

VeriBest Analog Simulation Engine Reference Manual

DLA022230

VB99

VERIBEST[®]
I N C O R P O R A T E D

Warranties and Liabilities

All warranties given by VeriBest, Inc. (hereinafter collectively called VeriBest), are set forth in the Software License Agreement, and nothing stated in, or implied by, this document or its contents shall be considered or deemed a modification or amendment of such warranties.

The information and the software discussed in this document are subject to change without notice and should not be construed as commitments by VeriBest. VeriBest assumes no responsibility for any errors that may appear in this document.

The software discussed in this document is furnished under a license and may be used or copied only in accordance with the terms of this license.

Copyright 1994, 1995 INTERGRAPH CORPORATION Copyright 1996 - 1998 VeriBest, Inc.

INCLUDING THIS DOCUMENTATION AND ANY SOFTWARE AND ITS FILE FORMATS AND AUDIO-VISUAL DISPLAYS DESCRIBED HEREIN; ALL RIGHTS RESERVED; MAY ONLY BE USED PURSUANT TO THE APPLICABLE SOFTWARE LICENSE AGREEMENT; CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF VERIBEST, INC. AND/OR OTHER THIRD PARTIES WHICH IS PROTECTED BY COPYRIGHT, TRADE SECRET AND TRADEMARK LAW AND MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE WITHOUT PRIOR WRITTEN AUTHORIZATION.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of Commercial Computer Software -- Restricted Rights at 48 CFR 52.227-19, as applicable.

Unpublished -- rights reserved under the Copyright Laws of the United States.
VeriBest, Inc.; Boulder, CO 80301

Registered Licensee's Limited Right of Reproduction

Neither the manual nor audio-visual displays may be copied, reproduced, translated, or reduced to any other human or machine-readable form without the express written consent of VeriBest, except that all registered licensees may copy, for the licensee's use only, one copy per license held by the licensee of this manual.

Trademarks

VeriBest® is the registered trademark of VeriBest, Inc.

Intergraph® is a registered trademark of Intergraph Corporation.

ACEPlus and VeriBest Design Capture are trademarks of Intergraph Corporation.

Other brands and product names are trademarks of their respective owners.

Table of Contents

Warranties and Liabilities	2
 Chapter 1	
Introduction	1-1
Overview	1-2
Types of Analysis	1-3
Types of Output	1-3
User-Defined Controlled Sources	1-3
User-Defined Charge & Flux Sources	1-3
Usage Notes	1-3
Documentation Conventions	1-3
Simulation Requirements	1-4
Circuit Description	1-4
Unique VBASE Instructions	1-6
Important Program Enhancements	1-6
Legal Names	1-7
VBASE Device Quick Reference	1-8
Legal Numbers	1-9
Functions	1-9
Statistical Spread	1-11
Formats	1-12
Examples	1-13
 Chapter 2	
Voltage and Current Sources	2-1
Arbitrary Sources	2-1
Formats	2-1
Examples	2-2
Dependent Sources	2-2
Voltage Controlled Voltage Source	2-3
Formats	2-3
Examples	2-4
Current Controlled Current Source	2-5
Formats	2-5
Examples	2-6
Voltage Controlled Current Source	2-7
Formats	2-7
Current Controlled Voltage Source	2-9

Formats	2-9
Examples.....	2-10
Independent Sources	2-11
Formats	2-11
Examples.....	2-11
Exponential Specification	2-12
Format	2-12
Example.....	2-12
Full-Wave Rectified Sinusoidal.....	2-13
Format	2-13
Examples.....	2-13
Half-Wave Rectified Sinusoidal	2-14
Format	2-14
Example.....	2-14
Pulse Specification	2-15
Format	2-15
Examples.....	2-15
Piecewise-Linear Specification.....	2-16
Formats	2-16
Examples.....	2-16
Single-Frequency FM Specification.....	2-17
Format	2-17
Example.....	2-17
Sinusoidal Specification	2-18
Format	2-18
Examples.....	2-18

Chapter 3

Passive Elements.....	3-1
Parameter Formats.....	3-1
General .MODEL Specification	3-1
Format	3-1
Example.....	3-2
Transformer Cores	3-2
Format	3-2
Examples.....	3-3
Transformer Core Models	3-3
Format	3-3
Transformer Model Parameters	3-4
Capacitors	3-4
Formats	3-4
Examples.....	3-5
Semiconductor Capacitor Models	3-6
Format	3-6
Capacitor Model Parameters.....	3-6

Example.....	3-6
Polynomial and User-Defined Capacitors	3-7
Format	3-7
Example.....	3-7
Mutual Inductors	3-8
Format	3-8
Example.....	3-8
Inductors	3-8
Formats	3-8
Examples.....	3-9
Semiconductor Inductor Models.....	3-9
Format	3-9
Examples.....	3-10
Polynomial and User-Defined Inductors.....	3-10
Format	3-10
Example.....	3-10
Resistors	3-11
Formats	3-11
Examples.....	3-12
Semiconductor Resistor Models.....	3-12
Format	3-12
Resistor Model Parameters.....	3-13
Switches	3-13
Formats	3-13
Examples.....	3-14
Switch Models	3-14
Formats	3-14
Example.....	3-14
Switch Model Parameters	3-14
Transmission Lines	3-15
Formats	3-15
Example.....	3-15
Uniform/Distributed RC Lines	3-16
Format	3-16
Example.....	3-16
Distributed RC Models	3-16
Format	3-16
Example.....	3-17
Distributed RC Model Parameters.....	3-17
Windings	3-17
Format	3-17
Example.....	3-18

Chapter 4

Semiconductor Devices	4-1
Formats	4-1
General .MODEL Specification	4-2
Format	4-2
Example.....	4-2
BJT Devices	4-3
Formats	4-3
Examples.....	4-3
BJT Models	4-4
Formats	4-4
Example.....	4-4
BJT Model Parameters.....	4-5
Usage Notes.....	4-6
Diode Devices	4-7
Formats	4-7
Examples.....	4-7
Diode Models	4-8
Format	4-8
Example.....	4-8
Diode Model Parameters.....	4-9
JFET Devices	4-10
Formats	4-10
Example.....	4-10
JFET Models	4-11
Formats	4-11
Example.....	4-11
JFET Model Parameters	4-12
MESFET Devices	4-13
Formats	4-13
Example.....	4-13
MESFET Models	4-14
Formats	4-14
Example.....	4-14
MESFET Model Parameters	4-15
Level 1	4-18
Level 2	4-19
Level 3	4-19
MOSFET Devices	4-20
Formats	4-20
Examples.....	4-21
Usage Notes.....	4-21
MOSFET Models.....	4-22
Formats	4-22
Example.....	4-23
MOSFET Model Parameters	4-23

Setup	4-23
Geometry	4-23
Stress Analysis	4-24
Overlap Capacitances	4-24
Junction Capacitance (bottom)	4-24
Junction Capacitance (Sidewall)	4-25
Leakage Current (Bottom)	4-25
Leakage Current (Sidewall)	4-25
Parasitic Resistances	4-25
Channel Capacitance	
QOPT=4 Parameters	4-25
Threshold Voltage	4-27
Noise	4-27
Mobility	4-27
Level 1	4-27
Level 2	4-28
Level 3	4-28
Level 4	4-28
Level 5	4-29
Level 6	4-31
MOB=1 Parameters	4-32
MOB=2 Parameters	4-32
MOB=3 Parameters	4-32
MOB=4 and MOB=5 Parameters	4-32
CLM=1 Parameters	4-32
CLM=2 Parameters	4-33
CLM=3 Parameters	4-33
CLM=4 Parameters	4-33
Level 8	4-33
Level 10	4-34
Level 11	4-38
Level 20	4-39
Usage Notes	4-39
MOSFET Parasitics	4-40
Sidewall Junction Capacitance (CJO _{SW})	4-40
Bottom Junction Capacitance (CJO)	4-40
Bottom Diode Leakage Current (ISD _{DIODE})	4-41
Sidewall Diode Leakage Current (ISSW _{DIODE})	4-41
Parasitic Resistances (RPAR)	4-41

Chapter 5

Digital Devices	5-1
General .MODEL Specification	5-2
Format	5-2
Example	5-2

Flip-Flop Devices	5-3
Formats	5-3
Example.....	5-4
Flip-Flop Models.....	5-4
Format	5-4
Example.....	5-4
Gate Devices	5-5
Format	5-5
Example.....	5-6
Gate Models	5-6
Format	5-6
Example.....	5-6
Digital Model Parameters	5-7
Input Voltage Transition Type.....	5-7
Input Voltage Threshold	5-7
Propagation Delay	5-7
Output Voltage Transition Time.....	5-7
Output Voltage Level	5-7
Clock Mode.....	5-8

Chapter 6

Subcircuits.....	6-1
Subcircuit Definitions.....	6-2
Formats	6-2
Examples.....	6-2
Usage Notes.....	6-3
Subcircuit Expansions	6-4
Formats	6-4
Example.....	6-4
Subcircuit Functions	6-5
Formats	6-5
Examples.....	6-6

Chapter 7

Instructions.....	7-1
VBASE Instructions	7-1
.AC	7-2
Formats	7-2
Examples.....	7-2
.ACQUIRE	7-3
Format	7-3
.ALTER.....	7-4
Format	7-4
Examples.....	7-4

Usage Notes.....	7-5
.DC	7-6
Formats	7-6
Examples.....	7-7
.DELETE	7-8
Formats	7-8
Examples.....	7-8
.DISTO	7-9
Format	7-9
Examples.....	7-9
.DISTR.....	7-10
Format	7-10
.DISTR Equations and Parameters	7-10
Uniform.....	7-10
Gaussian (Gauss).....	7-10
Double Exponential (Dexpon).....	7-11
Gamma.....	7-11
Log-normal (Lognor).....	7-12
Weibull (Weibul)	7-12
Bimodal (Bimod)	7-13
.DUMP	7-13
Format	7-13
Examples.....	7-14
Usage Notes.....	7-15
.END	7-15
Format	7-15
.ENDDEL.....	7-15
Format	7-15
Example.....	7-15
.FOUR	7-15
Format	7-15
Examples.....	7-16
.GLOBAL.....	7-16
Format	7-16
Example.....	7-16
.IC.....	7-16
Formats	7-16
Examples.....	7-17
Usage Notes.....	7-17
.INCLUDE.....	7-17
Formats	7-17
Examples.....	7-17
.LIB	7-18
.LIB Library File	7-18
Format	7-18
Example.....	7-18
.LIB Calls	7-18

Format	7-19
Example.....	7-19
.MEASURE.....	7-19
Format	7-19
Example.....	7-20
.MODEL.....	7-20
Format	7-20
Example.....	7-21
.MONTE	7-21
Format	7-21
Examples.....	7-22
.NODESET	7-22
Formats	7-22
Examples.....	7-23
.NOISE	7-23
Format	7-23
Examples.....	7-23
.OP	7-24
Formats	7-24
Examples.....	7-24
.OPTIONS	7-24
Format	7-24
Algorithm Simulation Options	7-25
Tolerance Simulation Options	7-27
Stress Simulation Options	7-28
Statistical Simulation Options	7-28
Input Options	7-29
Output Options	7-30
Example.....	7-30
Usage Notes.....	7-30
.PARAM.....	7-31
Format	7-31
Examples.....	7-31
.PLOT	7-32
Format	7-32
Usage Notes.....	7-33
Examples.....	7-33
.PRINT.....	7-33
Format	7-33
Examples.....	7-35
.RESTART.....	7-35
Format	7-35
Example.....	7-35
.SENS.....	7-35
Format	7-35
Examples.....	7-36
.SEQUEL.....	7-36

Format	7-36
Example.....	7-37
.SUBCKT.....	7-37
Formats	7-37
Example.....	7-38
Usage Notes.....	7-38
.TEMP	7-38
Format	7-38
Examples.....	7-39
.TF	7-39
Format	7-39
Examples.....	7-39
.TRAN.....	7-39
Formats	7-39
Examples.....	7-41
Usage Notes.....	7-41
.VARY.....	7-42
Formats	7-42
Examples.....	7-43
.WIDTH.....	7-43
Format	7-44
Examples.....	7-44

Appendix A

Circuit Examples	A-1
Example 1 - Differential Pair	A-2
Explanations	A-2
Example 2 - MOSFET Device	A-3
Explanations	A-3
Example 3 - RTL Inverter	A-4
Explanations	A-4
Example 4 - Four-bit Adder	A-5
Explanations	A-7

Appendix B

Model Equations.....	B-1
BJT Model.....	B-2
DC Current	B-2
Charge Storage	B-4
Temperature Dependence.....	B-5
Diode Model	B-8
DC Current	B-8
Diode Model (IBV and BV Model Parameters).....	B-9
Diode Charge	B-9

Temperature Dependence.....	B-10
JFET Model	B-13
DC Current	B-13
Temperature Dependence.....	B-13
MESFET Model	B-15
Level 1	B-15
DC Current	B-15
Charge.....	B-15
Level 2.....	B-17
DC Current	B-17
Charge.....	B-17
Level 3.....	B-19
DC Current	B-19
Charge.....	B-20
QOPT = 4	B-22
Charge.....	B-22
Temperature Dependence.....	B-23
MOSFET Model	B-24
All Levels	B-24
Bottom Junction Capacitance	B-24
Threshold Voltage	B-24
Device Dimensions.....	B-25
Mobility	B-25
Depletion Layer Width	B-25
Bias Quantities	B-26
Noise Equations	B-26
Level 1	B-27
Threshold Voltage	B-27
Drain Current	B-27
Level 2.....	B-28
Threshold Voltage	B-28
Mobility Reduction	B-29
Saturation Voltage	B-29
Channel Length Modulation.....	B-31
Drain Current.....	B-32
Level 3.....	B-33
Threshold Voltage	B-34
Mobility Reduction	B-35
Saturation Voltage	B-35
Drain Current	B-35
Channel Length Modulation.....	B-35
Subthreshold Conduction	B-36
Level 4.....	B-37
Threshold Voltage	B-37
Mobility Reduction	B-37
Effective Beta	B-38
Saturation Voltage	B-39

Drain Current	B-39
Subthreshold Conduction	B-40
Geometry Dependence	B-40
Level 5	B-41
Geometry Dependence	B-41
Threshold Voltage	B-41
Mobility Reduction	B-41
Drain-Induced Barrier Lowering.....	B-41
Drain Saturation Voltage	B-42
Impact Ionization	B-42
Drain Current	B-42
Effective Beta	B-43
Saturation Region.....	B-43
Weak Inversion / Subthreshold.....	B-44
Thermal Voltage	B-44
Subthreshold Swing.....	B-44
Voltage Offset.....	B-44
Transition Region { $V_{GLOW} < (v_{gs} - v_{th}) < V_{GHIGH}$ }	B-44
Level 6	B-45
Threshold Voltage	B-45
Weak Inversion.....	B-46
Saturation Voltage	B-47
Alternate Saturation Voltage.....	B-48
Mobility Reduction	B-48
Channel Length Modulation.....	B-50
Drain Current	B-51
Level 8	B-52
Threshold Voltage	B-52
Mobility Reduction	B-53
Saturation Voltage	B-53
Channel Length Modulation.....	B-53
Drain Current	B-54
Level 10	B-55
Threshold Voltage	B-55
Effective	B-58
Mobility	B-58
Drain Saturation Voltage	B-59
Effective V_{ds}	B-60
Drain Current Expression	B-61
Substrate Current	B-62
Drain/Source Resistance.....	B-62
Effective Channel Length and Width	B-63
Temperature Effects	B-64
Level 11	B-65
Threshold Voltage	B-65
Mobility Reduction	B-65
Saturation Voltage	B-65

Drain Current	B-65
Subthreshold Conduction	B-66
Geometry Dependence	B-67
Temperature Dependence.....	B-67
Level 4	B-71
Level 11	B-72
TLEV = 1	B-72
Level 20.....	B-73
Static Intrinsic Model Equations	B-73
Quasi-Static Model Equations	B-79
EKV Noise Model	B-81
Non-Quasi-Static (NQS) model equations.....	B-82
Yang-Chatterjee Charge Model.....	B-83
Accumulation Region ($v_{gs} \leq v_{fb} + v_{bs}$)	B-83
Subthreshold Region ($v_{fb} + v_{bs} < v_{gs} \leq v_{th}$)	B-84
Saturation Region ($v_{th} < v_{gs} \leq a_x v_{ds} + v_{th}$)	B-84
Linear Region ($v_{gs} > a_x v_{ds} + v_{th}$)	B-85
Meyer Charge Model.....	B-87
Cut-off Region ($v_{gs} \leq v_{th}$)	B-87
On Region ($v_{gs} > v_{th}$)	B-88
Peak Region ($v_{gs} - v_{th} < 0.1$)	B-89
Transition Region ($v_{gs} - v_{th} \geq 0.1, v_{ds} < 0.1$)	B-90
Saturation Region ($v_{gs} - v_{th} \geq 0.1, v_{ds} \geq v_{dsat}$)	B-90
Linear Region ($v_{gs} - v_{th} \geq 0.1, v_{ds} < v_{dsat}$)	B-90
Ward-Dutton Charge Model	B-91
Accumulation Region ($v_g \leq 0$)	B-91
Cut-off Region ($v_{gs} \leq v_{th}$)	B-92
On Region ($v_{gs} > v_{th}$)	B-93
BSIM Charge Model.....	B-95
Accumulation Region ($v_{gs} \leq V_{FB} + v_{bs}$)	B-95
Subthreshold Region ($V_{FB} + v_{bs} < v_{gs} \leq v_{th}$)	B-95
Saturation Region ($v_{th} < v_{gs} \leq a_x v_{ds} + v_{th}$)	B-96
Linear Region ($v_{gs} > a_x v_{ds} + v_{th}$)	B-96
BSIM2 Charge Model	B-98
Accumulation Region ($v_{gs} < v_{bs} + V_{FB}$)	B-98
Subthreshold Region ($v_{bs} + V_{FB} < v_{gs} < v_{th} + V_{GLOW}$)	B-98
Saturation Region ($v_{ds} > v_{dsat}$)	B-98
Linear Region ($v_{ds} < v_{dsat}$)	B-99
BSIM3 Charge Model	B-100
Dimension Dependence	B-100
Overlap Capacitance	B-100
Intrinsic Charges.....	B-101
BSIM3 Non-Quasi-Static (NQS) Model	B-105
BSIM3 Noise Model.....	B-106
Flicker Noise	B-107
ASPEC Charge Model.....	B-110
Gate-to-Bulk Capacitance	B-110

Gate-to-Source Capacitance	B-110
Gate-to-Drain Capacitance	B-112
Distributed RC Line Model.....	B-113

Appendix C

DIABLO Language Structure	C-1
Calling a DIABLO Function.....	C-2
Overview: An Example	C-2
General Description.....	C-2
Controlled Source Functions	C-2
Capacitors and Inductors.....	C-3
Multidimensional Capacitors and Inductors.....	C-5
Advanced Features	C-5
Writing a DIABLO Function	C-6
Basic Framework.....	C-6
Example.....	C-7
Function Body	C-8
Syntax Rules	C-8
Variables and Numbers	C-8
Algebraic Expressions and Function Calls	C-11
Conditional Statements	C-12
Return Statements.....	C-13
Print Statements	C-14
Special Features	C-14
VBASE Features	C-15
System Reserved Variables (internally set by VBASE).....	C-15
System Reserved Variables (set by the user)	C-16
Reserved Prefixes	C-18
System Flags.....	C-19
Special Function Types	C-19
Multi-Defined Function Names	C-20
Summary	C-21
Convergence Problems	C-21

Appendix D

Error Messages.....	D-1
Format	D-1
.INCLUDE, .LIBRARY, and .ENDLIB Errors	D-2
.NAME Instruction Errors.....	D-2
.NODESET Error	D-2
Circuit Checker Errors	D-2
Command Error.....	D-3
DC Solution Errors	D-3
General Errors.....	D-3

Initial Conditions Error	D-4
Input Processor Errors	D-4
Input Source Errors	D-5
Matrix Error.....	D-5
Model Specification Errors	D-5
MOSFET Table Model Errors.....	D-5
Mutual Inductor Error.....	D-6
Sensitivity Analysis Errors	D-6
Subcircuit Errors.....	D-6
Tolerance Setting Error	D-6
Transfer Function Error	D-6
Transient Analysis Errors	D-6
Transmission Line Errors	D-6
User-Defined Element Errors	D-7
Index	1

Chapter 1

Introduction

The VeriBest Analog Simulation Engine (VBASE) is a high performance circuit simulator. VBASE provides excellent simulation convergence, accuracy, charge conservation, and speed. It is SPICE-compatible, but not SPICE-based. Instead, it employs a unique set of proprietary algorithms for analog verification.

The major features of VBASE include its ability to simulate large designs (thousands of transistors) with high performance in convergence, accuracy and speed. These benefits combined with new model equations (for example, improved conservation of charge), improved time-step control, and simulation stop-restart for steady-state analysis provide the fastest, most accurate simulation available.

VBASE applies its single-engine algorithm approach to analog and mixed-signal designs that contain medium- and small-scale logic integration. This approach allows seamless analysis of both the electronic and timing properties in designs. VBASE simulates analog properties in logic devices, which is particularly advantageous in detecting defects in high speed designs. Such problems could include rise/fall time, overshoot, etc.

This document is intended as a reference manual for the advanced user as a supplement to the *VeriBest Analog User's Guide*.

The supporting manuals are:

- **VeriBest Analog User's Guide** - the user guide for VeriBest Analog. It includes instructions, explanations, and examples for using this software to help you design and simulate circuits.
- **VeriBest Analog Quick Start** - This document describes the steps needed to create the simulation environment and simulate an analog circuit. It is not intended to teach you everything about VBA, but to give you a firm grasp of how the software operates.
- **VeriBest Analog Model Library Reference Manual** and the **VeriBest Analog Model List** - These reference manuals contain descriptions of the models and a complete list of model libraries distributed with the VBA application.

This chapter contains the following sections:

- Overview
 - Types of Analysis
 - Types of Output
 - User-Defined Controlled Sources
 - User-Defined Charge & Flux Sources
- Usage Notes
 - Documentation Conventions
 - Simulation Requirements
 - Circuit Description
 - Statistical Spread

Overview

VeriBest Analog Simulation Engine (VBASE) is a fast and accurate circuit simulation program developed by VeriBest, Inc. VBASE performs nonlinear DC, nonlinear transient, and linear AC analysis. Circuits may contain voltage and current sources, resistors, capacitors, inductors, diodes, BJTs, JFETs, MOSFETs, MESFETs, distributed RC lines, transmission lines, and user-defined controlled sources. VBASE is compatible with the input formats, output formats, and models of UC Berkeley's SPICE2G6. In addition, VBASE has enhancements which make it compatible with modified versions of SPICE2.

VBASE has built-in models for the semiconductor devices. You need only specify the pertinent model parameter values.

You can use the diode model to model either PN junction diodes or Schottky barrier diodes.

The NPN and PNP BJT models in VBASE are an adaptation of the Gummel-Poon integral charge control model, which is compatible with the SPICE2G6 BJT model.

The N-type and P-type JFET models are SPICE2G6 compatible and are derived from the Shichman-Hodges FET model.

There are various default N-channel and P-channel MOSFET models available, based on models used in the SPICE program. These MOSFET models have been enhanced to account for charge conservation.

The MOS1 model, derived from Shichman and Hodges, is described by a square-law I-V characteristic.

Both MOS2 (an analytical model) and MOS3 (a semi-empirical model) include second-order effects in short-channel devices such as channel length modulation, subthreshold conduction, scattering-limited velocity saturation, and nonlinear capacitances.

The MOS4 is an enhanced version of the short channel BSIM model (described by Sheu) and is a process characterization-based model.

The MOS5 BSIM2 model is a semi-empirical, deep-submicron MOSFET model (described by Min-Chie Jeng).

The MOS6 is an enhanced version of the ASPEC model.

The MOS8 is an enhanced but empirical version of MOS2 model.

The MOS10 BSIM3 is a physics-based, deep-submicron MOSFET model developed by the Device Group at UC Berkeley.

The MOS11, is an enhanced implementation of the short channel CSIM model described by Poon and Scharfetter.

The MOS20 EKV is an analytical MOSFET model developed by Enz, Krummenacher, and Vittoz of the Swiss Federal Institute of Technology for low-voltage and low-current applications.

There are various N-channel and P-channel MESFET models available. The MES1 model is derived from the RCA quadratic model, the MES2 model is derived from the RCA cubic model, and the MES3 model is derived from the Raytheon model.

VBASE also provides you with the capability of incorporating proprietary MOSFET, MESFET, and BJT models, MOSFET table models, and user-defined elements.

Types of Analysis

Nonlinear DC analysis provides the initial quiescent state of the network. VeriBest has developed proprietary techniques to assure excellent convergence properties. DC analysis can also be used to generate DC transfer curves. Nonlinear transient analysis simulates the circuit operation as a function of time over a user-specified time interval. Linear AC analysis computes the frequency response of the network using the linearized component values computed at the DC operating point.

Types of Output

The DC analysis output is tables or plots of node voltages and branch currents for DC transfer curve analysis. A table of node voltages and device operating point information is the output for the DC operating point analysis. The transient analysis output is either tables or plots of node voltages and branch currents versus time. The linear AC analysis output is tables or plots of outputs versus frequency.

User-Defined Controlled Sources

VBASE allows you to generate your own equations for controlled sources in a C-like language format. For more information, see the DIABLO language section of this manual.

User-Defined Charge & Flux Sources

VBASE allows you to generate your own equations for capacitor and inductor elements in a C-like language format. For more information, see the DIABLO language section of this manual.

Usage Notes

Documentation Conventions

To help you locate and interpret information easily, the *VeriBest Analog Simulation Reference Manual* uses consistent visual cues and a few standard keyword formats. The conventions are detailed below.

In Text, This	Represents
BOLD	Upper-case bold letters are used to denote the minimum required characters or strings of characters.
text	Lower-case characters are used to represent replacement character strings or numeric values. For example, Rxxxxxxx denotes a string beginning with the letter R (or r) followed by one to 27 alphanumeric characters.
<tstart>	Fields enclosed in angle brackets are optional and may be nested. For example, <tstart<tmax>> indicates that tstart must be specified if tmax is to be specified. If the example was <tstart> <tmax>, it would indicate that the optional values could be specified independently.
p ₁ <p ₂ <p ₃ ...>>>	Ellipses (...) are used to indicate that repeated or continued values may be specified. For example, p ₁ <p ₂ <p ₃ ...>>> indicates that additional optional values may be specified beyond p ₃ .
() , =	When punctuation is indicated in general format, it should be used as shown, unless otherwise stated.

Simulation Requirements

A circuit simulation program requires the following input information:

- The circuit component topology: a description of how sources, passive elements, and active semiconductor devices are connected.
- The component values including the source value-time relationships, the element values, the individual semiconductor device values, and the corresponding device modeling parameters.
- Simulation controls and descriptions specifying the type of simulation desired, the initial conditions for simulation, the desired outputs, and the controlling parameters for numerical computations.

The circuit topology and values of a network component are typically described in one data line of the program input file. However, more complex semiconductor devices usually require an additional *model* instruction to define all of the model parameters.

You can describe the circuit to be simulated in a hierarchical manner using the concept of *subcircuits*. In the subcircuit concept, blocks of network components are defined and later used repeatedly to describe larger, more complex circuits. Nesting can be used within subcircuits.

Circuit Description

The circuit to be analyzed by VBASE is described by a file consisting of:

- **Title Statement**

The first statement in an input file must be a title statement. The title statement typically consists of the name of a circuit and must fit on one line. Its contents are printed verbatim as the heading for each section of output. For example, the title statement has the following format:

```
POWER AMPLIFIER CIRCUIT
TEST OF CAM CELL
```

- **Comment Lines**

Comments must be preceded by an asterisk (*) or a semi colon (;). The asterisk should be used for full line comments and the semi colon for in line comments. The text following the asterisk or semi colon is interpreted as a comment. Comment lines cannot be continued using the continuation character (+).

Note: Inline comment characters need to have at least one space between them and the preceding text in order to be parsed correctly.

Comments are formatted in the following manner:

```
* <any comment>
; <any comment>
```

For example,

```
*Data following describes the buffer*
+ RF = 1k ; GAIN SHOULD BE 100
*****BUFFERCKT*****
```

- **Source Description Lines**

Define specifications for dependent and independent voltage and current sources. See *Chapter 2, Voltage and Current Sources*, for a detailed discussion of voltage and current sources supported by VBASE.

- **Passive Element Description Lines**

Define specifications for resistors, capacitors, inductors, and other passive elements. See *Chapter 3, Passive Elements*, for a detailed discussion of passive elements supported by VBASE.

- **Active Semiconductor Device Description Lines**

Define the models for the semiconductor devices typically requiring the definition of multiple parameters for accurate specification. See *Chapter 4, Semiconductor Devices*, for a detailed discussion of semiconductor devices and the model parameters supported by VBASE.

- **Instruction Lines**

Instruction lines are defined by a keyword that begins with a period (.). Instructions are used to specify the types of analysis, desired parameters for controlling simulation behavior, device model parameters, and output specifications. Instruction lines can occur in any order, but must follow these rules:

- The title and .END lines must be the first and last, respectively.
- Subcircuit definitions are contained within a header (.SUBCKT...) and an end (.ENDS).
- Instructions cannot be inside subcircuit definitions (except for the .MODEL, .IC and .NODESET instructions).
- .WIDTH instructions affect only subsequent lines within the file.

Blank or comment lines between continuation lines are ignored. Names or numerical values should not be split between two lines. For example,

```
.MODEL HMOS4 NMOS (LEVEL=3 VTO=1.25 KP=2.9E-5
+ GAMMA=.36 CBD=12FF CBS=13FF IS=1.2E-16
+ ETA=1.1 KAPPA=0.4)
```

See *Chapter 7, Instructions*, for a detailed discussion of the instruction lines supported by VBASE.

- **Subcircuit Definition**

Subcircuits provide a way to conveniently specify multiple identical circuit blocks. They are primarily used to simplify data specifications. They consist of two basic components:

- A subcircuit definition, in which the components and the topology of the subcircuit are defined, as well as the external node connections of the subcircuit block
- Subcircuit expansions (sometimes referred to as calls or instantiations) which define how a subcircuit block is to be used in the circuit. Multiple expansions may be made of any subcircuit definition.

See *Chapter 6, Subcircuits*, for a detailed discussion of subcircuit definition, data blocks and expansions supported by VBASE.

- **DIABLO Function Blocks**

These function blocks can be used to generate component behavior as a series of statements which can be executed to achieve a computational objective. See *Appendix C, DIABLO Language Structure*, for a detailed discussion of calling and writing DIABLO functions.

- **.END Statement**

The last statement in an input file must be a .END statement. The period is a required part of the statement. The .END statement has the following format:

```
.END
```

Within data lines of a circuit description, a free input format is used. Fields may be separated by one or more spaces, tabs, commas, and equal signs. Left or right parentheses may be mandated by the syntax. By default, only the first 80 characters of an input line are processed. The .WIDTH instruction can be used to change this value.

The entire circuit description is contained between the first line, which is the title, and the last line, which is the .END instruction. Data within the file can be in any order; however, each subcircuit block must be specified as a unit. Blank lines can be inserted to improve readability.

Unique VBASE Instructions

VBASE is a SPICE-compatible circuit simulator. However, VBASE contains numerous enhancements over SPICE2G6 algorithms, circuit elements, input specifications, analysis instructions, and analysis capabilities.

For each instruction there are a number of data specification formats. These formats are shown under the general header “Formats.”

Most instructions have two types of formats, those that are SPICE-compatible and those that are enhanced VBASE. The SPICE-compatible formats are shown first.

Some instructions do not exist in SPICE; therefore, all of the formats shown for these instructions are VBASE specific. The following instructions are those that are VBASE specific:

- .ACQUIRE
- .DELETE (used in accord with .ALTER)
- .DISTR
- .DUMP
- .ENDDEL
- .ENDFUNC (see the DIABLO section of this manual)
- .EOM (an alias for .ENDS)
- .FUNC (see the DIABLO section of this manual)
- .GLOBAL
- .INCLUDE
- .LIB
- .MACRO (an alias for .SUBCKT)
- .MONTE
- .PARAM
- .RESTART
- .SEQUEL
- .VARY

Important Program Enhancements

VBASE has the following enhancements:

- You can use parameterized values, but only as shown in bold in the following example:

```
.SUBCKT foo 10 40 STNDW=5U
m12 10 20 30 40 depl W=STNDW L=3U
.ENDS
```

```
.IC v(50)=highV
v10 20 0 pwl 0n 0.0 5n highV 15n highV
.param highV=5v
```

- You can use node names in addition to node numbers, as shown in bold in the following example of a voltage divider:

```
vin 2 gnd dc 5v
r1 2 divNode 1K
r2 divNode gnd 1k
```

- You can use probing for branch currents on specific devices and subcircuit call terminals, as shown in the following example:

```
.PRINT TRAN I3(M5) I5(X2)
```

- You can also use probing for devices and nodes that are inside subcircuits, as shown in the following example:

```
.PLOT DC i(xbuff.vc3) v(x1.200)
.IC v(x1.200)=5v
```

Legal Names

Names are used in the data file to name components, nodes (optional), model types, subcircuits, and parameters. There is no limit on the number of characters in a name, but only the first 28 characters will be recognized. To avoid confusion, keep names in the circuit description globally unique from user-defined parameter names.

Case distinctions are not made for names; upper-case characters are accepted, but they are converted to lower-case. Names use the same character set as SPICE2G6. As with SPICE, the following characters are not acceptable in names: control characters, space, equal sign, comma, and parentheses. Names must begin with a letter (A through Z). See the VBASE Device Quick Reference chart on the following page.

Component names begin with a special first letter, where the letter identifies the component type. For example, a resistor name must begin with the letter R. Hence, R, R1, RSE, ROUT, and R3AC2Z& are all valid resistor names. These names can be of arbitrary length, but the first 28 characters must be unique.

Node names are names or positive integers that don't need to be numbered sequentially. However, the global numerical name for the datum (ground) node is zero (0), even within subcircuits. The names 'gnd' and 'ground' are equivalent to node 0. Leading zeros are discarded. Thus, 0035 and 35 denote the same node. If a node has a name, it must begin with a letter. Additionally, if you begin a node name with a digit, VBASE truncates the name at the first alpha character. For example, 123A and 123 are interpreted by VBASE as identical node names.

Model names can start with any character with the exception of the BJT, MESFET and MOSFET model types. These model types must begin with an alpha character.

Subcircuit names can be of any length, but the first 28 characters must be unique. These names can be the same as component or node names. Node, component, and model names within subcircuits are local to the subcircuit definition.

Names or nodes within subcircuits are specified by using the subcircuit name as a prefix, separated by periods. Nested subcircuit names are specified in the order of the highest to the lowest in the hierarchy. For example, *Xaa.Xbb.Xcc.name* references *name* within subcircuit expansion *Xcc* within expansion *Xbb* within expansion *Xaa*.

sion *Xbb* which is within expansion *Xaa*. Although the name for each subcircuit cannot exceed 28 characters, the complete path name length for the nested subcircuit is unlimited.

In the ASCII output reports, all node names are displayed with their full path information in the hierarchical structure in ascending order. Each character is compared based on the ASCII representation on the system. For example, V(110) is before V(2) because 1 is before 2.

VBASE Device Quick Reference

First Character	Device Type	Model Type
A	Digital	various
B	Arbitrary Source	-
C	Capacitor	C
D	Diode	D
E	Voltage-Controlled Voltage Source	-
F	Current-Controlled Current Source	-
G	Voltage-Controlled Current Source	-
H	Current-Controlled Voltage Source	-
I	Current Source	-
J	JFET	NJF, PJF
K	Mutual Inductor	-
L	Inductor	L
M	MOSFET	NMOS, PMOS
N	Not Used	-
O	Not Used	-
P	Core	TFM
Q	BJT	NPN, PNP
R	Resistor	R
S	Voltage Controlled Switch	SW
T	Transmission Line	-
U	Uniform Distributed RC Lines	DRC, URC
V	Voltage Source	-
W	Current Controlled Switch	CSW
X	Subcircuit Call	-
Y	Winding	-
Z	MESFET	NMES, PMES

Legal Numbers

Numbers may be integers (for example, 12 and -44) or floating point values (for example, 3.14159). Optionally, numbers may be followed by an integer exponent (for example, 1E-14, 2.65E3) or one of the following scale factors:

Scale Factor	Prefix	Multiplying Value
T	(tera-)	1E12
G	(giga-)	1E9
MEG	(mega-)	1E6
K, k	(kilo-)	1E3
M, m	(milli-)	1E-3
U, u	(micro-)	1E-6
MIL	[mils to meters]	25.4E-6
N, n	(nano-)	1E-9
P, p	(pico-)	1E-12
F, f	(femto-)	1E-15

Alphabetic characters immediately following a number and its scale factor are ignored. Thus, 1000, 1000.0, 1000HZ, 1E3, 1.0E3, 1KHZ, .10E+4V, 10000E-1VOLTS, and 1K all represent the *same* number. Similarly, M, MA, MSEC, and MMHOS all represent the same scale factor.

The combination of an exponent with scale factor is accepted. Thus, 10E-4KV is acceptable and has the value of 1.

Functions

Circuit element values may be defined using global parameters or functions. You can use algebraic and standard functions of parameters passed to subcircuits to define element and model parameter values. Global parameters or subcircuit parameters may be used to evaluate the function. Enclose these functions in double quotes (" ").

There is no restriction on the number of variables in each circuit definition block.

Formats

```
"namei = F (arg1, ...argn, name1, namei-1)"
F (arg1, ...argn, name1, namei-1)"
"namei = LF (arg1, ...argn, name1 ...namei-1) ?
F1 (arg1, ...argn, name1 ...namei-1) :
F2 (arg1, ...argn, name1 ...namei-1)"
"LF (arg1, ...argn, name1 ...namei-1) ?
F1 (arg1, ...argn, name1 ...namei-1) :
F2 (arg1, ...argn, name1 ...namei-1)"
```

where:

F, F1, F2 represent any arithmetic function. These functions can include the following:

Operands:	constants parameters
Delimiters:	{ } ()
Continuation:	+ (if 1st character in line)
Arithmetic Operators:	+ add - subtract * multiply / divide ** exponential
Functions:	abs: absolute value sqrt: square root ln: natural log function log10: base 10 log function exp: exp(x) is equal to e ** x sin: sin function cos: cosine function tan: tangent function asin: arcsin function acos: arccos function atan: arctan function sinh: hyperbolic sin function cosh: hyperbolic cosine function tanh: hyperbolic tangent function

LF represents any logical-arithmetic functions. A logical-arithmetic function is a function which may have all the operations of an arithmetic function as well as the logicals and relationals.

Logicals:	&& and or
Relationals:	> greater than < less than >= greater than or equal to <= less than or equal to == equal to

arg1,...argn are arguments. There are two kinds of args:

- 1) .PARAM defined parameters
- 2) parameters defined on the subcircuit definition. The actual values may be passed by subcall invocation.

namei the name of the ith arithmetic function defined.

Function formats 3 and 4 have the same meaning as Conditional Operators (? :) in the C language (for example, condition?true:false).

Functions can appear anywhere in the subcircuit as well as in the nominal circuit. In the nominal circuit all the parameters must be .PARAM parameters or the name of functions defined in previous lines of the input. In this case, name1 is global and is added to a .PARAM list. It can be used in any subcircuit.

A function is written in free format; blanks or tabs can appear anywhere. A line may be continued by putting a plus sign (+) as the first character in the following line as a part of the netlist card; otherwise, it is part of the free format function. The following examples show how functions are used.

Example 1

```
.
.PARAM ST=7
.
"STAM=0.25*COS(ST/7-1.)"
.
.
X1 3 2 4 HE1 PPD=6U
.
.SUBCKT HE1 1 2 3 THICK=9P PPD=5U
.
"PM=PPD > 16.Udcos(PPD) ? 5.U : 1./2. * PPD"
"PL=2 * (0.25 * PPD + 0.5 * PM) *
+ (ln(exp(cosh(STAM - 0.25))))"
M1 1 2 3 THE W=PL L=" -0.5U * 3. + 1.5 *
+ PM " AS=THICK AD =9P
+ PS=PL PD=PPD
.
```

Example 2

```
.SUBCKT SUB1 1 2 3 P1=2U P2=3U
M1 1 2 3 4 MOD L=P1 W=P2
.SUBCKT SUB1 1 2 3 P1=2U P2=3U
M1 1 2 3 4 MOD L="P1" W="P2"
```

In example 2, the first format runs faster, but both formats yield the same results.

Statistical Spread

The information in this section provides you with the syntax required for assigning a statistical spread to all circuit parameters. The following parameters may have statistical spread in the nominal circuit:

- resistance
- capacitance
- inductance
- transmission line
- polynomial coefficients (describing controlled sources)
- DC and small signal frequency values for:
 - independent sources
 - geometric parameters on semiconductor device cards
 - model parameters
 - parameters passed to subcircuits

Formats

In most instances, as defined above, a statistical spread can be entered directly using the STAT keywords followed by the statistical parameters. As shown in the following example:

```
Rxxxxxxx n1 n2 value <STAT tol1 :distname <STRAK=tol2> <TRAK=x>>
```

```
Rxxxxxxx n1 n2 value <STAT min max :distname <STRAK=tol2> <TRAK=x>>
```

In those instances where this is not possible, you can enter the statistical variables using the following formats:

```
parameter name = value <STAT tol1 :distname <STRAK=tol2> <TRAK=x>>
```

```
parameter name = value <STAT min max :distname> <STRAK=tol2> <TRAK=x>>
```

where:

parameter name is the name of the device parameter.

value is the value of the parameter. This can be a positive or negative number, but not zero.

STAT is the keyword for statistical parameters

tol1 is the percentage tolerance of the parameter for the corresponding distribution.

min is the minimum value of the parameter for a distribution. You must include a maximum value if you assign a minimum value.

max is the maximum value of the parameter for a distribution.

:distname is the name of the distribution, which can be any of the legal keywords (types) as well as names defined from the .DISTR card. The legal keywords (types) are:

UNIFORM
GAUSSIAN
DEXPON
GAMMA
LOGNOR
WEIBUL
BIMOD

Any circuit parameter that varies according to a user-named distribution (defined in a .DISTR statement) may track other parameters by means of the TRAK and/or STRAK options described below. Any distribution name you define must begin with an alphabetic character and can include numbers. All distributions are scaled to the tolerance or minimum and maximum values defined after STAT.

STRAK is the keyword for stochastic tracking. Use STRAK when you want two different parameters to track each other according to a distribution you have already defined or will define later. During stochastic tracking, one variable is tracked as before, but after tracking, the new variable is randomly chosen with Gaussian distribution where:

$$3 \text{ sigma} = \text{tol2\%} * \text{tracked variable.}$$

tol2 is the percentage tolerance for stochastic tracking.

TRAK	is the keyword for parameter tracking. Use TRAK when parameters are directly related. Parameters are associated by use of the same distribution name. VBASE multiplies the value of the variable .DISTR card by the value of TRAK, and scales accordingly. Two or more parameters may track each other in this way.
x	is the multiplication factor used for tracking the values of two or more parameters. x is any real number not equal to zero

Examples

```
.MODEL 2N2222 NPN BF=200 STAT 150 250 :GAUSS
  (where BF is the beta forward of a transistor)
XCMP4 3 3 2 4 5 6 7 LF155 I_BIAS=30P I_B_TC=3P I_OS=3P
+ STAT -20P 20P :GAUSS I_OS_TC=1P V_OS=3M STAT -5 5M
+ :GAUSS V_OS_TC=5U R_IN=1E12 V_NOIS=20N AV=200K
+ AV_TC=-8M CMRR=100 SR=5.0 STAT 3 10 :GAUSS
+ GBW=2.5MEG STAT 1.2MEG 5 MEG :GAUSS
+ GBW_TC=-8M R_OUT=50 +VO_SWP=13 VO_SWN=-13
+ I_SP=2M STAT 1M 4M :GAUSS I_SP_TC=-2M
XCMP3 1 2 AMP GAIN=1E3 STAT 10 :GAUSS
```

Chapter 2

Voltage and Current Sources

This chapter defines specifications for arbitrary, dependent and independent voltage and current sources. The following topics are discussed:

- Arbitrary Sources
- Dependent Sources
 - Voltage Controlled Voltage Source
 - Current Controlled Current Source
 - Voltage Controlled Current Source
 - Current Controlled Voltage Source
- Independent Sources
 - Exponential Specification
 - Pulse Specification
 - Piecewise-Linear Specification
 - Single-Frequency FM Specification
 - Sinusoidal Specification

Arbitrary Sources

Nonlinear, arbitrary dependent sources use the following formats:

Formats

Bxxxxxxx n+ n- **I**=expr

Bxxxxxxx n+ n- **V**=expr

where:

Bxxxxxx	represents the name of an arbitrary source.
n+ n-	represent the numbers or names (integers, alphanumerics) of the positive and negative connecting nodes, respectively.
I	the device is a current source. The value of the expression determines the current flowing through the device from n+ to n-.
V	the device is a voltage source. The value of the expression determines the voltage across the device between n+ and n-.
expr	an arithmetic expression of functions operating on voltages and currents through voltage sources in the system. In addition, constants, reserved constants, and parameters can be used.

Operands	constants reserved constants: e (natural base) pi parameters voltage and current variables
Delimiters:	{ } ()
Continuation:	+ (if first character in line)
Operators:	+ (add) - (subtract) * (multiply) / (divide) ^ (exponential)
Functions:	abs (absolute value) sqrt (square root) ln (natural log, base e) log (log, base 10) exp (exp(x) is equal to e^x) sin, cos, tan asin, acos, atan sinh, cosh, tanh

Examples

```
B1 0 1 I=cos(v(1)) + sin(v(2))  
B1 0 1 V=ln(cos(log(v(1,2)^2))) - v(3)^4 + v(2)^v(1)  
B1 3 4 I = 17  
B1 3 4 V = -5  
B1 3 4 V = exp( pi^i(vdd) )
```

Dependent Sources

All types of nonlinear dependent sources are treated as $Y=F(X)$, where $F(X)$ is a polynomial of dimension n ($n \geq 1$), or user-defined code (see the DIABLO section of this manual). The coefficients of the polynomial are specified in increasing order with respect to the power of the corresponding term in the polynomial.

For example, $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7$ and P_8 are specified sequentially for a dependent source with two dimensions (controlling variables X_1, X_2 specified in this sequence). The function is computed as follows:

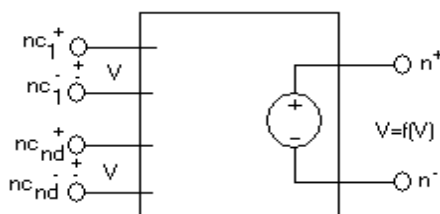
$$Y = P_0 + P_1 * X_1 + P_2 * X_2 + P_3 * X_1^2 + P_4 * X_1 * X_2 + P_5 * X_2^2 + P_6 * X_1^3 + P_7 * X_1^2 * X_2 + P_8 * X_1 * X_2^2$$

If the same set of coefficients is used for a three dimensional polynomial, then the function is computed as:

$$Y = P_0 + P_1 * X_1 + P_2 * X_2 + P_3 * X_3 + P_4 * X_1^2 + P_5 * X_1 * X_2 + P_6 * X_1 * X_3 + P_7 * X_2^2 + P_8 * X_2 * X_3$$

If a single coefficient is given, then the source is assumed to be controlled by a single controlling variable and the single coefficient specified is the linear term. For example, $y = P_0 X_1$.

Voltage Controlled Voltage Source



Formats

Linear:

Exxxxxxxx n⁺ n⁻ nc₁⁺ nc₁⁻ p₀ < **IC** = icval₁ >

Nonlinear Polynomial:

Exxxxxxxx n⁺ n⁻ < **POLY**(nd) > nc₁⁺ nc₁⁻ < .. nc_{nd}⁺ nc_{nd}⁻ > p₀ < p₁ .. >
+ < **IC** = icval₁ ... icval_{nd} >

Nonlinear Function:

Exxxxxxxx n⁺ n⁻ **FUNC**(nd) nc₁⁺ nc₁⁻ ... nc_{nd}⁺ nc_{nd}⁻ func_name
+ < p₀ ... > < **IC** = icval₁ ... icval_{nd} >

Linear Function:

Exxxxxxxx n⁺ n⁻ **LIN_FUNC**(nd) nc₁⁺ nc₁⁻ ... nc_{nd}⁺ nc_{nd}⁻ func_name
+ < p₀ ... > < **IC** = icval₁ ... icval_{nd} >

Step Function:

Exxxxxxxx n⁺ n⁻ **STEP_FUNC**(nd) nc₁⁺ nc₁⁻ ... nc_{nd}⁺ nc_{nd}⁻ func_name
+ < p₀ ... > < **IC** = icval₁ ... icval_{nd} >

Nonlinear Time Derivative Function:

```

Exxxxxxx  n+  n-  D_DT(nd)  nc1+ nc1- ... ncnd+ ncnd-  func_name
+ < p0 ... >  < IC = icval1 ... icvalnd >

```

where:

E xxxxxxx	represents the unique source name.
n ⁺ n ⁻	represent numbers or names (integers, alphanumerics) denoting the positive and negative nodes of the source respectively.
POLY	indicates that the voltage across E xxxxxxx is a polynomial function of the controlling voltages. An nd must be specified.
FUNC	indicates that the voltage across E xxxxxxx is a DIABLO function, func_name, of the controlling voltages (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
LIN_FUNC	indicates that the voltage across E xxxxxxx is a linear DIABLO function, func_name, of the controlling voltages (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
STEP_FUNC	indicates that the voltage across E xxxxxxx is a step DIABLO function, func_name, of the controlling voltages (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
D_DT	indicates that the voltage across E xxxxxxx is the time derivative of a DIABLO function, func_name, of the controlling voltages (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
nd	represents the dimension (number of variables) of the polynomial or function describing the voltage across E xxxxxxx. One pair of controlling nodes must be specified for each dimension. nd must be positive.
nc ₁ ⁺ nc ₁ ⁻	represent numbers or names (integers, alphanumerics) denoting the positive and negative nodes of the controlling voltage respectively.
func_name	represents the name of the DIABLO function (see <i>Appendix C, DIABLO Language Structure</i>).
p ₀ , p ₁ , ...	For POLY , p ₀ , p ₁ , .. represent the coefficients of the polynomial describing the voltage across E xxxxxxx. For FUNC , LIN_FUNC , STEP_FUNC and D_DT , p ₀ , p ₁ , .. represent the parameters which can be passed into the function. For a linear source, p ₀ represent the voltage gain.
IC	indicates that an initial guess of the controlling voltages for calculating the operating point is to be specified.
<icval ₁ .. icval _{nd} >	the initial estimate of the controlling voltages across corresponding nc ₁ ⁺ nc ₁ ⁻ ... nc _{nd} ⁺ nc _{nd} ⁻ .

Examples

```
EM1 1 0 93 94 10.0
```

represents the following relationship:

```
V(1,0)= 10.0 x V(93,94)
```

```
EP2 2 1 POLY(2) 1 4 21 20 0.1 0.001 0.3 10.0
```

represents the following relationship:

```
V(2,1)= 0.1 + 0.001 x V(1,4) + 0.3 x V(21,20) + 10.0 x V(1,4)2
```

```
EP2 2 1 FUNC(2) 1 4 21 20 DIAB1 0.1 0.001 0.3 10.0
```

represents the following relationship:

```
V(2,1)= DIAB1(V(1,4),V(21,20),0.1,0.001,0.3,10)
```

where DIAB1 is a user-defined DIABLO function.

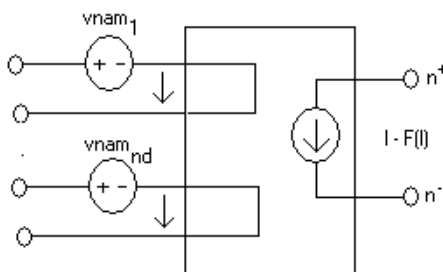
```
EP2 2 1 D_DT(2) 1 4 21 20 DIAB1 0.1 0.001 0.3 10.0
```

represents the following relationship:

```
V(2,1)= d(DIAB1(V(1,4),V(21,20),0.1,0.001,0.3,10))/dt
```

where DIAB1 is a user-defined DIABLO function.

Current Controlled Current Source



Formats

Linear:

```
Fxxxxxxx n+ n- vnam1 p0 < IC = icval1 >
```

Nonlinear Polynomial:

```
Fxxxxxxx n+ n- < POLY(nd) > vnam1 < ... vnamnd > p0 < p1 ... >
+ < IC = icval1 ... icvalnd >
```

Nonlinear Function:

```
Fxxxxxxx n+ n- FUNC(nd) vnam1 ... vnamnd func_name
+ < p0 ... > < IC = icval1 ... icvalnd >
```

Linear Function:

```
Fxxxxxxx n+ n- LIN_FUNC(nd) vnam1 ... vnamnd func_name
+ < p0 ... > < IC = icval1 ... icvalnd >
```

Step Function:

```
Fxxxxxxx n+ n- STEP_FUNC(nd) vnam1 ... vnamnd func_name
+ < p0 ... > < IC = icval1 ... icvalnd >
```

Nonlinear Time Derivative Function:

```
Fxxxxxxx n+ n- D_DT(nd) vnam1 ... vnamnd func_name
+ < p0 ... > < IC=icval1 ... icvalnd >
```

where:

Fxxxxxxx represents the unique source name.

n^+ n^- represent numbers or names (integers, alphanumerics) denoting the positive and negative nodes of the source respectively.

POLY	indicates that the current through Fxxxxxxx is a polynomial function of the controlling currents. An nd must be specified.
FUNC	indicates that the current through Fxxxxxxx is a DIABLO function, func_name , of the controlling currents (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
LIN_FUNC	indicates that the current through Fxxxxxxx is a linear DIABLO function, func_name , of the controlling currents (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
STEP_FUNC	indicates that the current through Fxxxxxxx is a step DIABLO function, func_name , of the controlling voltages (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
D_DT	indicates that the current through Fxxxxxxx is the time derivative of a DIABLO function, func_name , of the controlling currents (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
nd	the dimension (number of variables) of the polynomial or function describing the current through Fxxxxxxx . One controlling voltage source must be specified for each dimension. nd must be positive.
vnam₁, vnam₂ ...	names of the voltage sources through which the controlling currents flow. The controlling current flows from the positive node of the voltage source through the voltage source to its negative node.
func_name	represents the name of the DIABLO function (see <i>Appendix C, DIABLO Language Structure</i>).
p₀, p₁, ...	For POLY , p₀, p₁, .. represent the coefficients of the polynomial describing the current through Fxxxxxxx . For FUNC , LIN_FUNC , STEP_FUNC and D_DT , p₀, p₁, .. represent the parameters which can be passed into the function. For a linear source, p₀ represents the current gain.
IC	indicates that an initial guess of the controlling currents for calculating the operating point is to be specified.
<icval₁ ... icval_{nd}>	the initial estimate of the controlling currents through corresponding vnam₁ ... vnam_{nd} .

Examples

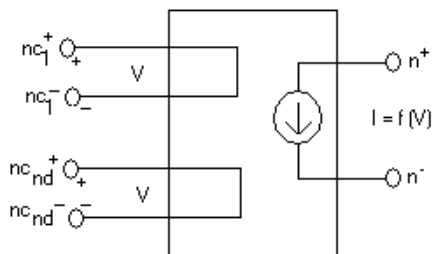
```
FM1 1 0 VCC 1MA
      represents the following relationship:
I(1,0)= 0.001 x I(VCC)

FP3 2 1 POLY(3) VOUT VDIFF VCLK 10MA 0.1 2.0 4.2
      represents the following relationship:
I(2,1)= 0.01 + 0.1 x I(VOUT) + 2.0 x I(VDIFF) + 4.2 x I(VCLK)

FP3 2 1 FUNC(3) VOUT VDIFF VCLK DIAB1 10MA 0.1 2.0 4.2
      represents the following relationship:
I(2,1)= DIAB1(I(VOUT),I(VDIFF),I(VCLK),10MA,0.1,2.0,4.2)
where DIAB1 is a user-defined DIABLO function.
```

FP3 2 1 D_DT(3) VOUT VDIFF VCLK DIAB1 10MA 0.1 2.0 4.2
 represents the following relationship:
 $I(2,1) = d(DIAB1(I(VOUT), I(VDIFF), I(VCLK), 10MA, 0.1, 2.0, 4.2)) / dt$
 where DIAB1 is a user-defined DIABLO function.

Voltage Controlled Current Source



Formats

Linear:

Gxxxxxxx n⁺ n⁻ nc₁⁺ nc₁⁻ p₀ < **IC** = icval₁ >

Nonlinear Polynomial:

Gxxxxxxx n⁺ n⁻ < **POLY**(nd) > nc₁⁺ nc₁⁻ < ... nc_{nd}⁺ nc_{nd}⁻ > p₀ < p₁ ... > + < **IC** = icval₁ ... icval_{nd} >

Linear Function:

Gxxxxxxx n⁺ n⁻ **LIN_FUNC**(nd) nc₁⁺ nc₁⁻ ... nc_{nd}⁺ nc_{nd}⁻ func_name + < p₀ ... > < **IC** = icval₁ ... icval_{nd} >

Step Function:

Gxxxxxxx n⁺ n⁻ **STEP_FUNC**(nd) nc₁⁺ nc₁⁻ ... nc_{nd}⁺ nc_{nd}⁻ func_name + < p₀ ... > < **IC** = icval₁ ... icval_{nd} >

Nonlinear Function:

Gxxxxxxx n⁺ n⁻ **FUNC**(nd) nc₁⁺ nc₁⁻ ... nc_{nd}⁺ nc_{nd}⁻ func_name + < p₀ ... > < **IC** = icval₁ ... icval_{nd} >

Nonlinear Time Derivative Function:

Gxxxxxxx n⁺ n⁻ **D_DT**(nd) nc₁⁺ nc₁⁻ ... nc_{nd}⁺ nc_{nd}⁻ func_name + < p₀ ... > < **IC** = icval₁ ... icval_{nd} >

where:

Gxxxxxxx represents the unique source name.

n⁺ n⁻ represent numbers or names (integers, alphanumerics) denoting the positive and negative nodes of the source respectively.

POLY indicates that the current through **Gxxxxxxx** is a polynomial function of the controlling currents. An nd must be specified.

FUNC	indicates that the current through $G_{xxxxxxx}$ is a DIABLO function, func_name, of the controlling voltages (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
LIN_FUNC	indicates that the current through $G_{xxxxxxx}$ is a linear DIABLO function, func_name, of the controlling voltages (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
STEP_FUNC	indicates that the current through $G_{xxxxxxx}$ is a step DIABLO function, func_name, of the controlling voltages (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
D_DT	indicates that the current through $G_{xxxxxxx}$ is the time derivative of a DIABLO function func_name, of the controlling voltages (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
nd	the dimension (number of variables) of the polynomial or function describing the current through $G_{xxxxxxx}$. One pair of controlling nodes must be specified for each dimension and must be positive.
$nc_1^+ nc_1^-$	represent numbers or names (integers, alphanumerics) denoting the positive and negative nodes of the controlling voltage respectively.
func_name	the name of the DIABLO function (see <i>Appendix C, DIABLO Language Structure</i>).
p_0, p_1, \dots	For POLY , p_0, p_1, \dots represent the coefficients of the polynomial describing the current through $G_{xxxxxxx}$. For FUNC , LIN_FUNC , STEP_FUNC and D_DT , p_0, p_1, \dots represent the parameters which can be passed into the function. For a linear source, p_0 represents the transconductance.
IC	indicates that an initial guess of the controlling voltages for calculating the operating point is to be specified.
$\langle icval_1 \dots icval_{nd} \rangle$	the initial estimate of the controlling voltages across corresponding $nc_1^+ nc_1^- \dots nc_{nd}^+ nc_{nd}^-$.

Examples

GM1 1 0 93 94 6.2MHO

represents the following relationship:

$I(1,0) = 6.2 \times V(93,94)$

GP2 2 1 POLY(2)10 14 11 21 0.0 0.01 0.3 0.4

represents the following relationship:

$I(2,1) = 0.0 + 0.01 \times V(10,14) + 0.3 \times V(11,21) + 0.4 \times V(10,14)^2$

GP2 2 1 FUNC(2) 10 14 11 21 DIAB1 0.0 0.01 0.3 0.4

represents the following relationship:

$I(2,1) = \text{DIAB1}(V(10,14), V(11,21), 0.0, 0.01, 0.3, 0.4)$

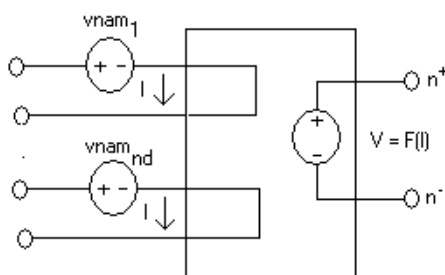
where DIAB1 is a user-defined DIABLO function.

GP2 2 1 D_DT(2) 10 14 11 21 DIAB1 0.0 0.01 0.3 0.4

$I(2,1) = d(\text{DIAB1}(V(10,14), V(11,21), 0.0, 0.01, 0.3, 0.4)) / dt$

where DIAB1 is a user-defined DIABLO function.

Current Controlled Voltage Source



Formats

Linear:

Hxxxxxxx n^+ n^- $vnam_1$ p_0 < **IC** = $icval_1$ >

Nonlinear Polynomial:

Hxxxxxxx n^+ n^- < **POLY**(nd) > $vnam_1$ < ... $vnam_{nd}$ > p_0 < p_1 ... >
+ < **IC** = $icval_1$... $icval_{nd}$ >

Nonlinear Function:

Hxxxxxxx n^+ n^- **FUNC**(nd) $vnam_1$... $vnam_{nd}$ $func_name$
+ < p_0 ... > < **IC** = $icval_1$... $icval_{nd}$ >

Linear Function:

Hxxxxxxx n^+ n^- **LIN_FUNC**(nd) $vnam_1$... $vnam_{nd}$ $func_name$
+ < p_0 ... > < **IC** = $icval_1$... $icval_{nd}$ >

Step Function:

Hxxxxxxx n^+ n^- **STEP_FUNC**(nd) $vnam_1$... $vnam_{nd}$ $func_name$
+ < p_0 ... > < **IC** = $icval_1$... $icval_{nd}$ >

Nonlinear Time Derivative Function:

Hxxxxxxx n^+ n^- **D_DT**(nd) $vnam_1$... $vnam_{nd}$ $func_name$
+ < p_0 ... > < **IC** = $icval_1$... $icval_{nd}$ >

where:

Hxxxxxxx represents the unique source name.

n^+ n^- represent numbers or names (integers, alphanumerics) denoting the positive and negative nodes of the source respectively.

POLY indicates that the voltage across **Hxxxxxxx** is a polynomial function of the controlling currents. An nd must be specified.

FUNC indicates that the voltage across **Hxxxxxxx** is a DIABLO function, $func_name$, of the controlling currents (see *Appendix C, DIABLO Language Structure*) and optional parameters.

LIN_FUNC indicates that the voltage across **Hxxxxxxx** is a linear DIABLO function, $func_name$, of the controlling currents (see *Appendix C, DIABLO Language Structure*) and optional parameters.

STEP_FUNC	indicates that the voltage across Hxxxxxxx is a step DIABLO function, func_name , of the controlling currents (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
D_DT	indicates that the voltage across Hxxxxxxx is the time derivative of a DIABLO function, func_name , of the controlling currents (see <i>Appendix C, DIABLO Language Structure</i>) and optional parameters.
nd	the dimension (number of variables) of the polynomial or function describing the voltage across Hxxxxxxx . One controlling voltage source must be specified for each dimension. nd must be positive.
vnam₁, vnam₂	names of the voltage sources through which the controlling currents flow. The controlling current flows from the positive node of the voltage source through the voltage source to its negative node.
func_name	the name of the DIABLO function (see <i>Appendix C, DIABLO Language Structure</i>).
p₀, p₁,...	For POLY , p₀, p₁, .. represent the coefficients of the polynomial describing the voltage across Hxxxxxxx . For FUNC , LIN_FUNC , STEP_FUNC and D_DT , p₀, p₁, .. represent the parameters which can be passed into the function. For a linear source, p₀ represents the transresistance.
IC	indicates that an initial guess of the controlling voltages for calculating the operating point is to be specified.
<icval₁ ... icval_{nd}>	the initial estimate of the controlling currents through corresponding vnam₁ ... vnam_{nd} .

Examples

```
HM1 1 0 VDD 1KOHM
```

represents the following relationship:

```
V(1,0) = 1000.0 x I(VDD)
```

```
HP3 2 1 POLY(3) VIN VSS VCLK 0.5 100.0 2000.0 20.0
```

represents the following relationship:

```
V(2,1)= 0.5 + 100 x I(VIN) + 2000.0 x I(VSS) + 20.0 x I(VCLK)
```

```
HP3 2 1 FUNC(3) VIN VSS VCLK DIAB1 0.5 100.0 2000.0 20.0
```

```
V(2,1)= DIAB1(I(VIN),I(VSS),I(VCLK),0.5,100.0,2000.0,20.0)
```

where **DIAB1** is a user-defined DIABLO function.

```
HP3 2 1 D_DT(3) VIN VSS VCLK DIAB1 0.5 100.0 2000.0 20.0
```

```
V(2,1)=d(DIAB1(I(VIN),I(VSS),I(VCLK),0.5,100.0,2000.0,20.0))/dt
```

where **DIAB1** is a user-defined DIABLO function.

Independent Sources

Independent voltage and current sources use the formats described below.

Formats

```

Ixxxxxxx n+ n- < DC dcval > < AC < mag < phase > > >
+ < type ( param param param ... ) >
Vxxxxxxx n+ n- < DC dcval > < AC < mag < phase > > >
+ < type ( param param param ... ) >

```

where:

Ixxxxxxx represents the name of a current source.

Vxxxxxxx represents the name of a voltage source.

n+ n- represent the positive and negative connections for the node.

DC indicates that the optional DC source value follows.

dcval DC (constant) source value. If not specified, the time=0 value of the time variable source function is used. If no time variable source function is specified, then dcval=0 is used.

AC indicates that optional AC source values follow.

mag magnitude of the linear AC source. If not specified, mag=1.0 is assumed.

phase phase of the AC source. If not specified, phase=0.0 is assumed.

type represents an optional source type keyword for a time-variable source. Valid source types are:

EXP	exponential source
FSIN	full-rectified sine
HSIN	half-rectified sine
PULSE	pulsed source
PWL	piecewise-linear source
SFFM	single-frequency FM source
SIN	sinusoidal source

Each of these source types is described in more detail in the following pages.

param parameter values that define the time-dependent source characteristics. The number and meaning of these parameters depends upon the type. These values do not need to be enclosed in parentheses.

The AC, DC, and type specification sets are not order dependent.

Examples

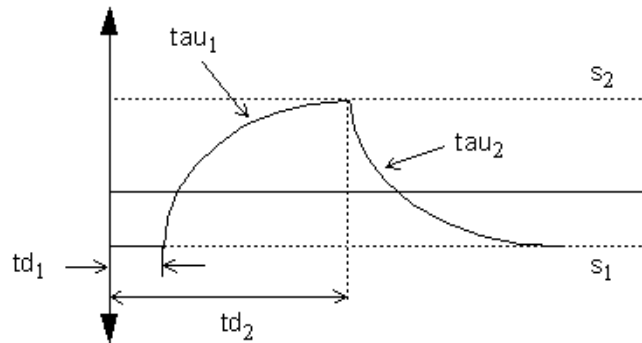
```

vcc 10 0 dc 6
IBIAS 14 2 5UA
vin1 13 0 pulse(0v 5v 10ns 10ns 10ns 100ns 200ns)

```

Exponential Specification

EXP defines an exponential source with both a rising and a falling waveform.



Format

EXP (s1 s2 < td₁ tau₁ td₂ tau₂ >)

where:

EXP indicates an exponential source. The parameters that follow do not need to be enclosed in parentheses.

s₁ is the initial value in volts or amps.

s₂ is the pulsed value in volts or amps.

td₁ is the rise delay time in seconds. The default value is zero.

tau₁ is the rise time constant in seconds. The default value is tstep.

td₂ is the fall delay time in seconds. The default value is td₁ + tstep.

tau₂ is the fall time constant in seconds. The default value is tstep.

If an optional parameter is omitted, the default value will be used. See the .TRAN instruction in *Chapter 7, Instructions* for additional information on tstep (the output time step) and tlast (the last simulation timepoint). An exponential source will have the following values versus time:

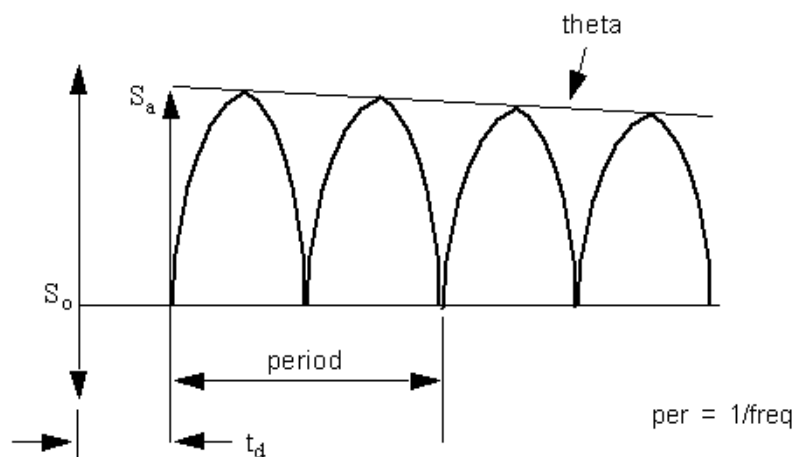
Time	Value
0-td ₁	s ₁
td ₁ -td ₂	$s_1 + (s_2 - s_1) \times \left(1 - e^{-\frac{(\text{time} - \text{td}_1)}{\text{tau}_1}}\right)$
td ₂ -tlast	$s_1 + (s_2 - s_1) \times \left(1 - e^{-\frac{(\text{time} - \text{td}_1)}{\text{tau}_1}}\right) + (s_1 - s_2) \times \left(1 - e^{-\frac{(\text{time} - \text{td}_2)}{\text{tau}_2}}\right)$

Example

VEXPON 29 0 EXP (-4V -1V 2NS 30NS 60NS 40NS)

Full-Wave Rectified Sinusoidal

FSIN defines a full-wave rectified sinusoidal source with a damping factor and an initial delay time.



Format

FSIN <USOIDAL>(s_o s_a freq t_d theta phase)

where:

- FSIN** indicates a damped full-wave rectified sinusoidal source. The parameters that follow do not need to be enclosed in parentheses.
- s_o is the offset of source in volts or amps
- s_a is the amplitude in volts or amps
- freq is the frequency in hertz. The default value is 1/tstop (tstop as specified on the .TRAN instruction).
- t_d is the delay in seconds. The default value is zero.
- theta is the damping factor in 1/second. The default value is zero.
- phase is the phase shift in radians. The default is zero

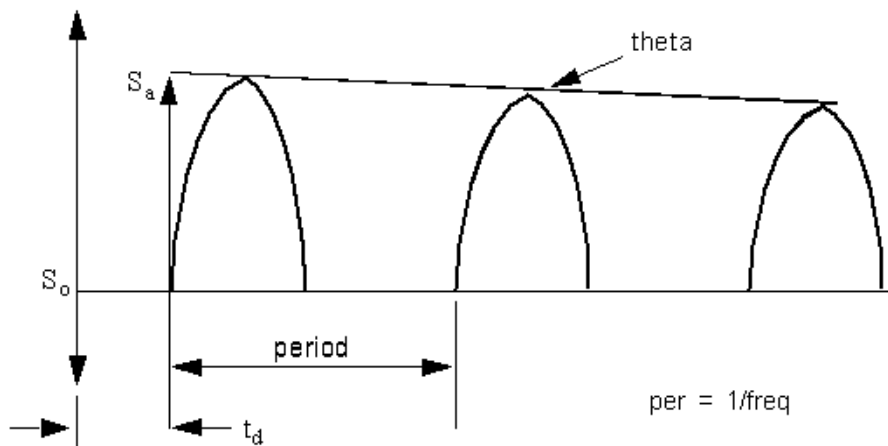
Examples

```
VIN 2 0 FSIN(0 1 1E7 0 0 0)
```

```
VIN 3 0 SIN(0 1 100MEG 1NS 1E10 0)
```

Half-Wave Rectified Sinusoidal

HSIN defines a half-wave rectified sinusoidal source with a damping factor and an initial delay time.



Format

HSIN <USOIDAL> (S_o S_a freq t_d theta phase)

where:

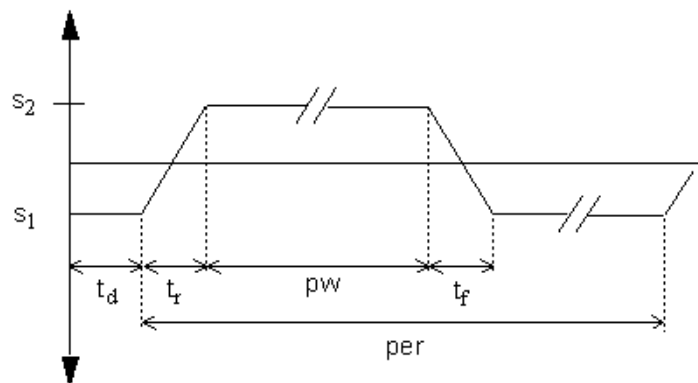
HSIN	indicates a damped half-wave rectified sinusoidal source. The parameters that follow do not need to be enclosed in parentheses.
S_o	is the offset of source in volts or amps
S_a	is the amplitude in volts or amps
freq	is the frequency in hertz. The default value is 1/tstop (tstop as specified on the .TRAN instruction).
t_d	is the delay in seconds. The default value is zero.
theta	is the damping factor in 1/second. The default value is zero.
phase	is the phase shift in radians. The default is zero

Example

```
VIN 2 0 HSIN(0 1 1E7 0 0 0)
```

Pulse Specification

PULSE specifies a regular, periodic waveform.



Format

PULSE (s_1 s_2 t_d t_r t_f pw per > > > >)
where:

- PULSE** indicates a pulsed source. The parameters that follow do not need to be enclosed in parentheses.
- s_1 is the initial value in volts or amps.
- s_2 is the pulsed value in volts or amps.
- t_d is the delay time in seconds. The default value is zero.
- t_r is the rise time in seconds. The default value is tstep.
- t_f is the fall time in seconds. The default value is tstep.
- pw is the pulse width in seconds. The default value is tlast (tlast as specified on the .TRAN instruction).
- per is the period in seconds. The default value is tlast.

If an optional parameter is omitted, the default value is used.

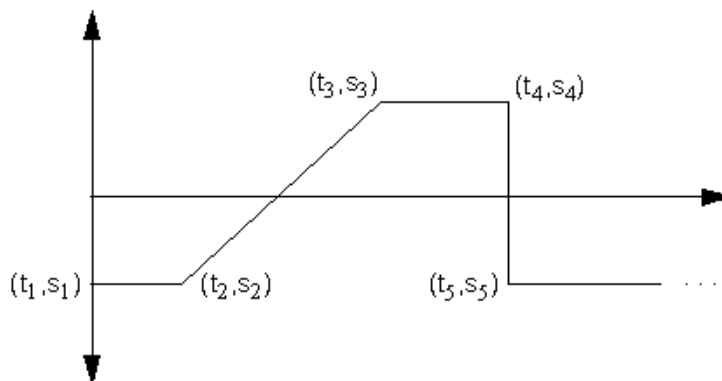
See the .TRAN instruction in *Chapter 7, Instructions* for additional information on tstep (the output time step) and tlast (the last simulation timepoint).

Examples

```
VIN 3 0 PULSE(-1 1 2NS 2NS 2NS 50NS 100NS)
istart 4 22 pulse 0ua 20ua 0 1ns 2ns 10ns
vp 14 36 pulse (2v 0v 1n 1n)
```

Piecewise-Linear Specification

PWL defines an arbitrary piecewise-linear waveform.



Formats

PWL ($t_1, s_1 < t_2, s_2 < t_3, s_3 < t_4, s_4 < \dots > > > >$)

PWL ($t_1, s_1 < t_2, s_2 < \dots t_n, s_n > >$) **R** $< tr > >$

PL ($s_1, t_1 (s_2, t_2 \dots s_n, t_n > >) < R < tr > >$

where:

PWL	indicates a piecewise-linear source. The values that follow do not need to be enclosed by parentheses.
PL	enables the specified order for the time point-value pairs to be reversed
t_1, s_1	represent the first breakpoint time and source values. Note that t_1 must equal zero. These values are required.
$t_2, s_2 \dots$	represent optional second, third, fourth, and so on, breakpoint times and source values. The time values must be monotonically increasing.
R	specification R causes the piecewise-linear waveform to repeat after time t_n
tr	after time t_n the same waveshape beginning at TR is repeated. tr must equal a breakpoint in the waveform and the values at tr and t_n must be the same.

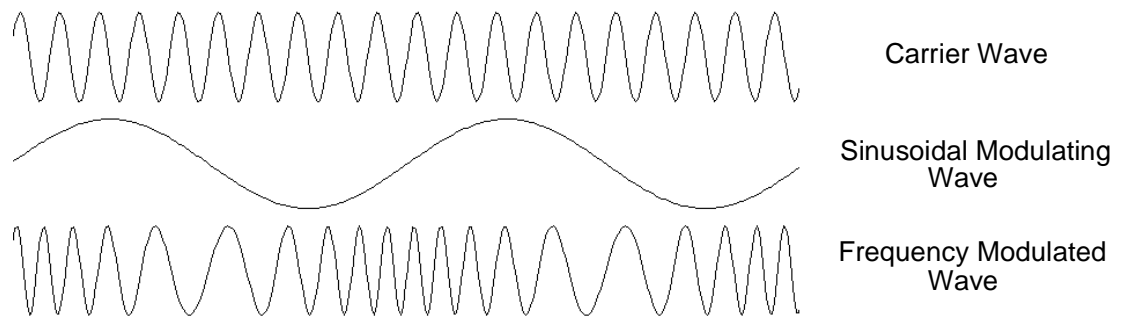
If a breakpoint time is greater than tlast (last timepoint of the simulation), the program properly interpolates the source value. This corrects a method used in SPICE.

Examples

```
VNEG 321 0 PWL (0NS -7V 10NS -7V 11NS -3V 17NS
+ -3V 18NS -7V 0NS -7V 52NS -3V 60NS -3V)
IPL 222 64 PWL(0US 1MA 20US .15MA 30US .2MA 35US
+.25MA)
```

Single-Frequency FM Specification

SFFM defines a single-frequency modulated carrier.



Format

SFFM (offs ampl < fc mdi fs >)

where:

SFFM	indicates a single-frequency FM source
offs	is the offset in volts or amps
ampl	is the amplitude in volts or amps
fc	is the carrier frequency in hertz. The default value is 1/tlast (tlast as specified on the .TRAN instruction).
mdi	is the modulation index. The default value is zero.
fs	is the signal frequency in hertz. The default value is 1/tlast (tlast as specified on the .TRAN instruction).

The waveform of an SFFM is described by the following expression:

$$Value = offs + ampl \times Sine[(2.0 \times \pi \times fc \times time) + mdi \times Sine(2.0 \times \pi \times fs \times time)]$$

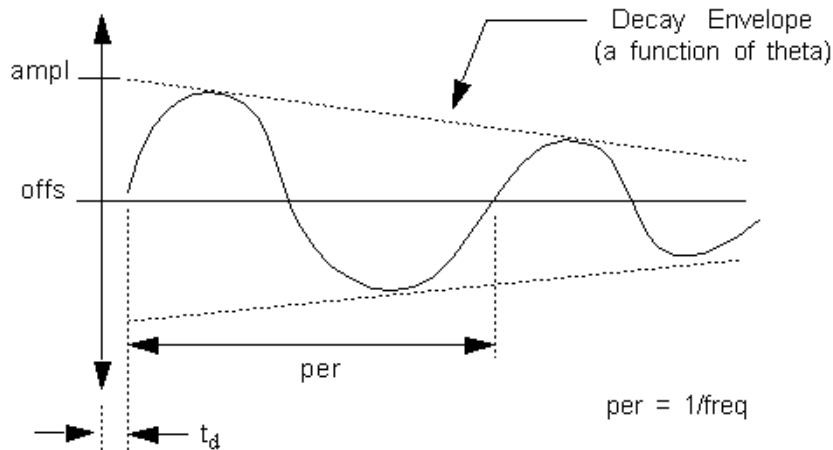
Note: See the .TRAN instruction *Chapter 7, Instructions* for additional information on tlast (the last simulation timepoint).

Example

```
IFM 77 45 SFFM (0 1UA 20KHZ 5 1KHZ)
```

Sinusoidal Specification

SIN defines a sinusoidal source with a damping factor and an initial delay time.



Format

SIN (offs ampl < freq < t_d < theta > > >)

where:

SIN indicates a damped sinusoidal source. The parameters that follow do not need to be enclosed in parentheses.

offs is the offset of source in volts or amps

ampl is the amplitude in volts or amps

freq is the frequency in hertz. The default value is 1/tlast (tlast as specified on the .TRAN instruction).

t_d is the delay in seconds. The default value is zero.

theta is the damping factor in 1/second. The default value is zero.

If an optional parameter is omitted, the default value will be used.

Note: See the .TRAN instruction *Chapter 7, Instructions* for additional information on tlast (the last simulation timepoint).

A sinusoidal source will have the following values versus time:

Time	Value
0- t_d	offs

t_d -tlast	$offs + ampl \times \text{Sine}[2.0\pi \times freq \times (time - t_d)] \times e^{-(time - t_d) \times theta}$
--------------	--

Examples

```
vsin 64 0 sin (10mv 50mv 200kHz 0 0)
IWAVE 136 137 SIN (0 200UA)
```

Chapter 3

Passive Elements

In this chapter, specifications for resistors, capacitors, inductors, and other passive elements are defined. All circuit parameters can have a statistical spread assigned to them. A definition of the statistical spread syntax shown below is found in *Chapter 1, Introduction*.

Parameter Formats

```
parameter name = value <STAT  tol1 :distname <STRAK=tol2>  <TRAK=x>>
```

```
parameter name = value <STAT  min max :distname <STRAK=tol2>  <TRAK=x>>
```

General .MODEL Specification

The .MODEL instruction specifies a set of model parameters that are referenced by one or more devices. Specific .MODEL instruction types are described in the section for each device type.

Format

.MODEL mname type (<pname=pval pname=pval> ...)

where:

.MODEL	indicates that model parameters are to be specified
mname	represents the model reference name
pname	represents a parameter keyword
pval	associated parameter value

type	represents the device type as follows:	
	C	Semiconductor Capacitor
	L	Inductor
	R	Semiconductor Resistor
	CSW	Current Controlled Switch
	SW	Voltage Controlled Switch
	D	Diode
	URC	Uniform Distributed RC Line
	DRC	Alternative name for URC model
	NPN	NPN BJT
	PNP	PNP BJT
	NJF	N-channel JFET
	PJF	P-channel JFET
	NMOS	N-channel MOSFET
	PMOS	P-channel MOSFET
	NMES	N-channel MESFET
	PMES	P-channel MESFET
	TFM	Transformer Core
	DIGITAL	Digital Model

Example

```
.MODEL DEPL NMOS (LEVEL=1 VTO=-4.0 KP=20U  
+GAMMA=1.31 LAMBDA=0.01 PHI=0.6)
```

Transformer Cores

Format

```
Pxxxxxxx nw mname LM=maglength A=area <LG=airgap> <G=slength>  
<F=freq> <B=b0>
```

where:

Pxxxxxxx	represents the unique transformer name.
nw	is the number of windings on the transformer core.
mname	is the name of the transformer model.
LM	indicates that the core's magnetic path length will be specified.
maglength	is the magnetic path length of the core (m).
A	indicates that the cross-sectional area will be specified.
area	is the cross-sectional area (m ²).
LG	indicates that the air gap size will be specified.
airgap	is the gap size (m).
G	indicates that the straight section length perpendicular to the gap will be specified.

<code>slength</code>	is the straight section length perpendicular to the gap. It affects the fringe field around the gap (m).
F	indicates that the frequency of the currents going through the transformer will be specified
<code>freq</code>	is the frequency of the currents going through the transformer. The core loss of the transformer is dependent on the frequency of the currents (Hz).
B	indicates that the residual magnetization in the core at time zero will be specified.
<code>b0</code>	is the residual magnetization in the core at time zero (tesla).

Examples

```
P1 2 TT A=1E-4 LM=2E-2
POUT 4 TMOD A=2E-4 LM=10.E-2 LG=1.E-4 G=3.E-2
+ F=100K B=.1
```

Transformer Core Models

Format

.MODEL mname **TFM** (< pname = pval > < pname = pval > ...)

where:

mname	represents the model name specified on the referencing transformer cores.
TFM	indicates a transformer core model specification.
pname	represents a reserved parameter name as described in the parameter table.
pval	parameter value associated with a parameter name. The pname=pval pairs need not be enclosed in parentheses.

The transformer core model uses hyperbolic curves to empirically match the hysteresis curves of magnetic core materials. Frequency and temperature dependencies are also included. The core loss is given by the area enclosed by the hysteresis loop during transient analysis. Eddy current and/or other frequency effects are described by the following equation:

$$HC'(f) = HC \times (FC1 + FC2 \times f^{FC3})$$

f (Hz) is the frequency. In AC analysis, you should model the loss using the ratio between the real and imaginary parts of the initial permeability, VI:

$$\tan(\delta) = \frac{Im(VI)}{Re(VI)}$$

Define the loss factor as follows:

$$\frac{\tan(\delta)}{VI} = DEL1 \times \left(\frac{f}{1.0 \times 10^6} \right)^{DEL1P} + DEL2 \times \left(\frac{f}{1.0 \times 10^6} \right)^{DEL2P}$$

Define the linear temperature dependency as follows:

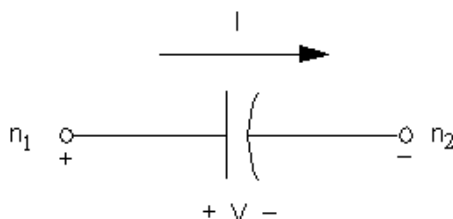
$$BS'(T) = BS \times (1 + TBS \times \delta T)$$

where δT = actual temperature – nominal temperature

Transformer Model Parameters

Name	Parameter	Units	Default
LEVEL	transformer model selector		1.0
BS	saturation inductance	T	0.2
BR	residual inductance	T	0.1
HC	coercive force	A/m	20
FC1	frequency coefficient	—	1.0
FC2	frequency coefficient	—	0.0
FC3	frequency coefficient	—	0.0
TBS	temperature coefficient for BS	—	0.0
TBR	temperature coefficient for BR	—	0.0
THC	temperature coefficient for HC	—	0.0
UI	initial relative permeability	—	5000
DEL1	loss factor coefficient	—	0.0
DEL1P	loss factor coefficient	—	0.0
DEL2	loss factor coefficient	—	0.0
DEL2P	loss factor coefficient	—	0.0
TUI	temperature coefficient for initial permeability	—	0.0
PMAX	maximum power	W	0.0

Capacitors



Formats

```

Cxxxxxxx  n1  n2  value  < M = pval > < IC = icval >
Cxxxxxxx  n1  n2  offset  mname  < L = lval >
+ < W = wval >  < M = pval > < IC = icval >
Cxxxxxxx  n1  n2  value  << TC =>  tc1 < tc2  >>
+ < M = pval > < IC = icval >
Cxxxxxxx  n1  n2  offset  mname  < L = lval >
+ < W = wval >  << SCALE = > scale >  < M = pval > < IC = icval >

```

where:

Cxxxxxxx represents the unique capacitor name.

n₁, n₂ represent numbers or names (integers, alphanumerics) of the connecting nodes.

value represents the nominal capacitance value (farads)

offset represents the offset capacitance (farads). It is used to calculate the capacitance for the associated capacitor .MODEL specification (see the section “Semiconductor Capacitor Models” of this chapter).

mname represents the model name that references a specific capacitor .MODEL instruction.

L indicates that a channel length will be specified.

lval defines the channel length (meters).

W indicates that a channel width will be specified.

wval defines the channel width (meters).

tc₁, tc₂ represent the first and the second-order temperature coefficients of the capacitor (Default: tc₁=0, tc₂=0).

SCALE indicates that a scaling factor will be specified.

M indicates that a capacitor multiplier will be specified.

pval represents the number of capacitors connected in parallel. The capacitance is computed as value * pval.

IC indicates that an initial guess of the controlling voltages for calculating the operating point is to be specified.

icval the initial estimate of the controlling voltages.

The capacitor value may be different depending upon the order of specification of the two nodes n₁ and n₂.

The capacitance as a function of temperature is:

$$C(temp) = value \times (1 + tc_1 \times (temp - T_{nom}) + tc_2 \times (temp - T_{nom})^2)$$

where T_{nom} is the nominal temperature.

Examples

```
CPYB 13 0 1UF
COSC 17 23 10U .025 .0001
C1 1 10 1UF mod L=1U
```

Semiconductor Capacitor Models

Format

```
.MODEL mname C (< pname = pval > <pname = pval> ...)
```

where:

.MODEL indicates that model parameters are to be specified.

mname represents the model reference name.

C indicates a capacitor model specification.

pname represents a reserved parameter name as described in the parameter table.

pval parameter value associated with a parameter name. The pname=pval pairs need not be enclosed in parentheses.

Capacitor Model Parameters

Name	Parameter	Units	Default
CJ	junction bottom capacitance	F/m ²	0.0
CJSW	junction side wall capacitance	F/m	0
DEFW	default width (not used)	m	1.0E-6
NARROW	narrowing due to side etching	m	0.0
VOMAX	maximum voltage	V	25E-03

The capacitance is computed as follows:

$$value = offset + CJ \times (L - NARROW) \times (W - NARROW) + 2.0 \times CJSW \times (L + W - 2.0 \times NARROW)$$

Example

```
.Model POLYCAP C CJ=10F CJSWDEFW=2U
+ NARROW=0.2U
```

Polynomial and User-Defined Capacitors

Format

```

Cxxxxxxx  n1  n2  POLY  p0  < p1 < p2  <  ... > > >
Cxxxxxxx  n1  n2  FUNC  func_name  < p0  < p1 < p2  <  ... > > >
Cxxxxxxx  n1  n2  LIN_FUNC  func_name  < p0  < p1 < p2  <  ... > > >
Cxxxxxxx  n1  n2  STEP_FUNC  func_name  < p0  < p1 < p2  <  ... > > >

```

where:

Cxxxxxxx represents the capacitor name.

n₁, n₂ represent numbers or names (integers, alphanumerics) of the connecting nodes.

POLY indicates that polynomial coefficients are to be used to define the capacitance value.

FUNC indicates that a DIABLO function is to be used to define the capacitance value (see *Appendix C, DIABLO Language Structure*).

LIN_FUNC indicates that a linear DIABLO function is to be used to define the capacitance value (see *Appendix C, DIABLO Language Structure*).

STEP_FUNC indicates that a step DIABLO function is to be used to define the capacitance value (see *Appendix C, DIABLO Language Structure*).

func_name indicates the name of the DIABLO function (see *Appendix C, DIABLO Language Structure*).

p₀ For **POLY**, p₀ represents the first polynomial coefficient. This must be specified.
For **FUNC**, p₀ represents a parameter which can be passed into the DIABLO function. It is optional (see *Appendix C, DIABLO Language Structure*).

p₁, p₂, ... For **POLY**, p₁, p₂, ... represent optional first, second, etc., order polynomial coefficients for the capacitor. For **FUNC**, p₁, p₂, ... represent parameters which can be passed into the DIABLO function (see *Appendix C, DIABLO Language Structure*).

The polynomial capacitor value is computed during simulation as a function of the voltage across the capacitor. Defining V as the voltage across the capacitor [V=V (n₁), – , V (n₂)], the capacitor value is computed from:

$$value = p_0 + p_1 \times V + p_2 \times V^2 + p_3 \times V^3 + etc$$

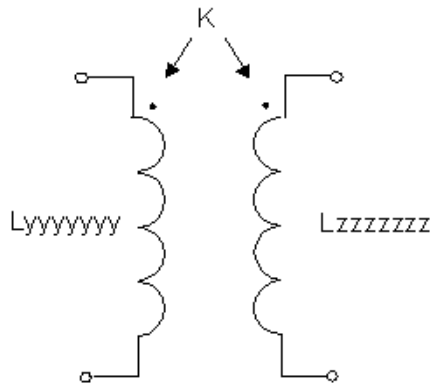
The capacitor value may be different depending upon the order of specification of the two nodes n₁ and n₂.

The high level descriptive language DIABLO is used for generating mixed discipline analog models. The language allows you to develop simulation models that can run under the VeriBest Analog (VBA) environment.

Example

```
CNONLIN 46 62 POLY 2PF .01 .00035
```

Mutual Inductors



Format

```
Kxxxxxxx Lyyyyyyy Lzzzzzzz value
```

where:

Kxxxxxxx represents the mutual inductor name. This name must be different than any other component name.

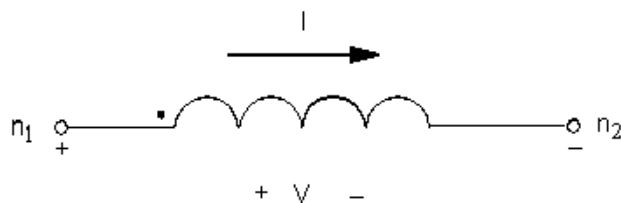
Lyyyyyyy **Lzzzzzzz** names of the two inductors which are to be mutually coupled. These inductors must be specified elsewhere in the circuit description.

value coupling coefficient of the two inductors. This value must be greater than or equal to zero and less than or equal to one.

Example

```
K24 L24 L25 .96
```

Inductors



Formats

```
Lxxxxxxx n1 n2 valtrn <<TC => tc1 < tc2 >> <NTURN = >nturn < M = pval >  
<IC = icval>
```

where:

Lxxxxxxx	represents the inductor name. This name must be different than any other inductor name.
n_1, n_2	represent numbers or names (integers, alphanumerics) of the connecting nodes.
value	inductance value (Henrys).
valtrn	inductance-per-turn (Henrys).
mname	represents the model name.
NTURN	indicates that number of turns will be specified.
nturn	number of turns (Default: nturn=1.0).
tc_1, tc_2	first and second-order temperature coefficients (Default: $tc_1=0, tc_2=0$).
M	indicates that an inductor multiplier will be specified.
pval	number of inductors connected in parallel.
IC	indicates that an initial guess of the controlling current for calculating the operating point is to be specified.
icval	the initial estimate of the controlling current.

Usage notes

The initial current is used only if VBASE does not perform a DC analysis. If a DC analysis is performed prior to the transient analysis, the initial current is computed from the DC analysis results. The conditions under which a DC analysis is performed are described in the .TRAN instruction.

The inductor model only contains the stress parameters. Value in this case has to be allowed with mname. There are no parameters in the inductor model from which value can be calculated.

Examples

```
lchoke 123 45 10mh
LPIN1 14 22 25PH
```

Semiconductor Inductor Models

Format

```
.MODEL mname L ( < pname = pval > )
```

where:

.MODEL	indicates that model parameters are to be specified.
mname	indicates an inductor model specification.
L	represents the model reference name.
IMAX	maximum current specifier.
n	maximum current in amps. The default is 250 mA.

Examples

```
.MODEL FLUX1 L IMAX=50MA
.MODEL ind1 L
```

Polynomial and User-Defined Inductors

Format

```
Lxxxxxxx n1 n2 POLY p0 < p1 < p2 < ... > > >
Lxxxxxxx n1 n2 FUNC func_name < p0 < p1 < p2 < ... > > >
Lxxxxxxx n1 n2 LIN_FUNC func_name < p0 < p1 < p2 < ... > > >
Lxxxxxxx n1 n2 STEP_FUNC func_name < p0 < p1 < p2 < ... > > >
```

where:

Lxxxxxxx represents the inductor name.

n₁, n₂ represent numbers or names (integers, alphanumerics) of the connecting nodes.

POLY indicates that polynomial coefficients are to be used to define the inductance value. The inductor flux may be different depending upon the order of specification of the two nodes **n₁** and **n₂**.

FUNC indicates that a DIABLO function is to be used to define the inductance value (see *Appendix C, DIABLO Language Structure*).

LIN_FUNC indicates that a linear DIABLO function is to be used to define the inductance value (see *Appendix C, DIABLO Language Structure*).

STEP_FUNC indicates that a step DIABLO function is to be used to define the inductance value (see *Appendix C, DIABLO Language Structure*).

func_name indicates the name of the DIABLO function (see *Appendix C, DIABLO Language Structure*).

p₀ For **POLY**, **p₀** represents the first polynomial coefficient. This must be specified.

For **FUNC**, **p₀** represents a parameter which can be passed into the DIABLO function. It is optional (see *Appendix C, DIABLO Language Structure*).

p₁, p₂, ... For **POLY**, **p₁, p₂, ...** represent optional first, second, etc., order polynomial coefficients for the inductor. For **FUNC**, **p₁, p₂, ...** represent parameters which can be passed into the DIABLO function (see *Appendix C, DIABLO Language Structure*).

The polynomial inductor value is computed during simulation as a function of the current through the inductor. When **I** is defined as the current through the inductor from **n₁** to **n₂** the inductance value is computed from:

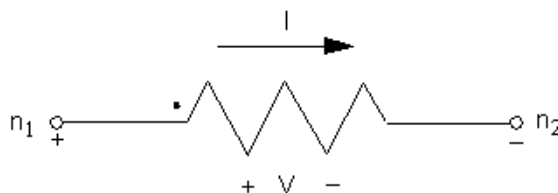
$$value = p_0 + p_1 \times I + p_2 \times I^2 + p_3 \times I^3 + etc$$

The high level descriptive language DIABLO is used for generating mixed discipline analog models. The language allows you to develop simulation models that can run under the VeriBest Analog (VBA) environment.

Example

```
LNONLIN 46 62 POLY 2PH 0.01 0.00035
```

Resistors



Formats

```

Rxxxxxxx n1 n2 value < TC = tc1 < tc2 << SCALE = > scale >>> < M = pval>
Rxxxxxx n1 n2 offset mname < L = lval >
+ < W = wval > < TC = tc1 < tc2 << SCALE = > scale >>> < M = pval >
Rxxxxxx n1 n2 value < TC = tc1 < tc2 << SCALE = > scale >>>
+ < AC = acval > < M = pval >
Rxxxxxx n1 n2 offset mname < L = lval >
+ < W = wval >< TC = tc1 < tc2<< SCALE => scale >>> <AC = acval><M = pval>
Rxxxxxx n1 n2 numsq <TC = tc1 < tc2 < rsheet > > >
+ < AC = acval > < M = pval >
Rxxxxxx n1 n2 offset mname < L = lval > < W = wval > <TC = tc1 < tc2
+ < rsheet > > > < AC = acval > < M = pval >

```

where:

Rxxxxxxx represents the unique resistor name.

n_1 , n_2 represent numbers or names (integers, alphanumerics) of the connecting nodes.

value represents the nominal resistance value (Ω). This value may be positive or negative but not zero.

numsq represents number of squares of area. Resistance is computed from $\text{numsq} * \text{rsheet}$.

TC indicates that temperature coefficients are to be specified.

tc_1 , tc_2 represent first and second-order temperature coefficients of the resistor (Default: $tc_1=0$, $tc_2=0$).

SCALE indicates that a scaling factor for computing the effective resistance from $\text{scale} * \text{value}$ will be specified.

scale scaling factor for computing the effective resistance from $\text{scale} * \text{value}$.

rsheet sheet resistance per square.

offset represents the offset resistance (Ω). It is used to calculate the resistance for the associated resistor .MODEL specification (see the section “Semiconductor Resistor Models” of this chapter).

mname represents the model name that references a specific resistor .MODEL instruction.

L indicates that a channel length will be specified.

lval	defines the channel length (m).
w	indicates that a channel width will be specified.
wval	defines the channel width (m).
AC	indicates that an AC value is specified.
acval	resistance value to be used during AC analysis (Default: acval=value).
M	indicates that a parallel multiplier value is to be specified.
pval	calculates the total resistance when there are pval resistors in parallel. The resistance value will be value/pval.

The resistance as a function of temperature is,

$$R(\text{temp}) = \text{value} \times (1 + tc_1 \times (\text{temp} - T_{\text{nom}}) + tc_2 \times (\text{temp} - T_{\text{nom}})^2)$$

where T_{nom} is the nominal temperature.

The multiplier can be used with resistors that reference a .MODEL instruction.

Examples

```
R1 1 2 100
RC1 12 17 1K TC=0.001,0.015
```

Semiconductor Resistor Models

Format

```
.MODEL mname R ( < pname = pval> < pname = pval> ... )
```

where:

.MODEL	indicates that model parameter are to be specified.
mname	represents the model reference name.
R	indicates a resistor model specification.
pname	represents a reserved parameter name as described in the parameter table.
pval	parameter value associated with a parameter name. The pname=pval pairs need not be enclosed in parentheses.

Resistor Model Parameters

Name	Parameter	Units	Default
TC1	first order temperature coefficient	$\Omega/^{\circ}\text{C}$	0.0
TC2	second order temperature coefficient	$\Omega/^{\circ 2}\text{C}$	0.0
RSH	sheet resistance	$\Omega/\text{sq.}$	0.0
DEFW	default width (not used)	m	1.0E-6
NARROW	narrowing due to side etching	m	0.0
PMAX	maximum power	W	0.0

Resistance is computed as follows:

$$R = \text{offset} + RSH \times \frac{L - NARROW}{W - NARROW}$$

Temperature effect is calculated as follows:

$$R(\text{temp}) = R(T_{nom}) \times [1 + tc_1 \times (\text{temp} - T_{nom}) + tc_2 \times (\text{temp} - T_{nom})^2]$$

Whether the temperature coefficient is specified on the device line or the model line, it has the same effect on the offset value. If different values are specified for the temperature coefficients on the device line and the model line, the values of the device line will override the values of the model line.

Switches

Formats

Sxxxxxxx n+ n- nc+ nc- mname < **ON/OFF** >

Wxxxxxxx n+ n- vname mname < **ON/OFF** >

where:

Sxxxxxxx / **W**xxxxxxx represents the unique switch names for the voltage/current controlled switches respectively.

n+, n- represent numbers or names (integers, alphanumerics) denoting the connecting nodes of the switch.

nc+, nc- represent numbers or names (integers, alphanumerics) denoting the positive and negative nodes of the controlling voltages respectively.

vname name of the voltage source through which the controlling current flows.

mname represents the model name that references a specific SWITCH .MODEL instruction.

ON/OFF indicates an initial ON or OFF condition for DC analysis.

Examples

```
S1 1 2 10 11 MOSSW ON
W2 10 21 VCLOCK CMOSSW
W3 11 23 VPHASE CMOSSW OFF
```

Switch Models

This model allows you to define an almost ideal switch. Since on and off resistance is available, the switch can have relatively infinite or close to zero resistance when compared to other circuit elements. A current or a voltage value controls the operation of the switch. When the controlling voltage or current increases past the threshold VT or IT respectively, the switch is considered *closed* with a resistance equal to RON. Conversely, the switch is considered *open* with a resistance of ROFF after the controlling voltage or current decreases under a threshold, VH or IH respectively.

Formats

```
.MODEL mname SW ( < pname = pval > < pname = pval > ... )
.MODEL mname CSW ( < pname = pval > < pname = pval > ... )
```

where:

.MODEL indicates that model parameters are to be specified.
mname represents the model name specified by the referencing SWITCH.
SW indicates a voltage controlled switch specification.
CSW indicates a current controlled switch specification.
pname represents a reserved parameter name as described next.
pval parameter values associated with a parameter name.

Example

```
.MODEL MOSSWITCH SW (VT=1.0 RON=10.0)
```

Switch Model Parameters

Name	Parameter	Units	Default
RON	On resistance	Ω	1.0
ROFF	Off resistance	Ω	1.0/GMIN
VT	Threshold voltage (SW only)	V	0.0
VH	Hysteresis voltage (SW only)	V	0.0
IT	Threshold current (CSW only)	A	0.0
IH	Hysteresis current (CSW only)	A	0.0

Transmission Lines

Formats

```
Txxxxxxx  n1i  n1r  n2i  n2r  Z0 = imped  TD = tdlay
+ < IC = v1, i1, v2, i2 >

Txxxxxxx  n1i  n1r  n2i  n2r  Z0 = imped  F = freq
+ < NL = length > < IC = v1, i1, v2, i2 >
```

where

Txxxxxxx represents the unique name of the transmission line. The Txxxxxxx component models only one propagation mode. To simulate two modes, two transmission lines are required.

n1i port 1 input node.

n1r port 1 reference node.

n2i port 2 input node.

n2r port 2 reference node.

Z0 indicates that the characteristic impedance is to be defined.

imped characteristic impedance of the transmission line.

TD indicates that the transmission line delay is to be defined.

tdlay transmission line delay (sec).

IC indicates that initial conditions are to be defined.

v₁ initial voltage across port 1.

i₁ initial branch current for port 1.

v₂ initial voltage across port 2 (Default: v₂=0.0).

i₂ initial branch current for port 2.

F indicates that the transmission-line frequency is to be specified.

freq transmission-line frequency (Default: freq=1.0E9).

NL indicates that the normalized length is to be specified.

length normalized transmission-line length. The line delay will be calculated from TD=length/freq (Default: length=0.25).

Very short transmission line delays compared to the analysis time will result in very long execution times.

The initial conditions are used only if VBASE does not perform a DC analysis. If a DC analysis is performed prior to the transient analysis, then the initial conditions will be computed from the DC analysis results. The conditions under which a DC analysis is performed are described in the .TRAN instruction.

Example

```
TRUN2 16 22 14 23 Z0=50 F=1.3E8 NL=.32
```

Uniform/Distributed RC Lines

Format

Uxxxxxxx n1 n2 n3 mname **L** = len < **N** = lumps >

where:

Uxxxxxxx represents the unique distributed RC name.

n1, n2 are two nodes connected by the RC line.

n3 is the common node to which capacitances or diodes are connected.

mname represents the model name that references a specific DRC/URC .MODEL instruction.

L indicates that a length value is specified.

len is the length of the line (meters).

N indicates that the number of lumps are specified.

lumps is the number of lumped segments for this line.

Example

U1 2 3 4 drcmod l=100u n=3

Distributed RC Models

The distributed RC model is derived from a model proposed by Levy Gerzberg in 1974. The distributed RC line is divided into 2N number of segments, where N is the number of lumps specified with the distributed RC device or with the distributed RC model. If N is not specified, it is computed using the following equation:

$$N = \frac{\ln \left[FMAX \times RPERL \times CPERL \times 2\pi \times len^2 \times \left(K - \frac{1}{K} \right)^2 \right]}{\ln(K)}$$

The RC value of the segments progresses geometrically toward the middle of the line, with K as a proportionality constant. Each segment contains resistors and capacitors, where one node of all the capacitors is connected to the common node. If you specify the model parameter ISPERL, then the capacitors are replaced by diodes, which will have a zero bias junction capacitance equal to the capacitor that was replaced. If you use diodes in place of capacitors, then you can specify any of the diode model parameters in the .MODEL instruction.

Format

.MODEL mname [**DRC** | **URC**] (< pnam = pval > < pnam = pval > ...)

where:

.MODEL indicates that model parameters are to be specified.

mname represents the model name specified by the referencing distributed RC devices.

DRC,URC indicates a uniform distributed RC line model specification. Either DRC or URC can be used.

pnam represents a reserved parameter name as described in the following pages.

pval parameter value associated with the parameter name. The pnam=pval pairs do not need to be enclosed in parentheses.

Example

```
.MODEL drcmod DRC k=1.5 rperl=10k cperl=10pf isperl=1ma
```

Distributed RC Model Parameters

Name	Parameter	Units	Default
K	Propagation constant	—	2.0
FMAX	Maximum frequency of interest	Hz	1.0G
RPERL	Resistance per unit length	Ω/m	1000
CPERL	Capacitance per unit length	F/m	1.0F
ISPERL	Diode saturation current per unit length	A/m	0.0
RSPERL	Diode resistance per unit length	Ω/m	0.0
TYPE	Type of diode connection. Diode cathode is connected to the common terminal if type = 1.0, and anode is connected to the common terminal if type = -1.0.	—	1.0
LUMPS	Number of lumps for this model. If not specified, it is computed.	—	—
TR1	First order temperature coefficient for resistance	$1/^\circ\text{C}$	0.0
TR2	Second order temperature coefficient for resistance	$1/(^\circ\text{C})^2$	0.0
TC1	First order temperature coefficient for capacitance	$1/^\circ\text{C}$	0.0
TC2	Second order temperature coefficient for capacitance	$1/(^\circ\text{C})^2$	0.0

Windings

Format

```
Yxxxxxxx n1 n2 bname nturn k < IC = icval >
```

where:

Yxxxxxxx is the name of the transformer winding.

n₁, n₂ represent numbers or names (integers, alphanumerics) of the connecting nodes.

pname is the name of the transformer core for this winding.

nturn is the number of turns in the winding.

k is the coupling constant between the core and the winding. The value has to be in the range $0 < k < 1$.

IC is an optional field indicating the initial current flowing from n_1 through the winding to n_2 . The initial conditions only apply if you specify the UIC option on the .TRAN statement.

icval defines the initial current flowing from n_1 through the winding to n_2 .

Example

```
.MODEL EXAM1 TFM FC1=2 DEL1=.5
PEX1 3 EXAM1 A=1.0E-4 LM=2.0E-2
YEX1 1 2 PEX1 4 .95
```

Chapter 4

Semiconductor Devices

The models for the semiconductor devices typically require many parameters for accurate specification. However, most devices in a circuit will have similar characteristics, and thus the same model parameters. To simplify the description, sets of model parameters are defined in separate .MODEL instructions, which are referenced by each semiconductor device specification.

Global default values for some geometric factors may be set with the .OPTIONS instruction. Other geometric factors associated with the individual devices can be specified on each device line.

All circuit parameters can have a statistical spread assigned to them. A definition of the statistical spread syntax below is found in *Chapter 1, Introduction*.

Formats

```
param name = value < STAT tol + :distname < STRAK = tol2 > < TRAK = x > >
```

```
param name = value < STAT min max + :distname < STRAK = tol2 > < TRAK = x > >
```

General .MODEL Specification

The .MODEL instruction specifies a set of model parameters that are referenced by one or more devices. Specific .MODEL instruction types are described in the section for each device type.

Format

```
.MODEL  mname  type  (  pname = pval  pname = pval...  )
```

where:

.MODEL indicates that model parameters are to be specified

mname represents the model reference name

type represents the device type as follows:

C	Semiconductor Capacitor
L	Inductor
R	Semiconductor Resistor
CSW	Current Controlled Switch
SW	Voltage Controlled Switch
D	Diode
URC	Uniform Distributed RC Line
DRC	Alternative name for URC model
NPN	NPN BJT
PNP	PNP BJT
NJF	N-channel JFET
PJF	P-channel JFET
NMOS	N-channel MOSFET
PMOS	P-channel MOSFET
NMES	N-channel MESFET
PMES	P-channel MESFET
TFM	Transformer Core
DIGITAL	Digital Model

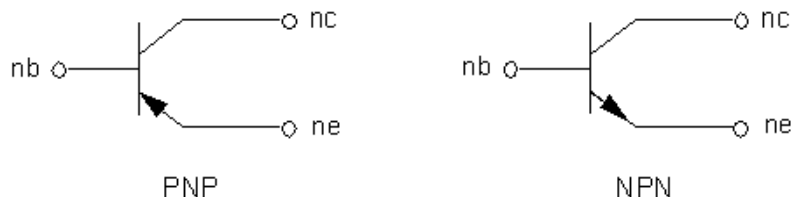
pname represents a parameter keyword

pval associated parameter value

Example

```
.MODEL DEPL NMOS (LEVEL=1 VTO=-4.0 KP=20U  
+ GAMMA=1.31 LAMBDA=0.01 PHI=0.6)
```

BJT Devices



Formats

```

Qxxxxxxx nc nb ne < ns > mname < area > < OFF >
+ < IC = vbe, vce >

Qxxxxxxx nc nb ne < ns > mname < area > < OFF >
+ < IC = vbe, vce > < M = pval >

Qxxxxxxx nc nb ne < ns > mname < AREA = area > < OFF >
+ < VBE = vbe > < VCE = vce > < M = pval >

```

where:

Qxxxxxxx represents a unique BJT name

nc, nb, ne collector, base and emitter nodes, respectively

ns substrate node name. If not specified, the ground node (0) is assumed (unless BULK= is specified).

mname represents the model name that references a specific NPN or PNP .MODEL instruction

AREA indicates that an area factor is specified

area base area factor. The base area factor scales several BJT parameters (see Parameter list). The default is area=1.0.

OFF indicates an initial OFF starting condition for DC analysis

IC indicates that an initial guess of the device voltages for DC analysis is specified

VBE, VCE indicates that an initial guess for the DC conditions is to be specified

vbe, vce define the initial guess for the DC conditions

M indicates that a multiplier factor to model transistors in parallel is to be specified. Affects all the currents, capacitances and resistances.

pval number of devices in parallel

Examples

```

Q23 10 26 4 20 MOD1
Q43B 11 21 13 QMOD IC=0.6,5.0

```

BJT Models

The bipolar junction transistor (BJT) model is an adaptation of the integral charge control model of Gummel and Poon. The original model has been extended to include high-bias-level effects. This model simplifies to the Ebers-Moll model when certain parameters are not specified.

The forward current gain characteristics of the DC model are defined by IS, BF, NF, ISE, IKF, and NE, and the reverse current gain characteristics are determined by IS, BR, NR, ISC, IKR, and NC. The output conductances for the forward and reverse regions are determined by VAF and VAR. Three ohmic resistances RC, RE, and RB are included, where RB can be current dependent. Base charge storage is modeled by the forward and reverse transit times, TF, and TR, with TF being bias-dependent if desired. The nonlinear depletion capacitances at the B-E junction is determined by CJE, VJE, and MJE. Similarly, B-C junction depletion capacitance is defined by CJC, VJC, and MJC; C-S (collector-substrate) junction depletion capacitance is related to CJS, VJS, and MJS.

The temperature dependence of saturation current IS is determined by the energy gap EG and saturation current temperature exponent XTI. The base current temperature dependence is modeled by the beta temperature exponent XTB.

Lateral geometry is assumed for PNP transistors. This is different from SPICE. Vertical geometry can be selected by using the model parameter SUBS or by using the options NOSUBS or SPICE in the .OPTIONS instruction.

Formats

```
.MODEL mname NPN ( < pnam = pval > < pnam = pval > ... )  
.MODEL mname PNP ( < pnam = pval > < pnam = pval > ... )
```

where:

.MODEL	indicates that model parameters are to be specified
mname	represents the model name specified on the referencing BJTs
NPN	indicates an NPN BJT model specification
PNP	indicates a PNP BJT model specification
pnam	represents a reserved parameter name as described in the following pages
pval	parameter values associated with a parameter name. The pnam=pval pairs do not need to be enclosed in parentheses.

Example

```
.MODEL QX_14A NPN (IS=2.3E-16 BF=205 RB=22OHMS)
```

BJT Model Parameters

An asterisk (*) indicates a parameter scaled by the AREA factor.

Name	Parameter	Units	Default
IS *	Transport saturation current	A	1.0E-16
LEVEL	BJT Model Selector (1=normal, 2=user-defined)	—	1
BF	ideal maximum forward beta	—	100(npn) 50 (pnp)
NF	Forward current emission coefficient.	—	1.0
VAF	Forward Early voltage	1/V	—
IKF *	Corner for forward beta high current roll-off	A	—
ISE *	B-E leakage saturation current	A	0.0
C2	B-E leakage saturation current coefficient	—	0.0
NE	B-E leakage emission coefficient	—	1.5
BR	Ideal maximum reverse beta	—	1.0 (npn) 10.0 (pnp)
NR	Reverse current emission coefficient	—	1.0
VAR	Reverse Early voltage	1/V	—
IKR *	Corner for reverse beta high current roll-off	A	—
ISC *	B-C leakage saturation current	A	0.0
C4	B-C leakage saturation current coefficient	—	0.0
NC	B-C leakage emission coefficient	—	2.0
RB *	Zero bias base resistance	Ω	0.0
IRB *	Current where base resistance falls halfway to its minimum value	A	0.0
RBM *	Minimum base resistance at high currents	Ω	RB
RE *	Emitter resistance	Ω	0.0
RC *	Collector resistance	Ω	0.0
CJE *	B-E zero bias depletion capacitance	F	0.0
VJE	B-E built-in potential	V	0.75
MJE	B-E junction exponential factor	—	0.33
TF	Ideal forward transit time	sec	0.0
XTF	Coefficient for bias dependence of TF	—	0.0
VTF	Voltage describing VBC dependence of TF	V	—
ITF *	High-current parameter for effect on TF	—	0.0
PTF	Excess phase at 1/(2*TF)Hz	—	0.0

Name	Parameter	Units	Default
CJC *	B-C zero bias depletion capacitance	F	0.0
VJC	B-C built-in potential	V	0.75
MJC	B-C junction exponential factor	—	0.33
XCJC	Fraction of B-C depletion capacitance connected to internal base node	—	1.0
TR	Ideal reverse transit time	sec	0.0
CJS *	Zero bias substrate junction capacitance	F	0.0
VJS	Substrate junction built-in potential	V	0.75
MJS	Substrate junction exponential factor	—	0.5
XTB	Forward and reverse beta temperature exponent	—	0.0
EG	Energy gap for temperature effect on IS	V	1.11
XTI	Temperature exponent for effect on IS	—	3.0
FC	Coefficient for forward-bias depletion capacitance formula	—	0.5
ISS *	Substrate junction saturation current	A	0.0
NS	Substrate junction emission coefficient	—	1.0
SUBS	Substrate connection selector		
	vertical geometry (NPN)	—	1
	lateral geometry (PNP)	—	-1
TRC1	Collector resistor first order temperature coefficient	1/°C	0.0
TRC2	Collector resistor second order temperature coefficient	1/(°C) ²	0.0
TRB1	Base resistor first order temperature coefficient	1/°C	0.0
TRB2	Base resistor second order temperature coefficient	1/(°C) ²	0.0
TRE1	Emitter resistor first order temperature coefficient	1/°C	0.0
TRE2	Emitter resistor second order temperature coefficient	1/(°C) ²	0.0
PMAX	Maximum power dissipation	W	—
KF	Flicker noise coefficient	—	0.0
AF	Flicker noise exponent	—	1.0

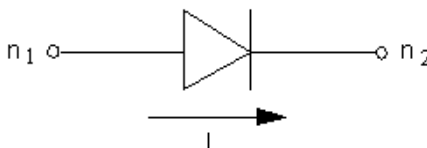
Usage Notes

If the options NOSUBS or SPICE are specified, then VBASE uses vertical geometry for NPN and PNP's. If these options are not specified and the model parameter SUBS is specified with a value of 1, then VBASE uses the vertical geometry for both NPN and PNP.

If the NOSUBS or SPICE options are not specified and the model parameter SUBS is specified with a value of -1, then VBASE uses lateral geometry for NPN and PNP.

In all other cases, VBASE uses vertical geometry for NPN and lateral geometry for PNP.

Diode Devices



Formats

```

Dxxxxxxx  n1  n2  mname  < area >  < OFF >  < IC = vd >
Dxxxxxxx  n1  n2  mname  < area  < jperi > >  < OFF >
+ < IC = vd > < M = pval >
Dxxxxxxx  n1  n2  mname  < AREA = area >  < PJ = jperi >  < OFF >
+ < IC = vd > < M = pval >

```

where:

Dxxxxxxx represents the unique name of the diode

n₁, n₂ anode and cathode nodes of the diode respectively. The diode will be forward-biased when the voltage at node n₁ is greater than at node n₂.

mname represents the model name that references a specific diode .MODEL instruction

AREA indicates that an area factor will be specified

area area factor value. The area factor scales the diode DC current IS, capacitance CJO, and resistance RS. The default is area=1.0

OFF indicates an initial OFF starting condition for DC analysis

IC indicates that an optional initial condition is to be specified

vd initial diode voltage. This initial condition can be overridden by a .IC instruction.

PJ indicates that the junction periphery is to be specified

jperi junction periphery value. It affects ISP and CJP. (Default: jperi=0.0)

M indicates that a multiplier factor is to be specified

pval number of diodes in parallel. It affects all currents, resistances, and capacitances.

Examples

```
dbridge 2 10 diodel
```

```
DCLMP 3 7 DMOD 3.0 IC=0.2
```

Diode Models

The DC characteristics of a diode are determined by the parameters IS, ISP, and N. An ohmic resistance, RS, is included.

Charge storage effects are modeled by a transit time, TT, and a nonlinear depletion layer capacitance determined by the area parameters CJO, VJ, and M, and periphery parameters CJP, VJP, and MP.

The temperature dependence of the saturation current is defined by the energy gap parameter EG and XTI, the saturation current temperature exponent.

Reverse breakdown is modeled by an exponential increase in the reverse diode current and is determined by the parameter BV.

Format

```
.MODEL mname D ( < pname = pval > < pname = pval >... )
```

where:

.MODEL	indicates that model parameters are to be specified
mname	represents the model name specified on the referencing diodes
D	indicates a diode model specification
pname	represents a reserved parameter name as described in the parameter table
pval	parameter value associated with a parameter name. The pname=pval pairs do not need to be enclosed in parentheses.

Example

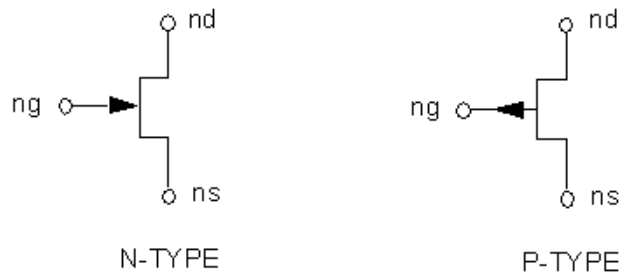
```
.MODEL JX123 D (IS=9.0E-15 N=1.02 CJO=1.3PF)
```

Diode Model Parameters

An asterisk (*) indicates a parameter scaled by the AREA factor. A pound sign (#) indicates a parameter scaled by periphery, PJ.

Name	Parameter	Units	Default
IS *	Saturation current per unit area	A	1.0E-14
ISP #	Sidewall saturation current per unit junction periphery	A/PJ	0.0
N	Emission coefficient for IS	—	1.0
NP	Emission coefficient for ISP	—	N
BV	Reverse breakdown voltage	V	infinite
RS *	Ohmic series resistance	Ω	0.0
CJO *	Zero bias junction capacitance per unit junction bottom wall area	F	0.0
M	Grading coefficient	—	0.5
FC	Coefficient for forward-bias depletion area capacitance formula	—	0.5
VJ	Junction potential for bottom wall	V	1.0
CJP #	Zero bias junction capacitance per unit junction periphery	F/PJ	0.0
MP	Grading coefficient for periphery	—	M
FCP	Coefficient for forward-bias depletion periphery junction capacitance formula	—	FC
VJP	Periphery junction potential	V	VJ
TT	Transit Time	sec	0
EG	Activation Energy	eV	1.11
XTI	Saturation-current temperature exponent (typically 2.0 for Schottky diodes)	—	3.0
TBV, TBV1	Temperature coefficient for breakdown voltage (linear)	1/°C	0.0
TBV2	Temperature coefficient for breakdown voltage (quadratic)	1/°C	0.0
TRS, TRS1	Resistance temperature coefficient (linear)	1/°C	0.0
TRS2	Resistance temperature coefficient (quadratic)	1/°C	0.0
CTA	Temperature coefficient for area junction capacitance	1/°C	0.0
CTP	Temperature coefficient for periphery junction capacitance	1/°C	0.0
PMAX	Maximum power dissipation	W	—
IBV	reverse breakdown current	A	0.0
KF	Flicker noise coefficient	—	0.0
AF	Flicker noise exponent	—	1.0
IKF	High-injection knee current	A	0.0
ISR	Recombination current parameter per unit area	A	0.0
NR	Emission coefficient for ISR	—	2.0
TIKF	IKF temperature coefficient	A	0.0
IKFP	Sidewall high-injection knee current	A	0.0
ISRP	Sidewall recombination current parameter per unit junction periphery	A	0.0
NRP	Emission coefficient for ISRP	—	NR
TIKFP	IKFP temperature coefficient	A	0.0

JFET Devices



Formats

```

Jxxxxxxx nd ng ns mname < area > < OFF > < IC = vds, vgs >
Jxxxxxxx nd ng ns mname < area > < OFF > < IC = vds, vgs >
+ < M = pval >
Jxxxxxxx nd ng ns mname < AREA = area >
+ < OFF > < IC = vds, vgs > < M = pval >

```

where:

Jxxxxxxx	represents the unique JFET name
nd, ng, ns	drain, gate and source nodes, respectively
mname	represents the model name that references a specific NJF or PJF .MODEL instruction
AREA	indicates that an area factor will be specified
area	area factor that affects BETA, RD, RS, CGS, and CGD
OFF	indicates an initial OFF starting condition for DC analysis
IC	indicates that an initial VDS and VGS are to be specified
vds, vgs	initial guess of the conditions to be used for DC analysis
M	indicates a multiplier factor specification
pval	indicates that pval JFETs are connected in parallel. It affects all currents, resistances and capacitances.

Example

```
J1 7 2 3 JM1 OFF
```

JFET Models

The Shichman-Hodges FET model is the basis for the JFET models. The DC characteristics are defined by the parameters VTO and BETA (which determine the variation of drain current with gate voltage), LAMBDA (which defines the output conductance), and IS (saturation current of the two gate junctions). Two ohmic resistances, RD and RS, are included. For both gate junctions, the charge storage is modeled by nonlinear depletion layer capacitances which vary as the inverse square root of the junction voltage. The nonlinear capacitances are defined by parameters CGS, CGD, and PB.

The threshold voltage VTO is negative for normally-on n-type JFETs and positive for normally-on p-type JFETs, and must be specified as such. This differs from SPICE, where both n- and p- type VTO values must be specified positive.

Formats

```
.MODEL  mname  NJF  ( < pnam = pval >  < pnam = pval >... )
.MODEL  mname  PJF  ( < pnam = pval >  < pnam = pval >... )
```

where:

.MODEL	indicates that model parameters are to be specified
mname	represents the model name specified by the referencing JFETs
NJF	indicates an n-type JFET model specification
PJF	indicates a p-type JFET model specification
pnam	represents a reserved parameter name as described in the following pages
pval	parameter value associated with a parameter name. The pnam=pval pairs do not need to be enclosed in parentheses.

Example

```
.MODEL JX23 NJF (VTO=-2V BETA=12U)
```

JFET Model Parameters

An asterisk (*) indicates a parameter scaled by the AREA factor.

Name	Parameter	Units	Default
VTO	Threshold voltage (negative for n-type) (positive for p-type)	V	-2.0
BETA*	Transconductance parameter	A/V ²	1E-4
BEX	Mobility temperature exponent correction for low field mobility	-	0.0
LAMBDA	Channel length modulation parameter	1/V	0.0
RD*	Drain ohmic resistance	Ω	0.0
RS*	Source ohmic resistance	Ω	0.0
CGS*	Zero bias G-S junction capacitance	F	0.0
CGD*	Zero bias G-D junction capacitance	F	0.0
PB	Gate junction potential	V	1.0
IS	Gate junction saturation current	A	1.0E-14
FC	Coefficient for forward-bias depletion capacitance formula	—	0.5
PMAX	Maximum power dissipation	W	—
KF	Flicker noise coefficient	—	0.0
AF	Flicker noise exponent	—	1.0
MJS	Source grading coefficient	—	0.5
MJD	Drain grading coefficient	—	0.5

MESFET Devices

Formats

```

Zxxxxxxx nd ng ns mname < AREA = val >
+ < OFF > < IC = vds, vgs > < M = pval >

Zxxxxxxx nd ng ns < nb > mname < W = val >
+ < L = val > < AD = val > < AS = val > < PD = val >
+ < PS = val > < NRD = val > < NRS = val > < NRG = val >
+ < NRB = val > < OFF > < IC = vds, vgs, vbs >
+ < M = pval >

Zxxxxxxx nd ng ns < nb > mname val val val ...
+ < OFF > < IC = vds, vgs, vbs > < M = pval >

```

where:

Zxxxxxxx	represents the unique name of the MESFET
nd, ng, ns	drain, gate, and source nodes, respectively
nb	bulk (substrate) node. You may specify this node in the .MODEL instruction. If you don't specify the bulk node, either here or in the .MODEL instruction, then the MESFET is assumed to be a three-terminal device.
mname	represents the model name that references a specific NMES or PMES .MODEL instruction
AREA	indicates that an area factor is specified
val	the specified parameter value
W, L	define the channel width and length (meters)
AD, AS	define the drain and source area (meters ²)
PD, PS	define the drain and source periphery factors (meters)
NRD, NRS, NRG, NRB	define the equivalent number of squares for computing the corresponding parasitic resistances from the sheet resistances
OFF	indicates an initial OFF starting condition for DC analysis
IC	indicates that an initial guess of the device voltages for DC analysis is specified
vds, vgs, vbs	specify the initial guess of the device voltages (volts)
M	indicates the multiplier factor which allows multiple devices to be connected in parallel
pval	number of devices in parallel

If you don't specify W, L, AD, AS, PD, PS, NRD, NRS, NRG, or NRB values, VBASE uses default values. The defaults for W and L are 1.0, and the defaults for the rest of the parameters are 0.0. You can change these defaults using the .OPTIONS instruction. Information about the .Options instruction can be found in Chapter 7: Instructions.

Example

```

Z1 4 5 3 gaas w=10

```

MESFET Models

VBASE provides various MESFET device models that differ in the formulation of the I-V and Q-V characteristics. The parameter LEVEL selects the model to be used, as follows:

LEVEL=1	RCA quadratic model
LEVEL=2	RCA cubic model
LEVEL=3	Raytheon model

The MESFET parasitics are represented by:

- parasitic resistances in series with each of the four nodes: nd, ng, ns, and nb (see the previous section, "MESFET Devices")
- constant parasitic capacitances between external drain and gate, external drain and source, and external source and gate nodes
- constant parasitic capacitances between internal drain and gate and internal drain and source and internal source and gate nodes
- constant capacitances between internal bulk and source and internal bulk and drain nodes

Additionally, VBASE includes the parasitic diodes between gate and source and gate and drain nodes with their junction saturation currents and variable capacitances. A series RC combination is included at the input and output to model the frequency dependence of the input and output impedances.

Formats

```
.MODEL mname NMES ( < pnam = pval > < pnam = pval > ... )  
.MODEL mname PMES ( < pnam = pval > < pnam = pval > ... )
```

where:

.MODEL	indicates that model parameters are to be specified
mname	represents the model name specified by the referencing MESFET devices
NMES	indicates an N-channel MESFET model specification
PMES	indicates a P-channel MESFET model specification
pnam	represents a reserved parameter name as described in the next page
pval	parameter value associated with the parameter name. The pnam=pval pairs do not need to be enclosed in parentheses.

Example

```
.MODEL gaas NMES vto=-2 beta=100u b=0.02
```


MESFET Model Parameters

An asterisk (*) indicates a parameter affected by scalm in the .OPTION instruction. A pound sign (#) indicates a parameter affected by AD or AS. A (@) sign indicates a parameter affected by W. A plus sign (+) indicates a parameter affected by PD or PS.

A dollar sign (\$) indicates a parameter affected by NR. An exclamation point (!) indicates a parameter affected by W*L. A percent sign (%) indicates a parameter affected by W/L.

Name	Parameter	Units	Default
LEVEL	MESFET model selector.	—	1
BULK	Bulk node assignment.	node	—
RD	Parasitic drain resistance.	Ω	0.0
RG	Parasitic gate resistance.	Ω	0.0
RS	Parasitic source resistance.	Ω	0.0
RB	Parasitic bulk resistance.	Ω	0.0
RSHDS\$	Parasitic sheet resistance for source and drain.	Ω/sq	0.0
RSHG\$	Parasitic sheet resistance for gate.	Ω/sq	0.0
RSHB\$	parasitic sheet resistance for bulk.	Ω/sq	0.0
TRD1	First order temperature coefficient for RD.	$1/^{\circ}\text{C}$	0.0
TRD2	Second order temperature coefficient for RD.	$1/^{\circ}\text{C}^2$	0.0
TRG1	First order temperature coefficient for RG.	$1/^{\circ}\text{C}$	0.0
TRG2	Second order temperature coefficient for RG.	$1/^{\circ}\text{C}^2$	0.0
TRS1	First order temperature coefficient for RS.	$1/^{\circ}\text{C}$	0.0
TRS2	Second order temperature coefficient for RS.	$1/^{\circ}\text{C}^2$	0.0
TRB1	First order temperature coefficient for RB.	$1/^{\circ}\text{C}$	0.0
TRB2	Second order temperature coefficient for RB.	$1/^{\circ}\text{C}^2$	0.0
TRSHDS1	First order temperature coefficient for RSHDS.	$1/^{\circ}\text{C}$	0.0
TRSHDS2	Second order temperature coefficient for RSHDS.	$1/^{\circ}\text{C}^2$	0.0
TRSHG1	First order temperature coefficient for RSHG.	$1/^{\circ}\text{C}$	0.0
TRSHG2	Second order temperature coefficient for RSHG.	$1/^{\circ}\text{C}^2$	0.0
TRSHB1	First order temperature coefficient for RSHB.	$1/^{\circ}\text{C}$	0.0
TRSHB2	Second order temperature coefficient for RSHB.	$1/^{\circ}\text{C}^2$	0.0
CGDPX	Parasitic capacitance between external g-d nodes.	F	0.0
CGSPX	Parasitic capacitance between external g-s nodes.	F	0.0
CGPX*@	Parasitic capacitance per unit width, between external g-s and g-d nodes.	F/width	0.0

Name	Parameter	Units	Default
CDSPX ^{*@}	Parasitic capacitance per unit width, between external d-s nodes.	F/width	0.0
CGDPI	Parasitic capacitance between internal g-d nodes.	F	0.0
CGSPI	Parasitic capacitance between internal g-s nodes.	F	0.0
CGPI ^{*@}	Parasitic capacitance per unit width, between internal g-s and g-d nodes.	F/width	0.0
CDSPI ^{*@}	Parasitic capacitance per unit width, between internal d-s nodes.	F/width	0.0
CBD	Parasitic capacitance between internal b-d nodes.	F	0.0
CBS	Parasitic capacitance between internal b-s nodes.	F	0.0
CB ^{*#}	Parasitic capacitance per unit area, between internal b-s and b-d nodes.	F/area	0.0
TCB1	First order temperature coefficient for bulk capacitance.	1/°C	0.0
TCB2	Second order temperature coefficient for bulk capacitance.	1/(°C) ²	0.0
CGD	Zero bias capacitance for g-d bottom junction diode.	F	0.0
CGS	Zero bias capacitance for g-s bottom junction diode.	F	0.0
CJ ^{*#}	Zero bias capacitance per unit area, for g-d and g-s bottom junction diode.	F/area	0.0
MJ	Grading coefficient for g-d and g-s bottom diode capacitances	—	0.5
PB	Junction potential for g-d and g-s bottom diode capacitances.	V	1.0
FCB	Forward bias junction capacitance coefficient for g-d and g-s bottom junctions.	—	0.5
CGDSW	Zero bias capacitance for g-d sidewall junction diode.	F	0.0
CGSSW	Zero bias capacitance for g-s sidewall junction diode.	F	0.0
CJSW ^{*#}	Zero bias capacitance per unit periphery, for g-d and g-s sidewall junction diode.	F/peri	0.0
MJSW	Grading coefficient for g-d and g-s sidewall diode capacitances.	—	0.5
PBSW	Junction potential for g-d and g-s sidewall diode capacitances.	V	1.0
FCBSW	Forward bias junction capacitance coefficient for g-d and g-s sidewall junctions.	—	0.5
TCJ1	First order temperature coefficient for g-d and g-s bottom junction capacitance.	1/°C	0.0
TCJ2	Second order temperature coefficient for g-d and g-s bottom junction capacitance.	1/(°C) ²	0.0
TCJSW1	First order temperature coefficient for g-d and g-s sidewall junction capacitance.	1/°C	0.0

Name	Parameter	Units	Default
TCJSW2	Second order temperature coefficient for g-d and g-s sidewall junction capacitance.	1/°C	20.0
IS	Bottom junction leakage current for g-d and g-s diodes.	A	1.0E-14
JS*#	Bottom junction leakage current per unit area, for g-d and g-s diodes.	A/area	0.0
ISSW	Sidewall junction leakage current for g-d and g-s diodes.	A	0.0
JSSW*+	Sidewall junction leakage current per unit periphery, for g-d and g-s diodes.	A/peri	0.0
N	Emission coefficient for g-d and g-s diodes.	—	1.0
BV	Reverse breakdown voltage for g-d and g-s diodes. This is specified as a positive number for both types of MESFETS. A value of zero indicates that breakdown is not to be modeled.	V	0.0
TTD	Transit time for g-d diode.	sec	0.0
TTS	Transit time for g-s diode.	sec	0.0
TT	Transit time for g-d and g-s diodes.	sec	0.0
EG	Activation energy for g-d and g-s diodes.	eV	1.42
XTI	Temperature exponent for leakage current for g-d and g-s diodes.	—	2.0
TBV1	First order temperature coefficient for g-d and g-s diode breakdown voltage.	1/°C	0.0
TBV2	Second order temperature coefficient for g-d and g-s diode breakdown voltage.	1/(°C) ²	0.0
RIN*@	Input series resistance per unit width.	Ω/width	0.0
CDGF*@	Feedback capacitance from g-d per unit width.	F/width	0.0
RDSS*@	Output series resistance per unit width.	Ω/width	0.0
CDSS*@	Output series capacitance per unit width.	F/width	0.0
CGDO	Zero bias g-d capacitance of the intrinsic MESFET.	F	0.0
CGSO	Zero bias g-s capacitance of the intrinsic MESFET.	F	0.0
CGO*!	Zero bias g-s and g-d capacitance per unit gate area, of the intrinsic MESFET.	F/area	0.0
M	Grading coefficient for g-d and g-s intrinsic capacitances.	—	0.5
VBI	Junction potential for g-d and g-s intrinsic capacitances.	V	1.0
FC	Forward bias junction capacitance coefficient for g-d and g-s intrinsic capacitances.	—	0.5
TCG1	First order temperature coefficient for g-d and g-s intrinsic capacitance.	1/°C	0.0

Name	Parameter	Units	Default
TCG2	Second order temperature coefficient for g-d and g-s intrinsic capacitance.	$1/(^{\circ}\text{C})^2$	0.0
CGDN	Normal bias g-d capacitance of the intrinsic MESFET.	F	0.0
CGSN	Normal bias g-s capacitance of the intrinsic MESFET.	F	0.0
CGN ^{*!}	Normal bias g-s and g-d capacitance, per unit gate area, of the intrinsic MESFET.	F/area	0.0
TCGN1	First order temperature coefficient for normal bias g-d and g-s intrinsic capacitance.	$1/^{\circ}\text{C}$	0.0
TCGN2	Second order temperature coefficient for normal bias g-d and g-s intrinsic capacitance.	$1/(^{\circ}\text{C})^2$	0.0
VTO	Pinch-off voltage. Proper sign must be used for N and P channel devices.	V	-2.0
K1	Pinch-off voltage increase (varies with square root of reverse bias on the bulk).	V	0.0
K2	Pinch-off voltage increase (proportional to reverse bias on the bulk).	—	0.0
TVT1	First order temperature coefficient for pinch-off voltage.	$1/^{\circ}\text{C}$	0.0
TVT2	Second order temperature coefficient for pinch-off voltage.	$1/(^{\circ}\text{C})^2$	0.0
TAU	Transit time under the gate.	sec	0.0
ALPHA	Saturation voltage parameter.	1/V	2.0
BETA [%]	Transconductance parameter.	A/V ²	1.0E-4
BETATEMP	Temperature exponent for BETA.	—	-1.5
DELTAEFF	Transition voltage range for charge model.	V	0.2
DELTA	Transition voltage range for charge model.	V	1/ALPHA
QOPT	Charge model selection. Four charge models are available. There is one charge model for the corresponding DC models of level 1, 2 and 3. The charge model corresponding to QOPT=4 is a charge conserving implementation of the level 3 charge model.	—	LEVEL
PMAX	Maximum power dissipation	W	—

Level 1

Name	Parameter	Units	Default
LAMBDA	Channel length modulation parameter.	1/V	0.0

Level 2

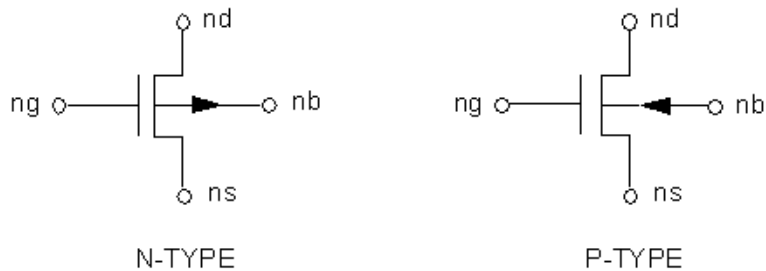
Name	Parameter	Units	Default
A0	Zero order coefficient in the cubic equation.	V^2	836.0
A1	First order coefficient in the cubic equation.	V	152.8
A2	Second order coefficient in the cubic equation.	—	-12.93
A3	Third order coefficient in the cubic equation.	1/V	-2.33
A5	Proportionality constant for transit time. If A5 is zero, transit time is specified by TAU.	sec/V	0.0
VDSO	VDS at which A0 through A3 are determined. Must be specified with proper sign for N and P channel devices.	V	6.0
GAMMA	Coefficient of pinch-off change.	1/V	0.0

Level 3

Name	Parameter	Units	Default
LAMBDA	Channel length modulation parameter	1/V	0.0
B	Effective doping parameter	1/V	0.0

Note: For the level 3 charge model, the interpretation of CGSO and CGDO is different from the one used in the Raytheon model. The model parameter CGDN is equivalent to the parameter CGDO used in the Raytheon model. Therefore, to get proper capacitance values, all model parameters CGSO, CGDO, CGSN and CGDN should be specified. For a symmetrical device, CGSO=CGDO and CGSN=CGDN. Alternatively, CGO and CGN should be specified, and the corresponding values of CGSO, CGDO, CGSN and CGDN will be computed for W and L of the device.

MOSFET Devices



Formats

```

Mxxxxxxx nd ng ns nb mname < L = val > < W = val >
+ < AD = val > < AS = val > < PD = val > < PS = val >
+ < NRD = val > < NRS = val > < OFF > < IC = vds, vgs, vbs >

Mxxxxxxx nd ng ns nb mname val val val ...
+ < OFF > < IC = vds, vgs, vbs >

Mxxxxxxx nd ng ns < nb > mname < L=val > < W = val >
+ < AD = val > < AS = val > < PD = val > < PS = val >
+ < NRD = val > < NRS = val > < OFF > < IC = vds, vgs, <vbs> >
+ < M = pval >

Mxxxxxxx nd ng ns < nb > mname val val val ...
+ < OFF > < IC = vdg, vgs, <vbs> > < M = pval >

```

where:

Mxxxxxxx	represents the unique name of the MOSFET
nd, ng, ns	drain, gate, and source nodes, respectively
nb	bulk (substrate) node
mname	represents the model name that references a specific NMOS or PMOS .MODEL instruction
val	the specified parameter value
L, W	define the channel length and width (meters) (see Notes 1, 2, 4, 5)
AD, AS	define the areas of the drain and source diffusions (square meters) (see Notes 1, 2, 3, 4)
PD, PS	define the perimeters of the drain and source junctions (meters) (see Notes 1, 2, 4)
NRD, NRS	define the equivalent number of squares for drain and source diffusion. It is used to compute drain and source resistance from RSH.
OFF	indicates an initial OFF starting condition for DC analysis
IC	indicates that an initial guess of the device voltages for DC analysis is to be specified
vds, vgs, vbs	specify the initial guess of the device voltages

M indicates the multiplier factor which allows for multiple devices in parallel
 pval number of devices in parallel

Examples

```
m1 24 2 0 20 type1
```

```
M31 2 17 6 10 MODM L=12U W=2U
```

```
M31 2 16 6 10 MODM 12U 2U
```

```
MAC3 2 9 3 0 MOD1 L=10U W=5U PS=40U AS=100P
```

```
mac3 2 9 3 mod1 10u 5u 2p 2p
```

Usage Notes

1. The parameters that follow mname may be specified in the following ways:
 As a sequence of XX=val assignments where the parameters can be specified in any order.
 As a sequence of numbers specified in the exact order shown, starting with the value for L.
2. If any of L, W, AD, AS, PD or PS values is not specified, default values are used. The user may change the values of these global default parameters using the .OPTIONS instruction. Information about the .OPTIONS instruction can be found in Chapter 7. Using defaults simplifies the preparation of the source file and changing the device geometries.
3. Since the AD and AS values are in square meters, the suffix P (1E-12) should be used rather than the suffix U (1E-6) to specify square micrometers.
4. The parameters L, W, AD, AS, PD, and PS are multiplied by the SCALE parameter on the .OPTION instruction.
5. The positions of W and L can be reversed on the MOSFET specification by using the WL option in the .OPTIONS instruction
6. VBASE does not treat parasitic source and drain diodes the same as SPICE. It is possible to set the parasitic source and drain junction currents to zero by setting IS = 0 and JS = 0 (default) on the .MODEL instruction, or IS = 0 on the .MODEL instruction and AD = 0 and AS = 0 (default) on the element (Mxxxxxx), whereas in SPICE they are always included. MOSFET parasitic junction capacitance is modeled in a manner consistent with SPICE.

MOSFET Models

VBASE provides various MOSFET device models that differ in the formulation of the I-V characteristic. The parameter LEVEL selects the model to be used as follows:

LEVEL=1	for MOS1, Shichman-Hodges model
LEVEL=2	for MOS2, an analytical model
LEVEL=3	for MOS3, a semi-empirical model
LEVEL=4	for BSIM, a short channel model
LEVEL=5	for BSIM2, a deep-submicron model
LEVEL=6	for ASPEC, an ASPEC compatible model
LEVEL=8	for MOS8, an enhanced MOS2 model
LEVEL=10	for BSIM3, a deep-submicron model
LEVEL=11	for CSIM, a short channel model
LEVEL=20	for EKV, MOSFET model

When installing User Models in VBASE, you can use the LEVEL parameter to select the appropriate User Model. If you don't specify the device parameters VTO, KP, PHI, or GAMMA, VBASE follows one of two procedures:

- calculates each parameter if either the NSUB or TOX process parameter is provided
- sets each parameter to its default value

VTO is negative for depletion mode N-channel devices. VTO is positive for depletion mode P-channel devices.

Charge storage is modeled by the following:

- three constant capacitors that represent overlap capacitances: CGSO, CGDO, and CGBO
- the nonlinear thin-oxide capacitance, distributed among the gate, source, drain, and bulk regions
- the nonlinear depletion-layer capacitances for both substrate junctions, divided into bottom and periphery. The depletion-layer capacitances vary directly as the MJ and MJSW power of junction voltage. They are determined by the parameters CJ, CJSW, MJ, MJSW, PB, PBSW, FC, CBS, and CBD.

The charge-conserving Yang-Chatterjee model is used to model the MOSFET channel capacitance. However, if you don't specify LEVEL=1 and the TOX parameter, VBASE does not model the channel charge. VBASE also provides the Meyer charge model and the Ward-Dutton charge model, the BSIM charge model, the ASPEC charge model, and a zero charge model.

You can describe the junction characteristics in multiple ways. For example, the reverse current can be set either using IS (amps) or JS (amps/m²). The first is an absolute value, whereas the second is multiplied by AD and AS (as given in the device element specification) to produce the reverse currents of the drain and source junctions respectively.

Formats

```
.MODEL mname NMOS ( < pname = pval > < pname = pval > ... )
.MODEL mname PMOS ( < pname = pval > < pname = pval > ... )
```


where:

.MODEL	indicates that model parameters are to be specified
mname	represents the model name specified by the referencing MOSFET
NMOS	indicates an N-channel MOS model specification
PMOS	indicates a P-channel MOS model specification
pname	represents a reserved parameter name as described in the following pages
pval	parameter value associated with a parameter name. The pname=pval pairs do not need to be enclosed in parentheses.

Example

```
.MODEL HMOS4 NMOS (LEVEL=3 VTO=1.2 KP=3.1E-6
+ GAMMA=.37 PHI=.63 CBD=15FF CBS=16FF IS=1.E-17)
```

MOSFET Model Parameters

An asterisk (*) indicates a parameter affected by scaln. A pound sign (#) indicates that you should see the Usage Notes at the end of this section.

Setup

Name	Parameter	Units	Default
LEVEL	MOSFET ids equation selector	—	1
BULK	Bulk node assignment	—	—
TLEV	ASPEC style temperature compensation choice	—	0.0
NUMDERIV	Use numerical derivatives instead of analytical derivatives for the MOSFET model (See .OPTION ABSDELTA and .OPTION RELDELTA)	—	0.0

Geometry

Name	Parameter	Units	Default
LD*	Lateral diffusion	m	0.0
LMLT	Length multiplier	—	1.0
DL (or XL)*	Correction added to L on the device card (except for BSIM where it is subtracted)	m	0.0
WD*	Lateral diffusion into channel from bulk along width	m	0.0
WMLT	Width multiplier	—	1.0
DEL	Channel length reduction on each side. DEL is not applicable to BSIM (LEVEL=4) $L_{\text{eff}} = L_{\text{drawn}} * LMLT + DL - 2 * (LD + DEL)$	m	0.0

Name	Parameter	Units	Default
DW (or XW)*	Correction added to the W on the device card (except for BSIMs where it is subtracted)	m	0.0
LDIF*	Length of lightly doped diffusion	m	0.0
HDIF	Length of heavily doped diffusion	m	0.0
ACM	ASPEC Compatibility Mode (= 1 \checkmark set)	—	1(level 6) 0

Stress Analysis

Name	Parameter	Units	Default
PMAX	Maximum power dissipation	W	—

Overlap Capacitances

Name	Parameter	Units	Default
CGBO*	Gate-bulk overlap capacitance per meter channel length. It is computed if not specified.	F/m	—
CGDO*	Gate-drain overlap capacitance per meter channel width. It is computed if not specified.	F/m	—
CGSO*	Gate-source overlap capacitance per meter channel width. It is computed if not specified.	F/m	—
FRINGE	Fringing field factor for G-S and G-D overlap capacitance calculation	m	0.0

Junction Capacitance (bottom)

Name	Parameter	Units	Default
CBS	Zero bias bulk-source junction capacitance	F	0.0
CBD	Zero bias bulk-drain junction capacitance	F	0.0
CJ	Zero bias bulk junction bottom capacitance	F/m ²	0.0 F/m (ACM=1)
MJ	Bulk junction bottom grading coefficient	—	0.5
PB	Diode bottom wall junction potential	V	0.8
FC	Forward-bias non-ideal junction capacitance coefficient	—	0.5
CTA	Temperature coefficient	1/°C	0.0
MJSW	Bulk junction sidewall grading coefficient	—	0.33
PBSW	Diode sidewall junction potential	V	PB
CTP	Temperature coefficient	1/°C	0.0

Junction Capacitance (Sidewall)

Name	Parameter	Units	Default
CJSW	Zero bias junction sidewall capacitance per meter of junction perimeter	F/m	0.0

Leakage Current (Bottom)

Name	Parameter	Units	Default
IS	Bulk junction saturation current	A	1.0E-14
JS*	Bulk junction saturation current per unit area	A/m ²	0.0 A/m (ACM=1)

Leakage Current (Sidewall)

Name	Parameter	Units	Default
ISSW	Sidewall junction saturation current	A	0.0
JSSW*	Sidewall junction saturation current per meter of junction perimeter	A/m	0.0

Parasitic Resistances

Name	Parameter	Units	Default
RSH	Drain and source diffusion sheet resistance	Ω/sq	0.0
RS	Source ohmic resistance	Ω	0.0
RD	Drain ohmic resistance	Ω	0.0
TRS	Temperature coefficient for drain and source diffusion resistance	1/ $^{\circ}\text{C}$	0.0
TRSH	Temperature coefficient for resistor	1/ $^{\circ}\text{C}$	0.0
TRD	Temperature coefficient for drain resistor	1/ $^{\circ}\text{C}$	0.0

Channel Capacitance

Name	Parameter	Units	Default
COX	Gate oxide capacitance	F/m ²	0.0
XQC	If .OPTION SPICE is not specified:		
	For the Ward-Dutton charge model (QOPT=2)		0.5
	For the BSIM charge model (QOPT=3)		0.0

Name	Parameter	Units	Default
	For all other charge models		1.0
	When the MOSFET model level is not equal to 4 (for example, when any MOSFET model other than BSIM is used), and when the BSIM charge model (QOPT=3) is used, then:		
	- XPC > 0.0 selects a 40/60 partition for drain/source charges in saturation		
	- XPC = 0.0 selects a 0/100 partition		
	If .OPTION SPICE is specified: XQC is used to determine the charge model. QOPT has no effect.		1.0
	- XQC ≤ 0.5 selects the Ward-Dutton charge model		
	- XQC > 0.5 selects the Meyer charge model for drain/source charges in saturation		
QOPT	Charge model selection	—	0.0
	0 - Yang-Chatterjee		
	1 - Meyer		
	2 - Ward-Dutton		
	3 - BSIM charge model		
	4 - ASPEC charge model		
	5 - Zero charge model		
	6 - BSIM2 charge model		
	7 - BSIM3 charge model (Only for BSIM3 MOSFET Device model)		
	8 - EKV model (Only for EKV MOSFET Device model)		
	This parameter is ignored with .OPTION SPICE or with .OPTION QOPT=n		

QOPT=4 Parameters

Name	Parameter	Units	Default
CF1	Transition from depletion to weak inversion	V	0.0
CF2	Transition from weak to strong inversion	V	0.1
CF3	Transition from saturation to linear	—	1.0
XCG	Capacitance multiplier	—	0.667

Threshold Voltage

Name	Parameter	Units	Default
VTO	Zero bias threshold voltage	V	0.0
NSS	Surface state density	1/cm ²	0.0
TPG	Type of gate material +1 opposite to substrate -1 same as substrate 0 aluminum gate	—	1.0
PHI	Surface potential for strong inversion. (For LEVEL=6, PHI=Fermi potential)	V	0.6
GAMMA	Bulk threshold parameter	V	0.0
NSUB	Substrate doping	1/cm ³	0.0
TCV	Temperature coefficient	V/°C	0.0

Noise

Name	Parameter	Units	Default
KF	Flicker noise coefficient	V ² F	0.0
AF	Flicker noise coefficient	—	1.0

Mobility

Name	Parameter	Units	Default
KP	Transconductance parameter	A/V ²	2.0E-5
UO	Surface mobility	cm ² /V-sec	600 (N-chl) 400 (P-chl)
BEX	Temperature exponent for KP	—	-1.5

Level 1

Name	Parameter	Units	Default
LAMBDA	Channel-length modulation	1/V	0.0
TOX	Gate oxide thickness	m	0.0

Level 2

Name	Parameter	Units	Default
LAMBDA	Channel-length modulation	1/V	0.0
DELTA	Width effect on threshold voltage	—	0.0
VMAX	Maximum drift velocity of carriers	m/sec	0.0
NEFF	Total channel charge coefficient	—	1.0
XJ	Metallurgical junction depth	m	0.0
UCRIT	Critical field for mobility degradation	V/cm	1.0E4
UEXP	Critical field exponent in mobility degradation	—	0.0
UTRA	Transverse field coefficient	—	0.0
TOX	Gate oxide thickness	m	1.0E-7

Level 3

Name	Parameter	Units	Default
ETA	Static feedback	—	0.0
DELTA	Width effect on threshold voltage	—	0.0
KAPPA	Saturation field factor	—	1.0
THETA	Mobility reduction	1/V	0.0
TOX	Gate oxide thickness	m	1.0E-7
XJ*	Metallurgical junction depth	m	0.0
VMAX	Maximum drift velocity of carriers	m/sec	0.0
NFS	Fast surface state density	1/cm ²	0.0

Level 4

Name	Parameter	Units	Default
VFB [#]	Flat-band voltage	V	0.0
PHI [#]	Surface inversion potential	V	0.6
K1 [#]	Body effect coefficient	√V	0.0
K2 [#]	Drain/source depletion charge sharing coefficient	—	0.0
ETA [#]	Zero bias drain-induced barrier lowering coefficient	—	0.0
MUZ	Zero bias mobility	cm ² /Vsec	600
DL [#]	Shortening of channel	m	0.0
DW	Narrowing of channel	m	0.0

Name	Parameter	Units	Default
U0 [#]	Zero bias transverse field mobility degradation coefficient	1/V	0.0
U1 [#]	Zero bias velocity saturation coefficient	m/V	0.0
X2MZ [#]	Sensitivity of mobility to substrate bias at Vds=0	cm ² /V ² sec	0.0
X2E [#]	Sensitivity of drain-induced barrier lowering effect to substrate bias	1/V	0.0
X3E [#]	Sensitivity of drain-induced barrier lowering effect to drain bias at Vds = Vdd	1/V	0.0
X2U0 [#]	Sensitivity of transverse field	1/V ²	0.0
X2U1 [#]	Sensitivity of velocity saturation effect to substrate bias	m/V ²	0.0
MUS [#]	Mobility at zero substrate bias at Vds = Vdd	cm ² /V ² sec	0.0
X2MS [#]	Sensitivity of mobility to substrate bias at Vds = Vdd	cm ² /V ² sec	0.0
X3MS [#]	Sensitivity of mobility to drain bias at Vds = Vdd	cm ² /V ² sec	0.0
X3U1 [#]	Sensitivity of velocity saturation effect on drain bias at Vds=Vdd	m/V ²	0.0
TOX	Gate oxide thickness	m	1.0E-7
VDD	Measurement bias range	V	0.0
XPART	Gate-oxide capacitance charge model flag XPART=0 selects a 40/60 drain source charge partition in saturation, while XPART = 1 selects a0/100 drain/source charge partition. XPART = 1 is recommended.	—	1.0
N0 [#]	Zero bias subthreshold slope coefficient	—	0.0
NB [#]	Sensitivity of subthreshold slope to substrate bias	—	0.0
ND [#]	Sensitivity of subthreshold slope to drain bias	—	0.0

Level 5

There are no default values for Level 5, all parameter values must be specified. For the level 5 MOS-FET model, parameters marked with a pound sign (#) also have corresponding parameters for their length and width dependency.

Name	Parameter	Units
VFB [#]	Flat-band voltage	V
PHI [#]	Strong inversion surface potential	V
K1 [#]	Bulk (body) effect coefficient	V ⁽⁻¹⁾
K2 [#]	Non-uniform channel doping coefficient	-
ETA0 [#]	Drain-induced barrier lowering at Vbs=0	-
ETAB [#]	Sensitivity of eta to Vbs	V ⁽⁻¹⁾
MU0 [#]	Low-field mobility; Vds=0, Vgs=Vth	cm ² /Vsec

MU0B [#]	Sensitivity of μ_0 to V_{bs}	cm^2/Vsec
MUS0 [#]	High-field mobility; $V_{ds}=V_{dd}$, $V_{gs}=V_{th}$	cm^2/Vsec
MUSB [#]	Sensitivity of μ_s to V_{bs}	cm^2/Vsec
MU20 [#]	Empirical parameter for output resistance	-
MU2B [#]	Sensitivity of μ_2 to V_{bs}	$1/V$
MU2G [#]	Sensitivity of μ_2 to V_{gs}	$1/V$
MU30 [#]	Empirical parameter for output resistance	$\text{cm}^2/V^2\text{sec}$
MU3B [#]	Sensitivity of μ_3 to V_{bs}	$\text{cm}^2/V^3\text{sec}$
MU3G [#]	Sensitivity of μ_3 to V_{gs}	$\text{cm}^2/V^3\text{sec}$
MU40 [#]	Empirical parameter for output resistance	$\text{cm}^2/V^3\text{sec}$
MU4B [#]	Sensitivity of μ_4 to V_{bs}	$\text{cm}^2/V^4\text{sec}$
MU4G [#]	Sensitivity of μ_4 to V_{gs}	$\text{cm}^2/V_4\text{sec}$
UA0 [#]	1st ord. vertical-field mobility reduction	$V^{(-1)}$
UAB [#]	Sensitivity of u_a to V_{bs}	$1/V^2$
UB0 [#]	2nd ord. vertical-field mobility reduction	$1/V^2$
UBB [#]	Sensitivity of u_b to V_{bs}	$1/V^3$
U10 [#]	Velocity saturation coefficient	$1/V$
U1B [#]	Sensitivity of u_1 to V_{bs}	$1/V^2$
U1D [#]	Sensitivity of u_1 to V_{ds}	$1/V^2$
N0 [#]	Subthreshold swing coefficient	-
NB [#]	Sensitivity of n to V_{bs}	\sqrt{V}
ND [#]	Sensitivity of n to V_{ds}	$1/V$
VOF0 [#]	V_{th} offset for subthreshold; $V_{ds}=0$ $V_{bs}=0$	-
VOFB [#]	Sensitivity of v_{of} to V_{bs}	$1/V$
VOFD [#]	Sensitivity of v_{of} to V_{ds}	$1/V$
AI0 [#]	Hot-electron coefficient; R_{out} degradation	-
AIB [#]	Sensitivity of a_i to V_{bs}	$1/V$
BI0 [#]	Hot-electron exponent; R_{out} degradation	V
BIB [#]	Sensitivity of b_i to V_{bs}	-
VGHIGH [#]	Upper bound of transition region	V
VGLOW [#]	Lower bound of transition region	V
VDD [#]	Drain supply voltage; maximum v_{ds}	V
VGG [#]	Gate supply voltage; maximum v_{gs}	V
VBB [#]	Body supply voltage; maximum v_{bs}	V

Level 6

Name	Parameter	Units	Default
DNS	Doping for LGAMMA computation	1/cm ³	0.0
LGAMMA	Multilevel threshold parameter. If LGAMMA is not specified, then LGAMMA is computed from DNS. When VBO is specified, then GAMMA is used if reverse bias on the bulk is less than VBO, and LGAMMA is used for larger reverse bias.	√V	0.0
GAMMA*	Junction depth, if VBO is not specified	m	0.0
VBO	Critical voltage for gamma switch	V	0.0
NWM	Narrow width modulation of gamma	—	0.0
SCM	Short channel modulation of gamma	1/V	0.0
VSH	Threshold voltage shift due to channel length	V	0.0
VFDS	Critical voltage for selection of FDS or UFDS. FDS is used if VDS < VFDS.	V	0.0
FDS	Field drain to source controls reduction of threshold due to source-drain electric field	—	0.0
UFDS	High field FD	—	0.0
NWE*	Narrow width effect on threshold voltage	—	0.0
KU	Velocity saturation switch. Alternate saturation model is used if KU > 1.	—	0.0
NU	Switch for mobility reduction due to lateral field. Mobility reduction is modeled if NU = 1.	—	1.0
KA	Short channel VDS scaling factor	—	1.0
MAL	Short channel exponent for VDS scaling factor	—	0.5
MBL	Short channel exponent for mobility reduction	—	1.0
NFS	Fast surface state density	1/cm ²	0.0
VMAX	Maximum drift field velocity of carriers. VMAX also determines the calculation scheme for saturation voltage. Use zero to indicate an infinite value.	cm/sec	0.0
TOX	Gate oxide thickness	Å	690
ECRIT	Critical lateral electric field	V/cm	0.0
MOB	Mobility equation selector	—	0.0
CLM	Channel length modulation equation	—	0.0
WIC	Weak inversion equation selector	—	0.0
WEX	Weak inversion equation exponent or WIC=2	—	0.0

MOB=1 Parameters

Name	Parameter	Units	Default
F1	Gate field mobility reduction	1/V	0.1
UTRA	Lateral field mobility reduction factor	1/V	0.0

MOB=2 Parameters

Name	Parameter	Units	Default
F1	Critical vertical field at which mobility reduction becomes significant	V/cm	0.0
UEXP	Mobility exponent	—	0.0
UTRA	Lateral field mobility reduction factor	1/V	0.0

MOB=3 Parameters

Name	Parameter	Units	Default
F1	Low-field mobility multiplier	1/V	0.0
UEXP	Mobility exponent	—	0.0
UTRA	High-field mobility multiplier	1/V	0.0
F4	Mobility summing constant	—	1.0
VF1	Critical voltage for low to high field multiplier switch	V	0.0

MOB=4 and MOB=5 Parameters

Name	Parameter	Units	Default
F1	Critical field for mobility reduction	V/cm	0.0
F2	Bulk mobility reduction factor	1/V ²	0.0
F3	Critical lateral field for mobility reduction (MOB=5)	V/cm	0.0
ECRIT	Critical lateral field for mobility reduction (MOB=4)	V/cm	0.0

CLM=1 Parameters

Name	Parameter	Units	Default
LAMBDA	Channel length modulation factor. If not specified, it is computed.	cm/ \sqrt{V}	0.0
KL	Empirical exponent.	—	0.0

CLM=2 Parameters

Name	Parameter	Units	Default
A1	First fringing field factor	—	0.2
A2	Second fringing field factor	—	0.6

CLM=3 Parameters

Name	Parameter	Units	Default
LAMBDA	Channel length modulation factor. If not specified, it is computed.	cm/ \sqrt{V}	0.0
KCL	Exponent for substrate bias scaling factor	—	0.0
MCL	Short channel exponent	—	1.0

CLM=4 Parameters

Name	Parameter	Units	Default
A1*	Junction depth	m	0.0
DND	Drain diffusion concentration	1/cm ³	1.0E20
KL	Grading coefficient exponent	—	0.333

Level 8

Name	Parameter	Units	Default
CAV	Thermal voltage multiplier for weak inversion equation	—	0.0
DELTA	Width effect on threshold voltage	—	0.0
ECRIT	Critical electric field for carrier velocity saturation. Typical values are 6000 for electrons and 24000 for holes. Use zero to indicate an infinite value.	V/cm	0.0
LAM1	Channel length modulation correction	1/m	0.0
LAMBDA	Channel length modulation factor	—	0.0
SNVB	Slope of doping concentration versus VBS	1/V*cm ³	0.0
UCRIT	Critical voltage for mobility degradation, if UEXP > 0.0	V/cm	0.0
UCRIT	Critical voltage for mobility degradation, if UEXP ≤ 0.0	1/V	0.0
UEXP	Critical field exponent for mobility degradation	—	0.0
UTRA	Transverse field coefficient for mobility degradation	1/V	0.0
TOX	Gate oxide thickness	m	1.0E-7

Level 10

Name	Parameter	Units	Default
TOX	Gate oxide thickness	m	1.5E-8
XJ #	Junction Depth	m	1.5E-7
NCH #	Channel doping concentration	1/cm ³	1.7E17
NSUB #	Substrate doping concentration	1/cm ³	6.0E16
NGATE #	Poly gate doping concentration	1/cm ³	0.0
K1 #	First-order body effect coefficient	√V	0.53
K2 #	Second-order body effect coefficient	—	-0.0186
K3 #	Narrow width coefficient		80.0
K3B #	Body effect coefficient of k3	—	0.0
DVT0W #	First coefficient of narrow width effect on Vth at small L	1/m	0.0
DVT1W #	Second coefficient of narrow width effect on Vth at small L	1/m	5.3E6
DVT2W #	Body-bias coefficient of narrow width effect on Vth at small L	1/V	-0.032
VBM#	Maximum applied body bias in Vth calculation	V	-3.0
W0 #	Narrow width parameter	m	2.5E-6
NLX #	Lateral non-uniform doping coefficient	m	1.74E-7
DVT0 #	First coefficient of short-channel effect on Vth	—	2.2
DVT1 #	Second coefficient of short-channel effect on Vth	—	0.53
DVT2 #	Body-bias coefficient of short-channel effect on Vth	1/V	-0.032
U0 #	Mobility at T-TNOM		
	NMOSFET	cm ² /(V·sec)	670.0
	PMOSFET		250.0
UA #	First-order mobility degradation coefficient	m/V	2.25E-9
UB #	Second-order mobility degradation coefficient	(m/V) ²	5.87E-19
UC #	Body-effect of mobility degradation coefficient	(m/V) ²	4.65E-11
		MOBMOD=1,2	-0.0465
		MOBMOD=3	1/V
VSAT #	Saturation velocity at T == TNOM	m/sec	8.0E4
A0 #	Bulk charge effect coefficient for channel length	—	1.0
AGS#	Gate bias coefficient of the Abulk	1/V	0.0
B0 #	Bulk charge effect coefficient for channel width	m	0.0

Name	Parameter	Units	Default
B1 #	Bulk charge effect width offset	m	0.0
KETA #	Body-bias coefficient of the bulk charge effect	1/V	-0.047
A1 #	First non-saturation factor	1/V	0.0
A2 #	Second non-saturation factor	—	1.0
RDSW #	Parasitic resistance per unit width	Ω/width	0.0
PRWG#	Gate bias effect coefficient of RDSW	1/V	0.0
PRWB #	Body-effect coefficient of RDSW	\sqrt{V}	0.0
WR #	Width offset from Weff for Rds calculation	—	1.0
WINT	Width offset fitting parameter from I-V without bias	m	0.0
LINT	Length offset fitting parameter from I-V without bias	m	0.0
DWG #	Coefficient of Weff's gate dependence	m/V	0.0
DWB #	Coefficient of Weff's substrate body bias dependences	m/ \sqrt{V}	0.0
VOFF #	Offset voltage in the subthreshold region at large W and L	V	-0.08
NFACTOR #	Subthreshold swing factor	—	1.0
ETA0 #	DIBL coefficient in subthreshold region	—	0.08
ETAB #	Body-bias coefficient for the subthreshold DIBL effect	1/V	-0.07
DSUB #	DIBL coefficient exponent in subthreshold region	—	DROUT
CIT #	Interface trap capacitance	F/m ²	0.0
CDSC #	Drain/Source to channel coupling capacitance	F/m ²	2.4E-4
CDSCD#	Drain-bias sensitivity of CDSC	F/m ²	0.0
CDSCB #	Body-bias sensitivity of CDSC	F/Vm ²	0.0
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	1/V	0.0086
PDIBLCB #	Body-effect coefficient of drain-induced barrier lowering correction parameters	1/V	0.0
DROUT #	L dependence coefficient of the DIBL correction parameter in Rout	—	0.56
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
DELTA #	Effective Vds parameter	V	0.01

Name	Parameter	Units	Default
ALPHA0 #	The first parameter of impact ionization current	m/V	0.0
BETA0 #	The second parameter of impact ionization current	V	30.0
MOBMOD	Mobility model selector	—	1
CAPMOD	Flag for the short channel capacitance model	—	2
VFBCV#	Flat band voltage parameter for CAPMOD=0	V	-1
XPART	Charge partitioning rate flag	—	0
CGSL #	Light doped source-gate region overlap capacitance	F/m	0.0
CGDL #	Light doped drain-gate region overlap capacitance	F/m	0.0
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
DLC	Length offset fitting parameter from C-V	m	LINT
DWC	Width offset fitting parameter from C-V	m	WINT
CF #	Fringing field capacitance	F/m	calculated
UTE #	Mobility temperature exponent	—	-1.5
KT1 #	Channel length sensitivity of temperature coefficient for threshold voltage	V	-0.11
KT2 #	Body-bias coefficient of the Vth temp effect	—	0.022
KT1L #	Temperature coefficient of Vth	Vm	0.0
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	m/V ²	-5.6E-11
		MOBMOD=1,2	-0.056
		MOBMOD=3	1/V
AT #	Temperature coefficient for saturation velocity	m/sec	3.3E4
PRT #	Temperature coefficient of parasitic resistance	Ωm/°C	0.0
LMIN	Minimum channel length	m	0.0
LMAX	Maximum channel length	m	1.0
WMIN	Minimum channel width	m	0.0
WMAX	Maximum channel width	m	1.0
WL	Coefficient of Length dependence for width offset	m ^{WLN}	0.0
WLN	Power of length dependence of width offset	—	1.0

Name	Parameter	Units	Default
WW	Coefficient of width dependence for width offset	m^{WWN}	0.0
WWN	Power of width dependence of width offset	—	1.0
WWL	Coefficient of Length and width cross term for width offset	$m^{WWN+WLN}$	0.0
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
LWN	Power of width dependence for length offset	—	1.0
LWL	Coefficient of length and width cross term for length offset	$m^{LWN+LLN}$	0.0
XT #	Doping depth	m	1.55E-7
VTH0 #	Zero bias threshold voltage	V	calculated
GAMMA1 #	Body-effect coefficient near the interface	\sqrt{V}	calculated
GAMMA2 #	Body-effect coefficient near the bulk	\sqrt{V}	calculated
PHI #	Surface potential at strong inversion	V	calculated
VBX #	Vbs at which the depletion width equals XT	V	calculated
NOIA#	Noise parameter A	—	(NMOS) 1.0E20 (PMOS) 9.9E18
NOIB#	Noise parameter B	—	(NMOS) 5.0E4 (PMOS) 2.4E3
NOIC#	Noise parameter C	—	(NMOS) -1.4E-12 (PMOS) 1.4E-12
EF#	Flicker frequency exponent for NOIMOD=2	—	1.0
EM#	Saturated field	V/m	4.1E7
NOIMOD	Noise model selector	—	1
ELM#	Elmore constant	—	5
NQSMOD	Flag for NQS model	—	0

Level 11

Name	Parameter	Units	Default
XJ [*]	Metallurgical junction depth	m	0.0
VTO [#]	Zero bias threshold voltage	V	0.0
GAMMA [#]	Bulk threshold parameter	\sqrt{V}	0.0
GAMMA2 [#]	Threshold voltage dependence on bulk bias	—	0.0
ETA [#]	Threshold voltage dependence on drain bias	1/V	0.0
UO [#]	Surface mobility	cm ² /Vsec	600
BETAO [#]	Zero field conductance coefficient (computed from UO, if not specified)	A/V ²	—
THETA1 [#]	Mobility reduction coefficient due to gate bias	1/V	0.0
THETA2 [#]	Mobility reduction coefficient due to drain bias	1/V	0.0
THETA3 [#]	Mobility reduction coefficient due to substrate bias	1/ \sqrt{V}	0.0
GAMMAFF	Substrate doping coefficient for modeling the substrate doping term in the drain current expression	—	1.0
FLGSUBTH	Flag to invoke the subthreshold current model. The five sub parameters that follow are used if FLGSUBTH is set to 1.0	—	0.0
SUBEXP [#]	Subthreshold exponent multiplier	—	1.0
SUBEXPB [#]	Effect of substrate bias on the subthreshold exponent multiplier	—	0.0
SUBEXPD [#]	Effect of drain bias on the subthreshold exponent multiplier	—	0.0
SUBMULT [#]	Subthreshold current multiplier	—	0.0
SUBLIMIT [#]	Subthreshold current limiting multiplier	—	1.0
LREF [*]	Length of the reference device	m	infinite
WREF [*]	Width of the reference device	m	infinite
TOX	Gate oxide thickness	m	1.0E-7

Level 20

Name	Parameter	Units	Default
XJ	Junction depth	m	0.1E-06
THETA	mobility reduction coefficient	1/V	0
UCRIT	longitudinal critical field	V/m	2.0
LAMBDA	depletion length coefficient		0.5
WETA	narrow channel effect coefficient		0.25
LETA	short channel effect coefficient		0.1
IBA	first impact ionization coefficient	1/m	0
IBB	second impact ionization coefficient	V/m	3.0E8
IBN	saturation voltage factor for impact ionization		
UCEX	longitudinal critical field temperature exponent		0.8
IBBT	temperature coefficient for IBB	1/K	9.0E-4
NQS	Non-Quasi-Static (NQS) operation switch		0
VTO	long-channel threshold voltage	V	0.0
DL	channel length correction	m	0
DW	channel width correction	m	0

Note: As VG, VTO is also referred to the bulk. DL and DW parameters generally have a negative value.

Usage Notes

The model parameters in the preceding tables that have the footnote mark # adjacent to them are functions of device dimensions for the level 11 model. Each of these parameters has three components. The reference component is represented by the parameter name, for example, VTO or THETA1. Dependence on the reciprocal of channel length or width is represented by a parameter whose name is formed by appending L or W to the reference parameter name, for example, VTOL, VTOW or THETA1L, THETA1W. The reference components are the parameters for a device having channel length of LREF and width of WREF.

If you don't specify them, the overlap capacitances CGSO, CGDO and CGBO are computed. This is different from SPICE.

For the level 4 and level 5 MOSFET models, parameters marked with a pound sign (#) also have corresponding parameters for their length and width dependency. For example, for parameter VFB (volts), the corresponding length and width dependency parameters are LVFB and WVFB (volt-meters). The formula to calculate the parameter based on the corresponding length and width dependency is:

$$P = P_0 + \frac{P_L}{L_{eff}} + \frac{P_W}{W_{eff}}$$

where:

$$L_{eff} = L - DL \quad W_{eff} = W - DW$$

For the level 10 MOSFET model, parameters marked with a pound sign (#) can have corresponding parameters for their length, width and square. For example, for parameter AT (m²/sec), the corresponding length, width and square dependency parameters are LAT (m²/sec), WAT (m²/sec) and PAT (m³/sec). The formula to calculate the parameter based on the corresponding length, width and square is:

$$P = P_0 + \frac{P_L}{L_{eff}} + \frac{P_W}{W_{eff}} + \frac{P_P}{L_{eff} \cdot W_{eff}}$$

Where L_{eff} , W_{eff} are defined in the MOSFET Model Level 10 (BSIM3) section of this manual.

The length and width dependency parameters are also affected by scalm.

MOSFET Parasitics

The computation of MOSFET parasitics is described in the following pages.

$$L_{eff} = L \times LMLT + XL - 2.0 \times (LD + DEL)$$

$$W_{eff} = W \times WMLT + XW - 2.0 \times WD$$

DEL is not used in the BSIM Model.

If CGSO or CGDO is not specified, then:

$$CGSO \text{ or } CGDO = COX \times (LD + FRINGE) \times W_{eff}$$

If CGBO is not specified, then:

$$CGBO = COX \times 2.0 \times WD \times L_{eff}$$

Sidewall Junction Capacitance (CJ0SW)

Let $WR = W \times WMLT + XW$

If $CJSW > 0.0$ then

If $HDIF > 0.0$ then

$$CJ0SW = CJSW \times 2.0 \times (WR + 2.0 \times HDIF)$$

else if $LDIF > 0.0$ then

$$CJ0SW = CJSW \times 2.0 \times (WR + 2.0 \times LR)$$

else

$$CJ0SW = CJSW \times (PSorPD)$$

else

$$CJ0SW = 0.0$$

Bottom Junction Capacitance (CJ0)

Let $LR = LD + LDIF$

If $CBS > 0.0$ AND $CBD > 0.0$ then

$$CJ0BS = CBS$$

$$CJ0BD = CBD$$

else if $HDIF > 0.0$ then

$$CJ0BS = CJ0BD = CJ \times WR \times HDIF$$

else if $LDIF > 0.0$ then

$$CJ0BS = CJ0BD = CJ \times WR \times LR$$

```

else
    CJ0BS = CJ * AS
    CJ0BD = CJ * AD

```

Bottom Diode Leakage Current (ISDIODE)

```

If JS > 0.0 then
If HDIF > 0.0 then
    ISDIODE = JS x WR x HDIF
else if LDIF > 0.0 then
    ISDIODE = JS x WR x LR
else if ( AS or AD ) > 0.0
    ISDIODE = JS x (AS or AD)
else
    ISDIODE = 0.0
else
    ISDIODE = IS

```

Sidewall Diode Leakage Current (ISSWDIODE)

```

If JSSW > 0.0 then
If HDIF > 0.0 then
    ISSWDIODE = JSSW x 2.0 X (WR + 2.0 x HDIF)
else if LDIF > 0.0 then
    ISSWDIODE = JSSW x 2.0 X (WR + 2.0 x LR)
else if ( PS or PD ) > 0.0
    ISSWDIODE = JSSW x (PS or PD)
else
    ISSWDIODE = 0.0
else
    ISSWDIODE = ISSW

```

Parasitic Resistances (RPAR)

```

Let R = RS or RD and NR = NRS or NRD
If HDIF > 0.0 then
If NR > 0.0 then

```

$$RPAR = NR \times RSH + \frac{LR \times R}{WR}$$

```

else

```

$$RPAR = \frac{HDIF \times RSH + LR \times R}{WR}$$

```

else if LDIF > 0.0

```

$$RPAR = NR \times RSH + \frac{LR \times R}{WR}$$

```

else if ( RSH * NR ) > 0.0 then

```

```

    RPAR = RSH x NR

```

```

else

```

```

    RPAR = R

```

Chapter 5

Digital Devices

VBASE is capable of performing transient analysis of circuits consisting of digital devices or a combination of analog and digital devices in a native analog mixed A/D simulation.

The built-in digital models (primitives) are limited to simulating switching characteristics such as propagation delays, rise/fall times, etc. Loading effects and driving capacity are modeled with additional circuit elements. These additional elements can be included by you in the circuit.

General .MODEL Specification

The .MODEL instruction specifies a set of model parameters that are referenced by one or more devices. Specific .MODEL instruction types are described in the section for each device type.

Format

```
.MODEL mname type ( pname=pval pname=pval ...)
```

where:

.MODEL indicates that model parameters are to be specified

mname represents the model reference name

type represents the device type as follows:

C	Semiconductor Capacitor
L	Inductor
R	Semiconductor Resistor
CSW	Current Controlled Switch
SW	Voltage Controlled Switch
D	Diode
URC	Uniform Distributed RC Line
DRC	Alternative name for URC model
NPN	NPN BJT
PNP	PNP BJT
NJF	N-channel JFET
PJF	P-channel JFET
NMOS	N-channel MOSFET
PMOS	P-channel MOSFET
NMES	N-channel MESFET
PMES	P-channel MESFET
TFM	Core
DIGITAL	Digital Model

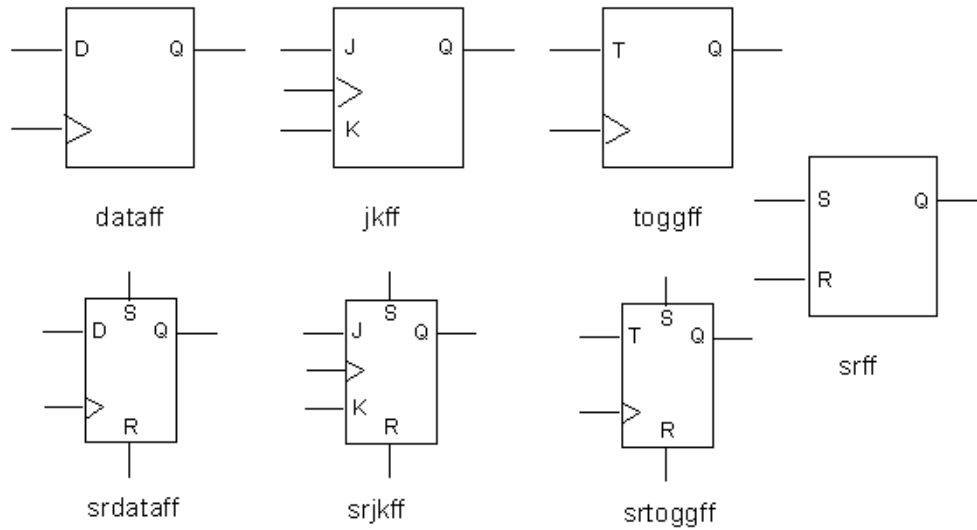
pname represents a parameter keyword as described in the parameter table

pval parameter value associated with a parameter name. The pname=pval pairs need not be enclosed in parenthesis

Example

```
.MODEL DEPL NMOS (LEVEL=1 VTO=-4.0 KP=20U
+GAMMA=1.31 LAMBDA=0.01 PHI=0.6)
```

Flip-Flop Devices



Formats

```

Axxxxxxx q type (n) clock <set> <reset> d mname IC=val ichold
Axxxxxxx q type (n) clock <set> <reset> j k mname IC=val ichold
Axxxxxxx q type (n) set reset mname IC=val ichold

```

where:

Axxxxxxx represents the unique flip-flop name

q the output node

type represents the function type keyword. Valid function types for flip-flops are:

Data flip-flops

dataff

sdataff

rdataff

srdataff

JK flip-flops

jkff

sjkff

rjkff

srjkff

Toggle flip-flops

toggff

stoggff

rtoggff

srtoggff

SR flip-flops

srf

srtff

n	the number of input nodes
clock	the clock input signal (active edge determined by the model parameter)
set	optional set input which asynchronously forces the output high (set overrides reset)
reset	optional reset input which asynchronously forces the output low
d	the input for DATA and TOGGLE flip-flops
j, k	the inputs for J-K flip-flops
mname	the model name
IC	indicates that an initial condition is to be specified. This is not optional. An initial condition must be specified.
val	the value of the initial condition (1 if on, 0 if off)
ichold	indicates whether the output is held at initial condition throughout the dc operating point calculation: 0 indicates false 1 indicates true

Example

```
AJKFF 6 JKFF(3) 1 2 3 FJK IC = 0 1
```

Flip-Flop Models

Format

```
.MODEL mname DIGITAL ( <pname = pval> <pname = pval>... )
```

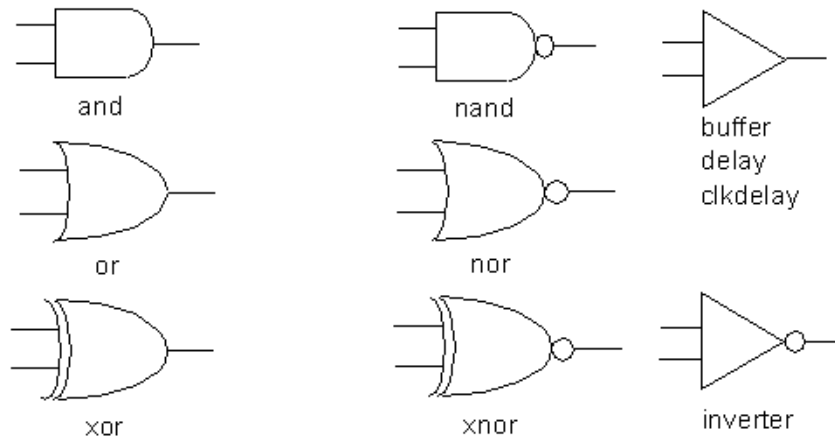
where:

.MODEL	indicates that model parameters are to be specified
mname	represents the model name specified by the digital device
DIGITAL	indicates that digital parameters are to be specified
pname	represents a parameter keyword as described in the parameter table
pval	parameter value associated with a parameter name. The pname=pval pairs need not be enclosed in parentheses.

Example

```
.MODEL FJK DIGITAL ENABLCLK = 3 INPUTFLG = 3 INPUTFLG = 5.0  
+ LOWLEV = 0.0 LHBITTRL = 2.0 HLBITTRL = 0.8
```


Gate Devices



Format

Axxxxxxx O type (n) I1 ... In mname IC=val ichold
where:

Axxxxxxx represents the unique gate names

O output node

type represents the function type keyword. Valid function types for gates are:

AND
NAND
OR
NOR
XOR
XNOR
BUFFER
DELAY
CLKDELAY
INVERTER

n the number of input nodes

I1 ... In input nodes 1 to n

mname the model name

IC indicates an initial condition is to be specified. This is not optional. An initial condition must be specified.

val the value of the initial condition (1 if on, 0 if off)

ichold indicates whether the output is held at initial condition throughout the dc operating point calculation:

0 indicates false
1 indicates true

Example

```
a2and 13 and(2) 1 2 andmod ic = 0 1
```

Gate Models

Format

```
.MODEL mname DIGITAL ( <pname = pval> <pname = pval>... )
```

where:

.MODEL	indicates that model parameters are to be specified
mname	represents the model name specified by the digital device
DIGITAL	indicates that digital parameters are to be specified
pname	represents a reserved parameter name as described in the parameter table
pval	parameter value associated with a parameter name. The pname=pval pairs do not need to be enclosed in parentheses

Example

```
.MODEL ANDMOD DIGITAL INPUTFLG = 3 HIGHLEV = 4.8 LOWLEV = 0.2  
+ LHBITTRL = 2.0 HLBITTRL = 0.8 LHDELAYT = 0.0 HLDELAYT = 0.0  
+ LHRISETM = 5E-9 HLRISETM = 5E-9
```

Digital Model Parameters

Input Voltage Transition Type

Name	Parameter	Units	Default
INPUTFLG	Type of transition: 0 single voltage level (BITTRNLV) 1 hysteresis state 3 single state calculated as an average (LHBITTRL+HLBITTRL)/2	—	0

Input Voltage Threshold

Name	Parameter	Units	Default
BITTRNLV	Bit transition level for INPUTFLG = 0 (1)	V	0.5
LHBITTRL	Low to High transition level (INPUTFLG = 1,2 or 3)	V	0.75
HLBITTRL	High to Low transition level (INPUTFLG = 1,2 or 3)	V	0.25

Propagation Delay

Name	Parameter	Units	Default
LHDELAYT	Low to High propagation delay time	sec	0.0
HLDELAYT	High to Low propagation delay time	sec	0.0

Output Voltage Transition Time

Name	Parameter	Units	Default
LHRISSTM	Low to High transition time	sec	1.0 E-9
HLRISSTM	High to Low transition time	sec	1.0 E-9

Output Voltage Level

Name	Parameter	Units	Default
HIGHLEV	logic 1 voltage level	V	1
LOWLEV	logic 0 voltage level	V	0

Clock Mode

Name	Parameter	Units	Default
ENABLCLK	Active State of flip flop clock input 0 - disabled, no clocking 1 - not used 2 - not used 3 - rising edge enabled 4 - falling edge enabled	---	3

Chapter 6

Subcircuits

Subcircuits provide a way to conveniently specify multiple identical circuit blocks. You can define a set of network components as a group, and then reference them repeatedly in a fashion similar to individual components. There is no limit on the size or complexity of subcircuits, and subcircuits may contain other subcircuits.

Subcircuits have two basic components:

- a subcircuit definition, in which the components and the topology of the subcircuit are defined, as well as the external node connections of the subcircuit block
- subcircuit expansions (sometimes referred to as calls or instantiations) which define how a subcircuit block is to be used in the circuit. Multiple expansions may be made of any subcircuit definition.

Note: Subcircuits do not reduce program execution time. Primarily, subcircuits provide simplified data specification.

Subcircuit Definitions

Formats

```
.SUBCKT name n1 < n2 < n3 < ... > > >
.....
xxxxxxxxxx
xxxxxxxxxx
.....
xxxxxxxxxx
xxxxxxxxxx
.ENDS <name>

.SUBCKT name n1 < n2 < n3 < ... > > > < parnam = pval ... >
.....
xxxxxxxxxx
xxxxxxxxxx
.....
.....
xxxxxxxxxx
xxxxxxxxxx
.ENDS <name>
```

where:

.SUBCKT	indicates the beginning of a subcircuit definition.
name	represents the subcircuit name. This name must be different than any other sub circuit name.
n ₁ , n ₂ , n ₃	represent the numbers or names (integers, alphanumerics) of the external nodes of the subcircuit. These node names must be distinct and must not include the ground node. There is no limit on the number of external node names.
xxxxxxxxxx xxxxxxxxxx	represent the component (model) description lines that define the subcircuit topology
parnam = pval	represents a parameter name set to a value for use only in the subcircuit, overridden by an assignment in the subcircuit call or by a value set in a .PARAM instruction.
.ENDS	indicates the end of a subcircuit definition.
name	(optional) should be the same as the name of a preceding and unterminated subcircuit definition. It indicates the end of that definition, including the subcircuit definitions nested within that subcircuit.

Examples

Example 1

```
.SUBCKT OPAMP 1 2 3 4
.SUBCKT ADDR 10 20 30 STND = 3U
...
.ENDS
```

Example 2

```
.SUBCKT COUNTER 1 2 3 4 TD = 10N TP = 100N
"C = TD/100K"
"TD1 = TP/10"
COUT 100 101 "C"
X11 1 2 3 4 FFLOP TDH = "TD1"

.SUBCKT FFLOP 1 2 3 4 TDH = 100N
...
...
...
.ENDS FFLOP
.ENDS COUNTER
```

Usage Notes

If name is omitted on a .ENDS instruction, all subcircuits currently being defined are terminated. Name is needed only for nested unterminated subcircuit definitions when it is not intended to terminate all of them.

Subcircuit definitions may contain other subcircuit definitions, device models, and subcircuit expansions. However, instructions (other than .MODEL) may not appear within a subcircuit definition.

Any device models or subcircuit definitions included in a subcircuit definition are strictly local to the definition. That is, they are not known outside the subcircuit definition and refer to entities different from external uses of the same model and subcircuit names.

The use of any model or subcircuit name that is not defined locally is assumed to refer to an external model or sub-circuit. The model/subcircuit is found by looking for a local definition of the name in a sequence of subcircuit definitions, starting with the one containing the use of the name. The search continues outward until either the name is found or there is no enclosing subcircuit.

Node names not included among n_1, n_2, n_3, \dots nodes on the .SUBCKT line are local. However, naming a node is considered part of the definition, with the result that no external node names are directly available within the sub-circuit. Ground (0) is the only exception: it is always global. External node names are available indirectly by using the replacement name from the subcircuit expansion.

You can use the .GLOBAL instruction to declare that certain nodes will have the same meaning both inside and outside subcircuits. Global specification is convenient for connecting power supplies to nodes within subcircuits.

Subcircuit Expansions

Formats

```

Xzzzzzzzz n1 < n2 < n3 < ... > > > name
+ <M = multiplier>

Xzzzzzzzz n1 < n2 < n3 < ... > > > name
+ < parnam=pval ... > < M = multiplier>

```

where:

Xzzzzzzzz	represents the pseudo-name of the subcircuit expansion
n_1, n_2, n_3	represent the connecting nodes for the expansion. The number of node names in the expansion should be the same as in the definition. The names in the expansion are paired in the order of occurrence with the node names in the subcircuit definition line. The expansion names are then substituted for those names within the definition, except where they are used within a nested subcircuit definition.
name	represents the name of the subcircuit being called
parnam=pval	represents a parameter name set to a value for use only in the subcircuit. This assignment overrides an assignment in the subcircuit definition, but is overridden by a value set in a .PARAM instruction.
M = multiplier	represents the number of multiple subcircuits in parallel. These subcircuits share the same nodes as the boundary of the subcircuit definition.

Assuming that the expanded subcircuit makes sense, node names in the subcircuit call do not need to be distinct from names external to the subcircuit expansion nor do they need to be non-zero.

Example

```
X1 2 4 17 3 2 1 MULTI
```


Subcircuit Functions

You can define subcircuit element values using functions. You can use algebraic and standard functions of global parameters and parameters passed to subcircuits to define element and model parameter values. Constants you define in the top-level circuit are passed as parameters and used inside the subcircuit to evaluate the function. Enclose these functions in double quotes (" ").

There is no restriction on the number of variables in each subcircuit definition block.

Formats

```
"namei = F ( arg1, ...argn, name1, ... namei-1 )"
"F ( arg1, ...argn, name1, ... namei-1 )"
"namei = LF ( arg1, ...argn, name1, ...namei-1 ) ?
F1 ( arg1, ...argn, name1, ...namei-1 ) :
F2 ( arg1, ...argn, name1, ...namei-1 )"
"LF ( arg1, ...argn, name1, ...namei-1 ) ?
F1 ( arg1, ...argn, name1, ...namei-1 ) :
F2 ( arg1, ...argn, name1, ...namei-1 )"

```

where:

F, F1, F2 represents any arithmetic function. These functions can include the following:

Operands:	constants parameters
Delimiters:	{ } ()
Continuation:	+ (if 1st character in line)
Arithmetic Operators:	+ add - subtract * multiply / divide ** exponential
Functions:	abs: absolute value sqrt: square root ln: natural log function log10: base 10 log function exp: exp(x) is equal to e ** x sin: sin function cos: cosine function tan: tangent function asin: arcsin function acos: arccos function atan: arctan function sinh: hyperbolic sin function cosh: hyperbolic cosine function tanh: hyperbolic tangent function

LF represents any logical-arithmetic functions. A logical-arithmetic function is a function which may have all the operations of an arithmetic function as well as the logicals and relationals.

Logicals:	&& and or
Relationals:	> greater than < less than >= greater than or equal to <= less than or equal to == equal to

`arg1, ...argn` are arguments. There are two kinds of args:

<code>.PARAM</code>	<ul style="list-style-type: none"> PARAM defined parameters parameters defined on the subcircuit definition. The actual values may be passed by subcall invocation.
---------------------	---

<code>name<i>i</i></code>	the name of the <i>i</i> th arithmetic function defined
---------------------------	---

Function formats 3 and 4 have the same meaning as Conditional Operators (`? :`) in the C language (for example, `condition?true:false`).

Functions can appear anywhere in the subcircuit as well as in the nominal circuit. In the nominal circuit all the parameters must be `.PARAM` parameters or the name of functions defined in previous lines of the input. In this case, `namei` is global and is added to a `.PARAM` list and can be used in any subcircuit.

A function is written in free format, which means that blanks or tabs can appear anywhere. A line may be continued by putting a plus sign (+) as the first character in the following line as a part of the netlist card; otherwise, it is part of the free format function.

Examples

Example 1

```
.
.PARAM ST=7
.
"STAM=0.25 * COS(ST/7-1.)"
.
.
X1 3 2 4 HE1 PPD=6U
.
.SUBCKT HE1 1 2 3 THICK=9P PPD=5U
.
"PM=PPD > 16.U*cos(PPD) ? 5.U : 1./2. *PPD"
"PL=2*(0.25*PPD + 0.5*PM)*
+ (log(exp(cosh(STAM - 0.25))))"
M1 1 2 3 THE W=PL L=" -0.5U*3. + 1.5*PM"
+ AS=THICK AD=9P PS=PL 0 PD=PPD
```

Example 2

```
.SUBCKT SUB1 1 2 3 P1=2U P2=3U  
M1 1 2 3 4 MOD L=P1 W=P2
```

```
.SUBCKT SUB1 1 2 3 P1=2U P2=3U  
M1 1 2 3 4 MOD L="P1" W="P2"
```

In this example, the first format runs faster, but both formats yield the same results.

Chapter 7

Instructions

Instructions are special data lines VBASE uses to define job control and other non-topological information. You define instructions with reserved instruction keywords that begin with a period (for example, .ALTER, .GLOBAL, and .OP).

The .MODEL instructions are described in *Chapter 4, Semiconductor Devices*, since they are closely associated with the device, *Chapter 3: Passive Elements*. The .SUBCKT and .ENDS instructions are described in *Chapter 6, Subcircuits*. For completeness, the .MODEL, .SUBCKT and .ENDS instructions are described briefly in this chapter.

Within subcircuit definitions, only .MODEL, .IC and .NODESET instructions and other .SUBCKTENDS instructions can be used.

VBASE Instructions

For each instruction there are a number of data specification formats. These formats are shown under the general header “Formats.”

Most instructions have two types of formats, those that are SPICE-compatible and those that are enhanced VBASE. The SPICE-compatible formats are shown first.

Some instructions do not exist in SPICE; therefore all of the formats shown for these instructions are VBASE specific. The following instructions are those that are VBASE specific:

- .ACQUIRE
- .DELETE (used in accord with .ALTER)
- .DISTR
- .DUMP
- .ENDDEL
- .ENDFUNC (see *Appendix C, The DIABLO Language Structure*)
- .EOM (an alias for .ENDS)
- .FUNC (see *Appendix C, The DIABLO Language Structure*)
- .GLOBAL
- .INCLUDE
- .LIB
- .MACRO (an alias for .SUBCKT)
- .MONTE
- .PARAM
- .RESTART
- .SEQUEL
- .VARY

The general SPICE instructions are not listed separately, but are included in alphabetical order within the following instructions.

.AC

The .AC instruction requests that VBASE perform a linear AC analysis with the specified frequency points.

Formats

```
.AC DEC ndec fstart flast
.AC OCT noct fstart flast
.AC LIN npts fstart flast
```

where:

.AC	indicates that a linear AC analysis is to be performed
DEC	requests decade frequency variation
ndec	number of points-per-decade
OCT	requests octave frequency variation (octaves are produced by dividing the difference of flast and fstart by 8)
noct	number of points per octave
LIN	requests linear frequency variation
npts	number of total points
fstart	starting frequency. fstart may be parameterized
flast	final frequency. flast may be parameterized

For AC analysis to be meaningful, at least one source must be specified with an AC value.

Examples

```
.AC DEC 10 1 10K
.AC DEC 10 1K 100MEG
.AC LIN 100 1HZ 100HZ
.AC DEC 10 LOWFREQ HIFREQ
```

.ACQUIRE

The .ACQUIRE instruction defines the output variables VBASE stores in a disk file for graphic display by the post-processor. The output values are stored at the computed time points rather than interpolated values. You can define up to eight output variables in one statement, and there is no limit to the number of acquire statements.

Format

```
.ACQUIRE  analysis-type  out-var1  <out-var2<...<out-varn>>>
```

where:

analysis-type is the type of analysis. Legal values for this field are DC, AC, or TRAN.

out-var1, are output variables. You can specify up to eight ($n \leq 8$) output variables. Output
out-var2 variables take different formats depending on the type of analysis and the variable type. See the .PLOT statement for the formats used to specify output variables.

Examples

```
.ACQUIRE  TRAN  V(4)  I(VIN)
```

```
.ACQUIRE  AC   VM(4,2)  VR(7)  VP(8,3)
```

```
.ACQUIRE  DC   V(2)  I(VSRC)  V(23,17)  PI (RI)
```

```
.ACQUIRE  TRAN  I(M11)  I1(Q2)  I2(Q2)  I3(Q2)  I(RI) + I(CAP)  PA(M11)  
PI(Q2)
```

.ALTER

The .ALTER instruction allows the program to be rerun with altered components and/or parameters. .ALTER also allows the user to delete components or to alter the network topology. See Usage Notes for more detail.

Format

```
.ALTER <TOPO>
...
...
Element, device, or model lines
...
...
.ALTER
...
```

where:

TOPO	indicates that subsequent component descriptions will modify the topology of the circuit. See the Usage Notes accompanying this instruction for more detail.
Element, device, or model lines	describe the components, node numbers, names, and values that will be re-arranged, added, changed, or deleted to alter the network topology

Examples

Example 1

```
R1 1 0 5K
VCC 3 0 10V
M1 3 2 0 0 MOD1 L=10U W=10U
.MODEL MOD1 NMOS (VTO=1.2 KP=2.0E-5 PHI=0.6
+ NSUB=5.0E15)
.ALTER
R1 1 0 3.5K
.MODEL MOD1 NMOS (VTO=0.8 KP=2.0E-5 PHI=0.75
+ NSUB=5.0E15)
M1 3 2 0 0 MOD1 L=10U W=2U
.ALTER
M1 3 2 0 0 MOD1 L=10U W=4U
.END
```

Example 2

In this example, capacitor C1 is deleted and replaced with two resistors in series, R2 and R3, using the TOPO option of .ALTER.


```
V1  10  0  PWL  0N  0V  5N  5V
R1  10  20  1K
C1  20  0  0.05P

.ALTER  TOPO

C1
R3  30  0  5K
R2  20  30  2K

.END
```

Usage Notes

The first .ALTER line ends the input file and causes a simulation run to be executed (that is, it acts like a .END). The lines following this .ALTER and preceding the next .ALTER (or .END) are then used to replace the parameters on the corresponding lines and a new simulation is performed. This process repeats until a .END is encountered. Subsequent .ALTER statements perform simulations using parameters of the previous changes and the changes requested by the new lines.

.ALTER uses the component names (Rxxx, Cxxx, etc.) and .MODEL names (.MODEL xxxxx) to define which components and/or model parameters are to be changed.

The TOPO option allows subsequent component descriptions to modify the topology of the circuit. That is, node numbers or names may be rearranged, new components may be added, and existing components deleted. Components are deleted by specifying only the component name (for example, C1234) without any subsequent data.

Input options for .OPTION that are related to device geometries will not be affected by .ALTER changes to a subsequent .OPTION instruction. For example:

```
.OPTION  DEFL=2U
.ALTER
.OPTION  DEFL=3U
```

In the preceding example, the default length of MOSFET devices will remain 2um.

.DC

The .DC instruction causes the program to perform successive DC analyses while sweeping the values of either one or two voltage or current sources. Parameter values and temperature may also be swept.

Formats

```
.DC srcnam start stop incr < src2 start2 stop2 incr2 >
.DC < xxxxx.> srcnam start stop incr < x xstart xstop xincr >
.DC TEMP tstart tstop tincr < x xstart xstop xincr >
.DC param pstart pstop pincr < x xstart xstop xincr >
.DC DEV devname pname pstart pstop pincr <lin>
+ < x xstart xstop xincr <list>>
.DC DEV devname pname pstart pstop pincr <log>
+ < x xstart xstop xincr <list>>
.DC semidev mname pname pstart pstop pincr <lin>
+ < x xstart xstop xincr <list>>
.DC semidev mname pname pstart pstop pincr <log>
+ < x xstart xstop xincr <list>>
```

where:

.DC	requests a DC transfer curve analysis
srcnam	represents the name of an independent voltage or current source
start	starting value of the source
stop	final value of the source
incr	incrementing value of the source
src ₂	optionally specifies a second source name and associated incrementing parameters.
start ₂	If specified, the first source will be swept over its range for each value of the second
stop ₂	source. This can be useful for defining transistor characteristics.
incr ₂	
xxxxx	represents a subcircuit expansion call name
TEMP	indicates a temperature sweep
tstart	starting temperature
tstop	final temperature
tincr	temperature increment
param	any parameter name for a sweep variable. The parameter name may not start with a V or I.
pstart	starting parameter value
pstop	final parameter value

pincr	parameter value increment. The default display is <lin> which produces a linear chart. If <log> is specified, a logarithmic chart is generated where the pincr number is the number of points per decade.
DEV	indicates that a device parameter is being changed
devnam	name of the device to sweep.
pname	name of the device or model parameters to sweep.
semidev	indicates that a device model parameter is being changed. semidev must be one of the following keywords: MOS, BJT, DIODE or JFET.
mname	name of the model to sweep.
<lin>	specifies that the analysis generates a linear table of pincr points
<log>	specifies that the analysis generates a logarithmic table of pincr points per decade
<list>	specifies that the analysis generates a table using an explicitly defined list with up to eight values specified
x	optional second source, temperature, parameter, device or model sweep specification
xstart	
xstop	
xincr	

Examples

```
.DC VIN 0.25V 5.0V 0.25V
.dc vds 0 10 0.5 vgs 0 5 1
.DC temp -55 125 10
.DC I_INP -50UA 50UA 1U
.dc STND_W 5U 15U 5U VCC 4.5 5.5 0.5
.dc XBV.VR 0.5 2.5 0.5
.DC DEV M12 W 10U 40U 2U MOS NCHMOS
+ VTO 0.2 0.5 0.8 1 1.5 LIST
.DC BJT MOD1 IS 1E-18 1E-12 10 LOG
+ TEMP -50 70 15
.DC DEV X1 P2 10 20 1
+ X1 1 2 3 SUB1
.SUBCKT SUB1 1 2 3 P1=5 P2=6 P3=7
```

.DELETE

The **.DELETE** instruction causes any subsequent entries to be deleted up to the **.ENDDEL** instruction. If you specify **LIB** or **INCLUDE** from the circuit description option, then the entries in the specified files are deleted and it is not necessary to specify the corresponding **.ENDDEL** instruction.

Formats

```
.DELETE  
.DELETE  LIB    'filename'    name  
.DELETE    'filename'    name  
.DELETE  INCLUDE  'filename'
```

where:

.DELETE	indicates that entries are to be deleted
LIB	indicates that library entries are to be deleted
INCLUDE	indicates that entries are to be deleted from a file
'filename'	represents the name of the file to be referenced. This name must be enclosed in single or double quotes
name	represents the name of the block within the library file whose contents are to be deleted

Examples

```
.DELETE  
.DELETE  LIB    'SYS$LIBRARY'    MODELS  
.DELETE    "/usr/model/library"    NOMINAL  
.DELETE  INCLUDE    "/usr/mosfet/controls"
```

.DISTO

The .DISTO instruction causes the program to compute the distortion characteristics of the circuit in a small-signal mode as part of the AC analysis. The analysis is performed assuming that one or two signal frequencies are imposed at the input. The first frequency f_1 is the nominal analysis frequency, set by the frequency sweep of the .AC instruction. The optional second frequency f_2 ($=\text{skw2} * f_1$) is set implicitly by specifying skw2. The program then computes the following distortion measures:

- HD2 second order harmonic distortion. The magnitude and phase of the frequency component $2 * f_1$ when f_2 is not present.
- HD3 third order harmonic distortion. The magnitude and phase of the frequency component $3 * f_1$ when f_2 is not present.
- SIM2 intermodulation distortion (sum). The magnitude and phase of the frequency component $f_1 + f_2$.
- DIM2 intermodulation distortion (difference). The magnitude and phase of the frequency component $f_1 - f_2$.
- DIM3 intermodulation distortion (second difference). The magnitude and phase of the frequency component $(2 * f_1) - f_2$.

Format

.DISTO rload <interval <skw2 <refpwr <spw2> > > >

where:

- .DISTO** requests a distortion analysis.
- rload the resistor element name of the output load resistor into which all distortion power products are to be computed
- interval the interval at which a distortion-measure summary is to be printed. If omitted or set to zero, summary will not be printed. It is specified in terms of number of frequency points. If it is equal to or greater than one, then the summary is printed for the first frequency, and once every interval frequency step thereafter. interval may be parameterized.
- skw2 the ratio (f_2 / f_1) of the second frequency f_2 to the nominal analysis frequency f_1 . If omitted, a value of 0.9 is assumed. skw2 may be parameterized.
- refpwr the reference power level used for computing the distortion products. If omitted, a value of 1 mw (dbm) is assumed. refpwr may be parameterized.
- spw2 the amplitude of the second frequency f_2 . If omitted, a value of 1.0 is assumed. spw2 may be parameterized.

The summary printout for each frequency is quite extensive. Use the interval parameter to control the amount of output generated.

Examples

```
.DISTO RL 5 0.85 2MW 0.8
.DISTO X1.R5 INTL 0.92 2MW 0.95
```

.DISTR

This statement defines the distribution for use with the statistical analysis package. The equations and parameters for the distributions are defined following the format description.

Format

```
.DISTR  name  type      <pname1 = pval1>      <pname2 = pval2> ...  
+ <pnamen = pvaln>
```

where:

.DISTR	indicates a distribution will be defined
name	is a user-selected name that refers to the distribution. name should contain only alphanumeric characters, beginning with a letter.
type	is one of the following keywords: UNIFORM GAUSS DEXPON GAMMA LOGNOR WEIBUL BIMOD
pname1... pnamen	are the names of the distribution's parameters
pval1 ... pvaln	are the parameter values

.DISTR Equations and Parameters

Uniform

In a uniform distribution there is an equal probability of a sample being chosen anywhere within the parameter range.

Examples

```
.DISTR  MN2  UNIFORM
```

```
.DISTR  D1   UNIFORM
```

Gaussian (Gauss)

Probability Density Function =

$$\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

with MIN < x < MAX

Parameter	Default	Range
MEAN (μ)	0.0	between MIN and MAX
STDEV (σ)	0.33	real
MAX	1	real
MIN	-1	real

Examples

```
.DISTR G1 GAUSS MEAN=0.1 STDEV=0.25
.DISTR D1 GAUSS MAX=1 MIN=-1 MEAN=0.5
+ STDEV=0.25
```

Double Exponential (Dexpon)

Probability Density Function =

$$\frac{1}{\beta} e^{-\frac{|x-\mu|}{\beta}}$$

with $-3 < x < 3$

Parameter	Default	Range
MEAN (μ)	0.0	between real -3 and 3
BETA (β)	1.0	real positive

Examples

```
.DISTR D1 DEXPON
.DISTR D2 DEXPON MEAN=1.0 BETA=0.6
```

Making beta large (>3), makes the double exponential distribution look more like the uniform distribution.

Gamma

Probability Density Function =

$$\frac{1}{\Gamma(\alpha)} k^{\alpha-1} e^{-\frac{k}{\beta}}$$

Parameter	Default	Range
ALPHA (α)	1	integer positive
BETA (β)	1.0	real positive

Examples

```
.DISTR GAM1 GAMMA
```

```
.DISTR GAM2 GAMMA ALPHA=2 BETA=0.75
```

For the default values, the gamma distribution reduces to the exponential distribution.

Log-normal (Lognor)

Use this distribution when the logarithm of the random variable has a normal distribution.

Parameter	Default	Range
MEAN	0.0	between MIN and MAX
STDEV	0.33	real
MAX	1	real
MIN	-1	real

Examples

```
.DISTR R1 LOGNOR
```

```
.DISTR DIST1 LOGNOR MEAN=0.32
```

Weibull (Weibul)

You generally use the Weibull distribution in the analysis of manufacturing tolerances and failures, especially in describing the distribution of the time of failure of given components.

Probability Density Function =

$$\alpha \beta^{-\alpha} x^{(\beta-1)} e^{-\left(\frac{x}{\beta}\right)^{\alpha}}$$

Parameter	Default	Range
ALPHA (α)	1.0	real positive
BETA (β)	1.0	real positive

Examples

```
.DISTR W1 WEIBUL
```

```
.DISTR W2 WEIBUL ALPHA=3 BETA=0.75
```

For ALPHA = 3.2, the Weibull distribution is nearly normal in two shapes.

Bimodal (Bimod)

The shape of the (default) bimodal distribution is two gaussians separated by six standard deviations.

Parameter	Default	Range
RMEAN	0.5	between 0.0 and 1.0
LMEAN	-0.5	between -1.0 and 0.0
RSTDEV	1/6	real positive
LSTDEV	1/6	real positive

Examples

```
.DISTR BIMOD
.DISTR BIMOD LMEAN=-0.7 RMEAN=0.4 LSTDEV=0.2
+ RSTDEV=0.1
```

The overlap of the two gaussians can be controlled by either the standard deviations or the position of the means.

.DUMP

The .DUMP instruction creates a save file that you can use for post processor graphics, and for use with the .RESTART and .SEQUEL instructions.

Format

```
.DUMP STYLE=sname < FILE=fname >
+ < TYPE=analysis > < VARS=vnames >
+ < FORMAT=format > < COMMENT="string">
```

where:

STYLE	indicates the specific output format for the file produced by the .DUMP instruction
sname	represents these STYLE options:
VIEW	for use with the VBASE post-processor graphics program
RESTART	for use with the .RESTART instruction to restart transient analysis
SEQUEL	for use with the .SEQUEL instruction to use the transient analysis results for analyzing another circuit
FILE	indicates that the output file name will be specified
fname	represents an acceptable file name on the computing system
	(Default:
	for STYLE=VIEW: FILE=VBASE.view
	for STYLE=RESTART: FILE=VBASE.res
	for STYLE=SEQUEL: FILE=VBASE.seq)
TYPE	indicates that the type of analysis results to be stored will be specified

analysis	represents the type of analysis results to be stored. Options are: TRAN transient analysis results will be stored DC DC analysis results will be stored AC AC analysis results will be stored ALL transient, DC, and AC analysis results will be stored
VARs	indicates that a keyword or variable name to report voltage or current will be specified. (Default: VARs = ALL)
vnames	represents one of the first three options or combinations of the variable names. Options: NODEVOLT specifies that voltage values will be stored for all nodes in the network CURRENT specifies that the current through all devices in the network will be stored ALL specifies that all device currents and all node voltages in the network will be stored The variables are described in the following format: V (n_1) voltage at node n_1 V (n_1, n_2) voltage difference between nodes n_1 and n_2 I (vsrc) current output through the voltage source vsrc I n (name) current output through the element name with terminal number n. The element name may be a subcircuit call (Xzzzzzzz).
FORMAT	indicates that the physical file format will be specified
format	represents the physical file specification as either ASCII or binary (Default: format = binary)
COMMENT	indicates that comments will be specified
"string"	represents a character string to provide user comments and identification of the target file. This string can be extended to more than one line by using the continuation character + as the first character of the continuation line.

Examples

```
.dump style=view vars=all type=tran file=temp
+ comments="test for this particular feature in all names that you can not
+ believe yet but we want to have repetitions" Format=asc

.dump style=view vars=v(2) v(5) il(m2) type=all
+ comments="test2" Format=bin

.dump style=restart type=tran comments="first phase"
+ Format=asc

.dump style=sequel vars=v(1) v(6) v(1021)
+ comments="first episode" Format=bin
```

Usage Notes

VBASE generates a file in response to the .DUMP instruction. Both binary and ASCII formats can be used for this file. The binary format provides smaller memory requirements on the system, while the ASCII format allows the post processor and VBASE to run on different machines.

The format of the file created by the .DUMP instruction is proprietary to VeriBest, Inc. and is subject to change without warning.

If STYLE = SEQUEL, only node voltage variables must be specified for the VARS qualifier.

The effect of using specified variables will send only the values of those variables to the target file.

If STYLE = RESTART or SEQUEL, only transient analysis is meaningful when specifying the TYPE qualifier.

.END

The last line of each circuit description must be the .END instruction. VBASE interprets all data following the .END to be another circuit description.

Format

```
.END
```

.ENDDEL

The .ENDDEL instruction stops the deletion of entries started by the .DELETE instruction.

Format

```
.ENDDEL
```

where:

`.ENDDEL` indicates the termination of the preceding .DELETE instruction

Example

```
.ENDDEL
```

.FOUR

The .FOUR instruction causes the program to perform a Fourier analysis as part of the transient analysis. The Fourier analysis is performed over the interval (tlast - period, tlast), where tlast is the last simulation timepoint specified for the transient analysis (see the .TRAN instruction), and period is one period of the fundamental frequency. The DC component and first nine components are computed. For maximum accuracy of Fourier analysis, tmax for transient analysis (.TRAN instruction) should be set to period/100.0 (or even less for circuits with high resonance factors). VBASE automatically sets tmax to period/20.0 for the requested period only.

Format

```
.FOUR freq out1 out2 out3 ...
```

where:

.FOUR requests a Fourier analysis
freq fundamental frequency. freq may be parameterized.
out₁ transient analysis output variables for which Fourier analysis is performed.
out₂
out₃ ...

Examples

```
.FOUR 200K V(1) I(VIN)
.FOUR STND_F I(X1.VPL) I1(X1.M1) V(50)
```

.GLOBAL

The .GLOBAL instruction provides a convenient means for connecting power supplies to nested subcircuits.

Format

```
.GLOBAL node1 node2 ...
```

where:

.GLOBAL indicates that globally defined nodes are to be named
node₁ represent node numbers or names (integers, alphanumerics) that are to be globally
node₂ ... defined. That is, these nodes each refer to the same circuit connection both within and
outside subcircuit definitions.

Globally defined nodes should not be used as external names for subcircuit expansions.

Example

```
.GLOBAL 45 100
```

.IC

The .IC instruction is used to set initial node voltages that are held fixed during the DC analysis. Assuming fixed values for the .IC and the source nodes, the DC analysis will be performed on all the other (unset) dependent nodes in the circuit to determine their DC values.

Formats

```
.IC V(node1)=val1 V(node2)=val2 ... V(noden)=valn
.IC V(<Xzzzz.>node)=val V(node)=val
```

where:

.IC	indicates an initial conditions instruction. The keyword .DCVOLT may be used instead of .IC .
V(node)	specifies that the voltage at node <i>node</i> is to be defined. <i>node</i> represents a node number or name
val	voltage value to be assigned to node <i>node</i> . <i>val</i> may be parameterized.
xzzzz	represents a subcircuit expansion call name

Examples

```
.IC V(11)=5 V(4)=01335 V(2)=2.2
.IC V(15)=5 V(x1.12)=5 V(xcell.15)=2.7 V(5)=LOW_V
```

Usage Notes

If all dependent nodes in the circuit are set to initial node voltages with **.IC** instructions, no DC analysis will be performed prior to transient analysis, whether or not the **UIC** option is specified on the **.TRAN** line. In fact, the only way to skip the DC analysis is to set *all* dependent nodes in the circuit with **.IC** instructions. If only some of the dependent nodes are set, then a DC analysis will be done to determine the DC values of the unset dependent nodes.

The **.IC** instruction is allowed in a subcircuit definition. When used in a subcircuit, all specified nodes in different subcircuit instances (calls) have the same initial condition. This can cause an inconsistent DC solution because of these inconsistent initial conditions.

For the **.OP** operating point analysis instruction and small signal AC analysis, ICs are recognized and used during the DC solution process. Nodes with IC's set are simply treated as frozen nodes and the final DC solution for those nodes is the specified values.

.INCLUDE

The **.INCLUDE** instruction allows data from another data file to be included in the program input file.

Formats

```
.INCLUDE 'filename'
```

where:

.INCLUDE	indicates that data from another file is to be included
'filename'	represents the filename (including any pathname) of the file to be included. Note that this name must be enclosed in either single or double quotes.

Examples

```
.include "/cad/moscells/model2"
.INCLUDE "<drive>:VERIBEST/VBA/libelec/vbamod/diode/1N914"
```

.LIB

The .LIB instruction is a program feature, similar to the .INCLUDE instruction, that allows data stored in outside library files to be referenced.

.LIB is used in two contexts:

- A library file containing multiple .LIB/.ENDL definitions
- LIB filename specifications within the original data file that reference a library file

.LIB Library File

The definition of a .LIB library file has the following structure:

Format

```
.LIB name1
...
...
any valid set of VBASE data
...
...
.ENDL name1
.LIB name2
...
any valid set of VBASE data
...
...
.ENDL name2
...
```

where:

.LIB indicates the beginning of a block of library data
name₁ represent reference names for each of the library blocks
name₂ ...

A library file can contain many sections defined by .LIB name and .ENDL name specifications.

Example

```
.LIB    NOMINAL
.MODEL    NM    NMOS    (LEVEL = 1
+ LAMBDA=3.29E-2    TOX=25.0N
+ VTO=0.75    GAMMA=0.605    PHI=0.8)
.MODEL    NP    PMOS    (LEVEL=1
+ VTO=-0.80    LAMBDA=6.8E-2    TOX=25.0N
+ PHI=0.7    GAMMA=0.313)
.ENDL    NOMINAL
```

.LIB Calls

The .LIB calls are used in the program data file to reference data in a library file.

Format

```
.LIB 'filename' name
```

where:

.LIB indicates a library call

'filename' represents the name of the library file to be referenced. This name should be enclosed in single or double quotes.

name represents the name of the block within the library file whose contents are to be included in the program data

Example

```
.lib "/usr/model/library" NOMINAL
```

.MEASURE

The .MEASURE instruction is used to extract either timing information from transient analysis, or frequency information from AC analysis.

Format

```
.MEASURE analysis name < TRIGGER >
+ outvar1 < VAL= val1 state1 cycle1 >
+< TARGET > outvar2 < VAL= val2 state2 cycle2 >
```

where:

.MEASURE indicates that items are to be extracted from the simulation results

analysis represents one of the following qualifiers:

TRAN specifies that a timing value from transient analysis will be extracted

AC specifies that a frequency value from AC analysis will be extracted

name specifies a name that will be used in the output report to identify the value extracted by .MEASURE

TRIGGER indicates that the variable that follows is to be used as the starting point for the measurement

TARGET indicates that the variable that follows is to be used as the ending point for the measurement

outvar₁ represents the variable used as the starting point for the measurement. The variable can have the following format:

V (n₁) voltage at node n₁

V (n₁,n₂) voltage difference between nodes n₁ and n₂

I (Vsrc) current output through the voltage source vsrc

In (name) current output through the element name with terminal number n. The element name may be a subcircuit call (xzzzzzzz).

<code>outvar₂</code>	represents the variable used as the ending point for the measurement. The variable can have the same format as <code>outvar₁</code> .
VAL	indicates that values will be specified
<code>val₁</code> <code>val₂</code>	represents numerical values used for the start and end of the measurement
<code>state₁</code> <code>state₂</code>	represents the state for the beginning or end of the measurement. This must be one of two states: rise or fall. These states represent the behavior of the variable. For example, <code>V(2) 1.0 rise</code> means use the value of <code>V(2)</code> when is greater than or equal to 1.0.
<code>cycle₁</code>	represents the number of repeated events of the described measurement point. For example, <code>V(2) 4.0 fall 3</code> means use the value of <code>V(2)</code> when <code>V(2)</code> falls below 4.0 for the third time.

The time interval or frequency between the two specified points (`outvar1` and `outvar2`) is computed and reported by the name identifier specified on the `.MEASURE` instruction.

The two output variables (`outvar1` and `outvar2`) can be identical.

Example

```
.MEASURE tran delay2 TRIGGER V(10) 0.3 rise
+ TARGET V(11) 4.2 fall 5
```

.MODEL

The `.MODEL` instruction specifies a set of model parameters that are referenced by one or more devices. Specific `.MODEL` instruction types are described in the section for each semiconductor device type.

Format

```
.MODEL mname type ( <pnam=pval> <pnam=pval> ... )
```

where:

.MODEL	indicates that a set of model parameters will be defined
<code>mname</code>	represents the user assigned model name
<code>type</code>	represents one of the following device type keywords:

C	Semiconductor Capacitor
L	Inductor
R	Semiconductor Resistor
CSW	Current Controlled Switch
SW	Voltage Controlled Switch
D	Diode
URC	Uniform Distributed RC Line
DRC	Alternative name for URC model
NPN	NPN BJT
PNP	PNP BJT
NJF	N-type JFET
PJF	P-type JFET
NMOS	N-channel MOSFET
PMOS	P-channel MOSFET
NMES	N-type MESFET
PMES	P-type MESFET
TFM	Transformer Core
DIGITAL	Digital Model
pnam	represents a reserved parameter name, depending upon the device type
pval	model parameter value. The pnam=pval pairs need not be enclosed in parentheses

Example

```
.MODEL DEPL NMOS (LEVEL=1 VTO=-4.0 KP=20U
+ GAMMA=1.31 LAMBDA=0.01 PHI=0.6)
```

.MONTE

This statement enables the VBASE algorithm that executes Monte Carlo analysis for DC, frequency, and transient analysis, and collects data for the nodes or the element currents specified. The parameters varied during Monte Carlo analysis are those followed by the keyword **STAT**. The options associated with MONTE are listed in the “.OPTIONS” section of this chapter.

Format

```
.MONTE <WRITE> type
or
.MONTE <WRITE> type <out-var1>... <out-varn>
```

where:

.MONTE	indicates a Monte Carlo instruction
WRITE	(Optional) displays the parameter values in the output file for each sample
type	represents one of the following keywords:
DC	enables VBASE to collect data during DC analysis
AC	enables VBASE to collect data during frequency analysis
TR	enables VBASE to collect data during transient analysis

DC, AC, and TR are mutually exclusive.

`out-var` (Optional) is a list of output voltage nodes for which data is to be collected during statistical analysis

Note: It is not necessary to specify desired outputs on the `.MONTE` line. You can write `.MONTE` type and then run the `.ACQUIRE` statement to specify outputs as you normally would for any other analysis.

Examples

```
.MONTE WRITE DC V(8)
.MONTE WRITE DC I(RRC) IB(QQ1)
.MONTE TR V(5) I(M23)
.MONTE WRITE AC V(10)
.MONTE WRITE TR V(7) I(RCC)
```

There is no limit to the number of `MONTE` statements that may appear in an input file, but there is a limit of eight output variables per statement.

Permissible output variable types are as follows:

DC: node voltages, currents through all elements

AC: node voltages, currents through voltage sources only

TR: node voltages, currents through all elements

.NODESET

The `.NODESET` instruction affects only the first iteration of the DC solution. The DC solution begins with the `.NODESET` nodes set to the given voltages, source nodes set to their DC values, and other nodes initialized to zero. The `.NODESET` constraints are dropped after the first iteration (or the number of specified iterations) and the DC solution continues normally.

Formats

```
.NODESET V(node1)=val1 V(node2)=val2 ... V(noden)=valn
.NODESET V(Xzzzz.node)=val V(node)=val
```

where:

<code>.NODESET</code>	indicates an initial node-set instruction
<code>V(node)</code>	specifies that the initial voltage at node <code>node</code> is to be defined. <code>node</code> represents a node number or name.
<code>val</code>	voltage value to be assigned to node <code>node</code> . <code>val</code> may be parameterized.
<code>Xzzzz</code>	represents a subcircuit expansion call name

The **`.NODESET`** command causes the specified nodes to be fixed to the specified voltages for a specified number of iterations in the DC solution process. If convergence results during those iterations, those node voltages will be allowed to vary in another DC solution process to ensure that the

correct solution is reached. The number of iterations in this case can be specified by the .OPTIONS NODESET instruction.

Examples

```
.NODESET V(12)=4.5 V(4)=2.23
.NODESET V(x1.250)=1.75 V(xBUFF.212)=5.2 V(50)=6
+ V(5)=HI_V
```

.NOISE

The .NOISE instruction causes the program to perform a noise analysis as part of the AC analysis. VBASE assumes each noise source is statistically uncorrelated to other noise sources in the circuit. VBASE computes the total output noise voltage by summing all the individual thermal noise contributions:

$$V^2 = \text{SUM}(Z * I)^2$$

V is the equivalent noise voltage
 Z is the equivalent impedance of an ideal noiseless resistor
 I is the equivalent noise current

VBASE computes the equivalent input noise by dividing the total output noise by the magnitude of the voltage at the output node. VBASE computes the equivalent output noise at the specified output and the equivalent input noise at the specified input. The contribution of each noise generator in the circuit is also printed. The units for output and input noise are volts/(Hz)^{1/2} or amps/(Hz)^{1/2} and are normalized with respect to the square root of the noise bandwidth. VBASE uses the model parameters KF and AF on the appropriate .MODEL instruction to simulate flicker noise sources.

Format

```
.NOISE outv insrc interval
```

where:

.NOISE requests a noise analysis
 outv the output voltage variable specifying the node at which the noise is summed
 insrc the name of an independent voltage or current source for use as noise input reference
 interval the interval at which a noise analysis summary is to be printed. If omitted or set to zero, the noise summary is not printed. The interval is specified in terms of the number of frequency points. If it is equal to or greater than 1, then the summary is printed for the first frequency, and once every interval frequency step thereafter. The interval may be parameterized.

Examples

```
.NOISE V(3) VINPUT 10
.NOISE V(X1.S) X1.I5 PR_INTL
```

The summary printout for each frequency is quite extensive. Use the interval parameter to control the amount of output generated.

.OP

The .OP instruction causes VBASE to compute the DC operating point.

The .OP instruction causes the program to write complete tables of the operating state of the circuit at one or more timepoints.

Formats

.OP

.OP <TRANSTAT> <format₁> <t₁> <format₂> <t₂> ... <format_n> <t_n>

where:

.OP	requests that tables of operating points be generated
TRANSTAT	prints the operating state at a specified time within a transient analysis without performing a DC solution
format	represents one of the following keywords:
ALL	requests a complete summary of all node voltages, branch currents, component values and power dissipation (Default)
DEBUG	equivalent to ALL
VOLTAGE	requests a table of node voltages only
CURRENT	requests node voltages, source values and power
BRIEF	equivalent to CURRENT
t₁, t₂, ...	timepoints at which the outputs are to be generated. If a format specification is not put before a time value, the program will use the last format value specified (or ALL , if none are specified).

Note: If not used judiciously, the .OP instruction can generate large amounts of output.

Examples

```
.OP
.OP ALL 0 VOL 40NS 60NS CUR 80NS
.OP VOLTAGE
.OP TRANSTAT ALL 10NS
```

.OPTIONS

The .OPTIONS instruction allows you to change and/or reset program control and user options for specific simulation requirements.

Format

.OPTIONS opt opt ... opt=val ... opt ...

where:

.OPTIONS	indicates that program options are to be redefined
opt	represents an option keyword as described in the following pages
val	represents a value to be assigned to certain options

Any combination of the options and assigned values may be included, in any order.

Listed in this section are the .OPTION keywords and an explanation of how each affects the program. An x represents a positive number. The options are listed in six categories: Algorithm Simulation Options, Tolerance Simulation Options, Stress Simulation Options, Statistical Simulation Options, Input Options, and Output Options.

Algorithm Simulation Options

ABSDELTA=x	absolute change for numerical derivative computation. Only applies to table MOSFET models and MOSFET models that have the model parameter NUM-DERIV set. The default is 0.0.
CMIN=x	resets the value of the added grounded capacitor to every user-specified node that has no explicit capacitors connected. Examples of explicit capacitors are CGSO and CGDO in the .MODEL instruction for MOSFET, and capacitor elements. The default is 1.0E-18 Farads.
DCMODE=name	<p>The following DC solution methods can be selected:</p> <p>dcmode=fast dcmode=stiff dcmode=spice dcmode=all</p> <p>dcmode=fast - (default, if no BJTs or controlled sources are present) only uses VBASE proprietary DC solution algorithms.</p> <p>dcmode=stiff - only uses VBASE proprietary DC solution algorithms; however, it uses them with branch voltage limiting for the initial pass through the VBASE Newton-Raphson algorithm. This may help circuits containing highly nonlinear device coverage.</p> <p>dcmode=spice - only uses the VBASE SPICE-like DC solution algorithm; the VBASE proprietary algorithms are not invoked.</p> <p>dcmode=all - (default, if BJTs or controlled sources are present) first attempts a DC solution using the VBASE SPICE-like DC solution algorithm. If convergence is not achieved after 100 iterations, the VBASE proprietary DC solution algorithms are invoked in the manner described above as dcmode=stiff (for example, branch voltage limiting is used for the initial pass through the VBASE Newton-Raphson algorithm).</p>
GMIN=x	resets the minimum conductance through transistors used by the SPICE-like DC solution algorithms. This option has no effect on the VBASE proprietary algorithms. The default is 1E-12.

ITL1=x	resets the DC iteration limit in the SPICE-like DC solution algorithm. This parameter should not be specified. If the parameter <i>is</i> specified, and this limit is reached, the DC phase is terminated, and the next phase begins. A warning message is printed when the limit is reached (see Usage Note 11). The default is option not specified.														
ITL5=x	resets the transient analysis total iteration limit. If this limit is reached, the transient analysis phase is terminated, and the next analysis phase begins. A warning message is printed to the error file when the limit is reached. The default is <i>no limit</i> .														
LIMPTS=x	resets the allowable total number of time points computed during transient analysis. If this limit is reached, the transient analysis phase is terminated, and the next analysis phase begins. A warning message is printed to the error file in this case (see Usage Note 5). The default is <i>no limit</i> .														
METHOD =name	sets the integration method to be used. Gear's method of the second order is the other choice (gear). The trapezoidal method usually produces results in a shorter time, while Gear's method is more robust. The default method is the trapezoidal method (the abbreviation trap can be used).														
NOBYPASS	disables the bypass process. Only use the bypass process with MOSFET devices. Never use the bypass process with any other devices (see Usage Note 7).														
NODESET=x	resets the number of iterations that the DC solution uses to fix the specified nodes to the specified voltages in the .NODESET instruction. The default is 1.														
NOLIMJFT	disables the branch-voltage-limiting for JFET devices. This type of limiting is enabled only when option trmode = stiff is set (see Usage Note 6).														
NOLIMMES	disables the branch-voltage-limiting for MESFET devices. This type of limiting is enabled only when option trmode = stiff is set (see Usage Note 6).														
NOLIMMOS	disables the branch-voltage-limiting for MOSFET devices. This type of limiting is enabled only when option trmode = stiff is set (see Usage Note 6).														
NOSUBS	tells the system to ignore model parameter SUBS for BJTs. Vertical structure is assumed for both NPN and PNP BJT transistors (see Usage Note 4).														
QOPT	selects a charge model for MOSFETs. If specified, this option overrides the charge model selection on the .MODEL instruction. <table> <tr> <th><u>Value</u></th><th><u>Model</u></th></tr> <tr> <td>0</td><td>Yang-Chatterjee</td></tr> <tr> <td>1</td><td>Meyer</td></tr> <tr> <td>2</td><td>Ward-Dutton</td></tr> <tr> <td>3</td><td>BSIM charge</td></tr> <tr> <td>4</td><td>ASPEC charge</td></tr> <tr> <td>5</td><td>Zero charge</td></tr> </table> <p>This parameter is ignored with .OPTION SPICE (see Usage Note 3).</p>	<u>Value</u>	<u>Model</u>	0	Yang-Chatterjee	1	Meyer	2	Ward-Dutton	3	BSIM charge	4	ASPEC charge	5	Zero charge
<u>Value</u>	<u>Model</u>														
0	Yang-Chatterjee														
1	Meyer														
2	Ward-Dutton														
3	BSIM charge														
4	ASPEC charge														
5	Zero charge														
RELDELTA=x	determines relative change for numerical derivative computation. This only applies to table MOSFET model and MOSFET models that have the model parameter NUMDERIV set. The default is 0.001.														

SPICE	selects SPICE interpretations where VBASE normally differs (see Usage Notes 3, 4, 5, and 10).
TNOM=x	resets the nominal temperature at which all circuit values are assumed to be specified. This is the reference value from which temperature adjusted values are calculated. The default is 27°C.
TRMODE=name	<p>has a default value of fast unless BJTs or controlled sources are present in the circuit, in which case the default is stiff.</p> <p>trmode=stiff trmode=fast</p> <p>trmode=stiff - selects a solution method for transient analysis that has slower execution speed, but can help some circuits to converge due to a conservative simulation scheme. This method uses branch voltage limiting and checks for current convergence on nonlinear branch currents through transistors in the transient analysis.</p> <p>trmode=fast - does not do voltage limiting and does not check for current convergence on nonlinear branch currents through transistors in the transient analysis.</p>
<p>Note: If you use the defaults suggested, VBASE and SPICE give different results for certain circuits. Therefore, use the FAST mode ONLY for FET designs with voltages in the 0-10 volt range, and non-exponential currents. It has been shown that in those cases where the FAST mode works (for example, MOSFET microprocessor designs and memories), you get identical results to SPICE with an average of 4X speedup.</p>	
TUSEIC	applies the initial node voltages set in the .IC instruction for each temperature analysis. Not specifying TUSEIC (default) applies the initial node voltages set in the .IC instruction for the first temperature only.
TUSENSET	indicates that the values set with the .NODESET instruction are used for the DC analysis for all temperatures. For example, if analysis is requested at 27, 55, and 90 degrees, nodeset values are used for the specified nodes under all these temperatures. Not specifying TUSENSET (default) applies the initial node voltages set in the .NODESET instruction for the first temperature only.

Tolerance Simulation Options

ABSTOL=x	absolute tolerance value for currents (see Usage Note 1). The default is 1.0×10^{-12} .
CHGTOL=x	resets the charge tolerance value. The default is 1.0×10^{-14} .
DCABSTOL=x	absolute tolerance value for currents for the DC solution only (see Usage Note 1). The default is 1.0×10^{-12} .
DCRELTOL=x	relative tolerance value for voltages and currents for the DC solution only (see Usage Note 1). The default is 0.001.
DCVNTOL=x	absolute tolerance value for voltages for the DC solution only. The default is 1.0×10^{-6} .
FLXTOL=x	resets the flux tolerance value. The default is 1.0×10^{-13} .
IABYPASS=x	absolute current tolerance for bypass. The default is 1.0×10^{-9} .
IRBYPASS=x	relative current tolerance for bypass. The default is 0.001.

RELTOL=x relative tolerance value voltages (see Usage Note 1). The default is 0.001.
VABYPASS=x absolute voltage tolerance for bypass. The default is 1.0×10^{-6} .
VNTOL=x absolute tolerance value for voltages. The default is 1.0×10^{-6} .
VRBYPASS=x relative voltage tolerance for bypass. The default is 0.001.

Stress Simulation Options

POWERTR prints out a report on the power dissipated in TRANSIENT by all elements that have the parameter IMAX, PMAX and/or VMAX specified as part of their model.
POWERDC prints out a report on the power dissipated in DC by all elements that have the parameters PMAX specified as part of their model.

Statistical Simulation Options

INDEP turns off model tracking for statistical analysis. All discrete devices will have independent parameter sets chosen. The default is option not specified, which means that all discretized models that have the same model will have identical parameters in a given run.
SEED=x sets the seed for the random number generator used in the selection of parameter values for statistical analysis. Modify this value to cause the same input file to select different sets of circuits. The default is 9999.
WORST performs worst case analysis, in addition to Monte Carlo <.MONTE> analysis. If you require only a worst case analysis, specify the NOMONT keyword. If you want to perform both worst case and Monte Carlo analysis in the same run, do not specify the NOMONT keyword.
NOMONT=x disables Monte Carlo analysis. This should only be used in conjunction with the WORST keyword.
NMONTE=x specifies the number of runs during statistical analysis.

DIABLO Options

MAXFUNC Maximum number of instructions plus stack entries for all the specified diablo functions. The default is 10000. Anytime this option is set, it must be placed at the beginning of the netlist entry.

Input Options

ASPEC	sets ASPEC compatibility. If you don't specify scale, then scale = 1.0×10^{-6} . If you don't specify scalm, then scalm = 1.0×10^{-6} . For MOSFET models, the defaults are LEVEL = 6, and TLEV = 1. The units of TOX are Angstroms, and the WL option is also invoked. If you don't specify the LEVEL parameter in the .MODEL instruction for MOSFET models then this option must be set before the .MODEL instruction (see Usage Note 9).
DEFAD=x	resets the default MOS drain diffusion area. The default is 0.0.
DEFAS=x	resets the default MOS source diffusion area. The default is 0.0.
DEFL=x	resets the default MOS channel length. The default is 100.0U.
DEFNRD=x	resets the default MOSFET drain parasitic resistance factor. The default is 0.0.
DEFNRS =x	resets the default MOSFET source parasitic resistance factor. The default is 0.0.
DEFPD=x	resets the default MOS drain perimeter. The default is 0.0.
DEFPS=x	resets the default MOS source perimeter. The default is 0.0.
DEFW=x	resets the default MOS channel width. The default is 100.0U.
SCALE=x	scales the device geometries (see Usage Note 10). The default is 1.0 (see Usage Note 8).
SCALM=x	scales the .MODEL parameters for MOSFETs (see MOSFET models). The default is 1.0.
TIMEPAIR	causes the .TRAN instruction to be interpreted according to the TIMEPAIR format. This option must precede the .TRAN instruction to have effect. See the ".TRAN" section of this chapter.
WL	reverses the default order on the MOSFET device specification to WL.
ZDEFAD=x	resets the default MESFET drain area factor. The default is 0.0.
ZDEFAS=x	resets the default MESFET source area factor. The default is 0.0.
ZDEFL=x	resets the default MESFET length. The default is 1.0m
ZDEFNRB=x	resets the default MESFET bulk parasitic resistance factor. The default is 0.0.
ZDEFNRD=x	resets the default MESFET drain parasitic resistance factor. The default is 0.0.
ZDEFNRG=x	resets the default MESFET gate parasitic resistance factor. The default is 0.0.
ZDERNRS=x	resets the default MESFET source parasitic resistance factor. The default is 0.0.
ZDEFPD=x	resets the default MESFET drain periphery factor. The default is 0.0.
ZDEFPS=x	resets the default MESFET source periphery factor. The default is 0.0.
ZDEFW=x	resets the default MESFET width. The default is 1.0m.

Output Options

ACCT	invokes the printout of accounting and run time statistics
CO=x	sets the maximum width of the output data file. The default value of x is an 80 character width. See the ".WIDTH" section of this chapter.
DUMPIC	invokes the printout of node voltages using the format for the .IC instruction after a DC solution
ECHOFILE=x	controls echoing of the include and library files. x=1 echoes the files to the output as they are being read. The default is 0.
NODE	invokes the printout for a node table that lists the elements connected to each node
NODECAPS	invokes the printout for the total capacitance associated with each node specified in the input listing. Nodes connected to voltage sources are not printed. A .OP instruction must also be specified to invoke the printout. When you specify .OP TRAN-STAT, the node capacitance during transient analysis is printed. (Default: NODECAPS is not invoked).
NOMOD	suppresses printout of model parameters
NOPAGE	suppresses page ejects
NUMDGT=x	resets the number of significant digits printed for output variable values. The limits of x are $0 < x < 8$. The default is 4 (see Usage Note 2).
OPTS	causes the option values to be printed
OUTFMT	displays numbers in scientific or engineer notation format. The two types are: SCIENTIF (default) and ENGINEER.

Example

```
.OPTIONS WL NOBYPASS RELTOL=5.0E-4 TNOM=55
```

Usage Notes

1. For RELTOL, ABSTOL, and VNTOL, VBASE automatically adjusts its equivalents of the SPICE tolerance parameters to be proportional to the specified change in the values.
2. For NUMDGT, the value defined is independent of the error tolerances that VBASE uses. The option only affects the program output; the internal program precision is unchanged.
3. Use MOSFET charge model selection with .OPTION SPICE. Use the Meyer charge model if XQC is greater than 0.5. Use the Ward-Dutton charge model if XQC is equal to or lesser than or 0.5. The charge model selection parameter QOPT on the .MODEL instruction is ignored.
4. If you specify .OPTION SPICE, then the model parameter SUBS is not used for BJTs. Vertical structure is assumed for both NPN and PNP BJT transistors. The default is MJS=0.0.
5. Use LIMPTS if you intend to stop the simulation to avoid long execution time. To complete the simulation of transient behavior, a reasonable upper boundary for simulation time is several thousand time points (such as 2000). Using ITL1 can result in incorrect DC solution values which could disable the transient analysis. Use a number such as 2000 for an upper boundary. Once the DC solution is stopped due to ITL1, the transient results may not be trustworthy.

6. NOLIMJFT, NOLIMMES, and NOLIMMOS options can have the effect of reducing simulation time. However, for highly nonlinear models, they may cause convergence problems. The VBASE built-in models for MOSFET, MESFET, and JFET are not considered highly nonlinear.
7. The NOBYPASS option turns on the computation of MOSFET characteristics that otherwise would be bypassed when the MOSFET branch voltages and currents are within the specified tolerances of the VABYPASS, VRBYPASS, IABYPASS, and IRBYPASS options. Using NOBYPASS will result in longer execution time and reduced memory usage.
8. The parameters L, W, AD, AS, PD, and PS for MOSFETs are multiplied by the SCALE option.
9. If you use the SPICE option, then the default value for the mobility parameter UO is the same for both N-channel and P-channel MOSFETs and equals the default value of the N-channel MOSFETs.
10. The default or user-specified values of dcmode and trmode are overridden when you use the SPICE option; dcmode is assigned the value spice and trmode is assigned the value stiff.
11. The use of ITL1 can result in incorrect DC solution values which could disable the transient analyses. Once the DC solution is stopped due to ITL1, the transient results may not be trustworthy.
12. Individual data files for each of the noise and distortion sweep runs are generated when a statistical analysis, in combination with a frequency analysis with noise and distortion, is run.

.PARAM

Use the .PARAM instruction to assign values to parameter variable names. These parameter names can then take the place of numerical values for component and .MODEL descriptions. You can also use parameters within subcircuit definitions. Certain values on analysis instructions can also be parameterized. Refer to each analysis card for valid parameters.

Format

```
.PARAM    pnam1 =valu1    pnam2 =valu2 ...
```

where:

.PARAM indicates that parameter value assignments are to be made

pnam₁ represent user assigned parameter names. These names should be globally unique.

pnam₂ ...

valu₁ numerical values assigned to the parameter names

valu₂ ...

Whenever a user-defined parameter name is used in the VBASE circuit description, the corresponding value is automatically substituted. The .PARAM values are global in nature, and can override the values set in subcircuits (subcircuit call or definition).

Examples

```
M12  10  20  30  40    L=LEN2    W=W1D2
.PARAM  LEN1=3U    LEN2=2.5U    WID2=10U
.param  mult=.82
```

```
.IC      V(8)=HI_V      V(12)=LO_V
.param   HI_V=5.0      LO_V=0.3      HOT=100      WARM=25
.TEMP     WARM      HOT
```

.PLOT

Each .PLOT instruction plots graphs of up to 40 outputs.

Format

```
.PLOT    <SPICE>    type    out1    <(lo,hi)>    out2    <(lo,hi)>
+ out3    <(lo,hi)> ... outn    <(lo,hi)>
```

where:

.PLOT indicates that a set of outputs are to be plotted in line-printer format

<SPICE> outputs ASCII plots in the SPICE type of format

type represents one of the following keywords:

TRAN	plot transient outputs
DC	plot DC sweep outputs
AC	plot linear AC outputs
DISTO	plot distortion analysis outputs
NOISE	plot noise analysis outputs

(lo,hi) represent optional coordinates for the preceding output specification(s)

out₁ represent output specifications. These can have the following formats, for DC, TRAN or AC types.

out₂

out_n...

V(n₁) voltage at node n₁

V(n₁,n₂) voltage difference between nodes n₁ and n₂

I(vsrc) current output through the voltage source vsrc

In(name) current output through the element name with terminal number n (as appears on an element line; for example, for a MOSFET I1=drain current, I2=gate current, and so on). The element name may be a subcircuit call (Xxxxxxxx).

PA(name) average power of element name

PI(name) instantaneous power of element name

For AC type, you can access five additional outputs by inserting the following letters immediately following V or I:

R real part

I imaginary part

M magnitude

P phase
DB $20 \times \log_{10}$ (magnitude)

For TRAN type, you can access an additional output by inserting the number 0 immediately following V or I. This causes the time = zero value of the output to be plotted as a constant.

Output specification must be ONOISE or INOISE for NOISE output.

Output specification must be HD2, HD3, SIM2, DIM2 or DIM3 for DISTO output.

Output specification may be immediately followed by (R), (I), (M), (P) or (DB) for NOISE and DISTO output. The meaning of these suffixes is explained above.

Usage Notes

The program automatically determines the minimum and maximum values of all output variables being plotted and scales the plot to fit. More than one scale is used if the output variable values warrant (for example, mixing output variables whose values are orders of magnitude different still gives readable plots).

The overlap of two or more traces on any plot is indicated by the letter X.

When more than one output is specified on the same plot, the first output specified is both printed and plotted. If you want a printout of all variables, then a companion .PRINT instruction must be included.

There is no limit on the number of .PLOT instructions specified for each type of analysis.

Plotted node voltages and currents may reference devices and nodes inside subcircuits. For example, I2 (X1.M5) specifies current through the gate terminal of M5 in subcircuit X1. V (x12.500) requests the voltage of node 500 in subcircuit x12.

In AC analysis, VBASE does not support branch current output for devices other than independent voltage sources.

Examples

```
.PLOT TRAN V(5) V(4) V(0,5) V(x1.7) I3(x1)
.plot dc i(vin) i2(rx3) i1(cout) i3(ml2) i4(x2.q5)
.PLOT NOISE ONOISE
.PLOT AC VDB(5) V(3)
.PLOT DISTO HD2(R)
.PRINT
```

.PRINT

A .PRINT instruction causes a table of voltages and branch currents to be printed.

Format

```
.PRINT type out1 out2 ... outn
```

where:

.PRINT	indicates that a set of outputs are printed in tabular format
type	represents one of the following keywords:
TRAN	print transient outputs
DC	print DC sweep outputs
AC	print linear AC outputs
DISTO	print distortion analysis outputs
NOISE	print noise analysis outputs
out₁ out₂ out₃ ...	represent output specifications. These can have the following formats, for DC, TRAN, or AC types:
V(n₁)	voltage at node n ₁
V(n₁,n₂)	voltage difference between nodes n ₁ and n ₂
I(vsrc)	current output through the voltage source vsrc
In(name)	current through the element name with terminal number n (as appears on an element line; for example, for a MOSFET I1=drain current, I2=gate current, and so on). The element name may be a subcircuit call (Xzzzzzzz).
PA(name)	average power of element name
PI(name)	instantaneous power of element name

For AC type, five additional outputs are accessed by inserting the following letters immediately following the V or I:

R	real part
I	imaginary part
M	magnitude
P	phase
DB	20 x log ₁₀ (magnitude)

For TRAN type, you can access an additional output by inserting the number 0 immediately following V or I. This causes the time = zero value of the output to print as a constant.

For NOISE output, output specification must be ONOISE or INOISE.

For DISTO output, output specification must be HD2, HD3, SIM2, DIM2, or DIM3.

For NOISE and DISTO output, output specification may be immediately followed by (R), (I), (M), (P), or (DB). The meaning of these suffixes is explained above.

There is no limit on the number of .PRINT instructions.

Output requests for node voltage and element terminal current may reference subcircuits. For example, I2 (x1.M5) specifies current through the gate terminal of M5 in subcircuit X1. V (x12.500) requests the voltage of node 500 in subcircuit x12.

In AC analysis, branch current output is not supported for devices other than independent voltage source.

Examples

```
.PRINT TRAN V(4) V(7) I(VBG3) V(22,5)
.print dc i2(r5) i4(x3.M8) i(vcc) I5(x1.xINV)
```

.RESTART

The .RESTART instruction uses the file created by the .DUMP instruction in a previous simulation to restart and continue a transient analysis starting from the last time point in the previous simulation.

Format

```
.RESTART filename
```

where:

.RESTART indicates that a transient analysis is to be restarted
 filename represents a file in the format created by the .DUMP instruction
 By specifying ASCII format in the dump phase, the results can be moved to a different machine and the transient analysis restarted with a copy of VBASE on that machine.

The circuit topology and node names can not be changed before restarting a transient analysis. However, it is possible to stop a transient analysis through the TERMCOND feature in .TRAN instruction, modify the circuit, and then restart the analysis. Changing only small numbers of device parameters and/or model parameters will produce acceptable results through a restart analysis. To avoid nonconvergence caused by discontinuity avoid using dramatic value changes.

The DC solution is not performed in a restarted transient analysis. Output is only provided for those time points after the restart time.

Example

```
.restart VBASE.res
```

.SENS

The .SENS instruction causes the program to determine the DC and AC small-signal sensitivities of specific outputs with respect to every circuit parameter.

Format

```
.SENS <DC> out1 <out2 < ...<outn>>>
.SENS AC out1 freq1 <freq2 < ...<freqn>>>
```

where:

.SENS requests a sensitivity analysis
 DC requests a DC sensitivity analysis

`out1` represents an output specification in one of the following formats:

$$\text{out}_2$$

out₃ ...

$V(n_1)$ Voltage at node n_1

V_(n₁,n₂) Voltage difference between nodes n₁ and n₂. These nodes cannot be connected across a source.

AC requests an AC sensitivity analysis

`freq1` a value specifying the frequency at which the AC sensitivity analysis is performed

freq₂

freq₃ ...

For large circuits, .SENS can generate a considerable amount of output.

Note: Conversion from rectangular to polar form:

The Phase Sensitivity from Rectangular sensitivity data is:

$$\frac{\partial \phi}{\partial \rho} = 57.29577 Im\left(\frac{1}{V_0} \frac{\partial V_0}{\partial p}\right)$$

The Magnitude Sensitivity from Rectangular sensitivity data is:

$$\frac{\partial |V_o|}{\partial \rho} = \frac{1}{|V_0|} \text{Re} \left(\bar{V}_0 \frac{\partial V_0}{\partial p} \right)$$

Examples

```
.SENS DC V(2,3) V(x1.9)
```

```
.SENS AC V(2,3) V(1) 100 10K 1M
```

.SEQUEL

The `.SEQUEL` instruction partitions a circuit into multiple blocks that can be analyzed sequentially with the help of the `.DUMP` instruction.

Format

```
.SEQUEL      filename  node1  node2 ...noden
```

where:

.SEQUEL indicates that a circuit is to be partitioned

`filename` represents the name of a file created by the `.DUMP` instruction

node₁ represent the names of the nodes specified in the .DUMP instruction

...

$$\text{node}_n$$

If the circuit is partitioned at nodes which have little coupling (minimal feedback) and the multiple blocks do not constitute any loop, then the sequel simulation will provide accurate results when compared to the simulation of the full circuit as a complete unit.

The sequence of the nodes specified on the .SEQUEL instruction needs to be exactly the same as the sequence specified in the .DUMP instruction.

If any voltage source is connected to the sequel node, the sequel values will override the voltage source values.

If an analysis other than transient analysis is requested with the .SEQUEL instruction, VBASE generates an error message and stops if one of the sequel nodes is connected to only one device in the circuit. If no other analysis is requested, the transient analysis is then performed without further error messages.

Example

```
.sequel VBASE.seq 5 16 4 21
```

.SUBCKT

Formats

```
.SUBCKT name n1 < n2 < n3 < ... > > >
.....
xxxxxxxxx
xxxxxxxxx
.....
xxxxxxxxx
xxxxxxxxx
.ENDS <name>

.SUBCKT name n1 < n2 < n3 < ... > > > < parnam = pval ... >
.....
xxxxxxxxx
xxxxxxxxx
.....
.....
xxxxxxxxx
xxxxxxxxx
.ENDS <name>
```

where:

.SUBCKT	indicates the beginning of a subcircuit definition
name	represents the subcircuit name. This name must be different from all other subcircuit names.
n ₁ , n ₂ , n ₃	represent the numbers or names (integers, alphanumerics) of the external nodes of the subcircuit. These node names must be distinct and must not include the ground node. There is no limit on the number of external node names.
xxxxxxxxx xxxxxxxxx	represent the component/model description lines that define the subcircuit topology

<code>parnam =</code> <code>pval</code>	represents the parameter name set to a value for use only in the subcircuit. This is overridden by an assignment in the subcircuit call or by a value set in a <code>.PARAM</code> instruction.
<code>.ENDS</code>	indicates the end of a subcircuit definition. The keywords <code>.MACRO</code> and <code>.EOM</code> may be used instead of <code>.SUBCKT</code> and <code>.ENDS</code> , respectively.
<code>name</code>	(optional) should be the same as the name of a preceding and unterminated subcircuit definition. If <code>name</code> is omitted in a <code>.ENDS</code> instruction, all subcircuits currently being defined are terminated. You only need to specify <code>name</code> for nested unterminated subcircuit definitions when it is not intended to terminate all of them.

Example

```
.SUBCKT  OPAMP  1  2  3  4
.SUBCKT  ADDR  10  20  30  STND=3U
...
...
...
.ENDS
```

Usage Notes

Subcircuit definitions may contain other subcircuit definitions, device models, initial conditions, nodesets, and subcircuit expansions. However, instructions other than `.MODEL`, `.IC`, or `.NODESET` may not appear within a subcircuit definition.

Any device models or subcircuit definitions included in a subcircuit definition are strictly local to the definition. That is, they are not known outside the subcircuit definition and refer to different entities than do external uses of the same model and subcircuit names.

The use of any model or subcircuit name that is not defined locally is assumed to refer to an external model or subcircuit. The model/subcircuit is found by looking for a local definition of the name in a sequence of subcircuit definitions, starting with the one containing the use of the name. The search continues outward until either the name is found or there is no enclosing subcircuit.

Node names not included among n_1, n_2, n_3, \dots nodes on the `.SUBCKT` line are strictly local. However, naming a node is considered part of the definition, with the result that no external node names are directly available within the sub-circuit. Ground (0) is the only exception; it is always global. External node names are available indirectly by using the replacement name from the subcircuit expansion.

You can use the `.GLOBAL` instruction to declare that certain nodes will have the same meaning both inside and outside subcircuits. Global specification is convenient for connecting power supplies to nodes within subcircuits.

.TEMP

The `.TEMP` instruction defines the temperature(s) at which simulation is performed. If not specified, a single temperature of 27°C is assumed.

Format

```
.TEMP   $t_1$   < $t_2$   <...< $t_n$ >>>
```

where:

.TEMP	indicates that simulation temperatures are to be specified
t_1	temperatures in degrees C at which simulation is performed. VBASE ignores values less than -273°C (0 °K). Temperatures may be parameterized.
$t_2 \dots$	

The reference temperature value can be changed using the TNOM = option on the .OPTIONS instruction.

Examples

```
.temp 125.0
.TEMP -55.0 25.0 125.0
.TEMP WARM HOT
```

.TF

The .TF instruction requests the DC small-signal transfer function analysis of the circuit.

Format

```
.TF out srcname
```

where:

.TF	requests a transfer function analysis
out	represents an output specification using one of the following formats:
V(n_1)	voltage at node n_1
V(n_1, n_2)	voltage across nodes n_1 and n_2
I(vsrc)	current through a voltage source
srcname	name of the input source

Examples

```
.TF V(333,222) VIN
.TF V(XIN.100) XIN.VPULSE
```

.TRAN

The .TRAN instruction causes the program to perform a transient analysis.

Formats

```
.TRAN tstep tlast <tstart <tmax>> <UIC>
+ <TERMCOND exp>
.TRAN <TIMEPAIR> <tstep1 tlast1 <TMAX=t1> ...
+ tstepn tlastn <TMAX=tn>> <START=val> <UIC>
+ <TERMCOND exp >
```

where:

.TRAN	requests a transient analysis
tstep ₁	time step printing or plotting increment for line printer output. tstep may be parameterized.
tstep ₂ ...	
tlast ₁	last simulation timepoint. tlast may be parameterized.
tlast ₂ ...	
tstart	starting timepoint for simulation output. If not specified, tstart = 0 is assumed. Transient simulation always starts from time = 0. Only use tstart for output printing.
START	starting timepoint for simulation output. If not specified, START = 0 is assumed. Transient simulation always starts from time = 0. Only use START for output printing.
TMAX	largest internal time step used during t ₁ ...t _n transient analysis. The largest time step is for the i th time segment which begins at tlast (i - 1) and ends at tlast (i).
UIC	use initial conditions flag for the .IC instruction (see Usage Note 1)
TIMEPAIR	invokes the TIMEPAIR format for the .TRAN instruction.
TERMCOND	invokes the conditional termination for transient analysis based on the expression exp
exp	an expression used to terminate transient analysis. Transient analysis terminates when the expression is true (see Usage Note 2).

Use the following items for evaluating the expression exp (see Usage Notes 3 and 4):

Operands:	constants parameters output variables
Reserved Keywords:	time -- simulated time value tempc -- simulated temperature in degree centigrade tempk -- simulated temperature in degree Kelvin
Separators:	space tabs
Delimiter:	{ }
Continuation:	+ (if 1st character in line)
Arithmetic Operators:	+ add - subtract * multiply / divide ** exponential
Relationals:	> greater than < less than >= greater than or equal to <= less than or equal to == equal to

Logicals:	&& and or
Functionals:	sin: sin function cos: cosine function tan: tangent function ln: natural log function log10: base 10 log function abs: absolute value atan: arctan function exp: $\exp(x)$ is equal to e^{**x}

Examples

Example 1

```
.TRAN 1NS 100NS
.tran 1ns 1000ns 500ns
.TRAN 10NS 1US UIC
.TRAN DEFAULT 1US 100NS 0NS 5US
.TRAN 2NS 100NS TERMCOND
+ { v(3) > 3.0 * { 2 + V(2,4) } } && { v(10) < 3.2 }
```

The preceding example stops transient analysis when:

v(3) is greater than $3.0 * \{ 2 + V(2,4) \}$
and
v(10) is less than 3.2.

Example 2

```
.tran 2ns 100ns termcond { v(5) > test + v(20) }
```

In the preceding example, test is defined as a parameter.

Example 3

```
.tran 5n 200u termcond { {v0(1)> 3 && v(2)<v(3) - v(5)} ||
+ {v0(1) < 3 && v(2) > v(4) * v(7) } }
```

In the preceding example, the time zero value of a voltage can be referenced, and the transient analysis stops when one of the following conditions is satisfied:

- Time is equal to 200 micro seconds
- v(1) at time zero is greater than 3 volts and v(2) is less than $v(3) - v(5)$
- v(1) at time zero is less than 3 volts and v(2) is greater than $v(4) * v(7)$

Usage Notes

1. If the initial conditions are completely specified using a .IC instruction, no DC solution is performed and the values on the .IC instruction are used as the initial conditions. If the initial conditions are not completely specified, node voltages specified in .IC instructions are held constant during DC analysis and all other dependent node voltages are computed normally.

2. When evaluating the conditional termination (TERMCOND) expression, numerical computation can only compute discrete time points. Thus, the output of the simulation may slightly pass the point where the condition is satisfied or may stop less than one tstep before the point.
3. When evaluating the conditional termination (TERMCOND) expression, parentheses () are *not* recognized. Instead, braces { } are used to group expressions.
4. When evaluating the conditional termination (TERMCOND) expression, constants can be numbers such as 0.3, 9.1E-10 (see the “Legal Numbers” section in *Chapter 1, Introduction*), output variables can be names such as V(10), V(3,5), I(vsrc), I1(m5) (see the “Legal Names” section in *Chapter 1, Introduction*), and parameters can be variables such as defined in .PARAM TEST = 3 (see the “.PARAM” section of this chapter).
5. Output variables can be names such as V0(10), V0(3,5), I0(vsrc), and I01(m5) which refer to the time = zero values of the output variables V(10), V(3,5), and I1(m5) respectively, in this case.

.VARY

The .VARY instruction is used to execute several runs together for analyzing the effects of variations of parameters on circuit behavior. Any parameter with a numerical value can be varied.

Formats

```
.VARY e-type tol
.VARY d-type dnam [pnam1 pval1 <pnam2 pval2< ...<pnamn pvaln>>>][val]
.VARY m-type mnam pnam1 pval1 <pnam2 pval2> ...
```

where:

.VARY indicates that several runs are to be executed together

e-type represents one of the following element keywords:

RESISTOR
CAPACITOR
INDUCTOR

Only the first three letters of the above keywords are required. RES, CAP, and IND refer to all devices of that type.

tol represents a list of space separated values tol indicating resistance, capacitance, and inductance tolerance. (Resistance, capacitance, and inductance are multiplied by 1 + tol where tol must be greater than -1.0.)

d-type represents one of the following device keywords:

DEVICE refers to a particular device
VOLTAGE refers to voltage sources that do not vary with time

Only the first three letters of the above keywords are required.

dnam represents the user assigned device name

pnam₁ represent the parameter names

pnam₂...

Some devices do not have this field, but do have values that refer to area factors. Other devices, like capacitors, also do not require this field but their values refer to tolerances as with the RES element types.

For semiconductor models, the parameter names are exactly the same as in the .MODEL statement except for contact resistance, which has a *TOL* in front of the word (for example, TOLRE for BJT) that refers to tolerance factors.

`pval1` Each pval represents a list of space separated parameter values (for example, 2U 3U 12U).
`pval2...`

`val` represents a list of space separated values (for example, 5 18 25). For devices like capacitors and resistors, val means tolerance. For devices like BJTs, val means area factor.

`m-type` represents one of the following model keywords:

MOSFET

MESFET

BJT

DIODE

JFET

Only the first three letters of the above keywords are required.

`mnam` represents the user assigned model name

A variation on a single statement will occur sequentially while different statements in the same file will take every combination as in a nested DO loop with the first .VARY statement as the innermost nest. For example:

```
.VARY DEV M1 L 1U W 1U 2U
.VARY DEV M2 L 2U 3U
```

will have four variations in this order:

1. M1 L=1U W=1U M2 L=2U
2. M1 L=1U W=2U M2 L=2U
3. M1 L=1U W=1U M2 L=3U
4. M1 L=1U W=2U M2 L=3U

Examples

```
.VARY MOS MODEL1 NSUB 1E15 1E16 TOX 1
+ BJT MOD2 RE -.5 .5
.VARY RESISTORS .1 .2 -.3
.VARY DEVICE R2 .4 .6 M1 L 2U 3U W 10U 11U 12U
.VARY VOLTAGE VIN 4.5 5
.VARY DEV Q33 2 4
```

.WIDTH

The .WIDTH instruction is used to define the maximum width of data input and output.

Note: The .WIDTH instruction is order dependent within the netlist. It must come after the title statement but before any of the topology/device statements.

Format

```
.WIDTH  <IN=inchr>  <OUT=outwid>
```

where:

.WIDTH	indicates that input and/or output widths are to be changed
IN	indicates that the maximum input line width is to be specified
inchr	number of characters (counting a tab as one character) that are read from an input file line. Continuation lines in this context are treated as independent lines. The default is inchr = 80.
OUT	indicates that the maximum output print width is specified
outwid	maximum width of the output file. This value must be less than 133. The default is outwid = 80.

Examples

```
.WIDTH  IN=72  OUT=80  
.WIDTH  OUT=132
```

Appendix A

Circuit Examples

This Appendix contains examples of the following circuits:

- Differential Pair
- MOSFET Device
- RTL Inverter
- Four-bit Adder

Example 1 - Differential Pair

This file determines the DC operating point and small-signal transfer function of a simple differential DC pair. In addition, the AC small-signal response is computed over the frequency range 100 Hz to 100 MHz. Explanations of the data lines follow the example.

```
(1)SIMPLE DIFFERENTIAL PAIR
(2)VCC 7 0 12V
(3)VEE 8 0 -12V
(4)VIN 1 0 AC 1
(5)RS1 1 2 1K
(6)RS2 6 0 1K
(7)Q1 3 2 4 MOD1
(8)Q2 5 6 4 MOD1
(9)RC1 7 3 10K
(10)RC2 7 5 10K
(11)RE 4 8 10K
(12).MODEL MOD1 NPN BF=100 VBF=50 IS=1.E-12 RB=100
(13)+ TF=.6NS CJC=.5PF
(14).TF V(5) VIN
(15).AC DEC 10 100 100MEG
(16).PLOT AC VM(5) VP(5)
(17).PRINT AC VM(5) VP(5)
(18).END
```

Explanations

- (1) Title line. Must be the first line in the file. This line will be printed verbatim on the output.
- (2),(3) DC voltage sources
- (4) Linear AC source with a value of 1.
- (5),(6) Resistors
- (7), (8) BJTs referencing the MOD1 .MODEL instruction.
- (9),(10),(11) Additional resistors
- (12),(13) .MODEL instruction to define the parameters of BJTs Q1 and Q2. Note use of continuation.
- (14) .TF requests a transfer function analysis of the voltage at node 5 as a function of VIN.
- (15) .AC requests a linear AC analysis. Decade variation, with 10 points-per-decade from 100 Hz to 100 MHz.
- (16),(17) .PLOT and .PRINT output of the voltage magnitude (VM) and voltage phase (VP) at node 5.
- (18) .END instruction. Must be the last line in the description.

Example 2 - MOSFET Device

This file computes the output characteristics of a MOSFET device over the range 0-10 V for VDS and 0-5 V for VGS. Explanations of the data lines follow the example.

```
(1)J. DOE MOS OUTPUT CHARACTERISTICS (FILE: XMOS1.DAT)
(2).OPTIONS OPTS NOBYPASS
(3)VDS 3 0
(4)VGS 2 0
(5)MX1 1 2 0 0 XMOS1 L=4U W=6U AD=10P AS=10P
(6).MODEL XMOS1 NMOS VTO=-2 NSUB=1.0E15 UO=550 RS=1
(7)*VIDS MEASURES ID
(8)*(IF VDS WAS USED, ID WOULD BE NEGATIVE)
(9)VIDS 3 1
(10).DC VDS 0V 10V .5V VGS 0V 5V 1V
(11).PRINT DC I(VIDS) V(2)
(12).PLOT DC I(VIDS)
(13).END
```

Explanations

- (1) Title Line. Should identify file and/or user.
- (2) .OPTIONS requests no bypass and that the options be printed.
- (3),(4) Voltage sources whose values are to be defined on the .DC instruction.
- (5) MOSFET device specification. References the following .MODEL instruction.
- (6) .MODEL instruction defines the parameters for MOSFET MX1.
- (7),(8) Comment lines. Note that a continuation (+) cannot be used.
- (9) Voltage source used to sense the MOSFET drain current. Value is assumed to be 0 V.
- (10) VDS will be swept in .5 volt increments from 0 volts to 10 volts for each value of VGS. VGS will be swept from 0 volts to 5 volts in 1 volt increments.
- (11),(12) Print and plot the DC transfer function outputs.
- (13) .END must end the file.

Example 3 - RTL Inverter

This file requests the DC transfer curve and the transient pulse response of a simple RTL inverter. Explanations of the data lines follow the example.

```
(1)j. smith simple rtl inverter
(2)vcc 4 0 5v
(3)vin 1 0 pulse 0v 5v 4ns 2ns 3ns 30ns
(4)rb 1 2 10k
(5)q45 3 2 0 q45
(6)rc 3 4 1k
(7).plot dc v(3)
(8).print tran v(3)
(9).model q45 npn bf 20 rb 100 tf .1ns cjc 2pf
(10).dc vin 0v 5v 0.1v
(11).tran 1ns 100ns
(12).end
```

Explanations

- (1) Title line.
- (2) Power supply voltage of 5 V.
- (3) Pulsed voltage source with an initial value of 0.0 volts and a pulsed value of 5.0 volts. The delay time is 4 ns, rise time is 2 ns and fall time is 3 ns. The pulse width is 30 ns.
- (4),(6) Resistors.
- (5) BJT specification. Note that the transistor name and model reference name are identical. This is not recommended but is permissible.
- (7) Specifies plotted output of the DC transfer function analysis.
- (8) Specifies tabular output of the transient analysis.
- (9) Model specification of the BJT q45.
- (10) Requests a DC transfer function, sweeping vin from 0 volts to 5 volts in .1 volt increments.
- (11) Requests a transient analysis from time=0.0 (default) to 100 ns. The output step will be 1 ns.
- (12) Last line in the file.

Example 4 - Four-bit Adder

This file defines a four-bit binary adder. The description uses nested subcircuits to describe the circuit in hierarchical fashion. Explanations of the data lines follow the example.

```
(1)ADDER - 4 BIT ALL-NAND-GATE BINARY ADDER
(2)
(3)*** SUBCIRCUIT DEFINITIONS ***
(4)
(5).SUBCKT NAND 1 2 3 4
(6)* NODES: INPUT(2), OUTPUT, VCC
(7)Q1 9 5 1 QMOD
(8)D1CLAMP 0 1 DIOD
(9)Q2 9 5 2 QMOD
(10)D2CLAMP 0 2 DIOD
(11)RB 4 5 4K
(12)R1 4 6 1.6K
(13)Q3 6 9 8 QMOD
(14)R2 8 0 1K
(15)RC 4 7 130
(16)Q4 7 6 10 QMOD
(17)DVBEDROP 10 3 DIOD
(18)Q5 3 8 0 QMOD
(19).ENDS NAND
(20)
(21).SUBCKT ONEBIT 1 2 3 4 5 6
(22)* NODES:INPUT(2),CARRY-IN,OUTPUT,CARRY-OUT, VCC
(23)X1 1 2 7 6 NAND
(24)X2 1 7 8 6 NAND
(25)X3 2 7 9 6 NAND
(26)X4 8 9 10 6 NAND
(27)X5 3 10 11 6 NAND
(28)X6 3 11 12 6 NAND
(29)X7 10 11 13 6 NAND
(30)X8 12 13 4 6 NAND
(31)X9 11 7 5 6 NAND
(32).ENDS ONEBIT
(33)
(34).SUBCKT TWOBIT 1 2 3 4 5 6 7 8 9
(35)* NODES:INPUT-BIT0(2)/BIT1(2),OUTPUT-BIT0/BIT1,
(36)*CARRY-IN, CARRY-OUT, VCC
(37)X1 1 2 7 5 10 9 ONEBIT
(38)X2 3 4 10 6 8 9 ONEBIT
(39).ENDS TWOBIT
(40)
(41).SUBCKT FOURBIT 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
(42)* NODES: INPUT - BIT0(2)/BIT1(2)/BIT2(2)/BIT3(2),
(43)* OUTPUT - BIT0/BIT1/BIT2/BIT3, CARRY-IN,
(44)CARRY-OUT, VCC
```

Circuit Examples

```
(45)X1 1 2 3 4 9 10 13 16 15 TWOBIT
(46)X2 5 6 7 8 11 12 16 14 15 TWOBIT
(47).ENDS FOURBIT
(48)
(49)
(50).MODEL DIOD * (IS=.9E-16)
(51).MODEL QMOD NPN (BF=75 RB=100 CJE=1PF CJC=3PF)
(52)VCC 99 0 DC 5V
(53)VIN1A 1 0 PULSE(0V 3V 0 10NS 10NS 10NS 50NS)
(54)VIN1B 2 0 PULSE(0V 3V 0 10NS 10NS 20NS 100NS)
(55)VIN2A 3 0 PULSE(0V 3V 0 10NS 10NS 40NS 200NS)
(56)VIN2B 4 0 PULSE(0V 3V 0 10NS 10NS 80NS 400NS)
(57)VIN3A 5 0 PULSE(0V 3V 0 10NS 10NS 160NS 800NS)
(58)VIN3B 6 0 PULSE(0V 3V 0 10NS 10NS 320NS 1600NS)
(59)VIN4A 7 0 PULSE(0V 3V 0 10NS 10NS 640NS 3200NS)
(60)VIN4B 8 0 PULSE(0V 3V 0 10NS 10NS 1280NS 6400NS)
(61)
(62)X1 1 2 3 4 5 6 7 8 9 10 11 12 0 13 99 FOURBIT
(63)RBIT0 9 0 1K
(64)RBIT1 10 0 1K
(65)RBIT2 11 0 1K
(66)RBIT3 12 0 1K
(67)RCOUT 13 0 1K
(68).PLOT TRAN V(1) V(2) V(3) V(4) V(5) V(6) V(7) V(8)
(69).PLOT TRAN V(9) V(10) V(11) V(12) V(13)
(70).PRINT TRAN V(1) V(2) V(3) V(4) V(5) V(6) V(7) V(8)
(71).PRINT TRAN V(9) V(10) V(11) V(12) V(13)
(72)
(73).TRAN 1NS 6400NS
(74)
(75).OPTIONS LIMPTS=6401
(76).END
```

Explanations

- (1) Title line must be first in file.
- (2),(4) Blank lines are allowed (and improve readability).
- (3) The * for a comment need not start in column 1.
- (5) Subcircuit NAND defined with four external nodes.
- (7) to (18) Subcircuit definition. Note that transistors and diodes reference .MODEL instructions outside the subcircuit definition (lines 50 & 51).
- (19) End of subcircuit definition. NAND is optional since there is no ambiguity.
- (21) Subcircuit ONEBIT has six external nodes.
- (22) Comments can be used anywhere in the data file.
- (23) to (31) Expansions (calls) of the nand-gate subcircuit.
- (32) End of the ONEBIT subcircuit.
- (34) to (39) Two-bit adder subcircuit uses onebit expansions.
- (41) to (47) Four-bit adder subcircuit uses twobit expansions.
- (50),(51) Model specifications for the diode and BJT.
- (53) to (60) Input sources form a binary counter.
- (62) Expansion of the four-bit adder.
- (68) to (71) Output specifications.
- (73) Transient analysis specification.
- (75) Options specification.
- (76) Last line in the file.

Appendix B

Model Equations

This appendix provides the equations that describe the following models:

- BJT
- Diode
- JFET
- MESFET
- MOSFET
- Distributed RC Line

BJT Model

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

$$I_{SE} = C_2 I_S$$

$$I_{SC} = C_4 I_S$$

DC Current

$$q_1 = \frac{1}{1 - \frac{v_{bc}}{V_{AF}} - \frac{v_{be}}{V_{AR}}}$$

$$q_2 = \frac{I_S}{I_{KF}} \left(e^{\frac{v_{be}}{v_t}} - 1 \right) + \frac{I_S}{I_{KR}} \left(e^{\frac{v_{bc}}{v_t}} - 1 \right)$$

$$q_b = \frac{1}{2} q_1 (1 + \sqrt{1 + 4q_2})$$

$$i_c = \frac{I_S}{q_b} \left(e^{\frac{v_{be}}{v_t}} - e^{\frac{v_{bc}}{v_t}} \right) - \frac{I_S}{\beta_R} \left(e^{\frac{v_{bc}}{v_t}} - 1 \right) - I_{SC} \left(e^{\frac{v_{bc}}{N_C v_t}} - 1 \right)$$

$$i_b = \frac{I_S}{\beta_F} \left(e^{\frac{v_{be}}{v_t}} - 1 \right) + I_{SE} \left(e^{\frac{v_{be}}{N_E v_t}} - 1 \right) + \frac{I_S}{\beta_R} \left(e^{\frac{v_{bc}}{v_t}} - 1 \right) + I_{SC} \left(e^{\frac{v_{bc}}{N_C v_t}} - 1 \right)$$

If $IRB = 0.0$, then

$$RB = RBM + \frac{RB - RBM}{q_b}$$

If $IRB > 0.0$, then

$$z = \frac{-1 + \sqrt{1 + 14.59025 \left(\frac{ib}{IRB} \right)}}{2.4317 \sqrt{\frac{ib}{IRB}}}$$

$$RB = RBM + 3(RB - RBM) \frac{(\tan(z) - z)}{z \tan^2(z)}$$

Charge Storage

$$q_{be} = TF \frac{1 + XTF e^{\frac{v_{bc}}{\sqrt{2}VTF}} \left[\frac{IS \left(e^{\frac{v_{be}}{V_i}} - 1 \right)}{IS \left(e^{\frac{v_{be}}{V_i}} \right) + ITF} \right]^2}{q_b} IS \left(e^{\frac{v_{be}}{V_i}} - 1 \right) + \int_0^{v_{be}} C_{be}(v) dv$$

$$q_{bc} = TR IS \left(e^{\frac{v_{bc}}{V_i}} - 1 \right) + XCJC \int_0^{v_{bc}} C_{bc}(v) dv$$

$$q_{bxc} = (1 - XCJC) \int_0^{v_{bc}} C_{bc}(v) dv$$

If $v < FC VJ$, then

$$C(v) = \frac{CJO}{\left(1 - \frac{v}{VJ} \right)^M}$$

If $v \geq FC VJ$, then

$$C(v) = \frac{CJO}{(1 - FC)^{(1+M)}} \left[1 - FC(1 + M) + \frac{v}{VJ} M \right]$$

Temperature Dependence

TNOM = nominal temperature

T = analysis temperature

Tp = previous analysis temperature (TNOM if first temperature analysis)

The model quantities at the current analysis temperature are written without any suffix. The quantities at the previous temperature are denoted by the suffix p. The quantities at the nominal temperature are denoted by the suffix NOM.

$$IS = IS_p e^{\frac{q\left(\frac{T}{T_p} - 1\right)EG}{kT}} \left(\frac{T}{T_p}\right)^{XTI}$$

$$BF = BF_p \left(\frac{T}{T_p}\right)^{XTB}$$

$$BR = BR_p \left(\frac{T}{T_p}\right)^{XTB}$$

$$ISE = ISE_p e^{\frac{q\left(\frac{T}{T_p} - 1\right)EG}{NE kT}} \left(\frac{T}{T_p}\right)^{\frac{XTI}{NE}} \left(\frac{T}{T_p}\right)^{-XTB}$$

$$ISC = ISC_p e^{\frac{q\left(\frac{T}{T_p} - 1\right)EG}{NC kT}} \left(\frac{T}{T_p}\right)^{\frac{XTI}{NC}} \left(\frac{T}{T_p}\right)^{-XTB}$$

$$VJ = \frac{kT}{q} \ln \frac{N_A N_D}{n_i^2}$$

$$E_{gT} = 1.16 - \frac{7.02 \times 10^{-4} T^2}{T + 1108}$$

$$VJ = \frac{T}{T_p} \left\{ VJ_p - \left\{ -2 \frac{kT_p}{q} \left[\frac{3}{2} \ln \frac{T_p}{TNOM} + q \left(-\frac{E_{gp}}{2kT_p} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right] \right\} \right\} \\ + \left\{ -2 \frac{kT}{q} \left[\frac{3}{2} \ln \frac{T}{TNOM} + q \left(-\frac{E_g}{2kT} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right] \right\}$$

$$CJ = CJ_{NOM} \left\{ 1 + M \left[0.0004(T - TNOM) - \frac{VJ - VJ_{NOM}}{VJ_{NOM}} \right] \right\}$$

$$CJ = CJ_p \left\{ 1 + M \left[0.0004(T - TNOM) - \frac{VJ - VJ_{NOM}}{VJ_{NOM}} \right] \right\}$$

$$RE = RE_{NOM} [1 + TRE1(T - TNOM) + TRE2(T - TNOM)^2]$$

$$RC = RC_{NOM} [1 + TRC1(T - TNOM) + TRC2(T - TNOM)^2]$$

$$RB = RB_{NOM}[1 + TRB1(T - TNOM) + TRB2(T - TNOM)^2]$$

$$RBM = RBM_{NOM}[1 + TRB1(T - TNOM) + TRB2(T - TNOM)^2]$$

Diode Model

DC Current

$$\begin{aligned}
 i = & IS \cdot \left(e^{\frac{v}{(N \cdot V_t)}} - 1 \right) \cdot \sqrt{\frac{IKF}{IKF + IS \cdot \left(e^{\frac{v}{(N \cdot V_t)}} - 1 \right)}} + \\
 & + ISR \cdot \left(e^{\frac{v}{(NR \cdot V_t)}} - 1 \right) \cdot \sqrt{\left[\left(1 - \frac{V}{VJ} \right)^2 + 0.005 \right]^M} + \\
 & + ISP \cdot \left(e^{\frac{v}{(NP \cdot V_t)}} - 1 \right) \cdot \sqrt{\frac{IKFP}{IKFP + ISP \cdot \left(e^{\frac{v}{(NP \cdot V_t)}} - 1 \right)}} + \\
 & + ISRP \cdot \left(e^{\frac{v}{(NRP \cdot V_t)}} - 1 \right) \cdot \sqrt{\left[\left(1 - \frac{V}{VJP} \right)^2 + 0.005 \right]^{MP}}
 \end{aligned}$$

Each exponential component in the above equations may be rewritten as:

$$i_{ec} = I_o \cdot \left(e^{\frac{v}{(m \cdot V_t)}} - 1 \right)$$

and be calculated for different regimes of operation according to:

Large Forward Bias ($v > m \cdot V_t \cdot (I_n(I_{max}) - I_n(I_o))$)

$$i_{ec} = I_{max} \cdot \left[1 + \frac{1}{m \cdot V_t} (v - m \cdot V_t (I_n(I_{max}) - I_n(I_o))) \right]$$

Normal Region ($v \geq -BV$)

$$i_{ec} = I_o \cdot \left(e^{\frac{v}{m \cdot V_t}} - 1 \right)$$

Breakdown Region ($v \geq -BV - v_t \cdot (I_n(I_{max}) - I_n(I_o))$)

$$i_{ec} = I_o \cdot \left(e^{\frac{-BV}{(m \cdot v_t)}} - e^{\frac{-(BV + v)}{v_t}} \right)$$

Large Reverse Bias ($v < -BV - v_t \cdot (\ln(I_{\max}) - \ln(I_o))$)

$$i_{ec} = I_o \cdot e^{\frac{-BV}{(m \cdot V_t)}} - I_{\max} \left[1 - \frac{1}{V_t} (V + BV + V_t \cdot (\ln(I_{\max}) - \ln(I_o))) \right]$$

where $I_{\max} = 10^5 \text{A}$.

Diode Model (IBV and BV Model Parameters)

There is a dependency between the *IBV* and *BV* model parameters. If only the *IBV* parameter is specified then:

$$BV = N \times v_t \times \ln \left(1 + \frac{IBV}{IS} \right) + \frac{IBV}{g_{\min}}$$

If both *IBV* and *BV* are specified, then *BV* is adjusted through an iterative algorithm that ensures the exponential behavior in the breakdown region for the specified *IBV*.

If *IBV* is not specified, then *BV* is used as given or is set to its default value.

Diode Charge

$$q = TTi + \int_0^v C_{\text{bottom}}(v) dv + \int_0^v C_{\text{periphery}}(v) dv$$

If $v < FC \cdot V_J$

$$C(v) = \frac{CJO}{\left(1 - \frac{v}{V_J}\right)^M}$$

If $v \geq FC \cdot V_J$

$$C(v) = \frac{CJO}{(1 - FC)^{(1+M)}} \left[1 - FC(1 + M) + \frac{v}{V_J} M \right]$$

Temperature Dependence

TNOM = nominal temperature

T = analysis temperature

Tp = previous analysis temperature (TNOM if first temperature analysis)

The model quantities at the current temperature are written without any suffix. The quantities at the previous temperature are denoted by the suffix p. The quantities at the nominal temperature are denoted by the suffix NOM.

If CTA \neq 0.0

$$CJ = CJ_{NOM}[1 + CTA(T - TNOM)]$$

If CTP \neq 0.0

$$CJP = CJP_{NOM}[1 + CTP(T - TNOM)]$$

$$VJ = \frac{kT}{q} \ln \frac{N_A N_D}{n_i^2}$$

$$VJ = \frac{T}{T_p} \left\{ VJ_p - \left[-2 \frac{kT_p}{q} \left[\frac{3}{2} \ln \frac{T_p}{TNOM} + q \left(-\frac{E_{gp}}{2kT_p} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right] \right] \right\} \\ + \left\{ -2 \frac{kT}{q} \left[\frac{3}{2} \ln \frac{T}{TNOM} + q \left(-\frac{E_g}{2kT} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right] \right\}$$

$$CJ = CJ_{NOM} \{ 1 + M[0.0004(T - TNOM)]$$

$$- \frac{VJ - VJ_{NOM}}{VJ_{NOM}} \}$$

$$CJ = CJ_p \{1 + M[0.0004(T - TNOM)]$$

$$-\frac{VJ - VJ_{NOM}}{VJ_{NOM}}\}$$

$$VJP = \frac{kT}{q} \ln \frac{N_A N_D}{n_i^2}$$

$$VJP = \frac{T}{T_p} \left\{ VJP_p - \left[-2 \frac{kT}{q} p \left[\frac{3}{2} \ln \frac{T_p}{TNOM} + q \left(-\frac{E_{gp}}{2kT_p} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right] \right] \right\}$$

$$+ \left\{ -2 \frac{kT}{q} \left[\frac{3}{2} \ln \frac{T}{TNOM} + q \left(-\frac{E_g}{2kT} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right] \right\}$$

$$CJP = CJP_{NOM} \{1 + MP[0.0004(T - TNOM)]$$

$$-\frac{VJP - VJP_{NOM}}{VJP_{NOM}}\}$$

$$CJP = CJP_p \{1 + MP[0.0004(T - TNOM)]$$

$$-\frac{VJP - VJP_{NOM}}{VJP_{NOM}}\}$$

$$IS = IS_p e^{\frac{q\left(\frac{T}{T_p} - 1\right)^{EG}}{NkT}} \left(\frac{T}{T_p}\right)^{\frac{XTI}{N}}$$

$$ISP = ISP_p e^{\frac{q\left(\frac{T}{T_p} - 1\right)^{EG}}{NPkT}} \left(\frac{T}{T_p}\right)^{\frac{XTI}{NP}}$$

$$ISR = ISR_p e^{\frac{q\left(\frac{T}{T_p} - 1\right)^{EG}}{NRkT}} \left(\frac{T}{T_p}\right)^{\frac{XTI}{NR}}$$

$$ISRP = ISRP_p e^{\frac{q\left(\frac{T}{T_p} - 1\right)^{EG}}{NRPkT}} \left(\frac{T}{T_p}\right)^{\frac{XTI}{NRP}}$$

$$BV = BV_{NOM} [1 + TBV1 \cdot (T - TNOM) + TBV2 \times (T - TNOM)^2]$$

$$RS = RS_{NOM}[1 + TRS1 \cdot (T - TNOM) + TRS2 \times (T - TNOM)^2]$$

$$IKF = IKF_{NOM}[1 + TIKF \cdot (T - TNOM)]$$

JFET Model

DC Current

$$i_{ds} = 0 \text{ if } v_{gs} - V_{TO} \leq 0$$

$$= \text{BETA} (v_{gs} - V_{TO})^2 (1 + \text{LAMBDA} v_{ds})$$

if $0 < v_{gs} - V_{TO} < v_{ds}$

$$= 2\text{BETA} (1 + \text{LAMBDA} v_{ds}) v_{ds} (v_{gs} - V_{TO} - 1/2 v_{ds})$$

if $0 < v_{ds} \leq v_{gs} - V_{TO}$

Temperature Dependence

TNOM = nominal temperature

T = analysis temperature

Tp = previous analysis temperature (TNOM if first temperature analysis)

Model Equations

The model quantities at the current temperature are written without any suffix. The quantities at the previous temperature are denoted by the suffix p. The quantities at the nominal temperature are denoted by the suffix NOM.

$$IS = IS_p e^{\frac{q\left(\frac{T}{T_p} - 1\right)^{1.11}}{kT}}$$

$$PB = \frac{T}{T_p} \left\{ PB_p - \left\{ \left(-2 \frac{kT_p}{q} \right) \left[\frac{3}{2} \ln \frac{T_p}{TNOM} + q \left(-\frac{E_{gp}}{2kT_p} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right] \right\} \right\} \\ + \left\{ -2 \frac{kT}{q} \left[\frac{3}{2} \ln \frac{T}{TNOM} + q \left(-\frac{E_g}{2kT} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right] \right\}$$

$$CGS = CGS_{NOM} \left\{ 1 + 0.5 \left[0.0004(T - TNOM) - \frac{PB - PB_{NOM}}{PB_{NOM}} \right] \right\}$$

$$CGS = CGS_p \left\{ 1 + 0.5 \left[0.0004(T - TNOM) - \frac{PB - PB_{NOM}}{PB_{NOM}} \right] \right\}$$

$$CGD = CGD_{NOM} \left\{ 1 + 0.5 \left[0.0004(T - TNOM) - \frac{PB - PB_{NOM}}{PB_{NOM}} \right] \right\}$$

$$CGD = CGD_p \left\{ 1 + 0.5 \left[0.0004(T - TNOM) - \frac{PB - PB_{NOM}}{PB_{NOM}} \right] \right\}$$

MESFET Model

Parasitic g-d and g-s diodes use the same equations as described for junction diodes.

Level 1

This is an RCA quadratic model.

DC Current

$$v_p = V_{TO} + K_1(-v_{bs})^{1/2} - K_2 v_{bs}$$

If $v_{gs} > v_p$

$$i_{ds} = BETA(1 + LAMBDA v_{ds}) \times (v_{gs} - v_p)^2 \tanh(ALPHA v_{ds})$$

Charge

$$q_{ds} = \tau_{AU} i_{ds}$$

$$q_{gs} = \int_0^{v_{gs}} C(v) dv \quad \text{where } CO = CGSO$$

$$q_{gd} = \int_0^{v_{gd}} C(v) dv \quad \text{where } CO = CGDO$$

If $v < FC \ VBI$, then

$$C(v) = \frac{CO}{\left(1 - \frac{v}{VBI}\right)^M}$$

If $v \geq FC \text{ VBI}$, then

$$C(v) = \frac{CO}{(1 - FC)^{1+M}}$$

$$\times \left[1 - FC(1 + M) + \frac{v}{VBI}M \right]$$

$$BETA = BETA_{NOM}(T/TNOM)^{BEX}$$

Level 2

This is an RCA cubic model.

DC Current

$$v1 = vgs[1 + \text{GAMMA}(\text{VDSO} - vds)]$$

$$ids = \text{BETA}(A0 + A1v1 + A2v1^2 + A3v1^3)$$

$$\times \tanh(\text{ALPHA}vds)$$

Charge

If $A5 > 0$, then

$$qds = A5vdsids$$

If $\text{TAU} > 0$

$$qds = \text{TAU}ids$$

$$qgs = \int_0^{vgs} C(v)dv \quad \text{where CO} = \text{CGSO}$$

$$qgd = \int_0^{vgd} C(v)dv \quad \text{where CO} = \text{CGDO}$$

If $v < \text{FC VBI}$, then

$$C(v) = \frac{\text{CO}}{\left(1 - \frac{v}{\text{VBI}}\right)^M}$$

If $v \geq FC \text{ VBI}$, then

$$C(v) = \frac{CO}{(1 - FC)^{1+M}} \left[1 - FC(1+M) + \frac{v}{VBI} M \right]$$

Level 3

This is a Raytheon model.

DC Current

$$v_p = V_{TO} + K_1(-v_{bs})^{1/2} - K_2 v_{bs}$$

If $v_{gs} - v_p \leq 0$, then

$$i_{ds} = 0$$

If $v_{ds} < 3/ALPHA$, then

$$i_{ds} = \frac{BETA(v_{gs} - v_p)^2}{1 + B(v_{gs} - v_p)} \\ \times \left[1 - \left(\frac{1 - ALPHA v_{ds}}{3} \right)^3 \right] \\ \times (1 + LAMBDA v_{ds})$$

If $v_{ds} \geq 3/ALPHA$, then

$$i_{ds} = \frac{BETA(v_{gs} - v_p)^2}{1 + B(v_{gs} - v_p)} \\ \times (1 + LAMBDA v_{ds})$$

Charge

$$q_{ds} = TA U_{ids}$$

$$v_{eff2} = 0.5[v_{gs} + v_{gd} - \sqrt{(v_{gs} - v_{gd})^2 + DELTA^2}]$$

$$v_{eff1} = 0.5[v_{gs} + v_{gd} + \sqrt{(v_{gs} - v_{gd})^2 + DELTA^2}]$$

$$v_{eff} = 0.5[v_{eff1} + v_p + \sqrt{(v_{eff1} - v_p)^2 + DELTA_{EFF}^2}]$$

$$q_g = \int_0^{v_{eff}} C(v) dv + CGDN v_{eff2}$$

If $v < FC \text{ VBI}$, then

$$C(v) = \frac{CGSO}{\left(1 - \frac{v}{VBI}\right)^M}$$

If $v \geq \text{FC VBI}$

$$C(v) = \frac{\text{CGSO}}{(1 - \text{FC})^{1+M}} \left[1 - \text{FC}(1+M) + \frac{v}{\text{VBI}} M \right]$$

$$\begin{aligned} \Delta q_{gs} &= 0.5 \{ [qg(vgs, vgd) - qg(vgs_{old}, vgd)] \\ &\quad + [qg(vgs, vgd_{old}) - qg(vgs_{old}, vgd_{old})] \} \\ \Delta q_{gd} &= 0.5 \{ [qg(vgs, vgd) - qg(vgs, vgd_{old})] \\ &\quad + [qg(vgs_{old}, vgd) - qg(vgs_{old}, vgd_{old})] \} \end{aligned}$$

$$c_{gs} = \frac{\partial \Delta q_{gs}}{\partial v_{gs}}$$

$$c_{gd} = \frac{\partial \Delta q_{gd}}{\partial v_{gd}}$$

QOPT = 4

Charge

$$q_{ds} = \text{TAUids}$$

$$v_{eff2} = 0.5[v_{gs} + v_{gd} - \sqrt{(v_{gs} - v_{gd})^2 + \text{DELTA}^2}]$$

$$v_{eff1} = 0.5[v_{gs} + v_{gd} + \sqrt{(v_{gs} - v_{gd})^2 + \text{DELTA}^2}]$$

$$v_{eff} = 0.5[v_{eff1} + v_p + \sqrt{(v_{eff1} - v_p)^2 + \text{DELTAEFF}^2}]$$

$$q_g = \int_0^{v_{eff}} C(v) dv + \text{CGDN} \cdot v_{eff2}$$

If $v < \text{FC VBI}$, then

$$C(v) = \frac{\text{CGSO}}{\left(1 - \frac{v}{\text{VBI}}\right)^M}$$

If $v \geq \text{FC VBI}$, then

$$C(v) = \frac{\text{CGSO}}{(1 - \text{FC})^{1+M}} \left[1 - \text{FC}(1+M) + \frac{v}{\text{VBI}} M \right]$$

$$q_s = - \int_0^{v_{eff}} C(v) dv$$

$$q_d = -\text{CGDN} v_{eff2}$$

Temperature Dependence

If the capacitance temperature coefficients are zero, then the capacitances are computed using equations similar to junction diode capacitance equations.

Junction built-in potential equations are similar to the equations described for the junction diodes.

MOSFET Model

The MOSFET representation consists of the parasitic diodes between source-bulk and drain-bulk, parasitic resistances, and the intrinsic MOSFET. This section describes all the MOSFET model equations for the intrinsic MOSFET. The equations used for parasitic diodes are the same as the equations described in the section "Diode Model."

All Levels

This section provides equations that are common to all levels.

Bottom Junction Capacitance

$$CJ = \sqrt{\frac{q\epsilon_{si} \text{NSUB}}{2PB}}$$

Threshold Voltage

$$PHI = 2 \frac{k \text{TNOM}}{q} \ln\left(\frac{\text{NSUB}}{n_i}\right)$$

$$n_i = 1.45 \times 10^{16}$$

$$COX = \frac{\epsilon_{ox}}{TOX}$$

$$GAMMA = \frac{\sqrt{2q\epsilon_{si}\text{NSUB}}}{COX}$$

$$E_g = 1.16 - \frac{7.02 \times 10^4 \times \text{TNOM}^2}{\text{TNOM} + 1108.0}$$

If TPG = 1, then

$$\Phi_{ms} = -0.05 - 0.5E_g - 0.5PHI$$

If TPG = -1, then

$$\Phi_{ms} = 0.5E_g - 0.5\text{PHI}$$

$$v_{fb} = \Phi_{ms} - \frac{qN_{SS}}{C_{OX}}$$

$$V_{TO} = v_{fb} + \text{PHI} + \text{GAMMA}\sqrt{\text{PHI}}$$

$$\begin{aligned} v_{bi} &= V_{TO} - \text{GAMMA}\sqrt{\text{PHI}} \\ &= v_{fb} + \text{PHI} \end{aligned}$$

Device Dimensions

$$l' = L \times \text{LMLT} + \text{XL} - 2 \times \text{LD}$$

$$w' = W \times \text{WMLT} + \text{XW} - 2 \times \text{WD}$$

Mobility

If KP is not specified, then

$$\text{KP} = \text{UO} \cdot C_{OX}$$

$$\beta = \text{KP} \frac{w'}{l'}$$

$$\text{UO} = \frac{\text{KP}}{C_{OX}}$$

Depletion Layer Width

$$x_d = \sqrt{\frac{2\epsilon_{si}}{qN_{SUB}}}$$

Bias Quantities

If $v_{bs} \leq 0.0$, then

$$sarg = \sqrt{PHI - v_{bs}}$$

If $v_{bs} > 0.0$, then

$$sarg = \frac{\sqrt{PHI}}{1 + 0.5 \frac{v_{bs}}{PHI} + 0.375 \frac{v_{bs}^2}{PHI^2}}$$

If $v_{bs} - v_{ds} \leq 0.0$, then

$$barg = \sqrt{PHI - (v_{bs} - v_{ds})}$$

If $v_{bs} - v_{ds} > 0.0$, then

$$barg = \frac{\sqrt{PHI}}{1 + 0.5 \frac{(v_{bs} - v_{ds})}{PHI} + 0.375 \frac{(v_{bs} - v_{ds})^2}{PHI^2}}$$

Noise Equations

Thermal Noise generation in the Drain and Source parasitic resistors

$$\text{thermal_noise_} R_d = \frac{4KT}{RD}$$

$$\text{thermal_noise_} R_s = \frac{4KT}{RS}$$

Shot Noise

$$\text{shot_noise} = \frac{8KT \cdot g_m}{3}$$

Flicker Noise

$$\text{flicker_noise} = \frac{KF \cdot i_{ds}^{AF}}{COX \cdot W_{eff} \cdot l_{eff} \cdot f}$$

Level 1

This is a Shichman-Hodges model.

Threshold Voltage

$$v_{th} = v_{bi} + \text{GAMMA}s_{arg}$$

Drain Current

If $v_{ds} \leq v_{gs} - v_{th}$ (linear region), then

$$i_{ds} = \beta(1 + \text{LAMBDA}v_{ds})\left(v_{gs} - v_{th} - \frac{1}{2}v_{ds}\right)v_{ds}$$

If $v_{ds} > v_{gs} - v_{th}$ (saturation region), then

$$i_{ds} = \frac{1}{2}\beta(1 + \text{LAMBDA}v_{ds})(v_{gs} - v_{th})^2$$

Level 2

This is an analytical model.

Threshold Voltage

$$F_N = \frac{1}{8} \frac{2\pi\epsilon_{si} \text{DELTA}}{\text{COX} w'}$$

$$\text{argss} = \frac{1}{2} \frac{\text{XJ}}{l'} \left[\sqrt{1 + 2 \left(\frac{\text{xdbsarg}}{\text{XJ}} \right)} - 1 \right]$$

$$\text{argsd} = \frac{1}{2} \frac{\text{XJ}}{l'} \left[\sqrt{1 + 2 \left(\frac{\text{xdbarg}}{\text{XJ}} \right)} - 1 \right]$$

$$\gamma_{SD} = \text{GAMMA}(1 - \text{argss} - \text{argsd})$$

$$\text{cfs} = q\text{NFS}$$

$$Q_{dep} = \text{COX} \gamma_{SD} \text{sarg}$$

$$\text{cd} = \frac{\partial Q_{dep}}{\partial v_{bs}}$$

$$\text{xn} = 1 + \frac{\text{cfs}}{\text{COX}} + \frac{\text{cd}}{\text{COX}} + \text{FN}$$

$$v_{th} = v_{bi} + F_N(\text{PHI} - v_{bs}) + \gamma_{SD} \text{sarg} + \frac{kT}{q} \text{xn}$$

$$v_{bin} = v_{bi} + F_N(\text{PHI} - v_{bs})$$

Mobility Reduction

$$\mu_{\text{fact}} = \left[\frac{\epsilon_{si} \text{UCRIT}}{\text{COX}(\text{vgs} - \text{vth} - \text{UTRA} \text{vds})} \right]^{\text{UEXP}}$$

$$\mu_{\text{eff}} = \mu_{\text{fact}} \text{UO}$$

Saturation Voltage

$$\eta = 1 + F_N$$

$$\gamma_D = \frac{\gamma_{SD}}{\eta}$$

If NFS = 0.0, then

$$\text{vgsx} = \text{vgs}$$

If NFS > 0.0 and vgs ≥ vth, then

$$\text{vgsx} = \text{vgs}$$

If NFS > 0.0 and vgs < vth, then

$$\text{vgsx} = \text{vth}$$

If vmax = 0.0, then

$$\text{vdsat} = \frac{\text{vgsx} - \text{vbin}}{\eta} + \frac{1}{2} \gamma_D^2$$

$$\times \left[1 - \sqrt{1 + \left(\frac{4}{\gamma_D^2} \right) \left(\frac{\text{vgsx} - \text{vbin}}{\eta} \right) + \text{PHI} - \text{vbs}} \right]$$

If vmax > 0.0 then vdsat is computed by solving the fourth order polynomial obtained as follows:

Model Equations

$$v_{\max} =$$

$$\frac{\mu_{\text{eff}} \left\{ \left(-v_{\text{bin}} - \frac{1}{2} \eta v_{\text{dsat}} \right) v_{\text{dsat}} - \frac{2}{3} \gamma_{\text{SD}} \left[\sqrt[3]{\text{PHI} - (v_{\text{bs}} - v_{\text{dsat}})} - \sqrt[3]{\text{PHI} - v_{\text{bs}}} \right] \right\}}{l' [v_{\text{gs}} - v_{\text{bin}} - \eta v_{\text{dsat}} - \gamma_{\text{SD}} \sqrt{\text{PHI} - (v_{\text{bs}} - v_{\text{dsat}})}]}$$

$$xv = \frac{V_{\text{MAX}} l'}{\mu_{\text{eff}}}$$

$$x = \sqrt{v_{\text{dsat}} + \text{PHI} - v_{\text{bs}}}$$

$$v_1 = \frac{v_{\text{gs}} - v_{\text{bin}}}{\eta} + \text{PHI} - v_{\text{bs}}$$

$$v_2 = \text{PHI} - v_{\text{bs}}$$

$$xv = \frac{\left(v_1 - \frac{v_2}{2} - \frac{x^2}{2} \right) (x^2 - v_2) - \frac{2}{3} \gamma_D (x^3 - \sqrt[3]{v_2})}{v_1 - \gamma_D x - x^2}$$

$$A = \frac{4}{3} \gamma_D$$

$$B = -2(v_1 + xv)$$

$$C = -2\gamma_D xv$$

$$D = 2v_1(v_2 + xv) - v_2^2 - \frac{4}{3} \gamma_D \sqrt[3]{v_2}$$

$$x^4 + Ax^3 + Bx^2 + Cx + D = 0$$

Channel Length Modulation

If $\text{LAMBDA} > 0.0$, then

$$l_{\text{fact}} = 1 - \text{LAMBDA} v_{\text{ds}}$$

If $\text{LAMBDA} \leq 0.0$ and $\text{NSUB} > 0.0$ and $\text{VMAX} = 0.0$, then

$$l_{\text{fact}} = 1 - \frac{x_d}{l'} \left[\frac{v_{\text{ds}} - v_{\text{dsat}}}{4} + \sqrt{1 + \left(\frac{v_{\text{ds}} - v_{\text{dsat}}}{4} \right)^2} \right]^{1/2}$$

If $\text{LAMBDA} \leq 0.0$ and $\text{NSUB} > 0.0$ and $\text{VMAX} > 0.0$, then

$$l_{\text{fact}} = 1 - \frac{x_d}{\sqrt{\text{NEFF}} \, l'} \times \left\{ \left[\left(\frac{\text{VMAX} x_d}{2 \sqrt{\text{NEFF}} \, \mu_{\text{eff}}} \right)^2 + v_{\text{ds}} - v_{\text{dsat}} \right]^{1/2} - \frac{\text{VMAX} \, x_d}{2 \sqrt{\text{NEFF}} \, \mu_{\text{eff}}} \right\}$$

$$x_{\text{wb}} = x_d \sqrt{\text{PB}}$$

$$l_{\text{eff}} = l' l_{\text{fact}}$$

If $l_{\text{eff}} < x_{\text{wb}}$, then

$$l_e = \frac{x_{\text{wb}}}{1 + \frac{x_{\text{wb}} - l_{\text{eff}}}{x_{\text{wb}}}} \quad \text{f}$$

$$l_{\text{fact}} = \frac{l_{\text{eff}}}{l'}$$

Drain Current

$$\beta_{\text{eff}} = \beta \frac{\mu_{\text{fact}}}{l_{\text{fact}}}$$

If $v_{ds} \leq v_{dsat}$, then

$$v_{dsx} = v_{ds}$$

If $v_{ds} > v_{dsat}$, then

$$v_{dsx} = v_{dsat}$$

$$i_{ds} = \beta_{\text{eff}} \left\{ \left(v_{gsx} - v_{bin} - \frac{1}{2} \eta v_{dsx} \right) - \frac{2}{3} \gamma_{SD} \left[\sqrt[3]{\text{PHI} - (v_{bs} - v_{dsx})} - \sqrt[3]{\text{PHI} - v_{bs}} \right] \right\}$$

If $v_{gs} \leq v_{th}$ and $NFS > 0.0$, then

$$i_{ds} = i_{ds} \times e^{\frac{q}{kT} \frac{v_{gs} - v_{th}}{xn}}$$

Level 3

This is a semi-empirical model.

Threshold Voltage

$$F_N = \frac{1}{4} \frac{2\pi\epsilon_{si} \text{DELTA}}{\text{COX} \cdot w'}$$

$$\sigma = \frac{\Omega \cdot \text{ETA}}{\text{COX}(l')^3}$$

$$\Omega = 8.15 \times 10^{-22}$$

$$W_p = \text{xdsarg}$$

$$d_0 = 0.0631353$$

$$d_1 = 0.8013292$$

$$d_2 = 0.01110777$$

$$\frac{W_c}{XJ} = d_0 + d_1 \frac{W_p}{XJ} + d_2 \frac{W_p^2}{XJ}$$

$$F_S = 1 - \frac{XJ}{l'} \left\{ \frac{LD + W_c}{XJ} \left[1 - \left(\frac{\frac{W_p}{XJ}}{1 + \frac{W_p}{XJ}} \right)^2 \right]^{1/2} - \frac{LD}{XJ} \right\}$$

$$x_n = 1 + \frac{q_{\text{NFS}}}{\text{COX}} + \frac{\text{GAMMA} F_s \text{sarg} + F_N (\text{PHI} - v_{\text{bs}})}{2(\text{PHI} - v_{\text{bs}})}$$

$$v_{\text{th}} = v_{\text{fb}} + \text{PHI} + \sigma \cdot v_{\text{ds}} + \text{GAMMA} F_s \text{sarg}$$

$$+ F_N (\text{PHI} - v_{\text{bs}}) + \frac{kT}{q} x_n$$

Mobility Reduction

$$v_{gsx} = \max(v_{gs}, v_{th})$$

$$\mu_{\text{fact}} = \frac{1}{1 + \text{THETA}(v_{gsx} - v_{th})}$$

$$\mu_s = \mu_{\text{fact}} UO$$

Saturation Voltage

$$F_B = \frac{\text{GAMMA} F_S}{4 \text{sarg}} + F_N$$

$$v_{\text{dsat}} = \frac{v_{gsx} - v_{th}}{1 + F_B} + \frac{V_{\text{MAX}} l'}{\mu_s}$$

$$- \sqrt{\left(\frac{v_{gsx} - v_{th}}{1 + F_B} \right)^2 + \left(\frac{V_{\text{MAX}} l'}{\mu_s} \right)^2}$$

$$v_{\text{dsx}} = \min(v_{\text{ds}}, v_{\text{dsat}})$$

Drain Current

$$F_{\text{drain}} = \frac{1}{1 + \frac{\mu_s}{V_{\text{MAX}} l'} v_{\text{dsx}}}$$

$$i_{\text{ds}} = \beta \mu_{\text{fact}} F_{\text{drain}} \left(v_{gsx} - v_{th} - \frac{1 + F_B}{2} v_{\text{dsx}} \right) v_{\text{dsx}}$$

Channel Length Modulation

If $v_{\text{ds}} > v_{\text{dsat}}$, then the channel length modulation factor is computed.

If VMAX = 0.0 , then

$$\Delta l = x_d \sqrt{KAPPA(v_{ds} - v_{dsat})}$$

If VMAX > 0.0 , then

$$idsat = ids$$

$$g_{dsat} = idsat(1 - F_{drain}) \frac{\mu_s}{l' VMAX}$$

$$E_p = \frac{idsat}{g_{dsat} l'}$$

$$\Delta l = \sqrt{\left(\frac{E_p x_d^2}{2}\right)^2 + KAPPA x_d^2 (v_{ds} - v_{dsat})}$$

$$-\frac{x_d^2 E_p}{2}$$

If $\Delta l > \frac{1}{2} l'$, then

$$\Delta l = l' - \frac{(l')^2}{4\Delta l}$$

$$l_{fact} = \frac{1}{1 - \frac{\Delta l}{l'}}$$

$$ids = idsat l_{fact}$$

Subthreshold Conduction

$$ids = idsat \left(e^{\frac{q}{kT} \frac{v_{gs} - v_{th}}{xn}} \right)$$

Level 4

This is the BSIM model, a short channel model.

Threshold Voltage

$$\eta = \text{ETA} + \text{X2E}v_{bs} + \text{X3E}(v_{ds} - V_{DD})$$

$$v_{th} = V_{FB} + \text{PHI} + \text{K1} \sqrt{\text{PHI} - v_{bs}} - \text{K2}(\text{PHI} - v_{bs}) - \eta v_{ds}$$

Mobility Reduction

$$U_{gs} = U_0 + \text{X2U0} v_{bs}$$

$$U_{ds} = \frac{U_1 + \text{X2U1} v_{bs} + \text{X3U1}(v_{ds} - V_{DD})}{l'}$$

Effective Beta

$$\beta_{0_0} = \text{MUZ} \cdot \text{cox} \frac{w'}{l'}$$

$$\beta_{0_b} = \text{X2MZ} \cdot \text{cox} \frac{w'}{l'}$$

$$\beta_{\text{vds}=0} = \beta_{0_0} + \beta_{0_b} \text{vbs}$$

$$\beta_{\text{VDD}} = \text{MUS} \cdot \text{cox} \frac{w'}{l'}$$

$$\beta_{\text{VDD}_b} = \text{X2MS} \cdot \text{cox} \frac{w'}{l'}$$

$$\beta_{\text{vds} = \text{VDD}} = \beta_{\text{VDD}} + \beta_{\text{VDD}_b} \text{vbs}$$

$$\beta_{\text{VDD}_d} = \text{X3MS} \cdot \text{cox} \frac{w'}{l'}$$

$$\left. \frac{\partial \beta}{\partial \text{vds}} \right|_{\text{vds}=\text{VDD}} = \beta_{\text{VDD}_d}$$

If vds > VDD, then

$$\beta_0 = \beta_{\text{vds}=\text{VDD}} + \left. \frac{\partial \beta}{\partial \text{vds}} \right|_{\text{vds}=\text{VDD}} (\text{vds} - \text{VDD})$$

If $v_{ds} \leq VDD$

$$\beta_0 = \beta_{v_{ds}=0} + v_{ds} \left\{ \left[\frac{2(\beta_{v_{ds}=VDD} - \beta_{v_{ds}=0})}{VDD} - \frac{\partial \beta}{\partial v_{ds}} \right]_{v_{ds}=VDD} + \left[\frac{-\beta_{v_{ds}=VDD} + \beta_{v_{ds}=0} + \frac{\partial \beta}{\partial v_{ds}} \Big|_{v_{ds}=VDD} VDD}{VDD^2} \right] v_{ds} \right\}$$

$$\beta = \frac{\beta_0}{1 + U_{gs}(v_{gs} - v_{th})}$$

Saturation Voltage

$$g = 1 - \frac{1}{1.744 + 0.8364(PHI - v_{bs})}$$

$$a = 1 + \frac{gK1}{2\sqrt{PHI - v_{bs}}}$$

$$vc = \frac{U_{ds}(v_{gs} - v_{th})}{a}$$

$$k = \frac{1 + vc + \sqrt{1 + 2vc}}{2}$$

$$v_{dsat} = \frac{v_{gs} - v_{th}}{a\sqrt{k}}$$

Drain Current

If $v_{ds} \leq v_{dsat}$, then

$$v_z = v_{ds}$$

If $v_{ds} > v_{dsat}$, then

$$v_z = v_{dsat}$$

$$i_{ds} = \frac{\beta}{(1 + U_{ds} v_z)} \left[(v_{gs} - v_{th}) v_z - \frac{a}{2} v_z^2 \right]$$

Subthreshold Conduction

$$n = N_0 + N_B v_{bs} + N_D v_{ds}$$

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

$$i_{exp} = \beta_{0_0} v_t^2 e^{1.8} e^{\frac{v_{gs} - v_{th}}{N v_t}} \left(1 - e^{-\frac{v_{ds}}{v_t}} \right)$$

$$i_{limit} = 4.5 \beta_{0_0} v_t^2$$

$$i_{subt} = \frac{i_{limit} \times i_{exp}}{i_{limit} + i_{exp}}$$

$$i_{ds} = i_{ds} + i_{subt}$$

Geometry Dependence

Each model parameter has three components: reference value, channel length dependence, and channel width dependence. The reference value is indicated by the parameter name and length and width dependence component names are formed by prefixing l and w to the parameter name.

$$l' = L - DL$$

$$w' = W - DW$$

$$param = param + \frac{l_{param}}{l'} + \frac{w_{param}}{w'}$$

Level 5

These equations describe the BSIM2 deep-submicron model.

Geometry Dependence

Each model parameter has three components: reference value, channel length dependence, and channel width dependence. The reference value is indicated by the parameter name and length and width dependence component names are formed by prefixing l and w to the parameter name.

$$l' = L - DL$$

$$w' = W - DW$$

$$param = param + \frac{lparam}{l'} + \frac{wparam}{w'}$$

Threshold Voltage

$$v_{th} = VFB + PHI + K1\sqrt{PHI - vbs} - K2(PHI - vbs) - \eta \times vds$$

Mobility Reduction

$$U_{vert} = 1 + U_a(vgs - v_{th}) + U_b(vgs - v_{th})^2$$

$$U_b = UA0 + UAB \times vbs$$

$$U_b = UB0 + UBB \times vbs$$

$$U_{1s} = U1S0 + U1SB \times vbs$$

$$U = U_{vert} + U_1 \times vds$$

$$U_1 = U_{1s} \left(1 - \frac{U1D(vds - vdsat)^2}{vdsat^2} \right) \quad \text{if } vds < vdsat$$

$$= U_{1s} \quad \text{if } vds \geq vdsat$$

Drain-Induced Barrier Lowering

$$\eta = ETA0 + ETAB \times vbs$$

Drain Saturation Voltage

$$vdsat = \frac{vgs - vth}{a \sqrt{K_k}}$$

$$a = 1 + \frac{g \times K1}{2 \sqrt{PHI - vbs}}$$

$$g = 1 - \frac{1}{1.744 + 0.8364(PHI - vbs)}$$

$$K_k = \frac{1 + vc + \sqrt{1 + 2vc}}{2}$$

$$vc = \frac{U_{1s}(vgs - vth)}{a \times U_{vert}}$$

Impact Ionization

$$fr = 1 + ai \times e^{\left(\frac{-bi}{vds - vdsat} \right)} \quad \text{if } vds \geq vdsat$$

$$= 1 \quad \text{if } vds < vdsat$$

$$ai = AIO + AIB \times vbs$$

$$abi = BIO + BIB \times vbs$$

Drain Current

Strong Inversion $\{(vgs - vth) \geq VGHIGH\}$

Linear Region $(vds < vdsat)$

$$ids = \frac{\beta \left(vgs - vth - \frac{a}{2} vds \right) vds}{U}$$

Effective Beta

$$\beta = \beta_0 + \beta_1 \times \tanh\left(\frac{\beta_2 \times v_{ds}}{v_{dsat}}\right) + \beta_3(v_{ds} - \beta_4)v_{ds}^2$$

$$\beta_0 = \beta_{00} + \beta_{0b} \times v_{bs}$$

$$\beta_{00} = MU0 \times cox \times \frac{w'}{l'}$$

$$\beta_{0b} = MU0B \times cox \times \frac{w'}{l'}$$

$$\beta_1 = \beta_s - (\beta_0 + \beta_3 \times v_{dd} - \beta_4 \times v_{dd}^2)$$

$$\beta_s = \beta_{s0} + \beta_{sb} \times v_{bs}$$

$$\beta_0 = MUS0 \times cox \times \frac{w'}{l'}$$

$$\beta_{sb} = MUSB \times cox \times \frac{w'}{l'}$$

$$\beta_2 = MU20 + MU2B \times v_{bs} + MU2G \times v_{gs}$$

$$\beta_3 = \beta_{30} + \beta_{3b} \times v_{bs} + \beta_{3g} \times v_{gs}$$

$$\beta_{30} = MU30 \times cox \times \frac{w'}{l'}$$

$$\beta_{3b} = MU3B \times cox \times \frac{w'}{l'}$$

$$\beta_{3g} = MU3G \times cox \times \frac{w'}{l'}$$

$$\beta_4 = \beta_{40} + \beta_{4b} \times v_{bs} + \beta_{4g} \times v_{gs}$$

$$\beta_{40} = MU40 \times cox \times \frac{w'}{l'}$$

Saturation Region

($v_{ds} \geq v_{dsat}$)

$$i_{ds} = \frac{\beta(v_{gs} - v_{th})^2}{2a \times K_k \times U_{vert}} \times fr$$

Weak Inversion / Subthreshold**{(vgs - vth) <= VGLOW; VGLOW < 0}**

$$ids = \beta \times vtm^2 \times e^{\left(\frac{vgs - vth}{n \times vtm} + vof\right)} \times fr$$

Thermal Voltage

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

Subthreshold Swing

$$n = N0 + \frac{NB}{\sqrt{PHI - vbs}} + ND \times vds$$

Voltage Offset

$$vof = VOF0 + VOFB \times vbs + VOFD \times vds$$

Transition Region {VGLOW < (vgs - vth) < VGHIGH}

The drain current equation used in the transition region is the same as that in the strong-inversion region except for the replacement of $(vgs - vth)$ with the effective gate voltage, $vgeff$, which is described by a cubic spline function of vgs .

Effective Gate Voltage:

$$vgeff = c0 + c1 \times vgs + c2 \times vgs^2 + c3 \times vgs^3$$

The coefficients, $c0 - c3$, are determined from the boundary conditions. The boundary conditions for this cubic spline function are chosen so that the drain current, ids , and its first derivative, $dids/dvgs$, are continuous at both bounds: $(vgs - vth) = VGLOW$ and $(vgs - vth) = VGHIGH$.

Level 6

This is the ASPEC model.

Threshold Voltage

If **LGAMMA > 0** and **VBO = 0** , then

$$\text{scf} = 1 - \left(\sqrt{1 + \frac{2\text{LAMBDA} \text{ sarg}}{\text{LGAMMA}}} - 1 \right) \frac{\text{LGAMMA}}{l'}$$

Else **scf = 1**

$$\text{gw} = 1 + \frac{\text{NWM}}{w'} \text{xd sarg}$$

$$\text{gl} = 1 - \frac{\text{XJ}}{l'} \left[\sqrt{1 + \frac{2\text{LAMBDA} \sqrt{\text{PHI} - \text{vbs} + \text{SCM} \text{vds}}}{\text{XJ}}} + -1 \right]$$

If **VBO = 0**, then

$$\gamma' = \text{GAMMA}$$

If **VBO > 0** and **(-vbs) < VBO**, then

$$\gamma' = \text{GAMMA}$$

If **VBO > 0** and **(-vbs) > VBO**, then

$$\gamma' = \text{LGAMMA}$$

$$\gamma = \gamma' \text{gwglscf}$$

$$\gamma_1 = \text{GAMMA} \text{gwglscf}$$

If **VBO > 0** and **(-vbs) > VBO**, then

$$\text{vfb} = \text{vfb} + (\gamma_1 - \gamma) \sqrt{\text{VBO} + \text{PHI}}$$

If GAMMA $\neq \gamma$, then

$$v_{fb} = v_{fb} + (\text{GAMMA} - \gamma) \sqrt{\text{PHI}}$$

$$v_{fb} = v_{fb} - \frac{\text{NWE}}{w'} - \frac{\text{LD}}{l'} V_{SH} - \frac{\epsilon_{si}}{\text{COX}} \frac{1}{l'}$$

$$\times \text{FDS} \min(v_{ds}, V_{FDS}) + \text{UFDS} \max(v_{ds} - V_{FDS}, 0)$$

$$v_{te} = v_{fb} + \text{PHI} + \gamma \text{ sarg}$$

$$v_g = v_{gs} - v_{bs} - v_{fb}$$

$$v_{gdrive} = v_{gs} - v_{te}$$

Weak Inversion

For WIC = 0:

$$f_{\text{weak}} = 1$$

$$v_{on} = v_g$$

For WIC = 1:

$$V_x = \frac{kT}{q} \left(1 + \frac{q\text{NFS}}{\text{COX}} + \frac{\gamma}{2\text{sarg}} \right)$$

$$v_{on} = \max(v_g, \text{PHI} - v_{bs} + \gamma \text{ sarg} + V_x)$$

$$f_{\text{weak}} = e^{\frac{v_g - v_{on}}{V_x}}$$

For WIC = 2:

$$V_x = \frac{kT}{q} \left(1 + \frac{qNFS}{COX} + \frac{\gamma}{2sarg} \right)$$

$$von = \max(vg, PHI - vbs + \gamma \text{ sarg} + V_x)$$

$$voff = \max(vg, PHI - vbs + \gamma \text{ sarg} - PHI)$$

$$f_{weak} = \left[1 - \frac{von - voff}{V_x + PHI} \right]^{WEX}$$

Saturation Voltage

$$\eta = 1 + \frac{NEW}{w'}$$

$$vsat0 = \left\{ \left[\left(\frac{\eta\gamma}{2} \right)^2 + \frac{von + \frac{NWE(PHI - vbs)}{w'}}{\eta} \right]^{1/2} - \frac{\gamma}{2\eta} \right\}^2$$

If ECRIT > 0, then

$$vc = ECRIT \cdot l'$$

If VMAX > 0, then

$$vc = \frac{VMAX \cdot l'}{\mu_{eff}}$$

$$vsat = vsat0 + vc - \sqrt{(vsat0 + vbs - PHI)^2 + vc^2}$$

Alternate Saturation Voltage

If $KU > 1$, then

$$\alpha = \frac{ECRIT \cdot I'}{vgdrive}$$

$$fu = 1 - \frac{KU}{\sqrt{\alpha^2 + KU^2} + \alpha(KU - 1)}$$

$$fa = KAfu^{2MAL}$$

If $KU \leq 1$, then

$$fu = 1$$

$$fa = 1$$

$$vd = vds - vbs + PHI$$

$$vde = \min\left(\frac{vd}{fa}, vsat\right)$$

Mobility Reduction

$$vdse = vde + vbs - PHI$$

For $MOB = 0$:

$$\mu_{fact} = 1$$

For $MOB = 1$:

$$\mu_{fact} = \frac{1}{1 + F1(vg - vs - UTRAvdse)}$$

For MOB = 2:

$$\mu_{\text{fact}} = \left(\frac{\frac{F1 \epsilon_{\text{si}}}{\text{COX}}}{v_{\text{g}} - v_{\text{s}} - \text{UTRA} v_{\text{dse}}} \right)^{\text{UEXP}}$$

For MOB = 3:

If $v_{\text{gdrive}}^{\text{UEXP}} \leq V_{\text{F1}}$, then

$$FF = F1$$

If $v_{\text{gdrive}}^{\text{UEXP}} > V_{\text{F1}}$, then

$$FF = \text{UTRA}$$

$$F = FF \cdot v_{\text{gdrive}}^{\text{UEXP}}$$

If $V_{\text{F1}} > 0$ and $v_{\text{gdrive}}^{\text{UEXP}} > V_{\text{F1}}$, then

$$F = F + (F1 + \text{UTRA}) V_{\text{F1}}$$

$$\mu_{\text{fact}} = \frac{1}{F4 + F}$$

For MOB = 4 or MOB = 5:

If MOB = 4, then

$$v_{\text{crit}} = \text{ECRIT} \cdot l'$$

If MOB = 5, then

$$v_{\text{crit}} = \text{UTRA} \cdot l'$$

$$\mu_{\text{fact}} = \frac{1}{1 + \frac{v_{\text{gdrive}} \cdot \text{COX}}{F1 \epsilon_{\text{si}}} + \frac{v_{\text{dse}}}{v_{\text{crit}}} + F2 \text{sarg}}$$

Effective Mobility:

$$\mu_{\text{eff}} = \mu_{\text{fact}} UO$$

Channel Length Modulation

For CLM = 0:

$$\Delta l = 0$$

For CLM = 1:

If not specified:

$$LAMBDA = \sqrt{\frac{2\epsilon_{\text{si}}}{qNSUB}}$$

$$\Delta l = LAMBDA \sqrt{v_d - v_{de}} \left(\frac{v_{\text{sat}} + v_{\text{bs}} - PHI}{v_{\text{sat}0} + v_{\text{bs}} - PHI} \right)^{KL}$$

For CLM = 2:

$$\Delta l = \frac{\epsilon_{\text{si}}}{COX} \frac{v_d - v_{de}}{A1(v_d - v_g) + A2(v_g - v_{de} + v_{\text{bs}} - PHI)}$$

For CLM = 3:

$$\Delta l = LAMBDA \cdot fu^{2MCL} \times (\sqrt{v_d - v_{\text{fa}} - v_{\text{sat}} + KCL \cdot v_{\text{bs}} + PHI} - \sqrt{KCL \cdot v_{\text{bs}} + PHI})$$

For CLM = 4:

$$\Delta l = \left(\frac{2A1\epsilon_{\text{si}}}{qNSUB \ln\left(\frac{DND}{NSUB}\right)} \right)^{KL} [(v_d - v_{de} + PHI)^{KL} - PHI^{KL}]$$

Drain Current

$$\beta_{\text{eff}} = \mu_{\text{eff}} \beta \text{fa}^{2\text{MBL}} \text{f}_{\text{weak}} \frac{l'}{l' - \Delta l}$$

$$v_s = \text{PHI} - v_{bs}$$

$$i_{ds} = \beta_{\text{eff}} \left\{ v_{on}(v_{de} - v_s) - \frac{1}{2}(v_{de} - v_s) \times \right.$$

$$\left[v_{de} + v_s + \frac{\text{NWE}}{w'}(v_{de} - v_s) \right] - \frac{2}{3} \gamma \left(\sqrt[3]{v_{de}} - \sqrt[3]{v_s} \right) \}$$

Level 8

Threshold Voltage

This is an enhanced MOS2 (analytical) model.

$$F_N = \frac{1}{8} \frac{2\pi\epsilon_{si} \text{DELTA}}{\text{COX}_w}$$

$$\text{NSUB}_{\text{fact}} = 1 - \frac{\text{SNVB}}{\text{NSUB}} \text{vbs}$$

$$\gamma = \text{GAMMA} \sqrt{\text{NSUB}_{\text{fact}}}$$

$$\Phi = \text{PHI} + 2 \frac{kT}{q} \ln(\text{NSUB}_{\text{fact}})$$

$$xd' = \frac{xd}{\sqrt{\text{NSUB}_{\text{fact}}}}$$

$$\text{argss} = \frac{1}{2} \frac{\text{XJ}}{l'} \left[\sqrt{1 + \frac{2xd' \text{sarg}}{\text{XJ}}} - 1 \right]$$

$$\text{argsd} = \frac{1}{2} \frac{\text{XJ}}{l'} \left[\sqrt{1 + \frac{2xd' \text{barg}}{\text{XJ}}} - 1 \right]$$

$$\gamma_{SD} = \gamma(1 - \text{argss} - \text{argsd})$$

$$\text{vbin} = \text{VTO} - \text{GAMMA} \sqrt{\text{PHI}} + F_N(\text{PHI} - \text{vbs})$$

$$\text{xn} = 1 + \frac{q\text{NFS}}{\text{COX}} \frac{1}{2} \frac{\gamma_{SD}}{\text{sarg}} + \frac{q\epsilon_{si}}{\text{COX}^2} \frac{\text{SNVB}}{\gamma_{SD}} \frac{\text{sarg}}{\gamma_{SD}} + F_N(\text{PHI} - \text{vbs})$$

$$\text{vth} = \text{vbin} + \gamma_{SD} + \frac{kT}{q} \text{xnCAV}$$

Mobility Reduction

If $U_{EXP} > 0.0$, then

$$\mu_{fact} = \left(\frac{\epsilon_{si} U_{CRIT}}{COX(v_{gs} - v_{th})} \right)^{U_{EXP}}$$

If $U_{EXP} \leq 0.0$, then

$$\mu_{fact} = \frac{1}{1 + U_{CRIT}(v_{gs} - v_{th})}$$

Saturation Voltage

$$v_{gsx} = \max(v_{gs}, v_{th})$$

$$\eta = 1 + F_N$$

$$\gamma_D = \frac{\gamma_{SD}}{\eta}$$

$$v_{dsat}' = \frac{v_{gsx} - v_{bin}}{\eta} + \frac{1}{2} \gamma_D^2 \times$$

$$\left\{ 1 - \sqrt{1 + \frac{4}{\gamma_D^2} \times \left[\frac{v_{gsx} - v_{bin}}{\eta} + PHI - v_{bs} \right]} \right\}$$

$$v_{dsat} = v_{dsat}' + E_{CRIT} \cdot 1' - \sqrt{(v_{dsat}')^2 + (E_{CRIT} \cdot 1')^2}$$

Channel Length Modulation

If $v_{ds} > v_{dsat}$, then

$$l_{fact} = 1 - \frac{LAMBDA}{1 + LAM1} \cdot (v_{ds} - v_{dsat})$$

If $v_{ds} \leq v_{dsat}$, then

$$I_{fact} = 1$$

Drain Current

$$v_{dsx} = \min(v_{ds}, v_{dsat})$$

$$\beta_{eff} = KP \frac{\mu_{fact}}{I_{fact}} \frac{1}{1 + \frac{UTRA}{I'} v_{dsx}}$$

$$i_{ds} = \beta_{eff} \left\{ \left(v_{gsx} - v_{bin} - \frac{1}{2} \eta v_{ds} \right) - \frac{2}{3} \gamma_{SD} \times \left[\sqrt[3]{\text{PHI} - (v_{bs} - v_{dsx})} - \sqrt[3]{\text{PHI} - v_{bs}} \right] \right\}$$

If $v_{gs} \leq v_{th}$ and $NFS > 0.0$, then

$$i_{ds} = i_{ds} e^{\frac{q}{kT} \frac{v_{gs} - v_{th}}{xn}}$$

Level 10

These equations describe the BSIM3 deep-submicron model.

Threshold Voltage

$$\begin{aligned}
 V_{th} &= V_{TH0} + K1 \cdot (\sqrt{PHI - V_{bseff}} - \sqrt{PHI}) - (K2 \cdot V_{bseff}) + K1 \times \\
 &\left(\sqrt{1 + \frac{NLX}{L_{eff}}} - 1 \right) \sqrt{PHI} + (K3 + K3B \cdot V_{bseff}) \cdot \frac{TOX}{W_{eff} + W_O} \cdot PHI \\
 &- DVTOW \cdot \left(\exp\left(-DVT1W \cdot \frac{W_{eff} L_{eff}}{2l_{tw}}\right) + 2\exp\left(-DVT1W \cdot \frac{W_{eff} L_{eff}}{l_{tw}}\right) \right) (V_{BI} - PHI) \\
 &- DVTO \cdot \left(\exp\left(-DVT1 \cdot \frac{L_{eff}}{2l_t}\right) + 2\exp\left(-DVT1 \cdot \frac{L_{eff}}{l_t}\right) \right) (V_{BI} - PHI) \\
 &- \left(\exp\left(-DSUB \cdot \frac{L_{eff}}{2l_{t0}}\right) + 2\exp\left(-DSUB \cdot \frac{L_{eff}}{l_{t0}}\right) \right) (ETA0 + ETAB \cdot V_{bseff}) V_{ds} \\
 l_{tw} &= \sqrt{\epsilon_{si} X_{dep} / COX} (1 + DVT2W \cdot V_{bseff}) \\
 l_t &= \sqrt{\epsilon_{si} X_{dep} / COX} (1 + DVT2W \cdot V_{bseff}) \\
 l_{t0} &= \sqrt{\epsilon_{si} X_{dep0} / COX} \\
 X_{dep} &= \sqrt{\frac{2\epsilon_{si}(PHI - V_{bseff})}{q \cdot NCH}} \\
 X_{dep0} &= \sqrt{\frac{2\epsilon_{si} \cdot PHI}{q \cdot NCH}}
 \end{aligned}$$

Model Equations

If VTH0 is not specified in the .MODEL card, it is calculated using

$$V_{TH0} = V_{FB} + PHI + K1 \cdot \sqrt{PHI}$$

where $V_{FB} = -1V$ in the BSIM3 model.

If VTH0 is given:

$$V_{FB} = V_{TH0} - PHI - (K1 \cdot \sqrt{PHI})$$

If K1 and K2 are not given, they are calculated using:

$$K1 = GAMMA2 - (2 \cdot K2 \cdot \sqrt{PHI - V_{BM}})$$

$$K2 = (GAMMA1 - GAMMA2) \cdot \frac{\sqrt{PHI - V_{BX}} - \sqrt{PHI}}{2\sqrt{PHI}(\sqrt{PHI - V_{BM}} - \sqrt{PHI}) + V_{BM}}$$

If NCH is not given and GAMMA1 is given, NCH is calculated from:

$$NCH = \frac{GAMMA1 \cdot GAMMA1 \cdot COX \cdot COX}{2q\epsilon_{si}}$$

If both GAMMA1 and NCH are not given, NCH defaults to $1.7E17cm^{-3}$ and GAMMA1 is calculated from NCH.

VBI is calculated using:

$$VBI = \phi_t \ln\left(\frac{NCH \cdot 1e26}{(n_i)^2}\right)$$

$$\phi_t = \frac{K_B T}{q}$$

PHI is calculated using:

$$PHI = 2\phi_t \ln\left(\frac{NCH}{n_i}\right)$$

If GAMMA1 is not given, it is calculated using:

$$\text{GAMMA1} = \frac{\sqrt{2q\epsilon_{\text{si}} \cdot \text{NCH}}}{\text{COX}}$$

If GAMMA2 is not given, it is calculated using:

$$\text{GAMMA2} = \frac{\sqrt{2q\epsilon_{\text{si}} \cdot \text{NSUB}}}{\text{COX}}$$

If VBX is not given, it is calculated using:

$$\text{VBX} = \text{PHI} - \frac{q \cdot \text{NCH} \cdot \text{XT} \cdot \text{XT}}{2\epsilon_{\text{si}}}$$

$$V_{\text{bseff}} = V_{\text{bcm}} + 0.5 \cdot \left(V_{\text{bs}} - V_{\text{bcm}} - \delta_1 + \sqrt{(V_{\text{bs}} - V_{\text{bcm}} - \delta_1)^2 + 4\delta_1 \cdot V_{\text{bcm}}} \right)$$

$$V_{\text{bcm}} = 0.9 \left(\text{PHI} - \frac{K1 \cdot K1}{4 \cdot K2 \cdot K2} \right)$$

$$\delta_1 = 0.001$$

Effective

Vgs - Vth

$$V_{gsteff} = \frac{2\eta\phi_t \ln\left(1 + \exp\left(\frac{V_{gs_eff} - V_{th}}{2\eta\phi_t}\right)\right)}{1 + 2nC_{ox} \sqrt{\frac{2\Phi_s}{q\epsilon_{si}N_{ch}}} \exp\left(-\frac{V_{gs_eff} - V_{th} - (2 \cdot VOFF)}{2\eta\phi_t}\right)}$$

$$V_{gs_eff} = V_{FB} + PHI + \frac{q\epsilon_{si} \cdot NGATE}{COX \cdot COX} \left(\sqrt{1 + \frac{2 \cdot COX \cdot COX \cdot (V_{gs} - V_{FB} - PHI)}{q \cdot \epsilon_{si} \cdot NGATE}} - 1 \right)$$

$$n = 1 + N_{factor} \frac{C_d}{C_{ox}}$$

$$+ \frac{(C_{dsc} + C_{dscd}V_{ds} + C_{dscb}V_{bseff}) \left(\exp\left(-D_{vtl} \frac{L_{eff}}{2l_t}\right) + 2\exp\left(-D_{vtl} \frac{L_{eff}}{l_t}\right) \right)}{C_{ox}} + \frac{C_{it}}{C_{ox}}$$

$$C_d = \frac{\epsilon_{si}}{X_{dep}}$$

Mobility

For mobmod=1:

$$\mu_{eff} = \frac{\mu_0}{1 + (UA + UC \cdot V_{bseff}) \left(\frac{V_{gsteff} + 2V_{th}}{TOX} \right) + UB \cdot \left(\frac{V_{gsteff} + 2V_{th}}{TOX} \right)^2}$$

For mobmod=2:

$$\mu_{\text{eff}} = \frac{\mu_0}{1 + (U_A + U_C \cdot V_{\text{bseff}}) \left(\frac{V_{\text{gsteff}}}{\text{TOX}} \right) + U_B \cdot \left(\frac{V_{\text{gsteff}}}{\text{TOX}} \right)^2}$$

For mobmod=3:

$$\mu_{\text{eff}} = \frac{\mu_0}{1 + \left[U_A \cdot \left(\frac{V_{\text{gsteff}} + 2V_{\text{th}}}{\text{TOX}} \right) + U_C \cdot \left(\frac{V_{\text{gsteff}} + 2V_{\text{th}}}{\text{TOX}} \right)^2 \right] (1 + U_C \cdot V_{\text{bseff}})}$$

Drain Saturation Voltage

For $R_{\text{ds}} > 0$ or $\lambda \neq 1$:

$$V_{\text{dsat}} = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

$$a = A_{\text{bulk}} \cdot W_{\text{eff}}^2 V_{\text{sat}} \cdot \text{COX} \cdot R_{\text{DS}} + \left(\frac{1}{\lambda} - 1 \right) A_{\text{bulk}}$$

$$b = -[(V_{\text{gsteff}} + 2\phi_t) \left(\frac{2}{\lambda} - 1 \right) + A_{\text{bulk}} E_{\text{sat}} L_{\text{eff}} + 3A_{\text{bulk}} (V_{\text{gsteff}} + 2\phi_t) (W_{\text{eff}} V_{\text{sat}} \cdot \text{COX} \cdot R_{\text{DS}})]$$

$$c = [(V_{\text{gsteff}} + 2\phi_t) E_{\text{sat}} L_{\text{eff}} + 2(V_{\text{gsteff}} + 2\phi_t)^2 (W_{\text{eff}} V_{\text{sat}} \cdot \text{COX} \cdot R_{\text{DS}})]$$

$$\lambda = A1 \cdot V_{\text{gsteff}} + A2$$

For Rds=0 and $\lambda=1$:

$$V_{dsat} = \frac{E_{sat} L_{eff} (V_{gsteff} + 2\phi_t)}{A_{bulk} E_{sat} L_{eff} + (V_{gsteff} + 2\phi_t)}$$

$$A_{bulk} = \left(1 + \frac{K1}{2\sqrt{\phi_{HI} - V_{bseff}}}\right) \left\{ \frac{A0 \cdot L_{eff}}{L_{eff} + 2\sqrt{XJ \cdot X_{dep}}} \right\} \cdot$$

$$\left[1 - AGS \cdot V_{gsteff} \left(\frac{L_{eff}}{L_{eff} + 2\sqrt{XJ \cdot X_{dep}}} \right)^2 + \frac{B0}{W_{eff} + B1} \right] \times \frac{1}{1 + KETA \cdot V_{bseff}}$$

$$E_{sat} = \frac{2v_{sat}}{\mu_{eff}}$$

Effective Vds

$$V_{dseff} = V_{dsat} - 0.5 \left(V_{dsat} - V_{ds} - DELTA + \sqrt{(V_{dsat} - V_{ds} - DELTA)^2 + 4 \cdot DELTA \cdot V_{dsat}} \right)$$

Drain Current Expression

$$I_d = \frac{I_{d0}}{1 + \frac{R_{ds} I_{d0}}{V_{dseff}}} \left(1 + \frac{V_{ds} - V_{dseff}}{V_A} \right) \left(1 + \frac{V_{ds} - V_{dseff}}{V_{ASCBE}} \right)$$

$$I_{d0} = \frac{W_{eff} \mu_{eff} \cdot COX \cdot V_{gsteff} \left(1 - A_{bulk} \frac{V_{dseff}}{2(V_{gsteff} + 2\phi_t)} \right) V_{dseff}}{L_{eff} \left(1 + \frac{V_{dseff}}{E_{sat} L_{eff}} \right)}$$

$$V_A = V_{Asat} + \left(1 + \frac{PVAG \cdot V_{gsteff}}{E_{sat} L_{eff}} \right) \left(\frac{1}{V_{ACLM}} + \frac{1}{V_{ADIBLC}} \right)^{-1}$$

$$V_{ACLM} = \frac{A_{bulk} E_{sat} L_{eff} + V_{gsteff}}{PCLM \cdot A_{bulk} Elitl} (V_{ds} - V_{dseff})$$

$$V_{ADIBLC} = \frac{(V_{gsteff} + 2\phi_t)}{\theta_{rout} (1 + PDIBLCB \cdot V_{bseff})} \left(1 - \frac{A_{bulk} V_{dsat}}{A_{bulk} V_{dsat} + V_{gsteff} + 2\phi_t} \right)$$

$$\theta_{rout} = PDIBLC1 \cdot \left[\exp \left(-DROUT \cdot \frac{L_{eff}}{2l_{t0}} \right) + 2 \exp \left(-DROUT \cdot \frac{L_{eff}}{l_{t0}} \right) \right] + PDIBLC2$$

$$V_{ASCBE} = \frac{L_{eff}}{PSCBE2} \cdot \exp \left(\frac{PSCBE1 \cdot litl}{V_{ds} - V_{dseff}} \right)$$

$$V_{Asat} = \frac{E_{sat} L_{eff} + V_{dsat} + 2R_{DS} V_{sat} \cdot COX \cdot W_{eff} V_{gstef} \left(1 - \frac{A_{bulk} V_{dsat}}{2(V_{gstef} + 2\phi_t)} \right)}{\frac{2}{\lambda} - 1 + 2R_{DS} V_{sat} \cdot COX \cdot W_{eff} V_{gstef}}$$

$$litl = \sqrt{\frac{\epsilon_{si} \cdot TOX \cdot XJ}{\epsilon_{ox}}}$$

Substrate Current

$$I_{sub} = \frac{ALPHA0}{L_{eff}} (V_{ds} - V_{dseff}) \exp\left(-\frac{BETA0}{V_{ds} - V_{dseff}}\right) \frac{I_{ds0}}{1 + \frac{R_{ds} I_{ds0}}{V_{dseff}}} \left(1 + \frac{V_{ds} - V_{dseff}}{V_A} \right)$$

Drain/Source Resistance

$$R_{ds} = \frac{RDSW \cdot [1 + PRWG \cdot V_{gstreff} + PRWB \cdot (\sqrt{PHI - V_{bseff}} - \sqrt{PHI})]}{(W_{eff})^{WR}}$$

Effective Channel Length and Width

$$L_{\text{eff}} = L - 2dL$$

$$W_{\text{eff}} = W - 2dW$$

$$W'_{\text{eff}} = W - 2dW'$$

$$dW = dW' + DWG \cdot V_{\text{gsteff}} + DWB \cdot (\sqrt{\text{PHI} - V_{\text{bseff}}} - \sqrt{\text{PHI}})$$

$$dW' = W_{\text{INT}} + \frac{WL}{L_{\text{WLN}}} + \frac{WW}{W_{\text{WWN}}} + \frac{WWL}{L_{\text{WLN}} \cdot W_{\text{WWN}}}$$

$$dL = L_{\text{INT}} + \frac{LL}{L_{\text{LLN}}} + \frac{LW}{L_{\text{LWN}}} + \frac{LWL}{L_{\text{LLN}} \cdot L_{\text{LWN}}}$$

Temperature Effects

$$V_{th}(T) = V_{th}(TNOM) + \left(\frac{KT1 + KT1L}{L_{eff}} + KT2 \cdot V_{bseff} \right) \left(\frac{T}{TNOM - 1} \right)$$

$$\mu_0(T) = \mu_0(TNOM) \cdot (T/TNOM - 1)^{UTE}$$

$$v_{sat}(T) = VSAT - AT \cdot (T/TNOM - 1)$$

$$R_{dsw}(T) = RDSW + PRT \cdot (T/TNOM - 1)$$

$$U_a(T) = UA + UA1 \cdot (T/TNOM - 1)$$

$$U_b(T) = UB + UB1 \cdot (T/TNOM - 1)$$

$$U_c(T) = UC + UC1 \cdot (T/TNOM - 1)$$

Level 11

This is the CSIM model, a short channel model.

Threshold Voltage

$$v_{th} = V_{TO} + \text{GAMMA}(\sqrt{\text{PHI} - v_{bs}} - \sqrt{\text{PHI}}) \\ + \text{GAMMA2 } v_{bs} - \text{ETA } v_{ds}$$

Mobility Reduction

$$\text{beta} = \frac{\text{BETA0}}{1 + \text{THETA1}(v_{gs} - v_{th}) + \text{THETA3} \times (\sqrt{\text{PHI} - v_{bs}} - \sqrt{\text{PHI}})}$$

Saturation Voltage

$$g = 1 - \frac{1}{1.744 + 0.8364(\text{PHI} - v_{bs})}$$

$$a = 1 + \frac{g \text{ GAMMA } \text{GAMMAFF}}{2\sqrt{\text{PHI} - v_{bs}}}$$

$$v_c = \frac{\text{THETA2}(v_{gs} - v_{th})}{a}$$

$$k = \frac{1 + v_c + \sqrt{1 + 2v_c}}{2}$$

$$v_{dsat} = \frac{v_{gs} - v_{th}}{a\sqrt{k}}$$

Drain Current

If $v_{ds} \leq v_{dsat}$, then

$$v_z = v_{ds}$$

If $v_{ds} > v_{dsat}$, then

$$v_z = v_{dsat}$$

$$i_{ds} = \frac{\beta}{1 + \text{THETA2}} \times \left[(v_{gs} - v_{th})v_z - \frac{a}{2}v_z^2 \right]$$

Subthreshold Conduction

The subthreshold component is added if the parameter SUBTHFLAG is greater than 0.0.

$$n = \text{SUBEXP} + \text{SUBEXPB} \cdot v_{bs} + \text{SUBEXPD} \cdot v_{ds}$$

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

$$i_{exp} = \text{SUBMULT} \cdot \text{BETA0} \cdot v_t^2 \cdot e^{1.8 \cdot \frac{v_{gs} - v_{th}}{N v_t}} \left(1 - e^{-\frac{v_{ds}}{v_t}} \right)$$

$$i_{sl} = \text{SUBLIMIT} \cdot \text{BETA0} \cdot \frac{(3v_t)^2}{2}$$

$$i_{subt} = \frac{i_{sl} \times i_{exp}}{i_{sl} + i_{exp}}$$

$$i_{ds} = i_{ds} + i_{subt}$$

Geometry Dependence

Each model parameter has three components: reference value, channel length dependence and channel width dependence. The reference value is indicated by the parameter name, and length and width dependence component names are formed by appending l and w to the parameter name.

$$l_{\text{eff}} = L - 2\text{DELTA}L$$

$$w_{\text{eff}} = W - 2\text{DELTA}W$$

$$l_{\text{reff}} = L_{\text{REF}} - 2\text{DELTA}L$$

$$w_{\text{reff}} = W_{\text{REF}} - 2\text{DELTA}W$$

$$\text{param} = \text{paraml} \left(\frac{1}{l_{\text{eff}}} - \frac{1}{l_{\text{reff}}} \right) + \text{paramw} \left(\frac{1}{w_{\text{eff}}} - \frac{1}{w_{\text{reff}}} \right)$$

Temperature Dependence

TNOM = nominal temperature

T = analysis temperature

Tp = previous analysis temperature (TNOM if first temperature analysis)

Model Equations

The model quantities at the current analysis temperature are written without any suffix. The quantities at previous temperature are denoted by the suffix p. The quantities at the nominal temperature are denoted by the suffix NOM.

$$UO = UO_p \left(\frac{T}{T_p} \right)^{BEX}$$

$$KP = KP_p \left(\frac{T}{T_p} \right)^{BEX}$$

$$E_g = 1.16 - \frac{7.02 \times 10^{-4} T^2}{T + 1108.0}$$

$$n_i = 1.45 \times 10^{16} \left(\frac{T}{T_{NOM}} \right)^{1.5} e^{-\frac{E_g - E_{gNOM}}{2kT}}$$

$$PHI = 2 \frac{kT}{q} \ln \frac{NSUB}{n_i}$$

$$PHI = \frac{T}{T_p} \left\{ PHI_p - \left[-2 \frac{kT_p}{q} \left(\frac{3}{2} \ln \frac{T_p}{T_{NOM}} + q \left(-\frac{E_{gp}}{2kT_p} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right) \right] \right\} \\ + \left\{ -2 \frac{kT}{q} \left[\frac{3}{2} \ln \frac{T}{T_{NOM}} + q \times \left(-\frac{E_g}{2kT} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right] \right\}$$

$$vbi = vbi_p - 0.5(E_g - E_{gp}) + 0.5(PHI - PHI_p)$$

$$vfb = vbi - PHI$$

$$VTO = vbi + GAMMA \sqrt{PHI}$$

$$IS \propto e^{-\frac{EG}{kT}}$$

$$IS = IS_p e^{-\left(\frac{E_g}{kT} - \frac{E_{gp}}{kT}\right)}$$

$$JS = JS_p e^{-\left(\frac{E_g}{kT} - \frac{E_{gp}}{kT}\right)}$$

$$ISSW = ISSW_p e^{-\left(\frac{E_g}{kT} - \frac{E_{gp}}{kT}\right)}$$

$$JSSW = JSSW_p e^{-\left(\frac{E_g}{kT} - \frac{E_{gp}}{kT}\right)}$$

$$PB = \frac{kT}{q} \ln \frac{N_A N_D}{n_i^2}$$

$$PB = \frac{T}{T_p} \left\{ PB_p - \left[-2 \frac{kT_p}{q} \left(\frac{3}{2} \ln \frac{T_p}{T_{NOM}} + q \left(-\frac{E_{gp}}{2kT_p} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right) \right] \right\} \\ + \left\{ -2 \frac{kT}{q} \left[\frac{3}{2} \ln \frac{T}{T_{NOM}} + q \left(-\frac{E_g}{2kT} + \frac{E_{gNOM}}{2kT_{NOM}} \right) \right] \right\}$$

$$CJ = CJ_{NOM} \{ 1 + MJ[0.0004(T - T_{NOM})$$

$$- \frac{PB - PB_{NOM}}{PB_{NOM}} \} \left. \vphantom{\frac{PB - PB_{NOM}}{PB_{NOM}}} \right]$$

$$CJ = CJ_p \{ 1 + MJ[0.0004(T - T_{NOM})$$

$$- \frac{PB - PB_{NOM}}{PB_{NOM}} \} \left. \vphantom{\frac{PB - PB_{NOM}}{PB_{NOM}}} \right]$$

$$\begin{aligned} \text{CJSW} = & \text{CJSW}_p \{ 1 + \text{MJ}[0.0004(\text{T} - \text{TNOM}) \\ & - \frac{\text{PB} - \text{PB}_{\text{NOM}}}{\text{PB}_{\text{NOM}}}] \} \end{aligned}$$

$$\begin{aligned} \text{CBD} = & \text{CBD}_p \{ 1 + \text{MJ}[0.0004(\text{T} - \text{TNOM}) \\ & - \frac{\text{PB} - \text{PB}_{\text{NOM}}}{\text{PB}_{\text{NOM}}}] \} \end{aligned}$$

$$\begin{aligned} \text{CBS} = & \text{CBS}_p \{ 1 + \text{MJ}[0.0004(\text{T} - \text{TNOM}) \\ & - \frac{\text{PB} - \text{PB}_{\text{NOM}}}{\text{PB}_{\text{NOM}}}] \} \end{aligned}$$

$$\text{RS} = \text{RS}_{\text{NOM}}[1 + \text{TRS}(\text{T} - \text{TNOM})]$$

$$\text{RD} = \text{RD}_{\text{NOM}}[1 + \text{TRD}(\text{T} - \text{TNOM})]$$

$$\text{RSH} = \text{RSH}_{\text{NOM}}[1 + \text{TRSH}(\text{T} - \text{TNOM})]$$

Level 4

$$\text{MUZ} = \text{MUZ}_p \left(\frac{T}{T_p} \right)^{\text{BEX}}$$

$$\text{X2MZ} = \text{X2MZ}_p \left(\frac{T}{T_p} \right)^{\text{BEX}}$$

$$\text{MUS} = \text{MUS}_p \left(\frac{T}{T_p} \right)^{\text{BEX}}$$

$$\text{X2MS} = \text{X2MS}_p \left(\frac{T}{T_p} \right)^{\text{BEX}}$$

$$\text{X3MS} = \text{X3MS}_p \left(\frac{T}{T_p} \right)^{\text{BEX}}$$

$$\Delta\text{PHI} = \text{PHI}_p - \text{PHI}$$

$$\text{VFB} = \text{VFB}_p + \Delta\text{PHI}$$

Level 11

$$\text{BETA0} = \text{BETA0}_p \left(\frac{T}{T_p} \right)^{\text{BEX}}$$

$$\text{BETA0L} = \text{BETA0L}_p \left(\frac{T}{T_p} \right)^{\text{BEX}}$$

$$\text{BETA0W} = \text{BETA0W}_p \left(\frac{T}{T_p} \right)^{\text{BEX}}$$

$$\Delta \text{VTO} = \text{VTO} - \text{VTO}_p$$

$$\text{VTO} = \text{VTOL}_p + \Delta \text{VTO}$$

$$\text{VTOW} = \text{VTOLW}_p + \Delta \text{VTO}$$

TLEV = 1

In this case, the following parameters are not computed using the equations described above. The following equations are used instead.

TCV should be specified with proper sign, similar to VTO specification. The sign for P-channel devices is opposite to that for N-channel devices.

$$\text{VTO} = \text{VTO}(\text{TNOM}) - \text{TCV}(T - \text{TNOM})$$

$$\text{CJ} = \text{CJ}_{\text{TNOM}} [1 + \text{CTA}(T - \text{TNOM})]$$

Level 20

These equations describe the EKV MOSFET model.

Static Intrinsic Model Equations

Intrinsic model equations are presented for a N-channel MOSFET. For a P-channel MOSFET, the reasoning and the explanations are the same but it is necessary to inverse the signs of polarity and to inverse the doping's nature, i.e. P-channel is dealt with as a pseudo-N-channel. The EKV model is formulated as a 'single expression', which preserves continuity of first- and higher-order derivatives with respect to any terminal voltage, in the entire range of validity of the model.

Voltages are all referred to the local substrate:

$$V_G = V_{GB}$$

$$V_S = V_{SB}$$

$$V_D = V_{DB}$$

where, V_{GB} , V_{SB} , V_{DB} are *intrinsic gate-to-bulk, source-to-bulk and drain-to-bulk voltages*.

Drain-to-Source Current

$$I_{DS} = I_S \times (i_f - i_r')$$

$$I_S = 2 \cdot n \cdot \beta \cdot V_t^2$$

$$V_t = \frac{K \cdot T}{q}$$

Slope Factor

$$n = 1 + \frac{GAMMA}{2 \cdot \sqrt{V_p + PHI + 4 \cdot V_t}}$$

Transconductance Factor

$$\beta = KP \cdot \frac{W_{eff}}{L_{eq}} \cdot \frac{1}{1 + THETA \cdot V_p'}$$

Effective Channel Length And Width

$$W_{eff} = M \cdot (W + DW)$$

$$L_{eff} = (L + DL)$$

Note: *DL* and *DW* normally have a *negative* value due to the above definition.

Equivalent Channel Length And Velocity Saturation

$$\begin{aligned}
 L_{eq} &= 0.5 \cdot \left(L' + \sqrt{L'^2 + L_{min}^2} \right) \\
 L' &= L_{eff} - \Delta L + \frac{V_{DS} + V_{ip}}{UCRIT} \\
 L_{min} &= 0.1 \cdot L_{eff} \\
 V_{ip} &= \sqrt{V_{DSS}^2 + \Delta V^2} - \sqrt{(V_{DS} - V_{DSS})^2 + \Delta V^2} \\
 V_{DS} &= \frac{V_D - V_S}{2} \\
 \Delta V &= 4 \cdot V_t \cdot \sqrt{LAMBDA \cdot \left(\sqrt{i_f} - \frac{V_{DSS}}{V_t} \right) + \frac{1}{64}} \\
 V_{DSS} &= V_c \cdot \left[\sqrt{\frac{1}{4} + \frac{V_t}{V_c} \cdot \sqrt{i_f}} - \frac{1}{2} \right] \\
 V_c &= UCRIT \cdot L_{eff}
 \end{aligned}$$

Channel Length Modulation

$$\Delta L = LAMBDA \cdot L_c \cdot \ln \left(1 + \frac{V_{DS} - V_{ip}}{L_c \cdot UCRIT} \right)$$

$$L_c = \sqrt{\frac{\epsilon_0 \cdot \epsilon_{SI}}{COX}} \cdot XJ$$

Drain-to-Source Saturation Voltage

$$V'_{DSS} = V_c \cdot \left[\sqrt{\frac{1}{4} + \frac{V_t}{V_c} \cdot \left(\sqrt{i_f} - \frac{3}{4} \cdot \ln(i_f) \right)} - \frac{1}{2} \right] + V_t \cdot \left[\ln \left(\frac{V_c}{2 \cdot V_t} \right) - 1 \right]$$

Normalized Currents and Interpolation Function

Forward Normalized Current:

$$i_f = F \left[\frac{V_P - V_S}{V_t} \right]$$

Reverse Normalized Current:

$$i_r = F \left[\frac{V_P - V_{DS} - V_S - \sqrt{V'^2_{DSS} + \Delta V^2} + \sqrt{(V_{DS} - V'_{DSS})^2 + \Delta V^2}}{V_t} \right]$$

Large Signal Interpolation Function:

$$F(v) = \begin{cases} i_a = \frac{e^v}{1 + c_{A0} \cdot e^v}, & v < -3 \\ i_b = \frac{c_{B0} \cdot e^{v \cdot c_{B1}}}{1 + c_{B2} \cdot e^{v \cdot c_{B3}}}, & -3 \leq v < -1 \\ i_c = \left(\frac{c_{c0} \cdot e^{v \cdot c_{c1}}}{1 + c_{C2} \cdot e^{v \cdot c_{c3}}} \right)^2, & -1 \leq v < 2.5 \\ i_d = [c_{D0} + c_{D1} \cdot v - c_{D2} \cdot \ln(c_{D3} + v)]^2 & v \geq -3 \end{cases}$$

where, interpolation function coefficients:

$$c_{A0} = 0.936$$

$$c_{B0} = 1.0773087$$

$$c_{B1} = 1.0131373$$

$$c_{B2} = 0.78365565$$

$$c_{B3} = 0.74462624$$

$$c_{C0} = 1.6913059$$

$$c_{C1} = 0.60520877$$

$$c_{C2} = 1.1709916$$

$$c_{C3} = 0.47326778$$

$$c_{D0} = 1.6107939$$

$$c_{D1} = 0.5085409$$

$$c_{D2} = 0.76603547$$

$$c_{D3} = 2.8864104$$

Pinch-off voltage including short and narrow channel effects

$$V_P' = 0.5 \cdot \left(V_P + \sqrt{V_P^2 + 2 \cdot V_t^2} \right)$$

$$V_P = \begin{cases} V_G' - PHI - \gamma' \cdot \left(\sqrt{V_G'^2 + \left(\frac{\gamma'}{2} \right)^2} - \frac{\gamma'}{2} \right), & V_G' > 0 \\ -PHI, & V_G' \leq 0 \end{cases}$$

where,

$$V_G' = V_G - V_{TO} + PHI + GAMMA \cdot \sqrt{PHI}$$

$$\gamma' = 0.5 \cdot \left(\gamma^\circ + \sqrt{\left(\gamma^\circ \right)^2 + 0.01 \cdot GAMMA^2} \right)$$

$$\gamma^\circ = GAMMA - \frac{\epsilon_0 \cdot \epsilon_{SI}}{COX} \cdot \left[\left(\frac{LETA}{L + DL} - \frac{3 \cdot WETA}{W + DW} \right) \cdot \sqrt{V_{S(D)}' + PHI} + \frac{LETA}{L + DL} \cdot \sqrt{V_D' + PHI} \right]$$

$$V_{S(D)}' = 0.5 \cdot \left[V_{S(D)} - PHI + \sqrt{\left(V_{S(D)} + PHI \right)^2 + \left(4 \cdot V_t \right)^2} \right]$$

Impact Ionization Current

$$I_B = \begin{cases} I_{DS} \cdot \frac{IBA}{IBB} \cdot V_{ib} \cdot \exp\left(\frac{-IBB \cdot L_c}{V_{ib}}\right), & V_{ib} > 0 \\ 0, & V_{ib} \leq 0 \end{cases}$$

$$V_{ib} = V_D - V_S - IBN \cdot V_{DSS}$$

Quasi-Static Model Equations**Reverse Normalized Current For Intrinsic Capacitances**

$$i_r = F \left[\frac{V_P - V_D}{V_t} \right]$$

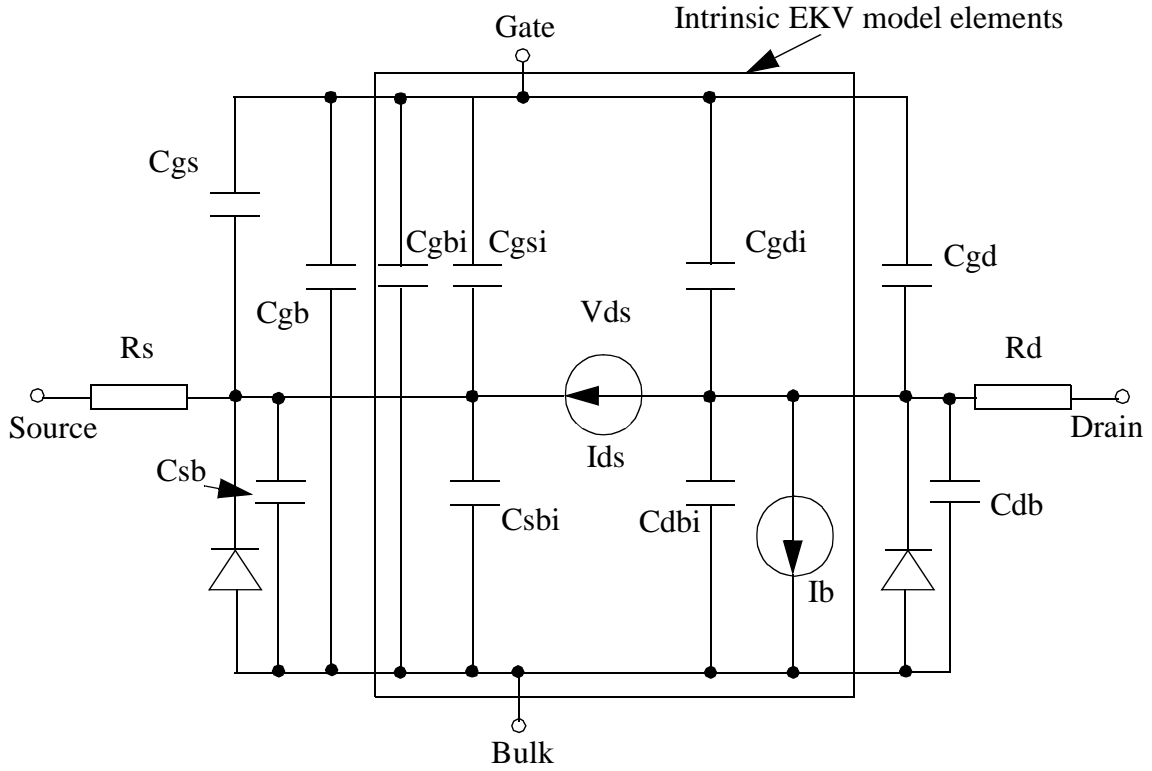
Interpolation Functions:

$$\begin{aligned}
c_{gsw}(i) &= \frac{i}{\sqrt{1 + 0.5 \cdot \sqrt{i} + i}} \\
c_{gss}(i_f i_r) &= \frac{2}{3} \cdot \left[1 - \frac{i_r}{(\sqrt{i_f} + \sqrt{i_r})^2} \right] \\
c_{gs}(i_f i_r) &= \frac{c_{gsw}(i_f) \cdot c_{gss}(i_f i_r)}{c_{gsw}(i_f) + c_{gss}(i_f i_r)} \\
c_{gbw} &= c_{gsw}(i_f) + c_{gsw}(i_r) \\
c_{gbs} &= \frac{2}{3} \cdot \left[1 + 2 \cdot \frac{\sqrt{i_f \cdot i_r}}{(\sqrt{i_f} + \sqrt{i_r})^2} \right] \\
c_{gb} &= \begin{cases} \frac{GAMMA}{2 \cdot \sqrt{V_P + PHI + GAMMA}} \cdot \left(1 - \frac{c_{gbw} \cdot c_{gbs}}{c_{gbw} + c_{gbs}} \right), & V_P > -PHI \\ \left(1 - \frac{c_{gbw} \cdot c_{gbs}}{c_{gbw} + c_{gbs}} \right), & V_P \leq -PHI \end{cases}
\end{aligned}$$

Intrinsic Capacitances Equations

$$\begin{aligned}
C_{ox} &= W_{eff} \cdot L_{eff} \cdot COX \\
C_{gsi} &= C_{ox} \cdot c_{gs}(i_f i_r) \\
C_{gdi} &= C_{ox} \cdot c_{gs}(i_r i_f) \\
C_{gbi} &= C_{ox} \cdot c_{gb} \\
C_{sbi} &= C_{ox} \cdot (n - 1) \cdot c_{gs}(i_f i_r) \\
C_{dbi} &= C_{ox} \cdot (n - 1) \cdot c_{gs}(i_r i_f)
\end{aligned}$$

Equivalent Circuit for Transient Analysis



EKV Noise Model

Channel Thermal Noise

$$S_{thermal} = 4 \cdot k \cdot T \cdot \gamma \cdot g_{ms} = 4 \cdot k \cdot T \cdot \gamma \cdot (g_m + g_{mbs} + g_{ds})$$

where, g_{ms} is the source transconductance and the noise factor defined as:

$$\gamma = \frac{1}{1 + i_f} \cdot \left[\frac{1}{2} \cdot (1 + \alpha) + \frac{2}{3} \cdot i_f \cdot \frac{1 + \alpha + \sqrt{\alpha}}{1 + \sqrt{\alpha}} \right]$$

$$\alpha = \frac{i_r}{i_f}$$

Flicker Noise

$$S_{flicker} = \frac{KF \cdot g_m^2}{W_{eff} \cdot L_{eff} \cdot COX \cdot f^{AF}}$$

Non-Quasi-Static (NQS) model equations

The EKV model includes a first order NQS model for small-signal (.AC) simulation. The equation of the NQS drain current is obtained from the quasi-static value of the drain current which is then 1st-order low-pass filtered according to:

$$I_{DS}^{(s)} = \frac{I_{DSq}^{(s)}}{1 + NQS \cdot s \cdot \tau}$$

where, the characteristic time constant depends on the bias according to:

$$\tau = \frac{\tau_0}{3} \cdot \frac{1}{\sqrt{1 + \frac{25}{16} \cdot \left(\frac{(\sqrt{i_f} + \sqrt{i_r})^3}{i_f + 3 \cdot \sqrt{i_f} \cdot i_r + i_r} \right)^2}}$$

$$\tau_0 = \frac{COX \cdot L_{eff}^2}{2 \cdot KP \cdot V_t} = \frac{L_{eff}^2}{2 \cdot \mu \cdot V_t}$$

The corresponding small-signal (.AC) transadmittances are then given by:

$$Y_m^{(s)} = \frac{g_m}{1 + NQS \cdot s \cdot \tau}$$

$$Y_{ms}^{(s)} = \frac{g_{ms}}{1 + NQS \cdot s \cdot \tau}$$

$$Y_{ds}^{(s)} = \frac{g_{ds}}{1 + NQS \cdot s \cdot \tau}$$

$$Y_{mbs}^{(s)} = Y_{ms}^{(s)} - Y_m^{(s)} - Y_{ds}^{(s)}$$

where, all transconductances and output conductance evaluated at the operating point.

Yang-Chatterjee Charge Model

$$\gamma = \frac{v_{th} - v_{bi}}{sarg}$$

$$\alpha_x = \frac{v_{gs} - v_{th}}{v_{dsat}}$$

$$C_o = COX w' l'$$

Accumulation Region ($v_{gs} \leq v_{fb} + v_{bs}$)

$$Q_g = C_o(v_{gs} - v_{fb} - v_{bs})$$

$$Q_b = -Q_g$$

$$Q_c = 0$$

$$Q_s = 0$$

$$Q_d = 0$$

Subthreshold Region ($v_{fb} + v_{bs} < v_{gs} \leq v_{th}$)

$$Q_g = C_o \frac{\gamma^2}{2} \left\{ -1 + \sqrt{\frac{4(v_{gs} - v_{fb} - v_{bs})}{\gamma^2}} \right\}$$

$$Q_b = -Q_g$$

$$Q_c = 0$$

$$Q_s = 0$$

$$Q_d = 0$$

Saturation Region ($v_{th} < v_{gs} \leq a_x v_{ds} + v_{th}$)

$$Q_g = C_o \left(v_{gs} - v_{fb} - \text{PHI} - \frac{v_{gs} - v_{th}}{3\alpha_x} \right)$$

$$Q_b = C_o \left[v_{fb} + \text{PHI} - v_{th} - \frac{(1 - \alpha_x)(v_{gs} - v_{th})}{3\alpha_x} \right]$$

$$Q_c = -\frac{2}{3} C_o (v_{gs} - v_{th})$$

$$Q_d = 0$$

$$Q_s = -\frac{2}{3} C_o (v_{gs} - v_{th})$$

Linear Region ($v_{gs} > a_x v_{ds} + v_{th}$)

$$Q_g = C_o \left[v_{gs} - v_{fb} - \text{PHI} - \frac{v_{ds}}{2} \right.$$

$$\left. + \frac{\alpha_x v_{ds}^2}{12 \left(v_{gs} - v_{th} - \frac{\alpha_x v_{ds}}{2} \right)} \right]$$

$$Q_b = C_o \left[v_{fb} + \text{PHI} - v_{th} + \frac{1 - \alpha_x}{2} v_{ds} \right.$$

$$\left. - \frac{(1 - \alpha_x) \alpha_x v_{ds}^2}{12 \left(v_{gs} - v_{th} - \frac{\alpha_x v_{ds}}{2} \right)} \right]$$

$$Q_c = -C_o \left[v_{gs} - v_{th} - \frac{\alpha_x}{2} v_{ds} \right.$$

$$\left. + \frac{\alpha_x^2 v_{ds}^2}{12 \left(v_{gs} - v_{th} - \frac{\alpha_x v_{ds}}{2} \right)} \right]$$

$$Q_d = -C_o \left[\frac{v_{gs} - v_{th}}{2} - \frac{3}{4} \alpha_x v_{ds} \right.$$

$$\left. + \frac{\alpha_x^2 v_{ds}^2}{8 \left(v_{gs} - v_{th} - \frac{\alpha_x v_{ds}}{2} \right)} \right]$$

$$Q_s = -C_o \left[\frac{v_{gs} - v_{th}}{2} - \frac{1}{4} \alpha_x v_{ds} \right.$$

$$\left. - \frac{\alpha_x^2 v_{ds}^2}{24 \left(v_{gs} - v_{th} - \frac{\alpha_x v_{ds}}{2} \right)} \right]$$

Meyer Charge Model

$$\gamma = \frac{v_{th} - v_{bi}}{sarg}$$

$$\Phi_f = \frac{1}{2}PHI$$

$$C_o = COXw'l'$$

$$v_{gb} = v_{gs} - v_{bs}$$

Cut-off Region ($v_{gs} \leq v_{th}$)

$$C_{GSC} = \frac{2}{3}C_o \frac{v_{gs} - (v_{th} - \Phi_f)}{\Phi_f}$$

If $v_{gs} \leq v_{th} - \phi_f$, then

$$C_{GS} = 0$$

If $v_{th} - \phi_f < v_{gs}$, $v_{ds} \geq 0.1$, then

$$C_{GS} = C_{GSC}$$

If $v_{th} - \phi_f < v_{gs}$, $v_{ds} < 0.1$, then

$$C_{GS} = C_{GSC} \left[\frac{0.1 + v_{ds}}{0.2} \right]$$

If $v_{gs} \leq v_{th} - \phi_f$, then

$$C_{GD} = 0$$

If $v_{th} - \phi_f < v_{gs}$, $v_{ds} \geq 0.1$, then

$$C_{GD} = 0$$

If $v_{th} - \phi_f < v_{gs}$, $v_{ds} < 0.1$

$$CGD = C_{GSC} \left[\frac{0.1 - v_{ds}}{0.2} \right]$$

If $v_{gb} > v_{fb}$, then

$$CGB = \frac{C_o}{\sqrt{1 + \frac{4}{\gamma^2}(v_{gb} - v_{fb})}}$$

If $v_{gb} \leq v_{fb}$

$$CGB = C_o$$

On Region ($v_{gs} > v_{th}$)

If $v_{gb} > v_{fb}$, then

$$C_{GBO} = \frac{C_o}{\sqrt{1 + \frac{4}{\gamma^2}(v_{gb} - v_{fb})}}$$

If $v_{gb} \leq v_{fb}$, then

$$C_{GBO} = C_o$$

If $v_{gs} < v_{th} + PHI$, then

$$CGB = C_{GBO} \frac{-v_{gs} + v_{th} + PHI}{PHI}$$

If $v_{gs} \geq v_{th} + PHI$, then

$$CGB = 0$$

Peak Region ($v_{gs} - v_{th} < 0.1$)
Where $v_{ds} < 0.1$

$$C_{GS1} = \frac{2}{3}C_o \frac{0.1 + v_{ds}}{0.2}$$

$$C_{GD1} = \frac{2}{3}C_o \frac{0.1 - v_{ds}}{0.2}$$

$$C_{GS2} = \frac{2}{3}C_o \left[1 - \frac{(0.1 - v_{ds})^2}{(0.2 - v_{ds})^2} \right]$$

$$C_{GD2} = \frac{2}{3}C_o \left[1 - \frac{0.01}{(0.2 - v_{ds})^2} \right]$$

$$C_{GS} = (C_{GS2} - C_{GS1}) \frac{v_{gs} - v_{th}}{0.1} + C_{GS1}$$

$$C_{GD} = (C_{GD2} - C_{GD1}) \frac{v_{gs} - v_{th}}{0.1} + C_{GD1}$$

Where $v_{ds} \geq 0.1$

$$C_{GS} = \frac{2}{3}C_o$$

$$C_{GD} = 0$$

Transition Region ($v_{gs} - v_{th} \geq 0.1$, $v_{ds} < 0.1$)

$$C_{GS} = \frac{2}{3}C_o \left[1 - \frac{(0.1 - v_{ds})^2}{(0.2 - v_{ds})^2} \right]$$

$$C_{GD} = \frac{2}{3}C_o \left[1 - \frac{0.01}{(0.2 - v_{ds})^2} \right]$$

Saturation Region ($v_{gs} - v_{th} \geq 0.1$, $v_{ds} \geq v_{dsat}$)

$$C_{GS} = \frac{2}{3}C_o$$

$$C_{GD} = 0$$

Linear Region ($v_{gs} - v_{th} \geq 0.1$, $v_{ds} < v_{dsat}$)

$$C_{GS} = \frac{2}{3}C_o \left[1 - \frac{(v_{dsat} - v_{ds})^2}{(2v_{dsat} - v_{ds})^2} \right]$$

$$C_{GD} = \frac{2}{3}C_o \left[1 - \frac{v_{dsat}^2}{(2v_{dsat} - v_{ds})^2} \right]$$

Ward-Dutton Charge Model

$$\gamma = \frac{v_{th} - v_{bi}}{s_{arg}}$$

$$C_o = COX w' l'$$

$$v_g = v_{gs} - v_{bs} - v_{bi} + PHI$$

$$v_s = PHI - v_{bs}$$

If $v_{ds} \leq v_{dsat}$, then

$$v_z = PHI - v_{bs} + v_{ds}$$

If $v_{ds} > v_{dsat}$, then

$$v_z = PHI - v_{bs} + v_{dsat}$$

Accumulation Region ($v_g \leq 0$)

$$Q_g = C_o v_g$$

$$Q_b = -Q_g$$

$$Q_c = 0$$

$$Q_d = 0$$

$$Q_s = 0$$

Cut-off Region ($v_{gs} \leq v_{th}$)

$$Q_g = C_o \gamma \left[\sqrt{\frac{\gamma^2}{4 + v_g}} - \frac{\gamma}{2} \right]$$

$$Q_b = -Q_g$$

$$Q_c = 0$$

$$Q_d = 0$$

$$Q_s = 0$$

On Region ($v_{gs} > v_{th}$)

$$i = v_g(\sqrt{v_z} + \sqrt{v_s}) - \frac{2}{3}\gamma(v_z + \sqrt{v_z}\sqrt{v_s} + v_s)$$

$$-\frac{1}{2}(v_z + v_s)(\sqrt{v_z} + \sqrt{v_s})$$

$$Q_g = C_o v_g - \frac{C_o}{i} \left\{ \frac{1}{2} v_g (v_z + v_s) (\sqrt{v_z} + \sqrt{v_s}) \right.$$

$$\left. -\frac{2}{5} \gamma [v_z^2 + \sqrt{v_z}\sqrt{v_s}(v_z + v_s) + v_s^2] \right.$$

$$\left. -\frac{1}{3}(\sqrt{v_z} + \sqrt{v_s})(v_z^2 + v_z v_s + v_s^2) \right\}$$

$$Q_b = \frac{-C_o \gamma}{i} \left\{ \frac{2}{3} v_g (v_z + \sqrt{v_z}\sqrt{v_s} + v_s) \right.$$

$$\left. -\frac{1}{2} \gamma (v_z + v_s) (\sqrt{v_z} + \sqrt{v_s}) \right.$$

$$\left. -\frac{2}{5} [v_z^2 + \sqrt{v_z}\sqrt{v_s}(v_z + v_s) + v_z v_s + v_s^2] \right\}$$

$$Q_c = -(Q_g + Q_b)$$

If $v_{ds} \leq v_{dsat}$, then

$$Q_d = \frac{1}{2} Q_c$$

If $v_{ds} > v_{dsat}$, then

$$Q_d = X Q_c \quad Q_c$$

If $v_{ds} \leq v_{dsat}$, then

$$Q_s = \frac{1}{2}Q_c$$

If $v_{ds} > v_{dsat}$, then

$$Q_s = (1 - X_{QC})Q_c$$

BSIM Charge Model

$$\gamma = \frac{v_{th} - V_{FB} - PHI}{sarg}$$

$$\alpha_x = \frac{v_{gs} - v_{th}}{v_{dsat}}$$

$$C_o = COX_w l'$$

Accumulation Region ($v_{gs} \leq V_{FB} + v_{bs}$)

$$Q_g = C_o(v_{gs} - V_{FB} - v_{bs})$$

$$Q_b = -Q_g$$

$$Q_d = 0$$

Subthreshold Region ($V_{FB} + v_{bs} < v_{gs} \leq v_{th}$)

$$Q_g = C_o \frac{\gamma^2}{2} \left\{ -1 + \sqrt{1 + \frac{4(v_{gs} - V_{FB} - v_{bs})}{\gamma^2}} \right\}$$

$$Q_b = -Q_g$$

$$Q_d = 0$$

Saturation Region ($v_{th} < v_{gs} \leq a_x v_{ds} + v_{th}$)

$$Q_g = C_o \left[v_{gs} - V_{FB} - \Phi_{SI} - \frac{(v_{gs} - v_{th})}{3\alpha_x} \right]$$

$$Q_b = C_o \left[V_{FB} + \Phi_{SI} - v_{th} - \frac{(1 - \alpha_x)(v_{gs} - v_{th})}{3\alpha_x} \right]$$

If $X_{PART} = 1$, then

$$Q_d = 0$$

If $X_{PART} = 0$, then

$$Q_d = -\frac{4}{15} C_o (v_{gs} - v_{th})$$

Linear Region ($v_{gs} > a_x v_{ds} + v_{th}$)

$$Q_g = C_o \left[v_{gs} - V_{FB} - \Phi_{SI} - \frac{v_{ds}}{2} + \frac{\alpha_x v_{ds}^2}{12 \left(v_{gs} - v_{th} - \frac{\alpha_x v_{ds}}{2} \right)} \right]$$

$$Q_b = C_o [V_{FB} + \Phi_{SI} - v_{th}]$$

$$+ \frac{1 - \alpha_x}{2} v_{ds} \frac{(1 - \alpha_x) \alpha_x v_{ds}^2}{12 \left(v_{gs} - v_{th} - \frac{\alpha_x v_{ds}}{2} \right)}$$

If $\text{XPART} = 1$, then

$$Q_d = -C_o \left[\frac{v_{gs} - v_{th}}{2} - \frac{3}{4} \alpha_x v_{ds} + \frac{\alpha_x^2 v_{ds}^2}{8 \left(v_{gs} - v_{th} - \frac{\alpha_x v_{ds}}{2} \right)} \right]$$

If $\text{XPART} = 0$, then

$$Q_d = -C_o \left\{ \frac{v_{gs} - v_{th}}{2} - \frac{\alpha_x v_{ds}}{2} + \frac{\alpha_x v_{ds}}{\left[v_{gs} - v_{th} - \frac{\alpha_x v_{ds}}{2} \right]^2} \right.$$

$$\left. \times \left[\frac{(v_{gs} - v_{th})^2}{6} - \frac{\alpha_x v_{ds} (v_{gs} - v_{th})}{8} + \frac{\alpha_x^2 v_{ds}^2}{40} \right] \right\}$$

BSIM2 Charge Model

$$C_o = c_{ox} \times w' \times l'$$

Accumulation Region ($v_{gs} < v_{bs} + V_{FB}$)

$$Q_g = C_o(v_{gs} - v_{bs} - V_{FB})$$

$$Q_b = -Q_g$$

$$Q_d = 0$$

Subthreshold Region ($v_{bs} + V_{FB} \leq v_{gs} \leq v_{th} + V_{GLOW}$)

$$Q_g = C_o(v_{gs} - v_{bs} - V_{FB}) \left(1 - \frac{v_{gs} - v_{bs} - V_{FB}}{v_{th} - v_{bs} - V_{FB}} + \frac{1}{3} \left(\frac{v_{gs} - v_{bs} - V_{FB}}{v_{th} - v_{bs} - V_{FB}} \right)^2 \right)$$

$$Q_b = -Q_g$$

$$Q_d = 0$$

Saturation Region ($v_{ds} \geq v_{dsat}$)

$$Q_g = \frac{2}{3} C_o \times v_{gst} + Q_{bulk}$$

$$Q_{bulk} = \frac{1}{3} C_o (v_{th} - v_{bs} - V_{FB})$$

$$Q_b = -Q_{bulk}$$

$$Q_d = \frac{4}{15} C_o \times v_{gst}$$

Linear Region ($v_{ds} < v_{dsat}$)

$$v_{dosat} = \frac{v_{ds}}{v_{dsat}}$$

$$Q_g = \frac{2}{3}C_o \times v_{gst} \left(\frac{3(1 - v_{dosat}) + v_{dosat}^2}{2 - v_{dosat}} \right) + Q_{bulk}$$

$$Q_b = -Q_{bulk}$$

$$Q_d = \frac{1}{3}C_o \times v_{gst} \left(\frac{3(1 - v_{dosat}) + v_{dosat}^2}{2 - v_{dosat}} + \frac{v_{dosat}(1 - v_{dosat}) + 0.2v_{dosat}^2}{(2 - v_{dosat})^2} \right)$$

BSIM3 Charge Model

Dimension Dependence

$$\delta W_{eff} = DWC + \frac{WL}{L^{WLN}} + \frac{WW}{W^{WWN}} + \frac{WWL}{L^{WLN} \cdot W^{WWN}}$$

$$\delta L_{eff} = DLC + \frac{LL}{L^{LLN}} + \frac{LW}{W^{LWN}} + \frac{LWL}{L^{LLN} \cdot W^{LWN}}$$

$$L_{active} = L - 2\delta L_{eff}$$

$$W_{active} = W - 2\delta W_{eff}$$

Overlap Capacitance

Bulk Overlap Capacitance

If CGBO is not given then it is calculated using:

$$C_{GBO} = 2 \cdot DWC \cdot COX$$

$$\frac{Q_{ov,s}}{L_{active}} = C_{GBO} \cdot V_{bs}^a \cdot V_{gs}^b$$

Source Overlap Capacitance

$$V_{gs,overlap} = 0.5 \cdot \left((V_{gs} + \delta_1) - \sqrt{(V_{gs} + \delta_1)^2 + 4\delta_1} \right), \delta_1 = 0.02$$

$$\frac{Q_{overlap,s}}{W_{active}} = CGSO \cdot V_{gs} - CGSL \cdot \left[V_{gs} - V_{gs,overlap} + \frac{CKAPPA}{2} \left(\sqrt{1 - \frac{4V_{gs,overlap}}{CKAPPA}} - 1 \right) \right]$$

If CGSO is not given then it is calculated using:

If (DLC is given and is greater than CGSL/COX) **THEN**

$$C_{GSO} = DLC \cdot COX - CGSL$$

ELSE

$$C_{GSO} = 0.6 \cdot XJ \cdot COX$$

$$C_{GSO} = C_{GSO} + CF$$

where, if CF is not given then it is calculated using:

$$CF = \frac{2\epsilon_{ox}}{\pi} 1n \left(1 + \frac{4 \cdot 10^{-7}}{TOX} \right)$$

Drain Overlap Capacitance

$$V_{gd,overlap} = 0.5 \cdot \left((V_{gd} + \delta_2) - \sqrt{(V_{gd} + \delta_2)^2 + 4\delta_2} \right) \delta_2 = 0.02$$

$$\frac{Q_{overlap,d}}{W_{active}} = CGDO \cdot V_{gd} - CGDL \cdot \left[V_{gd} - V_{gd,overlap} + \frac{CKAPPA}{2} \left(\sqrt{1 - \frac{4V_{gd,overlap}}{CKAPPA}} - 1 \right) \right]$$

If CGDO is not given then it is calculated using:

If (DLC is given and is greater than CGDL/COX) **THEN**

$$C_{GDO} = DLC \cdot COX - CGDL$$

ELSE

$$C_{GDO} = 0.6 \cdot XJ \cdot COX$$

$$CGDO = C_{GDO} + CF$$

Gate Overlap Capacitance

$$Q_{overlap,g} = -(Q_{overlap,s} + Q_{overlap,d})$$

Intrinsic Charges

$$Q_g = -(Q_{inv} + Q_{acc} + Q_{subo} + \delta Q_{sub})$$

$$Q_b = +(Q_{acc} + Q_{subo} + \delta Q_{sub})$$

$$C_0 = L_{active} \cdot W_{active} \cdot COX$$

$$V_{dsat,cv} = \frac{V_{gsteffCV}}{A'_{bulk}}$$

$$A'_{bulk} = A_{bulk} \left(1 + \left(\frac{CLC}{L_{eff}} \right)^{CLE} \right)$$

$$V_{gsteffCV} = n\phi_t 1n \left[1 + \exp \left(\frac{V_{gs_eff} - V_{th}}{2(n\phi_t)} \right) \right]$$

If CAPMOD=1, **THEN**

Model Equations

Q_{sub0}, Q_{acc} are divided into two regions:

if $V_{gs_eff} - V_{bseff} \leq V_{FB}$, **then**

$$Q_{sub0} = 0$$

$$Q_{acc} = -C_0 \cdot (V_{gs_eff} - V_{FB} - V_{bseff} - V_{gsteffCV})$$

else $\{V_{gs_eff} - V_{bseff} > V_{FB}\}$

$$Q_{sub0} = -C_0 \frac{K1 \cdot K1}{2} \left(-1 + \sqrt{1 + \frac{4V_{gs_eff} - V_{FB} - V_{bseff} - V_{gsteffCV}}{K1 \cdot K1}} \right)$$

$$Q_{acc} = 0$$

endif

$\delta Q_{sub}, Q_{inv}$ have a single region for V_{gs} but two regions for V_{ds}

if $V_{ds} \geq V_{dsat,cv}$, **then** {Saturation Region}

$$Q_{inv} = -\frac{2}{3} C_0 V_{gsteffCV}$$

$$\delta Q_{sub} = -\frac{1}{3} C_0 V_{gsteffCV} \left(1 - \frac{1}{A'_{bulk}} \right)$$

50/50 Charge Partition

$$Q_s = -\frac{1}{3} C_0 V_{gsteffCV}$$

40/60 Charge Partition

$$Q_s = -\frac{2}{5} C_0 V_{gsteffCV}$$

0/100 Charge Partition

$$Q_s = -\frac{2}{3} C_0 V_{gsteffCV}$$

else {Linear Region}

$$Q_{inv} = -C_0 \left[(V_{gsteff}CV - 0.5A'_{bulk} V_{ds}) + \frac{A'^2_{bulk} V_{ds}^2}{12(V_{gsteff}CV - 0.5A'_{bulk} V_{ds})} \right]$$

$$\delta Q_{sub} = C_0 \cdot (1 - A'_{bulk}) \left[0.5V_{ds} - \frac{A'_{bulk} V_{ds}^2}{12(V_{gsteff}CV - 0.5A'_{bulk} V_{ds})} \right]$$

50/50 Charge Partition

$$Q_s = 0.5(Q_{inv} - \delta Q_{sub})$$

40/60 Charge Partition

$$Q_s = -\frac{C_0}{2(V_{gsteff}CV - 0.5A'_{bulk} V_{ds})^2}$$

$$\left[V_{gsteff}^3 CV - \frac{4}{3} V_{gsteff}^2 CV (A'_{bulk} V_{ds}) + \frac{2}{3} V_{gsteff} CV (A'_{bulk} V_{ds})^2 - \frac{2}{15} (A'_{bulk} V_{ds})^3 \right]$$

0/100 Charge Partition

$$Q_s = -C_0 \left[\frac{V_{gsteff}CV}{2} + \frac{A'_{bulk} V_{ds}}{4} - \frac{(A'_{bulk} V_{ds})^2}{24(V_{gsteff}CV - 0.5A'_{bulk} V_{ds})} \right]$$

endif

ELSE {CAPMOD=2}

$$Q_{acc} = -C_0(V_{FBeff} - V_{FB})$$

$$V_{FBeff} = V_{FB} - 0.5 \cdot (V_3 + \sqrt{V_3^2 + 4\delta_3 \cdot V_{FB}})$$

$$V_3 = V_{FB} - V_{gs_eff} + V_{bseff} - \delta_3, \delta_3 = 0.02$$

$$Q_{sub0} = -C_0 \frac{K1 \cdot K1}{2} \left(-1 + \sqrt{1 + \frac{4(V_{gs_eff} - V_{FBeff} - V_{bseff} - V_{gsteffCV})}{K1 \cdot K1}} \right)$$

$$V_{cveff} = V_{dsat,cv} - 0.5 \cdot (V_4 + \sqrt{V_4^2 + 4\delta_4 V_{dsat,cv}})$$

$$V_4 = V_{dsat,cv} - V_{ds} - \delta_4, \delta_4 = 0.02$$

$$Q_{inv} = -C_0 \left[(V_{gsteffCV} - 0.5A'_{bulk} V_{cveff}) + \frac{A'^2_{bulk} V_{cveff}^2}{12(V_{gsteffCV} - 0.5A'_{bulk} V_{cveff})} \right]$$

$$\delta Q_{sub} = C_0 \cdot (1 - A'_{bulk}) \left[0.5V_{cveff} - \frac{A'_{bulk} V_{cveff}^2}{12(V_{gsteffCV} - 0.5A'_{bulk} V_{cveff})} \right]$$

50/50 Charge Partition

$$Q_s = 0.5(Q_{inv} - \delta Q_{sub})$$

40/60 Channel-Charge Partition

$$Q_s = -\frac{C_0}{2(V_{gsteffCV} - 0.5A'_{bulk} V_{cveff})^2}$$

$$\left[V_{gsteffCV}^3 - \frac{4}{3} V_{gsteffCV}^2 (A'_{bulk} V_{cveff}) + \frac{2}{3} V_{gsteffCV} (A'_{bulk} V_{cveff})^2 - \frac{2}{15} (A'_{bulk} V_{cveff})^3 \right]$$

0/100 Charge Partition

$$Q_s = -C_0 \left[\frac{V_{gsteff} CV}{2} + \frac{A'_{bulk} V_{cveff}}{4} - \frac{(A'_{bulk} V_{cveff})^2}{24(V_{gsteff} CV - 0.5 A'_{bulk} V_{cveff})} \right]$$

ENDIFF

$$Q_d = -(Q_g + Q_b + Q_s)$$

BSIM3 Non-Quasi-Static (NQS) Model

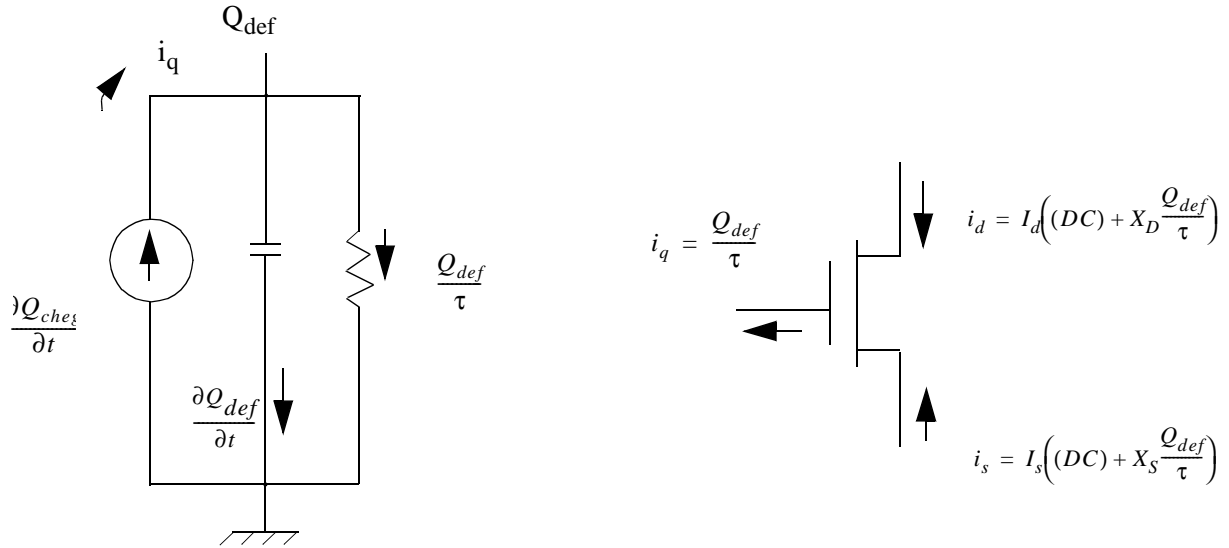
Quasi-static equilibrium channel charge:

$$Q_{cheq} = -(Q_g + Q_b)$$

Actual channel charge:

$$Q_{ch} = Q_{cheq} - Q_{def}$$

The state variable, Q_{def} , is an additional node created to keep track of the amount of deficit (or surplus) channel charge necessary to achieve equilibrium. The Q_{def} obtained from the subcircuit below:



by solving of the following equation:

$$\frac{\partial Q_{def}}{\partial t} = i_q - \frac{Q_{def}}{\tau} = -\frac{\partial(Q_g + Q_b)}{\partial t} - \frac{Q_{def}}{\tau},$$

with initial condition $Q_{def}(t = 0) = 0$.

The derivative of Q_{def} with respect to time is the gate charging current. This current is partitioned into separate drain and source current components. The elements in the NQS subcircuit above calculate as:

$$i_q = \frac{\partial Q_{cheq}}{\partial t} = -\frac{\partial(Q_g + Q_b)}{\partial t}$$

$$g_\tau = \frac{1}{\tau} = \frac{1}{\tau_{drift}} + \frac{1}{\tau_{diff}}$$

$$\tau_{drift} = \frac{\xi}{|Q_{cheq}|}$$

$$\xi = \frac{COX \cdot W_{active} L_{active}^3}{\mu_0(TNOM)(ELM)}$$

$$\tau_{diff} = \frac{L_{active}^2}{16\mu_0(TNOM)\phi_t}$$

and

$$i_d = I_{d(DC)} + X_D g_\tau Q_{def}$$

$$i_s = I_{s(DC)} + X_S g_\tau Q_{def}$$

$$i_g = -g_\tau Q_{def}$$

$$X_S = 0.6, X_D = 0.4, X_S + X_D = 1$$

BSIM3 Noise Model

Channel Thermal Noise

if NOIMOD=1:

$$\text{channel_thermal_noise} = \frac{8KT \cdot (g_m + g_{ds} + g_{mbs})}{3}$$

if NOIMOD=2:

$$\text{channel_thermal_noise} = \frac{4KT \cdot \mu_{eff} \cdot (-Q_{inv})}{L_{eff} \cdot L_{eff}}$$

$$Q_{inv} = -W_{eff} \cdot L_{eff} \cdot COX \cdot V_{gsteff} \left(1 - \frac{0.5A_{bulk} V_{dseff}}{(V_{gsteff} + 2 \cdot \Phi_t)} \right)$$

Flicker Noise

If **NOIMOD=1**:

$$\text{flicker_noise} = \frac{KF \cdot id_s^{AF}}{COX \cdot W_{eff} \cdot l_{eff} \cdot f^{EF}}$$

If **NOIMOD=2**:

if $V_{gs} \geq V_{th} + 0.1$ then:

Model Equations

flicker_noise =

$$S_{wi} = \frac{\phi_t q^2 I_d \cdot \mu_{eff}}{f_{eff}^{EF} L_{eff}^2 \cdot COX \cdot 10^8} \left(NOISEA \cdot \log \left(\frac{N_o + 2 \cdot 10^{14}}{N_l + 2 \cdot 10^{14}} \right) + NOISEB \cdot (N_o - N_l) + 0.5 \cdot NOISEC \cdot (N_o^2 - N_l^2) \right)$$

$$+ \frac{\phi_t \cdot I_d^2 \cdot \Delta L_{clm}}{f_{eff}^{EF} L_{eff}^2 W_{eff} \cdot 10^8} \cdot \frac{NOISEA + NOISEB \cdot N_l + NOISEC \cdot N_l^2}{(N_l + 2 \cdot 10^{14})^2}$$

$$N_o = \frac{COX(V_{gs} - V_{th})}{q}$$

$$N_l = \frac{COX(V_{gs} - V_{th} - V'_{ds})}{q}$$

$$V'_{ds} = \min(V_{ds}, V_{dsat})$$

$$\Delta L_{clm} = \begin{bmatrix} litl \cdot \log \left(\frac{V_{ds} - V_{dsat}}{E_{sat}} \right) \\ 0, (V_{ds} \leq V_{dsat}) \end{bmatrix}, V_{ds} > V_{dsat}$$

If $V_{gs} \leq V_{th} + 0.1$ then:

$$\text{flicker_noise} = \frac{S_{lim_it} \cdot S_{wi}}{S_{lim_it} + S_{wi}}$$

$$S_{wi} = \frac{NOISEA \cdot \phi_t \cdot I_d^2}{W_{eff} \cdot l_{eff} \cdot f^{FA} (4 \cdot 10^{36})}$$

Where S_{wi} is the strong inversion flicker noise calculated at $V_{gs} = V_{th} + 0.1$.

If NOIMOD=3:

 SPICE flicker noise model

 BSIM3 thermal noise model

If NOIMOD=4:

 BSIM3 flicker noise model

 SPICE thermal noise model.

ASPEC Charge Model

$$C_o = COXw'l'$$

$$v_{th} = v_{te}$$

$$v_{th}' = v_{th} - CF1$$

$$v_{ds}' = CF3 \ v_{ds}$$

Gate-to-Bulk Capacitance

For $v_{gs} \leq v_{fb} + v_{bs}$ (accumulation)

$$CGB = C_o$$

For $v_{fb} + v_{bs} < v_{gs} \leq v_{th}$ (depletion)

$$CGB = \frac{C_o}{\sqrt{1 + \frac{4}{\gamma^2}(v_{gs} - v_{fb} - v_{bs})}}$$

For $v_{gs} > v_{th}$ (inversion)

$$CGB = \frac{C_o G^+}{\sqrt{1 + \frac{4}{\gamma^2}(\gamma \sqrt{\phi_{HI} - v_{bs}} - \phi_{HI} - v_{bs})}}$$

Gate-to-Source Capacitance

$$CGS = x \ XCG \ C_o$$

For $v_{ds} < 0.1$ and $v_{gs} \leq v_{th}'$ (accumulation)

$$x = GD$$

For $v_{ds} < 0.1$ and $v_{th}' < v_{gs} < v_{th} + CF2$ (weak inversion)

$$x = \frac{v_{gs} - v_{th}'}{CF2} \left[1 - \frac{(CF2 - v_{ds})^2}{(2CF2 - v_{ds})^2} - D^- \right] + D^-$$

For $v_{ds} < 0.1$ and $v_{gs} \geq v_{th}' + CF2$ (inversion)

$$x = 1 - \frac{(v_{gs} - v_{th}' - v_{ds})^2}{(2(v_{gs} - v_{th}') - v_{ds})^2}$$

For $v_{ds} \geq 0.1$, $v_{gs} \leq v_{th}'$ and $CF1 = 0$ (accumulation)

$$x = G^-$$

For $v_{ds} \geq 0.1$, $v_{gs} \leq v_{th}'$ and $CF1 \neq 0$ (accumulation)

$$x = GD^+$$

For $v_{ds} \geq 0.1$, $v_{th}' < v_{gs} < v_{th}' + CF2$ and $CF1 = 0$ (weak inversion)

$$x = \frac{v_{gs} - v_{th}'}{CF2}$$

For $v_{ds} \geq 0.1$, $v_{th}' < v_{gs} < v_{th}' + CF2$ and $CF1 \neq 0$ (weak inversion)

$$x = \max \left[\frac{v_{gs} - v_{th}'}{CF2}, D^+ \right]$$

For $v_{ds} \geq 0.1$ and $v_{gs} - v_{th} \leq v_{ds}'$ (saturation)

$$x = 1$$

For $v_{ds} \geq 0.1$ and $v_{gs} - v_{th} > v_{ds}'$ (linear)

$$x = 1 - \frac{(v_{gs} - v_{th} - v_{ds})^2}{(2(v_{gs} - v_{th}) - v_{ds})^2}$$

Gate-to-Drain Capacitance

$$C_{GD} = x \cdot X_{CG} \cdot C_o$$

For $v_{ds} < 0.1$ and $v_{gs} \leq v_{th}'$ (accumulation)

$$x = G D^+$$

For $v_{ds} < 0.1$ and $v_{th}' < v_{gs} < v_{th}' + CF2$ (weak inversion)

$$x = D^+ + \frac{v_{gs} - v_{th}'}{CF2} \max\left(1 - \frac{CF2^2}{(2CF2 - v_{ds})^2}, D^+\right)$$

For $v_{ds} < 0.1$ and $v_{gs} \geq v_{th}' + CF2$ (inversion)

$$x = \max\left[D^+, 1 - \frac{(v_{gs} - v_{th}' - v_{ds})^2}{(2(v_{gs} - v_{th}') - v_{ds})^2}\right]$$

For $v_{ds} \geq 0.1$ and $v_{gs} < v_{th}'$ (accumulation)

$$x = G D^+$$

For $v_{ds} \geq 0.1$ and $v_{gs} - v_{th} \leq v_{ds}'$ (saturation)

$$x = D^+$$

For $v_{ds} \geq 0.1$ and $v_{gs} - v_{th} > v_{ds}'$ (linear)

$$x = \max\left[D^+, 1 - \frac{(v_{gs} - v_{th} - v_{ds})^2}{(2(v_{gs} - v_{th}) - v_{ds})^2}\right]$$

Distributed RC Line Model

$$N = \frac{\ln \left[FMAX \times RPERL \times CPERL \times 2 \times \pi \times len^2 \times \left(\frac{K-1}{K} \right)^2 \right]}{\ln K}$$

The line is divided into $2 \times N$ segments. The segments are symmetrical about the center of the line and are numbered starting from the end terminals and increasing toward the middle of the line. The segments increase toward the middle of the line in a geometric progression with K as the proportionality constant.

$$R_i = \left[\frac{RPERL \ln(K-1)}{2 \times K^N - 2} \right] \times K^{i-1}$$

$$C_i = \left[\frac{CPERL \ln(K-1)}{K^{N-1} \times (K+1) - 2} \right] \times K^{i-1}$$

$$IS_i = \left[\frac{ISPERL \ln(K-1)}{K^{N-1} \times (K+1) - 2} \right] \times K^{i-1}$$

$$GD_i = \left[\frac{\frac{1}{RSPERL \ln(K-1)}}{K^{N-1} \times (K+1) - 2} \right] \times K^{i-1}$$

where:

$i = 1 \dots N$

Distributed RC Line Model

$$N = \frac{\ln \left[FMAX \times RPERL \times CPERL \times 2 \times \pi \times len^2 \times \left(\frac{K-1}{K} \right)^2 \right]}{\ln K}$$

The line is divided into $2 \times N$ segments. The segments are symmetrical about the center of the line and are numbered starting from the end terminals and increasing toward the middle of the line. The segments increase toward the middle of the line in a geometric progression with K as the proportionality constant.

$$R_i = \left[\frac{RPERL \ln(K-1)}{2 \times K^N - 2} \right] \times K^{i-1}$$

$$C_i = \left[\frac{CPERL \ln(K-1)}{K^{N-1} \times (K+1) - 2} \right] \times K^{i-1}$$

$$IS_i = \left[\frac{ISPERL \ln(K-1)}{K^{N-1} \times (K+1) - 2} \right] \times K^{i-1}$$

$$GD_i = \left[\frac{\frac{1}{RSPERL \ln(K-1)}}{K^{N-1} \times (K+1) - 2} \right] \times K^{i-1}$$

where:

$i = 1 \dots N$

Appendix C

DIABLO Language Structure

DIABLO is VeriBest's high level description language used for generating mixed discipline analog models. The language allows you to develop simulation models that can run under the VeriBest Analog (VBA) environment. It is an alternative method to generating models of a more complex macro-model structure.

DIABLO is used to generate the component behavior as a series of statements which can be executed to achieve a computational objective. The topology is handled by the normal methods of model creation using the .SUBCKT keyword through a macromodel definition, and a graphical component symbol.

The DIABLO language supports numerous constants and arithmetic functions, for example Boltzmann's Constant, Absolute Temperature, etc. Other well known functions such as *sin*, *cos*, and *exp* are also recognized. Built-in functions include items such as *pow* (enabling you to raise one value of a variable to the power of a value).

DIABLO is a C type language which can be added to a VeriBest Analog Simulation Engine (VBASE) netlist. In VBASE, there are idealized elements (for example, controlled sources) that allow you to specify a polynomial function. The basic idea behind DIABLO is to extend this polynomial capability to C type functions.

The capability of VBASE has been enhanced with the addition of DIABLO. VBASE was originally intended to be a circuit simulator and has now become a general purpose differential equation solver. With such an attribute, VBASE can be used for device modeling, network design, and systems design. Also, it is not limited only to electrical circuits, but allows the simulation of mechanical, thermal, biological, and other systems described by the same set of equations used in circuit theory (time dependent ordinary differential equations).

This chapter documents DIABLO capabilities, methods of creating an idealized element, numerous examples, and enough information about the VBASE algorithms so that you can use the language in an efficient manner and avoid many pitfalls (like convergence problems). Transformation from a circuit description to other disciplines is also documented. This chapter discusses the following topics:

- Calling a DIABLO Function
 - Overview: An Example
 - General Description
 - Advanced Features
- Writing A DIABLO Function
 - Basic Framework
 - Function Body
 - Special Features
- Convergence Problems

Calling a DIABLO Function

In order to use a DIABLO model in a VBASE circuit description, you must be able to refer to it within the context of VBASE's primitives. If a circuit can be described with a VBASE format containing the keyword `poly`, you can replace `poly` with `func` and use a DIABLO model. Below is a simple example followed by a general description.

Overview: An Example

In standard VBASE, a Voltage Controlled Current Source can be represented as:

```
g1 1 2 poly(2) 2 3 4 5 a0 a1 a2 a3 a4
```

VBASE interprets this to be:

$$I(g1) = a0 + a1*v23 + a2*v45 + a3*v23*v23 + a4*v23*v45$$

where

$I(g1)$ is the current through $g1$

vnm is the voltage at node n minus the voltage at node m

ai represents either numbers or VBASE parameters

A DIABLO function can be referred to in a VBASE netlist by any device which uses the keyword `poly`. Instead of a polynomial representation of $I(g1)$ as in standard VBASE, you can write a general function by replacing the keyword `poly` with the keyword `func` and adding the function name after the input node list (described in the "Writing a DIABLO Function" section). For example:

```
g1 1 2 func(2) 2 3 4 5 DIABLO_FUNCTION a0 a1 a2 a3 a4
```

where $I(g1)$ is now a value returned by a user defined function (called `DIABLO_FUNCTION` in this case).

$$I(g1) = DIABLO_FUNCTION(v23, v45, a0, a1, a2, a3, a4)$$

It is important to note that the number (2) following `func` refers to the number of input nodal voltages (which change continuously during simulation), although the DIABLO function may be a function of several variables.

Note: The last 5 variables are constants which change only during parametric sweeps or Monte Carlo analysis.

General Description

Controlled Source Functions

A DIABLO function can be called by the following VBASE elements in a manner similar to the example in "Overview: An Example":

- Exxxx - Voltage Controlled Voltage Sources
- Gxxxx - Voltage Controlled Current Sources
- Hxxxx - Current Controlled Voltage Sources
- Fxxxx - Current Controlled Current Sources

Capacitors and Inductors

There are two ways to write one-dimensional capacitors and inductors.

- Using Controlled Source Functions, for example:

```
Vil1 3 1 0
Hl1 1 2 D_DT(1) Vil1 diablo_inductor
Gc1 1 2 D_DT(1) 1 2 diablo_capacitor
```

where:

`diablo_inductor` is the flux through the inductor calculated in the `diablo_inductor` function using the inductor current measured by the voltage source `Vil1`. The inductor is connected in between nodes 2 and 3.

`diablo_capacitor` is the charge on the capacitor

- Using capacitor and inductor functions. Capacitors and inductors can call a DIABLO function (the format is slightly different since capacitors and inductors are one dimensional functions of their controlled currents (inductors) or voltages (capacitors). For example:

```
L1 1 2 func DIABLO_INDUCTOR a1 a2
```

translates to:

```
v12 = d(DIABLO_INDUCTOR(I(L1),a1,a2))/dt,
```

and

```
C1 1 2 func DIABLO_CAPACITOR a1 a2
```

translates to:

```
I(C1) = d(DIABLO_CAPACITOR(v12,a1,a2))/dt.
```

Here $d(\dots)/dt$ is the time derivative.

As before, `DIABLO_CAPACITOR` is the charge on the capacitor and `DIABLO_INDUCTOR` is the flux through the inductor.

The respective currents and voltages are given by:

$$i_c = \frac{d(q(V_c))}{dt}$$

where:

i_c is the capacitor current

$q(V_c)$ is the capacitor charge depending on capacitor voltage

and

$$V_L = \frac{d(\phi(i_L))}{dt}$$

where:

v_L is the inductor voltage

$\phi(i_L)$ is the inductor flux depending on inductor current

However, if you do not have the charge-voltage function, but do have the capacitance-voltage function, you must express charge in terms of capacitance and voltage to use the method described above:

$$q(u) = \int C(V)dv$$

Then use it as the DIABLO function `DIABLO_CAPACITOR` mentioned above.

Sometimes it is more convenient to use the $L = f(i_L)$ function than $\varphi = f(i_L)$ to define a nonlinear inductor. In this case, to obtain $\varphi(i)$, an integral of the L function must be calculated:

$$\varphi(i) = \int L(i)di$$

Use it as the DIABLO function `DIABLO_INDUCTOR` mentioned above.

The following is an example of the linear inductors and capacitors using all previous methods:

```
L2 1 3 func ind_psi 1m 0

V3 4 5 0
H3 5 0 d_dt(1) V3 Diablo_ind 1m 0

.func
#ifdef ind_psi
#define ind_psi
ind_psi (i, L, IC)
{
    // i is the inductor current
    // L is the inductance
    // IC is the initial current
return (IC + i) * L
}
#endif

#ifdef Diablo_ind
#define Diablo_ind
{
return (IC + i) * L
}
#endif
.endfunc

C2 1 3 func cap_chrg 1n 0
G3 4 0 d_dt(1) 4 0 Diablo_cap 1n 0

.func
#ifdef cap_chrg
#define cap_chrg
cap_chrg (V, C, IC)
{
```



```
        // V is the capacitor voltage
        // C is the capacitance
        // IC is the initial current
return (IC + V) * C
}
#endif

#ifndef Diabla_cap
#define Diabla_cap
{
return (IC + V) * C
}
#endif
.endfunc
```

Multidimensional Capacitors and Inductors

In order to provide for multidimensional capacitors and inductors, the keyword `d_dt` can replace `func` on Controlled Sources. For example:

```
G1 1 2 d_dt(2) 3 4 5 6 MULTI_CAPACITOR a1 a2
```

translates to

```
I(G1) = d(MULTI_CAPACITOR(v34,v56,a1,a2))/dt
```

and

```
H1 1 2 d_dt(2) vx vy MULTI_INDUCTOR a1 a2
```

translates to

```
v12 = d(MULTI_INDUCTOR(I(vx),I(vy),a1,a2))/dt
```

where:

`I(vx)` and `I(vy)` are currents through independent voltage sources

`MULTI_CAPACITOR` describes a multidimensional charge

`MULTI_INDUCTOR` describes a multidimensional flux

Voltage Controlled Voltage Sources, Voltage Controlled Current Sources, and Current Controlled Current Sources may also use this keyword.

Note: The time derivative of the DIABLO function is used by VBASE, not the value of the function itself. A common mistake is modeling a capacitance or inductance rather than a charge or flux. Such modeling will result in erroneous data from VBASE.

Advanced Features

The following keywords can appear on the Controlled Sources (E, F, G, and H). These keywords are described in more detail in the “Special Features” section.

- **breakpoint** - can appear as one of the coefficients (the a's in the examples above) which allows you to set breakpoints in a transient analysis.
- **maxstep=value** - can appear at the end of Controlled Sources to override TMAX on the `.tran` line.
- **lin_func** or **step_func** - can replace `func` to tell VBASE that it is dealing with either a linear or a step function.

Writing a DIABLO Function

This section discusses the following topics:

- Basic Framework
- Function Body
 - Syntax Rules
 - Variables and Numbers
 - Comments
 - Algebraic Expressions and Function Calls
 - Conditional Statements
 - Return Statements
 - Print Statements
- Special Features
 - System Reserved Variables (internally set by VBASE)
 - time
 - last_time
 - timestep
 - freq
 - temp
 - old_value
 - System Reserved Variables (set by the user)
 - max_step
 - maxstep
 - breakpoint
 - Reserved Prefixes
 - old_
 - d_d
 - System Flags
 - phase
 - time_flag
 - status
 - deriv
 - Special Function Types
 - lin_func
 - step_func
 - Multi-Defined Function Names
 - Summary

Basic Framework

A DIABLO function can be written and/or edited using the Model Library Manager editor in VBA or it can be defined directly in the netlist.

Note: Since you are dealing with the DIABLO language in this section, you will deal directly with the netlist description of DIABLO.

One or more DIABLO functions can be defined between the keywords `.FUNC` and `.ENDFUNC` in the netlist.

```
.FUNC
DIABLO function 1
DIABLO function 2
.
.
.
.ENDFUNC
```

The DIABLO function can have the form:

```
name(arg1, arg2, ... argn)
{
    function body
}
```

Note that the function name has to be unique. VBASE uses the most recent definition, overwriting any previous ones bearing the same name. This name conflict may appear when user-defined library models containing DIABLO functions are being used in the circuit. In order to prevent this conflict, it is wise to use long function names especially with prefixes or suffixes associated to the model name.

Another technique is to use conditional statements as follows:

```
#ifndef name
#define name
name (arg1, arg2, ... argn)
{
    function body
}
#endif
```

This prevents overwriting the previous definition, (which presumably was working well) and makes it easier to find that the error is in the recently introduced DIABLO function.

Example

An example of a call to the DIABLO function using the basic framework is shown below:

```
g1 1 2 func(2) 2 3 4 5 function1 10
l1 1 2 func function2 1u

.func
#ifndef function1
#define function1
function1(a1,a2,a3)
{
    body of function1
}
#endif
#ifndef function2
#define function2
function2(a1,b2)
{
```

```
        body of function2
    }
    ##endif
    .endfunc
where:
```

<code>function1</code>	are the names referred to by the VBASE elements in “Calling a DIABLO
<code>function2</code>	Function.”
<code>a1 ...</code>	are the argument names passed by the element (an error occurs if the calling element has a different number of arguments than the DIABLO function and the simulation will terminate).

A DIABLO function does not distinguish between controlling voltages or currents, and VBASE parameters. Therefore the order of the argument list must match the calling sequence of the elements (controlling voltages or currents first).

The inductor element above has one parameter although `function2` has two. This is because the first argument refers to the implicit current through the inductor. The same is true for capacitors where the first argument is the voltage across a capacitor.

The function name is case independent but the arguments (as well as all variables in DIABLO except where specified) are case dependent. See the following section, “Function Body.”

Function Body

The body of a DIABLO function consists of comments, algebraic expressions with function calls, conditional statements, print statements, and a return statement. Variables are user-defined except for certain keywords which VBASE uses to pass information to the DIABLO function. The format of the body is loosely C type in appearance.

Syntax Rules

DIABLO’s syntax rules are simple. They are as follows:

- Every statement or expression is terminated with a semicolon (;).
- White spaces (blanks, tabs) are used as separators and are otherwise ignored. Use white spaces to make your DIABLO statements readable.
- The function body is enclosed within a left brace ({) and a right brace (}).
- The function name is followed by a set of arguments, separated by commas, enclosed in parentheses.
- Function variables are case sensitive and can be of any convenient length, but are restricted to system/UNIX limits.
- Keep line lengths to a convenient length, approximately 60 characters. Remember, a semi-colon is required to terminate a statement, and there is no requirement for a continuation mark.

If there is any syntax error, VBASE stops and the error is printed to the VBASE output file.

Variables and Numbers

Variables in DIABLO can be at most 100 characters long and are case dependent. They must start with a letter but can contain numbers or underscores (_), for example, `my_3rd_variable_is_long`.

There are a number of variables which are built into the DIABLO language and should not be overwritten (see “Algebraic Expressions and Function Calls”). These pre-determined variables include keywords, built-in constants, built-in functions, and scale factors. They are shown on the next few pages.

Keywords

- if
- else
- return
- print

Built-in Constants

Constant	Name	Value	Units
abszero or ABSZERO	absolute zero	-273.16	Centigrade
boltz or BOLTZ	Boltzmann constant	1.3806226E-23	Joules/Kelvin
charge or CHARGE	electronic charge	1.6021918E-19	Coulomb
ctok or CTOK	C to Kelvin	273.16	Kelvin
deg or DEG	degree	1.7453293E-2	Degree
epso or EPSO	epsilon (free space)	8.85421487E-12	Farads/meter
epsox or EPSOX	epsilon (silicon dioxide)	3.453143E-11	Farads/meter
epssil or EPSSIL	epsilon (silicon)	1.0359431E-10	Farads/meter
pi or PI	pi (π)	3.1415927	Radian
rad or RAD	radians	5.729578E1	
root2 or ROOT2	$\sqrt{2}$	1.41421356	
twopi or TWOPI	2pi (2π)	6.2831853	
xlog2 or XLOG2	ln2	6.9314718E-1	
xlog10 or XLOG10	ln10	2.3025851	

Built-in Functions

Function	Name	# of Arguments	
abs	absolute value	1	as in k=abs(-12)
acos	arccosine	1	as in beta=acos(0.777)
acosh	hyperbolic arccosine	1	as in beta=acosh(0.777)
asin	arcsine	1	as in gamma=asin(0.25)
asinh	hyperbolic arcsine	1	as in gamma=asinh(0.25)
atan	arctangent	1	as in theta=atan(1.0)
atanh	hyperbolic arctangent	1	as in theta=atanh(1.0)

Function	Name	# of Arguments	
cos	cosine	1	as in cos(pi)
cosh	hyperbolic cosine	1	as in cosh(pi)
exp	exponential	1	as in exp(x)
int	integer part	1	as in int(3.899)
ln	log _e	1	as in h=ln(2)
log10	log ₁₀	1	as in r=log10(3)
max	maximum	2	as in max(x,y)
min	minimum	2	as in min(x,y)
pow	exponentiation	2	as in pow(x,3)
sign	sign	1	as in sign(x)
sin	sine	1	as in sin(2*pi*f*t)
sinh	hyperbolic sine	1	as in sinh(2*pi*f*t)
sqrt	square root	1	as in y=sqrt(2)
tan	tangent	1	as in tan(pi/4)
tanh	hyperbolic tangent	1	as in tanh(pi/4)

Numbers

A number may be expressed in a DIABLO statement as:

- An integer (12, -44)
- A floating point number (3.14159)
- An integer or floating point number followed by an integer exponent (1E-14, 2.65E3)
- An integer or floating point number followed by one of the following scale factors:

Scale	Multiplier
T or t	10 ¹²
G or g	10 ⁹
MEG or meg	10 ⁶
K or k	10 ³
MIL or mil	25.4x10 ⁻⁶
M or m	10 ⁻³
U or u	10 ⁻⁶
N or n	10 ⁻⁹
P or p	10 ⁻¹²
F or f	10 ⁻¹⁵

Other Variables

The variables defined below can be used to define a DIABLO function. These variables can change during run time. See “Special Features” for a detailed explanation of each variable.

Transient Analysis Related Variables

- **time** - present time
- **last_time** - last converged time
- **timestep** - **time** minus **last_time**
- **time_flag** - transient analysis status
- **status** - first call in a transient analysis
- **max_step** - maximum step size
- **old_value** - last converged output value

Small-Signal AC Analysis Related Variable

- **freq** - present frequency

Temperature Related Variables

- **temp** - present temperature in degrees centigrade
- **vtherm** - thermal voltage (boltz * temp/charge) kT/q (T is in degrees Kelvin)

Other Variable

- **deri** - derivative flag
- **phase** - type of analysis

Reserved Prefixes

- **d_d** - derivatives
- **old_** - values at last converged time

Comments

Comments can appear anywhere between `.func` and `.endfunc`. All comments are preceded by `//` and terminate at the end of the line (see the examples in the next section).

Algebraic Expressions and Function Calls

You can assign local variables (that is, these variable names are recognized only within the DIABLO function body) using standard C algebraic expressions. To help clarify DIABLO's features, review the following example line by line.

```
g1 1 2 func(1) 3 4 function1 5
.func
function1(a1,a2)
{
    A_var = 3 * sqrt(a2*a2 + 1.1e1) + a2; //line 1//
    a2 = A_var;                          //line 2 //
```

The following is wrong

```

        b2 = a3;           // line 3
        pi = 9;           // line 4
        aa1=2             // line 5
    }
    .endfunc

```

where:

Line 1: A_var takes on a value 23 (notice how the function call sqrt is used).

Line 2: a2 (which was 5) will now be 23. It is important to note that since a2 is an argument passed to function1, that argument is updated. In other words, the next time function1 is called by VBASE, a2 is 23 (see “Convergence Problems”).

Line 3: This is an error since a3 has not been defined. The simulation will terminate during run time if the rest of the function is free of syntax errors (see line 5). If no syntax error occurs, an error message will be given which specifies the unknown variable and in which function it exists. The simulation will then terminate.

Line 4: This resets pi from 3.14159 to 9 which is not a good idea because, if pi is called by another DIABLO function, it will be 9.

Line 5: a ; is missing, so the simulation will stop with a message that says syntax error in or near above line.

Arithmetic Operators

The following arithmetic operators are available for use in expressions and statements:

- +
- -
- /
- *
- ^ (exponentiation)
- = (assignment)
- % (modulus)

Conditional Statements

Branching of DIABLO functions can be accomplished using the following C statements:

- if ()
- else
- else if ()
- {
- }
- return

Logical statements controlling branching are also C type and are as follows:

Statement	Example
if	if (x <= 1u x > 1p)
if ... else	if (v < vlim)x=1m; else x=1mil;

Statement	Example
if ...else if ... else	if (c != 1.0)y=c^2; else if (t < 1u) y=1/(c^2 +1); else y=pow(c,vlim);
if	if (y == z)
{	{
compound statements	q=charge^2;v=sqrt(1n * q); if
	(v < 1E-6)v=v * abs(v);
}	}
else	else
{	{
compound statements	q=...etc
}	}

The general form of DIABLO statements controlling branching are

```
if ( logical conditions ) {
    statements
} else {
    statements
}
```

OR

```
if ( logical conditions ) {
    statements
}
```

OR

```
if ( logical conditions ) {
    statements
} else if ( logical conditions ) {
    statements
}
```

Logical Operators

The following logical operators are available for use in if statements.

- < (less than)
- <= (less than or equal to)
- > (greater than)
- >= (greater than or equal to)
- == (equal)
- != (not equal)
- || (or)
- && (and)

Return Statements

Each DIABLO function should have at least one return which is the value of the function. If no returns are present, a value of 0 is given to the function (see “Convergence Problems”).

You can print constant values, values of expressions, strings, and control characters. Strings and control characters are surrounded by “`“`”, and every item printed must be separated by commas.

The formats for control characters are:

- Note:** "`\n`" should be at the end of each print statement or the lines will run into each other.

```
value1 = 3;
print ("value1 = ", value1, "\n");
```

```
value1 = 3
```

Note: If you forget "`\n`", the output looks like:

Special Features

A typical VBASE transient analysis can be described as in the figure below:



1. VBASE takes a step to t_2 and tries to converge by doing a number of solution trials. It fails to meet some internal convergence criteria and backtracks to t_3 .
2. VBASE is now at t_3 and tries to converge again doing a number of solution trials. It converges, so t_3 now becomes the last converged time and VBASE steps to t_4 .
3. VBASE tries to converge at time t_4 .

VBASE Features

- System Reserved Variables (internally set by VBASE)
 - time
 - last_time
 - timestep
 - freq
 - temp
 - old_value
 - vtherm
- System Reserved Variables (set by the user)
 - max_step
 - maxstep
 - breakpoint
- Reserved Prefixes
 - old_
 - d_d
- System Flags
 - phase
 - time_flag
 - status
 - deriv
- Special Function Types
 - lin_func
 - step_func
- Multi-Defined Function Names

System Reserved Variables (internally set by VBASE)

time

The `time` variable in DIABLO is always the present time with which VBASE is working. In the above example, in STEP1, `time=t2`; in STEP2, `time=t3`; and in STEP3, `time=t4`.

It is important to note that you have no information as to whether or not `time` will be a converged time point.

Note: `time` and all time related variables described below will be zero if you are not in the transient analysis mode.

last_time

The `last_time` variable is the last converged time point. In STEP1, while `time = t2`, `last_time` is `t1`. In STEP2, `last_time` is still `t1`, but in STEP3, `last_time` is `t3`.

timestep

`timestep = time - last_time`

freq

The keyword `freq` allows you to create your own frequency dependent element simply by writing a DIABLO function with that keyword. It must be noted that if `freq` is used in a DIABLO function, the gain (voltage or current derivatives) of the elements referring to the function becomes frequency dependent, and the function will be called for each frequency value `freq` (with `phase = 1`). If no `freq` is specified, the DC gain is used (`phase = 0`). This feature is used only in small-signal AC analysis. During a transient and DC analysis, `freq` is 0.

For elements which store charge (capacitors, inductors and controlled sources with `d_dt`) an imaginary gain value is stored and used during small-signal AC analysis, otherwise a real gain value will be stored.

```
// DIABLO - VERSION 1.0
skin (vsk) {
if(freq<0.5E6 || phase!=1) res=13.5;
else res=13.5+20E-6*(freq-0.5E6);
curr=vsk/res;
return curr;
}
```

This DIABLO function models a frequency dependent resistor. It may be used to model skin-effect in an inductor. It works only for Frequency analysis. Note the use of the `phase` variable.

Note: VBASE uses the voltage (or current) derivative of a DIABLO function as the gain. A common mistake is to assume that VBASE uses the value returned by DIABLO as the gain. Such an assumption gives erroneous results.

Note: It is also important to realize that the voltages or currents used in the DIABLO function to calculate the gain for a small-signal AC analysis are the (real) DC operating point voltages or currents (which do not change during a frequency sweep) not the (complex) voltages or currents which do change.

temp

Temperature is in degrees Centigrade.

old_value

The `old_value` variable is the returned value at the last converged timestep. In the example in Figure 3-1, in STEP1 and STEP2, `old_value` is the value at `t1`. In STEP3, `old_value` is the value at `t3`.

vtherm

The `vtherm` variable is the thermal voltage in degrees Kelvin, kT/q (boltz * temp/charge).

System Reserved Variables (set by the user)**max_step and maxstep**

`max_step` and `maxstep` are two variables which, though related, are entirely different in how they are used. `maxstep` is not part of the DIABLO function, but may be part of a Controlled Source call to a DIABLO function. `maxstep` sets the maximum step size globally to be `min(maxstep, tmax)` where `tmax` is on the `.TRAN` line. `max_step`, on the other hand is a keyword in a DIABLO function

and may lower (never raise) the global maximum step size (tmax). As an example, consider the following VBASE netlist:

```
.tran 10n 1000n 0 15n
g1 1 0 func(1) 1 0 fix_max 100ns 150ns 500ns maxstep=10ns
.func
fix_max(v1,t1,t2,t3)
{
if(time > t1 && time < t2)
    max_step=1ns;                // condition 1
else if(time >= t2 && time < t3)
    max_step=5ns;                // condition 2
else
    max_step = 1;                // condition 3
maxstep = 3*v1;
return(maxstep);
}
.endfunc
```

At the start of a run, the maximum step is set globally to 10ns. When the time is greater than 100ns, the maximum step is set to 1ns until time 150ns is reached or passed, at which point, it is set to 5ns. After 500ns, the maximum step is reset to 10ns

Note: The maximum step is the minimum of the global and `max_step`.

Note: A new variable, `maxstep`, was used which is no relation to `maxstep` on `g1`. In other words, `maxstep` is a keyword only on the element card and not part of a DIABLO function.

Note: `maxstep` must be the last parameter on the Controlled Source line (`g1` in the above example).

breakpoint

The `breakpoint` feature forces VBASE to perform a transient calculation at a specific time. In the example in Figure 3-1, if there was a breakpoint between `t3` and `t4`, the time step in VBASE would stop there rather than at `t4`.

DIABLO users can set breakpoints by specifying `breakpoint` as one of the parameters in a call to DIABLO for any of the Controlled Sources. The parameter in the DIABLO function which matches the call, will set a specified breakpoint greater than the present time (if the value is less than or equal to time, it will be ignored). The following example sets a breakpoint at the present time + 3 nanoseconds.

```
g1 1 2 func(1) 1 2 set_break 1 2 breakpoint 3 4
.func
set_break(v1,p1,p2,bk,a1,a2)
{
    if(v1 > p2)
        bk=time+3ns;
}
.endfunc
```

Reserved Prefixes

old_...

`old_` can be used as a prefix to any controlling voltage or current. In the explanation given in the System Reserved Variables section above, `old_v1` is the value of `v1` at the last controlled time step. `old_t1`, `old_t2` ... are not used because they do not refer to the controlling voltages and currents (they will always be set to zero).

d_d...

In order to find a solution in VBASE, the derivative with respect to the controlling voltages or currents is needed. VBASE finds these automatically using the classic numerical derivative technique:

$$\frac{(f(x+h, y, \dots) - f(x, y, \dots))}{h}$$
$$\frac{(f(x, y+h', \dots) - f(x, y, \dots))}{h'}$$
$$\vdots$$

for `h`, `h'` ... > 0

For example, if you have the following call to a DIABLO function:

```
G1 1 0 func(3) 1 2 3 4 5 6 three_d 5 6 7
.func
three_d(v1,v2,v3,p1,p2,p3)
.
.
.
.endfunc
```

and the `three_d` function does not call upon the use of partial derivatives, VBASE's internal algorithm performs the partial derivative when it linearizes a set of nonlinear differential equations.

The analytical equation for `d_dv1` is represented by $\frac{\partial i(g1)}{\partial v1}$.

For each iteration, `three_d` must be called 4 times: once for the calculation of `f` at (`v1`,`v2`,`v2`) and once for `f` at (`v1`+`h`,`v2`,`v3`), (`v1`,`v2`+`h'`,`v3`) and (`v1`,`v2`,`v3`+`h''`). In the first call, `deriv` is set to 0, then `deriv` is set to 1 for the other three calls.

In order to avoid all of these calls, you can write your own analytical derivatives with the preface `d_d` to the controlling voltages or currents. In the above example, there are 3 terms: `d_dv1`, `d_dv2` and `d_dv3` (a term `d_dp1` is ignored). If these values are specified, the numerical derivative algorithm is turned off and `three_d` is called only once for each iteration (`deriv` is always 0). See the "Convergence Problems" section.

d_dt

This function gives you access to the time derivative of a DIABLO function. It is called by replacing the keyword `FUNC` on the dependent source with the keyword `D_DT`. For example, if one specifies a Current Controlled Current Source:

```
G1 1 2 D_DT(2) 3 0 4 0 mycap
```

the current through `G1` is the time derivative of the value returned by `mycap`. So, if `mycap` returns a charge, `G1` is a capacitor controlled by two voltages.

System Flags

phase

This specifies the type of analysis being performed:

- 0 for DC
- 1 for small-signal AC
- 2 for transient

A transient analysis in VBASE first proceeds in finding a DC operating point at time = 0, in which the phase = 0. After time = 0, the phase equals 2.

(Figure 3-1 describes states which are in the middle of a transient analysis, therefore phase = 2 in all steps of Figure 3-1.)

time_flag

During a transient analysis:

- `time_flag = 1` for the first iteration of a new time point if VBASE moved forward in time and converged on the last time point.
- `time_flag = -1` for the first iteration of a new time point if VBASE moved backward in time and did not converge on the last time point.
- `time_flag = 0` for the rest of the iterations

For example, in STEP1, `time_flag = 1` on the first solution trial, then 0. In STEP2, `time_flag = -1` then 0; and in STEP3, `time_flag = 1` then 0.

status

The `status` System flag is used as a flag to initialize variables, for example, Initial Conditions, where:

```
status = 1 in the first Newton-Raphson iteration of the first time step
status = 0 during the rest of the runs
```

In the above example, `status` will always be 0.

deriv

The `deriv` system flag indicates that you are in the numerical derivative mode. `deriv = 0` for the first call to the DIABLO function. `deriv = 1` for subsequent calls to DIABLO functions. See “d_d” for more information.

Special Function Types

lin_func and step_func

`lin_func` and `step_func` tell VBASE what type of function a Controlled Source is calling. Such information speeds up simulation, as well as helps with convergence (see “Convergence Problems”).

`lin_func` refers to a linear or piece-wise linear function, and `step_func` refers to a step function. Both keywords may replace `func` on the Controlled Sources. For example:

```
g1 1 2 lin_func(1) 1 2 linear S3
g2 2 3 step_func(1) 2 3 step 4
.func
linear(v1,p1)
{
```

```

if(v1>p1)
    out=p1*p1;
else
    out=v1*p1;
return(out);
}

```

```

step(v1,p1)
{
if(v1>p1)
    out=1;
else
    out=0;
return(out);
}
.endfunc

```

`step_func` should be used if a DIABLO model has a step defined within the model that has no specified rise time. The `step_func` keyword forces VBASE not to calculate numerical derivatives for the specified function. This dramatically speeds simulation.

Note: If `step_func` is used on the wrong type of function (for example, any function which is not like a step function), the simulation may fail or produce incorrect results.

Note: `lin_func` can cause non-convergence if used on a non-linear DIABLO function.

Multi-Defined Function Names

Names assigned to DIABLO functions should be unique in order to prevent compiler errors. An exception occurs when a subcircuit containing one or more DIABLO function definitions is instantiated more than once. To instruct the compiler to ignore subsequent definitions of a particular DIABLO function, that function must be enclosed within the following directives.

```

#ifndef function_name
#define function_name
// DIABLO - VERSION 1.0

function_name(arguments)
{
function body
}
//
#endif

```


Summary

A summary of the values of various parameters for the example in Figure 3-1 is displayed in the following table.

step	time	last_time	timestep	phase	time_flag	status	deriv
STEP1	t2	t1	t2 - t1	2	1/0	0	0/1
STEP2	t3	t1	t3 - t1	2	-1/0	0	0/1
STEP3	t4	t3	t4 - t3	2	1/0	0	0/1

For `time_flag`, `x/x` represents:

the first try for convergence / the rest of the tries for convergence at present time = time

and for `deriv`, `x/x` represents:

the first call to the DIABLO function / subsequent calls to DIABLO functions to produce numerical derivatives

Convergence Problems

One of the biggest problems with most circuit simulators is convergence. Since you are dealing with a rather large set of non-linear differential equations which are solved numerically, a solution is not always attainable. However, if you have knowledge of the equations, you can develop algorithms to ensure a solution in most cases. This is one of the reasons why programs like VBASE work.

In a user-defined set of equations, on the other hand, you have no *prior* information of the equations, so the algorithms which are built into VBASE may not work. Ideally, a general set of converging algorithms is desirable so that an engineer designing a circuit with DIABLO can concentrate on the physical equations rather than the numerical algorithms.

Presently, the solution to DIABLO convergence problems are quite simple:

- You can specify certain keywords which let VBASE know what *type* function you are defining.
- Both numerical and analytical derivatives are allowed.
- Debugging tools in the form of print statements are provided.
- VBASE has built-in algorithms.

More general approaches as well as user-defined techniques will be provided in future releases.

One of the main reasons for non-convergence in non-linear systems is *bad* derivatives. A general algorithm for numerical solutions of such equations is called the Newton-Raphson technique. Here, you take a guess at the initial solution, calculate derivatives, and find a *better* solution. The derivatives tells the simulator in which direction the actual solution lies. If the directions to find the solution are wrong or confusing, a solution will not be found and non-convergence occurs.

The four types of *bad* derivatives are:

- wrong derivatives
- rapidly changing derivatives
- discontinuous derivatives
- discontinuous functions with continuous or discontinuous derivatives

Wrong derivatives can be avoided by allowing VBASE to perform numerical derivatives (see “deriv” in the “Special Features” section). However, this can slow down simulation somewhat due to the number of calls to a DIABLO function per iteration. That is why you are provided with the ability to perform analytical derivatives. However, great care must be taken that the analytical derivatives are correct since wrong derivatives will almost always cause non-convergence. Therefore, if you get non-convergence and you use the `d_d` feature, the first thing to do is check the derivatives.

Rapidly changing derivatives is a sign of an unsmooth function. In many cases, VBASE will work quite well due to its time control mechanism, although the time step will be small and the simulation may take a long time. An efficient way of coding is to use `max_step` in regions where things are changing rapidly (or `maxstep` if things are changing throughout the entire simulation). See “max_step and maxstep” in the “Special Features” section.

Special cases of rapidly changing functions are discontinuous functions and/or discontinuous derivatives. If this is the case, it is advisable to do curve fitting across the discontinuous region. Step functions are an exception to this rule since a mechanism is provided to let VBASE know when this case occurs (see “lin_func and step_func” in the “Special Features” section).

In order to find out where problems occur, the `print` statement has been provided (see “Print Statements”) in which you can monitor various bits of information during simulation. While this is not the ultimate debugging tool, it is a means to help a DIABLO user write efficient, trouble free code.

Appendix D

Error Messages

The types of errors VBASE recognizes during the reading of the source file, in increasing order of severity, are the following:

WARNING

SERIOUS ERROR

FATAL ERROR

A WARNING draws attention to a situation where an unintended effect may be generated.

A SERIOUS ERROR causes a message to be written, but execution continues.

A FATAL ERROR creates a situation so questionable that the simulation is aborted. However, the data file continues to be read to help identify other errors.

Format

<keyword> (input): <error_message>

where:

<keyword> indicates a token that is incorrect

(input) represents the filename/line #

error_message is one of the messages shown on the following pages

On the following pages are error and warning messages with commentary. The descriptions enclosed in square brackets appear where values occur in the messages. The number to the left in parentheses indicates the error level as follows:

(1) WARNING

(2) SERIOUS

(3) FATAL

.INCLUDE, .LIBRARY, and .ENDLIB Errors

- (2) floating point number [strg] is too small
 - (2) floating point number [strg] is too large
 - (3) include file names must be quoted
 - (3) file [strg] is included recursively
 - (3) entry [strg] in library [strg] is included recursively
 - (3) file [strg] seems to have vanished - can't be reopened
 - (3) cannot locate entry [strg] in library [strg]
-

.NAME Instruction Errors

- (1) [val] is too narrow for input width
 - (2) can't set [strg] flag to [strg]
 - (1) simulation temperature [val] is too low
 - (2) can't simulate below absolute zero
 - (2) simulation temperature [val] is too high
 - (1) only one [strg] line is allowed
 - (1) limits are ignored on [strg] lines
 - (2) [strg] is an undefined parameter
 - (1) parameter [strg] redefined
-

.NODESET Error

- (1) node setting for node [strg] redefined
-

Circuit Checker Errors

- (2) no elements connected to node [1strg]
 - (1) no elements connected to node [1strg] - node ignored
 - (2) only one element is connected to node [1strg]
 - (1) no elements connected to node [1strg] before flattening
 - (1) only one element is connected to node [1strg] before flattening
 - (1) strangely specified node [strg] is being changed to [strg]
 - (2) [strg] doesn't have [val] nodes
 - (1) node [1strg] is not referenced by an element -- ignored
 - (1) node [strg] in output request is not specified elsewhere (elsewhere -- ignored)
-

-
- (1) element [strg] in output request is not specified (elsewhere -- ignored)
 - (1) cannot put two nested subcircuit branch l probes on the same node [strg]. [strg] probe ignored.
 - (2) [val] is an invalid value for [strg]
 - (1) [val] is an invalid value for [strg]
 - (2) [strg] is too large for an integer
 - (1) strangely specified integer [strg] is being changed to [val]
 - (1) should use e to start exponent in [strg], not d
 - (2) [fval] is not a legal [strg]
 - (1) [strg] has insignificant digits
-

Command Error

- (3) usage -- [file ...] [-cCEfFhHiIMOqQSvV] [-e efile] [-o ofile] [-m numprocs]
-

DC Solution Errors

- (1) no DC solution - initial conditions specified for every node
 - (1) gmin of [fval] may be affecting DC solution -- consider rerunning with smaller gmin
 - (2) no DC convergence due to a singular jacobian
 - (2) no DC convergence
 - (3) cannot perform .DC param analysis with the parameter [strg]. It is referenced on an analysis control card.
 - (1) DC solution converged in steady state but KCL residual is in the [eval] amps range.
 - (1) DC solution stopped due to iteration limit -- ITL1 = [val]
-

General Errors

- (2) [strg] is a duplicate device
 - (2) no bulk node found for [1strg]
 - (1) substrate node for [1strg] is being defaulted to ground
 - (2) [strg] is a duplicate [strg]
 - (2) new device [strg] is not allowed in alter mode
 - (2) missing number
 - (3) terminated after [val] errors
 - (3) can't open [strg]
 - (1) extra characters at end of line
 - (1) empty [strg] line
-

- (1) node change from [1strg] to [1strg] for [1strg] is ignored
 - (3) error allocating memory for [strg]
 - (3) SX system: [strg] - cannot continue
 - (3) error in [strg] - cannot continue
 - (3) [strg] - cannot continue
 - (3) cannot push arena [strg]
 - (3) cannot create arena [strg]
 - (3) cannot proof arena [strg]
 - (3) can't find configuration file [strg]
 - (3) corrupted configuration file [strg]
 - (3) this product can no longer be used
 - (1) this product will expire in [val] days
-

Initial Conditions Error

- (1) initial condition for node [strg] redefined
-

Input Processor Errors

- (3) unexpected end-of-file
 - (3) can't open [strg]
 - (2) value for [strg] is missing on [strg] line
 - (2) [strg] is an illegal [strg]
 - (2) [strg] is an unknown [strg]
 - (1) [strg] is an unknown [strg]
 - (2) [strg] is an unsupported [strg]
 - (1) [strg] is an unsupported [strg]
 - (1) [strg] is being truncated to [1strg]
 - (2) [strg] is not a valid [strg]
 - (2) [strg] is not a [strg]
 - (2) [strg] must be between [fval] and [fval]
 - (2) [strg] must be less than [fval]
 - (2) [strg] must be greater than [fval]
 - (2) [strg] for [1strg] must be between [fval] and [fval]
 - (1) [strg] for [1strg] is too big -- changed to [fval]
 - (1) [strg] for [1strg] is too small -- changed to [fval]
-

-
- (2) [strg] for [1strg] must be less than [fval]
 - (2) [strg] for [1strg] must be greater than [fval]
 - (2) value for [strg] must be greater than value for [strg]
 - (3) .END line is missing
-

Input Source Errors

- (3) unrecognizable table file [strg] for MOSFET model [1strg]
 - (2) first and last values must be equal for zero delay repeats
 - (2) time series out of sequence
 - (1) no transient analysis - all nodes in circuit are connected to voltage sources
-

Matrix Error

- (3) singular jacobian in [strg]
-

Model Specification Errors

- (2) can't find model [1strg] for [1strg]
 - (1) model parameter [1strg] is unused in model [1strg]
 - (2) [fval] is an illegal value for the [1strg] parameter in the [1strg] model
 - (2) missing [strg] parameter in [1strg] model
 - (1) duplicate model parameter [1strg] - value [strg] used
 - (2) [1strg] is not a [strg] model
 - (2) table models are not supported for [strg] models
 - (3) cannot parameterize model [strg] with the param [strg]. It is defined on a subcircuit-call (x) instruction.
-

MOSFET Table Model Errors

- (1) width value [fval] is outside the range of [fval] and [fval] for MOSFET [1strg]
 - (1) length value [fval] is outside the range of [fval] and [fval] for MOSFET [1strg]
 - (1) subthreshold calculation disabled for MOSFET model [1strg]
 - (3) cannot create table file [strg] for MOSFET model [1strg]
 - (3) error while writing [strg] to table file [strg] for MOSFET model [1strg]
 - (3) cannot open table file [strg] for MOSFET model [1strg]
 - (3) error while reading [strg] from table file [strg] for MOSFET model [1strg]
-

Mutual Inductor Error

- (2) cannot find inductor [1strg] for mutual inductor [1strg]
-

Sensitivity Analysis Errors

- (1) no DC solution -- all nodes are source or .IC nodes
 - (1) sensitivity of [strg] cannot be computed
-

Subcircuit Errors

- (2) can't find subcircuit [1strg] for [1strg]
 - (2) subcircuit [1strg] and reference [1strg] nodes don't match
 - (2) not in a subcircuit
 - (2) not in subcircuit [strg]
 - (2) [strg] lines are not allowed in subcircuits
 - (2) [1strg] is a duplicate node in the definition for subcircuit [1strg]
-

Tolerance Setting Error

- (1) distortion analysis -- [strg]
 - (1) [strg]=0.0 requires exact arithmetic and maybe infinite loop, default values are instead used
-

Transfer Function Error

- (1) tf analysis -- [strg]
-

Transient Analysis Errors

- (2) no convergence in transient analysis at time [fval]
 - (2) no convergence in transient analysis on first time point -- check initial conditions
 - (1) [val] time points simulated exceeds option LIMPTS value
 - (1) [val] transient iterations exceeds option ITL5 value
-

Transmission Line Errors

- (2) z0 not specified for transmission line [1strg]
 - (2) transmission lines may not be run in partitioned simulation mode
-

User-Defined Element Errors

- (2) user-defined element type [char] ([strg]) is illegal
 - (2) user-defined element type [char] ([strg]) does not exist
-

Index

Symbols

.AC 7-2
.ACQUIRE 7-3
.ALTER 7-4
.DC 7-6
.DELETE 7-8
.DISTO 7-9
.DISTR 7-10
 Equations and Parameters 7-10
.DUMP 7-13
.END 1-5, 7-15
.ENDDEL 7-15
.ENDL 7-18
.FOUR 7-15
.GLOBAL 7-16
.IC 7-16
.INCLUDE 7-17
.LIB 7-18
 Calls 7-18
 Library File 7-18
.MEASURE 7-19
.MODEL 7-20
.MODEL Parameters 7-29
.MODEL Specification 3-1, 4-2, 5-2
.MONTE 7-21
.NODESET 7-22, 7-26, 7-27
.NOISE 7-23
.OP 7-24
.OPTIONS 7-24
.PARAM 7-31
.PLOT 7-32
.PRINT 7-33
.RESTART 7-13, 7-35
.SENS 7-35
.SEQUEL 7-13, 7-36
.SUBCKT 7-37
.TEMP 7-38
.TF 7-39
.TRAN 7-29, 7-39
.VARY 7-42
.WIDTH 7-43

A

ABSDelta 7-25
ABSTOL 7-27
AC Analysis 7-2
ACCT 7-30
Adder A-5
Advanced Features
 breakpoint C-5
 lin_func C-5
 maxstep C-5

 step_func C-5
Algebraic Expressions and Function Calls C-11
 Arithmetic Operators C-12
Algorithm Simulation Options 7-25
ABSDelta 7-25
CMIN 7-25
DCMODE 7-25
GMIN 7-25
ITL1 7-26
ITL5 7-26
LIMPTS 7-26
METHOD 7-26
NOBYPASS 7-26
NODESET 7-26
NOLIMJFT 7-26
NOLIMMES 7-26
NOLIMMOS 7-26
NOSUBS 7-26
QOPT 7-26
RELDelta 7-26
SPICE 7-27
TNOM 7-27
TRMODE 7-27
TUSEIC 7-27
TUSENSET 7-27
Altered Components 7-4
Analysis 1-3
Arbitrary Sources 2-1
Arbitrary 2-1
Arithmetic Operators C-12
ASPEC 7-29
 Charge Model 7-26
 Compatibility 7-29

B

Basic Framework C-6
Bimodal (Bimod) 7-10, 7-13
BJT Devices 4-3
 BJT Models 4-4
 Model Equations B-2
 Model Parameters 4-5
Body Functions
 Return Statements C-13
Branch-Voltage-Limiting 7-26
breakpoint C-5, C-17
BSIM Charge Model 7-26
BSIM2 Devices
 Model Equations B-98
Built-in Constants C-9
Built-in Functions C-9
Bypass 7-26

C

Capacitor
 Model Parameters 3-6

Index

- Capacitor Value 7-25
- Capacitors 3-4, C-3
 - Polynomial 3-7
 - Semiconductor Models 3-6
 - User-Defined 3-7
- CCCS 2-5
- CCVS 2-9
- Channel Length 7-29
- Channel Width 7-29
- Charge Model Selection 7-26
- Charge Tolerance 7-27
- CHGTOL 7-27
- Circuit Description 1-4
- Circuit Examples A-1
- Circuit Partition 7-36
- CMIN 7-25
- CO 7-30
- Comment Lines 1-4
- Comments C-11
- Conditional Statements C-12
- Controlled Source Functions C-2
- Convergence Problems C-21
- Current Controlled Current Source 2-5
- Current Controlled Voltage Source 2-9
- Current Tolerance 7-27

D

- d_d C-18
- d_dt C-18
- DC Analysis 7-6, 7-25
- DC Operating Point 7-24
- DCABSTOL 7-27
- DCMODE 7-25
- DCRELTOL 7-27
- DCVNTOL 7-27
- DEFAD 7-29
- DEFAS 7-29
- DEFL 7-29
- DEFNRD 7-29
- DEFNRS 7-29
- DEFPD 7-29
- DEFPS 7-29
- DEFW 7-29
- Dependent Sources 2-2
 - CCCS 2-5
 - CCVS 2-9
 - VCCS 2-7
 - VCVS 2-3
- deriv C-19
- Device 1-8
- DIABLO 1-5, 7-28
 - VBASE Features C-15
- DIABLO Function
 - Calling C-2
 - Writing C-6

- DIABLO function, Example C-7
- DIABLO Language Structure
 - Convergence Problems C-21
- DIABLO Options 7-28
 - MAXFUNC 7-28
- Differential Pair A-2
- Digital Devices 5-1
 - .MODEL Specification 5-2
 - Flip-Flops 5-3
 - Gates 5-5
 - Model Parameters 5-7
- Diode Devices 4-7
 - Diode Models 4-8
 - Model Equations B-8
 - Model Parameters 4-9
- Distortion Analysis 7-9
- Distributed RC Lines 3-16
 - Model Equations B-113
 - Model Parameters 3-17
 - Models 3-16
- Distribution 7-10
- Documentation Conventions 1-3
- Double Exponential (Dexpon) 7-10, 7-11
- Drain Area Factor 7-29
- Drain Diffusion Area 7-29
- DUMPIC 7-30

E

- ECHOFIELD 7-30
- END Statement 1-5
- Enhancements 1-6
- Equations B-1
- Equivalent Input Noise 7-23
- Error Messages D-1
 - .INCLUDE, .LIBRARY, and END Library Errors D-2
 - .NAME Instruction Errors D-2
 - .NODESET Error D-2
 - Circuit Checker Errors D-2
 - Command Error D-3
 - DC Solution Errors D-3
 - General Errors D-3
 - Initial Conditions Error D-4
 - Input Processor Errors D-4
 - Input Source Errors D-5
 - Matrix Error D-5
 - Model Specification Errors D-5
 - MOSFET Table Model Errors D-5
 - Mutual Inductor Error D-6
 - Sensitivity Analysis Errors D-6
 - Subcircuit Errors D-6
 - Tolerance Setting Error D-6
 - Transfer Function Error D-6
 - Transient Analysis Errors D-6
 - Transmission Line Errors D-6

-
- User-Defined Element Errors D-7
 - Example DIABLO function C-7
 - EXP 2-12
 - Exponential Source 2-12
 - F**
 - Flicker Noise 7-23
 - Flip-Flop Devices 5-3
 - Flip-Flop Models 5-4
 - Flux Tolerance 7-27
 - FLUXTOL 7-27
 - Fourier Analysis 7-16
 - freq C-16
 - Frequency Analysis 7-2
 - FSIN 2-13, 2-14
 - Full-Wave Rectified Sinusoidal Source 2-13, 2-14
 - func C-2
 - Function Body C-8
 - Algebraic Expressions and Function Calls C-11
 - Comments C-11
 - Conditional Statements C-12
 - Logical Operators C-13
 - Print Statement C-14
 - Syntax Rules C-8
 - Variables and Numbers C-8
 - Function Calls, Algebraic Expressions and C-11
 - Function Names, Multi-Defined C-20
 - Function Types, Special C-19
 - lin_func C-19
 - step_func C-19
 - Functions 1-9, 1-10, 1-11
 - G**
 - Gamma 7-10, 7-11
 - Gate Devices 5-5
 - Gate Models 5-6
 - Gaussian (Gauss) 7-10
 - Gear's Method 7-26
 - Geometries 7-29
 - Globally Defined Nodes 7-16
 - GMIN 7-25
 - Graphic Display 7-3
 - I**
 - IABYPASS 7-27
 - INDEP 7-28
 - Independent Sources 2-11
 - Exponential 2-12
 - Full-Wave Rectified Sinusoidal 2-13, 2-14
 - Piecewise-Linear 2-16
 - Pulse 2-15
 - Single-Frequency FM 2-17
 - Sinusoidal 2-17
 - Inductors 3-8, C-3
 - Mutual Inductors 3-8
 - Polynomial 3-10
 - Semiconductor Models 3-9
 - User-Defined 3-10
 - Initial Conditions 7-17
 - Initial Node Voltages 7-16
 - Initial Node-Set 7-22
 - Input Options 7-29
 - ASPEC 7-29
 - DEFAD 7-29
 - DEFAS 7-29
 - DEFL 7-29
 - DEFNRD 7-29
 - DEFNRS 7-29
 - DEFPD 7-29
 - DEFPS 7-29
 - DEFW 7-29
 - SCALE 7-29
 - SCALM 7-29
 - TIMEPAIR 7-29
 - WL 7-29
 - ZDEFAD 7-29
 - ZDEFAS 7-29
 - ZDEFL 7-29
 - ZDEFNRB 7-29
 - ZDEFNRD 7-29
 - ZDEFNRG 7-29
 - ZDEFPD 7-29
 - ZDEFPS 7-29
 - ZDEFW 7-29
 - ZDERNRS 7-29
 - Instruction Lines 1-5
 - Instructions 1-5, 7-1
 - .AC 7-2
 - .ACQUIRE 7-3
 - .ALTER 7-4
 - .DC 7-6
 - .DELETE 7-8
 - .DISTO 7-9
 - .DISTR 7-10
 - .DUMP 7-13
 - .END 7-15
 - .ENDDEL 7-15
 - .ENDL 7-18
 - .FOUR 7-15
 - .GLOBAL 7-16
 - .IC 7-16
 - .INCLUDE 7-17
 - .LIB 7-18
 - .MEASURE 7-19
 - .MODEL 7-20
 - .MONTE 7-21
 - .NODESET 7-22
 - .NOISE 7-23
 - .OP 7-24
 - .OPTIONS 7-24
-

- .PARAM 7-31
- .PLOT 7-32
- .PRINT 7-33
- .RESTART 7-35
- .SENS 7-35
- .SEQUEL 7-36
- .SUBCKT 7-37
- .TEMP 7-38
- .TF 7-39
- .TRAN 7-39
- .VARY 7-42
- .WIDTH 7-43
- VBAS 7-1
- Integration Method 7-26
- Inverter A-4
- IRBYPASS 7-27
- Iteration Limit 7-26
- ITL1 7-26
- ITL5 7-26

J

- JFET Devices 4-10
 - JFET Models 4-11
 - Model Equations B-13
 - Model Parameters 4-12
- Job Control 7-1

K

- Keyword
 - func C-2
 - poly C-2
- Keywords C-9

L

- last_time C-15
- Legal Names 1-7
- Legal Numbers 1-9
- Length Factor 7-29
- Library
 - Call 7-19
 - Data 7-18
 - File 7-18
- Limit 7-26
- LIMPTS 7-26
- lin_func C-5, C-19
- Logical Operators C-13
- Log-normal (Lognor) 7-10, 7-12

M

- max_step and maxstep C-16
- MAXFUNC 7-28
- maxstep C-5
- MESFET Devices 4-13
 - MESFET Models 4-14
 - Model Equations B-15

- Level 1 B-15
- Level 2 B-17
- Level 3 B-19
- QOPT = 4 B-22
- Model Parameters 4-15
- Messages D-1
- METHOD 7-26
- Meyer Charge Model 7-26
- Minimum Conductance 7-25
- Model Equations B-1
 - BJT B-2
 - BSIM2 B-98
 - BSIM3 Charge Model B-100
 - Diode B-8
 - Distributed RC Lines B-113
 - JFET B-13
 - MESFET B-15
 - MOSFET B-24
- Model Parameters
 - Digital 5-7
- Model Tracking 7-28
- Monte Carlo Analysis 7-21
- MOSFET Devices 4-20
 - Example A-3
 - Model Equations B-24, B-100
 - All Levels B-24
 - ASPEC Charge Model B-110
 - BSIM Charge Model B-95
 - Level 1 B-27
 - Level 10 B-55
 - Level 11 B-65
 - Level 2 B-28
 - Level 20 B-73
 - Level 3 B-33
 - Level 4 B-37
 - Level 6 B-45
 - Level 8 B-52
 - Meyer Charge Model B-87
 - Ward-Dutton Charge Model B-91
 - Yang-Chatterjee Charge Model B-83
- Model Parameters 4-23
- MOSFET Models 4-22
- Parasitics 4-40
- Multi-Defined Function Names C-20
- Multidimensional Capacitors C-5
- Multidimensional Inductors C-5
- Mutual Inductors 3-8

N

- Names 1-7
- NMONTE 7-28
- NOBYPASS 7-26
- NODE 7-30
- NODECAPS 7-30
- Nodes Initialized 7-22

NODESET 7-26
Noise Analysis 7-23
NOLIMJFT 7-26
NOLIMMES 7-26
NOLIMMOS 7-26
Nominal Temperature 7-27
NOMOD 7-30
NOMONT 7-28
Nonlinear Dependent Sources 2-2
NOPAGE 7-30
NOSUBS 7-26
Numbers 1-9, C-10
Numbers, Variables and C-8
NUMDGT 7-30
Numerical Derivative Computation 7-25, 7-26

O

old_ C-18
old_value C-16
Operating Points 7-24
Options, Simulation 7-24
OPTS 7-30
OUTFMT 7-30
Output 1-3
Output Options 7-30
 ACCT 7-30
 CO 7-30
 DUMPIC 7-30
 ECHOFILE 7-30
 NODE 7-30
 NODECAPS 7-30
 NOMOD 7-30
 NOPAGE 7-30
 NUMDGT 7-30
 OPTS 7-30
 OUTFMT 7-30

P

Parameter Formats
 Passive Elements 3-1
Parameter Names 7-31
Parasitic Resistance Factor 7-29
Parasitics
 MOSFET 4-40
Partitioned 7-36
Passive Element Description Lines 1-5
Passive Elements 3-1
 .MODEL Specification 3-1
 Capacitors 3-4
 Inductors 3-8
 Mutual Inductors 3-8
 Parameter Formats 3-1
 Resistors 3-11
 Transformer Core 3-2
 Transmission Lines 3-15

 Windings 3-17
Perimeter 7-29
Periphery Factor 7-29
phase C-19
Piecewise-Linear Source 2-16
Plot 7-32
PMAx 7-28
poly C-2
Polynomial Capacitors 3-7
Polynomial Inductors 3-10
Power 7-28
POWERDC 7-28
POWERTR 7-28
Print 7-33
Print Statements C-14
Program 1-2
Program Enhancements 1-6
Program Input File 7-17
PULSE 2-15
Pulsed Source 2-15
PWL 2-16

Q

QOPT 7-26

R

Random Number Generator 7-28
RC Lines 3-16
RELDELTA 7-26
RELTOL 7-28
Reserved Prefixes C-11, C-18
 d_d C-18
 d_dt C-18
 old_ C-18
Resistor
 Model Parameters 3-13
Resistors 3-11
 Semiconductor Models 3-12
Return Statements C-13
RTL Inverter A-4

S

SCALE 7-29
SCALM 7-29
SEED 7-28
Semiconductor Capacitor Models 3-6
Semiconductor Device 1-5
Semiconductor Devices 4-1
 .MODEL Specification 4-2
 BJT 4-3
 Diode 4-7
 JFET 4-10
 MESFET 4-13
 MOSFET 4-20
 Switches 3-16

Index

- Uniform Distributed RC Lines 3-16
- Semiconductor Inductor Models 3-9
- Semiconductor Resistor Models 3-12
- Sensitivity Analysis 7-35
- SFFM 2-17
- Simulation 1-4
- Simulation Options
 - Algorithm 7-25
 - Input 7-29
 - Output 7-30
 - Statistical 7-28
 - Stress 7-28
 - Tolerance 7-27
- Simulation Requirements 1-4
- SIN 2-17
- Single-Frequency FM Source 2-17
- Sinusoidal Source 2-17
- Small-Signal AC Analysis Related Variable C-11
- Solution Methods 7-25, 7-27
- Source Area Factor 7-29
- Source Description Lines 1-5
- Source Diffusion Area 7-29
- Sources 1-3, 2-1
 - Dependent 2-2
 - Independent 2-11
- Special Features C-14
 - Function Types, Special C-19
 - Multi-Defined Function Names C-20
 - Reserved Prefixes C-18
 - Summary C-21
 - System Flags C-19
 - System Reserved Variables (internally set by VBAS) C-15
 - System Reserved Variables (set by the user) C-16
- SPICE 1-2, 1-6, 1-7, 2-16, 4-11, 7-1, 7-25, 7-27, 7-30, 7-32
- SPICE Interpretations 7-27
- Statistical Analysis 7-10, 7-22, 7-28
- Statistical Simulation Options 7-28
 - INDEP 7-28
 - NMONTE 7-28
 - SEED 7-28
 - WORST 7-28
- Statistical Spread 1-11
- status C-19
- step_func C-5, C-19
- Stochastic Tracking 1-12
- Stress Simulation Options 7-28
 - POWERDC 7-28
 - POWERTR 7-28
- Subcircuit Definition 1-5, 7-31, 7-37, 7-38
- Subcircuits 1-9, 6-1
 - Definitions 6-2
 - Expansions 6-4

- Functions 6-5
- SUBS 7-26
- Summary C-21
- Switches 3-16
 - Model Parameters 3-14
 - Switch Models 3-14
- Syntax Rules C-8
- System Flags C-19
 - deriv C-19
 - phase C-19
 - status C-19
 - time_flag C-19
- System Reserved Variables (internally set by VBAS) C-15
 - freq C-16
 - last_time C-15
 - old_value C-16
 - temp C-16
 - time C-15
 - timestep C-15
 - vtherm C-16
- System Reserved Variables (set by the user) C-16
 - breakpoint C-17
 - max_step and maxstep C-16

T

- temp C-16
- Temperature Analysis 7-27, 7-38
- Temperature Related Variables C-11
- Thermal Noise 7-23
- time C-15
- Time Points 7-26
- time_flag C-19
- Time-dependent Source 2-11
- TIMEPAIR 7-29
- timestep C-15
- Timing Information 7-19
- Title Statement 1-4
- TNOM 7-27
- Tolerance Simulation Options 7-27
 - ABSTOL 7-27
 - CHGTOL 7-27
 - DCABSTOL 7-27
 - DCRELTOL 7-27
 - DCVNTOL 7-27
 - FLUXTOL 7-27
 - IABYPASS 7-27
 - IRBYPASS 7-27
 - RELTOL 7-28
 - VABYPASS 7-28
 - VNTOL 7-28
 - VRBYPASS 7-28
- Tolerance Value 7-27
- TOX 7-29
- Tracking 1-12

Transfer Function Analysis 7-39
Transformer Core 3-2
Transformer Windings 3-17
Transformers
 Model Parameters 3-4
 Models 3-3
Transient Analysis 7-27, 7-39
Transient Analysis Related Variables C-11
Transmission Lines 3-15
Trapezoidal Method 7-26
TRMODE 7-27
TUSEIC 7-27
TUSENSET 7-27

U

Uniform 7-10
Uniform Distributed RC Lines 3-16
 Model Parameters 3-17
 Models 3-16
Uniform Distribution 7-10
User-Defined
 Capacitors 3-7
 Inductors 3-10

V

VABYPASS 7-28
Variables and Numbers C-8
 Built-in Constants C-9
 Built-in Functions C-9
 Keywords C-9
 Reserved Prefixes C-11
Variables, Other C-11
 Other Variables C-11
 Small-Signal AC Analysis Related Variable C-11
 Temperature Related Variables C-11
 Transient Analysis Related Variables C-11
Variations of Parameters 7-42
VBAS Device Quick Reference 1-8
VBAS Instructions 7-1
VBAS Instructions, Unique 1-6
VBASE Features
 DIABLO C-15
VCCS 2-7
VCVS 2-3
VeriBest Analog Simulator 1-2
VNTOL 7-28
Voltage Controlled Current Source 2-7, C-2
Voltage Controlled Voltage Source 2-3
Voltage Tolerance 7-28
VRBYPASS 7-28
vtherm C-16

W

Ward-Dutton Charge Model 7-26

Weibull (Weibul) 7-10, 7-12
Width 7-43
Width Factor 7-30
Windings 3-17
WL 7-29
WORST 7-28
Worst Case Analysis 7-28
WRITE 7-21
Writing a DIABLO Function C-6
 Basic Framework C-6
 Function Body C-8
 Special Features C-14

Y

Yang-Chatterjee Charge Model 7-26

Z

ZDEFAD 7-29
ZDEFAS 7-29
ZDEFL 7-29
ZDEFNRB 7-29
ZDEFNRD 7-29
ZDEFNRG 7-29
ZDEFPD 7-29
ZDEFPS 7-29
ZDEFW 7-29
ZDERNRS 7-29
Zero Charge Model 7-26

