SANDIA REPORT

SAND2004-2896
Unlimited Release
Printed June 2004

Xyce™ Parallel Electronic Simulator Reference Guide, Version 2.0

Eric R. Keiter, Scott A. Hutchinson, Robert J. Hoekstra, Lon J. Waters, Thomas V. Russo, Eric L. Rankin, Roger P. Pawlowski, Deborah A. Fixel and Steven D. Wix

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2004-2896
Unlimited Release
Printed June 2004

Xyce™ Parallel Electronic Simulator

Reference Guide, Version 2.0

Eric R. Keiter, Scott A. Hutchinson, Robert J. Hoekstra,
Eric L. Rankin, Roger P. Pawlowski, Deborah A. Fixel
Computational Sciences

Thomas V. Russo, Lon J. Waters and Steven D. Wix
Component Information and Models

Sandia National Laboratories
P.O. Box 5800
Mail Stop 0316
Albuquerque, NM 87185-0316

June 30, 2004

Abstract

This document is a reference guide to the **Xyce** Parallel Electronic Simulator, and is a companion document to the **Xyce** Users' Guide. The focus of this document is (to the extent possible) exhaustively list device parameters, solver options, parser options, and other usage details of **Xyce**. This document is *not* intended to be a tutorial. Users who are new to circuit simulation are better served by the **Xyce** Users' Guide.

Acknowledgements

The authors would like to acknowledge the entire Sandia National Laboratories HPEMS (High Performance Electrical Modeling and Simulation) team, including Carolyn Bogdan, Regina Schells, Ken Marx, Steve Brandon, David Shirley and Bill Ballard, for their support on this project. We also appreciate very much the work of Becky Arnold and Mike Williamson for the help in reviewing this document.

Lastly, a very special thanks to Hue Lai for typesetting this document with LaTeX.

Trademarks

Contacts

Bug Reports	http://tvrusso.sandia.gov/bugzilla
Email	xyce-support@sandia.gov
World Wide Web	http://www.cs.sandia.gov/xyce

Contents

Tables

1. Introduction

Welcome to **Xyce**

The **Xyce** Parallel Electronic Simulator has been written to support, in a rigorous manner, the simulation needs of the Sandia National Laboratories electrical designers. It is targeted specifically to run on large-scale parallel computing platforms but also runs well on a variety of architectures including single processor workstations. It also aims to support a variety of devices and models specific to Sandia needs.

1.1 Overview

This document is intended to compliment the **Xyce** Users' Guide [1]. It contains comprehensive, detailed information about a number of topics pertainent to the usage of **Xyce**. Included in this document is a netlist reference for the input-file commands and elements supported within **Xyce**; a command line reference, which describes the available command line arguements for **Xyce**; and quick-references for users of other circuit codes, such as Orcad's PSpice [2] and Sandia's ChileSPICE.

1.2 How to Use this Guide

This guide is designed so you can quickly find the information you need to use **Xyce**. It assumes that you are familiar with basic Unix-type commands, how Unix manages applications and files to perform routine tasks (e.g., starting applications, opening files and saving your work). Note that while Windows versions of **Xyce** are available, they are ported to run in the Cygwin environment, which has a Unix/Linux look and feel.

Typographical conventions

Before continuing in this Reference Guide, it is important to understand the terms and typographical conventions used. Procedures for performing an operation are generally numbered with the following typographical conventions.

Notation	Example	Description
`Verbatim text`	`xmpirun -np 4`	Commands entered from the keyboard on the command line or text entered in a netlist.
Bold Roman Font	Set nominal temperature using the **TNOM** option.	SPICE-type parameters used in models, etc.
Gray Shaded Text	DEBUGLEVEL	Feature that is designed primarily for use by **Xyce** developers.
`[text in brackets]`	`Xyce [options] <netlist>`	Optional parameters.
`<text in angle brackets>`	`Xyce [options] <netlist>`	Parameters to be inserted by the user.
`<object with asterisk>*`	`K1 <ind. 1> [<ind. n>*]`	Parameter that may be multiply specified.
`<TEXT1|TEXT2>`	`.PRINT TRAN` `+ DELIMITER=<TAB|COMMA>`	Parameters that may only take specified values.

Table 1.1. Xyce typographical conventions.

This page is left intentionally blank

2. Netlist Reference

Chapter Overview

This chapter contains reference material directed towards working with circuit analyses in **Xyce** using the netlist interface. Included in are detailed command descriptions, start-up option definitions and a list of devices supported by the **Xyce** netlist interface.

2.1 Netlist Commands

This section outlines the netlist commands that can be used with **Xyce** to setup and control circuit analysis.

Analysis Types

DC Sweep Analysis

Calculates the operating point for the circuit for a range of values for voltage sources. Note that this may be repeated for multiple voltage sources.

General Form

```
.DC <voltage source> <start> <stop> <step>
+ [<voltage source><start> <stop> <step>]...
```

Examples

```
.DC VIN -10 15 1

.DC VDS 0 3.5 0.05 VGS 0 3.5 0.5
```

Transient Analysis

Calculates the time-domain response of a circuit for a specified duration.

General Form

```
.TRAN <print step value> <final time value>
+[<start time value> [<step ceiling value>]] [NOOP]
```

Examples

```
.TRAN 1us 100ms

.TRAN 1ms 100ms 0ms .1ms
```

Arguments and Options

`<print step value>`

Used to calculate the initial time step (see below).

`<final time value>`

Sets the end time (duration) for the analysis.

`[<start time value>]`

Sets the time at which the simulation is to begin. Defaults to zero.

`[<step ceiling value>]`

Sets a maximum time step. Defaults to ((final time value)-(start time value))/10, unless there are breakpoints (see below).

`[NOOP]`

Specifies that no operating point calculation is to be performed.

The transient analysis calculates the circuit's response over an interval of time beginning with `TIME=0` or `TIME=<start time value>` and finishing at `<final time value>`. Use a `.PRINT` (print) statement to get the results of the transient analysis.

Before calculating the transient response **Xyce** computes a bias point for the circuit that is different from the regular bias point. This is necessary because at the start of a transient analysis, the independent sources can have different values than their DC values. Specifying `NOOP` on the `.TRAN` line causes **Xyce** to begin the transient analysis without performing the usual bias point calculation.

Comments

The time integration algorithms within **Xyce** use adaptive time-stepping methods that adjust the time-step size according to the activity in the analysis. The default ceiling for the internal time step is `(<final time value>-<start time value>)/10`. This default ceiling value is automatically adjusted if breakpoints are present, to insure that there are always at least 10 time steps between breakpoints. If the user specifies a ceiling value, however, it overrides any internally generated ceiling values.

Xyce is not strictly compatible with SPICE in its use of the values on the `.TRAN` line. In **Xyce**, `<print step value>` is not used as the printing interval. It is used in determining the initial step size, which is chosen to be the smallest of three quantities: the print step value, the step ceiling value, and 1/200th of the time until the next breakpoint.

STEP Parametric Analysis

Calculates a full analysis (`.DC` or `.TRAN`) over a range of parameter values.

General Form

```
.STEP <parameter name> <initial value> <final value> <step size value>
```

Examples

```
.STEP TEMP -45 -55 -10

.STEP R1 45 50 5

.STEP C101:C 45 50 5

.STEP DLEAK:IS 1.0e-12 1.0e-11 1.0e-12

.STEP V1 20 10 -1
```

Arguments and Options

`<initial value>`

Initial value for the parameter.

`<final value>`

Final value for the parameter.

`<step size value>`

Value that the parameter is incremented at each step.

STEP parameter analysis will sweep a parameter from its initial value to its final value, at increments of the step size. At each step of this sweep, it will conduct a full analysis (`.DC` or `.TRAN`) of the circuit.

The specification is similar to that of a `.DC` sweep, except that each parameter gets its own `.STEP` line in the input file, rather than specifying all of them on a single line.

Comments

Output, as designated by a `.PRINT` statement, is slightly more complicated in the case of a `.STEP` simulation. If the user has specified a `.PRINT` line in the input file, **Xyce** will output a separate *.prn file for each step of the sweep. The name of the input file will be the same as it would have been without the `.STEP` specification, except that the string "STEP?" will be inserted before the ".prn" suffix, where the "?" character is an integer representing the step number.

This is a similar capability to that of PSPICE, and ChileSPICE, but not identical. In future releases, this capability will be fully compatible.

Device Modeling

.MODEL (Model Definition)

Modeled device definition.

.SUBCKT (subcircuit)

The .SUBCKT statement begins a subcircuit definition by giving its name, the number and order of its nodes and the names and default parameters that direct its behavior. The .ENDS statement signifies the end of the subcircuit definition.

General Form

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

Examples

```
.SUBCKT OPAMP 10 12 111 112 13
...
.ENDS

.SUBCKT FILTER1 INPUT OUTPUT PARAMS: CENTER=200kHz,
+ BANDWIDTH=20kHz
...
.ENDS

.SUBCKT PLRD IN1 IN2 IN3 OUT1
+ PARAMS: MNTYMXDELY=0 IO_LEVEL=1
...
.ENDS

.SUBCKT 74LS01 A B Y
+ PARAMS: MNTYMXDELY=0 IO_LEVEL=1
...
.ENDS
```

Arguments and Options

`<name>`

The name used to reference a subcircuit.

`[node]*`

An optional list of nodes. This is not mandatory since it is feasible to define a subcircuit without any interface nodes.

`[PARAMS:]`

Keyword that provides values to subcircuits as arguments for use as expressions in the subcircuit. Parameters defined in the `PARAMS:` section may be used in expressions within the body of the subcircut and will take the default values specified in the subcircuit definition unless overridden by a `PARAMS:` section when the subcircuit is instantiated.

Comments

A subcircuit designation ends with a `.ENDS` command. The entire netlist between `.SUBCKT` and `.ENDS` is part of the definition. Each time the subcircuit is called via an `X` device, the entire netlist in the subcircuit definition replaces the `X` device.

There must be an equal number of nodes in the subcircuit call and in its definition. As soon as the subcircuit is called, the actual nodes (those in the calling statement) substitute for the argument nodes (those in the defining statement).

Node zero cannot be used in this node list, as it is the global ground node.

Subcircuit references may be nested to any level. However, their definitions cannot be nested. That is, a `.SUBCKT` statement cannot be placed between a `.SUBCKT` and a `.ENDS` statements.

Subcircuits should include only device instantiations and possibly these statements:

- `.MODEL` (model definition)
- `.PARAM` (parameter)
- `.FUNC` (function)

Models, parameters, and functions defined within a subcircuit are scoped to that definition. That is they are only accessible within the subcircuit definition in which they are included. Further, if a `.MODEL`, `.PARAM` or a `.FUNC` statement is included in the main circuit netlist, it is accessible from the main circuit as well as all subcircuits.

Node, device, and model names are scoped to the subcircuit in which they are defined. It is allowable to use a name in a subcircuit that has been previously used in the main circuit netlist. When the subcircuit is flattened (expanded into the main netlist), all of its names are given a prefix via the subcircuit instance name. For example, `Q17` becomes `X3:Q17` after expansion. After expansion, all names are unique. The single exception occurs in the use of global node names, which are not expanded.

`.ENDS` (end subcircuit)

Marks the end of a subcircuit definition.

Output Control

.PRINT (print)

Send analysis results to an output file. **Xyce** supports several options on the .PRINT line of netlists:

General Form

```
.PRINT <analysis type> [FORMAT=<STD|NOINDEX|PROBE|TECPLOT>]
+ [FILE=<output filename>] [WIDTH=<print field width>]
+ [PRECISION=<floating point output precision>]
+ [FILTER=<absolute value below which a number outputs as 0.0>]
+ [DELIMITER=<TAB|COMMA>]
+ <output variable>
```

Examples

```
.PRINT TRAN FORMAT=PROBE FILE=foobar.csd V(1)

.PRINT DC FILE=foobar.txt WIDTH=19 PRECISION=15 FILTER=1.0e-10
+ I(VSOURCE5) I(VSOURCE6)

R1 1 0 100
X1 1 2 3 MySubcircuit
V1 3 0 1V
.SUBCKT MYSUBCIRCUIT 1 2 3
R1 1 2  100K
R2 2 4  50K
R3 4 3  1K
.ENDS
.PRINT DC V(X1:4) V(2) I(V1)
```

Arguments and Options

`<analysis type>`

Only one analysis type (DC or TRAN) may be given for each `.PRINT` netlist entry.

`[FORMAT=<STD|NOINDEX|PROBE|TECPLOT>]`

The output format may be specified using the `FORMAT` option. The `STD` format outputs the data divided up into data columns. The `NOINDEX` format is the same as the `STD` format except that the index column is omitted. The `PROBE` format specifies that the output should be formatted to be compatible with the PSpice Probe plotting utility. The `TECPLOT` format specifies that the output should be formatted to be compatible with the Tecplot plotting program.

`[FILE=<output filename>]`

Specifies the name of the file to which the output will be written.

`[WIDTH=<print field width>]`

Controls the output width used in formatting the output.

`[PRECISION=<floating point precision >]`

Number of floating point digits past the decimal for output data.

`[FILTER=<filter floor value>]`

Used to specify the absolute value below which output variables will be printed as `0.0`.

`[DELIMITER=<TAB|COMMA>]`

Used to specify an alternate delimiter in the STD or NOINDEX format output.

`<output variable>`

Following the analysis type and other options is a list of output variables. There is no upper bound on the number of these output variables. The output is divided up into data columns and output according to any specified options (see options given above). Each output variable must be of the form `V(<circuit node>)` to output the voltage at a point in the circuit or `I(<independent source>)` to output the current through an independent source. **Xyce** does not support output of currents through arbitrary devices.

`<circuit node>` is simply the name of any node in your top-level circuit, or `<subcircuit name>:<node>` to reference nodes that are internal to a subcircuit.

Comments

The values of the output variables are output as a series of columns (one for each output variable).

Netlist Processing

.END (End of Circuit)

End of netlist file.

.FUNC (function)

User defined functions that can be used in expressions appearing later in the same scope as the .FUNC statement.

General Form

```
.FUNC <name>([arg]*) <body>
```

Examples

```
.FUNC E(x) {exp(x)}
.FUNC DECAY(CNST) {E(-CNST*TIME)}
.FUNC TRIWAV(x) {ACOS(COS(x))/3.14159}
.FUNC MIN3(A,B,C) {MIN(A,MIN(B,C))}
```

Arguments and Options

.FUNC

Must precede the first use of the function name. Functions cannot be redefined and the function name must not be the same as any of the predefined functions (e.g., SIN and SQRT). .FUNC arguments cannot be node names.

<body>

May refer to other (previously defined) functions; the second example, DECAY, uses the first example, E.

[arg]

The number of arguments in the use of a function must agree with the number in the definition. Parameters, TIME, and other functions are allowed in the body of function definitions.

Comments

The <body> of a defined function is handled in the same way as any math expression; it must be enclosed in curly braces .

.INC or .INCLUDE (include file)

Include specified file in netlist.

The file name can be surrounded by double quotes, "filename", but this is not necessary. The directory for the include file is assumed to be the execution directory unless a full or

relative path is given as a part of the file name.

.PARAM (parameter)

User defined parameter that can be used in expressions throughout the netlist.

General Form

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

Examples

```
.PARAM A_Param=1K
.PARAM B_Param={A_Param*3.1415926535}
```

Miscellaneous Commands

* (Comment)

Create a netlist comment line.

; (In-line Comment)

Add a netlist in-line comment

+ (Line Continuation)

Continue the text of the previous line

.OPTIONS Statements

.OPTIONS (Analysis Options)

Set various simulation limits, analysis control parameters and output characters. In general, they use the following format:

```
.OPTIONS <PKG> [<TAG>=<VALUE>]*
```

Exceptions to this format include the OUTPUT and RESTART options that use their own format, which will be defined under their respective descriptions.

The designator PKG refers to the Unified Modeling Language (UML) *package* which refers loosely to a *module* in the code. Thus, the term is used here as identifying a specific *module* to be controlled via *options* set in the netlist input file. The packages which currently support .OPTIONS, and the keywords to use in place of <PKG> are:

Package	PKG keyword
Global:	GLOBAL
Device Model:	DEVICE
Time Integration:	TIMEINT
Nonlinear Solver:	NONLIN
Transient Nonlinear Solver:	NONLIN-TRAN
Continuation/Bifurcation Tracking:	LOCA
Linear Solver:	LINSOL
Parallel Distribution:	PARALLEL
Output:	OUTPUT
Restart:	RESTART

As an example, the following netlist line will set the value of **ABSTOL** in the time integration package to 1×10^{-8}:

Example: .OPTIONS TIMEINT ABSTOL=1E-8

Below is an outline of the supported packages and their respective options:

Device Package Options

The device package parameters listed in Table 2.1 outline the options available for specifying device specific parameters. Some of these (DEFAS, DEFAD, TNOM etc.) have the same meaning as they do for the .OPTION line from Berkeley SPICE (3f5). Parameters which are apply globally to all device models will tend to be specified here. Parameters specific to a particular device instance or model are specified elsewhere.

Device Model (PKG = DEVICE) Tag	Description	Default
DEFAS	MOS Drain Diffusion Area	0.0
DEFAD	MOS Source Diffusion Area	0.0
DEFL	MOS Default Channel Length	1.0E-4
DEFW	MOS Default Channel Width	1.0E-4
GMIN	Minimum Conductance	1.0E-12
TEMP	Temperature	26.85°C (300K)

Device Model (PKG = DEVICE) Tag	Description	Default
TNOM	Nominal Temperature	26.85°C (300K)
SCALESRC	Scaling factor for source scaling	0.0
NUMJAC	Numerical Jacobian flag (only use for small problems)	0 (FALSE)
VOLTLIM	Voltage limiting	1 (TRUE)
icFac	This is a multiplicative factor which is applied to right-hand side vector loads of .IC initial conditions during the DCOP phase.	10000.0
LAMBERTW	This flag determines if the lambert-W function should be applied in place of exponentials in hard-to-solve devices. Currently, this capability is implemented in the diode and BJT. Try this for BJT circuits that have convergence problems. For best effect, this option should be tried with voltlim turned off. A detailed explanation of the Lambert-W function, and its application to device modeling can be found in reference [3].	0 (FALSE)
MAXTIMESTEP	Maximum time step size	1.0E+99
DEBUGLEVEL	The higher this number, the more info is output	1
DEBUGMINTIMESTEP	First time-step debug information is output	0

Device Model (PKG = DEVICE) Tag	Description	Default
DEBUGMAXTIMESTEP	Last time-step of debug output	65536
DEBUGMINTIME	Same as DEBUGMINTIMESTEP except controlled by time (sec.) instead of step number	0.0
DEBUGMAXTIME	Same as DEBUGMAXTIMESTEP except controlled by time (sec.) instead of step number	100.0

Table 2.1: Options for Device Package

Time Integration Options

The time integration parameters listed in Table 2.2 give the available options for helping control the time integration algorithms for transient analysis.

Time Integration (PKG = TIMEINT) Tag	Description	Default
METHOD	Time integration method. This parameter is only relevant when running **Xyce** in transient mode. Supported methods: ■ 1 (Backward Euler) ■ 2 (Backward Differentiation 2) ■ 3 (Trapezoidal)	1 (Backward Euler)
CONSTSTEP	Constant time step flag. If set to true, the code will use a constant time step in transient mode.	0 (FALSE: use variable time step)
RELTOL	Relative error tolerance	1.0E-02
ABSTOL	Absolute error tolerance	1.0E-06

Time Integration (PKG = TIMEINT) Tag	Description	Default
DOUBLEDCOPSTEP	This option should only be set to TRUE for a PDE device run. PDE devices often have to solve an extra "setup" problem to get the initial condition. This extra setup problem solves a nonlinear poisson equation (see the device appendix for more details), while the normal step solves a full drift-diffusion(DD) problem. The name of this flag refers to the fact that the code is essentially taking two DC operating point steps instead of one. If you set this to TRUE, but have no PDE devices in the circuit, the code will repeat the same identical DCOP step twice. Generally there is no point in doing this.	• 0 (FALSE), if no PDE devices are present. • 1(TRUE) if at least one PDE device is present in the circuit.
FIRSTDCOPSTEP	This is the index of the first DCOP step taken in a simulation for which DOUBLEDCOPSTEP is set to TRUE. The special initialization (nonlinear Poisson) step is referred to as step 0, while the normal (drift-diffusion) step is indexed with a 1. These two options(FIRSTDCOPSTEP and LASTDCOPSTEP) allow you to set the 1st or second DCOP step to be either kind of step. If FIRSTDCOPSTEP and LASTDCOPSTEP are both set to 0, then only the initial setup step happens. If FIRSTDCOPSTEP and LASTDCOPSTEP are both set to 1, then the initialization step doesn't happen, and only the real DD problem is attempted, with a crude initial guess. You should *never* set FIRSTDCOPSTEP to 1 and SECONDDCOPSTEP to 0. Normally, they should always be left as the defaults.	0

Time Integration (PKG = TIMEINT) Tag	Description	Default
LASTDCOPSTEP	This is the second step taken in a simulation for which DOUBLEDCOPSTEP is set to TRUE.	1
DOUBLEDCOPALL	This flag should almost always be false. If it is true, the code will take an extra "setup" step at every step, including transient time steps. The nonlinear poisson equation is thus solved every step to give an initial guess for the current step. This has turned out to not work very well, and may be removed as an option later.	0 (FALSE)
RESETTRANNLS	The nonlinear solver resets its settings for the transient part of the run to something more efficient (basically a simpler set of options with smaller numbers for things like max Newton step). If this is set to false, this resetting is turned off. Normally should be left as default.	1 (TRUE)
BPENABLE	Flag for turning on/off breakpoints (1 = ON, 0 = OFF). It is unlikely anyone would ever set this to FALSE, except to help debug the breakpoint capability.	1 (TRUE:breakpoints are enabled)

Time Integration (PKG = TIMEINT) Tag	Description	Default
EXITTIME	If this is set to nonzero, the code will check the simulation time at the end of each step. If the total time exceeds the exittime, the code will ungracefully exit. This is a debugging option, the point of which is the have the code stop at a certain time during a run without affecting the step size control. If not set by the user, it isn't activated.	-
EXITSTEP	Same as EXITTIME, only applied to step number. The code will exit at the specified step. If not set by the user, it isn't activated.	-

Table 2.2: Options for Time Integration Package.

Nonlinear Solver Options

The nonlinear solver parameters listed in Table 2.3 provide methods for controlling the nonlinear solver for DC, Transient. Note that the transient parameters for the nonlinear solver operated under the time integrator may be overridden using the Transient Nonlinear Solver Package Options below.

Nonlinear Solver (PKG = NONLIN) Tag	Description	Default
NLSTRATEGY	Nonlinear solution strategy. Supported Strategies: ■ 0 (Newton) ■ 1 (Gradient) ■ 2 (Trust Region)	0 (Newton)

Nonlinear Solver (PKG = NONLIN) Tag	Description	Default
SEARCHMETHOD	Line-search method used by the nonlinear solver. Supported line-search methods: ■ 0 (Full Newton - no line search) ■ 1 (Interval Halving) ■ 2 (Quadratic Interpolation) ■ 3 (Cubic Interpolation) ■ 4 (More'-Thuente)	0 (Full Newton) (NOTE: for iterative linear solves, the default is Quadratic Linesearch - 2)
ABSTOL	Absolute residual vector tolerance	1.0E-12
RELTOL	Relative residual vector tolerance	1.0E-03
DELTAXTOL	Weighted nonlinear-solution update norm convergence tolerance	1.0
RHSTOL	Residual convergence tolerance (unweighted 2-norm)	1.0E-06
MAXSTEP	Maximum number of Newton steps	200
MAXSEARCHSTEP	Maximum number of line-search steps	9
NORMLVL	Norm level used by the nonlinear solver algorithms *(NOTE: not used for convergence tests)*	2
LINOPT	Linear optimization flag	0 (FALSE)
CONSTRAINTBT	Constraint backtracking flag	0 (FALSE)
CONSTRAINTMAX	Global maximum setting for constraint backtracking	DBL_MAX (Machine Dependent Constant)
CONSTRAINTMIN	Global minimum setting for constraint backtracking	-DBL_MAX (Machine Dependent Constant)
CONSTRAINTCHANGE	Global percentage-change setting for constraint backtracking	sqrt(DBL_MAX) (Machine Dependent Constant)
IN_FORCING	Inexact Newton-Krylov forcing flag	1 (TRUE)
AZ_TOL	Sets the minimum allowed linear solver tolerance. Valid only if IN_FORCING=1.	1.0E-12
DIRECTLINSOLV	Direct (vs. iterative) linear solution flag *(Serial Only!)*	0 (FALSE)

Nonlinear Solver (PKG = NONLIN) Tag	Description	Default
DLSDEBUG	Debug output for direct linear solver	0 (FALSE)
DEBUGLEVEL	The higher this number, the more info is output	1
DEBUGMINTIMESTEP	First time-step debug information is output	0
DEBUGMAXTIMESTEP	Last time-step of debug output	99999999
DEBUGMINTIME	Same as DEBUGMINTIMESTEP except controlled by time (sec.) instead of step number	0.0
DEBUGMAXTIME	Same as DEBUGMAXTIMESTEP except controlled by time (sec.) instead of step number	1.0E+99
RECOVERYSTEPTYPE	If using a line search, this option determines the type of step to take if the line search fails. Supported strategies: ■ 0 (Take the last computed step size in the line search algorithm) ■ 1 (Take a constant step size set by RECOVERYSTEP)	0
RECOVERYSTEP	Value of the recovery step if a constant step length is selected	1.0

Nonlinear Solver (PKG = NONLIN) Tag	Description	Default
CONTINUATION	Enables the use of Homotopy/Continuation algorithms for the nonlinear solve. Options are: ■ 0 (Standard nonlinear solve) ■ 1 (Natural parameter homotopy. See LOCA options list) ■ 2 (Specialized dual parameter homotopy for MOSFET circuits)	0 (Standard nonlinear solve)

Table 2.3: Options for Nonlinear Solver Package.

Transient Nonlinear Solver Package Options

The nonlinear solver selections listed in Table 2.4 provide methods for controlling the nonlinear solver for Transient analysis. These override the defaults and any that were set simply in the Nonlinear Solver Package (e.g., set MAXSTEP=5 for transient).

Nonlinear Solver (PKG = NONLIN-TRAN) Tag	Description	Default
NLSTRATEGY	Nonlinear solution strategy. Supported Strategies: ■ 0 (Newton) ■ 1 (Gradient) ■ 2 (Trust Region)	0 (Newton)
SEARCHMETHOD	Line-search method used by the nonlinear solver. Supported line-search methods: ■ 0 (Full Newton - no line search) ■ 1 (Interval Halving) ■ 2 (Quadratic Interpolation) ■ 3 (Cubic Interpolation) ■ 4 (More'-Thuente)	2 (Quadratic line search)

Nonlinear Solver (PKG = NONLIN-TRAN) Tag	Description	Default
ABSTOL	Absolute residual vector tolerance	1.0E-06
RELTOL	Relative residual vector tolerance	1.0E-02
DELTAXTOL	Weighted nonlinear-solution update norm convergence tolerance	0.33
RHSTOL	Residual convergence tolerance (unweighted 2-norm)	1.0E-02
MAXSTEP	Maximum number of Newton steps	20
MAXSEARCHSTEP	Maximum number of line-search steps	2
NORMLVL	Norm level used by the nonlinear solver algorithms *(NOTE: not used for convergence tests)*	2
LINOPT	Linear optimization flag	0 (FALSE)
CONSTRAINTBT	Constraint backtracking flag	0 (FALSE)
CONSTRAINTMAX	Global maximum setting for constraint backtracking	DBL_MAX (Machine Dependent Constant)
CONSTRAINTMIN	Global minimum setting for constraint backtracking	-DBL_MAX (Machine Dependent Constant)
CONSTRAINTCHANGE	Global percentage-change setting for constraint backtracking	sqrt(DBL_MAX) (Machine Dependent Constant)
IN_FORCING	Inexact Newton-Krylov forcing flag	1 (TRUE)
AZ_TOL	Sets the minimum allowed linear solver tolerance. Valid only if IN_FORCING=1.	1.0E-12
DIRECTLINSOLV	Direct (vs. iterative) linear solution flag *(Serial Only!)*	0 (FALSE)
DLSDEBUG	Debug output for direct linear solver	0 (FALSE)
DEBUGLEVEL	The higher this number, the more info is output	1

Nonlinear Solver (PKG = NONLIN-TRAN) Tag	Description	Default
DEBUGMINTIMESTEP	First time-step debug information is output	0
DEBUGMAXTIMESTEP	Last time-step of debug output	99999999
DEBUGMINTIME	Same as DEBUGMINTIMESTEP except controlled by time (sec.) instead of step number	0.0
DEBUGMAXTIME	Same as DEBUGMAXTIMESTEP except controlled by time (sec.) instead of step number	1.0E+99
RECOVERYSTEPTYPE	If using a line search, this option determines the type of step to take if the line search fails. Supported strategies: ■ 0 (Take the last computed step size in the line search algorithm) ■ 1 (Take a constant step size set by RECOVERYSTEP)	0
RECOVERYSTEP	Value of the recovery step if a constant step length is selected	1.0

Nonlinear Solver (PKG = NONLIN-TRAN) Tag	Description	Default
CONTINUATION	Enables the use of Homotopy/Continuation algorithms for the nonlinear solve. Options are: ■ 0 (Standard nonlinear solve) ■ 1 (Natural parameter homotopy. See LOCA options list) ■ 2 (Specialized dual parameter homotopy for MOSFET circuits)	0 (Standard nonlinear solve)

Table 2.4: Options for Nonlinear Solver Package under transient operation.

Continuation and Bifurcation Tracking Package Options

The continuation selections listed in Table 2.5 provide methods for controlling continuation and bifurcation analysis. These override the defaults and any that were set simply in the Continaution Package. This option block is only used if the nonlinear solver or transient nonlinear solver enable continaution through the CONTINUATION flag.

Continuation and Bifurcation (PKG = LOCA) Tag	Description	Default
STEPPER	Stepping algorithm to use: ■ 0 (Natural or Zero order continuation) ■ 1 (Arc-length continuation)	0 (Natural)
PREDICTOR	Predictor algorithm to use: ■ 0 (Tangent) ■ 1 (Secant) ■ 2 (Random) ■ 3 (Constant)	0 (Tangent)

Continuation and Bifurcation (PKG = LOCA) Tag	Description	Default
STEPCONTROL	Algorithm used to adjust the step size between continuation steps: ■ 0 (Constant) ■ 1 (Adaptive)	0 (Constant)
CONPARAM	Parameter in which to step during a continuation run	VA:V0
INITIALVALUE	Starting value of the continuation parameter	0.0
MINVALUE	Minimum value of the continuation parameter	-1.0E20
MAXVALUE	Maximum value of the continuation parameter	1.0E20
BIFPARAM	Parameter to compute during bifurcation tracking runs	VA:V0
MAXSTEPS	Maximum number of continuation steps (includes failed steps)	20
MAXNLITERS	Maximum number of nonlinear iterations allowed (set this parameter equal to the MAXSTEP parameter in the NONLIN option block	20
INITIALSTEPSIZE	Starting value of the step size	1.0
MINSTEPSIZE	Minimum value of the step size	1.0E20
MAXSTEPSIZE	Maximum value of the step size	1.0E-4
AGGRESSIVENESS	Value between 0.0 and 1.0 that determines how aggressive the step size control algorithm should be when increasing the step size. 0.0 is a constant step size while 1.0 is the most aggressive.	0.0

Table 2.5: Options for Continuation and Bifurcation Tracking Package.

Linear Solver Options

Xyce uses both a sparse-direct solver as well as Krylov iterative methods for the solution of the linear equations generated by Newton's method. For the advanced users, there are a variety of options that can be set to help improve these solvers. Many of these options (for the Krylov solvers) are simply passed through to the underlying Trilinos/Aztec solution settings and thus have an "AZ_" prefix on the flag; the "AZ_" options are all case-sensitive. The list in Table 2.6 only provides a partial list of the more commonly used Trilinos/Aztec

options. For a full list of the available options, please see the Aztec User's Guide [4] available for download at `http://www.cs.sandia.gov/CRF/Aztec1`. However, for most users, the default options should prove adequate.

Linear Solver (PKG = LINSOL) Tag	Description	Default
AZ_max_iter	Maximum number of iterative solver iterations	500
AZ_precond	Iterative solver preconditioner flag	AZ_dom_decomp (14)
AZ_solver	Iterative solver type	AZ_gmres (1)
AZ_conv	Convergence type	AZ_r0 (0)
AZ_pre_calc	Type of precalculation	AZ_recalc (1)
AZ_keep_info	Retain calculation info	AZ_true (1)
AZ_orthog	Type of orthogonalization	AZ_modified (1)
AZ_subdomain_solve	Subdomain solution for domain decomposition preconditioners	AZ_ilut (9)
AZ_ilut_fill	Approximate allowed fill-in factor for the ILUT preconditioner	3.0
AZ_drop	Specifies drop tolerance used in conjunction with LU or ILUT preconditioners	1.0E-03
AZ_reorder	Reordering type	AZ_none (0)
AZ_scaling	Type of scaling	AZ_none (0)
AZ_kspace	Maximum size of gmres kspace	500
AZ_tol	Convergence tolerance	1.0E-12
AZ_output	Output level	50
AZ_overlap	Schwarz overlap level for ilu preconditioners	0
AZ_rthresh	Diagonal shifting relative threshold for ilu preconditioners	1.0001
AZ_athresh	Diagonal shifting absolute threshold for ilu preconditioners	1.0E-04
TR_loadbalance	Perform load-balance on the linear system	0 (FALSE)

Linear Solver (PKG = LINSOL) Tag	Description	Default
TR_filter	Providing a non-zero value for THRESHOLD triggers filtering of the Jacobian entries by ROW_SUM * THRESHOLD	0.0
ksparse	Triggers use of KSparse versus SuperLU for direct solves	1 (TRUE)
TR_singleton_filter	Triggers use of singleton filter for linear system	0 (FALSE)
TR_reindex	Reindexes linear system parallel global indices in lexigraphical order, recommended with singleton filter	0 (FALSE)
TR_solvermap	Triggers remapping of column indices for parallel runs, recommended with singleton filter	0 (FALSE)
adaptive_solve	Triggers use of AztecOO adaptive solve algorithm for preconditioning of iterative linear solves	0 (FALSE)
use_ifpack_precond	Triggers use of IfPack for preconditioning of iterative linear solves	0 (FALSE)

Table 2.6: Options for Linear Solver Package.

Parallel Options

These options are used to control partitioning of the problem in distributed memory parallel environments. There are currently only 2 parameters available.

Parallel (PKG = PARALLEL) Tag	Description	Default
Partitioner	Choice of partitioners	0 (Chaco)
DistribIndSrcNodes	Cutoff threshold for Independent Source connectivity before distibution of source	1.0

Table 2.7: Options for Parallel Support.

Output Options

The main purpose of the .OPTIONS OUTPUT command is to allow control of the output frequency of data to files specified by .PRINT TRAN commands. The format is:

```
.OPTIONS OUTPUT INITIAL_INTERVAL=<interval> [<t0> <i0> [<t1> <i1>...]]
```

where INITIAL_INTERVAL=<interval> specifies the starting interval time for output and <tx> <ix> specifies later simulation times <tx> where the output interval will change to <ix>.

Note: Xyce will output data at the next time that is greater-than or equal to the current interval time. This means that output will not exactly correspond to the time intervals due to the adaptive time stepping algorithms.

Checkpointing and Restarting Options

The .OPTIONS RESTART command is used to control all checkpoint output and restarting.

- Checkpointing command format:

```
.OPTIONS RESTART [PACK=<0|1>] JOB=<job prefix>
+ [INITIAL_INTERVAL=<initial interval time>
+  [<t0> <i0> [<t1> <i1>...]]]
```

PACK=<0|1> indicates whether the restart date will be byte packed or not. Parallel restarts must always be packed while currently Windows/MingW runs are always not packed. Otherwise, by default data will be packed unless explicitly specified. JOB=<job prefix> identifies the prefix for restart files. The actual restart files will be the job name with the current simulation time appended (e.g. name1e-05 for JOB=name and simulation time 1e-05 seconds). Furthermore, INITIAL_INTERVAL=<initial interval time> identifies the initial interval time used for restart output. The <tx> <ix> intervals identify times <tx> at which the output interval (ix) should change. This functionality is identical that described for the .OPTIONS OUTPUT command.

To generate checkpoints at every time step (default):

Example: .OPTIONS RESTART JOB=checkpt

To generate checkpoints every 0.1 μs:

Example: .OPTIONS RESTART JOB=checkpt INITIAL_INTERVAL=0.1us

To generate unpacked checkpoints every 0.1 μs:

Example: `.OPTIONS RESTART PACK=0 JOB=checkpt INITIAL_INTERVAL=0.1us`

To specify an initial interval of 0.1 μs, at 1 μs change to interval of 0.5 μs, and at 10 μs change to interval of 0.1 μs:

Example:

```
.OPTIONS RESTART JOB=checkpt INITIAL_INTERVAL=0.1us 1.0us
+ 0.5us 10us 0.1us
```

To restart from an existing restart file, specify the file by either `FILE=<restart file name>` to explicitly use a restart file or by `JOB=<job name> START_TIME=<specified name>` to specify a file prefix and a specified time. The time must exactly match an output file time for the simulator to correctly identify the correct file. To continue generating restart output files, `INITIAL_INTERVAL=<interval>` and following intervals can be appended to the command in the same format as described above. New restart files will be packed according to the previous restart file read in. Here are several examples:

- Restarting command format:

```
.OPTIONS RESTART <FILE=<restart file name> |
+ JOB=<job name> START_TIME=<time>)>
+ [ INITIAL_INTERVAL=<interval> [<t0> <i0> [<t1> <i1> ...]]]
```

Example restarting from checkpoint file at 0.133 μs:

Example: `.OPTIONS RESTART JOB=checkpt START_TIME=0.133us`

To restart from checkpoint file at 0.133 μs:

Example: `.OPTIONS RESTART FILE=checkpt0.000000133`

Restarting from 0.133 μs and continue checkpointing at 0.1 μs intervals:

Example:

```
 .OPTIONS RESTART FILE=checkpt0.000000133 JOB=checkpt_again
+ INITIAL_INTERVAL=0.1us
```

2.2 Analog Devices

Xyce supports many analog devices, including sources, subcircuits and behavioral models. This section serves as a reference for the analog devices supported by **Xyce**. Each device is described separately and includes the following information, if applicable:

- a description and an example of the correct netlist syntax.
- the matching model types and their description.
- the matching list of model parameters and associated descriptions.
- the corresponding and characteristic equations for the model (as required).
- references to publications on which the model is based.

You can also create models and macromodels using the `.MODEL` (model definition) and `.SUBCKT` (subcircuit) statements, respectively.

Please note that the characteristic equations are provided to give a general representation of the device behavior. The actual **Xyce** implementation of the device may be slightly different in order to improve, for example, the robustness of the device.

Table 2.8 gives a summary of the analog device types and the form of their netlist formats. Each of these in described below in detail.

Device Type	Designator Letter	Typical Netlist Format
Capacitor	C	C<name> <+ node> <- node> [model name] <value> + [IC=<initial value>]
Inductor	L	L<name> <+ node> <- node> [model name] <value> + [IC=<initial value>]
Resistor	R	R<name> <+ node> <- node> [model name] <value> + [L=<length>] [W=<width>]
Diode	D	D<name> <anode node> <cathode node> + <model name> [area value]
Mutual Inductor	K	K<name> <inductor 1> [<ind. n>*] + <linear coupling or model>
Independent Voltage Source	V	V<name> <+ node> <- node> [[DC] <value>] + [transient specification]
Independent Current Source	I	I<name> <+ node> <- node> [[DC] <value>] + [transient specification]
Voltage Controlled Voltage Source	E	E<name> <+ node> <- node> <+ controlling node> + <- controlling node> <gain>

Device Type	Designator Letter	Typical Netlist Format
Voltage Controlled Current Source	G	G<name> <+ node> <- node> <+ controlling node> + <- controlling node> <transconductance>
Nonlinear Dependent Source (B Source)	B	B<name> <+ node> <- node> + <I or V>={<expression>}
Bipolar Junction Transistor (BJT)	Q	Q<name> <collector node> <base node> <emitter node> [substrate node] <model name> [area value]
MOSFET	M	M<name> <drain node> <gate node> <source node> + <bulk/substrate node> <model name> + [common model parameter]*
Transmission Line	T	T<name> <A port + node> <A port - node> + <B port + node> <B port - node> + <ideal specification>
Voltage Controlled Switch	S	S<name> <+ switch node> <- switch node> + <+ controlling node> <- controlling node> + <model name>
Subcircuit	X	X<name> [node]* <subcircuit name> + [PARAMS: [<name>=<value>]*]
PDE Devices	Z	Z<name> <node1> <node2> [node3] + [node4] <model name>

Table 2.8: Analog Device Quick Reference.

Capacitor

General Form

```
C<name> <(+) node> <(-) node> [model name] <value>
[IC=<initial value>][L=<length> W=<width>]
```

Age-aware Form

```
C<name> <(+) node> <(-) node> <value> [D=<coeff>][AGE=<age>
(hours)]
```

Examples

```
CM12 2 4 5.288e-13
CLOAD 1 0 4.540pF IC=1.5V
CFEEDBACK 2 0 CMOD 1.0pF
CAGED 2 3 4.0uF D=0.0233 AGE=86200
```

Model Form

```
.MODEL <model name> C [model parameters]
```

Parameters and Options

(+) and (-) nodes

Polarity definition for a positive voltage across the capacitor. The first node is defined as positive. Therefore, the voltage across the component is the first node voltage minus the second node voltage.

[model name]

If [model name] is omitted, then <value> is the capacitance in farads. If [model name] is given then the value is determined from the model parameters; see the capacitor value formula below.

<value>

The capacitance value in farads.

<initial value>

The initial voltage across the capacitor during the bias point calculation.

<coeff>

Degradation coefficient for age-aware capacitor model.

<age>

Capacitor age in hours.

<length>

Geometry length for semiconductor capacitor.

<width>

Geometry width for semiconductor capacitor.

Comments	Positive current flows through the capacitor from the (+) node to the (-) node. In general, capacitors should have a positive capacitance value (<value> property). In all cases, the capacitance must not be zero. However, cases exist when a negative capacitance value may be used. This occurs most often in filter designs that analyze an RLC circuit equivalent to a real circuit. When transforming from the real to the RLC equivalent, the result may contain a negative capacitance value. In a transient run, negative capacitance values may cause the simulation to fail due to instabilities they cause in the time integration algorithms. The age-aware capacitor can only be used without a model.

Model Parameters

Table 2.9 gives the available model parameters for the capacitor.

Model parameters	Description	Units	Default
C	Capacitance Multiplier		1.0
D	Degradation Coefficient	-	0.0233
AGE	Capacitor model age	hours	0.0
L	Semiconductor model length	m	1.0
W	Semiconductor model width	m	1.0E-6
CJ	Junction bottom capacitance	F/m^2	0.0
CJSW	Junction sidewall capacitance	F/m	0.0
NARROW	Narrowing due to side etching	m	0.0

Table 2.9: Capacitor Model Parameters.

Capacitor Equations

Capacitance Value Formula

Age-aware Formula

If $\mathbf{AGE}$ is given, then the capacitance is:

$$\mathbf{C}[1 - \mathbf{D}\log(\mathbf{AGE})]$$

Semiconductor Formula

If `[model name]` and **L** and **W** are given, then the capacitance is:

$$\mathbf{CJ}(\mathbf{L}-\mathbf{NARROW})(\mathbf{W}-\mathbf{NARROW})+2\cdot\mathbf{CJSW}(\mathbf{L}-\mathbf{W}+2\cdot\mathbf{NARROW})$$

Inductor

General Form	`L<name> <(+) node> <(-) node> <value> + [IC=<initial value>]`
Examples	`L1 1 5 3.718e-08` `LLOAD 3 6 4.540mH IC=2mA`
Model Form	`.MODEL <model name> I [model parameters]`
Parameters and Options	`(+)` and `(-)` nodes Polarity definition for a positive voltage across the inductor. The first node is defined as positive. Therefore, the voltage across the component is the first node voltage minus the second node voltage. `<initial value>` The initial current through the inductor during the bias point calculation.
Comments	In general, inductors should have a positive inductance value (`VALUE` property). In all cases, the inductance must not zero. However, cases exist when a negative value may be used. This occurs most often in filter designs that analyze an RLC circuit equivalent to a real circuit. When transforming from the real to the RLC equivalent, the result may contain a negative inductance value.

Mutual Inductors

General Form

```
K<name> L<inductor name> [L<inductor name>*] <coupling value>
+ [model name]
```

Examples

```
KTUNED L3OUT  L4IN .8
KTRNSFRM LPRIMARY LSECNDRY 1
KXFRM L1 L2  L3  L4 .98 KPOT_3C8
```

Model Form

```
.MODEL <model name> CORE [model parameters]
```

Parameters and Options

`L<inductor name> [L<inductor name>*]`

Identifies the inductors to be coupled. The inductors are coupled and in the dot notation the dot is placed on the first node of each inductor. The polarity is determined by the order of the nodes in the L devices and not by the order of the inductors in the K statement.

`<coupling value>`

The coefficient of mutual coupling, which must be between -1.0 and 1.0.

This coefficient is defined by the equation

$$\texttt{<coupling value>} = \frac{M_{ij}}{L_i L_j}$$

where

L_i is the inductance of the ith named inductor in the K-line

M_{ij} is the mutual inductance between L_i and L_j

For transformers of normal geometry, use 1.0 as the value. Values less than 1.0 occur in air core transformers when the coils do not completely overlap.

`<model name>`

If `<model name>` is present, four things change:

- The mutual coupling inductor becomes a nonlinear, magnetic core device.
- The inductors become windings, so the number specifying inductance now specifies the number of turns.
- The list of coupled inductors could be just one inductor.
- A model statement is required to specify the model parameters.

Model Parameters

Table 2.10 gives the available model parameters for mutual inductors.

Model parameters	Description	Units	Default
AREA	mean magnetic cross-sectional area	cm^2	0.01
GAP	effective air gap length	cm	0.0
PATH	total mean magnetic path length	cm	1.0
MS	saturation magnetization	amp/m	1.0e6
A	thermal energy parameter	amp/m	1000.0
C	domain flexing constant		0.2
ALPHA	domain coupling parameter		5.0e-5
KIRR	irreversible magnetization	amp/m	500.0
DELV	modeling parameter	V	0.001
BETAH	modeling parameter		0.00001
BETAM	modeling parameter		3.124e-5
K	alternative specification of KIRR for compatibility with PSpice	$\frac{A}{m}$	500.0
PACK	PSpice compatibility parameter, ignored by **Xyce**		
LEVEL	PSpice compatibility parameter, ignored by **Xyce**		

Table 2.10: Inductor Model Parameters.

Resistor

General Form	`R<name> <(+) node> <(-) node> [model name] <value> + [L=<length>] [W=<width`

Examples

```
R1 1 2 2K
RLOAD 3 6 RTCMOD 4.540
.MODEL RTCMOD R (TC1=.01 TC2=-.001)
RSEMICOND 2 0 RMOD L=1000u W=1u
.MODEL RMOD R (RSH=1)
```

Model Form	`.MODEL <model name> R [model parameters]`

Parameters and Options

`(+)` and `(-)` nodes

Polarity definition for a positive voltage across the resistor. The first node is defined as positive. Therefore, the voltage across the component is the first node voltage minus the second node voltage. Positive current flows from the positive node (first node) to the negative node (second node).

`[model name]`

If `[model name]` is omitted, then `<value>` is the resistance in Ohms. If `[model name]` is given then the value is determined from the model parameters; see the resistance value formula below.

`<length>` and `<width>`

Length and width for semiconductor resistance model

Comments

Resistors must have a positive resistance value (`<value>` property) - the resistance must not be zero.

Model Parameters

Table 2.11 gives the available model parameters for the resistor.

Model parameters	Description	Units	Default
R	Resistance Multiplier		1.0
TC1	Linear Temperature Coefficient	$°C^{-1}$	0.0
TC2	Quadratic Temperature Coefficient	$°C^{-2}$	0.0
TCE	Exponential Temperature Coefficient	%/°C	0.0
TNOM	Parameter Measurement Temperature	°C	27.0
RSH	Sheet Resistance	ohm	0.0
NARROW	Narrowing due to side etching	m	0.0

Table 2.11: Resistor Model Parameters.

Resistor Equations

Resistance Value Formula

If [model name] is included *and* **TCE** is given, then the resistance is:

$$< \texttt{value} > \cdot \mathbf{R} \cdot 1.01^{\mathbf{TCE}(T-T_0)}$$

where <value> is usually positive although it may be negative but never zero. T_0 is the nominal temperature (set using **TNOM** option).

If [model name] is included *and* **TCE** is not given, then the resistance is:

$$< \texttt{value} > \cdot \mathbf{R}(1 + \mathbf{TC1} \cdot (T - T_0) + \mathbf{TC2} \cdot (T - T_0)^2)$$

where <value> is typically positive although it may be negative but never zero.

If **L** and **W** are given, the resistance is:

$$\mathbf{RSH}\frac{[\mathbf{L} - \mathbf{NARROW}]}{[\mathbf{W} - \mathbf{NARROW}]}$$

Diode

General Form	`D<name> <(+) node> <(-) node> <model name> [area value]`
Examples	`DCLAMP 1 0 DMOD` `D2 15 17 SWITCH 1.5`
Model Form	`.MODEL <model name> D [model parameters]`

Parameters and Options	`<(+) node>`
	The anode.
	`<(-) node>`
	The cathode.
	`[area value]`
	Scales IS, ISR, IKF, RS, CJO, and IBV, and has a default value of 1. IBV and BV are both specified as positive values.

Comments	The diode is modeled as an ohmic resistance (`RS/area`) in series with an intrinsic diode. Positive current is current flowing from the anode through the diode to the cathode.

Diode Operating Temperature

Model parameters can be assigned unique measurement temperatures using the TNOM model parameter.

Diode level selection

Two distinct implementations of the diode are available. These are selected by using the `LEVEL` model parameter. The default implementation is based on SPICE 3F5, and may be explicitly specified using `LEVEL=1` in the model parameters, but is also selected if no `LEVEL` parameter is specified. The second implementation is based on the AIM-SPICE diode model[5] as modified to include radiation effects[6]. This second implementation is selected using the `LEVEL=3` model parameter. There is no `LEVEL=2` model in this version of **Xyce**.

Model Parameters

Table 2.12 gives the available model parameters for the `LEVEL=1` diode.

Model parameters	Description	Units	Default
BV	Reverse Breakdown Knee Voltage	volt	1E99

Model parameters	Description	Units	Default
CJO	Zero-bias p-n Depletion Capacitance	farad	0.0
EG	Activation Energy	eV	1.11
FC	Forward-bias Depletion Capacitance Coefficient	-	0.5
IBV	Reverse Breakdown Knee Current	amp	1E-3
IS	Saturation Current	amp	1E-14
M	p-n Grading Parameter	-	0.5
N	Emission Coefficient	-	1.0
RS	Parasitic Resistance	ohm	0.0
TT	Transit Time	sec	0.0
TNOM	Parameter Measurement Temperature	°C	27.0
VJ	p-n Potential	volt	1.0
XTI	**IS** Temperature Exponent		3.0

Table 2.12: Diode Model Parameters.

Table 2.13 gives the available model parameters for the LEVEL=3 diode. In all cases, voltage parameters related to the "radiation voltage" are relative to the radiation generation rate (e.g. if PULSEV2 is 4.3 and GRORDER is 26, the radiation dose at the top of the pulse is $4.3 \times 10^{26} m^{-3} s^{-1}$, or 1×10^{7} rads^{-s}).

Model parameters	Description	Units	Default
KF	Flicker Noise Coefficient	-	0.0
AF	Flicker Noise Exponent	-	1.0
BV	Reverse Breakdown Knee Voltage	volt	1E99
CJO	Zero-bias p-n Depletion Capacitance	farad	0.0
EG	Activation Energy	eV	1.11
FC	Forward-bias Depletion Capacitance Coefficient	-	0.5
IBV	Reverse Breakdown Knee Current	amp	1E-3
IS	Saturation Current	amp	1E-14
M	p-n Grading Parameter	-	0.5
N	Emission Coefficient	-	1.0
RS	Parasitic Resistance	ohm	0.0
TT	Transit Time	sec	0.0
TNOM	Parameter Measurement Temperature	°C	27.0
VJ	p-n Potential	volt	1.0

Model parameters	Description	Units	Default
XTI	**IS** Temperature Exponent	-	3.0
IKF	High-injection knee current	amp	0.0
ISR	Recombination current	amp	0.0
NR	Emission coefficient for ISR	-	2.0
ISNPP	Is this an n-p-p+ diode?	boolean	0 (false)
ISPNN	Is this an p-n-n+ diode?	boolean	0 (false)
FUNCTIONTYPE	Selects the shape of the radiation pulse	-	0 (none)
GRORDER	Order of magnitude of radiation generation rate	-	25
PERMITTIVITY	Dielectric permittivity of the semiconductor	F/m	1.0443E-10
NA	Doping density in p- region	m^{-3}	5E20
NAHI	Doping density in p+ region	m^{-3}	1E24
ND	Doping density in n- region	m^{-3}	3E25
NDHI	Doping density in n+ region	m^{-3}	3E25
NI	Intrinsic concentration	m^{-3}	1.45E16
WN	Width of the n- region	m	2E-6
WNHI	Width of the n+ region	m	1E-6
WP	Width of the n- region	m	5E-5
WPHI	Width of the n+ region	m	1E-5
PULSEV1	Initial value of radiation pulse (FUNCTIONTYPE=1)	volt	0.0
PULSEV2	Maximum value of radiation pulse (FUNCTIONTYPE=1)	volt	4.3
PULSETD	Delay of radiation pulse (FUNCTIONTYPE=1)	sec	0.0
PULSETR	Rise time of radiation pulse (FUNCTIONTYPE=1)	sec	1E-5
PULSETF	Fall time of radiation pulse (FUNCTIONTYPE=1)	sec	1E-5
PULSEPW	Width of radiation pulse (FUNCTIONTYPE=1)	sec	1E-5
PULSEPER	Period of radiation pulse (FUNCTIONTYPE=1)	sec	1E10
SINV0	Initial value of sinusoidal radiation voltage (FUNCTIONTYPE=2)	volt	0.0

Model parameters	Description	Units	Default
SINVA	Amplitude of sinusoidal radiation voltage (FUNCTIONTYPE=2)	volt	0.0
SINFREQ	Frequency of sinusoidal radiation voltage (FUNCTIONTYPE=2)	Hz	1.0
SINTD	Delay of sinusoidal radiation voltage (FUNCTIONTYPE=2)	sec	0.0
SINTHETA	Inverse of time constant for exponential envelope of sinusoidal radiation voltage (FUNCTIONTYPE=2)	sec^{-1}	0.0
EXPV1	Initial value of exponential cusp radiation voltage (FUNCTIONTYPE=3)	volt	0.0
EXPV2	Height of exponential cusp radiation voltage (FUNCTIONTYPE=3)	volt	0.0
EXPTD1	Rise delay time of exponential cusp radiation voltage (FUNCTIONTYPE=3)	sec	0.0
EXPTAU1	Rise time constant of exponential cusp radiation voltage (FUNCTIONTYPE=3)	sec	1e-2
EXPTD2	Fall delay time of exponential cusp radiation voltage (FUNCTIONTYPE=3)	sec	1E10
EXPTAU2	Fall time constant of exponential cusp radiation voltage (FUNCTIONTYPE=3)	sec	1E10
SFFMVO	Offset of single frequency FM curve (FUNCTIONTYPE=4)	volt	0.0
SFFMVA	Amplitude of single frequency FM curve (FUNCTIONTYPE=4)	volt	0.0
SFFMFC	Carrier frequency of single frequency FM curve (FUNCTIONTYPE=4)	Hz	1.0
SFFMMDI	Modulation index of single frequency FM curve (FUNCTIONTYPE=4)	-	0.0
SFFMFS	Signal frequency of single frequency FM curve (FUNCTIONTYPE=4)	Hz	0.0
PWLT1	Time of point 1 of piecewise linear curve (FUNCTIONTYPE=5)	sec	1
PWLT2	Time of point 2 of piecewise linear curve (FUNCTIONTYPE=5)	sec	2
PWLT3	Time of point 3 of piecewise linear curve (FUNCTIONTYPE=5)	sec	3
PWLT4	Time of point 4 of piecewise linear curve (FUNCTIONTYPE=5)	sec	4
PWLT5	Time of point 5 of piecewise linear curve (FUNCTIONTYPE=5)	sec	5

Model parameters	Description	Units	Default
PWLV1	Height of point 1 of piecewise linear curve (FUNCTIONTYPE=5)	volt	0
PWLV2	Height of point 2 of piecewise linear curve (FUNCTIONTYPE=5)	volt	0
PWLV3	Height of point 3 of piecewise linear curve (FUNCTIONTYPE=5)	volt	0
PWLV4	Height of point 4 of piecewise linear curve (FUNCTIONTYPE=5)	volt	0
PWLV5	Height of point 5 of piecewise linear curve (FUNCTIONTYPE=5)	volt	0
LINET1	Time of point 1 for linear radiation voltage (FUNCTIONTYPE=6)	sec	0
LINEV1	Height of point 1 for linear radiation voltage (FUNCTIONTYPE=6)	volt	0
LINEK	Slope linear radiation voltage (FUNCTIONTYPE=6)	-	0
RAUGN	Auger recombination rate for electrons in p-region	cm^6/s	1.1E-42
RAUGP	Auger recombination rate for holes in n-region	cm^6/s	3E-43
NDN	Diffusion constant of electrons in n-region at low injection	m^2/s	1E-3
NDP	Diffusion constant of holes in n-region at low injection	m^2/s	7E-4
TAUP0	Hole lifetime in n-region at low injection	sec	2e-8
TAUINFP	Hole lifetime in n-region at saturated SRH condition	sec	4E-8
PDN	Diffusion constant of electrons in p-region at low injection	m^2/s	4E-3
PDP	Diffusion constant of holes in p-region at low injection	m^2/s	5E-4
TAUN0	Electron lifetime in p-region at low injection	sec	1e-6
TAUINFN	Electron lifetime in p-region at saturated SRH condition	sec	2e-6
PDNHI	Diffusion constant of electrons in p+-region at low injection	m^2/s	5E-3
PDPHI	Diffusion constant of holes in p+-region at low injection	m^2/s	1E-4
TAUN0HI	Electron lifetime in p+-region at low injection	sec	1e-7

Model parameters	Description	Units	Default
TAUINFNHI	Electron lifetime in p+-region at saturated SRH condition	sec	2E-7
NDNHI	Diffusion constant of electrons in n+-region at low injection	m^2/s	1e-3
NDPHI	Diffusion constant of holes in n+-region at low injection	m^2/s	2e-4
TAUPOHI	Hole lifetime in n+-region at low injection	sec	2e-8
TAUINFPHI	Hole lifetime in n+-region at saturated SRH condition	sec	4e-8

Table 2.13: Diode Level 3 Model Parameters.

Diode Equations

The equations in this section use the following variables:

$$\begin{aligned} V_{di} &= \text{voltage across the intrinsic diode only} \\ V_{th} &= k \cdot T/q \text{ (thermal voltage)} \\ k &= \text{Boltzmann's constant} \\ q &= \text{electron charge} \\ T &= \text{analysis temperature (Kelvin)} \\ T_0 &= \text{nominal temperature (set using } \mathbf{TNOM} \text{ option)} \\ \omega &= \text{Frequency (Hz)} \end{aligned}$$

Other variables are listed above in the diode model parameters.

Level=1

The level 1 diode is based on the Spice3f5 level 1 model.

DC Current (Level=1)

The intrinsic diode current consists of forward and reverse bias regions where

$$I_D = \begin{cases} \mathbf{IS} \cdot \left[\exp\left(\frac{V_{di}}{\mathbf{N}V_{th}}\right) - 1\right], & V_{di} > -3.0 \cdot \mathbf{N}V_{th} \\ -\mathbf{IS} \cdot \left[1.0 + \left(\frac{3.0 \cdot \mathbf{N}V_{th}}{V_{di} \cdot q}\right)^3\right], & V_{di} < -3.0 \cdot \mathbf{N}V_{th} \end{cases}$$

When $\mathbf{BV}$ is explicitly given in the model statement, an exponential current model is substituted when $V_{di} < -\mathbf{BV}$.

This equation for I_D is

$$I_D = -\mathbf{IS} \cdot \exp\left(-\frac{BV + V_{di}}{\mathbf{N}V_{th}}\right)$$

Note here that if $\mathbf{IBV}$ is set by the user, then $\mathbf{BV}$ is changed internally to ensure consistency. This is the equation actually used by **Xyce** but the value of $\mathbf{BV}$ is not required to be specified by the user.

Capacitance (Level=1)

The p-n diode capacitance consists of a depletion layer capacitance C_d and a diffusion capacitance C_{dif}. The first is given by

$$C_d = \begin{cases} \mathbf{CJ} \cdot \mathbf{AREA} \left(1 - \frac{V_{di}}{\mathbf{VJ}}\right)^{-\mathbf{M}}, & V_{di} \leq \mathbf{FC} \cdot \mathbf{VJ} \\ \frac{\mathbf{CJ} \cdot \mathbf{AREA}}{\mathbf{F2}} \left(\mathbf{F3} + \mathbf{M}\frac{V_{di}}{\mathbf{VJ}}\right), & V_{di} > \mathbf{FC} \cdot \mathbf{VJ} \end{cases}$$

The diffusion capacitance (sometimes referred to as the transit time capacitance) is

$$C_{dif} = \mathbf{TT} G_d = \mathbf{TT} \frac{dI_D}{dV_{di}}$$

where G_d is the junction conductance.

Temperature Effects (Level=1)

The diode model contains explicit temperature dependencies in the ideal diode current, the generation/recombination current and the breakdown current. Further temperature dependencies are present in the diode model via the saturation current I_S, the depletion layer junction capacitance CJ, the junction potential V_J.

$$\begin{aligned}
V_t(T) &= \frac{kT}{q} \\
V_{tnom}(T) &= \frac{k\mathbf{TNOM}}{q} \\
E_g(T) &= E_{g0} - \frac{\alpha T^2}{\beta + T} \\
E_{gNOM}(T) &= E_{g0} - \frac{\alpha \mathbf{TNOM}^2}{\mathbf{TNOM} + \beta} \\
arg1(T) &= -\frac{E_g(T)}{2kT} + \frac{E_{g300}}{2kT_0} \\
arg2(T) &= -\frac{E_{gNOM}(T)}{2k\mathbf{TNOM}} + \frac{E_{g300}}{2kT_0} \\
pbfact1(T) &= -2.0 \cdot V_t(T) \left(1.5 \cdot \ln\left(\frac{T}{T_0}\right) + q \cdot arg1(T)\right) \\
pbfact2(T) &= -2.0 \cdot V_{tnom}(T) \left(1.5 \cdot \ln\left(\frac{\mathbf{TNOM}}{T_0}\right) + q \cdot arg2(T)\right) \\
pbo(T) &= (\mathbf{VJ} - pbfact2(T)) \frac{T_0}{\mathbf{TNOM}} \\
V_J(T) &= pbfact1(T) + \frac{T}{T_0} pbo(T)
\end{aligned}$$

$$
\begin{aligned}
gma_{old}(T) &= \frac{\mathbf{VJ} - pbo(T)}{pbo(T)} \\
gma_{new}(T) &= \frac{V_J(T) - pbo(T)}{pbo(T)} \\
CJ(T) &= \mathbf{CJ0}\frac{1.0 + \mathbf{M}\left(4.0 \times 10^{-4}\left(T - T_0\right) - gma_{new}(T)\right)}{1.0 + \mathbf{M}\left(4.0 \times 10^{-4}\left(\mathbf{TNOM} - T_0\right) - gma_{old}(T)\right)} \\
I_S(T) &= \mathbf{IS} \cdot \exp\left(\left(\frac{T}{\mathbf{TNOM}} - 1.0\right) \cdot \frac{\mathbf{EG}}{\mathbf{N}V_t(T)} + \frac{\mathbf{XTI}}{\mathbf{N}} \cdot \ln\left(\frac{T}{\mathbf{TNOM}}\right)\right)
\end{aligned}
$$

where, for silicon, $\alpha = 7.02 \times 10^{-4}\ eV/K$, $\beta = 1108\ ^\circ K$ and $E_{g0} = 1.16\ eV$.

Level=3

The level 3 diode implementation is based on [5] with radiation effects added[6].

DC Current (Level=3)

Here the I-V nature is modeled using a diffusion current I_D, a generation/recombination current I_{GR} and a breakdown current I_B

$$I = K_{HI} \cdot I_D + I_{GR} - I_B$$

where

$$I_D = \mathbf{IS} \cdot \left[\exp\left(\frac{V_{di}}{\mathbf{N}V_{th}}\right) - 1\right]$$

and the injection current factor is given by

$$K_{HI} = \begin{cases} \sqrt{\mathbf{IKF}/(\mathbf{IKF} + I_D)}, & \text{for } \mathbf{IKF} > 0 \\ 1, & \text{otherwise} \end{cases}$$

The generation/recombination current is given by

$$I_{GR} = \mathbf{ISR}\left[\left(1 - \frac{V_{di}}{\mathbf{VJ}}\right)^2 + 0.001\right]^{\mathbf{M}/2}\left[\exp\left(\frac{V_{di}}{\mathbf{NR}V_{th}}\right) - 1\right]$$

where the grading parameter $\mathbf{M}$ is $1/2$ for an abrupt junction and $1/3$ for a linearly graded junction. Lastly, the breakdown current is calculated using

$$I_B = \mathbf{IBV}\exp\left(-\frac{V_{di} + \mathbf{BV}}{\mathbf{N}V_{th}}\right)$$

Capacitance (Level=3)

The p-n diode capacitance consists of a depletion layer capacitance C_d and a diffusion capacitance C_{dif}. The first is given by

$$C_d = \begin{cases} \mathbf{CJO}\left(1 - \frac{V_{di}}{\mathbf{VJ}}\right)^{-\mathbf{M}}, & V_{di} \le \mathbf{FC} \cdot \mathbf{VJ} \\ \mathbf{CJO}(1 - \mathbf{FC})^{-(1+\mathbf{M})}\left[1 - \mathbf{FC}(1 + \mathbf{M}) + \mathbf{M}\frac{V_{di}}{\mathbf{VJ}}\right], & V_{di} > \mathbf{FC} \cdot \mathbf{VJ} \end{cases}$$

The diffusion capacitance (sometimes referred to as the transit time capacitance) is

$$C_{dif} = \mathbf{TT}G_d = \mathbf{TT}\frac{dI}{dV_{di}}$$

where G_d is the junction conductance and I is the diode dc current given above.

Temperature Effects (Level=3)

The diode model contains explicit temperature dependencies in the ideal diode current, the generation/recombination current and the breakdown current. Further temperature dependencies are present in the diode model via the saturation current I_S and the built-in or contact voltage V_{bi}

$$\begin{aligned} I_S(T) &= \mathbf{IS} \cdot \left(\frac{T}{T_0}\right)^{\mathbf{XTI/N}} \exp\left[\frac{\mathbf{EG} \cdot (T - T_0)}{\mathbf{N} \cdot qV_{th}T_0}\right] \\ V_{bi}(T) &= \frac{T}{T_0}\mathbf{VJ} - 2V_{th}\ln\left(\frac{T}{T_0}\right)^{1.5} - \frac{1}{q}\left[\frac{T}{T_0}E_{g0} - E_g(T)\right] \\ E_g(T) &= E_{g0} - \frac{\alpha T^2}{\beta + T} \end{aligned}$$

where, for silicon, $\alpha = 7.02 \times 10^{-4}\ eV/K$, $\beta = 1108\ ^\circ K$ and $E_{g0} = 1.16\ eV$.

Radiation Effects

The equations used to incorporate radiation effects into the `LEVEL=3` diode are beyond the scope of this manual. Consult [6] for a detailed analysis.

Caveats

The area parameter that can be specified on the instance line scales many of the device model parameters, and is usually used to specify a number of equivalent parallel devices rather than an actual device area. In the original AimSPICE and ChileSPICE implementation this same area parameter was used to scale the photocurrent. Many of the extracted BJT models in use at Sandia were designed to be used with the instance area parameter omitted. Using the radiation aware model with an area of 1.0 results in absurdly high photocurrents, but specifying any other area causes model parameters to be scaled away from the values originally extracted for the real devices.

To address this problem, the **Xyce** implementation of the radiation-aware device has defined a new model parameter, DEVICEAREA, that can be specified in the model statement instead of the instance line. In the **Xyce** implementation, the instance line area is used only to scale the model parameters in a manner consistent with SPICE 3F5, and the DEVICEAREA parameter is used only to scale the photocurrent. **This is incompatible with ChileSPICE and Aim-SPICE. Xyce will allow real device models to be used in the radiation context by adding the radiation specific parameters and the DEVICEAREA parameter to the model statement, whereas the other simulators will**

require that several level 1 model parameters be converted to per-unit-area values before use with the level 3 model.

For a more thorough description of p-n junction physics, see [9]. For a thorough description of the U.C. Berkeley SPICE models see Reference [11].

Independent Voltage Source

General Form	`V<name> <(+) node> <(-) node> + [ [DC] <DC|TRAN value> ]` `+ [transient specification]`
Examples	`VSLOW 1 22 SIN(0.5V 1.0ma 1KHz 1ms)` `VPULSE 1 3 PULSE(-1V 1V 2ns 2ns 2ns 50ns 100ns)`
Description	Positive current flows from the positive node through the source to the negative node. DC/TRAN is the source value during a DC or Transient analysis, respectively. For a transient analysis, all independent sources can be given a time-dependent value and its value at is used for the DC analysis.
Parameters and Options	`[transient specification]` There are five predefined time-varying functions for sources: ■ `PULSE(<parameters>)` - a pulse waveform ■ `SIN(<parameters>)` - a sinusoidal waveform ■ `EXP(<parameters>)` - an exponential waveform ■ `PWL(<parameters>)` - a piecewise linear waveform ■ `SFFM(<parameters>)` - a frequency-modulated waveform

Transient Specifications

This section outlines the available transient specifications. Δt and T_F are the time step size and simulation end-time, respectively.

Pulse

```
PULSE(V1 V2 TD TR TF PW PER)
```

Parameter	Default	Unit
V1 (Initial Value)	N/A	volt
V2 (Pulse Value)	N/A	volt
TD (Delay Time)	0.0	s
TR (Rise Time)	Δt	s
TF (Fall Time)	Δt	s
PW (Pulse Width)	T_F	s
PER (Period)	T_F	s

Sine

```
SIN(V0 VA FREQ TD THETA)
```

Parameter	Default	Unit
V0 (Offset)	N/A	volt
VA (Amplitude)	N/A	volt
FREQ (Frequency)	0.0	s
TD (Delay)	Δt	s
THETA (Attenuation Factor)	Δt	s

The waveform is shaped according to the following equations:

$$V = \begin{cases} V_0, & 0 < t < T_D \\ V_0 + V_A \sin[2\pi \cdot \mathbf{FREQ} \cdot (t - T_D)] \exp[-(t - T_D) \cdot \mathbf{THETA}], & T_D < t < T_F \end{cases}$$

Exponent

```
EXP(V1 V2 TD1 TAU1 TD2 TAU2)
```

Parameter	Default	Unit
V1 (Initial Phase)	N/A	volt
VA (Amplitude)	N/A	volt
TD1 (Rise Delay Time)	0.0	s
TAU1 (Rise Time Constant)	Δt	s
TD2 (Delay Fall Time)	TD1 $+\Delta t$	s
TAU2 (Fall Time Constant)	Δt	s

The waveform is shaped according to the following equations:

$$V = \begin{cases} V_1, & 0 < t < \text{TD1} \\ V_1 + (V_2 - V_1)\{1 - \exp[-(t - \text{TD1})/\text{TAU1}]\}, & \text{TD1} < t < \text{TD2} \\ V_1 + (V_2 - V_1)\{1 - \exp[-(t - \text{TD1})/\text{TAU1}]\} & \\ \quad +(V_1 - V_2)\{1 - \exp[-(t - \text{TD2})/\text{TAU2}]\}, & \text{TD2} < t < T_2 \end{cases}$$

Piecewise Linear

```
PWL [TIME_SCALE_FACTOR=<value>]
```

```
+ [VALUE_SCALE_FACTOR=<value>] <corner points>

PWL [TIME_SCALE_FACTOR=<value>]
+ [VALUE_SCALE_FACTOR=<value>] FILE "<name>"
```

Parameter	Default	Unit
<tn> (Time at Corner)	s	none
<vn> (Voltage at Corner)	volt	none
<n> (Number of Repetitions)	positive integer, 0, or -1	none

When the FILE option is given, **Xyce** will read the corner points from the file specified in the <name> field.

Frequency Modulated

```
SFFM (<voff> <vampl> <fc> <mod> <fm>)
```

Parameter	Default	Unit
<voff> (Offset Voltage)	volt	none
<vampl> (Peak Voltage Amplitude)	volt	none
<fc> (Carrier Frequency)	hertz	1/**TSTOP**
<mod> (Modulation Index)		0
<fm> (Modulation Frequency)	hertz	1/**TSTOP**

The waveform is shaped according to the following equations:

$$V = \mathbf{voff} + \mathbf{vampl} \cdot \sin(2\pi \cdot \mathbf{fc} \cdot \mathbf{TIME} + \mathrm{mod} \cdot \sin(2\pi \cdot \mathbf{fm} \cdot \mathbf{TIME}))$$

where **TIME** is the current simulation time.

Independent Current Source

General Form	`I<name> <(+) node> <(-) node> + [ [DC] <DC/TRAN value> ]` `+ [transient specification]`
Examples	`ISLOW 1 22 SIN(0.5ma 1.0ma 1KHz 1ms)` `IPULSE 1 3 PULSE(-1mA 1mA 2ns 2ns 2ns 50ns 100ns)`
Description	Positive current flows from the positive node through the source to the negative node. DC/TRAN is the source value during a DC or transient analysis, respectively. For a transient analysis, all independent sources can be given a time-dependent value and its value at is used for the DC analysis.
Parameters and Option	`[transient specification]` There are five predefined time-varying functions for sources: ■ PULSE(<parameters>) - a pulse waveform ■ SIN(<parameters>) - a sinusoidal waveform ■ EXP(<parameters>) - an exponential waveform ■ PWL(<parameters>) - a piecewise linear waveform ■ SFFM(<parameters>) - a frequency-modulated waveform

Transient Specifications

This section outlines the available transient specifications. Δt and T_F are the time step size and simulation end-time, respectively.

Pulse

```
PULSE(I1 I2 TD TR TF PW PER)
```

Parameter	Default	Unit
I1 (Initial Value)	N/A	amp
I2 (Pulsed Value)	N/A	amp
TD (Delay Time)	0.0	s
TR (Rise Time)	Δt	s
TF (Fall Time)	Δt	s
PW (Pulse Width)	T_F	s
PER (Period)	T_F	s

Sine

```
SIN(IO IA FREQ TD THETA)
```

Parameter	Default	Unit
IO (Offset)	N/A	amp
IA (Amplitude)	N/A	amp
FREQ (Frequency)	0.0	s
TD (Delay)	Δt	s
THETA (Attenuation Factor)	Δt	s

The waveform is shaped according to the following equations:

$$I = \begin{cases} I_0, & 0 < t < T_D \\ I_0 + I_A \sin[2\pi \cdot \mathbf{FREQ} \cdot (t + T_D)] \ \exp[-(t + T_D) \cdot \mathbf{THETA}], & T_D < t < T_F \end{cases}$$

Exponent

```
EXP(I1 I2 TD1 TAU1 TD2 TAU2)
```

Parameter	Default	Unit
I1 (Initial Phase)	N/A	amp
IA (Amplitude)	N/A	amp
TD1 (Rise Delay Time)	0.0	s
TAU1 (Rise Time Constant)	Δt	s
TD2 (Delay Fall Time)	$\mathbf{TD1} + \Delta t$	s
TAU2 (Fall Time Constant)	Δt	s

The waveform is shaped according to the following equations:

$$I = \begin{cases} I_1, & 0 < t < \text{TD1} \\ I_1 + (I_2 - I_1)(1 - \exp[-(t - \text{TD1}/\text{TAU1}]), & \text{TD1} < t < \text{TD2} \\ I_1 + (I_2 - I_1)(1 - \exp[-(t - \text{TD1}/\text{TAU1}]) & \\ \quad +(I_1 - I_2)(1 - \exp[-(t - \text{TD2}/\text{TAU2}]), & \text{TD2} < t < T_2 \end{cases}$$

Piecewise Linear

```
PWL[TIME_SCALE_FACTOR=<value>] +
[VALUE_SCALE_FACTOR=<value>] (corner_points)

PWL [TIME_SCALE_FACTOR=<value>]
+ [VALUE_SCALE_FACTOR=<value>] FILE "<name>"
```

Parameter	Default	Unit
<tn> (Time at Corner)	s	none
<vn> (Voltage at Corner)	volt	none
<n> (Number of Repetitions)	positive integer, 0, or -1	none

When the FILE option is given, **Xyce** will read the corner points from the file specified in the <name> field.

Frequency Modulated

```
SFFM (<ioff> <iampl> <fc> <mod>  <fm>)
```

Parameter	Default	Unit
<ioff> (Offset Current)	amp	none
<iampl> (Peak Current Amplitude)	amp	none
<fc> (Carrier Frequency)	hertz	1/TSTOP
<mod> (Modulation Index)		0
<fm> (Modulation Frequency)	hertz	1/TSTOP

The waveform is shaped according to the following equations:

$$I = \mathbf{ioff} + \mathbf{iampl} \cdot \sin(2\pi \cdot \mathbf{fc} \cdot \mathbf{TIME} + \mathrm{mod} \cdot \sin(2\pi \cdot \mathbf{fm} \cdot \mathbf{TIME}))$$

Voltage Controlled Voltage Source

General Form	`E<name> <(+) node> <(-) node> <(+) controlling node>`
	`+ <(-) controlling node> <gain>`
	`E<name> <(+) <node> <(-) node> VALUE = { <expression> }`
	`E<name> <(+) <node> <(-) node> TABLE { <expression> } •`
	`+ < <input value>,<output value> >*`

Examples	`EBUFFER 1 2 10 11 5.0`
	`ESQROOT 5 0 VALUE = {5V*SQRT(V(3,2))}`
	`ET2 2 0 TABLE {V(ANODE,CATHODE)} = (0,0) (30,1)`

Description	In the first form, a specified voltage drop elsewhere in the circuit controls the voltage-source output. The second and third forms using the `VALUE` and `TABLE` keywords, respectively, are used in analog behavioral modeling. These two forms are automatically converted within **Xyce** to its prinicipal ABM device, the `B` nonlinear dependent source device. See the **Xyce** User's Guide for more information on analog behavioral modeling.
Parameters and Options	`(+)` and `(-)` nodes Output nodes. Positive current flows from the `(+)` node through the source to the `(-)` node. `<(+) controlling node>` and `<(-) controlling node>` Node pairs that define a set of controlling voltages. A given node may appear multiple times and the output and controlling nodes may be the same.

Current Controlled Current Source

General Form	`F<name> <(+) node> <(-) node>`
	`+ <controlling V device name> <gain>`
	`F<name> <(+) node> <(-) node> POLY(<value>)`
	`+ <controlling V device name>*`
	`+ < <polynomial coefficient value> >*`

Examples	`FSENSE 1 2 VSENSE 10.0`
	`FAMP 13 0 POLY(1) VIN 0 500`
	`FNONLIN 100 101 POLY(2) VCNTRL1 VCINTRL2 0.0 13.6 0.2 0.005`

Description	In the first form, a specified current elsewhere in the circuit controls the current-source output. The second form using the `POLY` keyword is used in analog behavioral modeling. This form is automatically converted within **Xyce** to its prinicipal ABM device, the `B` nonlinear dependent source device. See the **Xyce** User's Guide for more information on analog behavioral modeling.
Parameters and Options	`(+)` and `(-)` nodes Output nodes. Positive current flows from the `(+)` node through the source to the `(-)` node. `<controlling V device name>` The controlling voltage source which must be an independent voltage source (V device).

Current Controlled Voltage Source

The syntax of this device is exactly the same as for a Current Controlled Current Source. For a Current-Controlled Voltage Source just substitute an H for the F. The H device generates a voltage, whereas the F device generates a current.

Voltage Controlled Current Source

General Form

```
G<name> <(+) node> <(-) node> <(+) controlling node>
+ <(-) controlling node> <transconductance>
G<name> <(+) <node> <(-) node> VALUE = { <expression> }
G<name> <(+) <node> <(-) node> TABLE { <expression> } =
+ < <input value>,<output value> >*
```

Examples

```
GBUFFER 1 2 10 11 5.0
GPSK 11 6 VALUE = {5MA*SIN(6.28*10kHz*TIME+V(3))}
GA2 2 0 TABLE {V(5)} = (0,0) (1,5) (10,5) (11,0)
```

Description

In the forst form, a specified voltage drop elsewhere in the circuit controls the current-source output. The second and third forms using the `VALUE` and `TABLE` keywords, respectively, are used in analog behavioral modeling. These two forms are automatically converted within **Xyce** to its prinicipal ABM device, the `B` nonlinear dependent source device. See the **Xyce** User's Guide for more information on analog behavioral modeling.

Parameters and Options

`(+)` and `(-)` nodes

Output nodes. Positive current flows from the `(+)` node through the source to the `(-)` node.

`<(+) controlling node>` and `<(-) controlling node>`

Node pairs that define a set of controlling voltages. A given node may appear multiple times and the output and controlling nodes may be the same.

Nonlinear Dependent Source

General Form	`B<name> <(+) node> <(-) node> V={ABM expression}` `B<name> <(+) node> <(-) node> I={ABM expression}`

Examples	`B1 2 0 V={sqrt(V(1))}`
	`B2 4 0 V={V(1)*TIME}`
	`B3 4 2 I={I(V1) + V(4,2)/100}`
	`B4 5 0 V={Table {V(5)}=(0,0) (1.0,2.0) (2.0,3.0) (3.0,10.0)}`

Description	The nonlinear dependent source device, also known as the B-source device, is used in analog behavioral modeling (ABM). The `(+)` and `(-)` nodes are the output nodes. Positive current flows from the `(+)` node through the source to the `(-)` node.
Comments	See the "Analog Behavioral Modeling" chapter of the **Xyce** User's Guide for more information on the Bsource device and ABM expressions, and the "Parameters and Expressions" section of the User's Guide for more information on expressions in general. **Note: the braces surrounding all expressions are required.**

Bipolar Junction Transistor (BJT)

General Form	`Q<name> < collector node> <base node> <emitter node>` `+ [substrate node] <model name> [area value]`

Examples	`Q2 10 2 9 PNP1`
	`Q12 14 2 0 1 NPN2 2.0`
	`Q6 VC 4 11 [SUB] LAXPNP`

Model Form	`.MODEL <model name> NPN [model parameters]` `.MODEL <model name> PNP [model parameters]`

Parameters and Options	`[substrate node]`
	Optional and defaults to ground. Since **Xyce** permits alphanumeric node names and because there is no easy way to make a distinction between these and the model names, the name (not a number) used for the substrate node must be enclosed in square brackets `[ ]`. Otherwise, nodes would be interpreted as model names. See the third example above.
	`[area value]`
	The relative device area with a default value of 1.

Comments	The BJT is modeled as an intrinsic transistor using ohmic resistances in series with the collector (RC/area), with the base (value varies with current, see BJT equations) and with the emitter (RE/area).For model parameters with optional names, such as VAF and VA (the optional name is in parentheses), either may be used.For model types NPN and PNP, the isolation junction capacitance is connected between the intrinsic-collector and substrate nodes. This is the same as in SPICE and works well for vertical IC transistor structures. For lateral IC transistor structures there is a third model, LPNP, where the isolation junction capacitance is linked between the intrinsic-base and substrate nodes.

BJT Level selection

Xyce supports two BJT models: the level 1 model is based on the standard SPICE 3F5 BJT model, and the level 2 model augments the level 1 model with radiation effects [6]. Specify the model parameter **LEVEL** to select the model.

BJT Operating Temperature

Model parameters may be assigned unique measurement temperatures using the **TNOM** model parameter. See BJT model parameters for more information.

Level=1 Model Parameters

Table 2.14 gives the available model parameters for the level 1 BJT.

Model parameters	Description	Units	Default
BF	Ideal Maximum Forward Beta		100.0
BR	Ideal Maximum Reverse Beta		1.0
CJC	Base-collector Zero-bias p-n Capacitance	farad	0.0
CJE	Base-emitter Zero-bias p-n Capacitance	farad	0.0
CJS (CCS)	Substrate Zero-bias p-n Capacitance	farad	0.0
EG	Bandgap Voltage (Barrier Height)	eV	1.11
FC	Forward-bias Depletion Capacitor Coefficient		0.5
IKF (IK)	Corner for Forward-beta High-current Roll-off	amp	1E99
IKR	Corner for Reverse-beta High-current Roll-off	amp	1E99
IRB	Current at which Rb Falls off by half	amp	0.0
IS	Transport Saturation Current	amp	1E-16
ISC	Base-collector Leakage Saturation Current	amp	0.0
ISE	Base-emitter Leakage Saturation Current	amp	0.0
ITF	Transit Time Dependency on Ic	amp	0.0
KF	Flicker Noise Coefficient		0.0
MJC	Base-collector p-n Grading Factor		0.33
MJE	Base-emitter p-n Grading Factor		0.33
MJS	Substrate p-n Grading Factor		0.0
NC	Base-collector Leakage Emission Coefficient		2.0
NE	Base-emitter Leakage Emission Coefficient		1.5
NF	Forward Current Emission Coefficient		1.0
NR	Reverse Current Emission Coefficient		1.0
PTF	Excess Phase @ $1/(2\pi \cdot \mathrm{TF})$Hz	degree	0.0
RB	Zero-bias (Maximum) Base Resistance	ohm	0.0
RBM	Minimum Base Resistance	ohm	0.0
RC	Collector Ohmic Resistance	ohm	0.0
RE	Emitter Ohmic Resistance	ohm	0.0
TF	Ideal Forward Transit Time	sec	0.0
TR	Ideal Reverse Transit Time	sec	0.0

Model parameters	Description	Units	Default
TNOM	Parameter Measurement Temperature	°C	27.0
VAF	Forward Early Voltage	volt	1E99
VAR	Reverse Early Voltage	volt	1E99
VJC (PC)	Base-collector Built-in Potential	volt	0.75
VJE (PE)	Base-emitter Built-in Potential	volt	0.75
VJS (PS)	Substrate Built-in Potential	volt	0.75
VTF	Transit Time Dependency on V_{bc}	volt	1E99
XCJC	Fraction of **CJC** Connected Internally to **RB**		1.0
XTB	Forward and Reverse Beta Temperature Coefficient		0.0
XTF	Transit Time Bias Dependence Coefficient		0.0

Table 2.14: BJT Model Parameters.

Level=2 Model Parameters

Table 2.15 gives the available model parameters for the level 2 radiation-aware BJT.

Model parameters	Description	Units	Default
BF	Ideal Maximum Forward Beta		100.0
BR	Ideal Maximum Reverse Beta		1.0
CJC	Base-collector Zero-bias p-n Capacitance	farad	0.0
CJE	Base-emitter Zero-bias p-n Capacitance	farad	0.0
CJS (CCS)	Substrate Zero-bias p-n Capacitance	farad	0.0
EG	Bandgap Voltage (Barrier Height)	eV	1.11
FC	Forward-bias Depletion Capacitor Coefficient		0.5
IKF (IK)	Corner for Forward-beta High-current Roll-off	amp	1E99
IKR	Corner for Reverse-beta High-current Roll-off	amp	1E99
IRB	Current at which Rb Falls off by half	amp	0.0
IS	Transport Saturation Current	amp	1E-16
ISC	Base-collector Leakage Saturation Current	amp	0.0
ISE	Base-emitter Leakage Saturation Current	amp	0.0
ITF	Transit Time Dependency on Ic	amp	0.0
KF	Flicker Noise Coefficient		0.0

Model parameters	Description	Units	Default
MJC	Base-collector p-n Grading Factor		0.33
MJE	Base-emitter p-n Grading Factor		0.33
MJS	Substrate p-n Grading Factor		0.0
NC	Base-collector Leakage Emission Coefficient		2.0
NE	Base-emitter Leakage Emission Coefficient		1.5
NF	Forward Current Emission Coefficient		1.0
NR	Reverse Current Emission Coefficient		1.0
PTF	Excess Phase @ $1/(2\pi \cdot \mathrm{TF})$Hz	degree	0.0
RB	Zero-bias (Maximum) Base Resistance	ohm	0.0
RBM	Minimum Base Resistance	ohm	0.0
RC	Collector Ohmic Resistance	ohm	0.0
RE	Emitter Ohmic Resistance	ohm	0.0
TF	Ideal Forward Transit Time	sec	0.0
TR	Ideal Reverse Transit Time	sec	0.0
TNOM	Parameter Measurement Temperature	°C	27.0
VAF	Forward Early Voltage	volt	1E99
VAR	Reverse Early Voltage	volt	1E99
VJC (PC)	Base-collector Built-in Potential	volt	0.75
VJE (PE)	Base-emitter Built-in Potential	volt	0.75
VJS (PS)	Substrate Built-in Potential	volt	0.75
VTF	Transit Time Dependency on V_{bc}	volt	1E99
XCJC	Fraction of **CJC** Connected Internally to **RB**		1.0
XTB	Forward and Reverse Beta Temperature Coefficient		0.0
XTF	Transit Time Bias Dependence Coefficient		0.0
BC	Doping density in base	m^{-3}	1E22
BDN	Diffusion constant of electrons in base at low injection	m^2/s	1E-3
BDP	Diffusion constant of holes in base at low injection	m^2/s	2E-4
BTAUINF	Electron lifetime in base at saturated SRH condition	sec	4E-8
BTAU0	Electron lifetime in base at low injection	sec	2E-8
BW	Width of base	m	1E-6
CC	Doping density in collector	m^{-3}	5E20

Model parameters	Description	Units	Default
CCHI	Doping density in n^+-region (used when ISCHILO=1)	m^{-3}	1E24
CDN	Diffusion constant of electrons in collector at low injection	m^2/s	5E-3
CDNHI	Diffusion constant of electrons in n^+-region at low injection (when ISCHILO=1)	m^2/s	4E-3
CDP	Diffusion constant of holes in collector at low injection	m^2/s	7E-4
CDPHI	Diffusion constant of holes in n^+-region at low injection (when ISCHILO=1)	m^2/s	8E-4
CTAUINF	Hole lifetime in collector at saturated SRH condition	sec	2E-6
CTAUINFHI	Hole lifetime in collector in n^+-region at saturated SRH condition (when ISHILO=1)	sec	2E-7
CTAUO	Hole lifetime in collector at low injection	sec	1E-6
CTAUOHI	Hole lifetime in collector in n^+-region at low injection (when ISHILO=1)	sec	1E-7
CW	Width of collector	m	4E-6
CWHI	Width of n^+-region (when ISCHILO=1)	m	5E-5
DEVICEAREA	Area of device	m^2	1.0
EC	Doping density in emitter	m^{-3}	1E25
EDN	Diffusion constant of electrons in emitter at low injection	m^2/s	3E-3
EDP	Diffusion constant of holes in emitter at low injection	m^2/s	2E-4
ETAUINF	Hole lifetime in emitter at saturated SRH condition	sec	4E-8
ETAUO	Hole lifetime in emitter at low injection	sec	2E-8
EW	Width of emitter	m	2E-6
ISCHILOW	"Is the collector a Hi-Lo structure?" (0=no, 1=yes) *CAUTION: Tor Fjeldly's paper refers to a parameter "ISCHILO", but the code actually references "ISCHILOW"! If "ISCHILO" is used in a netlist then the parameter is ignored and the default value for* ISCHILOW *is used*	-	0 (no)
GRORDER	Order of magnitude of radiation generation rate	-	25
NI	Intrinsic concentration	m^{-3}	1.45E16

Model parameters	Description	Units	Default
PERMITTIVITY	Dielectric permittivity of the semiconductor	F/m	1.0443E-10
RAUGN	Auger recombination rate for electrons in p-region	cm^6/s	1.1E-42
RAUGP	Auger recombination rate for holes in n-region	cm^6/s	3E-43
FUNCTIONTYPE	Selects the shape of the radiation pulse. May take the following values: 0 (none), 1 (pulse wave form), 2 (sinusoidal wave form), 3 (exponential wave form), 4 (piecewise linear wave form), 5 (frequency modulated wave form), 6 (linearly ramped radiation), or 7 (take radiation wave form from data file)	-	0 (none)
PULSEV1	Initial value of radiation pulse (FUNCTIONTYPE=1)	volt	0.0
PULSEV2	Maximum value of radiation pulse (FUNCTIONTYPE=1)	volt	4.3
PULSETD	Delay of radiation pulse (FUNCTIONTYPE=1)	sec	0.0
PULSETR	Rise time of radiation pulse (FUNCTIONTYPE=1)	sec	1E-5
PULSETF	Fall time of radiation pulse (FUNCTIONTYPE=1)	sec	1E-5
PULSEPW	Width of radiation pulse (FUNCTIONTYPE=1)	sec	1E-5
PULSEPER	Period of radiation pulse (FUNCTIONTYPE=1)	sec	1E10
SINVO	Initial value of sinusoidal radiation voltage (FUNCTIONTYPE=2)	volt	0.0
SINVA	Amplitude of sinusoidal radiation voltage (FUNCTIONTYPE=2)	volt	0.0
SINFREQ	Frequency of sinusoidal radiation voltage (FUNCTIONTYPE=2)	Hz	1.0
SINTD	Delay of sinusoidal radiation voltage (FUNCTIONTYPE=2)	sec	0.0
SINTHETA	Inverse of time constant for exponential envelope of sinusoidal radiation voltage (FUNCTIONTYPE=2)	sec^{-1}	0.0
EXPV1	Initial value of exponential cusp radiation voltage (FUNCTIONTYPE=3)	volt	0.0

Model parameters	Description	Units	Default
EXPV2	Height of exponential cusp radiation voltage (FUNCTIONTYPE=3)	volt	0.0
EXPTD1	Rise delay time of exponential cusp radiation voltage (FUNCTIONTYPE=3)	sec	0.0
EXPTAU1	Rise time constant of exponential cusp radiation voltage (FUNCTIONTYPE=3)	sec	1e-2
EXPTD2	Fall delay time of exponential cusp radiation voltage (FUNCTIONTYPE=3)	sec	1E10
EXPTAU2	Fall time constant of exponential cusp radiation voltage (FUNCTIONTYPE=3)	sec	1E10
SFFMV0	Offset of single frequency FM curve (FUNCTIONTYPE=4)	volt	0.0
SFFMVA	Amplitude of single frequency FM curve (FUNCTIONTYPE=4)	volt	0.0
SFFMFC	Carrier frequency of single frequency FM curve (FUNCTIONTYPE=4)	Hz	1.0
SFFMMDI	Modulation index of single frequency FM curve (FUNCTIONTYPE=4)	-	0.0
SFFMFS	Signal frequency of single frequency FM curve (FUNCTIONTYPE=4)	Hz	0.0
PWLT1	Time of point 1 of piecewise linear curve (FUNCTIONTYPE=5)	sec	1
PWLT2	Time of point 2 of piecewise linear curve (FUNCTIONTYPE=5)	sec	2
PWLT3	Time of point 3 of piecewise linear curve (FUNCTIONTYPE=5)	sec	3
PWLT4	Time of point 4 of piecewise linear curve (FUNCTIONTYPE=5)	sec	4
PWLT5	Time of point 5 of piecewise linear curve (FUNCTIONTYPE=5)	sec	5
PWLV1	Height of point 1 of piecewise linear curve (FUNCTIONTYPE=5)	volt	0
PWLV2	Height of point 2 of piecewise linear curve (FUNCTIONTYPE=5)	volt	0
PWLV3	Height of point 3 of piecewise linear curve (FUNCTIONTYPE=5)	volt	0
PWLV4	Height of point 4 of piecewise linear curve (FUNCTIONTYPE=5)	volt	0
PWLV5	Height of point 5 of piecewise linear curve (FUNCTIONTYPE=5)	volt	0

Model parameters	Description	Units	Default
LINET1	Time of point 1 for linear radiation voltage (FUNCTIONTYPE=6)	sec	0
LINEV1	Height of point 1 for linear radiation voltage (FUNCTIONTYPE=6)	volt	0
LINEK	Slope linear radiation voltage (FUNCTIONTYPE=6)	-	0
PULSEDATA	Filename of (T,V) pairs to use instead of internal radiation source (FUNCTIONTYPE=7) (filename must be in double quotation marks, e.g. PULSEDATA="pulse.dat'	-	-

Table 2.15: Level 2 BJT Model Parameters.

BJT Equations

The BJT implementation within **Xyce** is based on [1]. The equations in this section describe an NPN transistor. For the PNP device, reverse the signs of all voltages and currents. The equations use the following variables:

V_{be} = intrinsic base-intrinsic emitter voltage
V_{bc} = intrinsic base-intrinsic collector voltage
V_{bs} = intrinsic base-substrate voltage
V_{bw} = intrinsic base-extrinsic collector voltage (quasi-saturation only)
V_{bx} = extrinsic base-intrinsic collector voltage
V_{ce} = intrinsic collector-intrinsic emitter voltage
V_{js} = (NPN) intrinsic collector-substrate voltage
= (PNP) intrinsic substrate-collector voltage
= (LPNP) intrinsic base-substrate voltage
V_t = kT/q (thermal voltage)
V_{th} = threshold voltage
k = Boltzmann's constant
q = electron charge
T = analysis temperature (K)
T_0 = nominal temperature (set using $\mathbf{TNOM}$ option)

Other variables are listed above in BJT Model Parameters.

DC Current

The BJT model is based on the Gummel and Poon model [7] where the different terminal currents are written

$$\begin{aligned} I_b &= -I_{be} - I_{bc} + I_{re} + I_{rc} + S\frac{dQ_b}{dt} \\ I_e &= -I_{cc} - I_{be} + I_{re} + S\frac{dQ_{dife}}{dt} + C_{de}\frac{dV_{be}}{dt} \\ I_c &= -I_{cc} - I_{bc} + I_{rc} + S\frac{dQ_{difc}}{dt} + C_{dc}\frac{dV_{bc}}{dt} \end{aligned}$$

Here, Q_{dife} and Q_{difc} are the hole charges per unit area in the base affiliated with the electrons introduced across the emitter-base and collector-base junctions, respectively. Also, Q_{be} and Q_{bc} are donations to the hole charge of the base affiliated with the differences in the depletion regions of the emitter-base and collector-base junctions, respectively. Lastly, S is the cross-sectional area of the device. The intermediate currents used are defined as

$$\begin{aligned} -I_{be} &= \frac{\mathbf{IS}}{\mathbf{BF}}\left[\exp\left(\frac{V_{be}}{\mathbf{NF}V_{th}}\right) - 1\right] \\ -I_{cc} &= \frac{Q_{bo}}{Q_b}\mathbf{IS}\left[\exp\left(\frac{V_{be}}{\mathbf{NF}V_{th}}\right) - \exp\left(\frac{V_{bc}}{\mathbf{NF}V_{th}}\right)\right] \\ -I_{bc} &= \frac{\mathbf{IS}}{\mathbf{BR}}\left[\exp\left(\frac{V_{bc}}{\mathbf{NE}V_{th}}\right) - 1\right] \\ I_{re} &= \mathbf{ISE}\left[\exp\left(\frac{V_{be}}{\mathbf{NE}V_{th}}\right) - 1\right] \\ I_{rc} &= \mathbf{ISC}\left[\exp\left(\frac{V_{bc}}{\mathbf{NC}V_{th}}\right) - 1\right] \end{aligned}$$

where the last two terms are the generation/recombination currents related to the emitter and collector junctions, respectively. The charge Q_b is the majority carrier charge in the base at large injection levels and is a key difference in the Gummel-Poon model over the earlier Ebers-Moll model.

Capacitance Terms

The p-n diode capacitance consists of a depletion layer capacitance C_d and a diffusion capacitance C_{dif}. The first is given by

$$C_d = \begin{cases} \mathbf{CJO}\left(1 - \frac{V_{di}}{\mathbf{VJ}}\right)^{-\mathbf{M}} & V_{di} \leq \mathbf{FC}\cdot\mathbf{VJ} \\ \mathbf{CJO}\,(1-\mathbf{FC})^{-(1+\mathbf{M})}\left[1 - \mathbf{FC}(1+\mathbf{M}) + \mathbf{M}\frac{V_{di}}{\mathbf{VJ}}\right] & V_{di} > \mathbf{FC}\cdot\mathbf{VJ} \end{cases}$$

The diffusion capacitance (sometimes referred to as the transit time capacitance) is

$$C_{dif} = \mathbf{TT}G_d = \mathbf{TT}\frac{dI}{dV_{di}}$$

where G_d is the junction conductance and I is the diode dc current given above.

Temperature Effects

The diode model contains explicit temperature dependencies in the ideal diode current, the generation/recombination current and the breakdown current. Further temperature dependencies are present in the diode model via the saturation current and the built-in or contact voltage

$$
\begin{aligned}
I_S(T) &= \mathbf{IS} \cdot \left(\frac{T}{T_0}\right)^{\mathbf{XTI/N}} \exp\left[\frac{\mathbf{EG} \cdot (T - T_0)}{\mathbf{N} \cdot qV_{th}T_0}\right] \\
V_{bi}(T) &= \frac{T}{T_0}\mathbf{VJ} - 2V_{th}\ln\left(\frac{T}{T_0}\right)^{1.5} - \frac{1}{q}\left[\frac{T}{T_0}E_{g0} - E_g(T)\right] \\
E_g(T) &= E_{g0} - \frac{\alpha T^2}{\beta + T}
\end{aligned}
$$

where, for silicon, $\alpha = 7.02 \times 10^{-4}\ eV/K$, $\beta = 1108\ K$ and $E_{g0} = 1.16\ eV$.

Radiation Effects

The equations for the radiation effects are beyond the scope of this manual. See [6] for details.

Caveats

It has been determined that the handling of current crowding effects in the level 2 BJT is not correct at this time. Specifying a finite value for **IRB** (current at which base resistance is at half of maximum) enables the current-dependence of the base resistance due to current crowding; using this option with the radiation aware BJT will cause anomalous currents due to the mishandling of this term in the presence of photocurrent. Until this aspect of the model is fixed, we recommend setting **IRB** to zero, disabling the current crowding effect.

The area parameter that can be specified on the instance line scales many of the device model parameters, and is usually used to specify a number of equivalent parallel devices rather than an actual device area. In the original AimSPICE and ChileSPICE implementation this same area parameter was used to scale the photocurrent. Many of the extracted BJT models in use at Sandia were designed to be used with the instance area parameter omitted. Using the radiation aware model with an area of 1.0 results in absurdly high photocurrents, but specifying any other area causes model parameters to be scaled away from the values originally extracted for the real devices.

To address this problem, the **Xyce** implementation of the radiation-aware device has defined a new model parameter, **DEVICEAREA**, that can be specified in the model statement instead of the instance line. In the **Xyce** implementation, the instance line area is used only to scale the model parameters in a manner consistent with SPICE 3F5, and the **DEVICEAREA** parameter is used only to scale the photocurrent. **This is incompatible with ChileSPICE and Aim-SPICE. Xyce will allow real device models to be used in the radiation context by adding the radiation specific parameters and the DEVICEAREA parameter to the model statement, whereas the other simulators will**

require that several level 1 model parameters be converted to per-unit-area values before use with the level 2 model.

For further information on BJT models, see [7]. For a thorough description of the U.C. Berkeley SPICE models see Reference [8].

MOS Field Effect Transistor (MOSFET)

General Form

```
M<name> <drain node> <gate node> <source node>
+ <bulk/substrate node> <model name>
+ [L=<value>] [W=<value>]
+ [AD=<value>] [AS=<value>]
+ [PD=<value>] [PS=<value>]
+ [NRD=<value>] [NRS=<value>]
+ [M=<value]
```

Examples

```
M5 4 12 3 0 PNOM L=20u W=10u
M3 5 13 10 0 PSTRONG
M6 7 13 10 0 PSTRONG M=2
M8 10 12 100 100 NWEAK L=30u W=20u
+ AD=288p AS=288p PD=60u PS=60u NRD=14 NRS=24
```

Model Form

```
.MODEL <model name> NMOS [model parameters]
.MODEL <model name> PMOS [model parameters]
```

Parameters and Options

`L` and `W`

The MOSFET channel length and width that are decreased to get the actual channel length and width. They may be given in the device `.MODEL` or `.OPTIONS` statements. The value in the device statement overrides the value in the model statement, which overrides the value in the `.OPTIONS` statement. Defaults for `L` and `W` may be set in the `.OPTIONS` statement. If `L` or `W` values are not given, their default value is 100 u.

`AD` and `AS`

The drain and source diffusion areas. Defaults for `AD` and `AS` can be set in the `.OPTIONS` statement. If `AD` or `AS` defaults are not set, their default value is 0.

`PD` and `PS`

The drain and source diffusion perimeters. Their default value is 0.

Parameters and Options (cont.)	NRD, NRS	Multipliers (in units of squares) that can be multiplied by RSH to yield the parasitic (ohmic) resistances of the drain (RD) and source (RS), respectively. NRD, NRS default to 0. Consider a square sheet of resistive material. Analysis shows that the resistance between two parallel edges of such a sheet depends upon its composition and thickness, but is independent of its size as long as it is square. In other words, the resistance will be the same whether the square's edge is 2 mm, 2 cm, or 2 m. For this reason, the sheet resistance of such a layer, abbreviated RSH, has units of ohms per square.
	M	If specified, the value is used as a number of parallel MOSFETs to be simulated. For example, if M=2 is specified, we simulate two identical mosfets connected to the same nodes in parallel.
Comments	The simulator provides three MOSFET device models, which differ in the formulation of the I-V characteristic. The LEVEL parameter selects among different models as shown below.	

MOSFET Operating Temperature

Model parameters may be assigned unique measurement temperatures using the TNOM model parameter. See the MOSFET model parameters for more information.

Model Parameters

Table 2.16 gives the available model parameters for the MOSFET.

All MOSFET models

The parameters shared by all MOSFET model levels are principally parasitic element values (e.g., series resistance, overlap capacitance, etc.).

Model levels 1, and 3

The DC behaviors of the level 1 and 3 MOSFET models are defined by the parameters VTO, KP, LAMBDA, PHI, and GAMMA. The simulator calculates these if the process parameters (e.g., TOX, and NSUB) are specified, but these are always overridden by any user-defined values. The VTO value is positive (negative) for modeling the enhancement mode and negative (positive) for the depletion mode of N-channel (P-channel) devices.

For MOSFETs, the capacitance model enforces charge conservation, influencing just the

Level 1 and 3 models.

Effective device parameter lengths and widths are calculated as follows:

$$P_i = P_0 + P_L/L_e + P_W/W_e$$

where

$$\begin{aligned} L_e &= \text{effective length} = \mathbf{L} - (2 \cdot \mathbf{LD}) \\ W_e &= \text{effective width} = \mathbf{W} - (2 \cdot \mathbf{WD}) \end{aligned}$$

See `.MODEL` (model definition) for more information.

Model level 9 (BSIM3 version 3.1)

The University of California, Berkeley BSIM3 model is a physical-based model with a large number of dependencies on essential dimensional and processing parameters. It incorporates the key effects that are critical in modeling deep-submicrometer MOSFETs. These include threshold voltage reduction, nonuniform doping, mobility reduction due to the vertical field, bulk charge effect, carrier velocity saturation, drain-induced barrier lowering (DIBL), channel length modulation (CLM), hot-carrier-induced output resistance reduction, subthreshold conduction, source/drain parasitic resistance, substrate current induced body effect (SCBE) and drain voltage reduction in LDD structure.

The BSIM3 Version 3.1 model is a deep submicron MOSFET model with several major enhancements (over earlier versions). These include a single I-V formula used to define the current and output conductance for operating regions, improved narrow width device modeling, a superior capacitance model with improved short and narrow geometry models, a new relaxation-time model to better transient modeling and enhanced model fitting of assorted W/L ratios using a single parameter set. This version preserves the large number of integrated dependencies on dimensional and processing parameters of the Version 2 model. For further information, see Reference [9].

Additional notes

1. If any of the following BSIM3 3.1 model parameters are not specified, they are computed via the following:

   If $\mathbf{VTHO}$ is not specified, then:

   $$\mathbf{VTHO} = \mathbf{VFB} + \phi_s \mathbf{K1}\sqrt{\phi_s}$$

   where:

   $$\mathbf{VFB} = -1.0$$

   If $\mathbf{VTHO}$ is given, then:

   $$\begin{aligned} \mathbf{VFB} &= \mathbf{VTHO} - \phi_s + \mathbf{K1}\sqrt{phi_s} \\ \mathbf{VBX} &= \phi_s - \frac{q \cdot \mathbf{NCH} \cdot \mathbf{XT}^2}{2\varepsilon_{si}} \\ \mathbf{CF} &= \left(\frac{2\varepsilon_{ox}}{\pi}\right) \ln\left(1 + \frac{1}{4 \times 10^7 \cdot \mathbf{TOX}}\right) \end{aligned}$$

where:

$$E_g(T) = \text{the energy bandgap at temperature } T = 1.16 - \frac{T^2}{7.02 \times 10^4 (T + 1108)}$$

2. If **K1** and **K2** are not given then they are computed via the following:

$$\begin{aligned} \mathbf{K1} &= \mathbf{GAMMA2} - 2 \cdot \mathbf{K2}\sqrt{\phi_s - \mathbf{VBM}} \\ \mathbf{K2} &= \frac{(\mathbf{GAMMA1} - \mathbf{GAMMA2})(\sqrt{\phi_s - \mathbf{VBX}} - \sqrt{\phi_s})}{2\sqrt{\phi_s}(\sqrt{\phi_s - \mathbf{VBM}} - \sqrt{\phi_s}) + \mathbf{VBM}} \end{aligned}$$

where:

$$\begin{aligned} \phi_s &= 2V_t \ln\left(\frac{\mathbf{NCH}}{n_i}\right) \\ V_t &= kT/q \\ n_i &= 1.45 \times 10^{10} \left(\frac{T}{300.15}\right)^{1.5} \exp\left(21.5565981 - \frac{E_g(T)}{2V_t}\right) \end{aligned}$$

3. If **NCH** is not specified and **GAMMA1** is, then:

$$\mathbf{NCH} = \frac{\mathbf{GAMMA1^2} \times \mathbf{COX^2}}{2q\varepsilon_{si}}$$

If **GAMMA1** and **NCH** *are not* specified, then **NCH** defaults to $1.7 \times 10^{23}\ m^{-3}$ and **GAMMA1** is computed using **NCH**:

$$\mathbf{GAMMA1} = \frac{\sqrt{2q\varepsilon_{si} \cdot \mathbf{NCH}}}{\mathbf{COX}}$$

If **GAMMA2** is not specified, then:

$$\mathbf{GAMMA2} = \frac{\sqrt{2q\varepsilon_{si} \cdot \mathbf{NSUB}}}{\mathbf{COX}}$$

4. If **CGSO** is not specified and **DLC** > 0, then:

$$\mathbf{CGSO} = \begin{cases} 0, & ((\mathbf{DLC} \cdot \mathbf{COX}) - \mathbf{CGSL}) < 0 \\ 0.6 \cdot \mathbf{XJ} \cdot \mathbf{COX}, & ((\mathbf{DLC} \cdot \mathbf{COX}) - \mathbf{CGSL}) \geq 0 \end{cases}$$

5. If **CGDO** is not specified and **DLC** > 0, then:

$$\mathbf{CGDO} = \begin{cases} 0, & ((\mathbf{DLC} \cdot \mathbf{COX}) - \mathbf{CGSL}) < 0 \\ 0.6 \cdot \mathbf{XJ} \cdot \mathbf{COX}, & ((\mathbf{DLC} \cdot \mathbf{COX}) - \mathbf{CGSL}) \geq 0 \end{cases}$$

Model parameters	Description	Units	Default
	All Levels		

Model parameters	Description	Units	Default
AF	Flicker Noise Exponent		1.0
CBD	Zero-bias Bulk-drain p-n Capacitance	farad (F)	0.0
CBS	Zero-bias Bulk-source p-n Capacitance	F	0.0
CGBO	Gate-bulk Overlap Capacitance/Channel Length	F	0.0
CGDO	Gate-drain Overlap Capacitance/Channel Width	F	0.0
CGSO	Gate-source Overlap Capacitance/Channel Width	F	0.0
CJ	Bulk p-n Zero-bias Bottom Capacitance/Area	F/m^2	0.0
CJSW	Bulk p-n Zero-bias Sidewall Capacitance/Area	F/m^2	0.0
FC	Bulk p-n Forward-bias Capacitance Coefficient		0.5
IS	Bulk p-n Saturation Current	amp	1E-14
JS	Bulk p-n Saturation Current/Area	amp/m^2	0.0
KF	Flicker Noise Coefficient		0.0
L	Channel Length	m	**DEFL**
LEVEL	Model Index		**1**
MJ	Bulk p-n Bottom Grading Coefficient		0.5
MJSW	Bulk p-n Sidewall Grading Coefficient		0.33
PB	Bulk p-n Bottom Potential	volt	0.8
RSH	Drain, Source Diffusion Sheet Resistance	ohm	0.0
TT	Bulk p-n Transit Time	second (s)	0.0
TNOM	Parameter Measurement Temperature	°C	27.0
W	Channel Width	m	**DEFW**
	Levels 1and 3		
DELTA	Width Effect on Threshold		0.0
ETA	Static Feedback (Level 3)		0.0
GAMMA	Bulk Threshold Parameter	$volt\ (V)^{1/2}$	
KP	Transconductance Coefficient	amp/V^2	2.0E-5
KAPPA	Saturation Field Factor (Level 3)		0.2
LAMBDA	Channel-length Modulation (Levels 1)	V^{-1}	0.0
LD	Lateral Diffusion (Length)	m	0.0
NFS	Fast Surface State Density	cm^{-2}	0.0

Model parameters	Description	Units	Default
NSS	Surface State Density	cm^{-2}	
NSUB	Substrate Doping Density	cm^{-3}	0.0
PHI	Surface Potential	V	0.6
RD	Drain Ohmic Resistance	ohm	0.0
RS	Source Ohmic Resistance	ohm	0.0
THETA	Mobility Modulation (Level 3)	V^{-1}	0.0
TOX	Oxide Thickness	m	
TPG	Gate Material Type: +1 = opposite of substrate −1 = same as substrate 0 = aluminum		+1
UTRA	(Not Used) Mobility Degradation Transverse Field Coefficient		0.0
UO	Surface Mobility	$cm^{-2}/(V \cdot s)$	600
VMAX	Maximum Drift Velocity	m/s	0.0
VTO	Zero-bias Threshold Voltage	V	0.0
WD	Lateral Diffusion (Width)	m	0.0
XJ	Metallurgical Junction Depth (Level 3)	m	0.0
XQC	Fraction of Channel Charge Attributed to Drain		1.0
Level 9: Control Parameters			
CAPMOD	Flag for the Short-channel Capacitance Model		2.0
MOBMOD	Mobility Model Selector		1
NOIMOD	Flag for Noise Model		1
NQSMOD	Flag for NQS Model		0
PARAMCHK	Flag for Model Parameter Checking		0
Level 9: Capacitance Parameters			
CF	Fringing Field Capacitance	F/m	
CKAPPA	Coefficient for Lightly Doped Region Overlap Capacitance Fringing Field Capacitance	F/m	0.6
CLC	Constant Term for the Short-Channel Model	m	0.1E-6
CLE	Exponential Term for the Short-Channel Model		0.6

Model parameters	Description	Units	Default
CGBO	Gate-bulk Overlap Capacitance per Unit Channel Length	F/m	0.0
CGDL	Light-doped Drain-gate Region Overlap Capacitance	F/m	0.0
Level 9: Capacitance Parameters (cont.)			
CGDO	Non-LDD Region Drain-gate Overlap Capacitance per Unit Channel Length	F/m	
CGSL	Light-doped Source-gate Region Overlap Capacitance	F/m	0.0
CGSO	Non-LDD Region Source-gate Overlap Capacitance per Unit Channel Length	F/m	
CJ	Bottom Junction Capacitance per Unit Area	F/m^2	5.0E-4
CJSW	Source/Drain Side Junction Capacitance per Unit Periphery	F/m	5.0E-10
CJSWG	Source/Drain Gate Sidewall Junction Capacitance per Unit Width	F/m	**CJSW**
DLC	Length Offset Fitting Parameter from C-V	m	**LINT**
DWC	Width Offset Fitting Parameter from C-V	m	**WINT**
MJ	Bottom Junction Capacitance Grading Coefficient		0.5
MJSW	Source/Drain Side Junction Capacitance Grading Coefficient		0.33
MJSWG	Source/Drain Gate Sidewall Junction Capacitance Grading Coefficient		**MJSW**
PB	Bottom Built-in Potential	V	1.0
PBSW	Source/Drain Side Junction Built-in Potential	V	1.0
PBSWG	Source/Drain Gate Sidewall Junction Built-in Potential	V	**PBSW**
VFBCV	Flat-band Voltage Parameter (for **CAPMOD** = 0 only)	V	-1.0
XPART	Charge Partitioning Rate Flag		0.0
Level 9: Bin Description Parameters			
BINUNIT	Bin Unit Scale Selector		1.0
LMAX	Maximum Channel Length	m	1.0
LMIN	Minimum Channel Length	m	0.0
WMAX	Maximum Channel Width	m	1.0
WMIN	Minimum Channel Width	m	0.0
Level 9: DC Parameters			

Model parameters	Description	Units	Default
A0	Bulk Charge Effect Coefficient for Channel Length		1.0
A1	First Non-saturation Effect Parameter	V^{-1}	0.0
A2	Second Non-saturation Factor		1.0
AGS	Gate-bias Coefficient of Abulk	V^{-1}	0.0
ALPHA0	First Parameter of Impact-ionization Current	m/V	0.0
B0	Bulk Charge Effect Coefficient for Channel Width	m	0.0
B1	Bulk Charge Effect Width Offset	m	0.0
BETA0	Second Parameter of Impact-ionization Current	V	30.0
CDSC	Drain/Source to Channel Coupling Capacitance	F/m^2	2.4E-4
CDSCB	Body-bias Sensitivity of **CDSC**	F/Vm^2	0.0
CDSCD	Drain-bias Sensitivity of **CDSC**	F/Vm^2	0.0
CIT	Interface Trap Capacitance	F/m^2	0.0
DELTA	Effective **Vds** Parameter	V	0.01
DROUT	L-dependence Coefficient of the **DIBL** Correction Parameter in **Rout**		0.56
DSUB	**DIBL** Coefficient Exponent in Subthreshold Region		**DROUT**
DVT0	First Coefficient of Short-channel Effect on Threshold Voltage		2.2
DVT0W	First Coefficient of Narrow-width Effect on Threshold Voltage for Small-channel Length	m^{-1}	0.0
DVT1	Second Coefficient of Short-channel Effect on Threshold Voltage		0.53
DVT2	Body-bias Coefficient of Short-channel Effect on Threshold Voltage	V^{-1}	-0.032
DVTW1	Second Coefficient of Narrow-channel Effect on Threshold Voltage for Small Channel Length	m^{-1}	5.3E6
DVTW2	Body-bias Coefficient of Narrow-width Effect for Small Channel Length	V^{-1}	-0.032
DWB	Coefficient of Substrate Body Bias Dependence of **Weff**	$m/V^{1/2}$	0.0
DWG	Coefficient of Gate Dependence of **Weff**	m/V	0.0

Model parameters	Description	Units	Default
ETA0	**DIBL** Coefficient in Subthreshold Region		0.08
ETAB	Body-bias Coefficient for the Subthreshold **DIBL** Effect	V^{-1}	-0.07
JS	Source-drain Junction Saturation Current per Unit Area	amp/m^2	1.0E-4
JSW	Sidewall Saturation Current per Unit Length	amp/m	0.0
K1	First-order Body Effect Coefficient	$V^{1/2}$	0.5
K2	Second-order Body Effect Coefficient		0.0
K3	Narrow Width Coefficient		80.0
K3B	Body Effect Coefficient of **K3**	V^{-1}	0.0
KETA	Body-bias Coefficient of Bulk Charge Effect	V^{-1}	-0.047
LINT	Length Offset Fitting Parameter from I-V Without Bias	m	0.0
NFACTOR	Subthreshold Swing Factor		1.0
NGATE	Poly Gate Doping Concentration	cm^{-3}	0.0
NLX	Lateral Non-uniform Doping Parameter	m	1.74E-7
PCLM	Channel Length Modulation Parameter		1.3
PDIBLC1	First Output Resistance **DIBL** Effect Correction Parameter		0.39
PDIBLC2	Second Output Resistance **DIBL** Effect Correction Parameter		0.0086
PDIBLCB	Body Effect Coefficient of **DIBL** Correction Parameter	V^{-1}	0.0
PRWB	Body Effect Coefficient of **RDSW**	$V^{-1/2}$	0.0
PRWG	Gate-bias Effect Coefficient of **RDSW**	V^{-1}	0.0
PSCBE1	First Substrate Current Body Effect Parameter	V/m	4.24E8
PSCBE2	Second Substrate Current Body Effect Parameter	V/m	1.0E-5
PVAG	Gate Dependence of Early Voltage		0.0
RDSW	Parasitic Resistance per Unit Width	ohm-μm^{WR}	0.0
RSH	Source-drain Sheet Resistance	ohm/ square?	0.0
U0	Mobility at Temp = **TNOM** **NMOS** **PMOS**	$cm^2/V \cdot s$	670.0 250.0
UA	First-order Mobility Degradation Coefficient	m/V	2.25E-9

Model parameters	Description	Units	Default
UB	Second-order Mobility Degradation Coefficient	$(m/V)^2$	5.87E-19
UC	Body Effect of Mobility Degradation Coefficient **MOBMOD** = 1 **MOBMOD** = 3	m/V^2 $1/V$	-4.65E-11 -0.046
VBM	Maximum Applied Body-bias in Threshold Voltage Calculation	V	-3.0
VOFF	Offset Voltage in the Subthreshold Region at Large W and L	V	-0.08
VSAT	Saturation Velocity at Temp = **TNOM**	m/s	8.0E4
VTHO	Threshold Voltage at **Vbs** = 0 for Large **L**	V	0.7 (**NMOS**) -0.7 (**PMOS**)
WO	Narrow-width Parameter	m	2.5E-6
WINT	Width-offset Fitting Parameter from I-V Without Bias	m	0.0
WR	Width-offset from Weff for **Rds** Calculation		1.0
Level 9: Flicker Noise Parameters			
AF	Frequency Exponent		1.0
EF	Flicker Exponent		1.0
EM	Saturation Field	V/m	4.1E7
KF	Flicker Noise Parameter		0.0
NOIA	Noise Parameter A **NMOS** **PMOS**		1.0E20 9.9E18
NOIB	Noise Parameter B **NMOS** **PMOS**		5.0E4 2.4E3
NOIC	Noise Parameter C **NMOS** **PMOS**		-1.4E-12 1.4E-12
Level 9: NQS Parameter			
ELM	Elmore Constant of the Channel		5.0
Level 9: Process Parameters			
GAMMA1	Body Effect Coefficient Near the Surface	$V^{1/2}$	
GAMMA2	Body Effect Coefficient in the Bulk	$V^{1/2}$	
NCH	Channel Doping Concentration	cm^{-3}	1.7E17
NSUB	Substrate Doping Concentration	cm^{-3}	6.0E16
TOX	Gate-oxide Thickness	m	1.5E-8
VBX	**Vbs** at Which the Depletion Region = **XT**	V	
XJ	Junction Depth	m	1.5E-7
XT	Doping Depth	m	1.55E-7
Level 9: Temperature Parameters			

Model parameters	Description	Units	Default
AT	Temperature Coefficient for Saturation Velocity	m/s	3.3E4
KT1	Temperature Coefficient for Threshold Voltage	V	-0.11
KT1L	Channel Length Dependence of the Temperature Coefficient for Threshold Voltage	V · m	0.0
KT2	Body-bias Coefficient of Threshold Voltage Temperature Effect		0.022
NJ	Emission Coefficient of Junction		1.0
PRT	Temperature Coefficient for **RDSW**	ohm-μm	0.0
TNOM	Temperature at which Parameters are Extracted	°C	27.0
UA1	Temperature Coefficient for **UA**	m/V	4.31E-9
UB1	Temperature Coefficient for **UB**	$(m/V)^2$	-7.61E-18
UC1	Temperature Coefficient for **UC** **MOBMOD** = 1 or 2 **MOBMOD** = 3	m/V^2 1/V	-5.6E-11 -0.056
UTE	Mobility Temperature Exponent		-1.5
XT1	Junction Current Temperature Exponent Coefficient		3.0
	Level 9: W and L Parameters		
LL	Coefficient of Length Dependence for Length Offset	m^{LLN}	0.0
LLN	Power of Length Dependence for Length Offset		1.0
LW	Coefficient of Width Dependence for Length Offset	m^{LWN}	0.0
LWL	Coefficient of Length and Width Cross Term for Length Offset	$m^{LWN+LLN}$	0.0
LWN	Power of Width Dependence for Length Offset		1.0
WL	Coefficient of Length Dependence for Width Offset	m^{WLN}	0.0
WW	Coefficient of Width Dependence for Width Offset	m^{WWN}	0.0
WWL	Coefficient of Length and Width Cross Term for Width Offset	$m^{WWN+WLN}$	0.0
WWN	Power of Width Dependence of Width Offset		1.0

Table 2.16: MOSFET Model Parameters.

MOSFET Equations

The following equations define an N-channel MOSFET. The P-channel devices use a reverse the sign for all voltages and currents. The equations use the following variables:

V_{bs} = intrinsic substrate-intrinsic source voltage
V_{bd} = intrinsic substrate-intrinsic drain voltage
V_{ds} = intrinsic drain-substrate source voltage
V_{dsat} = saturation voltage
V_{gs} = intrinsic gate-intrinsic source voltage
V_{gd} = intrinsic gate-intrinsic drain voltage
V_t = kT/q (thermal voltage)
V_{th} = threshold voltage
C_{ox} = the gate oxide capacitance per unit area
f = noise frequency
k = Boltzmann's constant
q = electron charge
$Leff$ = effective channel length
$Weff$ = effective channel width
T = analysis temperature ($^\circ K$)
T_0 = nominal temperature (set using TNOM option)

Other variables are listed in the BJT Model Parameters Table 2.14.

All Levels

I_g = gate current $= 0$

I_b = bulk current $= I_{bs} + I_{bd}$

where

I_{bs} = bulk-source leakage current $= I_{ss}\left(e^{V_{bs}/(N-V_t)} - 1\right)$

I_{ds} = bulk-drain leakage current $= I_{ds}\left(e^{V_{bd}/(N-V_t)} - 1\right)$

where

if

$\mathbf{JS} = 0$, or $\mathbf{AS} = 0$ or $\mathbf{AD} = 0$

then

$I_{ss} = \mathbf{IS}$

$I_{ds} = \mathbf{IS}$

else

$I_{ss} = \text{AS} \times \mathbf{JS} + \text{PS} \times \mathbf{JSSW}$

$I_{ds} = \text{AD} \times \mathbf{JS} + \text{PD} \times \mathbf{JSSW}$

I_d = drain current $= I_{drain} - I_{bd}$

I_s = source current $= -I_{drain} - I_{bs}$

Level 1: Idrain

Normal Mode: $V_{ds} > 0$

Case 1

For cutoff region: $V_{gs} - V_{to} < 0$

$$I_{drain} = 0$$

Case 2

For linear region: $V_{ds} < V_{gs} - V_{to}$

$$I_{drain} = (W/L)(\mathbf{KP}/2)(1 + \mathbf{LAMBDA} \times V_{ds})V_{ds}(2(V_{gs} - V_{to}) - V_{ds})$$

Case 3

For saturation region: $0 \leq V_{gs} - V_{to} \leq V_{ds}$

$$I_{drain} = (W/L)(\mathbf{KP}/2)(1 + \mathbf{LAMBDA} \cdot V_{ds})(V_{gs} - V_{to})^2$$

where

$$V_{to} = \mathbf{VTO} + \mathbf{GAMMA} \cdot \left((\mathbf{PHI} - V_{bs})^{1/2}\right)^{1/2}$$

Inverted Mode: $V_{ds} < 0$

Here, simply switch the source and drain in the normal mode equations given above.

Level 3: Idrain

See Reference [10] below for detailed information.

Capacitance

Level 1 and 3

C_{bs} = bulk-source capacitance = area cap. + sidewall cap. + transit time cap.

C_{bd} = bulk-drain capacitance = area cap. + sidewall cap. + transit time cap.

where

if

CBS = 0 and **CBD** = 0

then

$$C_{bs} = \text{AS} \cdot \mathbf{CJ} \cdot C_{bsj} + \text{PS} \cdot \mathbf{CJSW} \cdot C_{bss} + \mathbf{TT} \cdot G_{bs}$$
$$C_{bd} = \text{AD} \cdot \mathbf{CJ} \cdot C_{bdj} + \text{PD} \cdot \mathbf{CJSW} \cdot C_{bds} + \mathbf{TT} \cdot G_{ds}$$

else

$$C_{bs} = \text{CBS} \cdot C_{bsj} + \text{PS} \cdot \mathbf{CJSW} \cdot C_{bss} + \mathbf{TT} \cdot G_{bs}$$
$$C_{bd} = \text{CBD} \cdot C_{bdj} + \text{PD} \cdot \mathbf{CJSW} \cdot C_{bds} + \mathbf{TT} \cdot G_{ds}$$

where

G_{bs} = DC bulk-source conductance = dI_{bs}/dV_{bs}
G_{bd} = DC bulk-drain conductance = dI_{bd}/dV_{bd}

if

$V_{bs} \leq \mathbf{FC} \cdot \mathbf{PB}$

then

$$C_{bsj} = (1 - V_{bs}/\mathbf{PB})^{-\mathbf{MJ}}$$
$$C_{bss} = (1 - V_{bs}/\mathbf{PBSW})^{-\mathbf{MJSW}}$$

if

$V_{bs} > \mathbf{FC} \cdot \mathbf{PB}$

then

$$C_{bsj} = (1 - \mathbf{FC})^{-(1+\mathbf{MJ})}(1 - \mathbf{FC}(1 + \mathbf{MJ}) + \mathbf{MJ} \cdot V_{bs}/\mathbf{PB})$$
$$C_{bss} = (1 - \mathbf{FC})^{-(1+\mathbf{MJSW})}(1 - \mathbf{FC}(1 + \mathbf{MJSW}) + \mathbf{MJSW} \cdot V_{bs}/\mathbf{PBSW})$$

if

$V_{bd} \leq \mathbf{FC} \cdot \mathbf{PB}$

then

$$C_{bdj} = (1 - V_{bd}/\mathbf{PB})^{-\mathbf{MJ}}$$
$$C_{bds} = (1 - V_{bd}/\mathbf{PBSW})^{-\mathbf{MJSW}}$$

if

$V_{bd} > \mathbf{FC} \cdot \mathbf{PB}$

then

$$C_{bdj} = (1 - \mathbf{FC})^{-(1+\mathbf{MJ})}(1 - \mathbf{FC}(1 + \mathbf{MJ}) + \mathbf{MJ} \cdot V_{bd}/\mathbf{PB})$$
$$C_{bds} = (1 - \mathbf{FC})^{-(1+\mathbf{MJSW})}(1 - \mathbf{FC}(1 + \mathbf{MJSW}))$$

C_{gs} = gate-source overlap capacitance = $\mathbf{CGSO} \cdot \mathbf{W}$
C_{gd} = gate-drain overlap capacitance = $\mathbf{CGDO} \cdot \mathbf{W}$
C_{gb} = gate-bulk overlap capacitance = $\mathbf{CGBO} \cdot \mathbf{L}$

Temperature Effects

All Levels

$$\mathbf{IS}(T) = \mathbf{IS} \cdot \exp\left(E_g(T_0) \cdot T/T_0 - E_g(T)\right)/V_t$$
$$\mathbf{JS}(T) = \mathbf{JS} \cdot \exp\left(E_g(T_0) \cdot T/T_0 - E_g(T)\right)/V_t$$
$$\mathbf{JSSW}(T) = \mathbf{JSSW} \cdot \exp\left(E_g(T_0) \cdot T/T_0 - E_g(T)\right)/V_t$$
$$\mathbf{PB}(T) = \mathbf{PB} \cdot T/T_0 - 3V_t \ln(T/T_0) - E_g(T_0) \cdot T/T_0 + E_g T$$
$$\mathbf{PBSW}(T) = \mathbf{PBSW} \cdot T/T_0 - 3V_t \ln(T/T_0) - E_g(T_0) \cdot T/T_0 + E_g T$$
$$\mathbf{PHI}(T) = \mathbf{PHI} \cdot T/T_0 - 3V_t \ln(T/T_0) - E_g(T_0) \cdot T/T_0 + E_g T$$

where

$$E_g(T) = \text{silicon bandgap energy} = 1.16 - 0.000702T^2/(T + 1108)$$
$$\mathbf{CBD}(T) = \mathbf{CBD} \cdot (1 + \mathbf{MJ} \cdot (0.0004(T - T_0) + (1 - \mathbf{PB}(T)/\mathbf{PB})))$$
$$\mathbf{CBS}(T) = \mathbf{CBS} \cdot (1 + \mathbf{MJ} \cdot (0.0004(T - T_0) + (1 - \mathbf{PB}(T)/\mathbf{PB})))$$
$$\mathbf{CJ}(T) = \mathbf{CJ} \cdot (1 + \mathbf{MJ} \cdot (0.0004(T - T_0) + (1 - \mathbf{PB}(T)/\mathbf{PB})))$$
$$\mathbf{CJSW}(T) = \mathbf{CJSW} \cdot (1 + \mathbf{MJSW} \cdot (0.0004(T - T_0) + (1 - \mathbf{PB}(T)/\mathbf{PB})))$$
$$\mathbf{KP}(T) = \mathbf{KP} \cdot (T/T_0)^{-3/2}$$
$$\mathbf{UO}(T) = \mathbf{UO} \cdot (T/T_0)^{-3/2}$$
$$\mathbf{MUS}(T) = \mathbf{MUS} \cdot (T/T_0)^{-3/2}$$
$$\mathbf{MUZ}(T) = \mathbf{MUZ} \cdot (T/T_0)^{-3/2}$$
$$\mathbf{X3MS}(T) = \mathbf{X3MS} \cdot (T/T_0)^{-3/2}$$

For a thorough description of MOSFET models see [11, 10, 12, 13, 14, 8, 9, 15, 16, 17].

Voltage-controlled Switch

General Form	`S<name> <(+) switch node> <(-) switch node>`
	`+ <(+) controlling node> <(-) controlling node>`
	`+ <model name> [ON] [OFF]`

Examples	`S1 21 23 12 10 SMOD1` `SSET 15 10 1 13 SRELAY`
Description	The voltage-controlled switch is a particular type of voltage-controlled resistor. This model is designed to help reduce numerical issues. See Special considerations below.
Comments	The resistance between the `<(+) switch node>` and the `<(-) switch node>` is dependent on the voltage between the `<(+) controlling node>` and the `<(-) controlling node>`. The resistance changes in a continuous manner between the `RON` and `ROFF` model parameters. A resistance of `1/GMIN` (`GMIN` can be specified as a `.OPTIONS DEVICE` parameter) is attached across the controlling nodes to prevent them from floating. Even though evaluating the switch model is computationally inexpensive, for transient analysis, **Xyce** steps through the transition section using small time- steps in order to accurately calculate the waveform. Thus, a circuit with many switch transitions can result in lengthy run times. The ON and OFF parameters are used to specify the initial state of the switch at the first step of the operating point calculation; this does not force the switch to be in that state, it only gives the operating point solver an initial state to work with. If it is known that the switch should be in a particular state in the operating point it could help convergence to specify one of these keywords.

Model Parameters

Table 2.17 gives the available model parameters for the voltage-controlled switch.

Model parameters	Description	Units	Default
`ROFF`	Off Resistance	ohm	1.0×10^6
`RON`	On Resistance	ohm	1.0
`VOFF`	Control Voltage for Off State	volt	0.0
`VON`	Control Voltage for On State	volt	1.0

Table 2.17. Voltage-controlled Switch Model Parameters.

Special Considerations

- Due to numerical limitations, **Xyce** can only manage a dynamic range of approximately 12 decades. Thus, it is recommended the user limit the ratio $\mathbf{ROFF/RON}$ to less than 10^{12}.
- Furthermore, it is a good idea to limit the narrowness of the transition region. This is because in the transition region, the switch has gain and the narrower the region, the higher the gain and the more potential for numerical problems. The smallest value allowed for $\|\mathbf{VON} - \mathbf{VOFF}\|$ is 1×10^{-12}.

Voltage-controlled switch equations

The equations in this section use the following variables:

$$\begin{array}{rcll} R_s & = & \text{switch resistance} & \\ V_c & = & \text{voltage across control nodes} & \\ L_m & = & \text{log-mean of resistor values} & = \ln\left(\sqrt{\mathbf{RON} \cdot \mathbf{ROFF}}\right) \\ L_r & = & \text{log - ratio of resistor values} & = \ln\left(\mathbf{RON}/\mathbf{ROFF}\right) \\ V_d & = & \text{difference of control voltages} & = \mathbf{VON} - \mathbf{VOFF} \end{array}$$

Switch Resistance

To compute the switch resistance, **Xyce** first calculates the "switch state" S as $S = (V_c - \mathbf{VOFF})/V_d$. The switch resistance is then:

$$R_s = \begin{cases} \mathbf{RON}, & S \geq 1.0 \\ \mathbf{ROFF}, & S \leq 0.0 \\ \exp\left(L_m + 0.75L_r(2S-1) - 0.25L_r(2S-1)^3\right), & 0 < S < 1 \end{cases}$$

Lossless (Ideal) Transmission Line

General Form	`T<name> <A port (+) node> <A port (-) node>`
	`+ <B port (+) node> <B port (-) node> [model name]`
	`+ Z0=<value> [TD=<value>] [F=<value> [NL=<value>]]`

Examples	`Tline inp inn outp outn Z0=50 TD=1us` `Tline2 inp inn outp outn Z0=50 F=1meg NL=1.0`
Description	The lossless transmission line device is a two port (`A` and `B`), bi-directional delay line. The `(+)` and `(-)` nodes define the polarity of a positive voltage at a port.
Comments	`Z0` is the characteristic impedance. The transmission line's length is specified by either `TD` (a delay in seconds) or by `F` and `NL` (a frequency and relative wavelength at `F`). `NL` defaults to 0.25 (`F` is the quarter-wave frequency). If `F` is given, the time delay is computed as $\frac{NL}{F}$. While both `TD` and `F` are optional, at least one of them must be given.

Model Parameters

Table 2.18 gives the available model parameters for the lossless transmission line.

Model parameters	Description	Units	Default
Z0	Characteristic impedance	ohm	-
TD	Transmission delay	second	-
F	Frequency for **NL**	Hz	-
NL	Relative wavelength	-	0.25

Table 2.18. Lossless (Ideal) Transmission Line Parameters.

Subcircuit

A subcircuit can be introduced into the circuit netlist using the specified nodes to substitute for the argument nodes in the definition. It provides a building block of circuitry to be defined a single time and subsequently used multiple times in the overall circuit netlists.

General Form

```
X<name> [node]* <subcircuit name> [PARAMS: [<name> = <value>]*]
```

Examples

```
X12 100 101 200 201 DIFFAMP
XBUFF 13 15 UNITAMP
XFOLLOW IN OUT VCC VEE OUT OPAMP
XFELT 1 2 FILTER PARAMS: CENTER=200kHz
XNANDI 25 28 7 MYPWR MYGND PARAMS: IO_LEVEL=2
```

Parameters and Options

`<subcircuit name>`

The name of the subcircuit's definition.

`PARAMS:`

Passed into subcircuits as arguments and into expressions inside the subcircuit.

Comments

There must be an equal number of nodes in the subcircuit call and in its definition.

Subcircuit references may be nested to any level. However, the nesting cannot be circular. For example, if subcircuit `A`'s definition includes a call to subcircuit `B`, then subcircuit `B`'s definition cannot include a call to subcircuit `A`.

2.3 PDE Devices

Semiconductor device simulation, which is based on a coupled set of partial differential equations (PDE's) is supported in **Xyce**. Such devices can be invoked from the circuit netlist, in a similar manner to traditional SPICE-style analog devices. One dimensional and two dimensional devices are supported, with the dimensionality determined by the device model level.

General Form, 1D:	Z<name> <node> <node> [model name] [na=<value>] [nd=<value>] [nx=<value>] [area=<value>] [graded=<value>] [wj=<value>] [l=<value>] [w=<value>] [tecplotlevel=<value>] [sgplotlevel=<value>] [gnuplotlevel=<value>] [node=<tabular data>] [region=<tabular data>] [bulkmaterial=<string>] [temp=<value>]
General Form, 2D:	Z<name> <node> <node> [node] [node] [model name] [na=<value>] [nd=<value>] [meshfile=<filename.msh>] [nx=<value>] [ny=<value>] [l=<value>] [w=<value>] [type=<string> [node=<tabular data>] [region=<tabular data>] [x0=<value>] [cyl=<value>] [tecplotlevel=<value>] [sgplotlevel=<value>] [gnuplotlevel=<value>] [txtdatalevel=<value>] [ph.a1=<value>] [ph.type=<string>] [ph.tstart=<value>] [ph.tstop=<value>] [photogen=<value>] [ph.td=<value>] [ph.tr=<value>] [ph.tf=<value>] [ph.pw=<value>] [ph.per=<value>] [bulkmaterial=<string>] [temp=<value>]
Comments:	Most of the PDE parameters are specified on the instance level. At this point the model statement is only used for specifying if the device is 1D or 2D, via the level parameter. Both the 1D and the 2D devices can construct evenly spaced meshes, internally. The 2D device also has the option of reading in an unstructured mesh from a external mesh file. The electrode tabular data specification is explained in detail in table 2.23 Similarly, the doping region tabular data specification is explained in detail in table 2.21.

PDE Device Parameters

Most PDE parameters are specified on the instance level.

Instance parameters	Description	Units	Default	Device Type
	All Levels			
`name`	The instance name must start with a Z.	-	-	1D, 2D
`node`	Minimum of 2 connecting circuit nodes. The 2D device may have as many as 4 nodes, while the 1D device can only have 2. The node parameter is a tabular parameter, which specifies all the electrode attributes. See table 2.23 for a list.	-	-	1D, 2D
`region`	Specifies doping regions. Like the node parameter, this is a tabular parameter, containing several attributes.. See table 2.21 for a list.	-	-	1D, 2D
`area`	Cross sectional area of the device.	-	1.0	1D, 2D
`tecplotlevel`	Setting for Tecplot output: 0 - no Tecplot files 1 - Tecplot files, each output in a separate file. 2 - Tecplot file, each output appended to a single file. Tecplot files will have the `.dat` suffix, and the prefix will be the name of the device instance	-	1	1D, 2D
`sgplotlevel`	Flag for sgplot output. 0 - no sgplot files. 1 - sgplot files. sgplot is a plotting program that comes as part of the SG Framework [18]. sgplot files will have the *.res suffix, and the prefix will be the name of the device instance	-	0	1D, 2D
`gnuplotlevel`	Flag for gnuplot output. 0 - no gnuplot files. 1 - gnuplot files. gnuplot is an open source plotting program that is usually installed on Linux systems. gnuplot files will have the *Gnu.dat suffix, and the prefix will be the name of the device instance.	-	0	1D, 2D

Instance parameters	Description	Units	Default	Device Type
txtdatalevel	Flag for volume-averaged text output. 0 - no text files. 1 - text files. txtdataplot files will have the *.txt suffix, and the prefix will be the name of the device instance.	-	0	2D
bulkmaterial	Material of bulk material.	-	si	1D, 2D
mobmodel	mobility model.	-	carr	1D, 2D
type	P-type or N-type - this is only relevant if using the default dopings	-	PNP	1D, 2D
temp	Temperature	K	300.15	1D, 2D
nx	Number of mesh points, x-direction.	-	11	1D, 2D
l, w	Device length and width. These parameters mean the same thing for the 1D device.	-	1.0e-3	1D,2D
graded	Flag for graded junction vs. abrupt junction. (1=graded, 0=abrupt)	-	0	1D
wj	Junction width.	-	1.0e-4	1D
	Level 2 (2D) only			
ny	Number of mesh points, y-direction. Similar to nx (see above).	-	11	2D
meshfile	This is a required field for a 2D simulation. If the user specifies `meshfile = internal.mesh`, then **Xyce** will create a cartesian mesh. If the user specifies anything else (for example `meshfile = diode.msh`), **Xyce** will attempt to read in an external mesh file (in the example, named `diode.msh`) which is in the format of the SG Framework [18].	-	-	2D

Instance parameters	Description	Units	Default	Device Type
x0	This is the scaling factor for length. The code will do all of its scaling internally, so it is generally not neccessary to specify it manually. This is provided primarily for testing purposes.	-	max length of device	2D
Photogeneration terms (2D only)				
ph.a1	Photocurrent source magnitude.	$cm^{-3}s^{-1}$	0.0	2D
ph.tstart	Starting time	seconds	0.0	2D
ph.tstop	Ending time	seconds	1.0e+100	2D
photogen	This is an on/off flag. Setting to 1 turns on photogeneration terms	-	0	2D
ph.type	Uniform is the only available type.	-	UNIFORM	2D
ph.td	Delay time	seconds	0.0	2D
ph.tr	Rise time	seconds	0.0	2D
ph.tf	Fall time	seconds	0.0	2D
ph.pw	Pulse width	seconds	0.0	2D
ph.per	Period length	seconds	0.0	2D

Table 2.19: PDE Device Instance Parameters.

There is only one PDE device model parameter, the level.

Model parameters	Description	Units	Default
LEVEL	The level determines if this is a 1D or a 2D device. 1=1D, 2=2D.	-	1

Table 2.20. PDE Device Model Parameters.

Instance parameters	Description	Units	Default	Device Type
	All Levels			
function	functional form of doping region. Options are uniform, gaussian, and step.		1D,2D	uniform
type	Ntype of Ptype		1D,2D	ntype
nmax	Maximum value of impurity concentration.	cm^{-3}	1D,2D	1.0e15
nmin	Minimum value of impurity concentration.	cm^{-3}	1D,2D	1.0e15
xloc	Peak location	cm	1D,2D	0.0
xwidth	Distance from nmax to nmin, if applicable. This is only applicable for the function=gaussian case.		1D,2D	1.0e-3
flatx	This parameter determines if we're doing a half gaussian or a full gaussian. See table 2.22	-	1D,2D	0
	Level 2 (2D) only			
yloc	Same as xloc, but for the y-direction.	cm	2D	0.0
ywidth	Same as xwidth, but for the y-direction.	cm	2D	1.0e-3
flaty	Same as flatx, but for the y-direction.	-	2D	0

Table 2.21: PDE Device Doping Region Parameters. These correspond to the region instance parameter.

flatx or flaty value	Description	1D Cross Section
0	Gaussian on both sides of the peak (xloc) location.	
+1	Gaussian if x>xloc, flat (constant at the peak value) if x<xloc.	
-1	Gaussian if x<xloc, flat (constant at the peak value) if x>xloc.	

Table 2.22: Description of the flatx, flaty doping parameters

Electrode parameters	Description	Units	Default
	Level 2 (2D) only		
name	Electrode name	-	anode
bc	Carrier Density Boundary condition type (dirichlet or neumann)	-	dirichlet
start	Starting location	cm	0.0
end	Ending location	cm	0.0
side	Side specification (top, bottom, left or right)	-	top
material	Contact material		neutral
oxidebndryflag	Oxide layer boolean	-	false (0)
oxthick	Oxide thickness	cm	0.0
oxcharge	Oxide charge	C	0.0

Table 2.23: PDE Device Electrode Parameters.

Physical Models

This section contains information about physical models used in **Xyce** for the PDE devices. This includes various mobility models, expressions for calculating the effective mass for electrons and holes, an expression for intrinsic carrier concentration as a function of temperature, expressions which describe contacts to metal as well as contacts to metal-oxide-semiconductor devices.

Material Models and Parameters

This section describes some of the basic material properties that are available in **Xyce**. Described here are the models for effective mass, intrinsic carrier concentration, and the bandgap. This information is needed for the more complex models described in the mobility section (section 2.3) and the boundary condition section (section 2.3).

Effective Mass

Xyce includes functions which return the effective mass of electrons and holes for a number of semiconductor materials.

Electron Effective Mass

The electron effective mass is calculated as

$$m_{de} = (m_l^* m_t^{*2})^{1/3} \tag{2.1}$$

where m_l and m_t are the effective masses along the longitudinal and transverse directions of the ellipsoidal energy surface.

Hole Effective Mass

The hole effective mass is calculated as

$$m_{dh} = (m_{lh}^{*3/2} + m_{hh}^{*3/2})^{2/3} \tag{2.2}$$

where m_{lh} and m_{hh} are the "light" and "heavy" hole masses, respectively.

Intrinsic Carrier Concentration

The intrinsic carrier concentration in a semiconductor is obtained from the "np" product

$$np = n_i^2 = N_C N_V exp(-E_g/kT) \tag{2.3}$$

or

$$n_i = \sqrt{N_C N_V} e^{-E_g/2kT} \tag{2.4}$$

The expression used in **Xyce** to calculate the intrinsic carrier concentration comes from this and is given by

$$n_i = 4.9 \times 10^{15} (\frac{m_{de} m_{dh}}{m_0^2})^{3/4} M_c^{1/2} T^{3/2} e^{-E_g/2kT} \tag{2.5}$$

where M_c is the number of equivalent minima in the conduction band for the semiconductor, m_{de} is the density-of-state effective mass for electrons, m_{dh} is the density-of-state effective mass for holes, and m_0 is the free-electron mass.

Semiconductor	Symbol	$M_c^{1/2}$	n_i at room temperature
Silicon	si	$\sqrt{6.00}$	$1.25x10^{10}$
Germanium	ge	2.00	$2.5x10^{13}$
Galium Arsenide	gaas	1.00	$2.0x10^{6}$

Table 2.24: Intrinsic Carrier Concentration Parameters

Bandgap

The bandgap is a material and temperature-dependent quantity. The bandgap model for semiconductor materials, is based on Thurmond [19]. This model is given by:

$$E_g = E_{g0} - A * \left(\frac{T^{2.0}}{T + T_{off}} \right) \tag{2.6}$$

where E_g is the bandgap (eV) and T is the temperature (K). A, E_{g0}, and T_{off} are all material-dependent constants. Insulating materials, such as silicon dioxide, are assumed to have constant bandgaps, so their bandgaps are given by:

$$E_g = E_{g0} \tag{2.7}$$

where E_{g0} is a material-dependent constant. The values for the material-dependent constants used by equations 2.6 and 2.7 are given in Table 2.25.

Material	Symbol	E_{g0} (eV)	A	T_{off} (K)
Silicon	si	1.17	4.73e-4	636.0
Germanium	ge	0.7437	4.774e-4	235.0

Material	Symbol	E_{g0} (eV)	A	T_{off} (K)
Galium Arsenide	gaas	1.519	5.405e-4	204.0
Silicon Dioxide	sio2	9.00	NA	NA
Silicon Nitride	wdi	4.7	NA	NA
Sapphire	cu	4.7	NA	NA

Table 2.25: Bandgap constants

Mobility Models

A number of mobility models are included in **Xyce**. The analytic, arora, and carrier-carrier scattering models are considered to be low-field mobility models. The Lombardi surface mobility model is a transverse-field dependent model which also incorporates the mobility of the bulk silicon.

Analytic Mobility

This is a concentration- and temperature-dependent empirical mobility model, based on the work of Caughey and Thomas [20], which combines the effects of lattice scattering and ionized impurity scattering. The equation for the mobility of electrons is:

$$\mu_{0n} = \mu_{nmin} + \frac{\mu_{nmax}(\frac{T}{T_{ref}})^{nun} - \mu_{nmin}}{1 + (\frac{T}{T_{ref}})^{xin}(N_{total}/N_n^{ref})^{\alpha_n}} \tag{2.8}$$

and the equation for the mobility of holes is:

$$\mu_{0p} = \mu_{pmin} + \frac{\mu_{pmax}(\frac{T}{T_{ref}})^{nup} - \mu_{pmin}}{1 + (\frac{T}{T_{ref}})^{xip}(N_{total}/N_p^{ref})^{\alpha_p}} \tag{2.9}$$

where N_{total} is the local total impurity concentration (in $\#/cm^3$), T_{ref} is a reference temperature (300.15K), and T is the temperature (in degrees K). The parameters N_n^{ref} and N_p^{ref} are reference values for the doping concentration. The analytic mobility model can be selected by using the statement "mobmodel=analytic" in the netlist.

The parameters for the analytic mobility model are given in Table 3.

Parameter	Silicon	GaAs
μ_{nmin}	55.24	0.0
μ_{nmax}	1429.23	8500.0
N_n^{ref}	1.072e17	1.69e17
nun	-2.3	-1.0
xin	-3.8	0.0
α_n	0.73	0.436
μ_{pmin}	49.70	0.0
μ_{pmax}	479.37	400.0
N_p^{ref}	1.606e17	2.75e17
nup	-2.2	-2.1
xip	-3.7	0.0
α_p	0.70	0.395

Table 2.26: Analytic Mobility Parameters

Arora Mobility

This mobility model is also an analytic model which depends on impurity concentration and temperature. It comes from the work of Arora, *et al.* [21] and is based on both experimental data and the modified Brooks-Herring theory of mobility. The equation for the mobility of electrons is:

$$\mu_{0n} = \mu_{n1}(\frac{T}{T_{ref}})^{exn1} + \frac{\mu_{n2}(\frac{T}{T_{ref}})^{exn2}}{1 + (\frac{N_{total}}{Cn(\frac{T}{T_{ref}})^{exn3}})^{\alpha_n}} \tag{2.10}$$

and the equation for the mobility of holes is:

$$\mu_{0p} = \mu_{p1}(\frac{T}{T_{ref}})^{exp1} + \frac{\mu_{p2}(\frac{T}{T_{ref}})^{exp2}}{1 + (\frac{N_{total}}{Cp(\frac{T}{T_{ref}})^{exp3}})^{\alpha_p}} \tag{2.11}$$

where

$$\alpha_n = An(\frac{T}{T_{ref}})^{exn4} \tag{2.12}$$

and

$$\alpha_p = Ap(\frac{T}{T_{ref}})^{exp4} \tag{2.13}$$

The Arora mobility model can be selected by including the statement "mobmodel=arora" in the netlist. The parameters for the arora mobility model are given in Table 4.

Parameter	Silicon	GaAs
μ_{n1}	88.0	8.5e3
μ_{n2}	1252.0	0.0
Cn	1.26e17	1.26e17
An	0.88	0.0
exn1	-0.57	-0.57
exn2	-2.33	0.0
exn3	2.4	0.0
exn4	-0.146	0.0
μ_{p1}	54.3	4e2
μ_{p2}	407.0	0.0
Cp	2.35e17	2.35e17
Ap	0.88	0.0
exp1	-0.57	0.0
exp2	-2.23	0.0
exp3	2.4	0.0
exp4	-0.146	0.0

Table 2.27: Arora Mobility Parameters

Carrier-Carrier Scattering Mobility

This mobility model is based on the work of Dorkel and Leturq [22]. It incorporates carrier-carrier scattering effects, which are important when high concentrations of electrons and holes are present in the device. This model also takes lattice scattering and ionized impurity scattering into account. One important difference between the carrier-carrier scattering mobility model and the two previous mobility models (analytic and arora models) is that the carrier-carrier scattering mobility model depends upon the actual carrier concentrations in the device. This model is important for modeling breakdown as well as various radiation effects, which often result in very high carrier densities.

The expressions for the carrier-carrier model are as follows:

$$\mu_L = \mu_{L0}(\frac{T}{T_{ref}})^{-\alpha} \tag{2.14}$$

where μ_L is the lattice mobility, which has to do with scattering due to acoustic phonons.

$$\mu_I = \frac{AT^{3/2}}{N}[ln(1+\frac{BT^2}{N}) - \frac{BT^2}{N+BT^2}]^{-1} \tag{2.15}$$

where μ_I is the impurity mobility which is related to the interactions between the carriers and the ionized impurities.

$$\mu_{ccs} = \frac{2 \times 10^{17} T^{3/2}}{\sqrt{pn}}[ln(1+8.28 \times 10^8 T^2 (pn)^{-1/3})]^{-1} \tag{2.16}$$

where μ_{ccs} is the carrier-carrier scattering mobility, which is very important when both types of carriers are at high concentration.

$$X = \sqrt{\frac{6\mu_L(\mu_I + \mu_{ccs})}{\mu_I \mu_{ccs}}} \tag{2.17}$$

is an intermediate term and

$$\mu = \mu_L[\frac{1.025}{1+(X/1.68)^{1.43}} - 0.025] \tag{2.18}$$

is the carrier mobility. The carrier-carrier scattering mobility can be selected by including the statement "mobmobel=carr" in the netlist. The parameters for the carrier-carrier mobility model are given in Table 5.

Parameter	Carrier	Silicon	GaAs
Al	e^-	1430.0	8.50e3
Bl	e^-	-2.2	0.0
Ai	e^-	4.61e17	4.61e17
Bi	e^-	1.52e15	1.52e15
Al	h^+	495.0	4.0e2
Bl	h^+	-2.2	0.0
Ai	h^+	1.00e17	1.00e17
Bi	h^+	6.25e14	6.25e14

Table 2.28: Carrier-Carrier Mobility Parameters

Lombardi Surface Mobility Model

This mobility model combines expressions for mobility at the semiconductor-oxide interface and in bulk silicon. It is based on the work of Lombardi *et al.* [23]. The overall mobility is found using Mathiessen's rule:

$$\frac{1}{\mu} = \frac{1}{\mu_{ac}} + \frac{1}{\mu_b} + \frac{1}{\mu_{sr}} \tag{2.19}$$

where μ_{ac} is the carrier mobility due to scattering with surface acoustic phonons, μ_b is the carrier mobility in bulk silicon, and μ_{sr} is the carrier mobility limited by surface roughness scattering.

The Lombardi model is a more physics-based surface mobility model. It is a semi-empirical model for carrier mobility, and the expressions for the individual scattering mechanisms were extracted from experimental data taken in appropriate experimental conditions.

The expressions used in this model are given below:

$$\mu_{ac,n} = \frac{bn}{E_\perp} + \frac{cnN^{exn4}}{T(E_\perp)^{1/3}} \tag{2.20}$$

is the expression for electron mobility for acoustic phonon scattering,

$$\mu_{ac,p} = \frac{bp}{E_\perp} + \frac{cpN^{exp4}}{T(E_\perp)^{1/3}} \tag{2.21}$$

is the expression for hole mobility for acoustic phonon scattering,

$$\mu_{b,n} = \mu_{n0} + \frac{\mu_{max,n} - \mu_{n0}}{1 + (N/crn)^{exn1}} - \frac{\mu_{n1}}{1 + (csn/N)^{exn2}} \tag{2.22}$$

is the expression for bulk mobility for electrons, where

$$\mu_{max,n} = \mu_{n2}(\frac{T}{T_{ref}})^{-exn3} \tag{2.23}$$

and

$$\mu_{b,p} = \mu_{p0}exp(-pc/N) + \frac{\mu_{max,p}}{1 + (N/crp)^{exp1}} - \frac{\mu_{p1}}{1 + (csp/N)^{exp2}} \tag{2.24}$$

is the expression for bulk mobility for holes, where

$$\mu_{max,p} = \mu_{p2}(\frac{T}{T_{ref}})^{-exp3} \tag{2.25}$$

The expression for electrons for surface roughness scattering is

$$\mu_{sr,n} = (\frac{dn}{E_{\perp}^{exn8}}) \tag{2.26}$$

and the expression for holes for surface roughness scattering is

$$\mu_{sr,p} = (\frac{dp}{E_{\perp}^{exp8}}) \tag{2.27}$$

The parameters for the lombardi surface mobility model are given in Table 6.

Parameter	Silicon	GaAs
μ_{n0}	52.2	0.0
μ_{n1}	43.4	0.0
μ_{n2}	1417.0	1e6
crn	9.68e16	9.68e16
csn	3.43e20	0.0
bn	4.75e7	1e10
cn	1.74e5	0.0
dn	5.82e14	1e6
exn1	0.680	0.0
exn2	2.0	0.0
exn3	2.5	0.0
exn4	0.125	0.0
exn8	2.0	0.0
μ_{p0}	44.9	0.0
μ_{p1}	29.0	0.0
μ_{p2}	470.5	1.0
crp	2.23e17	2.23e17
csp	6.1e20	0.0
bp	9.93e6	1e10
cp	8.84e5	0.0
dp	2.05e14	1e6
exp1	0.719	0.0
exp2	2.0	0.0
exp3	2.2	0.0
exp4	0.0317	0.0
exp8	2.0	0.0
pc	9.23e16	0.0

Table 2.29: Lombardi Surface Mobility Parameters

Edge Mobilities

Mobility values are calculated along the edge connecting two nodes. In the case of the analytic, arora, and surface mobility models, the edge mobilities are calculated by taking the average of the mobilities at the two nodes. Then, the mobility along the edge connecting nodes 1 and 2 is:

$$\mu_{edge} = (\mu[1] + \mu[2])/2.0 \tag{2.28}$$

In the case of the carrier-carrier scattering mobility, the edge mobilities were calculated differently. The electron and hole concentrations were first calculated at the midpoint of the edge using a "product" average and then these values of "n" and "p" were used in the function to calculate the mobility at the midpoint of the edge. For example, if n[1] and n[2] are the electron concentrations at nodes 1 and 2, the electron concentration along the edge is given by:

$$n_{edge} = \sqrt{n[1] * n[2]} \tag{2.29}$$

Subsequently, the mobility at the midpoint of an edge is found by using the values of electron and hole concentration at the midpoint of the edge when calling the function which returns the mobility, calcMob().

$$\mu_{n,edge}^{carrier} = f(n_{edge}) \tag{2.30}$$

This method makes more sense, especially when the electron and hole concentrations vary by several orders of magnitude. Then it approximates taking the average of the logarithms.

Boundary Conditions for Electrode Contacts

This section describes various boundary conditions that need to be applied to the semiconductor boundary. **Xyce** is predominantly an analog circuit simulator, and the PDE-based device modeling that has been implemented in **Xyce** takes external circuit information as input. This input consists of voltages and currents which are applied as boundary conditions to the semiconductor domain.

The physical connection from the circuit to the device generally includes a variety of materials, including metals and oxides. Electrical differences between the semiconductor and the contact material can result in a potential barrier that must be included in the imposed voltage boundary condition.

There are three general types of contacts between the circuit and the PDE device which are currently handled by **Xyce**. The first is the "neutral" contact, in which it is simply assumed that the electrode material does not impose any addition potential barrier to that of the Fermi level differences in the semiconductor. The second is the Schottky contact, in which the electrode is a specified metal, and a potential barrier is imposed to account for the workfunction difference between the metal and the semiconductor. The last type of contact is the metal-oxide-semiconductor contact, in which the workfunction difference, and the voltage drop across the oxide must be accounted for.

Neutral Contacts

A neutral contact refers to the case in which the contact is made to the semiconductor itself, and barrier heights due to material differences are not considered. This is the simplest type of contact in **Xyce**, and problems which use this type of contact are generally easier to solve, compared with other types of contacts. In this case, the boundary is given by

$$V_{bc} = V_{ckt} + V_{bi} \tag{2.31}$$

where V_{ckt} is the potential applied by the circuit and V_{bi} is the "built-in" potential of the semiconductor. For a p-type substrate, the built-in potential is given by

$$V_{bi} = -\frac{kT}{q} ln(\frac{N_A}{n_i}) \tag{2.32}$$

and for an n-type substrate, the built-in potential is given by

$$V_{bi} = \frac{kT}{q} ln(\frac{N_D}{n_i}) \tag{2.33}$$

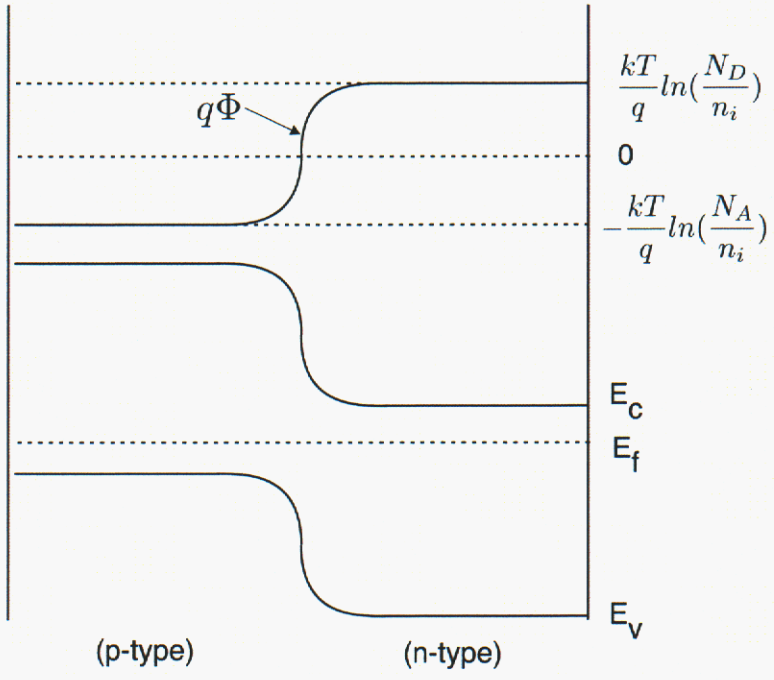

Figure 2.1. Neutral Contacts.

V_{bi} represents the extent of the energy band bending due to the doping of a device. While most of the dramatic changes will happen away from the contact, near junctions, it is still incorporated into the voltage boundary condition to maintain a flat potential near the contacts. Figure 2.1 shows the energy band variation accross a PN junction, and the corresponding electrostatic potential. This variation is due to the internal physics of the device, and needs to be there even in the event of zero applied voltage. This is partially enforced by the solution to Poisson's equation, and also by the application of equation 2.31.

Schottky Contacts

In the case of a metal-semiconductor contact, it is necessary to add the workfunction difference, Φ_{ms}, to the potential in the semiconductor [24]. Φ_m is a constant for a given metal, and Φ_s is a function of the doping in the semiconductor. The workfunction potential, Φ, when multiplied by q, is the difference between the Fermi level and vacuum in the material. In essence, the workfunction difference represents the distance between the Fermi level in the metal and the Fermi level in the semiconductor when considering the individual band structures.

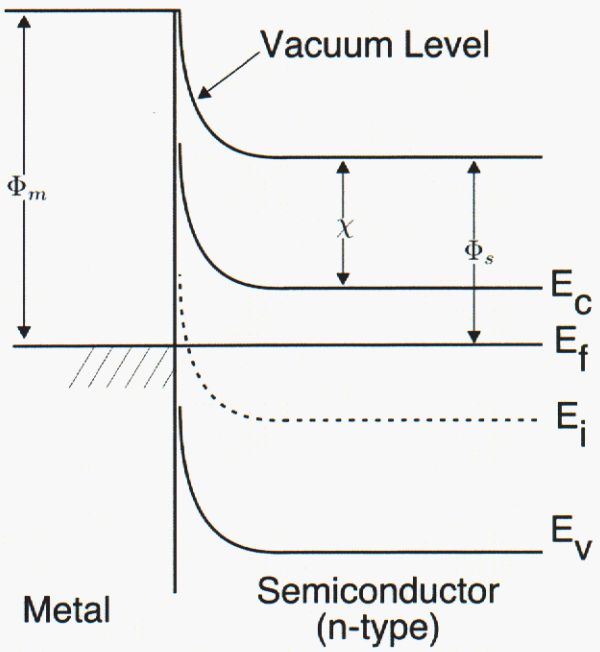

Figure 2.2. Schottky Contact, N-type.

In the case of an n-type semiconductor, the semiconductor workfunction can be represented as

$$\Phi_s = \chi + (E_C - E_{FS})/q \tag{2.34}$$

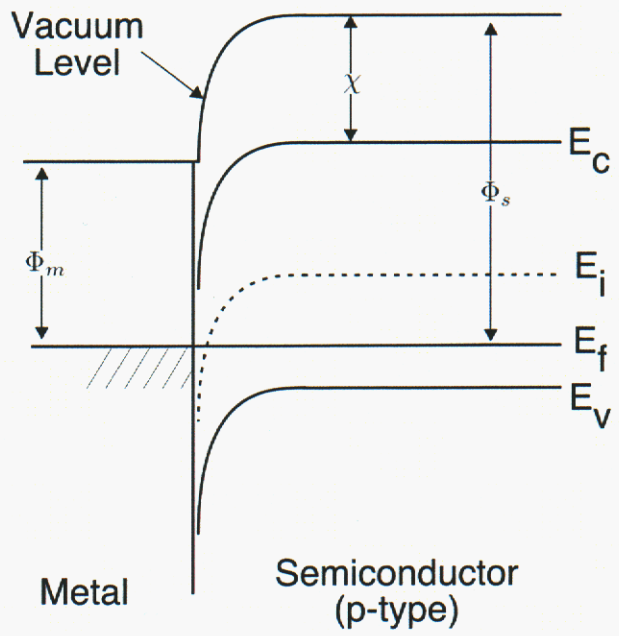

Figure 2.3. Schottky Contact, P-type.

where χ is the electron affinity in the semiconductor and qχ is the distance between the conduction band and vacuum in the semiconductor. E_C is the conduction band energy and E_{FS} is the Fermi level of the semiconductor. Rewriting this expression in terms of the doping concentration, it becomes

$$\Phi_s = \chi + E_g/2 - V_t ln(\frac{N_d}{n_i}) \tag{2.35}$$

In the case of a p-type semiconductor, the semiconductor workfunction can be represented as

$$\Phi_s = \chi + E_g/2 + (E_i - E_{FS})/q \tag{2.36}$$

where E_i is the intrinsic value of the Fermi level, and can be approximated as the halfway point between the conduction band (E_C) and the valance band (E_V). Rewriting this expression in terms of the doping concentration

$$\Phi_s = \chi + E_g/2 + V_t ln(\frac{N_a}{n_i}) \tag{2.37}$$

For the PDE devices in **Xyce**, for a node at a metal-semiconductor contact, the quantity $\Phi_m - \Phi_s$ is added to the potential at the node to account for the metal-semiconductor barrier. The current values of metal workfunctions used in **Xyce** are given in Table 2.30. The values for electron affinity are given in Table 2.31. The boundary condition for a metal electrode in **Xyce** is given by

$$V_{bc} = V_{ckt} + V_{bi} + \Phi_{ms} \tag{2.38}$$

where V_{ckt} is the potential applied by the circuit to the electrode and V_{bi} is the "built-in" potential of the semiconductor, a function of the semiconductor doping.

Metal	Symbol	Workfunction, Φ_m (Volts)
aluminum	al	4.10
p+-polysilicon	ppoly	5.25
n+-polysilicon	npoly	4.17
molybdenum	mo	4.53
tungsten	w	4.63
molybdenum disilicide	modi	4.80
tungsten disilicide	wdi	4.80
copper	cu	4.25
platinum	pt	5.30
gold	au	4.80

Table 2.30: Material workfunction values

Semiconductor	Symbol	Electron Affinity, χ (Volts)
Silicon	si	4.17
Germanium	ge	4.00
Galium Arsenide	gaas	4.07
Silicon Dioxide	sio2	0.97
Nitride	nitride	0.97
Sapphire	sapphire	0.97

Table 2.31: Electron affinities

Metal-Oxide-Semiconductor Contacts

To date in **Xyce**, only semiconductor material is included in the PDE solution domain. Metals and oxide materials are currently only included through boundary conditions. This is an adequate approach for a lot of problems. For some problems (such as modeling of low-dose radiation effects) modeling the oxide in more detail, as a PDE, will become neccessary. However, since oxides are usually very thin, compared with the semiconductor domain, meshing both materials as part of the same simulation is difficult. Therefore, incorporating the effects of a gate oxide as part of the gate boundary condition is a reasonable approach.

In the case of a contact to a metal-oxide-semiconductor structure, the separation of the Fermi energies in the metal and the semiconductor at equilibrium is due to two effects: the workfunction difference between the metal and the semiconductor, and the effective interface charge. These two effects cause the bands to bend at the surface in equilibrium. The flatband voltage is the sum of these two terms [24]:

$$V_{FB} = \Phi_{ms} - \frac{Q_i}{C_i} \tag{2.39}$$

where Φ_{ms} is the metal-semiconductor workfunction difference, Q_i is the value of interface charge (in C/cm^2), and C_i is the oxide capacitance per unit area, which is given by

$$C_i = \frac{\epsilon_{ox}\epsilon_0}{x_o} \tag{2.40}$$

The voltage V_{FB} is the amount of bias which, when applied to the gate, causes the electron energy bands to be flat. This is the potential that is added to a boundary node in **Xyce** to account for a metal-oxide-semiconductor barrier. The overall boundary condition for a contact to a metal-oxide-semiconductor structure is given by

$$V_{bc} = V_{ckt} + V_{bi} + \Phi_{ms} - Q_i/C_i \tag{2.41}$$

where V_{ckt} is the potential applied by the circuit and V_{bi} is the "built-in" potential of the semiconductor.

NMOS Device

The default NMOS device currently used in **Xyce** has a substrate doping concentration of $1.0 \times 10^{16}/cm^3$ and an oxide thickness of $1.0 \times 10^{-6}cm$. Since the ideal threshold voltage V_T is given by

$$V_T = 2\phi_F + \frac{\epsilon_s}{\epsilon_{ox}} x_o \sqrt{\frac{2qN_A\phi_F}{\epsilon_s\epsilon_0}} \tag{2.42}$$

V_T is equal to 0.892 V. for this device. Note that

$$\phi_F = \frac{1}{q}[E_i(bulk) - E_F] = \frac{kT}{q} ln(\frac{N_A}{n_i}) \tag{2.43}$$

for a p-type semiconductor substrate and

$$\phi_F = -\frac{kT}{q} ln(\frac{N_D}{n_i}) \tag{2.44}$$

for an n-type substrate.

This page is left intentionally blank

3. Command Line Arguments

Xyce supports a handful of command line arguments which must be given *before* the netlist filename. While most of these are intended for general use, others simply give access to new features that, while supported, are not enabled by default. These options are designated as *trial* options. The general usage is as follows:

```
runxyce [arguments] <netlist filename>
```

Table 3.1 gives a complete lists of command line options. In this table, the shaded rows indicate the trial options. *DEPRECATED* options are no longer supported and will be removed from future releases.

Argument	Description	Usage	Default
-h	Help option. Prints usage and exits.	`-h`	-
-v	Prints the version banner and exits.	`-v`	-
-delim	Set the output file field.	`-delim <TAB|COMMA|string>`	-
-o	Place the results into specified file.	`-o <file>`	-
-l	Place the log output into specified file.	`-l <file>`	-
-nox	Use the NOX nonlinear solver.	`-nox <on|off>`	on
-dva	Use faster direct vector access.	`-dva <on|off>`	on
-dma	Use faster direct matrix access.	`-dma <on|off>`	on
-ndl	Set the nonlinear solver debug level. (overrides netlist)	`-ndl <0..N>`	-
-tdl	Set the time integrator debug level. (overrides netlist)	`-tdl <0..N>`	-

Argument	Description	Usage	Default
-ddl	Set the device package debug level. (overrides netlist)	-ddl <0..N>	-
-sdl	Set the device sensitivity debug level. (overrides netlist)	-ddl <0..N>	-

Table 3.1: List of **Xyce** command line arguments.

4. Runtime Environment

There are two ways to start **Xyce**, using either the `runxyce` (or `runxyce.bat`) for serial **Xyce**, or `xmpirun` for parallel **Xyce**. These scripts set up the run time environment and call the **Xyce** executable. In addition to this, `xmpirun` serves as a wrapper to `mpirun` and `dmpirun`.

Environment variables are used to configure **Xyce**. Users may override the defaults by applying new values. Table 4.1 lists the environment variables and their usage.

Name	Purpose
`XYCE_METADATA_PATH`	Sets the path to the metadata configuration files.

Table 4.1. Environment variables used to control **Xyce**.

The version of `mpirun` called by `xmpirun` varies with each operating system. Table 4.2 shows the actual command executed on the each of the supported platforms. Execution in parallel is highly dependent on the MPI runtime environment. The table only lists the method used when the `mpirun` program is in the default configuration.

System	Actual MPI command line
HP/Compaq OSF	`dmpirun -np 2 Xyce_alpha-osf1_MPI anExampleNetlist.cir`
SGI / IRIX 6.5	`mpirun -np 2 Xyce_sgi10k_MPI anExampleNetlist.cir`
Intel X86 / RedHat Linux with MPICH	`mpirun.mpich -np 2 Xyce_linux_MPI anExampleNetlist.cir`
Intel X86 / RedHat Linux with LAM MPI	`mpirun -np 2 Xyce_linux_MPILAM anExampleNetlist.cir`
Intel X86 / FreeBSD with MPICH	`/usr/local/mpich/bin/mpirun -np 2 Xyce_freebsd_MPI anExampleNetlist.cir`
Intel X86 / FreeBSD with LAM MPI	`mpirun -np 2 Xyce_freebsd_MPILAM anExampleNetlist.cir`

Table 4.2. Typical command lines for parallel execution.

Running **Xyce** under MPICH

The MPICH implementation of MPI requires a file containing a list of machines on which to run. On RedHat Linux the file is installed in /usr/lib/mpich/share. On FreeBSD it is installed in /usr/local/mpich/share. This file must contain one line for each machine on which a process may be started. If you do not have write access this file, specify an alternate file with the `-machinefile <machinefilename>` option to `mpirun`.

MPICH executes parallel jobs by using the remote shell (rsh) or secure shell (ssh) to connect to the target machine. Therefore, you may be prompted for a password when starting a multiple processor job.

Running **Xyce** under LAM MPI

Unlike MPICH, LAM MPI requires a daemon process to run on each machine that will service parallel jobs. The process is started with `lamboot`. By default, `lamboot` will run a daemon (`lamd`) on the local machine, but it may be given a file name containing a list of machines for multiple-machine jobs. Consult the `bhost` man page for the format of the file.

`lamd` will run until it is explicitly halted. As long as `lamd` is running you may continue to run parallel jobs. Halt `lamd` using the `lamhalt` command.

Running **Xyce** under HP/Compaq's OSF MPI implementation

On HP/Compaq systems running the OSF operating system, the appropriate command to run parallel jobs is `dmpirun`. Its usage is otherwise similar to the MPICH and LAM `mpirun` commands.

This page is left intentionally blank

5. Setting Convergence Parameters for **Xyce**

Because the solution algorithms and methods within **Xyce** are different than those used by other circuit simulation tools (e.g., ChileSPICE), the overall convergence behavior is sometimes different as are the parameters which control this behavior.

With **Xyce** Release 1.1 several of the default transient parameters have been changed to improve overall performance and help make **Xyce** more competitive with other codes. While these defaults have been extensively tested to ensure acceptable accuracy, it is possible that some accuracy has been lost in order to provide improved performance. While this is likely negligible, accuracy can be improved by tightening the error tolerances and/or changing the time-integration order from their default values. As this is typically an issue for transient simulations, the following discussion applies to that context only. For a complete list of available solution control options, go to 2.1 in chapter 2.

Adjusting Transient Analysis Error Tolerances

As of Release 1.1, the default for **Xyce** is to use a Backward Euler (BDF-1) time integration method (also known as a 1-step Gear method) for performing transient analysis [25]. This is equivalent to using a Gear method 1 in ChileSPICE (the ChileSPICE default is to use the trapezoid rule). In general, this method provides acceptable accuracy and improved performance for circuits with highly-oscillatory behavior, which seem to be prevalent among those modeled at Sandia. However, the previous default, BDF-2, is a more stable and more accurate method than the backward Euler method and in circuits which have a "smoother" behavior (e.g., digital circuits), can provide a much faster and more accurate solution. Users who are modeling digital or other "smooth" circuits should select this method via the netlist with an options line:

```
.OPTIONS TIMEINT METHOD=2
```

Setting RELTOL and ABSTOL

In **Xyce**, there is currently RELTOL and ABSTOL settings for both the time integration package and the nonlinear solver package. Some general guidelines for settings parameters are [25]:

- Use the *same* RELTOL and ABSTOL values for both the TIMEINT and the NONLIN-TRAN .OPTIONS statements.
- For a conservative approach (i.e., safe), set RELTOL=1.0E-(m+1) where m is the desired number of significant digits of accuracy.
- Set ABSTOL to the smallest value at which the solution components (either voltage or current) are essentially insignificant.
- *Note* that the above suggests that ABSTOL < RELTOL.

The current defaults for these parameters are ABSTOL=1.0E-6 and RELTOL = 1.0E-2. For a complete list of the time integration parameters, see chapter 2.1.

Adjusting Nonlinear Solver Parameters (in transient mode)

In **Xyce**, the nonlinear solver options for transient analysis are set using the .OPTIONS NONLIN-TRAN line in a netlist. This subsection gives some guidelines for setting this parameters.

- For guidelines on setting RELTOL and ABSTOL, see above.
- RHSTOL – This is the maximum residual error for each nonlinear solution. **Xyce** uses this as a "safety" check on nonlinear convergence. Typically, 1.0E-2 (the default) works well.
- DELTAXTOL – This is the weighted update norm tolerance and is the primary check for nonlinear convergence. Since it is weighted (i.e., normalized using RELTOL and ABSTOL), a value of 1.0 would give it the same accuracy as the time integrator. For robustness, the default is 0.33 but sometimes a value of 0.1 may help prevent "time-step too small" errors. A value of 0.01 is considered quite small.
- MAXSTEP – This is the maximum number of Newton (nonlinear) steps for each nonlinear solve. In transient analysis, the default is 20 but can be increased to help prevent "time-step too small" errors. This is roughly equivalent to ITL4 in ChileSPICE.

6. Quick Reference for Orcad PSpice Users

This chapter describes many of the differences between **Xyce** and Orcad PSpice with an eye towards providing the ability for those familiar with using PSpice to begin using **Xyce** quickly. Currently, **Xyce** is still in very active state of development and so this section will change as capability is added to **Xyce** in the future. Also, note that there is also a quick reference chapter for ChileSPICE users 7, and many of the issues convered in that chapter are also applicable to PSpice.

GUI Support

Probably the most noticeable difference between **Xyce** and PSpice is the lack of a GUI in the current release version of **Xyce**. One, namely ChileCAD, is being developed and is expected to be available for the next release with some basic functionality. Thus, currently the only way to run **Xyce** is from the command line and using standard netlists.

Command Line Options

Command line arguments are supported in **Xyce** but they are different than those of PSPICE. For a complete reference, see chapter 3.

Device Support

In this release version of **Xyce**, not all the devices commonly found in circuit simulation tools are supported. For the complete list, please see the Analog Device Summary in Table 2.8.

Netlist Support

To the extent that specific devices or models are supported in **Xyce**, it supports most of the standard netlist inputs as may be found in standard SPICE. However, the `.OPTIONS` command has several additional features used to expose capabilities specific to **Xyce**. In particular, **Xyce** does not currently support the standard PSpice format `.OPTIONS` line in netlists. Instead, package specific `.OPTIONS` lines are supported according to the following

format: `.OPTIONS {PKG} <<TAG=>VALUE> ...` The **Xyce** packages which currently support `.OPTIONS` are:

Package	PKG keyword
Global:	GLOBAL
Device Model:	DEVICE
Time Integration:	TIMEINT
Nonlinear Solver:	NONLIN
Transient Nonlinear Solver:	NONLIN-TRAN
Continuation/Bifurcation Tracking:	LOCA
Linear Solver:	LINSOL
Parallel Distribution:	PARALLEL
Output:	OUTPUT
Restart:	RESTART

For a complete description of the supported options, see section 2.1.

Xyce does not support the ".PROBE" statement. Output of Probe format files is done using the ".PRINT" netlist statement. See chapter 2 for syntax.

Xyce does not support PSPICE style abbreviations in the ".PRINT" statement. For example, to print out the value of the voltage at node `A` in a transient simulation you must request `.PRINT TRAN V(A)`, not `.PRINT TRAN A`.

Converting PSpice ABM Models for Use in **Xyce**

As of the **Xyce** Version 2.0 release, **Xyce** is fully compatible with PSpice with respect to analog behavioral models. This includes the `E`, `F`, `G`, and `G` device types.

Usage of `.STEP` Analysis

The implementation of `.STEP` in **Xyce** is not yet fully compatible with that of PSpice. This will be corrected in subsequent releases of **Xyce**.

Sweep Type

PSpice supports four different sweep types: linear, octave, decade, and list. In **Xyce**, only the linear type is supported. Also, the **Xyce** parser will not understand a keyword specifying the sweep type. As such, this example will cause an error in **Xyce**:

Example:

```
.step LIN VCE 0V 10V .5V
```

However, it should work fine if you remove the LIN keyword.

Global .PARAM Sweeps

PSpice also supports sweeps over variables specified in .PARAM lines. This is not supported in **Xyce**. This block of text will not work in **Xyce**:

```
VAB 2 0 5
VAC 1 0 variable
.param variable=0
.step param variable 0 5 1
.dc VAB 4 5 1
```

An equivalent block of code that will work in **Xyce** is:

```
VAB 2 0 5
VAC 1 0 5
.step VAC 0 5 1
.dc VAB 4 5 1
```

Model Parameter Sweeps

PSpice requires extra keywords to apply a .STEP statement to a model parameter. **Xyce** handles model parameters differently, and is actually somewhat more flexible. Unfortunately, this means that the two specifications are not compatible.

A model parameter in PSpice would be handled like this:

```
R1 1 2 RMOD 1
.model RMOD RES(R=30)
.step RES RMOD(R) 30 50 5
```

The equivalent way to specify this in **Xyce** would be:

```
R1 1 2 RMOD 1
.model RMOD RES(R=30)
.step RMOD:R 30 50 5
```

Note that **Xyce** does not require the RES keyword on the .STEP line. In PSpice, this keyword is needed to specify what type of model is being used. **Xyce** actually has more flexibility

than PSpice in this regard - any model or instance variable can be set, on the .STEP line, using the same syntax.

Example: `.step D101:IS 1.0e-3 5.0e-3 1.0e-3`

In this example, D101 is the name of a model, or instance, and IS is the name of the parameter within that model or instance.

Other differences

Some other differences between **Xyce** and PSpice are described in table 6.1.

Issue	Comment
.VECTOR, .WATCH, and .PLOT output control analysis are not supported.	**Xyce** currently does not support these commands. If enough users request them, they may be supported in the future.
.AC, .FOUR, .FOUR, .NOISE and .TF analysis types are not supported.	**Xyce** fully supports .DC and .TRAN analysis. .OP and .SENS are partially supported. .AC is a planned feature. Other analyses, such as .NOISE may be supported in the future, if enough users request them.
.MC and .WCASE statistical analyses are not supported.	**Xyce** currently does not support these commands. If enough users request them, they may be supported in the future.
.DISTRIBUTION, which defines a user distribution for tolerances, is not supported.	**Xyce** does not support this command. This command goes along with .MC and .WCASE statistical analyses, which are also not supported.
.LOADBIAS, .SAVEBIAS, and .NODESET initial condition commands are not supported.	**Xyce** does not support these commands.
.ALIASES, .ENDALIASES, are not supported.	**Xyce** does not support these commands.
.STIMULUS is not supported.	**Xyce** does not support this command.
.TEXT is not supported.	**Xyce** does not support this command.
.SAVE does not work	**Xyce** does not support this. Use .PRINT instead.
.PROBE does not work	**Xyce** does not support this. Use the FORMAT=PROBE option of .PRINT instead. See section 2 for syntax.
.OP is incomplete	An .OP netlist will run in **Xyce**, but will not produce the extra output normally associated with the .OP statement.

Issue	Comment
.SENS is only supported in the development builds of **Xyce**	This is currently a research issue in **Xyce**, and will be available in later release versions of **Xyce**.
Pulsed source rise time of zero	A requested pulsed source rise/fall time of zero really is zero in **Xyce**. In other simulators, requesting a zero rise/fall time causes them to use the printing interval found on the tran line.
Mutual Inductor Model	Not the same as PSpice. This is a Sandia developed model.
.PRINT line shorthand	Output variables have to be specified as a V(node) or I(source). Listing the node alone will not work.
BSIM3 level	In **Xyce** the BSIM3 level=9. In PSpice the BSIM3 is level=8.
Node names vs. device names	Currently, circuit nodes and devices *MUST* have different names in **Xyce**. Some simulators can handle a device and a node with the same name, but **Xyce** cannot.
Interactive mode	**Xyce** does not have an interactive mode.
Time integrator default tolerances	**Xyce** has much tighter default solver tolerances than some other simulators (e.g., PSpice), and thus often takes smaller time steps. As a result, it will often take a greater number of total time steps for a given time interval. To have **Xyce** take time steps comparable to those of PSpice, set the RELTOL and ABSTOL time integrator options to larger values (e.g., RELTOL=1.0E-2, ABSTOL=1.0E-6).
PSpice-specific "operating point voltage sources"	These are not currently supported within **Xyce**. The capacitor and inductor, the only devices that currently support the "IC=" parameter, both insert the equivalent of operating-point voltage sources automatically when an initial condition is given.
.OPTIONS statements	**Xyce** does not support PSpice style .OPTION statements. In **Xyce**, the various packages all (potentially) have their own separate .OPTIONS line in the netlist. For a complete description, see section 2.1.
DTMAX	**Xyce** does support a maximum time step-size control on the .tran line, but we discourage its use. The time integration algorithms within **Xyce** use adaptive time-stepping methods that adjust the time-step size according to the activity in the analysis. If the simulator is not providing enough accuracy, the RELTOL and ABSTOL parameters should be decreased for both the time integration package (.OPTIONS TIMEINT) and the transient nonlinear solver package (.OPTIONS NONLIN-TRAN). We have found that in most cases specifying the same maximum timestep that PSpice requires for convergence actually slows Xyce down by preventing it from taking larger timesteps when the behavior warrants.
Nonlinear Dependent Source (B source) syntax	**Xyce** requires curly braces around all ABM expressions, where PSpice does not. See section 2.

Issue	Comment
.TRAN "UIC" keyword	PSpice requires the use of a keyword UIC on the .TRAN line in order to use initial conditions via IC keywords on instance lines. Doing so also tells PSpice not to perform an operating point calculation. In **Xyce**, UIC is ignored and produces a warning message. **Xyce** always uses initial conditions specified with IC keywords, and the case of inductors and capacitors automatically inserts a fictitious voltage source around the device that guarantees the correct potential drop across the device during the operating point. If the user desires that Xyce not perform an operating point calculation, but rather use an initial condition for a transient run of all zero voltages, then the user should specify NOOP instead.
.TRAN **Tstart** parameter	PSpice uses the third parameter of the .TRAN line as the time at which to begin outputting results. **Xyce** simply uses it as an offset to be applied to the time variable. To discard early parts of a run in **Xyce**, use a .OPTIONS OUTPUT line.
Temperature specification	Device temperatures in **Xyce** are specified through the .OPTIONS DEVICE line. PSpice allows a .TEMP line that is not recognized (and is ignored) by **Xyce**.

Table 6.1: Incompatibilities with PSpice.

7. Quick Reference for ChileSPICE Users

A large number of potential **Xyce** users have experience using Sandia's ChileSPICE circuit simulator, which is a shared-memory parallel code based on Berkely's SPICE version 3f5. Table 7.1 lists some of the differences between ChileSPICE and **Xyce**. Many of these are the same as the differences between PSpice and **Xyce**, which were listed in table 6.1.

Issue	Comment
`.SAVE` does not work	**Xyce** does not support this. Use .PRINT instead.
`.PROBE` does not work	**Xyce** does not support this. Use the FORMAT=PROBE option of .PRINT instead. See section 2 for syntax.
`.OP` is incomplete	An .OP netlist will run in **Xyce**, but will not produce the extra output normally associated with the .OP statement.
Pulsed source rise time of zero	A requested pulsed source rise/fall time of zero really is zero in **Xyce**. In other simulators, requesting a zero rise/fall time causes them to use the printing interval found on the tran line.
Mutual Inductor Model	Not the same as PSPICE. This is a Sandia developed model.
`.PRINT` line shorthand	Output variables have to be specified as a V(node) or I(source). Listing the node alone will not work.
BSIM3 level	In **Xyce** the BSIM3 level=9. In ChileSPICE the BSIM3 is level=8.
Node names vs. device names	Currently, circuit nodes and devices *MUST* have different names in **Xyce**. Some simulators can handle a device and a node with the same name, but **Xyce** cannot.
Interactive mode	**Xyce** does not have an interactive mode.
Time integrator default tolerances	**Xyce** has much tighter default solver tolerances than some other simulators (e.g., ChileSPICE), and thus often takes smaller time steps. As a result, it will often take a greater number of total time steps for a given time interval. To have **Xyce** take time steps comparable to those of ChileSPICE, set the `RELTOL` and `ABSTOL` time integrator options to larger values (e.g., `RELTOL=1.0E-2`, `ABSTOL=1.0E-6`).
ChileSPICE-specific "operating point voltage sources"	These are not currently supported within **Xyce**. The capacitor and inductor, the only devices that currently support the "IC=" parameter, both insert the equivalent of operating-point voltage sources automatically when an initial condition is given.

Issue	Comment
.OPTIONS statements	**Xyce** does not support ChileSPICE style .OPTION statements. In **Xyce**, the various packages all (potentially) have their own separate .OPTIONS line in the netlist. For a complete description, see section 2.1.
DTMAX	**Xyce** does support a maximum time step-size control on the .tran line, but we discourage its use. The time integration algorithms within **Xyce** use adaptive time-stepping methods that adjust the time-step size according to the activity in the analysis. If the simulator is not providing enough accuracy, the RELTOL and ABSTOL parameters should be decreased for both the time integration package (.OPTIONS TIMEINT) and the transient nonlinear solver package (.OPTIONS NONLIN-TRAN). We have found that in most cases specifying the same maximum timestep that ChileSPICE requires for convergence actually slows Xyce down by preventing it from taking larger timesteps when the behavior warrants.
Nonlinear Dependent Source (B source) syntax	**Xyce** requires curly braces around all ABM expressions, where ChileSPICE does not. See section 2.
.TRAN "UIC" keyword	SPICE 3F5 requires the use of a keyword UIC on the .TRAN line in order to use initial conditions via IC keywords on instance lines. Doing so also tells SPICE 3F5 not to perform an operating point calculation. In **Xyce**, UIC is ignored and produces a warning message. **Xyce** always uses initial conditions specified with IC keywords, and the case of inductors and capacitors automatically inserts a fictitious voltage source around the device that guarantees the correct potential drop across the device during the operating point. If the user desires that Xyce not perform an operating point calculation, but rather use an initial condition for a transient run of all zero voltages, then the user should specify NOOP instead.
.TRAN Tstart parameter	SPICE 3F5 uses the third parameter of the .TRAN line as the time at which to begin outputting results. **Xyce** simply uses it as an offset to be applied to the time variable. To discard early parts of a run in **Xyce**, use a .OPTIONS OUTPUT line.
Temperature specification	Device temperatures in **Xyce** are specified through the .OPTIONS DEVICE line. ChileSPICE allows a .TEMP line that is not recognized (and is ignored) by **Xyce**.
AREA parameter for radiation aware devices	The diode and BJT support a parameter on the instance line called AREA, and in both standard and radiation aware devices this is used to scale certain model parameters. In ChileSPICE this same parameter is used to scale the photocurrent. In **Xyce** it is used only to scale the model parameters, and a new model parameter DEVICEAREA is used to scale the photocurrent. This enables SPUDS data sets for real devices to be used unmodified except for the addition of the radiation-specific parameters. ChileSPICE requires that the scaled model parameters be adjusted to take into account the device area parameter.

Issue	Comment
.STEP syntax is not the same.	This issue is well covered in the PSpice quick reference. See section 6.

Table 7.1: Incompatibilities with ChileSPICE.

This page is left intentionally blank

Bibliography

[1] Scott A. Hutchinson, Eric R. Keiter, Robert J. Hoekstra, Lon J. Waters, Thomas V. Russo, Eric L. Rankin, Roger P. Pawlowski, and Steven D. Wix. Xyce parallel electronic simulator: User's guide, version 2.0. Technical Report SAND2004-xxxx, Sandia National Laboratories, Albuquerque, NM, June 2004.

[2] Orcad PSpice User's Guide. Technical report, Orcad, Inc., 1998.

[3] T. Banwell. Bipolar transistor circuit analysis using the lamber w-function. *IEEE Transactions on Circuits and Systems*, 47(11):1621–1633, November 2000.

[4] R. S. Tuminaro, M. A. Heroux, S. A. Hutchinson, and J. N. Shadid. Official Aztec User's Guide: Version 2.1. Technical report, Sandia National Laboratories, Albuquerque, NM, December 1999.

[5] T. A. Fjeldly, T. Ytterdal, and M. Shur. *Introduction to Device Modeling and Circuit Simulation*. Wiley InterScience, 1998.

[6] T. A. Fjeldly, Yanqing Deng, and M. Shur. Physics-based modeling of high intensity ionizing radiation effects in bipolar junction transistors. Technical report, 2000.

[7] A. S. Grove. *Physics and Technology of Semiconductor Devices*. John Wiley and Sons, Inc., 1967.

[8] P. Antognetti and G. Massobrio. *Semiconductor Device Modeling with SPICE*. McGraw-Hill, 1988.

[9] J.H. Huang, Z.H. Liu, M.C. Jeng, K. Hui, M. Chan, P.K. KO, and C. Hu. BSIM3 Manual. Technical report, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720.

[10] A. Vladimirescu and S. Lui. The Simulation of MOS Integrated Circuits using SPICE2. Technical Report Memorandum No. M80/7, February 1980.

[11] H. Shichman and D. A. Hodges. Modeling and simulation of insulated-gate field-effect transistor switching circuits. *IEEE Journal of Solid-State Circuits*, SC-3:285, September 1968.

[12] B. J. Sheu, D. L. Scharfetter, P.-K. Ko, and M.-C. Jeng. Bsim: Berkeley short-channel igfet model for mos transistors. *IEEE Journal of Solid-State Circuits*, SC-22:558–566, August 1987.

[13] J. R. Pierret. A MOS Parameter Extraction Program for the BSIM Model. Technical Report Memorandum No. M84/99 and M84/100, November 1984.

[14] Ping Yang, Berton Epler, and Pallab K. Chatterjee. An investigation of the charge conservation problem for mosfet circuit simulation. *IEEE Journal of Solid-State Circuits*, SC-18(1), February 1983.

[15] BSIM3v3.1 Manual. Technical report, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720.

[16] J. C. Bowers and H. A. Neinhaus. SPICE2 Computer Models for HEXFETs. Technical Report Application Note 954A, International Rectifier Corporation, Berkeley, CA 94720.

[17] M. Bucher, C. Lallement, C. Enz, F. Theodoloz, and F. Krummenacher. The EPFL/EKV MOSFET Model Equations for Simulation Technical Report: Model Version 2.6. Technical report, Electonics Laboratories, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, September 1997.

[18] K. M. Kramer and W. N. G. Hitchon. *Semiconductor Devices: A Simulation Approach*. Prentice-Hall, 1997.

[19] C. D. Thurmond. The standard thermodynamic function of the formation of electrons and holes in ge, si, gaas, and gap. *J. Electrochem. Soc.*, 122:1133, 1975.

[20] D.M. Caughey and R.E. Thomas. Carrier mobilities in silicon empirically related to doping and field. *Proc. IEEE*, 55:2192–2193, 1967.

[21] N.D. Arora, J.R. Hauser, and D.J. Roulston. Electron and hole mobilities in silicon as a function of concentration and temperature. *IEEE Transactions on Electron Devices*, ED-29:292–295, 1982.

[22] J.M. Dorkel and Ph. Leturq. Carrier mobilities in silicon semi-empirically related to temperature, doping, and injection level. *Solid-State Electronics*, 24(9):821–825, 1981.

[23] C. Lombardi, S. Manzini, A. Saporito, and M. Vanzi. A physically based mobility model for numerical simulation of nonplanar devices. *IEEE Transactions on Computer-Aided Design*, 7(11):1164–1170, November 1988.

[24] Ben G. Streetman and Sanjay Banerjee. *Solid State Electronic Devices*. Prentice-Hall, Upper Saddle River, New Jersey, 2000.

[25] K.E. Brenan, S.L Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM, 1996.

Index

DISTRIBUTION:

1 Steven P. Castillo
Klipsch School of Electrical and Computer Engineering
New Mexico State University
Box 3-o
Las Cruces, NM 88003

1 Kwong T. Ng
Klipsch School of Electrical and Computer Engineering
New Mexico State University
Box 3-o
Las Cruces, NM 88003

1 Nick Hitchon
Electrical and Computer Engineering
University of Wisconsin
1415 Engineering Drive
Madison, WI 53706

1 Mark Kushner
Department of Electrical and Computer Engineering
University of Illinois
1406 W. Green Street
Urbana, IL 61801

1 Ron Kielkowski
RCG Research, Inc
8605 Allisonville Rd, Suite 370
Indianapolis, In 46250

1 Mike Davis
Software Federation, Inc.
211 Highview Drive
Boulder, Co 80304

1 Wendland Beezhold
Idaho Accelerator Center
1500 Alvin Ricken Drive
Pocatello, Idaho 83201

1 Kartikeya Mayaram
Department of Electrical and Computer Engineering
Oregon State University
Corvallis, OR 97331-3211

1 Linda Petzold
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106-5070

1 Jaijeet Roychowdhury
4-174 EE/CSci Building
200 Union Street S.E.
University of Minnesota
Minneapolis, MN 55455

1 C.-J. Richard Shi
VLSI and Electronic Design Automation
210 EE/CSE Bldg.
Box 352500
University of Washington
Seattle, WA 98195

1 Homer F. Walker
WPI Mathematical Sciences
100 Institute Road
Worcester, MA 01609

1 Dan Yergeau
CISX 334
Via Ortega
Stanford, CA 94305-4075

1 Masha Sosonkina
319 Heller Hall
10 University Dr.
Duluth, MN 55812

1 Misha Elena Kilmer
113 Bromfield-Pearson Bldg.
Tufts University
Medford, MA 02155

1 Tim Davis
P.O. Box 116120
University of Florida
Gainesville, FL 32611-6120

1 Achim Basermann
C&C Research Laboratories,
NEC Europe Ltd.
Rathausallee 10
D-53757 Sankt Augustin
Germany

1 Philip A. Wilsey
Clifton Labs
7450 Montgomery Road
Suite 300
Cincinnati, Ohio 45236

1 Dale E. Martin
Clifton Labs
7450 Montgomery Road
Suite 300
Cincinnati, Ohio 45236

1 Lon Waters
CoMet Solutions, Inc.
11811 Menaul Blvd NE
Suite No. 1
Albuquerque, NM 87112

1 MS 0151
Tom Hunter, 09000

1 MS 0513
Al Romig, 01000

1 MS 0457
John Stichman, 02000

1 MS 0321
Bill Camp, 09200

1 MS 0841
Thomas C. Bickel, 09100

1 MS 1079
Marion Scott, 01700

1 MS 9003
Kenneth E. Washington, 08900

1 MS 0318
Paul Yarrington, 09230

1 MS 1071
Mike Knoll, 01730

1 MS 0310
Robert Leland, 09220

1 MS 0316
Sudip Dosanjh, 09233

1 MS 0525
Paul V. Plunkett, 01734

1 MS 0835
Steven N. Kempka, 09141

1 MS 0826
John D. Zepper, 09143

1 MS 0824
Jaime L. Moya, 09130

1 MS 0828
Martin Pilch, 09133

1 MS 0828
Anthony A. Giunta, 09133

1 MS 0139
Stephen E. Lott, 09905

1 MS 0310
Mark D. Rintoul, 09212

1 MS 1110
David Womble, 09214

1 MS 1111
Bruce Hendrickson, 09215

1 MS 1110
Neil Pundit, 09223

1 MS 1110
Doug Doerfler, 09224

1 MS 0822
Philip Heermann, 09227

1 MS 0819
Edward Boucheron, 09231

1 MS 0820
Patrick Chavez, 09232

1 MS 0316
John Aidun, 09235

1 MS 0316
Scott A. Hutchinson, 09233

10 MS 0316
Eric R. Keiter, 09233

1 MS 0316
Deborah Fixel, 09233

1 MS 0316
Robert J. Hoekstra, 09233

1 MS 0316
Joseph P. Castro, 09233

1 MS 0316
David R. Gardner, 09233

1 MS 0316
Gary Hennigan, 09233

1 MS 0316
Roger Pawlowski, 09233

1 MS 0316
Richard Schiek, 09233

1 MS 1111
John N. Shadid, 09233

1 MS 1111
Andrew Salinger, 09233

1 MS 0316
Paul Lin, 09233

1 MS 0316
Siriphone C. Kuthakun, 09233

1 MS 0807
David N. Shirley, 9328

1 MS 0807
Philip M. Campbell, 9328

1 MS 0847
Scott Mitchell, 09211

1 MS 0847
Mike Eldred, 09211

1 MS 0847
Tim Trucano, 09211

1 MS 0847
Bart van Bloemen Waanders, 09211

1 MS 0196
Elebeoba May, 09212

1 MS 1110
Todd Coffey, 09214

1 MS 1110
David Day, 09214

1 MS 1110
Mike Heroux, 09214

1 MS 1110
James Willenbring, 09214

1 MS 1111
Karen Devine, 09215

1 MS 0310
Jim Ang, 09220

1 MS 1109
Robert Benner, 09224

1 MS 0822
Pat Crossno, 09227

1 MS 0822
David Rogers, 09227

1 MS 0316
Harry Hjalmarson, 09235

1 MS 0525
Steven D. Wix, 01734

1 MS 0525
Thomas V. Russo, 01734

1 MS 0525
Regina Schells, 01734

1 MS 0525
Carolyn Bogdan, 01734

1 MS 0525
Mike Deveney, 01734

1 MS 0525
Raymond B. Heath, 01734

1 MS 0525
Ronald Sikorksi, 01734

1 MS 0525
Albert Nunez, 01734

1 MS 1081
Paul E. Dodd, 01762

1 MS 0660
Roger F. Billau, 09519

1 MS 0874
Robert Brocato, 01751

1 MS 1081
Charles E. Hembree, 01739

1 MS 0889
Neil R. Sorenson, 01832

1 MS 0311
Greg Lyons, 02616

1 MS 0311
Martin Stevenson, 02616

1 MS 0328
Fred Anderson, 02612

1 MS 0537
Perry Molley, 02331

1 MS 0537
Siviengxay Limary, 02331

1 MS 0537
John Dye, 02331

1 MS 0537
Barbara Wampler, 02331

1 MS 0537
Doug Weiss, 02333

1 MS 0537
Scott Holswade, 02333

1 MS 0537
George R. Laguna, 02333

1 MS 0405
Todd R. Jones, 12333

1 MS 0405
Thomas D. Brown, 12333

1 MS 0405
Donald C. Evans, 12333

1 MS 0405
Matthew T. Kerschen, 12333

1 MS 9101
Rex Eastin, 08232

1 MS 9101
Seung Choi, 08235

1 MS 9409
William P. Ballard, 08730

1 MS 9202
Kathryn R. Hughes, 08205

1 MS 9202
Rene L. Bierbaum, 08205

1 MS 9202
Kenneth D. Marx, 08205

1 MS 9202
Stephen L. Brandon, 08205

1 MS 9202
Jason Dimkoff, 08205

1 MS 9202
Brian E. Owens, 08205

1 MS 9401
Donna J. O'Connell, 08751

1 MS 9217
Stephen W. Thomas, 08950

1 MS 9217
Tamara G. Kolda, 08950

1 MS 9217
Kevin R. Long, 08950

1 MS 9915
Mitchel W. Sukalski, 08961

1 MS 1153
Larry D. Bacon, 15333

1 MS 1179
Leonard Lorence, 15341

1 MS 1179
David E. Beutler, 15341

1 MS 1179
Brian Franke, 15341

1 MS 0835
Randy Lorber, 09141

1 MS 1152
Mark L. Kiefer, 01642

1 MS 1137
Greg D. Valdez, 06224

1 MS 1137
Mark A. Gonzales, 06224

1 MS 1138
Rebecca Arnold, 06223

1 MS 1138
Charles Michael Williamson, 06223

1 MS 1138
Harvey C. Ogden, 06223

1 MS 9018
Central Technical Files, 8945-1

2 MS 0899
Technical Library, 9616