collected. For more information, read the description of DC analysis.

Definition

```
nodeset <node=value> ...
```

This statement takes a list of signals as an argument. The concept of nodes has been generalized to signals where a signal is a value associated with a topological node of the circuit or some other unknown that is solved by the simulator, such as the current through a inductor or the voltage of the internal node in a diode. Topological nodes can be either at the top-level or in a subcircuit.

For example,

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

where 7=0 implies that node 7 should be about 0V, node `out` should be about 1V, node `comp` in subcircuit `OpAmp1` should be about 5V, and the current through the first terminal of `L1` should be about 1uA.

Parameter Soft Limits (param_limits)

Description

The parameter values passed to Spectre components and analysis are subject to both hard and soft limits. If you set a parameter to a value that violates a hard limit, such as giving z0=0 to a transmission line, Spectre issues an error message and quits. If the given parameter value violates a soft limit, only a warning is issued, but Spectre uses the value of the component as given. Hard limits are used to prevent you from using values that would cause Spectre to fail or put a model in an invalid region. Soft limits are used to call attention to unusual parameter values that might have been given mistakenly. If a parameter value violates a soft limit, a message similar to one of the following sample messages is printed:

```
Parameter rb has the unusually small value of 1uOhms.
```

or

```
Parameter rb has the unusually large value of 1MOhms.
```

Spectre has built-in soft limits on a few parameter values. However, it is possible for you to override these limits, or provide limits on parameters that do not have built-in limits. To do so, create a parameter range limits file, and invoke Spectre giving the name of the file after the +param command line option. For example,

```
spectre +param limits-file input-file
```

Limits are given using the following syntax:

```
[PrimitiveName] [model] [LowerLimit <[=]] [|]Param[|] [<[=] UpperLimit]
```

The limits can be given as strict (using <=) or nonstrict (using <). If the limits are strict, there can be no space between < and =. The limits for one parameter are given on one line. There is no way of continuing the specification of the limits for a parameter over more than one line. If a parameter is given more than once, the limits given the last time override earlier limits. The primitive name must be a Spectre primitive name, not a name used for SPICE compatibility. So, for example, mos3 must be used rather than mos. Parameter limits can be written using Spectre native mode metric scale factors. Thus a limit of f <= 1.0e6 could also written as f <= 1M.

Here are some examples.

```
mos3       0.5u <= l <= 100u
           0.5u <= w
```

```
              0 < as <= 1e-8
              0 < ad <= 1e-8
     model |vto| <= 3
```

Notice that it is not necessary to give the primitive name each time. If not given, it is assumed to be the same as the previous parameter. Upper and lower limits may be given, and if not given there will be no limit on the parameter value. Thus, in the example, if w is less than 0.5um, a warning will be issued, but there is no limit on how large w can be. If a parameter is mentioned, but no limits given, then all limits are disabled for that parameter. Limits are placed on model parameters by giving the model keyword. If the model keyword is not given, the limits are applied to instance parameters. Notice that you can also place upper or lower limits on the absolute value of a parameter. For example,

```
resistor 0.1 < |r| < 1M
```

indicates that the absolute value of r should be greater than 0.1 Ohm and less than 1 MOhm. There can be no spaces between the absolute value symbols and the parameter name.

Here are some more examples.

```
1 <= x < 0.5
1 <= y <= 1
1 < z < 1
```

In the first case the lower bound is larger than the upper bound, which indicates that the range of x is all real numbers except those from 0.5 to 1 and 0.5 itself. The limits are applied separately, thus x must be both greater than or equal to 1 (1 <= x) and less than 0.5 (x < 0.5). The second case specifies that y should be 1 and the third case specifies that z should not be 1.

It is possible to specify limits for any scalar parameter that takes either a real number, an integer, or an enumeration. To specify the limits of a parameter that takes enumerations use the indices associated with the enumerations. For example, consider the region parameter of the bjt. There are four possible regions: off, fwd, rev, or sat (see `spectre -help bjt`). Each enumeration is assigned a number starting at 0 and counting up. Thus, off=0, fwd=1, rev=2, and sat=3. The specification

```
bjt        3 <= region <= 1
```

indicates that a warning should be printed if region=rev because the conditions (3 <= region) and (region <= 1) exclude only (region=2) and region 2 is rev.

It is possible to read a parameter limits file from within another file. To do so, use an include statement. For example,

```
include "filename"
```

will temporarily suspend the reading of the current file until the contents of `filename` have been read. Include statements may be nested arbitrarily deep with the condition that the operating system may limit the number of files that Spectre may have open at once. Paths in file names are taken to be relative to the directory that contains the current file, not from the directory in which Spectre was invoked.

Spectre can be instructed to always read a parameter limits file by using the SPECTRE_DEFAULTS environment variable. For example, if you put the following in you shell initialization file (.profile for sh, .cshrc for csh)

```
setenv SPECTRE_DEFAULTS "+param /cds/etc/spectre/param.lmts"
```

Spectre would always read the specified limits file.

Netlist Parameters (parameters)

Description

The Spectre native netlist language allows parameters to be specified and referenced in the netlist, both at the top-level scope and within subcircuit declarations (run `spectre -h subckt` for more details on parameters within subcircuits).

Definition

```
parameters <param=value> [param=value] ...
```

Examples

```
    simulator lang=spectre
    parameters p1=1 p2=2           // declare some parameters
    r1 (1 0) resistor r=p1         // use a parameter, value=1
    r2 (1 0) resistor r=p1+p2      // use parameters in an expression, value=3
    x1 s1 p4=8      // "s1" is defined below, pass in value 8 for "p4"
    subckt s1
    parameters p1=4 p3=5 p4=6      // note: no "p2" here, p1 "redefined"
    r1 (1 0) resistor r=p1         // local definition used: value=4
    r2 (1 0) resistor r=p2         // inherit from parent(top-level) value=2
    r3 (1 0) resistor r=p3         // use local definition, value=5
    r4 (1 0) resistor r=p4         // use passed-in value, value=8
    r5 (1 0) resistor r=p1+p2/p3   // use local+inherited/local = (4+2/5) = 4.4
    ends
time_sweep tran start=0 stop=(p1+p2)*50e-6 // use 5*50e-6 = 150 us
```

```
dc_sweep dc param=p1 values=[0.5 1 +p2 (sqrt(p2*p2)) ]  // sweep p1
```

Parameter Declaration

Parameters can be declared anywhere in the top-level circuit description or on the first line of a subcircuit definition. Parameters must be declared before they are used (referenced). Multiple parameters can be declared on a single line. When parameters are declared in the top-level, their values must be specified. When parameters are declared within subcircuits, their default values are optionally specified.

Parameter Inheritance

Subcircuit definitions inherit parameters from their parent (enclosing subcircuit definition, or top-level definition). This inheritance continues across all levels of nesting of subcircuit definitions, that is, if a subcircuit s1 is defined, which itself contains a nested subcircuit definition s2, then any parameters accessible within the scope of s1 are also accessible from within s2. Also, any parameters declared within the top-level circuit description are also accessible within both s1 and s2. However, any subcircuit definition can redefine a parameter that it has inherited. In this case, if no value is specified for the redefined parameter when the subcircuit is instantiated, then the redefined parameter uses the locally defined default value, rather than inheriting the actual parameter value from the parent.

Parameter Namespace

Parameter names must not conflict with device or analysis instance names, that is, it is not possible to reference a parameter called "r1" if there is an instance of a resistor (or other device or analysis) called "r1". Parameter names must also not be used where a node name is expected.

Parameter Referencing

Spectre netlist parameters can be referenced 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. This includes referencing of parameters in expressions (run `spectre -h expressions` for more details on netlist expression handling), as indicated in the preceding examples. You can use expressions containing parameter references when specifying device or analysis instance parameter values (for example specifying the resistance of a resistor or the stop time of a transient analysis, as outlined in the preceding example), when specifying model parameter values in model cards (for example specifying "bf=p1*0.8" for a bipolar model parameter, bf), or when specifying initial conditions and nodesets for individual circuit nodes.

Altering/Sweeping Parameters

Just as certain Spectre analyses (for example `sweep`, `alter`, `ac`, `dc`, `noise`, `sp`, `xf`) can sweep device instance or model parameters, they can also sweep netlist parameters. Run `spectre -h <analysis>` to see the particular details for any of these analyses, where <analysis> is the analysis of interest.

Temperature as a parameter

You can use the reserved parameters `temp` and `tnom` anywhere that an expression can be used, including within expressions and user-defined functions. The "temp" parameter always represents the simulator (circuit) temperature, and "tnom" always represents the measurement temperature. All expressions involving "temp" or "tnom" are re-evaluated any time the circuit temperature or measurement temperature changes.

You can also alter or sweep the "temp" and "tnom" parameters using any of the techniques available for altering or sweeping netlist or subcircuit parameters (with the exception of altergroups).

This capability allows you to write temperature dependent models for example, by using "temp" in an equation for a model or instance parameter. For example

```
r1 1 0 res r=(temp-tnom)*15+10k // temp is temperature
o1 options temp=55      // causes a change in above resistor r1
```

Reserved Parameters

The following parameters are reserved, and may not be declared as either top-level parameters or subcircuit parameters: temp, tnom, scale, scalem, freq, time.

Parameter Set - Block of Data (paramset)

Description

A parameter set is a block of data, which can be referenced by a sweep analysis. Within a paramset the first row contains an array of top-level netlist parameters. All other rows contain numbers which are used to alter the value of the parameters during the sweep. Each row represents an iteration of the sweep. This data should be bound within braces. The opening brace is required at the end of the line defining the paramset. The paramset cannot be defined within subcircuits or cannot be nested.

Definition

```
<Name> paramset {
  <list of parameter names>
  <list of number>
  [more rows of numbers]
}
```

Example

```
data paramset {
   p1  p2  p3
   1.1 2.2 3.3
   4.4 5.5 6.6
}
```

Output Selections (save)

Description

The `save` statement indicates that the values of specific nodes or signals should be saved in the output file. It works in conjunction with the `save` parameter on most analyses. The output file is written in Cadence Waveform Storage Format (WSF), Cadence Parameter Storage Format (PSF) or in Nutmeg/SPICE3 format as controlled by a a command line argument or a global option (see the options statement). The proper postprocessor should be used to view the output, generate plots, or do any further processing.

Definition

```
save <node|component|terminal> ...
```

This statement takes a list of signals as an argument. The concept of nodes has been generalized to signals where a signal is a value associated with a topological node of the circuit or some other unknown that is solved by the simulator, such as the current through a inductor or the voltage of the internal node in a diode. Topological nodes can be either at the top-level or in a subcircuit.

For example,

```
save 7 out OpAmp1.comp M1:currents D3:oppoint L1:1 R4:pwr
```

which tells that node `7`, node `out`, node `comp` in subcircuit `OpAmp1`, the currents through the terminals of `M1`, the oppoint information for diode D3, the current through the first terminal of

L1, and the instantaneous power dissipated by R4 should be saved. These outputs are saved in addition to any outputs specified with the `save` parameter on the analysis.

To specify a component terminal current, give the name of the component and the name or the index of the terminal separated by a colon. If `currents` is specified after the component and the colon, then all the terminal currents for the component are saved unless the component has only two terminals, in which case only the current through the first terminal is saved. Current is positive if it enters the terminal flowing into the component.

If a component name is followed by a colon and `oppoint`, then the operating point information associated with the component is computed and saved. If the colon is followed by an operating point parameter name (see each component for list of operating point parameters), then the value of that parameter is output.

If only a component name is given, all available information about the component, including the terminal currents and the operating point parameter values, is saved.

Sensitivity Analyses (sens)

Description

Use the `sens` control statement to find sensitivities of the output variables with respect to component and instance parameters for the list of the analyses performed. Currently DC and AC sensitivity analyses are supported. The results of the sensitivity analyses are stored in the output files written in Cadence Parameter Storage Format (PSF). In addition, you can use `+sensdata filename` command line argument or a global option (see the options statement) to direct sensitivity analyses results into a specified ASCII file.

Definition

```
sens (output_variables_list) to (design_parameters_list) for (analyses_list)
```

where

```
output_variables_list = ovar1 ovar2 ...
design_parameters_list = dpar1 dpar2 ...
analyses_list = anal1 anal2 ...
```

The list of the design parameters may include valid instance and model parameters. You can also specify device instances or device models without a modifier. In this case Spectre will attempt to compute sensitivities with respect to all corresponding instance or model parameters. Caution should be exercised in using this option as warnings or errors may be generated since many instance and model parameters cannot be modified. If no design parameters are specified then all the instance and model parameters are added. The list of the output variables for both AC and DC analyses may include node voltages and branch currents. For DC analyses, it also may include device instance operating point parameters.

Examples

```
sens (q1:betadc 2 Out) to (vcc:dc nbjt1:rb) for (analDC)
```

For this statement DC sensitivities of betadc operating point parameter of transistor `q1` and of nodes 2 and `Out` will be computed with respect to the `dc` voltage level of voltage source `vcc` and the model parameter `rb` for the DC analysis `analDC`. The results will be stored in the output file analDC.sens.dc.

```
sens (1 n2 7) to (q1:area nbjt1:rb) for (analAC)
```

For this statement AC sensitivities of nodes 1, n2, 7 will be computed with respect to the area parameter of transistor `q1` and the model parameter `rb` for each frequency of the AC analysis analAC. The results will be stored in the output file analAC.sens.ac.

```
sens (1 n2 7) for (analAC)
```

For this statement AC sensitivities of nodes 1, n2, 7 will be computed with respect to all instance and model parameters of all devices in the design for each frequency of the AC analysis analAC. The results will be stored in the output file analAC.sens.ac.

sens (vbb:p q1:int_c q1:gm 7) to (q1:area nbjt1:rb) for (analDC1)

For this statement DC sensitivities of the branch current `vbb:p`, the operating point parameter `gm` of the transistor `q1`, the internal collector voltage `q1:int_c` and the node `7` voltage will be computed with respect to the instance parameter `area` for instance `q1` and the model parameter `rb` for model `nbjt1`.

SpectreHDL Usage and Language Summary (spectrehdl)

Description

SpectreHDL is a proprietary analog hardware description language. It allows analog circuit behavior to be described at a high level of abstraction, using a language which is similar to Verilog-A (run `spectre -h veriloga` for some details on the Verilog-A modeling language supported by spectre). Behavioral descriptions of modules/components may be instantiated in a Spectre netlist along with regular Spectre primitives.

SpectreHDL descriptions are written in file(s) separate from the Spectre netlist file. These descriptions are written in modules (see the module alpha below). To include a module in the Spectre netlist, first add the line

```
ahdl_include "Ahdlfile.def"
```

to the Spectre netlist file (where Ahdlfile.def is the name of the file in which the required module is defined). The module is instantiated in the Spectre netlist in the same manner as Spectre primitives. For example,

```
name (node1 node2) alpha arg1=4.0 arg2=2 arg3="parameterized resistor"
```

This instantiates an element `alpha`, having two nodes and three parameters.

SpectreHDL modules can be debugged using `hdldebug`. `hdldebug` has a GUI and a command line mode. Please refer to the *Verilog-A Debugging Tool User Guide* for more information.

Module Template

The following is a SpectreHDL module template

```
module alpha( n1, n2 ) ( arg1, arg2, arg3 )
node [V,I] n1, n2;
parameter real arg1 = 2.0;
parameter integer arg2;
parameter string  arg3;
{
        real local;
        // this is a comment
        initial {
                // initializations performed before the
                // start of an analysis.
        }
        analog {
                // module behavioral description
                V(n1, n2) <- I(n1, n2) * arg1;
        }
        final {
                // tasks performed at the end of an analysis
        }
}
```

Language Summary

The following provides a summary of the SpectreHDL analog hardware description language. For more information refer to the *SpectreHDL Reference Manual*.

Derivative and Integral Operators

`dot(x)`	Differentiate x wrt time.
`integ(x <, ic <, assert>>)`	Integrate x wrt time. Output = ic during DC analysis. assert causes the integration to be reset.
`idtmod(x, <ic <, modulus <, offset> > > )`	Circular Integration of x wrt time. Output = ic during DC analysis. Integration is performed with given offset and modulus if specified.

Built-In Mathematical Functions

abs(x)	Absolute value
floor(x)	Largest integer < *x*
ceil(x)	Smallest integer > *x*
ln(x)	Natural logarithm
log(x)	Decimal logarithm
exp(x)	Exponential
sqrt(x)	Square root
min(x, y)	Minimum
max(x, y)	Maximum
pow(x, y)	x to the power of y

Simulator Time-Step Control Functions

`$threshold(x, direction <, abstol <, reltol_factor>>)`	Set breakpoint when x crosses zero.
`$bound_step(max_step)`	Limit time step, (time step <= max_step).
`$break_point(target <, period>)`	Set breakpoints at time = target and at times = N*period + target if period is specified.
`$last_crossing(x, direction)`	Return time when expression last crossed zero in a given direction.

Waveform Filter Functions

`$transition(x <, delay <, trise <, tfall>>>)`	Specify details of signal transitions. For efficient simulation, it is recommended that x not be a continuous signal, i.e. a function of a probe. See the *SpectreHDL Reference* manual for further explanation of this issue.
`$slew(x <, SRpos <, SRneg>>)`	Model slew rate behavior.
`$tdelay(x, time_delay, max_delay)`	Response(t) = x(t - time_delay).
`$zdelay(x <, period <, ttransition <, sample offset time <, ic>>>>)`	Fixed period sample and hold function.
`$zi_nd(x, numer, denom, period <, ttransition <,sample offset time>>)`	z-domain filter function, numerator-denominator form.
`$zi_zd(x, zeros, denom, period <, ttransition <,sample offset time>>)`	z-domain filter function, zero-denominator form.
`$zi_np(x, numer, poles, period <, ttransition <,sample offset time>>)`	z-domain filter function, numerator-pole form.
`$zi_zp(x, zeros, poles, period <, ttransition <,sample offset time>>)`	z-domain filter function, zero-pole form.
`$laplace_nd(x, numer, denom, <, abstol > )`	s-domain filter function, numerator-denominator form.

`$laplace_zd(x, zeros, denom, <, abstol > )`	s-domain filter function, zero-denominator form.
`$laplace_np(x, numer, poles, <, abstol > )`	s-domain filter function, numerator-pole form.
`$laplace_zp(x, zeros, poles, <, abstol > )`	s-domain filter function, zero-pole form.

Noise Functions

`$white_noise( power <, tag > )`	Generates white noise with given power. Noise contributions with the same tag are combined for a module.
`$flicker_noise( power, exp <, tag > )`	Generates pink noise with given power at 1 Hz that varies in proportion to 1/f^exp. Noise contributions with the same tag are combined for a module.
`$noise_table( vector <, tag > )`	Generates noise where power is determined by linear interpolation from the given vector of frequency-power pairs. Noise contributions with the same tag are combined for a module.

AC Analysis Stimuli

`$ac_stim( <analysis_name <, mag > > )`	Small signal source of specified magnitude, active for given analysis.

Interpolation Functions

`$build_table(type, response, inVec1, sizeVec1 <, inVec2, sizeVec2 ...>)`	Build a table for B-Spline interpolation.

$interpolate(interp_table, v1<, v2 <, v3 <, v4 >>>)	Perform interpolation at given point.

Simulator IO Functions

$strobe("format string" <, arguments>)	Print data to stdout every time step.
$debug("format string" <, arguments>)	Print data to stdout every iteration.
$fstrobe(fptr, "format string" <, arguments>)	Print data to a file every time step.
$fdebug(fptr, "format string" <, arguments>)	Print data to a file every iteration.
$fread(fptr, "format string" <, arguments>)	Read data from a file.
$warning("format string" <, arguments>)	Warning message.
$error("format string" <, arguments>)	Error message. Abort analysis.
$fatal("format string" <, arguments>)	Fatal message. Abort simulation.
$fopen("filename", mode)	Open a file.
$fflush(fptr)	Flush a file to disk.
$fclose(fptr)	Close a file.
$popen("command", "mode")	Open a pipe with given command in given mode.
$pclose(fptr)	Close a pipe.
$read_table("filename", table_id <, comment_str<, delim_str> >)	Read from a file into a 2-D real array.
$write_table("filename", table_id <,delim_str>)	Write from a 2-D real array to a file.
$halt("string")	Halt the simulation, printing given string.
$system("system command")	Sends a command to the operating system.
$str("format_string" < ,arg1 < ,arg2 < ..etc> > >)	Create a string from arguments in given format.
$strcmp(str1, str2)	Compares two strings lexicographically.
$strtoint(int_as_str)	Converts a string, int_as_str, to an integer.
$strtoreal(real_as_str)	Converts a string, real_as_str, to a real.

$strcpy(des_str, src_str)	Copies src_str to des_src.
$strcat(des_str, src_str)	Appends src_str to des_src.
$strlen(str)	Returns the number of characters in str.
$substr(input_str, start_pos, end_pos)	Returns the substring of input_str between start_pos and end_pos.
$strstr(input_str, sub_str)	Returns the first position where sub_str is found in input_str.
$strchr(input_str, character)	Returns the first position where character is found in input_str.
$strrchr(input_str, character)	Returns the last position where character is found in input_str.
$strspn(input_str, span_set)	Returns the number of continuous characters from the start of input_str that are in span_set.
$strcspn(input_str, span_set)	Returns the number of continuous characters from the start of input_str that are not in span_set.
$ascii(character)	Returns the ascii code of character.

Simulator Environment Functions

$time()	Returns current simulation time.
$temp()	Returns ambient simulation temperature.
$vt(<temp>)	Returns thermal voltage. If temp is defined, returns the thermal voltage at temp.
$analysis(analysis_string1<, analysis_string2 <, ...>>)	Returns true(1) if the current analysis phase matches one of the given analyses strings. Valid analyses strings are dc, tran, ac, pss, noise, pdisto, pac, pnoise, pxf, sp, tdr, xf, static, or ic.

Simulator Tolerance Functions

`$reltol()`	Returns relative tolerance.
`$abstol(`*`name`*`)`	Returns absolute tolerance of quantity *`name`*.

Parameter Functions

`$param_given(`*`param`*`)`	Returns 1 if *`param`* was set. *`param`* can be a model parameter or an instance parameter.
`$pwr(` *`x`* `)`	Assignment of model power consumption. Adds the expression *`x`* to the `pwr` parameter of a module.

Data Types

`integer`	Discrete numerical type.
`real`	Continuous numerical type.
`string`	Text string type.
`stream`	File pointer and text stream type.
`enum { name1 <, name2 <, name3 <, ... >>> }`	Discrete name type.
`void`	Null or empty type.
`table`	Interpolation table type.
`node [ PotentialName, FlowName ]`	Interconnection point type.

Data Qualifiers

`parameter`	Indicates that a variable is a parameter and so may be given a different value when the module is instantiated and may have a range specifier.
`const`	Indicates that a variable must be given a constant value when declared that can never be changed.

`global`	Used only for internal nodes. Means that the internal node is an alias for a global node of the same name in the netlist.

Structural Statements

Structural statements are used inside the module block but outside the analog, final and initial blocks

`model module_or_primative new_model (<param1 = expr1 <,..>>);`	Used to create a model called new_model from module_or_primative.
`module_or_primative inst_name (<node1 <, ..>> )(<param1 = expr1 <,..>>);`	Creates a new instance of module_or_primative called inst_name.

SpectreRF Summary (spectrerf)

Description

SpectreRF is a optional collection of analyses that are useful for circuits that are driven with a large periodic signal. Examples include mixers, oscillators, switched-capacitor filters, sample-and-holds, chopper stabilized amplifiers, frequency multipliers, frequency dividers, and samplers. They efficiently and directly compute the periodic and quasiperiodic steady-state solution of such circuits. They are capable of computing the large-signal and small-signal behavior, including noise behavior. Thus, SpectreRF is capable of computing the noise figure or intermodulation distortion of a mixer, the phase noise and harmonic distortion of an oscillator, and the frequency-response and noise behavior of a switched-capacitor filter. For more information on the SpectreRF analyses, run `spectre -help analysisName` where analysisName is `pss`, `pac`, `pxf`, `pnoise`, `pdisto`, or `envlp`.

Subcircuit Definitions (subckt)

Description

Hierarchical Circuit Description

The `subckt` statement is used to define a subcircuit. Subcircuit definitions are simply circuit macros that can be expanded anywhere in the circuit any number of times. When an instance in your input file refers to a subcircuit definition, the instances specified within the subcircuit are inserted into the circuit. Subcircuits may be nested. Thus a subcircuit definition may contain instances of other subcircuits. Subcircuits may also contain component, analysis or model statements. Subcircuit definitions may also be nested, in which case the innermost subcircuit definition can only be referenced from within the subcircuit in which it is defined, and cannot be referenced from elsewhere.

Instances that instantiate a subcircuit definition are referred to as subcircuit calls. The node names (or numbers) specified in the subcircuit call are substituted, in order, for the node names given in the subcircuit definition. All instances that refer to a subcircuit definition must have the same number of nodes as are specified in the subcircuit definition and in the same order. Node names inside the subcircuit definition are strictly local unless declared otherwise in the input file with a global statement.

Subcircuit Parameters

Parameter specification in subcircuit definitions is optional. In the case of nested subcircuit definitions, any parameters which have been declared for the outer subcircuit definition are also available within the inner subcircuit definition. Any parameters that are specified are referred to by name optionally followed by an equals sign and a default value. If, when making a subcircuit call, you do not specify a particular parameter, this default value is used in the macro expansion. Subcircuit parameters can be used in expressions within the subcircuit consisting of subcircuit parameters,

constants, and various mathematical operators. Run `spectre -h expressions` for more details on Spectre expression handling capability. Run `spectre -h parameters` for more details on how Spectre handles netlist parameters, including subcircuit parameters, and how they inherit within nested subcircuit definitions.

Subcircuits always have an implicitly defined parameter `m`. This parameter is passed to all components in the subcircuit and each component is expected to multiply it by its own multiplicity factor. In this way, it is possible to efficiently model several copies of the subcircuit in parallel. It is an error to attempt to explicitly define `m` on a `parameters` line. Also, because `m` is only implicitly defined, it is not available for use in expressions in the subcircuit.

Inline Subcircuits

An inline subckt is a special case of a subckt where one of the devices or models instantiated within this subckt does not get its full hierarchical name, but rather inherits the subckt call name itself. An inline subckt is syntactically denoted by the presence of the keyword `inline` before the `subckt`. It is called in the same manner as a regular subcircuit. The body of the inline subcircuit can typically contain one of the following, corresponding to different use models:

- multiple device instances (one of which is the "inline" component)
- multiple device instances, (one of which is "inline") and one or more parameterized models
- a single "inline" device instance and a parameterized model to which the device instance refers

The `inline` component is denoted by giving it the same name as the inline subcircuit itself. When the subcircuit is flattened, the `inline` component does not take on a hierarchical name such as X1.M1, but rather takes on the name of the subckt call itself, such as X1. Any non-inline components in the subckt take on the regular hierarchical name, just as if the subcircuit were a regular subckt (i.e. not an inline subckt).

Probing the Inline Device

Spectre allows the following list of items to be saved or probed for primitive devices. These would also apply to devices modeled as the inline components of inline subcircuits:

1. all terminal currents e.g. `save q1:currents`"

2. specific (index) terminal current e.g. `save q1:1` (#1=collector)

3. specific (named) terminal current e.g. `save q1:b` ("b"=base)

4. save all operating point info e.g. `save q1:oppoint`

5. save specific operating point info e.g. save `q1:vbe`

6. save all currents and oppoint info e.g. `save q1`

Parameterized Models and Inline Subckts

Inline subckts can be used in the same way as regular subcircuits to implement parameterized models, however inline subckts provide some powerful new options. When an inline subcircuit contains both a parameterized model and an inline device which references that model, then the user can create instances of the device, and each device will automatically get an appropriately scaled model assigned to it. For example, the instance parameters to an inline subckt could represent something like emitter width and length of a BJT device and within the subckt a model card could be created which is parameterized for emitter width and length and scales accordingly. When the designer instantiates the macro, he/she supplies the values for the emitter width and length, and a device is instantiated with an appropriate geometrically scaled model. Again, the inline device does not get a hierarchical name, and can be probed in the same manner as the inline device in the previous section on modeling parasitics, that is, it can be probed just as if it was a simple device, and not actually embedded in a subckt.

Automatic Model Selection using Inline Subckts

See `spectre -h if` for a description on how to combine Spectre's "structural if" statement with inline subckts to perform automatic model selection based on *any* netlist/subckt parameter.

Definition

```
[inline] subckt <Name> (<node1> ... <nodeN>)
    [parameters <name1>=<value1> ... <nameN>=<valueN>]
        ...
        <component, analysis, and/or model statements>
        ...

ends [Name]
```

Example 1: subckt

```
subckt coax (i1 o1 i2 o2)
 parameters zin=50 zout=50 vin=1 vout=1 len=0
  inner i1 o1 i2 o2  tline z0=zin vel=vin len=len
  outer o1 0  o2 0   tline z0=zout vel=vout len=len
ends coax
```

defines a parameterized coaxial transmission line macro from two ideal transmission lines. To instantiate this subcircuit, one could use an instance statement such as:

```
Coax1 pin nin out gnd coax zin=75 zout=150 len=35m
```

Example 2: inline subckt - Parasitics

Consider the following example of an inline subcircuit, which contains a mosfet instance, and two parasitic capacitances:

```
inline subckt s1 (a b)                // "s1" is name of subckt
 parameters p1=1u p2=2u
```

```
 s1 (a b 0 0) mos_mod l=p1 w=p2       // "s1" is "inline" component
 cap1 (a 0) capacitor c=1n
 cap2 (b 0) capacitor c=1n
ends s1
```

The following circuit creates a simple mos device instance M1, and calls the inline subcircuit `s1` twice (M2 & M3)

```
M1 (2 1 0 0) mos_mod

M2 (5 6) s1 p1=6u p2=7u

M3 (6 7) s1
```

This expands/flattens to:

```
M1 (2 1 0 0) mos_mod

M2 (5 6 0 0) mos_mod l=6u w=7u  // the "inline" component, inherits
call name

M2.cap1 (5 0) capacitor c=1n    // a regular hierarchical name

M2.cap2 (6 0) capacitor c=1n

M3 (6 7 0 0) mos_mod l=1u w=2u  // the "inline" component, inherits
call name

M3.cap1 (6 0) capacitor c=1n

M3.cap2 (7 0) capacitor c=1n
```

Here the final flattened names of the three mosfets (one for each instance) are M1, M2 and M3, rather than M1, M2.s1 and M3.s1 as they would be if s1 was a regular subcircuit. The parasitic capacitors (which the user is not really interested in, or perhaps even aware of, if the inline subckt definition was written by a separate modeling engineer) have full hierarchical names however.

Example 3: inline subckt - Scaled Models

Consider the following example, in which a parameterized model is declared within an inline subcircuit for a bipolar transistor. The model parameters are the emitter width, length, and area, and also the temperature delta (trise) of the device above nominal. Ninety nine instances of a 4*4 transistor are then placed, and one instance of a transistor with area=50 is placed. Each transistor gets an appropriately scaled model.

```
* declare a subckt, which instantiates a transistor with
* a parameterized model. The parameters are emitter width
*and length.
inline subckt bjtmod (c b e s)
parameters le=1u we=2u area=le*we trise=0
model mod1 bjt type=npn bf=100+(le+we)/2*(area/1e-12)
+       is=1e-12*(le/we)*(area/1e-12)
bjtmod (c b e s) mod1 trise=trise       // "inline" component
ends bjtmod
* some instances of this subckt
q1 (2 3 1 0) bjtmod le=4u we=4u         // trise defaults to zero
q2 (2 3 2 0) bjtmod le=4u we=4u trise=2
q3 (2 3 3 0) bjtmod le=4u we=4u
.
.
q99 (2 3 99 0) bjtmod le=4u we=4u
q100 (2 3 100 0) bjtmod le=1u area=50e-12
```

Verilog-A Usage and Language Summary (veriloga)

Description

Verilog-A is an analog hardware description language standard from Open Verilog International. It allows analog circuit behavior to be described at a high level of abstraction, using a language which is similar to SpectreHDL (run `spectre -h spectrehdl` for some details on the SpectreHDL modeling language). Behavioral descriptions of modules/components may be instantiated in a Spectre netlist along with regular Spectre primitives. For more information about using the SpectreHDL product, see the *SpectreHDL Reference* manual. For more information about using Verilog-A, see the *CadenceVerilog-A Language Reference* manual.

Verilog-A descriptions are written in file(s) separate from the Spectre netlist file. These descriptions are written in modules (see the module alpha below). To include a module in the Spectre netlist, first add the line

```
ahdl_include "VerilogAfile.va"
```

to the Spectre netlist file (where VerilogAfile.va is the name of the file in which the required module is defined). The module is instantiated in the Spectre netlist in the same manner as Spectre primitives for example,

```
name (node1 node2) alpha arg1=4.0 arg2=2
```

This instantiates an element `alpha`, having two nodes and two parameters.

Verilog-A modules can be debugged using *hdldebug*. *hdldebug* has a GUI and a command line mode. Please refer to the *Verilog-A Debugging Tool User Guide* for more information.

Module Template

The following is a Verilog-A module template

```
include "discipline.h"
include "constants.h"
module alpha( n1, n2 );
electrical n1, n2;
parameter real arg1 = 2.0;
parameter integer arg2 = 0;
 real local;
 // this is a comment
 analog begin
  @ ( initial_step ) begin
   // performed at the first timestep of an analysis
   end
  // module behavioral description
   V(n1, n2) <+ I(n1, n2) * arg1;
  @ ( final_step ) begin
   // performed at the last time step of an analysis
   end
 end
endmodule
```

Language Summary

The following provides a summary of the Verilog-A analog hardware description language. For more information refer to the *Verilog-A Language Reference* manual.

Analog Operators/Waveform Filters

`ddt(x <,abstol> )`	Differentiate x wrt time.
`idt(x, ic <, assert <, abstol> > )`	Integrate x wrt time. Output = ic during dc analysis and when assert is 1.
`idtmod(x, <ic <, modulus <, offset> > > )`	Circular Integration of x wrt time. Output = ic during DC analysis. Integration is performed with given offset and modulus if specified.

`transition(x <, delay <, trise <, tfall>>>)`	Specify details of signal transitions. For efficient simulation, it is recommended that x not be a continuous signal, i.e. a function of a probe. See the *Cadence Verilog-A Language Reference* manual for further explanation of this issue.
`slew(x <, SRpos <, SRneg>>)`	Model slew rate behavior.
`delay(x, time_delay, max_delay)`	Response(t) = x(t - time_delay).
`zi_nd(x, numer, denom, period, < ttransition <,sample offset time > )`	z-domain filter function, numerator-denominator form.
`zi_zd(x, zeros, denom, period, < ttransition <,sample offset time > )`	z-domain filter function, zero-denominator form.
`zi_np(x, numer, poles, period, < ttransition <,sample offset time > )`	z-domain filter function, numerator-pole form.
`zi_zp(x, zeros, poles, period, < ttransition <,sample offset time > )`	z-domain filter function, zero-pole form.
`laplace_nd(x, numer, denom, <, abstol > )`	s-domain filter function, numerator-denominator form.
`laplace_zd(x, zeros, denom, <, abstol >`	s-domain filter function, zero-denominator form.
`laplace_np(x, numer, poles, <, abstol > )`	s-domain filter function, numerator-pole form.
`laplace_zp(x, zeros, poles, <, abstol > )`	s-domain filter function, zero-pole form.

Built-In Mathematical Functions

abs(x)	Absolute value
exp(x)	Exponential if x < 80
ln(x)	Natural logarithm
log(x)	Log base 10
sqrt(x)	Square root
min(x,y)	Minimum
max(x,y)	Maximum

pow(x,y)	x to the power of y

Noise Functions

white_noise(power <, tag >)	Generates white noise with given power. Noise contributions with the same tag are combined for a module.
flicker_noise(power, exp <, tag >)	Generates pink noise with given power at 1 Hz that varies in proportion to 1/f^exp. Noise contributions with the same tag are combined for a module.
noise_table(vector <, tag >)	Generates noise where power is determined by linear interpolation from the given vector of frequency-power pairs. Noise contributions with the same tag are combined for a module.

AC Analysis Stimuli

ac_stim(<analysis_name <, mag > >)	Small signal source of specified magnitude, active for given analysis.

Analog Events

Analog events must be contained in an analog event detection statement; @(analog_event) statement.

cross(x, direction <, timetol <, abstol >>)	Generates an event when x crosses zero.
timer(start_time <, period>)	Set (optionally periodic) breakpoint event at time = start_time.
initial_step< (arg1 <, arg2 <, etc... > >)	Generate an event at the initial step of an analysis. arg1, arg2, etc. may be any of: dc, tran, ac, pss, noise, pdisto, pac, pnoise, pxf, sp, tdr, xf, static, or ic.

`final_step< ( arg1 <, arg2 <, etc... > > )`	Generate an event at the final step of an analysis. arg1, arg2, etc. may be any of:dc, `tran`, `ac`, `pss`, `noise`, `pdisto`, `pac`, `pnoise`, `pxf`, `sp`, `tdr`, `xf`, `static`, or `ic`.

Timestep Control

`bound_step(max_step)`	Limit timestep, (timestep <= max_step).
`last_crossing(x, direction)`	Return time when expression last crossed zero in a given direction.
`discontinuity(n)`	Hint to simulator that discontinuity is present in nth derivative.

Simulator IO Functions

`$display(argument_list)`	Print data to stdout. Formatting strings may be interspersed between arguments/data.
`$fdisplay(fptr, argument_list)`	Print data to a file. Formatting strings may be interspersed between arguments/data.
`$strobe(argument_list)`	Print data to stdout. Formatting strings may be interspersed between arguments/data.
`$fstrobe(fptr, argument_list)`	Print data to a file. Formatting strings may be interspersed between arguments/data.
`$fopen("filename")`	Open a file for writing.
`$fclose(fptr)`	Close a file.
`$finish<(n)>`	Finish the simulation.
`$stop<(n)>`	Stop the simulation.

Simulator Environment Functions

`$realtime`	Returns current simulation time.
`$temperature`	Returns ambient simulation temperature.

`$vt`	Returns thermal voltage.
`$vt(temp)`	Returns thermal voltage at given temp.
`$analysis(analysis_string1<, analysis_string2 <, ...>>)`	Returns true(1) if the current analysis phase matches one of the given analyses strings. Valid analyses strings are; `dc`, `tran`, `ac`, `pss`, `noise`, `pdisto`, `pac`, `pnoise`, `pxf`, `sp`, `tdr`, `xf`, `static`, or `ic`.

Parameter Functions

$pwr(x)	Assignment of model power consumption. Adds the expression x to the pwr parameter of a module.

Data Types

`integer`	Discrete numerical type.
`real`	Continuous numerical type.

Data Qualifiers

`parameter`	Indicates that a variable is a parameter and so may be given a different value when the module is instantiated, and that it may not be assigned a different value inside the module.

Structural Statements

`module_or_primative #(<.param1(expr1)<,...>>) inst_name (<node1 <, ..>> );`	Creates a new instance of module_or_primative called inst_name.

A

References

This section gives additional details about the source documents referred to in the text.

[antognetti88] Paolo Antognetti, Giuseppe Massobrio. *Semiconductor Device Modeling with SPICE*. McGraw-Hill, New York, 1988.

[gear71] C. William Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, 1971.

[hammerstad80] E. Hammerstad, O. Jensen. “Accurate models for microstrip computer-aided design.” *IEEE MTT-S 1980 International Microwave Symposium Digest*, pages 407-409.

[jansen83] Rolf H. Jansen, Martin Kirschning. “Arguments and an accurate model for the power-current formulation of microstrip characteristic impedance.” *Arch. Elek. Ubertragung* (AEU), vol. 37, 1983, pages 108-112.

[kirschning82] M. Kirschning, R. H. Jansen. “Accurate model for effective dielectric constant of microstrip with validity up to millimetre-wave frequencies.” *Electronic Letters*, vol. 18, no. 6, 18 March 1982, pages 272-273.

[kundert90]Kenneth S. Kundert, Jacob K. White, Alberto Sangiovanni-Vincentelli. *Steady-State Methods for Simulating Analog and Microwave Circuits*. Kluwer Academic Publishers, 1990.

[nagel75]Laurence W. Nagel. *SPICE2: A Computer Program to Simulate Semiconductor Circuits*. Ph. D. dissertation, University of California at Berkeley, May 1975. Available through Electronics Research Laboratory Publications, U. C. B., 94720; Memorandum No. UCB/ERL M520.

[quarles89] Thomas L. Quarles. *Analysis of Performance and Convergence Issues for Circuit Simulation*. Ph. D. dissertation, University of California at Berkeley, April 1989. Extensively documents the Spice3 program. Available through Electronics Research Laboratory Publications, U. C. B., 94720; Memorandum No. UCB/ERL M89/42.

[statz87]Hermann Statz, Paul Newman, Irl W. Smith, Robert A. Pucel, Hermann A. Haus. “GaAs FET device and circuit simulation in SPICE.” *IEEE Transactions on Electron Devices*, vol. ED-34, no. 2, pages 160-169, February 1987.

[vladimirescu81]A. Vladimirescu, Kaihe Zhang, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli. *SPICE Version 2G User’s Guide*, August 1981. Available through Industrial Liaison Program Software Distribution office, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 94720.

[yang82]Ping Yang, Pallab K. Chatterjee. “SPICE modeling for small geometry MOSFET circuits.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-1, no. 4, pages 169-182, October 1982.