Using This Manual

This manual describes the Star-Hspice circuit and device simulation software and how to use it.

Audience

This manual is intended for design engineers who use Star-Hspice to develop, test, analyze, and modify circuit designs.

How this Manual is Organized

The manual set is divided into four volumes as follows:

- Volume I (Chapters 1 through 13) describes how to run simulations with Star-Hspice and evaluate the results.
- Volume II (Chapters 14 through 19) describes how Star-Hspice models passive circuit elements and active devices of all types.
- Volume III (Chapters 20 through 23) describes MOSFET models.
- Volume IV contains detailed applications and examples of how to use Star-Hspice for a wide variety of circuit simulations (Chapters 24 through 31). Volume IV also contains reference material (Appendices A through F).

Related Documents

The following documents pertain to this guide:

- Star-Hspice, Star-Time, and AvanWaves Installation Guide
- Star-Sim and Star-Time User Guides
- Star-Hspice and AvanWaves Release Notes
If you have questions or suggestions about this documentation, send them to:

techpubs@avanticorp.com

Conventions

Avant! documents use the following conventions, unless otherwise specified:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>menuName &gt; commandName</td>
<td>Indicates the name of the menu and the command name. For example:</td>
</tr>
<tr>
<td></td>
<td>Cell &gt; Open Refers to the Open command in the Cell menu.</td>
</tr>
<tr>
<td>Tool: menuName &gt; commandName</td>
<td>Indicates that a command is accessible only through an application tool. Tool is the tool through which you access the command, menuName is the name of the menu, and commandName is the name of the command. For example:</td>
</tr>
<tr>
<td></td>
<td>Data Prep: Pin Solution &gt; Via Refers to the Via command on the Pin Solution menu, which you access by selecting Data Prep from the Tools menu in Apollo.</td>
</tr>
<tr>
<td>courier</td>
<td>In text, this font indicates a function or keyword that you must type exactly as shown. In examples, this font indicates system prompts, text from files, and messages printed by the system.</td>
</tr>
<tr>
<td>Convention</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>courier italic</code></td>
<td>Arguments appear in this font when the value of an argument is a string. The string must be enclosed with quotation marks.</td>
</tr>
<tr>
<td><code>times italic</code></td>
<td>Indicates commands, functions, arguments, file names, and variables within a line of text.</td>
</tr>
<tr>
<td></td>
<td>When a variable is included in italicized text, the variable is enclosed by angle brackets (<code>&lt;&gt;</code>). For example, “the name of the technology file is <code>&lt;libraryName&gt;.tf, where </code>&lt;libraryName&gt;` is the name of the library.”</td>
</tr>
<tr>
<td><code>[ ]</code></td>
<td>Denotes optional arguments, such as: <code>pin1 [pin2, ...pinN]</code> In this example, you must enter at least one pin name, the other arguments are optional.</td>
</tr>
<tr>
<td><code>{instanceName orientation} ...</code></td>
<td>Indicates that you can repeat the construction enclosed in braces.</td>
</tr>
<tr>
<td>.</td>
<td>Indicates that text was omitted.</td>
</tr>
<tr>
<td>'(item1 item2)'</td>
<td>An apostrophe followed by parentheses indicate that the text within the parentheses enclose a list. When the list contains multiple items, the items are separated by spaces. Type this information exactly as it appears in the syntax.</td>
</tr>
</tbody>
</table>
Obtaining Customer Support

If you have a maintenance contract with Avant!, you can obtain customer support by:

- Contacting your local Application Engineer (AE)
- Calling the Avant! Corporate office from 8:00 AM through 5:00 PM Pacific Standard Time (PST) at:
  1-800-346-5953
  Ask the receptionist for customer support.
- Emailing a description of the problem to the Hspice Support Center at:
  hspice_nw@avanticorp.com

Other Sources of Information

The Avant! external web site provides information for various products. You can access our web site at:

http://www.avanticorp.com

From our web site, you can register to become a member of the Avant! Users Research Organization for Real Applications (AURORA) user’s group. By participating, you can share and exchange information pertaining to Integrated Circuit Design Automation (ICDA).
Contents

Using This Manual ........................................................................................................ iii
  Audience ........................................................................................................ iii
  How this Manual is Organized ........................................................................ iii
  Related Documents ............................................................................................ iii
  Conventions ........................................................................................................ iv
  Obtaining Customer Support ........................................................................... vi
  Other Sources of Information .......................................................................... vi

Contents ....................................................................................................................... vii

Chapter 1 - Introducing Star-Hspice ................................................................. 1-1
  Star-Hspice Applications ............................................................................... 1-2
  Star-Hspice Features ...................................................................................... 1-3
  Star-Hspice Platforms .................................................................................... 1-6
  Examining the Simulation Structure ............................................................ 1-7
  Understanding the Data Flow ......................................................................... 1-9
    Simulation Process Overview ...................................................................... 1-10

Chapter 2 - Getting Started ............................................................................... 2-1
  AC Analysis of an RC Network ...................................................................... 2-2
  Transient Analysis of an RC Network ............................................................ 2-5
  Transient Analysis of an Inverter .................................................................... 2-7
Chapter 3 - Specifying Simulation Input and Controls ................................. 3-1

Using Netlist Input Files .............................................................................. 3-2
  Input Netlist File (<design>.sp) Guidelines ............................................ 3-2
  Input Netlist File Sections and Chapter References .............................. 3-7

Input Netlist File Composition .................................................................. 3-9
  Title of Simulation and .TITLE Statement ............................................ 3-9
  Comments ............................................................................................... 3-9
  Element and Source Statements ............................................................. 3-10
    .SUBCKT or .MACRO Statement ....................................................... 3-12
    .ENDS or .EOM Statement ................................................................. 3-14
  Subcircuit Call Statement ..................................................................... 3-14
  Element and Node Naming Conventions .............................................. 3-16
    .GLOBAL Statement .......................................................................... 3-19
    .TEMP Statement ............................................................................. 3-20
    .DATA Statement ............................................................................. 3-21
    .INCLUDE Statement ....................................................................... 3-29
    .MODEL Statement ........................................................................... 3-30
    .LIB Call and Definition Statements ............................................ 3-32
    .OPTIONS SEARCH Statement ....................................................... 3-36
    .PARAM Statement .......................................................................... 3-37
    .PROTECT Statement ..................................................................... 3-40
    .UNPROTECT Statement ................................................................... 3-40
    .ALTER Statement ........................................................................ 3-40
    .DEL LIB Statement ......................................................................... 3-42
    .END Statement ............................................................................... 3-45

Using Subcircuits ......................................................................................... 3-47
  Hierarchical Parameters ....................................................................... 3-48
  Undefined Subcircuit Search ............................................................... 3-49

Discrete Device Libraries .......................................................................... 3-50
  DDL Library Access ........................................................................... 3-50
  Vendor Libraries ................................................................................ 3-51
  Subcircuit Library Structure .................................................................. 3-52
Chapter 5 - Using Sources and Stimuli

Independent Source Elements
Source Element Conventions
Independent Source Element

Star-Hspice Independent Source Functions
Pulse Source Function
Sinusoidal Source Function
Exponential Source Function
Piecewise Linear (PWL) Source Function
Data Driven Piecewise Linear Source Function
Single-Frequency FM Source Function
Amplitude Modulation Source Function

Using Voltage and Current Controlled Elements
Polynomial Functions
Piecewise Linear Function

Voltage Dependent Voltage Sources — E Elements
Voltage Controlled Voltage Source (VCVS)
Behavioral Voltage Source
Ideal Op-Amp
Ideal Transformer

Voltage Dependent Current Sources — G Elements
Voltage Controlled Current Source (VCCS)
Behavioral Current Source
Voltage Controlled Resistor (VCR)
Voltage Controlled Capacitor (VCCAP)

Dependent Voltage Sources — H Elements
Current Controlled Voltage Source — (CCVS)
Current Dependent Current Sources — F Elements
Current Controlled Current Source (CCCS)
Digital and Mixed Mode Stimuli ................................................................. 5-50
U Element Digital Input Elements and Models ........................................... 5-50
Specifying a Digital Vector File................................................................. 5-60
Defining Tabular Data ............................................................................. 5-64
Defining Vector Patterns ......................................................................... 5-67
Modifying Waveform Characteristics...................................................... 5-73

Chapter 6 - Multi-Terminal Networks ......................................................... 6-1
Using Scattering Parameter Element .......................................................... 6-2
Frequency Table Model .......................................................................... 6-4

Chapter 7 - Parameters and Functions ....................................................... 7-1
Using Parameters in Simulation ................................................................. 7-2
Parameter Definition .................................................................................. 7-2
Parameter Assignment ............................................................................. 7-3
User-Defined Function Parameters ............................................................ 7-4
Subcircuit Default Definitions ................................................................. 7-5
Predefined Analysis Function ................................................................... 7-6
Measurement Parameters ......................................................................... 7-6
Multiply Parameter ................................................................................... 7-7
Using Algebraic Expressions ..................................................................... 7-8
Algebraic Expressions for Output ............................................................. 7-8
Built-In Functions ..................................................................................... 7-9
User-Defined Functions .......................................................................... 7-11
Parameter Scoping and Passing ................................................................. 7-13
Library Integrity ....................................................................................... 7-13
Reusing Cells ............................................................................................ 7-14
Creating Parameters in a Library .............................................................. 7-14
Parameter Defaults and Inheritance ......................................................... 7-17
Parameter Passing Problems ..................................................................... 7-19
Chapter 8 - Specifying Simulation Output ..................................................... 8-1

Using Output Statements ............................................................................... 8-2
Output Commands ......................................................................................... 8-2
Output Variables ......................................................................................... 8-3

Displaying Simulation Results ..................................................................... 8-5
.PRINT Statement ......................................................................................... 8-5
.PLOT Statement .......................................................................................... 8-8
.PROBE Statement ......................................................................................... 8-9
.GRAPH Statement ........................................................................................ 8-10
Print Control Options .................................................................................. 8-14
Subcircuit Output Printing ........................................................................... 8-19

Selecting Simulation Output Parameters ...................................................... 8-21
DC and Transient Output Variables .............................................................. 8-21
AC Analysis Output Variables ................................................................... 8-29
Element Template Output ........................................................................... 8-35

Specifying User-Defined Analysis (.MEASURE) ........................................... 8-37
Measure Parameter Types .......................................................................... 8-38
.MEASURE Statement: Rise, Fall, and Delay ............................................... 8-39
FIND and WHEN Functions ....................................................................... 8-43
Equation Evaluation .................................................................................... 8-45
Average, RMS, MIN, MAX, INTEG, and Peak-To-Peak ................................ 8-46
INTEGRAL Function .................................................................................... 8-48
DERIVATIVE Function ............................................................................... 8-48
ERROR Function ......................................................................................... 8-51

Element Template Listings ........................................................................... 8-55

Chapter 9 - Specifying Simulation Options .................................................. 9-1

Setting Control Options .............................................................................. 9-2
.OPTIONS Statement .................................................................................... 9-2
General Control Options ............................................................................ 9-5

Input and Output Options ................................................................. 9-5
CPU Options ..................................................................................... 9-11
Interface Options ........................................................................... 9-11
Analysis Options ............................................................................ 9-13
Error Options .................................................................................. 9-15
Version Options ............................................................................. 9-15
Model Analysis Options ................................................................. 9-16
General Options ............................................................................. 9-16
MOSFET Control Options .............................................................. 9-18
Inductors ........................................................................................ 9-19
BJTs .................................................................................................. 9-19
Diodes ............................................................................................ 9-19
DC Operating Point, DC Sweep, and Pole/Zero ......................... 9-20
Accuracy ........................................................................................ 9-20
Matrix-Related ............................................................................... 9-23
Input and Output ........................................................................... 9-28
Convergence .................................................................................. 9-29
Pole/Zero Control Options ............................................................ 9-35
Transient and AC Small Signal Analysis ..................................... 9-38
Accuracy ........................................................................................ 9-38
Speed ............................................................................................ 9-41
Timestep ........................................................................................ 9-43
Input and Output ........................................................................... 9-50

Chapter 10 - DC Initialization and Operating Point Analysis ....... 10-1
Understanding the Simulation Flow ............................................... 10-2
Performing Initialization and Analysis ......................................... 10-3
Setting Initial Conditions for Transient Analysis ....................... 10-5
Using DC Initialization and Operating Point Statements .......... 10-6
.OP Statement — Operating Point ............................................... 10-6
Element Statement IC Parameter ............................................... 10-9
.IC and .DCVOLT Initial Condition Statements ........................................ 10-9
.NODESET Statement ........................................................................ 110-10
Using .SAVE and .LOAD Statements .............................................. 110-11
.DC Statement—DC Sweeps ................................................................ 110-14
Using Other DC Analysis Statements .............................................. 110-19
.SENS Statement — DC Sensitivity Analysis .................................... 110-19
.TF Statement — DC Small-Signal Transfer Function Analysis ........ 110-20
.PZ Statement— Pole/Zero Analysis ............................................... 110-21
Setting DC Initialization Control Options ...................................... 110-22
Option Descriptions ........................................................................ 110-22
Pole/Zero Analysis Options .............................................................. 110-31
Specifying Accuracy and Convergence .......................................... 110-34
Accuracy Tolerances ...................................................................... 110-34
Accuracy Control Options ................................................................. 110-36
Convergence Control Option Descriptions .................................... 110-36
Autoconverge Process .................................................................. 110-42
Reducing DC Errors ........................................................................ 110-45
Shorted Element Nodes .................................................................. 110-47
Conductance Insertion Using DCSTEP .......................................... 110-47
Floating Point Overflow................................................................. 110-48
Diagnosing Convergence Problems .............................................. 110-49
Nonconvergence Diagnostic Table ................................................. 110-49
Traceback of Nonconvergence Source ......................................... 110-51
Solutions for Nonconvergent Circuits ........................................... 110-51

Chapter 11 - Performing Transient Analysis ..................................... 111-1
Understanding the Simulation Flow ................................................. 111-2
Understanding Transient Analysis .................................................. 111-3
Initial Conditions for Transient Analysis ....................................... 111-3
Chapter 13 - Statistical Analysis and Optimization ........................................ 13-1

Specifying Analytical Model Types .............................................................. 13-2
Simulating Circuit and Model Temperatures .................................................. 13-4
  Temperature Analysis .............................................................................. 13-5
  .TEMP Statement .................................................................................. 13-6
Performing Worst Case Analysis .................................................................. 13-8
  Model Skew Parameters .......................................................................... 13-8
Performing Monte Carlo Analysis ................................................................. 13-14
  Monte Carlo Setup ................................................................................ 13-14
  Monte Carlo Output ............................................................................... 13-15
  .PARAM Distribution Function Syntax ................................................ 13-16
  Monte Carlo Parameter Distribution Summary .................................... 13-18
  Monte Carlo Examples .......................................................................... 13-18
Worst Case and Monte Carlo Sweep Example .............................................. 13-26
  HSPICE Input File .............................................................................. 13-26
  Transient Sigma Sweep Results ............................................................ 13-28
  Monte Carlo Results ............................................................................. 13-30
Optimization ............................................................................................... 13-35
  Optimization Control ........................................................................... 13-36
  Simulation Accuracy ............................................................................. 13-36
  Curve Fit Optimization ......................................................................... 13-37
  Goal Optimization ............................................................................... 13-37
  Performing Timing Analysis ................................................................. 13-37
  Understanding the Optimization Syntax .............................................. 13-38
Optimization Examples .................................................................................. 13-44
  MOS Level 3 Model DC Optimization .................................................. 13-44
  MOS Level 13 Model DC Optimization ............................................... 13-48
  RC Network Optimization .................................................................... 13-51
CMOS Tristate Buffer Optimization ..................................................... 13-55
BJT S Parameters Optimization .......................................................... 13-60
BJT Model DC Optimization ................................................................. 13-63
GaAsFET Model DC Optimization ....................................................... 13-67
MOS Op-amp Optimization ................................................................. 13-70

Chapter 14 - Using Passive Device Models ....................................... 14-1
Resistor Device Model and Equations .................................................. 14-2
  Wire RC Model .............................................................................. 14-2
  Resistor Model Equations ............................................................... 14-5
Capacitor Device Model and Equations .............................................. 14-9
  Capacitance Model ...................................................................... 14-9
  Capacitor Device Equations ........................................................... 14-10
Inductor Device Model and Equations .............................................. 14-12
  Inductor Core Models .................................................................. 14-12
  Magnetic Core Element Outputs .................................................... 14-16
  Inductor Device Equations ............................................................... 14-17
Jiles-Atherton Ferromagnetic Core Model ......................................... 14-19

Chapter 15 - Using Diodes ................................................................. 15-1
Understanding the Diode Types ......................................................... 15-3
Using Diode Model Statements ......................................................... 15-4
  Setting Control Options ................................................................. 15-4
Specifying Junction Diode Models ..................................................... 15-6
  Using the Junction Model Statement ............................................. 15-7
  Using Junction Model Parameters ................................................ 15-8
  Providing Geometric Scaling for Diode Models .............................. 15-14
  Defining Diode Models ................................................................. 15-16
  Determining Temperature Effects on Junction Diodes .................. 15-19
Using Junction Diode Equations ....................................................... 15-22
Using Junction DC Equations ............................................................... 15-23
Using Diode Capacitance Equations .................................................... 15-25
Using Noise Equations .......................................................................... 15-28
Temperature Compensation Equations ................................................. 15-28
Using the Junction Cap Model................................................................. 15-33
    Setting Juncap Model Parameters ....................................................... 15-35
Theory ................................................................................................... 15-37
    JUNCAP Model Equations ................................................................... 15-38
Using the Fowler-Nordheim Diode.......................................................... 15-48
Converting National Semiconductor Models............................................. 15-50

Chapter 16 - Using BJT Models ............................................................... 16-1
Using BJT Models ....................................................................................... 16-2
    Selecting Models ..................................................................................... 16-2
Understanding the BJT Model Statement ............................................... 16-4
    Using BJT Basic Model Parameters....................................................... 16-5
    Handling BJT Model Temperature Effects ........................................... 16-14
Using BJT Device Equivalent Circuits ..................................................... 16-20
    Scaling ................................................................................................... 16-20
    Understanding the BJT Current Convention ........................................ 16-20
    Using BJT Equivalent Circuits.............................................................. 16-21
Using BJT Model Equations (NPN and PNP) ........................................... 16-28
    Understanding Transistor Geometry in Substrate Diodes..................... 16-28
    Using DC Model Equations .................................................................. 16-29
    Using Substrate Current Equations ....................................................... 16-31
    Using Base Charge Equations ............................................................... 16-32
    Using Variable Base Resistance Equations.......................................... 16-32
Using BJT Capacitance Equations ............................................................. 16-34
    Using Base-Emitter Capacitance Equations.......................................... 16-34
    Determining Base Collector Capacitance ............................................. 16-36
Using Substrate Capacitance ................................................................. 16-38
Defining BJT Noise Equations ................................................................. 16-40
Using BJT Temperature Compensation Equations ................................. 16-42
Using Energy Gap Temperature Equations ............................................. 16-42
Using Saturation and Beta Temperature Equations,
TLEV=0 or 2 ..................................................................................... 16-42
Using Saturation and Temperature Equations, TLEV=1 ......................... 16-44
Using Saturation Temperature Equations, TLEV=3 ............................... 16-45
Using Capacitance Temperature Equations .......................................... 16-46
Using Parasitic Resistor Temperature Equations ................................. 16-49
Using BJT LEVEL=2 Temperature Equations ....................................... 16-49
Using the BJT Quasi-Saturation Model .................................................... 16-50
Using Epitaxial Current Source Iepi ....................................................... 16-51
Using Epitaxial Charge Storage Elements Ci and Cx ............................. 16-52
Converting National Semiconductor Models ......................................... 16-55
Using the VBIC Bipolar Transistor Model .............................................. 16-57
Understanding the History of VBIC ....................................................... 16-57
VBIC Parameters .................................................................................. 16-57
Noise Analysis ...................................................................................... 16-58
LEVEL 6 Philips Bipolar Model (MEXTRAM LEVEL 503) ....................... 16-68
LEVEL 6 Element Syntax ....................................................................... 16-69
LEVEL 6 Model Parameters ................................................................. 16-69
LEVEL 6 Philips Bipolar Model (MEXTRAM LEVEL 504) ....................... 16-75
Notes for HSPICE Users ..................................................................... 16-76
LEVEL 6 Model Parameters (504) ......................................................... 16-77
LEVEL 8 HiCUM Model ......................................................................... 16-93
What is the HiCUM Model? ................................................................. 16-93
HiCUM Model Advantages .................................................................. 16-93
Star-Hspice HiCUM Model vs. Public HiCUM Model ........................... 16-95
Model Implementation ......................................................................... 16-95
Chapter 17 - Using JFET and MESFET Models................................. 17-1

Understanding JFETs .................................................................................. 17-2
Specifying a Model ...................................................................................... 17-3
Understanding the Capacitor Model ............................................................ 17-5
  Model Applications ...................................................................................... 17-5
  Control Options ......................................................................................... 17-6
Using JFET and MESFET Equivalent Circuits ........................................... 17-7
  Scaling ..................................................................................................... 17-7
  Understanding JFET Current Convention ............................................... 17-7
  JFET Equivalent Circuits ........................................................................ 17-8
Using JFET and MESFET Model Statements ........................................... 17-13
  JFET and MESFET Model Parameters .................................................... 17-13
  Gate Diode DC Parameters .................................................................... 17-15
  JFET and MESFET Capacitances ............................................................. 17-25
    Capacitance Comparison (CAPOP=1 and CAPOP=2) ......................... 17-29
  JFET and MESFET DC Equations .......................................................... 17-31
JFET and MESFET Noise Models.............................................................. 17-35
  Noise Parameters ................................................................................... 17-35
  Noise Equations .................................................................................... 17-35
  Noise Summary Printout Definitions ....................................................... 17-37
JFET and MESFET Temperature Equations .............................................. 17-38
  Temperature Compensation Equations .................................................. 17-41
Using TriQuint Model (TOM) Extensions to LEVEL=3 ......................... 17-45
  TOM Model Parameters ........................................................................ 17-46

Chapter 18 - Using Transmission Lines.................................................. 18-1

Using Transmission Line Equations and Parameters............................. 18-3
Using Frequency-Dependent Resistance and Conductance Matrices ..... 18-4
Determining Matrix Properties................................................................. 18-4
Understanding Wave Propagation on Transmission Lines .................. 18-5
Propagating a Voltage Step in a Transmission Line .............................. 18-6
Handling Line-to-Line Junctions ............................................................. 18-8
Using the W Element ............................................................................. 18-10
W Element and Field Solver Updates ..................................................... 18-12
W Element Transmission Line Properties Inputs.................................... 18-14
Tabular RLGC Model for W Element ..................................................... 18-27
W Element Syntax for a Tabular RLGC Model....................................... 18-27
Tabular RLGC Model Syntax ................................................................ 18-28
Extracting Transmission Line Parameters .............................................. 18-34
Filament Method .................................................................................... 18-34
Modeling Geometries .............................................................. 18-35
Solver Limitations ............................................................... 18-36
Using the Field-Solver Statement Syntax ............................................ 18-36
Defining Material Properties ............................................................... 18-36
Creating Layer Stacks .................................................................... 18-37
Defining Shapes ................................................................................ 18-38
Field-Solver Options ....................................................................... 18-42
Using the Field Solver Model ............................................................. 18-43
Field Solver Examples ....................................................................... 18-45
Example 1: Cylindrical Conductor Above a Ground Plane .................. 18-45
Example 2: Stratified Dielectric Media............................................... 18-47
Example 3: Two Traces Between Two Ground Planes ......................... 18-50
Example 4: Using Field Solver with Monte Carlo Analysis................. 18-52

Chapter 19 - Using IBIS Models.............................................................. 19-1
Understanding IBIS Conventions ......................................................... 19-3
Terminology ...................................................................................... 19-4
Limitations and Restrictions.............................................................. 19-5
Buffers

<table>
<thead>
<tr>
<th>Type</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Buffer</td>
<td>19-6</td>
</tr>
<tr>
<td>Output Buffer</td>
<td>19-8</td>
</tr>
<tr>
<td>Tristate Buffer</td>
<td>19-11</td>
</tr>
<tr>
<td>Input/Output Buffer</td>
<td>19-15</td>
</tr>
<tr>
<td>Open Drain, Open Sink, Open Source Buffers</td>
<td>19-18</td>
</tr>
<tr>
<td>I/O Open Drain, I/O Open Sink, I/O Open Source Buffers</td>
<td>19-18</td>
</tr>
<tr>
<td>Input ECL Buffer</td>
<td>19-19</td>
</tr>
<tr>
<td>Output ECL Buffer</td>
<td>19-20</td>
</tr>
<tr>
<td>Tristate ECL Buffer</td>
<td>19-21</td>
</tr>
<tr>
<td>Input-Output ECL Buffer</td>
<td>19-23</td>
</tr>
</tbody>
</table>

Specifying Common Keywords ................................................................. 19-26

<table>
<thead>
<tr>
<th>Type</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required Keywords</td>
<td>19-26</td>
</tr>
<tr>
<td>Optional Keywords</td>
<td>19-26</td>
</tr>
</tbody>
</table>

Differential Pins ......................................................................................... 19-36

Scaling Buffer Strength ............................................................................ 19-38

<table>
<thead>
<tr>
<th>Type</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffers in subcircuits</td>
<td>19-39</td>
</tr>
</tbody>
</table>

Example ..................................................................................................... 19-42

Additional Notes ........................................................................................ 19-44

<table>
<thead>
<tr>
<th>Type</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keywords</td>
<td>19-44</td>
</tr>
<tr>
<td>Voltage Thresholds</td>
<td>19-44</td>
</tr>
<tr>
<td>.OPTION D_IBIS</td>
<td>19-44</td>
</tr>
</tbody>
</table>

Understanding Warning and Error Messages ............................................ 19-46

References .................................................................................................. 19-49

Chapter 20 - Introducing MOSFETs ............................................................ 20-1

<table>
<thead>
<tr>
<th>Type</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding MOSFET Models</td>
<td>20-3</td>
</tr>
<tr>
<td>Selecting Models</td>
<td>20-4</td>
</tr>
</tbody>
</table>

Selecting MOSFET Model LEVELs ................................................................. 20-4

Selecting MOSFET Capacitors .................................................................... 20-7
Selecting MOS Diodes...........................................................................  20-9
Setting MOSFET Control Options ...................................................  20-11
Using the General MOSFET Model Statement ...............................  20-14
Using Nonplanar and Planar Technologies.......................................  20-15
   Using Field Effect Transistors......................................................  20-16
Using MOSFET Equivalent Circuits ...............................................  20-20
Using a MOSFET Diode Model .....................................................  20-26
   Selecting MOSFET Diode Models ...............................................  20-26
   Enhancing Convergence...........................................................  20-26
   Using MOSFET Diode Model Parameters..................................  20-27
   Using an ACM=0 MOS Diode....................................................  20-31
   Using an ACM=1 MOS Diode....................................................  20-34
   Using an ACM=2 MOS Diode....................................................  20-37
   Using an ACM=3 MOS Diode....................................................  20-41
Using MOS Diode Equations.......................................................  20-45
   DC Current ..............................................................................  20-45
   Using MOS Diode Capacitance Equations.................................  20-46
Using Common Threshold Voltage Equations................................  20-50
   Common Threshold Voltage Parameters..................................  20-50
   Calculating PHI, GAMMA, and VTO..........................................  20-51
Performing MOSFET Impact Ionization.........................................  20-53
   Using Impact Ionization Model Parameters..............................  20-53
   Calculating the Impact Ionization Equations............................  20-53
   Calculating Effective Output Conductance...............................  20-54
   Cascoding Example.................................................................  20-55
   Cascode Circuit......................................................................  20-56
Using MOS Gate Capacitance Models..........................................  20-57
   Selecting Capacitor Models......................................................  20-57
   Introducing Transcapacitance.................................................  20-59
   Using the Operating Point Capacitance Printout.........................  20-61
Using the Element Template Printout ................................................... 20-62
Calculating Gate Capacitance ............................................................... 20-64
Using MOS Gate Capacitance Model Parameters ............................... 20-69
Specifying XQC and XPART for CAPOP=4, 9, 11, 12 and 13 ................. 20-72
Using Overlap Capacitance Equations ............................................... 20-73
Defining CAPOP=0 — SPICE Meyer Gate Capacitances ...................... 20-74
Defining CAPOP=1 — Modified Meyer Gate Capacitances ................. 20-77
Defining CAPOP=2—Parameterized Modified Meyer Capacitances ....... 20-81
Defining CAPOP=3 — Gate Capacitances (Simpson Integration) ......... 20-86
Defining CAPOP=4 — Charge Conservation Capacitance Model ......... 20-87
Defining CAPOP=5 — Gate Capacitance ............................................ 20-94
Defining CAPOP=6 — AMI Gate Capacitance Model .......................... 20-94
Defining CAPOP=13 — BSIM1-based Charge-Conserving Gate Capacitance Model ............................................................... 20-97
Defining CAPOP=39 — BSIM2 Charge-Conserving Gate Capacitance Model ................................................................. 20-97
Calculating Effective Length and Width for AC Gate Capacitance ....... 20-97

Using Noise Models .............................................................................. 20-99
Using Noise Parameters ....................................................................... 20-99
Using Noise Equations ....................................................................... 20-99
Noise Summary Printout Definitions ................................................. 20-101

Using Temperature Parameters and Equations ................................. 20-102
Temperature Parameters .................................................................... 20-102
Using Temperature Equations ............................................................. 20-105

Chapter 21 - Selecting MOSFET Models: LEVEL 1-40 ..................... 21-1

LEVEL 1 IDS: Schichman-Hodges Model ............................................. 21-2
LEVEL 1 Model Parameters .................................................................. 21-2
LEVEL 1 Model Equations .................................................................. 21-5
LEVEL 2 IDS: Grove-Frohman Model ................................................... 21-7
LEVEL 2 Model Parameters .................................................................. 21-7
LEVEL 2 Model Equations .................................................................. 21-11
LEVEL 27 SOSFET Model ................................................................. 21-136
  LEVEL 27 Model Parameters ....................................................... 21-138
Non-Fully Depleted SOI Model ..................................................... 21-142
  Obtaining Model Parameters ..................................................... 21-143
  Fully Depleted SOI Model Considerations ................................. 21-146
LEVEL 28 Modified BSIM Model .................................................... 21-147
  LEVEL 28 Model Parameters ..................................................... 21-148
LEVEL 28 Model Equations ............................................................. 21-155
LEVEL 38 IDS: Cypress Depletion Model ....................................... 21-162
  LEVEL 38 Model Parameters ..................................................... 21-164
  LEVEL 38 Model Equations ....................................................... 21-169
  Example Model File ................................................................ 21-177
LEVEL 39 BSIM2 Model ................................................................. 21-179
  LEVEL 39 Model Parameters ..................................................... 21-179
  LEVEL 39 Model Equations ....................................................... 21-186
  Geometry and Bias Adjustment of Model Parameters ................. 21-186
  Compatibility Notes ............................................................... 21-190
  Prevention of Negative Output Conductance ......................... 21-193
  Charge-based Gate Capacitance Model (CAPOP=39) .............. 21-194
  Star-Hspice Enhancements ..................................................... 21-196
  Modeling Example ............................................................... 21-200
  Typical BSIM2 Model Listing .................................................. 21-203
LEVEL 40 HP a-Si TFT Model ........................................................ 21-207
  Model Parameters ............................................................... 21-207
  Using the HP a-Si TFT Model in Star-Hspice ....................... 21-208
  LEVEL 40 Model Equations ...................................................... 21-210
  LEVEL 40 Model Topology ....................................................... 21-211
Comparing MOS Models ............................................................. 21-218
  History and Motivation .......................................................... 21-218
  Future for Model Developments ............................................. 21-220
  Model Equation Evaluation Criteria .................................... 21-221
Potential for Good Fit to Data ............................................................. 21-221
Ease of Fit to Data............................................................................... 21-222
Robustness and Convergence Properties............................................. 21-223
Behavior Follows Actual Devices In All Circuit Conditions.............. 21-224
Ability to Simulate Process Variation.................................................. 21-225
Gate Capacitance Modeling ................................................................. 21-225
Examples of Data Fitting................................................................. 21-227

Chapter 22 - Selecting MOSFET Models: LEVEL 47-62 ..................... 22-1

LEVEL 47 BSIM3 Version 2 MOS Model ............................................. 22-2

LEVEL 47 Model Parameters ................................................................. 22-2
Leff and Weff Equations for BSIM3 Version 2.0................................. 22-9
LEVEL 47 Model Equations ................................................................. 22-10
PMOS Model......................................................................................... 22-18

LEVELs 49 and 53 BSIM3v3 MOS Models ............................................. 22-19

Selecting Model Versions ..................................................................... 22-20
Version 3.2 Features.............................................................................. 22-22
Nonquasi-Static (NQS) Model.............................................................. 22-23
Star-Hspice Enhancements................................................................... 22-24
Using BSIM3v3 in Star-Hspice............................................................ 22-32
LEVEL 49, 53 Model Parameters ......................................................... 22-34
Parameter Range Limits......................................................................... 22-49
LEVEL 49, 53 Equations ...................................................................... 22-52
.MODEL CARDS NMOS Model ......................................................... 22-53
PMOS Model......................................................................................... 22-55

LEVEL 50 Philips MOS9 Model............................................................ 22-56

LEVEL 50 Model Parameters ................................................................. 22-56
JUNCAP Model Parameters................................................................. 22-61
Using the Philips MOS9 Model in Star-Hspice .................................... 22-62
Star-Hspice Model Statement............................................................... 22-63

LEVEL 54 BSIM4.0 Model..................................................................... 22-66
LEVEL 54 Model Parameters ............................................................... 22-67

LEVEL 55 EPFL-EKV MOSFET Model .................................................. 22-85

  Single Equation Model .................................................................. 22-85
  Effects Modeled ........................................................................... 22-85
  Coherence of Static and Dynamic Models ...................................... 22-86
  Bulk Reference and Symmetry ....................................................... 22-86
  Equivalent Circuit ....................................................................... 22-87
  Device Input Variables .................................................................. 22-88
  EKV Intrinsic Model Parameters .................................................... 22-88
  Static Intrinsic Model Equations .................................................... 22-92
  Quasi-static Model Equations ......................................................... 22-103
  Non-Quasi-Static (NQS) Model Equations ....................................... 22-106
  Intrinsic Noise Model Equations .................................................... 22-107
  Operating Point Information ......................................................... 22-108
  Estimation and Limits of Static Intrinsic Model Parameters .......... 22-109
  Model Updates Description ........................................................... 22-111

LEVEL 57 UC Berkeley BSIM3-SOI Model ........................................... 22-114

  LEVEL 57 Model Parameters .......................................................... 22-117
  LEVEL 57 Template Output ............................................................. 22-128
  LEVEL 57 Updates to BSIM3-SOI PD versions 2.2, 2.21, and 2.22 .. 22-131

LEVEL 58 University of Florida SOI Model ........................................... 22-133

  LEVEL 58 FD/SOI MOSFET Model Parameters .............................. 22-134
  LEVEL 58 NFD/SOI MOSFET Model Parameters ............................. 22-138
  LEVEL 58 Template Output ............................................................ 22-144

LEVEL 59 UC Berkeley BSIM3-SOI FD Model ...................................... 22-147

  LEVEL 59 Model Parameters .......................................................... 22-149
  LEVEL 59 Template Output ............................................................. 22-157

LEVEL 60 UC Berkeley BSIM3-SOI DD Model ...................................... 22-161

  Model Features ............................................................................. 22-161
  Syntax ........................................................................................... 22-162
  Level 60 Model Parameters ......................................................... 22-164
LEVEL 61 RPI a-Si TFT Model .............................................................. 22-175
  Model Features .................................................................................... 22-175
  Using LEVEL 61 with Star-Hspice ..................................................... 22-175
  LEVEL 61 Model Parameters ............................................................. 22-176
  Equivalent Circuit ............................................................................... 22-178
  Model Equations ................................................................................. 22-178
LEVEL 62 RPI Poli-Si TFT Model ........................................................... 22-182
  Model Features .................................................................................... 22-182
  Using LEVEL 62 with Star-Hspice ..................................................... 22-183
  LEVEL 62 Model Parameters ............................................................. 22-184
  Equivalent Circuit ............................................................................... 22-186
  Model Equations ................................................................................. 22-186

Chapter 23 - Using the Common Model Interface ................................. 23-1
  Understanding CMI ................................................................................ 23-2
  Examining the Directory Structure ...................................................... 23-3
  Running Simulations with CMI Models ................................................ 23-4
  Supported Platforms ............................................................................. 23-5
  Adding Proprietary MOS Models ....................................................... 23-6
    Creating a CMI Shared Library .......................................................... 23-6
  Testing CMI Models ............................................................................. 23-12
  Model Interface Routines ..................................................................... 23-13
  Interface Variables ............................................................................. 23-18
    pModel, pInstance ............................................................................. 23-19
    CMI_ResetModel .............................................................................. 23-20
    CMI_ResetInstance .......................................................................... 23-22
    CMI_AssignModelParm ................................................................... 23-22
    CMI_AssignInstanceParm .................................................................. 23-23
    CMI_SetupModel .............................................................................. 23-24
    CMI_SetupInstance .......................................................................... 23-25
<table>
<thead>
<tr>
<th>Function Calling Protocol</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMI_Evaluate</td>
<td>23-26</td>
</tr>
<tr>
<td>CMI_DiodeEval</td>
<td>23-27</td>
</tr>
<tr>
<td>CMI_Noise</td>
<td>23-29</td>
</tr>
<tr>
<td>CMI_PrintModel</td>
<td>23-31</td>
</tr>
<tr>
<td>CMI_FreeModel</td>
<td>23-32</td>
</tr>
<tr>
<td>CMI_FreeInstance</td>
<td>23-32</td>
</tr>
<tr>
<td>CMI_WriteError</td>
<td>23-33</td>
</tr>
<tr>
<td>CMI_Start</td>
<td>23-35</td>
</tr>
<tr>
<td>CMI_Conclude</td>
<td>23-35</td>
</tr>
<tr>
<td>CMI Function Calling Protocol</td>
<td>23-36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Internal Routines</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supporting Extended Topology</td>
<td>23-40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conventions</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bias Polarity Conventions for N- and P-channel Devices</td>
<td>23-42</td>
</tr>
<tr>
<td>Source-Drain Reversal Conventions</td>
<td>23-43</td>
</tr>
<tr>
<td>Thread-Safe Model Code</td>
<td>23-44</td>
</tr>
</tbody>
</table>

**Chapter 24 - Performing Cell Characterization**  

Determine Typical Data Sheet Parameters  
Rise, Fall, and Delay Calculations  
Ripple Calculation  
Sigma Sweep versus Delay  
Delay versus Fanout  
Pin Capacitance Measurement  
Op-amp Characterization of ALM124  
Cell Characterization Using Data Driven Analysis  

**Chapter 25 - Signal Integrity**  

Preparing for Simulation  
Signal Integrity Problems  
Analog Side of Digital Logic
Chapter 26 - Performing Behavioral Modeling ........................................  26-1

Understanding the Behavioral Design Process .........................................  26-2
Using Behavioral Elements ........................................................................  26-3
Using Voltage and Current Controlled Elements ........................................  26-6
  Polynomial Functions ..................................................................................  26-7
  Piecewise Linear (PWL) Function ..............................................................  26-10
Dependent Voltage Sources — E Elements ...............................................  26-11
  Voltage Controlled Voltage Source (VCVS) ..............................................  26-11
  Behavioral Voltage Source .......................................................................  26-11
  Ideal Op-Amp .........................................................................................  26-11
  Ideal Transformer ....................................................................................  26-12
Dependent Current Sources — G Elements ...............................................  26-16
  Voltage Controlled Current Source (VCCS) .............................................  26-16
  Behavioral Current Source ......................................................................  26-16
  Voltage Controlled Resistor (VCR) .........................................................  26-16
  Voltage Controlled Capacitor (VCCAP) .................................................  26-17
Dependent Voltage Sources — H Elements ..............................................  26-23
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Controlled Voltage Source (CCVS)</td>
<td>26-23</td>
</tr>
<tr>
<td>Current Dependent Current Sources — F Elements</td>
<td>26-27</td>
</tr>
<tr>
<td>Current Controlled Current Source (CCCS)</td>
<td>26-27</td>
</tr>
<tr>
<td>Modeling with Digital Behavioral Components</td>
<td>26-31</td>
</tr>
<tr>
<td>Behavioral AND and NAND Gates</td>
<td>26-31</td>
</tr>
<tr>
<td>Behavioral D-Latch</td>
<td>26-32</td>
</tr>
<tr>
<td>Behavioral Double-Edge Triggered Flip-Flop</td>
<td>26-36</td>
</tr>
<tr>
<td>Calibrating Digital Behavioral Components</td>
<td>26-40</td>
</tr>
<tr>
<td>Building Behavioral Lookup Tables</td>
<td>26-40</td>
</tr>
<tr>
<td>Optimizing Behavioral CMOS Inverter Performance</td>
<td>26-46</td>
</tr>
<tr>
<td>Optimizing Behavioral Ring Oscillator Performance</td>
<td>26-50</td>
</tr>
<tr>
<td>Using Analog Behavioral Elements</td>
<td>26-53</td>
</tr>
<tr>
<td>Behavioral Integrator</td>
<td>26-53</td>
</tr>
<tr>
<td>Behavioral Differentiator</td>
<td>26-55</td>
</tr>
<tr>
<td>Ideal Transformer</td>
<td>26-57</td>
</tr>
<tr>
<td>Behavioral Tunnel Diode</td>
<td>26-58</td>
</tr>
<tr>
<td>Behavioral Silicon Controlled Rectifier (SCR)</td>
<td>26-59</td>
</tr>
<tr>
<td>Behavioral Triode Vacuum Tube Subcircuit</td>
<td>26-60</td>
</tr>
<tr>
<td>Behavioral Amplitude Modulator</td>
<td>26-62</td>
</tr>
<tr>
<td>Behavioral Data Sampler</td>
<td>26-63</td>
</tr>
<tr>
<td>Using Op-Amps, Comparators, and Oscillators</td>
<td>26-65</td>
</tr>
<tr>
<td>Star-Hspice Op-Amp Model Generator</td>
<td>26-65</td>
</tr>
<tr>
<td>Op-Amp Element Statement Format</td>
<td>26-66</td>
</tr>
<tr>
<td>Op-Amp .MODEL Statement Format</td>
<td>26-66</td>
</tr>
<tr>
<td>Op-Amp Subcircuit Example</td>
<td>26-74</td>
</tr>
<tr>
<td>741 Op-Amp from Controlled Sources</td>
<td>26-76</td>
</tr>
<tr>
<td>Inverting Comparator with Hysteresi</td>
<td>26-79</td>
</tr>
<tr>
<td>Voltage Controlled Oscillator (VCO)</td>
<td>26-81</td>
</tr>
<tr>
<td>LC Oscillator</td>
<td>26-83</td>
</tr>
<tr>
<td>Using a Phase Locked Loop Design</td>
<td>26-88</td>
</tr>
</tbody>
</table>
Phase Detector Using Multi-Input NAND Gates ................................. 26-88
PLL BJT Behavioral Modeling ......................................................... 26-92
References .................................................................................... 26-100

Chapter 27 - Performing Pole/Zero Analysis ................................. 27-1
Understanding Pole/Zero Analysis .................................................. 27-2
Using Pole/Zero Analysis ............................................................... 27-3
.PZ (Pole/Zero) Statement ............................................................... 27-3
Pole/Zero Analysis Examples ......................................................... 27-5

Chapter 28 - Performing FFT Spectrum Analysis ......................... 28-1
Using Windows In FFT Analysis ..................................................... 28-3
Using the .FFT Statement .............................................................. 28-7
Examining the FFT Output ............................................................. 28-10
AM Modulation .............................................................................. 28-13
Balanced Modulator and Demodulator ......................................... 28-16
Signal Detection Test Circuit ......................................................... 28-24
References .................................................................................... 28-29

Chapter 29 - Modeling Filters and Networks ............................... 29-1
Understanding Transient Modeling ................................................ 29-2
Using G and E Elements ............................................................... 29-4
Laplace Transform Function Call .................................................. 29-4
Element Statement Parameters ..................................................... 29-9
Laplace Band-Reject Filter ........................................................... 29-12
Laplace Low-Pass Filter ............................................................... 29-14
Circular Convolution Example ....................................................... 29-17
Laplace and Pole-Zero Modeling ................................................... 29-20
Laplace Transform (LAPLACE) Function ...................................... 29-20

Laplace Transform POLE (Pole/Zero) Function ................................. 29-28
AWE Transfer Function Modeling .................................................. 29-35
Y Parameter Line Modeling ........................................................... 29-39
Comparison of Circuit and Pole/Zero Models ......................... 29-43
Modeling Switched Capacitor Filters ........................................... 29-48
Switched Capacitor Network ......................................................... 29-48
Switched Capacitor Filter Example .............................................. 29-50
References .................................................................................... 29-55

Chapter 30 - Timing Analysis Using Bisection................................. 30-1
Understanding Bisection ................................................................. 30-2
Understanding the Bisection Methodology ................................. 30-5
Measurement .............................................................................. 30-5
Optimization .............................................................................. 30-5
Using Bisection ............................................................................. 30-6
Examining the Command Syntax .................................................. 30-7
Setup Time Analysis ................................................................. 30-10
Minimum Pulse Width Analysis .................................................. 30-16

Chapter 31 - Running Demonstration Files ................................... 31-1
Using the Demo Directory Tree ....................................................... 31-2
Running the Two-Bit Adder Demo ................................................. 31-3
Running the MOS I-V and C-V Plotting Demo ......................... 31-7
Running the CMOS Output Driver Demo ...................................... 31-12
Running the Temperature Coefficients Demo ......................... 31-18
Simulating Electrical Measurements .......................................... 31-20
Modeling Wide-Channel MOS Transistors ............................. 31-23
Examining the Demonstration Input Files ................................. 31-26
Appendix A - FAQ/Troubleshooting .............................................................. A-1
Analysis ........................................................................................................ A-2
Documentation ............................................................................................ A-4
Environment Variables .............................................................................. A-6
Error Messages ............................................................................................ A-7
Input ............................................................................................................ A-12
Installation Issues ...................................................................................... A-13
Licensing/Access Issues ........................................................................... A-15
Limitations .................................................................................................. A-17
Miscellaneous ............................................................................................ A-18
Models ........................................................................................................ A-20
MS Windows/PC Issues ............................................................................ A-23
Netlist/Options ............................................................................................ A-27
Output .......................................................................................................... A-32
W Element/Field Solver ........................................................................... A-34
Waveform Viewing .................................................................................. A-35

Appendix B - Interfaces For Design Environments ..................................... B-1
AvanLink to Cadence Composer and Analog Artist .................................. B-2
Features ...................................................................................................... B-2
Environment ............................................................................................... B-3
AvanLink Design Flow ............................................................................... B-4
Schematic Entry and Library Operations ................................................ B-5
Netlist Generation ..................................................................................... B-6
Simulation .................................................................................................. B-7
Waveform Display .................................................................................... B-7
AvanLink for Design Architect ................................................................. B-8
Features ...................................................................................................... B-8
Attenuation in Transmission Lines ................................................. C-88
The Lossy Transmission Line Model............................................. C-91
References .................................................................................... C-95

Appendix D - Finding Device Libraries .......................................... D-1
Selecting Models Automatically ...................................................... D-2
Examining the Library Listings ....................................................... D-5
  Analog Device Models ................................................................. D-5
  Behavioral Device Models .......................................................... D-8
  Bipolar Transistor Models ......................................................... D-9
  Burr-Brown Devices ................................................................. D-9
  Comlinear Device Models ......................................................... D-10
  Diode Models ............................................................................ D-10
  FET Models ............................................................................... D-12
  Linear Technology Device Models ............................................ D-14
  Intel PCI Speedway Models ...................................................... D-15
  Signetics Device Models ........................................................... D-15
  Texas Instruments Device Models ............................................. D-16
  Transmission Line Models ......................................................... D-17
  Xilinx Device Models .............................................................. D-17

Appendix E - Performing Library Encryption ................................ E-1
Understanding Library Encryption .............................................. E-2
  Controlling the Encryption Process ........................................ E-2
  Library Structure ..................................................................... E-2
Knowing the Encryption Guidelines ............................................ E-5
Installing and Running the Encryptor .......................................... E-7
  Installing the Encryptor .......................................................... E-7
  Running the Encryptor ........................................................... E-7
Understanding Metaencrypt Features ......................................... E-9
  New 8-Byte Key Encryption ................................................... E-9
Appendix F - Full Simulation Examples

Full Simulation Example with AvanWaves

Input Netlist and Circuit
Execution and Output Files
Simulation Graphical Output in AvanWaves

Full Simulation Example with Cosmos-Scope

Input Netlist and Circuit
Execution and Output Files
Using Cosmos-Scope to View Star-Hspice Results

Index — All Volumes
Chapter 1

Introducing Star-Hspice

The Star-Hspice optimizing analog circuit simulator is Avant!’s industrial-grade circuit analysis product for the simulation of electrical circuits in steady-state, transient, and frequency domains. Circuits are accurately simulated, analyzed, and optimized from DC to microwave frequencies greater than 100 GHz.

Star-Hspice is ideal for cell design and process modeling and is the tool of choice for signal-integrity and transmission-line analysis.

This chapter covers the following topics:

- Star-Hspice Applications
- Star-Hspice Features
- Star-Hspice Platforms
- Examining the Simulation Structure
- Understanding the Data Flow
**Star-Hspice Applications**

Star-Hspice is unequalled for fast, accurate circuit and behavioral simulation. It facilitates circuit-level analysis of performance and yield utilizing Monte Carlo, worst case, parametric sweep, and data-table sweep analysis while employing the most reliable automatic convergence capability. Star-Hspice forms the cornerstone of a suite of Avant! tools and services that allow accurate calibration of logic and circuit model libraries to actual silicon performance.

The size of the circuits simulated by Star-Hspice is limited only by the virtual memory of the computer being used. Star-Hspice software is optimized for each computer platform with interfaces available to a variety of design frameworks.
Star-Hspice is compatible with most SPICE variations, and has the following additional features:

- Superior convergence
- Accurate modeling, including many foundry models
- Hierarchical node naming and reference
- Circuit optimization for models and cells, with incremental or simultaneous multiparameter optimizations in AC, DC, and transient simulations
- Interpreted Monte Carlo and worst-case design support
- Input, output, and behavioral algebraics for parameterizable cells
- Cell characterization tools for calibrating library models for higher-level logic simulators

Figure 1-1: Star-Hspice Design Features
Star-Hspice Features

- Geometric lossy coupled transmission lines for PCB, multi-chip, package, and IC technologies
- Discrete component, pin, package, and vendor IC libraries
- AvanWaves interactive waveform graphing and analysis from multiple simulations

**Figure 1-2: Star-Hspice Circuit Analysis Types**

![Diagram of Star-Hspice Circuit Analysis Types]

- Parametric
- Operating Point
- Pole-Zero
- Monte Carlo
- Frequency
- Transient
- S-parameter
- Optimization
- Mixed
- AC/Transient
- Data Driven
Simulation at the integrated circuit level and at the system level requires careful planning of the organization and interaction between transistor models and subcircuits. Methods that worked for small circuits might have too many limitations when applied to higher-level simulations.

You can organize simulation circuits and models to run using the following Star-Hspice features:

- Explicit include files – .INC statement
- Implicit include files – .OPTION SEARCH = ‘lib_directory’
- Algebraics and parameters for devices and models – .PARAM statement
- Parameter library files – .LIB statement
- Automatic model selector – LMIN, LMAX, WMIN, WMAX model parameters
- Parameter sweep – SWEEP analysis statement
- Statistical analysis – SWEEP MONTE analysis statement
- Multiple alternative – .ALTER statement
- Automatic measurements – .MEASURE statement
Star-Hspice Platforms

Star-Hspice is available for the following platforms and operating systems:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Sparc and Ultra</td>
<td>Solaris 5.5, 5.6, 5.7 and 5.8</td>
</tr>
<tr>
<td>HP PA</td>
<td>UX 10.20 +</td>
</tr>
<tr>
<td>IBM RS6000</td>
<td>AIX 4.1 +</td>
</tr>
<tr>
<td>DEC Alpha</td>
<td>OSF 4.0 +</td>
</tr>
<tr>
<td>SGI</td>
<td>IRIX 6.5</td>
</tr>
<tr>
<td>PC</td>
<td>Windows 95, 98, 2000 and NT 4.0</td>
</tr>
<tr>
<td>Linux</td>
<td>RedHat 7.0 (Does not support MOSFET level</td>
</tr>
<tr>
<td></td>
<td>29 and level 45).</td>
</tr>
</tbody>
</table>

**Note:** + means “and higher versions”
Examining the Simulation Structure

Figure 1-4 shows the program structure for simulation experiments.

Figure 1-4: Simulation Program Structure

Analysis and verification of complex designs are typically organized around a series of experiments. These experiments are simple sweeps or more complex Monte Carlo, optimization, and setup and hold violation analyses that analyze DC, AC, and transient conditions.

For each simulation experiment, tolerances and limits must be specified to achieve the desired goals, such as optimizing or centering a design. Common factors for each experiment are process, voltage, temperature, and parasitics.
Two terms are used to describe experimental methods using Star-Hspice:

- **Single point** – a single point experiment is a simple procedure that produces a single result, or a single set of output data.
- **Multipoint** – an analysis (single point) sweep is performed for each value in an outer loop (multipoint) sweep.

The following are examples of multipoint experiments:

- Process variation – Monte Carlo or worst case model parameter variation
- Element variation – Monte Carlo or element parameter sweeps
- Voltage variation – VCC, VDD, and substrate supply variation
- Temperature variation – design temperature sensitivity
- Timing analysis – basic timing, jitter, and signal integrity analysis
- Parameter optimization – balancing complex constraints such as speed versus power or frequency versus slew rate versus offset for analog circuits
Understanding the Data Flow

Star-Hspice accepts input and simulation control information from a number of different sources. It can output results in a number of convenient forms for review and analysis. The overall Star-Hspice data flow is shown in Figure 1-5.

To begin the design entry and simulation process, create an input netlist file. Most schematic editors and netlisters support the SPICE or Star-Hspice hierarchical format. The analyses specified in the input file are executed during the Star-Hspice run. Star-Hspice stores the simulation results requested in either an output listing file or, if .OPTIONS POST is specified, a graph data file. If POST is specified, the complete circuit solution (in either steady state, time, or frequency domain) is stored. The results for any nodal voltage or branch current can then be viewed or plotted using a high-resolution graphic output terminal or laser printer. Star-Hspice has a complete set of print and plot variables for viewing analysis results.

The Star-Hspice program has a textual command line interface. For example, the program is executed by entering the hspice command, the input file name, and the desired options at the prompt in a UNIX shell, on a DOS command line, or by clicking on an icon in a Windows environment. You can have the Star-Hspice program simulation output appear in either an output listing file or in a graph data file. Star-Hspice creates standard output files to describe initial conditions (.ic extension) and output status (.st0 extension). In addition, Star-Hspice creates various output files in response to user-defined input options—for example, a <design>.tr0 file in response to a .TRAN transient analysis statement.

The AvanWaves output display and analysis program has a graphical user interface. Execute AvanWaves operations using the mouse to select options and execute commands in various AvanWaves windows. Refer to the AvanWaves User Guide for instructions on using AvanWaves.
Simulation Process Overview

Figure 1-6 is a diagram of the Star-Hspice simulation process. The following section summarizes the steps in a typical simulation.
Perform these steps to execute a Star-Hspice simulation.
1. **Invocation**
   Invoke Star-Hspice with a UNIX command such as:

   
   ```
   hspice demo.sp > demo.out &
   ```

   The Star-Hspice shell is invoked, with an input netlist file `demo.sp` and an output listing file `demo.out`. The “&” at the end of the command invokes Star-Hspice in the background so that the window and keyboard can still be used while Star-Hspice runs.

2. **Script execution**
   The Star-Hspice shell starts the `hspice` executable from the appropriate architecture (machine type) directory. The UNIX run script launches a Star-Hspice simulation. This procedure is used to establish the environment for the Star-Hspice executable. The script prompts for information, such as the platform you are running on and the version of Star-Hspice you want to run. (Available versions are determined when Star-Hspice is installed.)

3. **Licensing**
   Star-Hspice supports the FLEXlm licensing management system. With FLEXlm licensing, Star-Hspice reads the environment variable `LM_LICENSE_FILE` for the location of the `license.dat` file.

   If there is an authorization failure, the job terminates at this point, printing an error message in the output listing file.

4. **Simulation configuration**
   Star-Hspice reads the appropriate `meta.cfg` file. The search order for the configuration file is the user login directory and then the product installation directory.

5. **Design input**
   Star-Hspice opens the input netlist file. If the input netlist file does not exist, a “no input data” error appears in the output listing file.

   Three scratch files are opened in the `/tmp` directory. You can change this directory by resetting the TMPDIR environment variable in the Star-Hspice command script.
Star-Hspice opens the output listing file. If you do not have ownership of the
current directory, Star-Hspice terminates with a “file open” error.

An example of a simple Star-Hspice input netlist is:

Inverter Circuit

```
.OPTIONS LIST NODE POST
.TRAN 200P 20N SWEEP TEMP -55 75 10
.PRINT TRAN V(IN) V(OUT)
M1 VCC IN OUT VCC PCH L = 1U W = 20U
M2 OUT IN 0 0 NCH L = 1U W = 20U
VCC VCC 0 5
VIN IN 0 0 PULSE .2 4.8 2N 1N 1N 5N 20N CLOAD OUT 0 .75P
.MODEL PCH PMOS
.MODEL NCH NMOS
.ALTER
CLOAD OUT 0 1.5P
.END
```

6. **Library input**

Star-Hspice reads any files specified in .INCLUDE and .LIB statements.

7. **Operating point initialization**

Star-Hspice reads any initial conditions specified in .IC and .NODESET
statements, finds an operating point (that can be saved with a .SAVE
statement), and writes any operating point information you requested.

8. **Multipoint analysis**

Star-Hspice performs the experiments specified in analysis statements. In
the above example, the .TRAN statement causes Star-Hspice to perform a
multipoint transient analysis for 20 ns for temperatures ranging from -55°C
to 75°C in steps of 10°C.

9. **Single-point analysis**

Star-Hspice performs a single or double sweep of the designated quantity
and produces one set of output files.
10. Worst case .ALTER

Simulation conditions may be varied and the specified single or multipoint analysis repeated. In the above example, CLOAD is changed from 0.75 pF to 1.5 pF, and the multipoint transient analysis is repeated.

11. Normal termination

After completing the simulation, Star-Hspice closes all files that it opened and releases all license tokens.
Chapter 2

Getting Started

The examples in this chapter show you how to run Star-Hspice to perform some simple analyses.

This chapter includes the following examples:

- AC Analysis of an RC Network
- Transient Analysis of an RC Network
- Transient Analysis of an Inverter
AC Analysis of an RC Network

Figure 2-1 shows a simple RC network with a DC and AC source applied. The circuit consists of two resistors, R1 and R2, capacitor C1, and the source V1. Node 1 is the connection between the source positive terminal and R1. Node 2 is where R1, R2, and C1 are connected. Star-Hspice ground is always node 0.

The Star-Hspice netlist for the RC network circuit is:

A SIMPLE AC RUN
.OPTIONS LIST NODE POST
.OP
.AC DEC 10 1K 1MEG
.PRINT AC V(1) V(2) I(R2) I(C1)
V1 1 0 10 AC 1
R1 1 2 1K
R2 2 0 1K
C1 2 0 .001U
.END

Follow the procedure below to perform an AC analysis for the RC network circuit.
1. Type the above netlist into a file named quickAC.sp.

2. Run a Star-Hspice analysis by typing

   hspice quickAC.sp > quickAC.lis

   When the run finishes Star-Hspice displays

   >info:   ***** hspice job concluded

   followed by a line that shows the amount of real time, user time, and system
   time needed for the analysis.

   The following new files are present in your run directory:

   quickAC.ac0
   quickAC.ie
   quickAC.lis
   quickAC.st0.

3. Use an editor to view the .lis and .st0 files to examine the simulation results
   and status.

4. Run AvanWaves and open the .sp file. Select the quickAC.ac0 file from the
   Results Browser window to view the waveform. Display the voltage at
   node 2, using a log scale on the x-axis.

   Figure 2-2 shows the waveform that was produced by sweeping the response of
   node 2 as the frequency of the input was varied from 1 kHz to 1 MHz.

   **Figure 2-2: RC Network Node 2 Frequency Response**

![Waveform](image.png)
The file *quickAC.lis* displays the input netlist, details about the elements and topology, operating point information, and the table of requested data as the input is swept from 1 kHz to 1 MHz. The files *quickAC.ic* and *quickAC.st0* contain information about the DC operating point conditions and the Star-Hspice run status, respectively. The operating point conditions can be used for subsequent simulation runs using the .LOAD statement.
Transient Analysis of an RC Network

As a second example, run a transient analysis using the same RC network as in Figure 2-1, but adding a pulse source to the DC and AC sources.

1. Type the following equivalent Star-Hspice netlist into a file named quickTRAN.sp.

   A SIMPLE TRANSIENT RUN
   .OPTIONS LIST NODE POST
   .OP
   .TRAN 10N 2U
   .PRINT TRAN V(1) V(2) I(R2) I(C1)
   V1 1 0 10 AC 1 PULSE 0 5 10N 20N 20N 500N 2U
   R1 1 2 1K
   R2 2 0 1K
   C1 2 0 .001U
   .END

   Note that the V1 source specification has added a pulse source. The syntax for pulse sources and other types of sources is described in “Using Sources and Stimuli” on page 5-1.

2. Type the following to run Star-Hspice.

   hspice quickTRAN.sp > quickTRAN.lis

3. Use an editor to view the .lis and .st0 files to examine the simulation results and status.

4. Run AvanWaves and open the .sp file. Select the quickTRAN.tr0 file from the Results Browser window to view the waveform. Display the voltage at nodes 1 and 2 on the x-axis.

   The waveforms are shown in Figure 2-3.
Figure 2-3: Voltages at RC Network Circuit Node 1 and Node 2
Transient Analysis of an Inverter

As a final example, analyze the behavior of the simple MOS inverter shown in Figure 2-4.

**Figure 2-4: MOS Inverter Circuit**

1. Type the following netlist data into a file named `quickINV.sp`.
   
   ```
   Inverter Circuit
   .OPTIONS LIST NODE POST
   .TRAN 200P 20N
   .PRINT TRAN V(IN) V(OUT)
   M1 OUT IN VCC VCC PCH L = 1U W = 20U
   M2 OUT IN 0 0 NCH L = 1U W = 20U
   VCC VCC 0 5
   VIN IN 0 0 PULSE .2 4.8 2N 1N 1N 5N 20N
   CLOAD OUT 0 .75P
   .MODEL PCH PMOS LEVEL = 1
   .MODEL NCH NMOS LEVEL = 1
   .END
   ```

2. Type the following to run Star-Hspice.
   
   ```
   hspice quickINV.sp > quickINV.lis
   ```
Use AvanWaves to examine the voltage waveforms at the inverter IN and OUT nodes. The waveforms are shown in Figure 2-5.

**Figure 2-5: Voltage at MOS Inverter Node 1 and Node 2**
Chapter 3

Specifying Simulation Input and Controls

This chapter describes the input requirements, methods of entering data, and Star-Hspice statements used to enter input. This chapter covers the following topics:

- Using Netlist Input Files
- Input Netlist File Composition
- Using Subcircuits
- Discrete Device Libraries
- Using Standard Input Files
- Output Files
- Using the Star-Hspice Command
- Improving Simulation Performance Using Multithreading
Using Netlist Input Files

This section describes how to use standard Star-Hspice netlist input files.

Input Netlist File (<design>.sp) Guidelines

Star-Hspice operates on an input netlist file and stores results in either an output listing file or a graph data file. The Star-Hspice input file, with the name <design>.sp (although the filename can be anything, we recommend this form for clarity), contains the following:

- Design netlist (with subcircuits and macros, power supplies, and so on)
- Statement naming the library to be used (optional)
- Specification of the analysis to be run (optional)
- Specification of the output desired (optional)

Input netlist and library input files are generated by a schematic netlister or with a text editor.

Statements in the input netlist file can be in any order, except that the first line is a title line, and the last .ALTER submodule must appear at the end of the file before the .END statement.

Note: If there is no .END statement at the end of the input netlist file or no carriage return after the .END statement, an error message is issued.

Input Line Format

- The input netlist file cannot be in a packed or compressed format.
- The Star-Hspice input reader can accept an input token, such as a statement name, a node name, or a parameter name or value. A valid string of characters between two token delimiters is accepted as a token. See “Delimiters” on page 3-4.
- Input filename length, statement length, and equation length can be up to 1024 characters.
Upper and lower case are ignored, except in quoted filenames.

A statement may be continued on the next line by entering a plus (+) sign as the first nonnumeric, nonblank character in the next line.

All statements, including quoted strings such as paths and algebraics, are continued with a backslash (\) or a double backslash (\\) at the end of the line to be continued. The single backslash preserves white space and the double backslash squeezes out any white space between the continued lines. The double backslash guarantees that path names are joined without interruption. Input lines can be 1024 characters long, so folding and continuing a line is generally only necessary to improve readability.

Comments are added at any place in the file. Lines beginning with an asterisk (*) are comments. Place a comment on the same line as input text by entering a dollar sign ($), preceded by one or more blanks, after the input text.

An error is issued when a special control character is encountered in the input netlist file. Since most systems cannot print special control characters, the error message is ambiguous because the erroneous character cannot be shown in the error message. Use the .OPTIONS BADCHAR statement to locate such errors. The default for BADCHAR is “off”.

Names

Names must begin with an alphabetic character, but thereafter can contain numbers and the following characters:

! # $ % * + - / < > [ ] _

Names are input tokens that must be preceded and followed by token delimiters. See “Delimiters” below.

Names can be 1024 characters long.

Names are not case sensitive.
Delimiters
- An input token is any item in the input file recognized by Star-Hspice. Input token delimiters are: tab, blank, comma, equal sign (=), and parentheses “( )”.
- Single or double quotes delimit expressions and filenames.
- Element attributes are delimited by colons (“M1:beta”, for example).
- Hierarchy is indicated by periods. For example, “X1.A1.V” is the V node on subcircuit A1 of circuit X1.

Nodes
- Node identifiers can be up to 1024 characters long, including periods and extensions.
- Numerical node names are valid in the range of 0 through 9999999999999999 (1-1E16).
- Leading zeros are ignored in node numbers.
- Trailing characters are ignored in node numbers. For example, node 1A is the same as node 1. Exception: Star-Hspice recognizes the following special alphabetic trailing characters (d, e, f, g, i, m, n, o, p, u, x, k, t).
- A node name can begin with any of the following characters: # _ ! %.
- Nodes are made global across all subcircuits by a .GLOBAL statement.
- Node 0, GND, GND!, and GROUND all refer to the global Star-Hspice ground.

Instance Names
- The names of element instances begin with the element key letter (for example, M for a MOSFET element, D for a diode, R for a resistor, and so on), except in subcircuits.
- Subcircuit instance names begin with “X”. (Subcircuits are sometimes called macros or modules.)
- Instance names are limited to 1024 characters.
- .OPTIONS LENNAM controls the length of names in Star-Hspice printouts (default = 8).
Hierarchy Paths
- Path hierarchy is indicated by a period.
- Paths can be up to 1024 characters long.
- Path numbers compress the hierarchy for post-processing and listing files.
- Path number cross references are found in the listing and in the `<design>.pa0` file.
- `.OPTIONS PATHNUM` controls whether full path names or path numbers are shown in list files.

Numbers
- Numbers are entered as integer or real.
- Numbers can use exponential format or engineering key letter format, but not both (1e-12 or 1p, but not 1e-6u).
- Exponents are designated by D or E.
- Exponent size is limited by `.OPTIONS EXPMAX`.
- Trailing alphabetic characters are interpreted as units comments.
- Units comments are not checked.
- `.OPTIONS INGOLD` controls the format of numbers in printouts.
- `.OPTIONS NUMDGT = x` controls the listing printout accuracy.
- `.OPTIONS MEASDGT = x` controls the measure file printout accuracy.
- `.OPTIONS VFLOOR = x` specifies the smallest voltage for which the value will be printed. Smaller voltages are printed as 0.

Parameters and Expressions
- Parameter names follow Star-Hspice name syntax rules.
- Parameter hierarchy overrides and defaults are defined by `.OPTIONS PARHIER = global | local`.
- The last parameter definition or `.OPTIONS` statement is used if multiple definitions exist. This is true even if the last definition or `.OPTIONS` statement is later in the input than a reference to the parameter or option. No warning is issued when a redefinition occurs.
- If a parameter is used in another parameter definition, the first parameter must be defined before the second parameter definition.
- In your design parameter name selection, be careful to avoid conflicts with parameterized libraries.
- Expressions are delimited by single or double quotes and are limited to 256 characters.
- A line can be continued to improve readability by using a double slash at end of the line (//).
- Function nesting is limited to three levels.
- No user-defined function may have more than two arguments.
- Use the PAR(expression or parameter) function to evaluate expressions in output statements.

**Input Netlist File Structure**

A Star-Hspice input netlist file should consist of one main program and one or more optional submodules. Use a submodule (preceded by an .ALTER statement) to automatically change an input netlist file and rerun the simulation with different options, netlist, analysis statements, and test vectors.

You can use several high-level call statements to restructure the input netlist file modules. These are the .INCLUDE, .LIB and .DEL LIB statements. These statements can call netlists, model parameters, test vectors, analysis, and option macros into a file from library files or other files. The input netlist file also can call an external data file that contains parameterized data for element sources and models.

**Schematic Netlists**

Star-Hspice circuits typically are generated from schematics by netlisters. Star-Hspice accepts either hierarchical or flat netlists. The normal SPICE netlisters flatten out all subcircuits and rename all nodes to numbers. Avoid flat netlists if possible.

The process of creating a schematic involves:
- Symbol creation with a symbol editor
- Circuit encapsulation
- Property creation
- Symbol placement
- Symbol property definition
- Wire routing and definition

**Input Netlist File Sections and Chapter References**

<table>
<thead>
<tr>
<th>Sections</th>
<th>Examples</th>
<th>Ch</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>.TITLE</td>
<td>3</td>
<td>The first line is the input netlist file title</td>
</tr>
<tr>
<td>Set-up</td>
<td>.OPTIONS</td>
<td>9</td>
<td>Sets conditions for simulation</td>
</tr>
<tr>
<td></td>
<td>.IC or .NODESET</td>
<td>10</td>
<td>Initial values in circuit and subcircuit</td>
</tr>
<tr>
<td></td>
<td>.PARAM</td>
<td>7</td>
<td>Set parameter values in the netlist</td>
</tr>
<tr>
<td></td>
<td>.GLOBAL</td>
<td>7</td>
<td>Set node name globally in netlist</td>
</tr>
<tr>
<td>Sources</td>
<td>Sources (I or V) and digital inputs</td>
<td>5</td>
<td>Sets input stimuli</td>
</tr>
<tr>
<td>Netlist</td>
<td>Circuit elements</td>
<td>3-4</td>
<td>Circuit for simulation</td>
</tr>
<tr>
<td></td>
<td>.SUBKCT, .ENDS</td>
<td>3</td>
<td>Subcircuit definitions</td>
</tr>
<tr>
<td>Analysis</td>
<td>.DC, .TRAN, .AC, etc.</td>
<td>10-12</td>
<td>Statements to perform analyses</td>
</tr>
<tr>
<td></td>
<td>.SAVE and .LOAD</td>
<td>10</td>
<td>Save and load operating point info</td>
</tr>
<tr>
<td></td>
<td>.DATA</td>
<td>3</td>
<td>Create table for data-driven analysis</td>
</tr>
<tr>
<td></td>
<td>.TEMP</td>
<td>3</td>
<td>Set analysis temperature</td>
</tr>
<tr>
<td>Output</td>
<td>.PRINT, .PLOT, .GRAPH, .PROBE</td>
<td>8</td>
<td>Statements to output variables</td>
</tr>
<tr>
<td></td>
<td>.MEASURE</td>
<td>8</td>
<td>Statement to evaluate and report user-defined functions of a circuit</td>
</tr>
<tr>
<td>Sections</td>
<td>Examples</td>
<td>Ch</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------</td>
<td>----</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Library, Model and File Inclusion</td>
<td>.INCLUDE</td>
<td>3</td>
<td>General include files</td>
</tr>
<tr>
<td></td>
<td>.MODEL</td>
<td>3, 8</td>
<td>Element model descriptions</td>
</tr>
<tr>
<td></td>
<td>.LIB</td>
<td>3</td>
<td>Library</td>
</tr>
<tr>
<td></td>
<td>.&lt;UN&gt;PROTECT</td>
<td>3</td>
<td>Control printback to output listing</td>
</tr>
<tr>
<td>Alter blocks</td>
<td>.ALTER</td>
<td>3</td>
<td>Sequence for in-line case analysis</td>
</tr>
<tr>
<td></td>
<td>.DELETE LIB</td>
<td>3</td>
<td>Removes previous library selection</td>
</tr>
<tr>
<td>End of netlist</td>
<td>.END</td>
<td>3</td>
<td>Required statement to end the netlist</td>
</tr>
</tbody>
</table>
Input Netlist File Composition

Title of Simulation and .TITLE Statement

The simulation title is set using the first line of the input file. This line is always read and used as the title of the simulation regardless of the contents of this line. The title is printed verbatim in each section heading of the output listing file of the simulation.

The .TITLE statement, as shown in the first syntax below, can be used on the first line of the netlist to set the title, although the .TITLE syntax is not necessary. In the second form shown below, the string is the first line of the input file. The first line of the input file is always the implicit title. If a Star-Hspice statement appears as the first line in a file, it is interpreted as a title and is not executed.

An .ALTER statement does not support the usage of .TITLE. To change a title for a .ALTER statement, place the title content in the .ALTER statement itself.

Syntax

```
.TITLE <string of up to 72 characters>
```

or

```
<string of up to 72 characters>
```

Comments

An asterisk (*) as the first nonblank character or an inline dollar sign ($) indicates a comment statement.

Syntax

```
* <comment on a line by itself>
```

or

```
<HSPICE statement> $ <comment following HSPICE input>
```
Example

*RF = 1K   GAIN SHOULD BE 100
$ MAY THE FORCE BE WITH MY CIRCUIT
VIN 1 0 PL 0 0 5V 5NS $ 10v 50ns
R12 1 0 1MEG $ FEED BACK

You can place comment statements anywhere in the circuit description.

The * must be in the first space on the line.

The $ must be used for comments that do not begin at the first space on a line
(for example, for comments that follow Star-Hspice input on the same line). The
$ must be preceded by a space or comma if it is not the first nonblank character.
The $ is allowed within node or element names.

Element and Source Statements

Element statements describe the netlists of devices and sources. Elements are
connected to one another by nodes, which can either be numbers or names.
Element statements specify

■ Type of device
■ Nodes to which the device is connected
■ Parameter values that describe the operating electrical characteristics of the
device

Element statements also can reference model statements that define the electrical
parameters of the element.

Element statements for the various types of Star-Hspice elements are described
in the chapters on those types of elements.

Syntax

elname <node1 node2 ... nodeN> <mname>  
+ <pname1 = val1> <pname2 = val2> <M = val>

or

elname <node1 node2 ... nodeN> <mname>  
+ <pname = ’expression’> <M = val>
or

\texttt{elname <node1 node2 ... nodeN> <mname>}
+ \texttt{<val1 val2 ... valn>}

where:

- \texttt{elname} Element name that cannot exceed 1023 characters, and must begin with a specific letter for each element type:
  - B: IBIS buffer
  - C: Capacitor
  - D: Diode
  - E,F,G,H: Dependent current and voltage sources
  - I: Current source
  - J: JFET or MESFET
  - K: Mutual inductor
  - L: Inductor
  - M: MOSFET
  - Q: BJT
  - R: Resistor
  - T,U,W: Transmission line
  - V: Voltage source
  - X: Subcircuit call

- \texttt{node1 ...} Node names are identifiers of the nodes to which the element is connected. Node names must begin with a letter that may be followed by up to 1023 additional alphanumeric characters. The following characters are not allowed in node names: = ( ) , ‘ <space>

- \texttt{mname} Model reference name is required for all elements except passive devices.

- \texttt{pname1 ...} Element parameter name used to identify the parameter value that follows this name.

- \texttt{expression} Any mathematical expression containing values or parameters, i.e., ‘param1 * val2’
val1 ...

Value assigned to the parameter pname1 or to the corresponding model node. The value can be a number or an algebraic expression.

M = val

Element multiplier. Replicates the element “val” times in parallel.

Example

Q1234567 4000 5000 6000 SUBSTRATE BJTMODEL AREA = 1.0

The previous example specifies a bipolar junction transistor with its collector connected to node 4000, its base connected to node 5000, its emitter connected to node 6000, and its substrate connected to node SUBSTRATE. The transistor parameters are described in the model statement referenced by the name BJTMODEL.

M1 ADDR SIG1 GND SBS N1 10U 100U

The previous example specifies a MOSFET called M1, whose drain, gate, source, and substrate nodes are named ADDR, SIG1, GND, and SBS, respectively. The element statement calls an associated model statement, N1. MOSFET dimensions are specified as width = 100 microns and length = 10 microns.

M1 ADDR SIG1 GND SBS N1 w1+w l1+l

The previous example specifies a MOSFET called M1, whose drain, gate, source, and substrate nodes are named ADDR, SIG1, GND, and SBS, respectively. The element statement calls an associated model statement, N1. MOSFET dimensions are also specified as algebraic expressions, width = w1+w and length = l1+l.

.SUBCKT or .MACRO Statement

The syntax is:

.SUBCKT subnam n1 < n2 n3 ...> < parnam = val ...>

or

.MACRO subnam n1 < n2 n3 ... > < parnam = val ...>
where:

**subnam**
Specifies reference name for the subcircuit model call

**n1 …**
Node numbers for external reference; cannot be ground node (zero). Any element nodes appearing in the subcircuit but not included in this list are strictly local, with three exceptions:

1. the ground node (zero)
2. nodes assigned using BULK = node in the MOSFET or BJT models
3. nodes assigned using the .GLOBAL statement

**parnam**
A parameter name set to a value. For use only in the subcircuit, overridden by an assignment in the subcircuit call or by a value set in a .PARAM statement.

**Example**

```plaintext
*FILE SUB2.SP TEST OF SUBCIRCUITS
.OPTIONS LIST ACCT

* V1 1 0 1
.PARAM P5 = 5 P2 = 10

* .SUBCKT SUB1 1 2 P4 = 4
  R1 1 0 P4
  R2 2 0 P5
  X1 1 2 SUB2 P6 = 7
  X2 1 2 SUB2
  .ENDS

* .MACRO SUB2 1 2 P6 = 11
  R1 1 2 P6
  R2 2 0 P2
  .EOM

* X1 1 2 SUB1 P4 = 6
  X2 3 4 SUB1 P6 = 15
  X3 3 4 SUB2
```

Star-Hspice Manual, Release 20001.2 3-13
The above example defines two subcircuits: SUB1 and SUB2. These are resistor divider networks whose resistance values have been parameterized. They are called with the X1, X2, and X3 statements. Since the resistor value parameters are different in each call, these three calls produce different subcircuits.

**.ENDS or .EOM Statement**

The syntax is:

```
.ENDS <SUBNAM>
```

or

```
.EOM <SUBNAM>
```

**Example**

```
.ENDS OPAMP
.EOM MAC3
```

This statement must be the last for any subcircuit definition. The subcircuit name, if included, indicates which definition is being terminated. Subcircuit references (calls) may be nested within subcircuits.

**Subcircuit Call Statement**

The syntax is:

```
Xyyy n1 <n2 n3 ...> subnam <parnam = val ...> <M = val>
```

where:

- **Xyyy**: Subcircuit element name. Must begin with an “X”, which may be followed by up to 15 alphanumeric characters.
- **n1 ...**: Node names for external reference
- **subnam**: Subcircuit model reference name
parnam
A parameter name set to a value (val) for use only in the subcircuit. It overrides a parameter value assigned in the subcircuit definition, but is overridden by a value set in a .PARAM statement.

$M$
Multiplier. Makes the subcircuit appear as $M$ subcircuits in parallel. This is useful in characterizing circuit loading. No additional calculation time is needed to evaluate multiple subcircuits.

Example

X1 2 4 17 31 MULTI WN = 100 LN = 5

The above example calls a subcircuit model named MULTI. It assigns the parameters WN = 100 and LN = 5 to the parameters WN and LN given in the .SUBCKT statement (not shown). The subcircuit name is X1. All subcircuit names must begin with X.

Example

.SUBCKT YYY NODE1 NODE2 VCC = 5V
.IC NODEX = VCC
R1 NODE1 NODEX 1
R2 NODEX NODE2 1
.EOM
XYYY 5 6 YYY VCC = 3V

The above example defines a subcircuit named YYY. The subcircuit consists of two 1 ohm resistors in series. The subcircuit node, NODEX, is initialized with the .IC statement through the passed parameter VCC.

---

**Note:** A warning message is generated if a nonexistent subcircuit node is initialized. This can occur if an existing .ic file (initial conditions) is used to initialize a circuit modified since the .ic file was created.
Element and Node Naming Conventions

Node Names

Nodes are the points of connection between elements in the input netlist file. In Star-Hspice, nodes are designated by either names or by numbers. Node numbers can be from 1 to 999999999999999; node number 0 is always ground. Letters that follow numbers in node names are ignored. Node names must begin with a letter and are followed by up to 1023 characters.

In addition to letters and digits, the following characters are allowed in node names:

+ plus sign
- minus sign or hyphen
* asterisk
/ slash
$ dollar sign
# pound sign
[] left and right square brackets
! exclamation mark
<> left and right angle brackets
_ underscore
% percent sign

Braces, “{ }”, are allowed in node names, but Star-Hspice changes them to square brackets, “[ ]”.

The following are not allowed in node names:
( left and right parentheses
, comma
= equal sign
‘ apostrophe
blank space
The period is reserved for use as a separator between the subcircuit name and the node name:

<subcircuitName>.<nodeName>.

The sorting order for operating point nodes is

a-z, !, #, $, %, *, +, -, /

**Instance and Element Names**

Star-Hspice elements have names that begin with a letter designating the element type, followed by up to 1023 alphanumeric characters. Element type letters are R for resistor, C for capacitor, M for a MOSFET device, and so on (see “Element and Source Statements” on page 3-10).

**Subcircuit Node Names**

Subcircuit node names are assigned two different names in Star-Hspice. The first name is assigned by concatenating the circuit path name with the node name through the (.) extension – for example, X1.XBIAS.M5.

---

**Note:** Node designations starting with the same number followed by any letter are all the same. For example, 1c and 1d represent the same node.

---

The second subcircuit node name is a unique number that Star-Hspice assigns automatically to an input netlist file subcircuit. This number is concatenated using the (: ) extension with the internal node name, giving the entire subcircuit’s node name (for example, 10:M5). The node name is cross referenced in the output listing file Star-Hspice produces.

The ground node must be indicated by either the number 0, the name GND, or !GND. Every node should have at least two connections, except for transmission line nodes (unterminated transmission lines are permitted) and MOSFET substrate nodes (which have two internal connections). Floating power supply nodes are terminated with a 1 megohm resistor and a warning message.
**Path Names of Subcircuit Nodes**

A path name consists of a sequence of subcircuit names, starting at the highest level subcircuit call and ending at an element or bottom level node. The subcircuit names are separated by periods in the path name. The maximum length of the path name, including the node name, is 1024 characters.

You can use path names in the .PRINT, .PLOT, .NODESET, and .IC statements as an alternative method to reference internal nodes (nodes not appearing on the parameter list). Any node, including any internal node, can be referenced by its path name. Subcircuit node and element names follow the rules illustrated in Figure 3-1.

**Figure 3-1: Subcircuit Calling Tree with Circuit Numbers and Instance Names**

The path name of the node named `sig25` in subcircuit X4 in Figure 3-1 is `X1.X4.sig25`. You can use this path in Star-Hspice statements such as:

```
.PRINT v(X1.X4.sig25)
```

**Abbreviated Subcircuit Node Names**

You can use circuit numbers as an alternative to path names to reference nodes or elements in .PRINT, .PLOT, .NODESET, or .IC statements. Star-Hspice assigns a circuit number to all subcircuits on compilation, creating an abbreviated path name:

```
<subckt-num>:<name>
```
Every occurrence of a node or element in the output listing file is prefixed with the subcircuit number and colon. For example, 4:INTNODE1 denotes a node named INTNODE1 in a subcircuit assigned the number 4.

Any node not in a subcircuit is prefixed by 0: (0 references the main circuit). All nodes and subcircuits are identified in the output listing file with a circuit number referencing the subcircuit where the node or element appears. Abbreviated path names allow use of DC operating point node voltage output as input in a .NODESET for a later run; the part of the output listing titled “Operating Point Information” can be copied or typed directly into the input file, preceded by a .NODESET statement. This eliminates recomputing the DC operating point in the second simulation.

**Automatic Node Name Generation**

Star-Hspice has an automatic system for assigning internal node names. You can check both nodal voltages and branch currents by printing or plotting with the assigned node name. Several special cases for node assignment occur in Star-Hspice. Node 0 is automatically assigned as a ground node.

For CSOS (CMOS Silicon on Sapphire), if the bulk node is assigned the value -1, the name of the bulk node is B#. Use this name for printing the voltage at the bulk node. When printing or plotting current, for example .PLOT I(R1), Star-Hspice inserts a zero-valued voltage source. This source inserts an extra node in the circuit named V\textit{nn}, where \textit{nn} is a number automatically generated by Star-Hspice and appears in the output listing file.

**.GLOBAL Statement**

The .GLOBAL statement globally assigns a node name. This means that all references to a global node name used at any level of the hierarchy in the circuit, will be connected to the same node.

The .GLOBAL statement is most often used when subcircuits are included in a netlist file. This statement assigns a common node name to subcircuit nodes. Power supply connections of all subcircuits are often assigned using a .GLOBAL statement. For example, .GLOBAL VCC connects all subcircuits with the internal node name VCC. Ordinarily, in a subcircuit the node name is
given as the circuit number concatenated to the node name. When a .GLOBAL statement is used, the node name is not concatenated with the circuit number and is only assigned the global name. This allows exclusion of the power node name in the subcircuit or macro call.

Syntax

```plaintext
.GLOBAL node1 node2 node3 ...
```

where:

```
node1 ...
```

Specifies global nodes, such as supply and clock names; overrides local subcircuit definitions.

Example

This example shows global definitions for the nodes VDD and input_sig.

```plaintext
.GLOBAL VDD input_sig
```

**.TEMP Statement**

The temperature of a circuit for a Star-Hspice run is specified with the .TEMP statement or with the TEMP parameter in the .DC, .AC, and .TRAN statements. The circuit simulation temperature set by either of these is compared against the reference temperature set by the TNOM control option. The difference between the circuit simulation temperature and the reference temperature, TNOM, is used in determining the derating factors for component values. Temperature analysis is discussed in “Temperature Analysis” on page 13-5.

Syntax

```plaintext
.TEMP t1 <t2 <t3 ...>>
```

where:

```
t1 t2 ...
```

Specifies temperatures, in °C, at which the circuit is to be simulated.

Example

```plaintext
.TEMP -55.0 25.0 125.0
```
The .TEMP statement sets the circuit temperatures for the entire circuit simulation. Star-Hspice uses the temperature set in the .TEMP statement, along with the TNOM option setting (or the TREF model parameter) and the DTEMP element temperature, and simulates the circuit with individual elements or model temperatures.

**Example**
```
.TEMP 100
D1 N1 N2 DMOD DTEMP = 30
D2 NA NC DMOD
R1 NP NN 100 TC1 = 1 DTEMP = -30
.MODEL DMOD D IS = 1E-15 VJ = 0.6 CJA = 1.2E-13 CJP = 1.3E-14
TREF = 60.0
```

The circuit simulation temperature is given from the .TEMP statement as 100°C. Since TNOM is not specified, it will default to 25°C. The temperature of the diode is given as 30°C above the circuit temperature by the DTEMP parameter; that is, $D1_{temp} = 100°C + 30°C = 130°C$. The diode D2 is simulated at 100°C. R1 is simulated at 70°C. Since TREF is specified at 60°C in the diode model statement, the diode model parameters given are derated by 70°C ($130°C - 60°C$) for diode D1 and by 40°C ($100°C - 60°C$) for diode D2. The value of R1 is derated by 45°C ($70°C - TNOM$).

**.DATA Statement**

Data-driven analysis allows the user to modify any number of parameters, then perform an operating point, DC, AC, or transient analysis using the new parameter values. An array of parameter values can be either inline (in the simulation input file) or stored as an external ASCII file. The .DATA statement associates a list of parameter names with corresponding values in the array.

Data-driven analysis syntax requires a .DATA statement and an analysis statement that contains a $DATA = dataname$ keyword.

The .DATA statement provides a means for using concatenated or column laminated data sets for optimization on measured I-V, C-V, transient, or s-parameter data.
You can also use the .DATA statement for a first or second sweep variable in cell characterization and worst case corners testing. Data measured in a lab, such as transistor I-V data, is read one transistor at a time in an outer analysis loop. Within the outer loop, the data for each transistor (IDS curve, GDS curve, and so on) is read one curve at a time in an inner analysis loop.

The .DATA statement specifies the parameters for which values are to be changed and gives the sets of values that are to be assigned during each simulation. The required simulations are done as an internal loop. This bypasses reading in the netlist and setting up the simulation, and saves computing time. Internal loop simulation also allows simulation results to be plotted against each other and printed in a single output.

You can enter any number of parameters in a .DATA block. The .AC, .DC, and .TRAN statements can use external and inline data provided in .DATA statements. The number of data values per line does not need to correspond to the number of parameters. For example, it is not necessary to enter 20 values on each line in the .DATA block if 20 parameters are required for each simulation pass: the program reads 20 values on each pass no matter how the values are formatted.

(DATA statements are referred to by their datanames, so each dataname must be unique. Star-Hspice supports three .DATA statement formats:

- Inline data
- Data concatenated from external files
- Data column laminated from external files

These formats are described below. The keywords MER and LAM tell Star-Hspice to expect external file data rather than inline data. The keyword FILE denotes the external filename. Simple file names like out.dat can be used without the single or double quotes (’ ’ or “”) , but using them prevents problems with file names that start with numbers like 1234.dat. Remember that file names are case sensitive on UNIX systems.

See chapters 9, 10, and 11 for more details about using the .DATA statement in the different types of analysis. In short, the syntax is:
Operating point:
  .DC DATA = dataname

DC sweep:
  .DC vin 1 5 .25 SWEEP DATA = dataname

AC sweep:
  .AC dec 10 100 10meg SWEEP DATA = dataname

TRAN sweep:
  .TRAN 1n 10n SWEEP DATA = dataname

For any data driven analysis, make sure that the start time (time 0) is specified in the analysis statement, to ensure that the stop time is calculated correctly.

Inline .DATA Statement

Inline data is parameter data listed in a .DATA statement block. It is called by the datanm parameter in a .DC, .AC, or .TRAN analysis statement.

Syntax

```
.DATA datanm pnami <pnami2 pnami3 ... pnami.xxx >
  + pvali<pvali2 pvali3 ... pvali.xxx>
  + pvali’ <pvali2’ pvali3’ ... pvali.xxx’>
.ENDDATA
```

where:

- **datanm** Specifies the data name referred to in the .TRAN, .DC or .AC statement
- **pnami** Specifies the parameter names used for source value, element value, device size, model parameter value, and so on. These names must be declared in a .PARAM statement.
- **pvali** Specifies the parameter value

The number of parameters read in determines the number of columns of data. The physical number of data numbers per line does not need to correspond to the
number of parameters. In other words, if 20 parameters are needed, it is not necessary to put 20 numbers per line.

Example

```
.TRAN 1n 100n SWEEP DATA = devinf
.AC DEC 10 1hz 10khz SWEEP DATA = devinf
.DC TEMP -55 125 10 SWEEP DATA = devinf
.DATA devinf width length thresh cap
+ 50u 30u 1.2v 1.2pf
+ 25u 15u 1.0v 0.8pf
+ 5u 2u 0.7v 0.6pf
+ ... ... ... ...
.ENDDATA
```

Star-Hspice performs the above analyses for each set of parameter values defined in the .DATA statement. For example, the program first takes the width = 50u, length = 30u, thresh = 1.2v, and cap = 1.2pf parameters and performs .TRAN, .AC and .DC analyses. The analyses are then repeated for width = 25u, length = 15u, thresh = 1.0v, and cap = 0.8pf, and again for the values on each subsequent line in the .DATA block.

This is an example of .DATA as the inner sweep:

```
M1 1 2 3 0 N W = 50u L = LN
VGS 2 0 0.0v
VBS 3 0 VBS
VDS 1 0 VDS
.PARAM VDS = 0 VBS = 0 L = 1.0u
.DC DATA = vdot
.DATA vdot
VBS VDS L
0 0.1 1.5u
0 0.1 1.0u
0 0.1 0.8u
-1 0.1 1.0u
-2 0.1 1.0u
-3 0.1 1.0u
0 1.0 1.0u
0 5.0 1.0u
.ENDDATA
```
In the above example, a DC sweep analysis is performed for each set of VBS, VDS, and L parameters in the "DATA vdot" block. That is, eight DC analyses are performed, one for each line of parameter values in the .DATA block.

This is an example of .DATA as an outer sweep:

```
.PARAM W1 = 50u W2 = 50u L = 1u CAP = 0
.TRAN 1n 100n SWEEP DATA = d1
.DATA d1
  W1    W2    L      CAP
  50u    40u   1.0u   1.2pf
  25u    20u   0.8u   0.9pf
.ENDDATA
```

In the previous example, the default start time for the .TRAN analysis is 0, the time increment is 1 ns, and the stop time is 100 ns. This results in transient analyses at every time value from 0 to 100 ns in steps of 1 ns, using the first set of parameter values in the "DATA d1" block. Then the next set of parameter values is read, and another 100 transient analyses are performed, sweeping time from 0 to 100 ns in 1 ns steps. The outer sweep is time, and the inner sweep varies the parameter values. Two hundred analyses are performed: 100 time increments times 2 sets of parameter values.

**External File .DATA Statement**

**Syntax**

This is the syntax for concatenated data files.

```
.DATANNM MER
FILE = 'filename1' pnamel = colnum <pame2 = colnum
  ...
<FILE = 'filename2' pnamel = colnum <pame2 = colnum
  ...
<OUT = 'fileout'>
.ENDDATA
```
where:

- **datanm** Specifies the data name referred to in the .TRAN, .DC or .AC statement.
- **MER** Specifies concatenated (series merging) data files to be used.
- **filenamei** Specifies the name of the data file to be read. Files are concatenated in the order they appear in the .DATA statement. A maximum of 10 files can be specified.
- **pnami** Specifies the parameter names used for source value, element value, device size, model parameter value, and so on. These names must be declared in a .PARAM statement.
- **colnum** Specifies the column number in the data file for the parameter value. The column need not be the same between files.
- **fileouti** Specifies the name of the data file to be written with all the data concatenated. This file will have the complete syntax for an inline .DATA statement, and can replace the .DATA statement that created it in the netlist. Outputting the file is optional, and can be used to generate one data file from many.

Concatenated data files are files with the same number of columns placed one after another. For example, if the three files A, B, and C are concatenated,

<table>
<thead>
<tr>
<th>File A</th>
<th>File B</th>
<th>File C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a a a</td>
<td>b b b</td>
<td>c c c</td>
</tr>
<tr>
<td>a a a</td>
<td>b b b</td>
<td>c c c</td>
</tr>
<tr>
<td>a a a</td>
<td>b b b</td>
<td>c c c</td>
</tr>
</tbody>
</table>

the data appears as follows:

```
a a a
a a a
a a a
b b b
b b b
b b b
```

```
c c c
c c c
```
Note: The number of lines (rows) of data in each file need not be the same. It is assumed that the associated parameter of each column of file A is the same as each column of the other files.

Example

```
.DATA inputdata MER
FILE = 'file1' p1 = 1 p2 = 3 p3 = 4
FILE = 'file2' p1 = 1
FILE = 'file3'
.ENDDATA
```

In the above listing, `file1`, `file2`, and `file3` are concatenated to form the dataset `inputdata`. The data in `file1` is at the top of the file, followed by the data in `file2`, and `file3`. The `inputdata` in the .DATA statement references the dataname given in either the .DC, .AC or .TRAN analysis statements. The parameter fields give the column that the parameters are in (the parameter names must have already been defined in .PARAM statements). For example, the values for parameter `p1` are in column 1 of `file1` and `file2`. The values for parameter `p2` are in column 3 of `file1`.

For data files with fewer columns than others, the missing parameters will be given values of zero.

**Column Laminated .DATA Statement**

**Syntax**

This is the syntax for column laminated data files.

```
.DATA datanm LAM
FILE = 'filename1' pnam1 = colnum <pnam2 = colnum
...>
<FILE = 'filename2' pnam1 = colnum <pnam2 = colnum
...>>
...
<OUT = 'fileout'>
.ENDDATA
```
where:

- \texttt{datanm} Specifies the data name referred to in the .TRAN, .DC or .AC statement
- \texttt{LAM} Specifies column laminated (parallel merging) data files to be used
- \texttt{filenamei} Specifies the name of the data file to be read. Files are concatenated in the order they appear in the .DATA statement. A maximum of 10 files can be specified.
- \texttt{pnami} Specifies the parameter names used for source value, element value, device size, model parameter value, and so on. These names must be declared in a .PARAM statement.
- \texttt{colnum} Specifies the column number in the data file for the parameter value. The column need not be the same between files.
- \texttt{fileouti} Specifies the name of the data file to be written with all the data concatenated. This file will have the complete syntax for an inline .DATA statement, and can replace the .DATA statement that created it in the netlist. Outputting the file is optional, and can be used to generate one data file from many.

Column lamination means that the columns of files with the same number of rows are arranged side-by-side. For example, for three files \( D \), \( E \), and \( F \) containing the following columns of data,

\[
\begin{array}{ccc}
\text{File D} & \text{File E} & \text{File F} \\
d1 & d2 & d3 & e4 & e5 & f6 \\
d1 & d2 & d3 & e4 & e5 & f6 \\
d1 & d2 & d3 & e4 & e5 & f6 \\
\end{array}
\]

the laminated data appears as follows:

\[
\begin{align*}
d1 & \ d2 & \ d3 & \ e4 & \ e5 & \ f6 \\
d1 & \ d2 & \ d3 & \ e4 & \ e5 & \ f6 \\
d1 & \ d2 & \ d3 & \ e4 & \ e5 & \ f6 
\end{align*}
\]
The number of columns of data need not be the same in the three files.

**Note:** The number of lines (rows) of data in each file need not be the same. Data points missing will be interpreted as zero.

**Example**

```
.DATA dataname LAM
FILE = 'file1' p1 = 1 p2 = 2 p3 = 3
FILE = 'file2' p4 = 1 p5 = 2
OUT = 'fileout'
.ENDDATA
```

This listing takes columns from file1, and file2, and laminates them into the output file, fileout. Columns one, two, and three of file1, columns one and two of file2 are designated as the columns to be placed in the output file. There is a limit of 10 files per .DATA statement.

**Note:** When Star-Hspice is run on a different machine than the one on which the input data files reside (such as when working over a network), use full path names instead of aliases whenever possible, since aliases may have different definitions on different machines.

**.INCLUDE Statement**

**Syntax**

```
.INCLUDE '<filepath> filename'
```

where:

- `filepath` Path name of a file for computer operating systems supporting tree structured directories.
- `filename` Name of a file to include in the data file. The file path plus file name can be up to 1024 characters in length and can be any valid file name for the computer’s operating system. The file path and name must be enclosed in single or double quotation marks.
.MODEL Statement

Syntax

.MODEL mname type <VERSION = version_number>
+ <pname1 = val1 pname2 = val2 ...>

where:

mname Model name reference. Elements must refer to the model by this name.

Note: Model names that contain periods (.) can cause the Star-Hspice automatic model selector to fail under certain circumstances.

type Selects the model type, which must be one of the following:

AMP operational amplifier model
C capacitor model
CORE magnetic core model
D diode model
L magnetic core mutual inductor model
NJF n-channel JFET model
NMOS n-channel MOSFET model
NPN npn BJT model
OPT optimization model
PJF p-channel JFET model
PLOT plot model for the .GRAPH statement
PMOS p-channel MOSFET model
PNP pnp BJT model
R resistor model
U lossy transmission line model (lumped)
W lossy transmission line model
SP S parameter
**pname1 ...** Parameter name. The model parameter name assignment list (pname1) must be from the list of parameter names for the appropriate model type. Default values are given in each model section. The parameter assignment list can be enclosed in parentheses and each assignment can be separated by either blanks or commas for legibility. Continuation lines begin with a plus sign (+).

**VERSION** Star-Hspice version number, used to allow portability of the BSIM (LEVEL=13), BSIM2 (LEVEL = 39) models between Star-Hspice releases. Star-Hspice release numbers and the corresponding version numbers are:

<table>
<thead>
<tr>
<th>Star-Hspice release</th>
<th>Version number</th>
</tr>
</thead>
<tbody>
<tr>
<td>9007B</td>
<td>9007.02</td>
</tr>
<tr>
<td>9007D</td>
<td>9007.04</td>
</tr>
<tr>
<td>92A</td>
<td>92.01</td>
</tr>
<tr>
<td>92B</td>
<td>92.02</td>
</tr>
<tr>
<td>93A</td>
<td>93.01</td>
</tr>
<tr>
<td>93A.02</td>
<td>93.02</td>
</tr>
<tr>
<td>95.3</td>
<td>95.3</td>
</tr>
<tr>
<td>96.1</td>
<td>96.1</td>
</tr>
</tbody>
</table>

The VERSION parameter is only valid for LEVEL 13 and LEVEL 39 models, and in Star-Hspice releases starting with Release H93A.02. Using the parameter with any other model or with a release prior to H93A.02 results in a warning message, but the simulation continues. **Note:** VERSION is also used to denote the BSIM3v3 version number only in model LEVELs 49 and 53. For LEVELs 49 and 53, the parameter HSPVER is used to denote the Star-Hspice release number. Please see “Selecting Model Versions” on page 22-20 for more information.

**Example**

```plaintext
.MODEL MOD1 NPN BF=50 IS=1E-13 VBF=50 AREA=2 PJ=3, N=1.05
```
.LIB Call and Definition Statements

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

You also can place a .LIB call statement in an .ALTER block.

 Лиbrary Call Statement

Syntax

```
.LIB '<filepath> filename' entryname
```

where:

- **filepath**
  
  Path to a file. Used where a computer supports tree-structured directories. When the LIB file (or alias) resides in the same directory in which Star-Hspice is run, no directory path need be specified; the netlist runs on any machine. You can use the “../” syntax in the filepath to designate the parent directory of the current directory.

- **filename**
  
  Name of a file to include in the data file. The combination of filepath plus filename may be up to 256 characters long, structured as any valid filename for the computer’s operating system. File path and name must be enclosed in single or double quotation marks. You can use the “../” syntax in the filename to designate the parent directory of the current directory.

- **entryname**
  
  Entry name for the section of the library file to include. The first character of an entryname cannot be an integer.

Example

```
.LIB 'MODELS' cmos1
```
.LIB Library File Definition Statement

You can build libraries by using the .LIB statement in a library file. For each macro in a library, a library definition statement (.LIB entryname) and an .ENDL statement is used. The .LIB statement begins the library macro, and the .ENDL statement ends the library macro.

Syntax
.
.LIB entryname1
.
$ ANY VALID SET OF Star-Hspice STATEMENTS
.
.ENDL entryname1
.LIB entryname2
.
$ ANY VALID SET OF Star-Hspice STATEMENTS
.
.ENDL entryname2
.LIB entryname3
.
$ ANY VALID SET OF Star-Hspice STATEMENTS
.
.ENDL entryname3

The text following a library file entry name must consist of valid Star-Hspice statements.

.LIB Nested Library Calls

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

Example

Shown below are an illegal example and a legal example for a library assigned to library “file3.”
Illegal:
.LIB MOS7
...
.LIB 'file3' MOS7 $ This call is illegal within library MOS7
...
...
.ENDL

Legal:
.LIB MOS7
...
.LIB 'file1' MOS8
.LIB 'file2' MOS9
.LIB CTT $ file2 is already open for CTT entry point
.ENDL

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

Library Building Rules

1. A library cannot contain .ALTER statements.

2. A library may contain nested .LIB calls to itself or other libraries. The depth of nested calls is only limited by the constraints of your system configuration.

3. A library cannot contain a call to a library of its own entry name within the same library file.

4. A library cannot contain the .END statement.

5. .LIB statements within a file called with an .INCLUDE statement cannot be changed by .ALTER processing.
The simulator accesses the models and skew parameters through the .LIB statement and the .INCLUDE statement. The library contains parameters that modify .MODEL statements. The following example of a .LIB of model skew parameters features both worst case and statistical distribution data. The statistical distribution median value is the default for all non-Monte Carlo analysis.

**Example**

```
.LIB TT
$TYPICAL P-CHANNEL AND N-CHANNEL CMOS LIBRARY
$ PROCESS: 1.0U CMOS, FAB7
$ following distributions are 3 sigma ABSOLUTE GAUSSIAN
.PARAM TOX = AGAUSS(200,20,3) $ 200 angstrom +/- 20a
+ XL = AGAUSS(0.1u,0.13u,3) $ polysilicon CD
+ DELVTON = AGAUSS(0.0,.2V,3) $ n-ch threshold change
+ DELVTOP = AGAUSS(0.0,.15V,3) $ p-ch threshold change
.INC '/usr/meta/lib/cmos1_mod.dat' $ model include file
.ENDL TT

.LIB FF
$HIGH GAIN P-CH AND N-CH CMOS LIBRARY 3SIGMA VALUES
.PARAM TOX = 220 XL = -0.03 DELVTON = -.2V DELVTOP = -0.15V
.INC '/usr/meta/lib/cmos1_mod.dat' $ model include file
.ENDL FF
```

The model would be contained in the include file `/usr/meta/lib/cmos1_mod.dat`

```
.MODEL NCH NMOS LEVEL = 2 XL = XL TOX = TOX DELVTO = DELVTON
......
.MODEL PCH PMOS LEVEL = 2 XL = XL TOX = TOX DELVTO = DELVTOP
......
```

---

**Note:** The model keyword (left-hand side) is being equated to the skew parameter (right-hand side). A model keyword can be the same as a skew parameter.
.OPTIONS SEARCH Statement

This statement allows a library to be accessed automatically.

Syntax

.OPTIONS SEARCH = 'directory_path'

Example

.OPTIONS SEARCH = '$installdir/parts/vendor'

The above example sets the search path to find models by way of a vendor subdirectory under the installation directory, $installdir/parts (see Figure 3-2). The DDL subdirectories are contained in the parts/ directory.

Figure 3-2: Vendor Library Usage

Note: The '/usr' directory is in the Star-Hspice install directory.
Automatic Library Selection

Automatic library selection allows a search order for up to 40 directories. The \textit{hspice.ini} file sets the default search paths. Use this file for any directories that should always be searched. Star-Hspice searches for libraries in the order in which libraries are specified in \texttt{.OPTIONS SEARCH} statements. When Star-Hspice encounters a subcircuit call, the search order is as follows:

1. Read the input file for a \texttt{.SUBCKT} or \texttt{.MACRO} with call name.
2. Read any \texttt{.INC} files or \texttt{.LIB} files for a \texttt{.SUBCKT} or \texttt{.MACRO} with the call name.
3. Search the directory that the input file was in for a file named \texttt{call\_name.inc}.
4. Search the directories in the \texttt{.OPTIONS SEARCH} list.

By using the Star-Hspice library search and selection features you can, for example, simulate process corner cases, using \texttt{.OPTIONS SEARCH = ‘<libdir>’} to target different process directories. If you have an input/output buffer subcircuit stored in a file named \texttt{iobuf.inc}, create three copies of the file to simulate fast, slow and typical corner cases. Each file contains different Star-Hspice transistor models representing the different process corners. Store these files (all named \texttt{iobuf.inc}) in separate directories.

\texttt{.PARAM} Statement

Use the \texttt{.PARAM} statement to define parameters. Parameters in Star-Hspice are names that have associated numeric values. You can use any of the following methods to define parameters:

- Simple Parameter
- Algebraic Parameter
- User-Defined Function
- Subcircuit Default Definition
- Predefined Analysis
- Measurement Parameters
**Simple Parameter**

A simple parameter assignment is a constant real number. The parameter keeps this value unless there is a later definition that changes its value, or it is assigned a new value by an algebraic expression during simulation. There is no warning if a parameter is reassigned.

**Syntax**

```
.PARAM (ParamName)=<RealNumber>
```

**Algebraic Parameter**

Algebraic parameter assignments can be made by an algebraic expression of real values, predefined function, user-defined function, or circuit or model values. A complex expression must be enclosed in single quotes to invoke the Star-Hspice algebraic processor unless the expression begins with an alphabetic character and contains no blank spaces. A simple expression consists of a single parameter name.

**Syntax**

```
.PARAM <ParamName>='<Expression>'
```

or

```
.PARAM <ParamName1>=<ParamName2>
```

To use an algebraic expression as an output variable in a .PRINT, .PLOT, or .PROBE statement, use the PAR keyword:

```
.PRINT DC v(3) gain=PAR('v(3)/v(2)')
+ PAR('V(4)/V(2)')
```

**Example**

```
para x=cos(2)+sin(2)
```

**User-Defined Function**

A user-defined function assignment is similar to the definition of an algebraic parameter definition. Star-Hspice extends the algebraic parameter definition to
include function parameters that are used in the algebraic that defines the function.

User-defined functions cannot be nested more than three deep.

**Syntax**

```
.PARAM <ParamName>(<pv1>[<pv2>])='<Expression>'
```

**Subcircuit Default Definition**

The specification of hierarchical subcircuits allows you to pick default values for circuit elements. This is typically used in cell definitions so that the circuit can be simulated with typical values.

**Syntax**

```
.SUBCKT <SubName><PinList>[<SubDefaultsList>]
```

where `SubDefaultsList` is:

```
<SubParam1>=<Expression>[<SubParam1>=<Expression>...]
```

**Predefined Analysis**

Star-hspice has specialized analysis types, primarily Optimization and Monte Carlo, that require a method of controlling the analysis. The syntax for these uses of `.PARAM` are described in “.PARAM Distribution Function Syntax” on page 13-16.

**Measurement Parameters**

`.MEASURE` statements in Star-Hspice produce a type of parameter called a measurement parameter. In general, the rules for measurement parameters are the same as those for standard parameters with the exception of the definition: measurement parameters are not defined in a `.PARAM` statement but directly in the `.MEASURE` statement. For more information, see “Measure Parameter Types” on page 8-38.
.PROTECT Statement

Use the .PROTECT statement to keep models and cell libraries private. The .PROTECT statement suppresses the printout of the text from the list file, like the option BRIEF. The .UNPROTECT command restores normal output functions. In addition, any elements and models located between a .PROTECT and an .UNPROTECT statement inhibit the element and model listing from the option LIST. Any nodes that are contained within the .PROTECT and .UNPROTECT statements are not listed in the .OPTIONS NODE nodal cross reference, and are not listed in the .OP operating point printout.

Syntax

   .PROTECT

.UNPROTECT Statement

The .UNPROTECT statement restores normal output functions from a .PROTECT statement. Any elements and models located between .PROTECT and .UNPROTECT statements inhibit the element and model listing from the option LIST. Any nodes contained within the .PROTECT and .UNPROTECT statements are not listed in either the .OPTIONS NODE nodal cross reference, or in the .OP operating point printout.

Syntax

   .UNPROTECT

.ALTER Statement

You can use the .ALTER statement to rerun a simulation using different parameters and data.

Print and plot statements must be parameterized to be altered. The .ALTER block cannot include .PRINT, .PLOT, .GRAPH or any other input/output statements. You can include all analysis statements (.DC, .AC, .TRAN, .FOUR, .DISTO, .PZ, and so on) in only one .ALTER block in an input netlist file, but only if the analysis statement type has not been used previously in the main program. The .ALTER sequence or block can contain:
Specifying Simulation Input and Controls

Input Netlist File Composition

- Element statements (except source elements)
- .DATA statements
- .DEL LIB statements
- .INCLUDE statements
- .IC (initial condition) and .NODESET statements
- .LIB statements
- .MODEL statements
- .OP statements
- .OPTIONS statements
- .PARAM statements
- .TEMP statements
- .TF statements
- .TRAN, .DC, and .AC statements

Altering Design Variables and Subcircuits

The following rules are used when altering design variables and subcircuits.

1. If the name of a new element, .MODEL statement, or subcircuit definition is identical to the name of an original statement of the same type, the new statement replaces the old. New statements are added to the input netlist file.

2. Element and .MODEL statements within a subcircuit definition can be changed and new element or .MODEL statements can be added to a subcircuit definition. Topology modifications to subcircuit definitions should be put into libraries and added with .LIB and deleted with .DEL LIB.

3. If a parameter name of a new .PARAM statement in the .ALTER module is identical to a previous parameter name, the new assigned value replaces the old.

4. If elements or model parameter values were parameterized when using .ALTER, these parameter values must be changed through the .PARAM statement. Do not redescribe the elements or model parameters with numerical values.

5. Options turned on by an .OPTION statement in an original input file or a .ALTER block can be turned off.
6. Only the actual altered input is printed for each .ALTER run. A special .ALTER title identifies the run.

7. .LIB statements within a file called with an .INCLUDE statement cannot be revised by .ALTER processing, but .INCLUDE statements within a file called with a .LIB statement can be accepted by .ALTER processing.

Using Multiple .ALTER Statements

For the first run, Star-Hspice reads the input file only up to the first .ALTER statement and performs the analyses up to that .ALTER statement. After the first simulation is completed, Star-Hspice reads the input between the first .ALTER and the next .ALTER or .END statement. These statements are then used to modify the input netlist file. Star-Hspice then resimulates the circuit.

For each additional .ALTER statement, Star-Hspice performs the simulation preceding the first .ALTER statement, then performs another simulation using the input between the current .ALTER statement and the next .ALTER statement or the .END statement. If you do not want to rerun the simulation preceding the first .ALTER statement every time, put the statements preceding the first .ALTER statement in a library and use the .LIB statement in the main input file, and put a .DEL LIB statement in the .ALTER section to delete that library.

Syntax

```
.ALTER <title_string>
```

The `title_string` is any string up to 72 characters. The appropriate title string for each .ALTER run is printed in each section heading of the output listing and the graph data (.tr#) files.

.DEL LIB Statement

The .DEL LIB statement is used with the .ALTER statement to remove library data from memory. The .DEL LIB statement causes the .LIB call statement with the same library number and entry name to be removed from memory the next time the simulation is run. A .LIB statement can then be used to replace the deleted library.
Specifying Simulation Input and Controls

Input Netlist File Composition

Syntax

`.DEL LIB '<filepath>filename' entryname`
`.DEL LIB libnumber entryname`

where:

`entryname` Entry name used in the library call statement to be deleted.

`filename` Name of a file for deletion from the data file. The file path plus file name can be up to 64 characters in length and can be any file name that is valid for the operating system being used. The file path and name must be enclosed in single or double quote marks.

`filepath` Path name of a file, if the operating system supports tree-structured directories.

`libnumber` Library number used in the library call statement to be deleted.

Example

FILE1: ALTER1 TEST CMOS INVERTER
.OPTIONS ACCT LIST
.TEMP 125
.PARAM WVAL = 15U VDD = 5
*
.OP
.DC VIN 0 5 0.1
.PLOT DC V(3) V(2)
*
VDD 1 0 VDD
VIN 2 0
*
M1 3 2 1 1 P 6U 15U
M2 3 2 0 0 N 6U W = WVAL
*
.LIB 'MOS.LIB' NORMAL
.ALTER
.DEL LIB 'MOS.LIB' NORMAL $removes LIB from memory $PROTECTION
**Input Netlist File Composition**

*.PROT

.PROT $protect statements below .PROT

.LIB 'MOS.LIB' FAST $get fast model library

.UNPROT

.ALTER

.OPTIONS NOMOD OPTS $suppress model parameters printing

* and print the option summary

.TEMP -50 0 50 $run with different temperatures

.PARAM WVAL = 100U VDD = 5.5 $change the parameters

VDD 1 0 5.5 $using VDD 1 0 5.5 to change the

$power supply VDD value doesn't

$work

VIN 2 0 PWL 0NS 0 2NS 5 4NS 0 5NS 5 $change the input source

.OP VOL $node voltage table of operating

$points

.TRAN 1NS 5NS $run with transient also

M2 3 2 0 0 N 6U WVAL $change channel width

.MEAS SW2 TRIG V(3) VAL = 2.5 RISE = 1 TARG V(3)

+ VAL = VDD CROSS = 2 $measure output

* .END

Example 1 calculates a DC transfer function for a CMOS inverter. The device is first simulated using the inverter model NORMAL from the MOS.LIB library. By using the .ALTER block and the .LIB command, a faster CMOS inverter, FAST, is substituted for NORMAL and the circuit is resimulated. With the second .ALTER block, DC transfer analysis simulations are executed at three different temperatures and with an n-channel width of 100 µm instead of 15 µm. A transient analysis also is conducted in the second .ALTER block, so that the rise time of the inverter can be measured (using the .MEASURE statement).

This is a second example:

FILE2: ALTER2.SP CMOS INVERTER USING SUBCIRCUIT

.OPTIONS LIST ACCT

.MACRO INV 1 2 3

M1 3 2 1 1 P 6U 15U

M2 3 2 0 0 N 6U 8U

.LIB 'MOS.LIB' NORMAL

.EOM INV

XINV 1 2 3 INV

VDD 1 0 5

VIN 2 0

.DC VIN 0 5 0.1

In this example, the .ALTER block adds a resistor and capacitor network to the circuit. The network is connected to the output of the inverter and a DC small-signal transfer function is simulated.

.END Statement

The Star-Hspice input netlist file must have an .END statement as the last statement. The period preceding END is a required part of the statement. Any text that follows the .END statement is treated as a comment and has no effect on that simulation. A Star-Hspice input file that contains more than one Star-Hspice run must have an .END statement for each Star-Hspice run. Any number of simulations may be concatenated into a single file.

Syntax

.END <comment>

Example

MOS OUTPUT
.OPTIONS NODE NOPAGE
VDS 3 0
VGS 2 0
M1 1 2 0 0 MOD1 L = 4U W = 6U AD = 10P AS = 10P
Specifying Simulation Input and Controls

```
.MODEL MOD1 NMOS VTO = -2 NSUB = 1.0E15 TOX = 1000 UO = 550
VIDS 3 1
.DC   VDS 0 10 0.5   VGS 0 5 1
.PRINT DC I(M1) V(2)
.END MOS OUTPUT

MOS CAPS
.OPTIONS SCALE = 1U SCALM = 1U WL ACCT
.OP
.TRAN .1 6
V1 1 0 PWL 0 -1.5V 6 4.5V
V2 2 0 1.5VOLTS
MODN1 2 1 0 0 M 10 3
.MODEL M NMOS VTO = 1 NSUB = 1E15 TOX = 1000 UO = 800
LEVEL = 1
+ CAPOP = 2
.PLOT TRAN V(1) (0,5) LX18(M1) LX19(M1) LX20(M1) (0,6E-13)
.END MOS CAPS
```
Using Subcircuits

Reusable cells are the key to saving labor in any CAD system, and this also applies to circuit simulation. To create a reusable circuit, it must be constructed as a subcircuit. Use parameters to expand the utility of a subcircuit. SPICE includes the basic subcircuit but does not provide for the consistent naming of nodes. Star-Hspice provides a simple method for the naming of the subcircuit nodes and elements: simply prefix the node or element with the subcircuit call name.

Figure 3-3: Subcircuit Representation

The following Star-Hspice input creates an instance named X1 of the INV cell macro, which consists of two MOSFETs, MN and MP:

```
X1 IN OUT VD_LOCAL VS_LOCAL inv W = 20
.MACRO INV IN OUT VDD VSS W = 10 L = 1 DJUNC = 0
    MP OUT IN VDD VDD PCH W = W L = L DTEMP = DJUNC
    MN OUT IN VSS VSS NCH W = ‘W/2’ L = L DTEMP = DJUNC
.EOM
```

Note: To access the name of the MOSFET inside of the subcircuit INV called by X1, the names are X1.MP and X1.MN. So to print the current through the MOSFETs:

```
.PRINT I (X1.MP)
```
Hierarchical Parameters

The most basic subcircuit parameter is the M or multiply parameter. This is actually a keyword common to all elements including subcircuits, except for voltage sources. The multiply parameter multiplies the internal component values to give the effect of making parallel copies of the element or subcircuit. To simulate the effect of 32 output buffers switching simultaneously, you only need to place one subcircuit:

X1 in out buffer M = 32

Multiply works hierarchically. A subcircuit within a subcircuit is multiplied by the product of both levels.

Figure 3-4: Hierarchical Parameters Simplify Flip-flop Initialization

Example

X1 D Q Qbar CL CLBAR dlatch flip = 0
macro dlatch
+ D Q Qbar CL CLBAR flip = vcc
.nodeset v(din) = flip
xinvl din qbar inv
xinv2 Qbar Q inv
m1 q CLBAR din nch w = 5 l = 1
m2 D CL din nch w = 5 l = 1
.eom
There is no limit to the size or complexity of subcircuits; they may contain subcircuit references and any model or element statement. To specify subcircuit nodes in .PRINT or .PLOT statements, give the full subcircuit path and node name.

**Undefined Subcircuit Search**

When a subcircuit call is in a data file that does not contain the subcircuit description, Star-Hspice automatically searches the:

1. Present directory for the file
2. Directories specified in any .OPTION SEARCH = “directory_path_name” statement
3. Directory where the Discrete Device Library is located.

Star-Hspice searches for the model reference name file with an .inc suffix. For example, if an undefined subcircuit such as “X 1 1 2 INV” is included in the data file, Star-Hspice searches the system directories for the file called inv.inc and when found, places it in the calling data file.
Discrete Device Libraries

Avant!’s Discrete Device Library (DDL) is a collection of Star-Hspice models of discrete components. The $installdir/parts directory contains the various subdirectories that make up the DDL. BJT, MESFET, JFET, MOSFET, and diode models are derived from laboratory measurements using Avant!’s ATEM discrete device characterization system. Behavior of op-amp, comparator, timer, SCR and converter models closely resembles that described in manufacturers’ data sheets. Op-amp models are created using the built-in Star-Hspice op-amp model generator.

Note: $installdir is an environment variable whose value is the path name to the directory in which Star-Hspice is installed. That directory is called the installation directory. The installation directory contains subdirectories such as /parts and /bin, as well as certain files, such as a prototype meta.cfg file and Star-Hspice license files.

DDL Library Access

To include a DDL library component in a data file, use the X subcircuit call statement with the DDL element call. The DDL element statement includes the model name that is used in the actual DDL library file. For example, the following Star-Hspice element statement creates an instance of the 1N4004 diode model:

X1 2 1 D1N4004

where D1N4004 is the model name. See “Element and Source Statements” on page 3-10 and the chapters on specific types of devices for descriptions of element statements.

Optional parameter fields in the element statement can override the internal specification of the model. For example, for op-amp devices, the designer can override offset voltage and gain and offset current. Since the DDL library devices are based on Star-Hspice circuit level models, the effects of supply voltage, loading, and temperature are automatically compensated for in a simulation.
Star-Hspice accesses DDL models in several ways on most computers:

1. An *hspice.ini* initialization file is created when the installation script is run. The search path for the DDL and vendor libraries is written into a .OPTIONS SEARCH = ‘<lib_path>’ statement to give all users immediate access to all libraries. The models are automatically included on usage in the input netlist. When a model or subcircuit is referenced in the input netlist, the directory to which the = DDLPATH environment variable points is searched for a file with the same name as the reference name. This file is an include file, so its filename has the suffix .inc. The DDLPATH variable is set in the *meta.cfg* configuration file when Star-Hspice is installed.

2. Set .OPTIONS SEARCH = ‘<library_path>’ in the input netlist. This method allows you to list personal libraries to be searched. The default libraries referenced in the *hspice.ini* file are searched first. Libraries are searched in the order in which they are encountered in the input file.

3. Directly include a specific model using the .INCLUDE statement. For example, to use a model named T2N2211, store the model in a file named *T2N2211.inc* and put the following statement in the input file:

   .INCLUDE <path>/T2N2211.inc

   Since this method requires that each model be stored in its own .inc file, it is not generally useful, but it can be used for debugging new models when the number of models to be tested is small.

### Vendor Libraries

The interface between commercial parts and circuit or system simulation is the vendor library. ASIC vendors provide comprehensive cells corresponding to inverters, gates, latches, and output buffers. Memory and microprocessor vendors generally supply input and output buffers. Interface vendors supply complete cells for simple functions and output buffers for generic family output. Analog vendors supply behavioral models. To avoid name and parameter conflicts, vendor cell libraries should keep their models within the subcircuit definitions.
Subcircuit Library Structure

Your library structure must adhere to the Star-Hspice implicit subcircuit .INCLUDE statement specification feature. This Star-Hspice function allows you to specify the directory that the subcircuit file (\texttt{<subname>.inc}) resides in, and then reference the name \texttt{<subname>} in each subcircuit call.

Component naming conventions require that each subcircuit be of the form \texttt{<subname>.inc} and stored in a directory that is accessible through the \texttt{.OPTIONS SEARCH = `<libdir>’} statement.

Create subcircuit libraries in a hierarchical structure. This typically implies that the top-level subcircuit describes the input/output buffer fully and any hierarchy is buried inside. The buried hierarchy can include lower level components, model statements, and parameter assignments. Your library cannot use the Star Hspice .LIB or .INCLUDE statements anywhere in the hierarchy.
Using Standard Input Files

This section describes how to use standard input files.

Design and File Naming Conventions

The design name identifies the circuit and any related files, including schematic and netlist files, simulator input and output files, design configuration files and hardcopy files. Both Star-Hspice and AvanWaves extract the design name from their input files and perform subsequent actions based on that name. For example, AvanWaves reads the `<design>.cfg` configuration file to restore node setups used in previous AvanWaves runs.

Both Star-Hspice and AvanWaves read and write files related to the current circuit design. All files related to a design generally reside in one directory, although the output file is standard output on UNIX platforms and can be redirected.

Star-Hspice input file types and their standard names are listed in Table 3-1. These files are described in the following sections.

<table>
<thead>
<tr>
<th>Input File Type</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output configuration file</td>
<td><em>meta.cfg</em></td>
</tr>
<tr>
<td>Initialization file</td>
<td><em>hspice.ini</em></td>
</tr>
<tr>
<td>DC operating point initial conditions file</td>
<td><code>&lt;design&gt;.ic</code></td>
</tr>
<tr>
<td>Input netlist file</td>
<td><code>&lt;design&gt;.sp</code></td>
</tr>
<tr>
<td>Library input file</td>
<td><code>&lt;library_name&gt;</code></td>
</tr>
<tr>
<td>Analog transition data file</td>
<td><code>&lt;design&gt;.d2a</code></td>
</tr>
</tbody>
</table>
Configuration File (*meta.cfg*)

This file sets up the printer, plotter, and terminal. It includes a line, 
`default_include = file name`, which sets up a path to the default `.ini` file
(`hspice.ini`, for example).

The `default_include` file name is case sensitive (except for the PC and Windows
versions of Star-Hspice).

Initialization File (*hspice.ini*)

User defaults are specified in an `hspice.ini` initialization file. If an `hspice.ini` file
exists in the run directory, Star-Hspice includes its contents at the top of the
Star-Hspice input file.

Other ways to include initialization files are to define
“DEFAULT_INCLUDE = <filename>” in the system or in a `meta.cfg` file.

Typical uses of an initialization file are to set options (with an .OPTIONS
statement) and for library access, as is done in the Avant! installation procedure.

DC Operating Point Initial Conditions File (*<design>.ic*)

The `<design>.ic` file is an optional input file that contains initial DC conditions
for particular nodes. You can use it to initialize DC conditions, with either a
.NODESET or an .IC statement.

The .SAVE statement creates a `<design>.ic` file. A subsequent .LOAD
statement initializes the circuit to the DC operating point values in the
`<design>.ic` file.
Output Files

Star-Hspice produces various types of output files, as listed in the following table.

Table 3-2: Star-Hspice Output Files and Suffixes

<table>
<thead>
<tr>
<th>Output File Type</th>
<th>Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output listing</td>
<td>.lis, or user-specified</td>
</tr>
<tr>
<td>Transient analysis results</td>
<td>.tr# †</td>
</tr>
<tr>
<td>Transient analysis measurement results</td>
<td>.mt#</td>
</tr>
<tr>
<td>DC analysis results</td>
<td>.sw# †</td>
</tr>
<tr>
<td>DC analysis measurement results</td>
<td>.ms#</td>
</tr>
<tr>
<td>AC analysis results</td>
<td>.ac# †</td>
</tr>
<tr>
<td>AC analysis measurement results</td>
<td>.ma#</td>
</tr>
<tr>
<td>Hardcopy graph data (from meta.cfg PRTDEFAULT)</td>
<td>.gr# ††</td>
</tr>
<tr>
<td>Digital output</td>
<td>.a2d</td>
</tr>
<tr>
<td>FFT analysis graph data</td>
<td>.ft# †††</td>
</tr>
<tr>
<td>Subcircuit cross-listing</td>
<td>.pa#</td>
</tr>
<tr>
<td>Output status</td>
<td>.st#</td>
</tr>
<tr>
<td>Operating point node voltages (initial conditions)</td>
<td>.ic</td>
</tr>
</tbody>
</table>

# is either a sweep number or a hardcopy file number.
† Only created if a .POST statement is used to generate graphical data.
†† Requires a .GRAPH statement or a pointer to a file exists in the meta.cfg file.
††† This file is not generated by the PC version of Star-Hspice.
†††† Only created if a .FFT statement is used.

The files are listed in Table 3-2 and described below.
Output listing can appear as output_file (no file extension), output_file.lis, or have a user-specified file extension, depending upon which format is used to start the simulation. Output_file is the output file specification, less extension. This file includes the following information:

- Name and version of simulator used
- Avant! message block
- Input file name
- User name
- License details
- Copy of the input netlist file
- Node count
- Operating point parameters
- Details of volt drop, current, and power for each source and subcircuit
- Low resolution plots originating from the .PLOT statement
- Results of .PRINT statement
- Results of .OPTIONS statements

Transient analysis results are placed in output_file.tr#, where # is specified as 0-9 or a-z following the -n argument. This file contains a list of transient analysis numerical results. It is the result of an input file .TRAN statement together with an .OPTION POST statement to create a post-analysis file. The output file is in proprietary binary format if POST = 0 or 1, or ASCII format if POST = 2. The explicit expressions POST = BINARY, POST=ASCII may also be used.

Transient analysis measurement results are written to output_file.mt#. This output file is the result of an input file .MEASURE TRAN statement.

DC analysis results appear in output_file.sw#, which is produced as a result of a .DC statement. This file contains the results of the applied stepped or swept DC parameters defined in that statement. The results may include noise, distortion, or network analysis.

DC analysis measurement results are given in the file output_file.ms# when a .MEASURE DC statement exists in the input file.
AC analysis results are placed in output_file.ac#. These results contain a listing of output variables as a function of frequency, according to user specification following the .AC statement.

AC analysis measurement results appear in output_file.ma# when a .MEASURE AC statement exists in the input file.

Hardcopy graph data are placed in output_file.gr#, which is produced as a result of a .GRAPH statement. It is in the form of a printer file, typically in Adobe PostScript or HP PCL format. This facility is not available in the PC version of Star-Hspice.

Digital output contains data converted to digital form by the U element A2D conversion option.

FFT analysis graph data contains the graphical data needed to display the FFT analysis waveforms.

Subcircuit cross-listing is automatically generated and written into output_file.pa# when the input netlist includes subcircuits. It relates the subcircuit node names in subcircuit call statements to the node names used in the corresponding subcircuit definitions.

Output status is named with the output file specification, with a .st# extension, and contains the following runtime reports:

- Start and end times for each CPU phase
- Options settings with warnings for obsolete options
- Status of preprocessing checks for licensing, input syntax, models, and circuit topology
- Convergence strategies used by Star-Hspice on difficult circuits

The information in this file is useful in diagnosing problems, particularly when communicating with Avant! Customer Support.

Operating point node voltages are DC operating point initial conditions stored by the .SAVE statement.
Using the Star-Hspice Command

You can start Star-Hspice in either a prompting mode or a nonprompting command line mode.

Prompting Script Mode

Use the following procedure to start Star-Hspice in the prompting mode.

1. cd to your Star-Hspice run directory and type:
   
   hspice

2. The following prompt appears:
   
   Enter input file name:

3. Enter the name of your Star-Hspice input netlist file. If you do not include a file name extension, Star-Hspice looks for the file name with an .sp extension.

   If no file name exists with the name you enter, the following message appears and the Star-Hspice startup script terminates:
   
   **error** Cannot open input file <filename>

4. The following prompt appears:
   
   Enter output file name or directory:
   
   [<filename>.lis]

5. Enter the path and name you want to give the Star-Hspice output listing file. The default is the input file name with a .lis extension.

6. A numbered list of the Star-Hspice versions that are available appears, followed by a prompt to specify the version you want to run. Enter the number in the list of the Star-Hspice version you want to run.

7. For releases of Star-Hspice prior to Release H93A.02, the following prompt appears:
   
   How much memory is needed for this run?
Enter the number of 8-byte words of memory you want to allocate for the Star-Hspice run.

8. The following prompt appears:
   Run Star-Hspice at a lower priority? (y,n) [n]

9. To use the default priority, enter n, or just press Return.
   To specify the priority, enter y. The following prompt appears:
   HINT: The larger the number the lower the priority.
   Enter the priority scheduling factor: (5 10 15 20) [15]

   The default is 15. Enter your choice from the list of factors.

The Star-Hspice run begins.

Nonprompting Command Line Mode

Star-Hspice accepts the following arguments when run in the nonprompting command line mode:

   hspice <-i> <path/>input_file <-v HSPICE_version>
   +<-n number> <-a arch> <-o path>/output_file>

where:

   input_file       Specifies the input netlist file name, for which an extension <ext> is optional. If no input filename extension is provided in the command, Star-Hspice searches for a file named <input_file>.sp. The input file can be preceded by -i. The input filename is used as the root filename for the output files. Star-Hspice also checks to see if there is an initial conditions file (.ic) with the input file root name. The following is an example of an input file name: /usr/sim/work/rb_design.sp where: /usr/sim/work/ is the directory path to the design rb_design is the design root name .sp is the filename suffix

   -v               Specifies the version of Star-Hspice to use.
-n Specifies the number at which to start numbering output data file revisions (output_file.tr#, output_file.ac#, output_file.sw#, where # is the revision number).

-a Is an argument that overrides the default architecture.

Available Star-Hspice command arguments are listed in Table 3-3.

### Table 3-3: Star-Hspice Command Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a &lt;arch&gt;</td>
<td>Platform architecture. Choices are: • sun4, sol4 (SparcStation, Ultra) • pa (HP 700/800/9000) • alpha (DEC ALPHA) • rs (IBM RS6000) • sgi (SGI) • pc (Windows 95/NT)</td>
</tr>
<tr>
<td>-i &lt;input_file&gt;</td>
<td>Name of the input netlist file. If no extension is given, .sp is assumed.</td>
</tr>
<tr>
<td>-m &lt;mem_needed&gt;</td>
<td>Amount of memory requested for the simulation, in 8-byte words (only required for Star-Hspice releases prior to Release H93A.01)</td>
</tr>
<tr>
<td>-n &lt;number&gt;</td>
<td>Revision number at which to start numbering .gr#, .tr#, and other output files. By default, the file numbers start at zero: .gr0, .tr0, and so on. This option allows you to specify the number (-n 7 for .gr7, .tr7, for example).</td>
</tr>
<tr>
<td>-o &lt;output_file&gt;</td>
<td>Name of the output file. If no extension is given, .lis is assigned.</td>
</tr>
<tr>
<td>-r &lt;remote_host&gt;</td>
<td>Name of the machine on which to run the simulation</td>
</tr>
<tr>
<td>-v &lt;version&gt;</td>
<td>Star-Hspice version. Choices are determined at the time of installation by the Star-Hspice installation script.</td>
</tr>
<tr>
<td>-x</td>
<td>Displays the Star-Hspice script on the window as it runs</td>
</tr>
</tbody>
</table>
You do not need to include a filename extension in the output file specification. Star-Hspice names it `output_file.lis`. In output file names, Star-Hspice considers everything up to the final period to be the root filename, and everything following the last period to be the filename extension.

If you do not enter an output filename with the `-o` option, the input root filename is used as the output file root filename. If you include the extension `.lis` in the filename you enter with `-o`, Star-Hspice does not append another `.lis` extension to the output file root filename.

If no output file is specified, output is directed to the terminal. Use the following syntax to redirect the output to a file instead of the terminal:

```
hspice input_file <-v HSPICE_version> <-n number> <-a arch> > output_file
```

For example, for the invocation command

```
hspice demo.sp -v /usr/meta/96 -n 7 -a sun4 > demo.out
```

where:

- `demo.sp` Is the input netlist file; the `.sp` extension to the input filename is optional
- `-v /usr/meta/96` Specifies the version of Star-Hspice to use
- `-n 7` Starts the output data file revision numbers at 7: `demo.tr7`, `demo.ac7`, and `demo.sw7`
- `-a sun4` Overrides the default platform
- `>` Redirects the program output listing to `demo.out`
Sample Star-Hspice Commands

Some additional examples of Star-Hspice commands are explained below.

- **hspice -i demo.sp**
  
  “demo” is the root filename. Output files are named *demo.lis*, *demo.tr0*, *demo.st0*, and *demo.ic*.

- **hspice -i demo.sp -o demo**
  
  “demo” is the output file root name (designated by the -o option). Output files are named *demo.lis*, *demo.tr0*, *demo.st0*, and *demo.ic*.

- **hspice -i rbdir/demo.sp**
  
  “demo” is the root name. Output files *demo.lis*, *demo.tr0*, and *demo.st0* are written in the directory where the Star-Hspice command is executed. Output file *demo.ic* is written in the same directory as the input source, that is, *rbdir*.

- **hspice -i a.b.sp**
  
  “a.b” is the root name. The output files are */a.b.lis*, */a.b.tr0*, */a.b.st0*, and */a.b.ic*.

- **hspice -i a.b -o d.e**
  
  “a.b” is the root name for the input file. “d.e” is the root for output file names except for the .ic file, which is given the input file root name “a.b”. The output files are *d.e.lis*, *d.e.tr0*, *d.e.st0*, and *a.b.ic*.

- **hspice -i a.b.sp -o outdir/d.e**
  
  “a.b” is the root for the .ic file. The .ic file is written in a file named *a.b.ic*.
  
  “d.e” is the root for other output files. Output files are *outdir/d.e.lis*, *outdir/d.e.tr0*, and *outdir/d.e.st0*.

- **hspice -i indir/a.b.sp -o outdir/d.e.lis**
  
  “a.b” is the root for the .ic file. The .ic file is written in a file named *indir/a.b.ic*.
  
  “d.e” is the root for the output files.


**Improving Simulation Performance Using Multithreading**

Star-Hspice simulations involve both model evaluations and matrix solutions. Running model evaluations concurrently on multiple CPUs using multithreading can significantly improve simulation performance. In most cases, the model evaluation will dominate. To determine how much time is spent in model evaluation and solving, specify `.option acct = 2` in the netlist. Using multithreading results in faster simulations with no loss of accuracy.

Multithreaded (MT) Star-Hspice is supported on Sun Solaris 2.5.1 (SunOS 5.5.1) and on Windows NT as a prerelease version using win32 threads.

**Running Star-Hspice-MT**

You can run Star-Hspice-MT using the syntax described below.

**Sun Solaris Platform**

Enter on the command line:

```
hspace -mt #num -i input_filename -o output_filename
```

**Windows NT Platform**

Under the Windows NT DOS prompt type:

```
hsp_mt -mt #num -i input_filename -o output_filename
```

**Note:** If the `#num` is omitted, the number of threads will be set to the number of online CPUs.

If you omit the `-o output_file` option, the result will be printed to the standard output.

Under Windows NT explorer:

1. Double click the `hsp_mt` application icon.
2. Select the **File/ Simulate** button to select the input netlist file.
In Windows, the program will automatically detect and use the number of online CPUs.

Under the Avant! HSPUI interface:

1. Select the correct version of hsp_mt.exe in the Version Combo Box.
2. Select the correct number of processors in the MT Option Box.
3. Click the Open button to select the input netlist file.
4. Click the Simulate button to start the simulation.

**Performance Improvement Estimations**

For multithreaded Star-Hspice, the CPU time is:

\[ T_{mt} = T_{serial} + T_{parallel}/N_{cpu} + T_{overhead} \]

where:

- **\( T_{serial} \)** Represents the Star-Hspice calculations that are not threaded
- **\( T_{parallel} \)** Represents the threaded Star-Hspice calculations
- **\( N_{cpu} \)** The number of CPUs used. **\( T_{overhead} \)** is the overhead from multithreading. Typically, this represents a small fraction of the total run time.

For example, for a 151-stage nand ring oscillator using LEVEL 49, **\( T_{parallel} \)** is about 80% of **\( T_{1cpu} \)** (the CPU time associated with a single CPU), if you run with two threads on a multi-CPU machine. Ideally, assuming **\( T_{overhead} = 0 \)**, you can achieve a speedup of:

\[ T_{1cpu}/(0.2T_{1cpu} + 0.8T_{1cpu}/2cpus) = 1.67 \]

For six CPUs the speedup is:

\[ T_{1cpu}/(0.2T_{1cpu} + 0.8T_{1cpu}/6cpus) = 3.0 \]

The typical value of **\( T_{parallel} \)** is 0.6 to 0.7 for moderate to large circuits.
Chapter 4

Using Elements

This chapter describes the syntax for the basic elements of a circuit netlist. Please refer to the device model chapters in Volumes II and III for detailed syntax descriptions and model descriptions.

This chapter covers the following topics:

- Passive Elements
- Active Elements
- Transmission Lines
- Buffers
Resistors

The general syntax for including a resistor element in a Star-Hspice netlist is:

**General form:**

\[ \text{Rxxx n1 n2 <mname> <R = >resistance <<TC1 = >val> <<TC2 = >val> + <SCALE = val> <M = val> <AC = val> <DTEMP = val> <L = val> + <W = val> <C = val> } \]

where the resistance can be either a value (in units of ohms) or an equation. The only required fields are the two nodes and the resistance or the model name. If the parameter labels are used, the optional arguments may come in any order, although the nodes and model name must come first. If a resistor model is specified (see “Using Passive Device Models” on page 14-1), the resistance value is optional.

The arguments are defined as:

- **Rxxx**  
  Resistor element name. Must begin with “R”, which can be followed by up to 1023 alphanumeric characters.

- **n1**  
  Positive terminal node name

- **n2**  
  Negative terminal node name

- **mname**  
  Resistor model name. This name is used in elements to reference a resistor model.

- **R = resistance**  
  Resistance value at room temperature. This may be a numeric value or parameter in ohms, or a function of any node voltages, branch currents, or any independent variables such as time, frequency (HERTZ), or temperature.

- **TC1**  
  First-order temperature coefficient for the resistor. Refer to “Resistor Temperature Equations” on page 14-8 for temperature-dependent relations.
Using Elements

Passive Elements

**TC2**
Second-order temperature coefficient for the resistor

**SCALE**
Element scale parameter; scales resistance by its value.
Default = 1.0.

**M**
Multiplier used to simulate parallel resistors. For example, to represent two parallel instances of a resistor, set M = 2 to multiply the number of resistors by 2. Default = 1.0.

**AC**
AC resistance used in the AC analysis. Default = Reff.

**DTEMP**
Temperature difference between the element and the circuit in Celsius. Default = 0.0.

**L**
Resistor length in meters. Default = 0.0, if L is not specified in a resistor model.

**W**
Resistor width. Default = 0.0, if W is not specified in the model.

**C**
Capacitance connected from node n2 to bulk. Default = 0.0, if C is not specified in a resistor model.

**Example**

In the following example, resistor R1 is connected from node Rnode1 to node Rnode2 with a resistance of 100 ohms.

R1 Rnode1 Rnode2 100

Resistor RC1 connected from node 12 to node 17 with a resistance of 1 kilohm, and temperature coefficients of 0.001 and 0.

RC1 12 17 R = 1k TC1 = 0.001 TC2 = 0

Resistor Rterm connected from node input to ground with a resistance determined by the square root of the analysis frequency (nonzero for AC analysis only).

Rterm input gnd R = ’sqrt (HERTZ)’

Resistor Rxxx from node 98999999 to node 87654321 with a resistance of 1 ohm for DC and time-domain analyses, and 10 gigohms for AC analyses.

Rxxx 98999999 87654321 1 AC = 1e10
Capacitors

The general syntax for including a capacitor element in a Star-Hspice netlist is:

**General form:**

```
Cxxx n1 n2 <mname> <C = >capacitance <<TC1 = >val>
<<TC2 = >val>
+ <SCALE = val> <IC = val> <M = val> <W = val> <L = val>
+ <DTEMP = val>
```

or

```
Cxxx n1 n2 <C = >’equation’ <CTYPE = val> <above options...>
```

**Polynomial form:**

```
Cxxx n1 n2 POLY c0 c1... <above options...>
```

where the capacitance can be specified as a numeric value in units of farads, as an equation or as a polynomial of the voltage. The only required fields are the two nodes and the capacitance or model name. If the parameter labels are used, the optional arguments may come in any order, although the nodes and model name must come first. If a capacitor model is specified (see “Using Passive Device Models” on page 14-1), the capacitance value is optional.

If the equation form of the capacitance specification is used, the CTYPE parameter is used to determine the method of capacitance charge calculation. The calculation is different depending on whether a self-referential voltage is used in the equation (that is, the voltage across the capacitor whose capacitance is determined by the equation).

To avoid syntactic conflicts, if a capacitor model exists using the same name as a parameter used to specify the capacitance, the model name is taken. In the following example, C1 assumes the value of capacitance determined using the model and not the parameter.

```
.PROCESS CAPXX = 1
C1 1 2 CAPXX
.MODEL CAPXX C CAP = 1
```
The arguments are defined as:

- **Cxxx**  
  Capacitor element name. Must begin with a “C”, which can be followed by up to 1023 alphanumeric characters.

- **n1**  
  Positive terminal node name

- **n2**  
  Negative terminal node name

- **mname**  
  Capacitance model name. This name is used in elements to reference a capacitor model.

- **C = capacitance**  
  Capacitance at room temperature as a numeric value or parameter in farads.

- **TC1**  
  First-order temperature coefficient for the capacitor. Refer to “Capacitance Temperature Equation” on page 14-11 for temperature-dependent relations.

- **TC2**  
  Second-order temperature coefficient for the capacitor

- **SCALE**  
  Element scale parameter, scales capacitance by its value. Default = 1.0.

- **IC**  
  Initial voltage across the capacitor in volts. This value is used as the DC operating point voltage when UIC is specified in the .TRAN statement and is overridden by the .IC statement.

- **M**  
  Multiplier used to simulate multiple parallel capacitors. Default = 1.0

- **W**  
  Capacitor width in meters. Default = 0.0, if W is not specified in a capacitor model.

- **L**  
  Capacitor length in meters. Default = 0.0, if L is not specified in a capacitor model.

- **DTEMP**  
  Element temperature difference with respect to the circuit temperature in Celsius. Default = 0.0.

- **C = ‘equation’**  
  Capacitance at room temperature specified as a function of any node voltages, branch currents, or any independent variables such as time, frequency (HERTZ), or temperature.
Passive Elements Using Elements

**Example**

In the following example, capacitor C1 is connected from node 1 to node 2 with a capacitance of 20 picofarads:

```
C1 1 2 20p
```

Cshunt refers to three capacitors in parallel connected from node output to ground, each with a capacitance of 100 femtofarads.

```
Cshunt output gnd C = 100f M = 3
```

Capacitor Cload connected from node driver to node output with a capacitance determined by the voltage on node capcontrol times 1E-6, and an initial voltage across the capacitor of 0 volts.

```
Cload driver output C = '1u*v(capcontrol)' CTYPE = 1 IC = 0v
```

Capacitor C99 connected from node in to node out with a capacitance determined by the polynomial \( C = c0 + c1*v + c2*v*v \), where \( v \) is the voltage across the capacitor.

```
C99 in out POLY 2.0 0.5 0.01
```

**Inductors**

The general syntax for including an inductor element in a Star-Hspice netlist is:
General Form:
Lxxx n1 n2 <L = >inductance <<TC1 = >val> <<TC2 = >val>
+ <SCALE = val> <IC = val> <M = val> <DTEMP = val> <R = val>
or
Lxxx n1 n2 L = ‘equation’ <LTYPE = val> <above options...>

Polynomial form:
Lxxx n1 n2 POLY c0 c1... <above options...>

Magnetic Winding form:
Lxxx n1 n2 NT = turns <above options...>

where the inductance can be either a value (in units of henries), an equation, a polynomial of the current or a magnetic winding. The only required fields are the two nodes and the inductance or model name. If the parameter labels are used, the optional arguments may come in any order, although the nodes and model name must come first. If an inductor model is specified (see “Using Passive Device Models” on page 14-1), the inductance value is optional.

The arguments are defined as:

Lxxx Inductor element name. Must begin with L, which can be followed by up to 1023 alphanumeric characters.
n1 Positive terminal node name.
n2 Negative terminal node name.
TC1 First-order temperature coefficient for the inductor. Refer to “Inductor Temperature Equation” on page 14-17 for temperature-dependent relations.
TC2 Second-order temperature coefficient for the inductor.
SCALE Element scale parameter; scales inductance by its value. Default = 1.0.
IC Initial current through the inductor in amperes. This value is used as the DC operating point voltage when UIC is specified in the .TRAN statement and is overridden by the .IC statement.
Passive Elements

Example

In the following example, inductor L1 is connected from node coilin to node coilout with an inductance of 100 nanohenries.

    L1 coilin coilout 100n

Inductor Lloop connected from node 12 to node 17 with an inductance of 1 microhenry, and temperature coefficients of 0.001 and 0.

$L = \text{inductance}$  Inductance value. This may be a numeric value or parameter in henries, or a function of any node voltages, branch currents, or any independent variables such as time, frequency (HERTZ), or temperature.

$M$  Multiplier used to simulate parallel inductors. Default = 1.0.

$DTEMP$  Temperature difference between the element and the circuit in Celsius. Default = 0.0.

$R$  Resistance of inductor in ohms. Default = 0.0.

$L = \text{equation}$  Inductance at room temperature specified as a function of any node voltages, branch currents, or any independent variables such as time, frequency (HERTZ), or temperature.

$LTYPE$  Determines inductance flux calculation for elements with inductance equations. If inductance equation is a function of i(Lxxx), set $LTYPE = 1$. This setting must be used correctly to ensure proper inductance calculations and hence simulation results. Default = 0.

$POLY$  Keyword to specify inductance given by a polynomial.

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

$NT = \text{turns}$  Number representing the number of turns of an inductive magnetic winding.
Using Elements

Passive Elements

Lloop 12 17 L = 1u TC1 = 0.001 TC2 = 0  
Inductor Lcoil connected from node input to ground with an inductance determined by the product of the current through the inductor and 1E-6.  
Lcoil input gnd L = '1u*i(input)' LTYPE = 0  
Inductor L99 connected from node in to node out with an inductance determined by the polynomial \( L = c0 + c1*i + c2*i*i \), where \( i \) is the current through the inductor. The inductor is also specified to have a DC resistance of 10 ohms.  
L99 in out POLY 4.0 0.35 0.01 R = 10  
Inductor L connected from node 1 to node 2 as a magnetic winding element with 10 turns of wire.  
L 1 2 NT = 10

Mutual Inductors

The general syntax for including a mutual inductor element in a Star-Hspice netlist is:

**General form:**

\[ \text{Kxxx Lyyy Lzzz } \text{<K = >coupling} \]

**Mutual Core form:**

\[ \text{Kaaa Lbbb } \text{<Lccc ... <Lddd>> mname <MAG = magnetization>} \]

where “coupling” is a unitless value from zero to one representing the coupling strength. If the parameter labels are used, the optional arguments may come in any order, although the nodes and model name must come first. If an inductor model is specified (see “Using Passive Device Models” on page 14-1), the inductance value is optional.

The arguments are defined as:

**Kxxx**

Mutual inductor element name. Must begin with “K”, which can be followed by up to 1023 alphanumeric characters.

**Lyyy**

Name of the first of two coupled inductors.
The coupling coefficient should be determined by the user based on any geometric and spatial information known. The final coupling inductance will be determined by dividing the coupling coefficient by the square-root of the product of the two self-inductances.

When using the mutual inductor element to calculate the coupling between more than two inductors, Star-Hspice can automatically calculate an approximate second-order coupling. See the third example below for a specific situation.

**Warning:** The automatic inductance calculation is only an estimation and is accurate for a subset of geometries. The second-order coupling coefficient is simply the product of the two first-order coefficients, which is not correct for many geometries.
Example

Inductors Lin and Lout are coupled with a coefficient of 0.9.

\[
K_1 \text{ Lin} \text{ Lout} 0.9
\]

Inductors Lhigh and Llow are coupled with a coefficient equal to the value of the parameter COUPLE.

\[
K_{xfmr} \text{ Lhigh} \text{ Llow} K = \text{COUPLE}
\]

The two mutual inductors K1 and K2 couple L1 and L2, and L2 and L3, respectively. The coupling coefficients are 0.98 and 0.87. Star-Hspice automatically calculates the mutual inductance between L1 and L3, with a coefficient of 0.98*0.87 = 0.853.

\[
K_1 \text{ L1} \text{ L2} 0.98
\]

\[
K_2 \text{ L2} \text{ L3} 0.87
\]
Active Elements

Diode Element

The general syntax for including a diode element in a Star-Hspice netlist is:

Geometric (**LEVEL = 1**) and Non-geometric (**LEVEL = 3**) form:

```plaintext
Dxxx nplus nminus mname <<AREA = >area> <<PJ = >val> 
<P = val>
+ <LP = val> <WM = val> <LM = val> <OFF> <IC = vd> <M = val>
+ <DTEMP = val>
```

or

```plaintext
Dxxx nplus nminus mname <W = width> <L = length> <WP = val>
+ <LP = val> <WM = val> <LM = val> <OFF> <IC = vd> <M = val>
+ <DTEMP = val>
```

Fowler-Nordheim (**LEVEL = 2**) form:

```plaintext
Dxxx nplus nminus mname <W = val <L = val>> <WP = val> <OFF>
+ <IC = vd> <M = val>
```

The only required fields are the two nodes and the model name. If the parameter labels are used, the optional arguments may come in any order, although the nodes and model name must come first.

The arguments are as follows:

- **Dxxx**: Diode element name. Must begin with “D”, which can be followed by up to 1023 alphanumeric characters.
- **nplus**: Positive terminal (anode) node name. The series resistor of the equivalent circuit is attached to this terminal.
- **nminus**: Negative terminal (cathode) node name
- **mname**: Diode model name reference
### Using Elements

#### AREA
Area of the diode (unitless for diode model LEVEL = 1 and square meters for diode model LEVEL = 3). This affects saturation currents, capacitances and resistances (diode model parameters IK, IKR, JS, CJO and RS). Area factor for diode model LEVEL = 1 is not affected by the SCALE option. Default = 1.0. Overrides AREA from the diode model. If unspecified, is calculated from width and length specifications.

#### PJ
Periphery of junction (unitless for diode model LEVEL = 1 and meters for diode model LEVEL = 3). Overrides PJ from the diode model. If unspecified, calculated from the width and length specifications.

#### WP
Width of polysilicon capacitor in meters (for diode model LEVEL = 3 only). Overrides WP in diode model. Default = 0.0.

#### LP
Length of polysilicon capacitor in meters (for diode model LEVEL = 3 only). Overrides LP in diode model. Default = 0.0.

#### WM
Width of metal capacitor in meters (for diode model LEVEL = 3 only). Overrides WM in diode model. Default = 0.0.

#### LM
Width of metal capacitor in meters (for diode model LEVEL = 3 only). Overrides LM in diode model. Default = 0.0.

#### OFF
Sets initial condition to OFF for this element in DC analysis. Default = ON.

#### IC = vd
Initial voltage across the diode element. This value is used when the UIC option is present in the .TRAN statement and is overridden by the .IC statement.

#### M
Multiplier to simulate multiple diodes in parallel. All currents, capacitances and resistances are affected by the setting of M. Default = 1.

#### DTEMP
The difference between the element temperature and the circuit temperature in Celsius. Default = 0.0.
Active Elements

Example
Diode D1 with anode and cathode connected to nodes 1 and 2 where the diode model is given by diode1.

\[ \text{D1 1 2 diode1} \]

Diode Dprot with anode and cathode connected to node output and ground references diode model firstd and specifies an area of 10 (unitless for LEVEL = 1 model) with the diode OFF as an initial condition.

\[ \text{Dprot output gnd firstd 10 OFF} \]

Diode Ddrive with anode and cathode connected to nodes driver and output with a width and length of 500 microns and references diode model model_d.

\[ \text{Ddrive driver output model_d W = 5e-4 L = 5e-4 IC = 0.2} \]

Bipolar Junction Transistors (BJTs) Element
The general syntax for including a BJT element in a Star-Hspice netlist is:

General form:

\[ \text{Qxxx nc nb ne <ns> mname <area> <OFF> IC = vbeval,vceval} + \text{ <M = val> <DTEMP = val>} \]

or

\[ \text{Qxxx nc nb ne <ns> mname AREA = area <AREAB = val> + <AREAC = val> <OFF> <VBE = vbeval> <VCE = vceval> <M = val> + <DTEMP = val>} \]

The only required fields are the collector, base and emitter nodes, and the model name. The nodes and model name must come first.
Using Elements

The arguments are as follows:

- **Qxxx**: BJT element name. Must begin with “Q”, which can be followed by up to 1023 alphanumeric characters.
- **nc**: Collector terminal node name
- **nb**: Base terminal node name
- **ne**: Emitter terminal node name
- **ns**: Substrate terminal node name, which is optional. Can also be set in the BJT model with the parameter BULK.
- **mname**: BJT model name reference
- **area**, **AREA = area**: Emitter area multiplying factor which affects currents, resistances and capacitances. Default = 1.0.
- **OFF**: Sets initial condition to OFF for this element in DC analysis. Default = ON.
- **IC = vbeval, vceval, VBE, VCE**: Initial internal base-emitter voltage (vbeval) and collector-emitter voltage (vceval). These are used when UIC is present in the .TRAN statement and is overridden by the .IC statement.
- **M**: Multiplier to simulate multiple BJTs in parallel. All currents, capacitances and resistances are affected by the setting of M. Default = 1.
- **DTEMP**: The difference between the element temperature and the circuit temperature in Celsius. Default = 0.0.
- **AREAB**: Base area multiplying factor that affects currents, resistances and capacitances. Default = AREA.
- **AREAC**: Collector area multiplying factor that affects currents, resistances and capacitances. Default = AREA.

**Example**

BJT Q1 with collector, base, and emitter connected to nodes 1, 2 and 3 where the BJT model is given by model_1.
BJT Qopamp1 with collector, base, and emitter connected to nodes c1, b3 and e2 and the substrate connected to node s. The BJT model is given by 1stagepnp and the area factors AREA, AREAB and AREAC are 1.5, 2.5 and 3.0, respectively.

Qopamp1 c1 b3 e2 s 1stagepnp AREA = 1.5 AREAB = 2.5 AREAC = 3.0

BJT Qdrive with collector, base, and emitter connected to nodes driver, in and output with an area factor of 0.1 and references BJT model model_npn.

Qdrive driver in output model_npn 0.1

JFETs and MESFETs

The general syntax for including a JFET or MESFET element in a Star-Hspice netlist is:

**General form:**

\[
\text{Jxxx} \ nd \ ng \ ns <nb> \ mname \ <<\text{AREA} = \text{area} \ | \ <W = \text{val}> \ <L = \text{val}></br>
+ <OFF> \ <IC = \text{vdsval,vgsvval} > \ <M = \text{val} > \ <DTEMP = \text{val} >
\]

or

\[
\text{Jxxx} \ nd \ ng \ ns <nb> \ mname \ <<\text{AREA} = \text{area} > \ | \ <W = \text{val}> \ <L = \text{val}></br>
+ <OFF> \ <VDS = \text{vdsval} > \ <VGS = \text{vgsvval} > \ <M = \text{val} > \ <DTEMP = \text{val} >
\]

The only required fields are the drain, gate and source nodes, and the model name. The nodes and model name must come first.

The arguments are as follows:

- **Jxxx** JFET or MESFET element name. Must begin with “J”, which can be followed by up to 1023 alphanumeric characters.
- **nd** Drain terminal node name
Example

JFET J1 with drain, source, and gate connected to nodes 1, 2 and 3 where the JFET model is given by model_1.

```
J1 1 2 3 model_1
```

JFET Jopamp1 with drain, gate, and source connected to nodes d1, g3 and s2 and the bulk connected to node b. The JFET model is given by 1stage and the area is given as 100 microns.

```
Jopamp1 d1 g3 s2 b 1stage AREA = 100u
```
Active Elements

Using Elements

JFET Jdrive with drain, gate, and source connected to nodes driver, in and output with a width and length of 10 microns and references JFET model model_jfet.

Jdrive driver in output model_jfet W = 10u L = 10u

MOSFETs

The general syntax for including a MOSFET element in a Star-Hspice netlist is:

General form:

Mxxx nd ng ns <nb> mname <<L = >length> <<W = >width>
<AD = val>
+ <AS = val> <PD = val> <PS = val> <NRD = val> <NRS = val>
+ <RDC = val> <RSC = val> <OFF> <IC = vds,vgs,vbs> <M = val>
+ <DTEMP = val> <GEO = val> <DELVTO = val>

or

.OPTION WL

Mxxx nd ng ns <nb> mname <width> <length> <other options...>

The only required fields are the drain, gate and source nodes, and the model name. The nodes and model name must come first. The second syntax is used in conjunction with the .OPTION WL statement that allows exchanging the width and length options when no label is given.

The arguments are as follows:

Mxxx MOSFET element name. Must begin with “M”, which can be followed by up to 1023 alphanumeric characters.

nd Drain terminal node name

ng Gate terminal node name

ns Source terminal node name

nb Bulk terminal node name, which is optional. Can be set in MOSFET model using parameter BULK.

mname MOSFET model name reference
### MOSFET Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L</strong></td>
<td>MOSFET channel length in meters. This parameter overrides DEFL in an OPTIONS statement. Default = DEFL with a maximum of 0.1m.</td>
</tr>
<tr>
<td><strong>W</strong></td>
<td>MOSFET channel width in meters. This parameter overrides DEFW in an OPTIONS statement. Default = DEFW.</td>
</tr>
<tr>
<td><strong>AD</strong></td>
<td>Drain diffusion area. Overrides DEFAD in the OPTIONS statement. Default = DEFAD only when the MOSFET model parameter ACM = 0.</td>
</tr>
<tr>
<td><strong>AS</strong></td>
<td>Source diffusion area. Overrides DEFAS in the OPTIONS statement. Default = DEFAS only when the MOSFET model parameter ACM = 0.</td>
</tr>
<tr>
<td><strong>PD</strong></td>
<td>Perimeter of the drain junction, including the channel edge. Overrides DEFPD in the OPTIONS statement. Default = DEFAD when the MOSFET model parameter ACM = 0 or 1, and default = 0.0 when ACM = 2 or 3.</td>
</tr>
<tr>
<td><strong>PS</strong></td>
<td>Perimeter of the source junction, including the channel edge. Overrides DEFPS in the OPTIONS statement. Default = DEFAS when the MOSFET model parameter ACM = 0 or 1, and default = 0.0 when ACM = 2 or 3.</td>
</tr>
<tr>
<td><strong>NRD</strong></td>
<td>Number of squares of drain diffusion for resistance calculations. Overrides DEFNRD in the OPTIONS statement. Default = DEFNRD when the MOSFET model parameter ACM = 0 or 1, and default = 0.0 when ACM = 2 or 3.</td>
</tr>
<tr>
<td><strong>NRS</strong></td>
<td>Number of squares of source diffusion for resistance calculations. Overrides DEFNRS in the OPTIONS statement. Default = DEFNRS when the MOSFET model parameter ACM = 0 or 1, and default = 0.0 when ACM = 2 or 3.</td>
</tr>
<tr>
<td><strong>RDC</strong></td>
<td>Additional drain resistance due to contact resistance with units of ohms. This value overrides the RDC setting in the MOSFET model specification. Default = 0.0.</td>
</tr>
</tbody>
</table>
Active Elements Using Elements

RSC
Additional source resistance due to contact resistance with units of ohms. This value overrides the RSC setting in the MOSFET model specification. Default = 0.0.

OFF
Sets initial condition to OFF for this element in DC analysis. Default = ON.
Note: this command does not work for depletion devices.

IC = vds, vgs, vbs
Initial voltage across the external drain and source (vds), gate and source (vgs), and bulk and source terminals (vbs). These are used when UIC is present in the .TRAN statement and are overridden by the .IC statement.

M
Multiplier to simulate multiple MOSFETs in parallel. All channel widths, diode leakages, capacitances and resistances are affected by the setting of M. Default = 1.

DTEMP
The difference between the element temperature and the circuit temperature in Celsius. Default = 0.0.

GEO
Source/drain sharing selector for MOSFET model parameter value ACM = 3. Default = 0.0.

DELVTO
Zero-bias threshold voltage shift. Default = 0.0.

Example
MOSFET M1 with drain, gate, and source connected to nodes 1, 2 and 3 where the MOSFET model is given by model_1.

\[ M 1 \ 2 \ 3 \ \text{model}_1 \]

MOSFET Mopamp1 with drain, gate, and source connected to nodes d1, g3 and s2 and the bulk connected to node b. The MOSFET model is given by 1stage and the length and width of the gate are given as 2 and 10 microns, respectively.

\[ \text{Mopamp1} \ d1 \ g3 \ s2 \ b \ \text{1stage} \ \text{L} = 2\mu \ \text{W} = 10\mu \]

MOSFET Mdrive with drain, gate, and source connected to nodes driver, in and output with a width and length of 3 and 0.25 microns, respectively. This device references MOSFET model bsim3v3 and specifies a temperature for the device that is 4 degrees Celsius above the circuit temperature.

\[ \text{Mdrive} \ \text{driver} \ \text{in} \ \text{output} \ \text{bsim3v3} \ \text{W} = 3\mu \ \text{L} = 0.25\mu \ \text{DTEMP} = 4.0 \]
Transmission Lines

W Element Statement

The general syntax for including a lossy (W Element) transmission line element in a Star-Hspice netlist is:

**RLGC file form:**

```
Wxxx in1 <in2 <...inx>> refin out1 <out2 <...outx>> refout
+ <RLGCfile = fname> N = val L = val
```

**U-model form:**

```
Wxxx in1 <in2 <...inx>> refin out1 <out2 <...outx>> refout
+ <Umodel = mname> N = val L = val
```

**Field Solver form:**

```
Wxxx in1 <in2 <...inx>> refin out1 <out2 <...outx>> refout
+ <FSmodel = mname> N = val L = val
```

where the number of ports on a single transmission line are not limited. One input and output port, the ground references, a model or file reference, a number of conductors and a length are all required.

The arguments are defined as:

- **Wxxx** Lossy (W Element) transmission line element name. Must begin with a “W”, which can be followed by up to 1023 alphanumeric characters.
- **inx** Signal input node for the x\textsuperscript{th} transmission line (in1 is required).
- **refin** Ground reference for input signal
- **outx** Signal output node for the x\textsuperscript{th} transmission line (each input port must have a corresponding output port).
- **refout** Ground reference for output signal.
Transmission Lines

Example

Lossy transmission line W1 connected from node in to node out with both signal references grounded, using the RLGC file named cable.rlgc and length of 5 meters.

\[ W1 \text{ in gnd out gnd RLGCfile = cable.rlgc N = 1 L = 5 } \]

Two-conductor lossy transmission line Wcable is connected from nodes in1 and in2 to out1 and out2 with grounds on both signal references. References the U-model named umod_1 and is 10 meters in length.

\[ Wcable \text{ in1 in2 gnd out1 out2 gnd Umodel = umod_1 N = 2 L = 10 } \]

Five-conductor lossy transmission line Wnet1 connected from nodes i1, i2, i3, i4 and i5 to nodes o1, o3, o5 and the second and fourth outputs grounded with both signal references grounded as well. References the Field Solver model named board1 and is 1 millimeter long.

\[ Wnet1 \text{ i1 i2 i3 i4 i5 gnd o1 gnd o3 gnd o5 gnd FSmodel = board1 } \]
\[ + \text{ N = 5 L = 1m } \]
The order of parameters in the W-element card does not matter and the number of signal conductors, \( N \), can be specified after the list of nodes. Moreover, the nodes and parameters in the W-element card can be mixed freely.

Only one of the RLGC file, FS model, or U model models can be specified in a single W-element card.

Figure 4-1 shows the node numbering convention for the element syntax.

**Figure 4-1: Terminal Node Numbering for W Element**

**T Element Statement**

The general syntax for including a lossless (T Element) transmission line element in a Star-Hspice netlist is:

*General form:*

\[
\text{Txxx in refin out refout Z0 = val TD = val } < \text{L = val} > \\
< \text{IC = v1,i1,v2,i2} >
\]

or

\[
\text{Txxx in refin out refout Z0 = val F = val } < \text{NL = val} > \\
< \text{IC = v1,i1,v2,i2} >
\]

*U-model form:*

\[
\text{Txxx in refin out refout mname L = val}
\]

where only one input and output port is allowed.
The arguments are defined as:

- **Txxx**: Lossless transmission line element name. Must begin with a “T”, which can be followed by up to 1023 alphanumeric characters.
- **in**: Signal input node
- **refin**: Ground reference for input signal
- **out**: Signal output node
- **refout**: Ground reference for output signal
- **Z0**: Characteristic impedance of the transmission line
- **TD**: Signal delay from the transmission line in units of seconds per meter
- **L**: Physical length of the transmission line in units of meters. Default = 1.
- **IC = v1,i1,v2,i2**: Initial conditions of the transmission line. Specify the voltage on the input port (v1), current into the input port (i1), voltage on the output port (v2) and the current into the output port (i2).
- **F**: Frequency at which the transmission line has the electrical length given by NL.
- **NL**: Normalized electrical length of the transmission line at the frequency, specified in the F parameter, in units of wavelengths per line length. Default = 0.25, which corresponds to a quarter-wavelength.
- **mname**: U-model reference name. A lossy transmission line model, used here to represent the characteristics of the lossless transmission line.

**Example**

Transmission line T1 connected from node in to node out with both signal references grounded, with a 50 ohm impedance and a 5 nanosecond per meter transmission delay and a length of 5 meters.
Using Elements Transmission Lines

T1 in gnd out gnd Z0 = 50 TD = 5n L = 5

Transmission line Tcable is connected from node in1 to out1 with grounds on both signal references, a 100 ohm impedance, and a normalized electrical length of 1 wavelength at 100 kHz.

Tcable in1 gnd out1 gnd Z0 = 100 F = 100k NL = 1

Transmission line Tnet1 connected from node driver to node output with both signal references grounded. References the U-model named Umodel1 and is 1 millimeter long.

Tnet1 driver gnd output gnd Umodel1 L = 1m

U Element Statement

The general syntax for including a lossy (U Element) transmission line element in a Star-Hspice netlist is:

**General form:**

Uxxx in1 <in2 <...in5>> refin out1 <out2 <...out5>> refout mname
+ L = val <LUMPS = val>

where the number of ports on a single transmission line are limited to five in and five out. One input and output port, the ground references, a model reference and a length are all required.

The arguments are defined as:

- **Uxxx**
  Lossy (U Element) transmission line element name. Must begin with a “U”, which can be followed by up to 1023 alphanumeric characters.

- **inx**
  Signal input node for the x\(^{th}\) transmission line (in1 is required).

- **refin**
  Ground reference for input signal

- **outx**
  Signal output node for the x\(^{th}\) transmission line (each input port must have a corresponding output port).

- **refout**
  Ground reference for output signal

- **mname**
  U-model lossy transmission-line model reference name
Transmission Lines

$L$  
Physical length of the transmission line in units of meters.

$LUMPS$  
Number of lumped-parameter sections used in the simulation of this element.

Example

Lossy transmission line $U1$ connected from node $in$ to node $out$ with both signal references grounded, using the U-model named $umodel\_RG58$ and length of 5 meters.

$U1 \ in \ gnd \ out \ gnd \ umodel\_RG58 \ L = 5$

Two-conductor lossy transmission line $Ucable$ is connected from nodes $in1$ and $in2$ to $out1$ and $out2$ with grounds on both signal references. References the U-model named $twistpr$ and is 10 meters in length.

$Ucable \ in1 \ in2 \ gnd \ out1 \ out2 \ gnd \ twistpr \ L = 10$

Five-conductor lossy transmission line $Unet1$ connected from nodes $i1$, $i2$, $i3$, $i4$ and $i5$ to nodes $o1$, $o3$, $o5$ and the second and fourth outputs grounded with both signal references grounded as well. References the U-model named $Umodel1$ and is 1 millimeter long.

$Unet1 \ i1 \ i2 \ i3 \ i4 \ i5 \ gnd \ o1 \ gnd \ o3 \ gnd \ o5 \ gnd \ Umodel1 \ L = 1m$
Buffers

The general syntax of an element card for input/output buffers is:

**General Form**

```
bname node_1 node_2 ... node_N keyword_1 = value_1
...
+ [keyword_M = value_M]
```

where:

- `bname` is the buffer name and starts with the letter B.
- `node_1 node_2 ... node_N` is a list of input/output buffer external nodes. The number of nodes and their meaning are specific to different buffer types.
- `keyword_i = value_i` assigns value `value_i` to the keyword `keyword_i`. Optional keywords are given in square brackets.

See “Using IBIS Models” on page 19-1 for information on the keywords.

**Example**

```
B1 nd_pc nd_gc nd_in nd_out_of_in
  + buffer = 1
  + file = 'test.ibs'
  + model = 'IBIS_IN'
```

This example represents an input buffer, B1, with the 4 terminals `nd_pc`, `nd_gc`, `nd_in` and `nd_out_of_in`. The IBIS model `IBIS_IN` is located in the IBIS file named `test.ibs`. Note that nodes `nd_pc` and `nd_gc` are connected by Star-Hspice to the voltage sources. Therefore, users should not connect these nodes to voltage sources. See “Using IBIS Models” on page 19-1 for more examples.
Chapter 5
Using Sources and Stimuli

This chapter describes element and model statements for independent sources, dependent sources, analog-to-digital elements, and digital-to-analog elements. It also provides explanations of each type of element and model statement. Explicit formulas and examples show how various combinations of parameters affect the simulation.

The chapter covers the following topics:

- Independent Source Elements
- Star-Hspice Independent Source Functions
- Using Voltage and Current Controlled Elements
- Voltage Dependent Voltage Sources — E Elements
- Voltage Dependent Current Sources — G Elements
- Dependent Voltage Sources — H Elements
- Current Dependent Current Sources — F Elements
- Digital and Mixed Mode Stimuli
Independent Source Elements

Use independent source element statements to specify DC, AC, transient, and mixed independent voltage and current sources. Some types of analysis use the associated analysis sources. For example, in a DC analysis, if both DC and AC sources are specified in one independent source element statement, the AC source is taken out of the circuit for the DC analysis. If an independent source is specified for an AC, transient, and DC analysis, transient sources are removed for the AC analysis and DC sources are removed after the performance of the operating point. Initial transient value always overrides the DC value.

Source Element Conventions

Voltage sources need not be grounded. Positive current is assumed to flow from the positive node through the source to the negative node. A positive current source forces current to flow out of the N+ node through the source and into the N- node.

You can use parameters as values in independent sources. Do not identify these parameters using any of the following reserved keywords:

- AC
- ACI
- AM
- DC
- EXP
- PE
- PL
- PU
- PULSE
- PWL
- R
- RD
- SFFM
- SIN

Independent Source Element

The general syntax for including an independent source in a Star-Hspice netlist is:

**General Form**

\[
V_{xxx} \, n+ \, n- \, \langle\langle DC \rangle\rangle \, dcval \, \langle\langle tranfun \rangle\rangle \, \langle\langle AC=acmag, \, acphase\rangle\rangle \\
or \\
I_{yy} \, n+ \, n- \, \langle\langle DC \rangle\rangle \, dcval \, \langle\langle tranfun \rangle\rangle \, \langle\langle AC=acmag, \, acphase\rangle\rangle \\
+ \, \langle\langle M=val \rangle\rangle 
\]
The arguments are defined as follows:

- **Vxxx**: Independent voltage source element name. Must begin with a “V”, which can be followed by up to 1023 alphanumeric characters.

- **Iyyy**: Independent current source element name. Must begin with an “I”, which can be followed by up to 1023 alphanumeric characters.

- **n+**: Positive node

- **n-**: Negative node

- **DC=dcval**: DC source keyword and value in volts. The “tranfun” value at time zero overrides the DC value. Default=0.0.

- **tranfun**: Transient source function (one or more of: AM, DC, EXP, PE, PL, PU, PULSE, PWL, SFFM, SIN). The functions specify the characteristics of a time-varying source. See the individual functions for syntax.

- **AC**: AC source keyword for use in AC small-signal analysis

- **acmag**: Magnitude (RMS) of the AC source in volts

- **acphase**: Phase of the AC source in degrees. Default=0.0.

- **M**: Multiplier used for simulating multiple parallel current sources. The source current value is multiplied by M. Default=1.0.

**Example**

Voltage source VX has a 5 volt DC bias, and the positive terminal is connected to node 1 while the negative terminal is grounded.

```
VX 1 0 5V
```
Voltage source VB has a DC bias specified by the parameter ‘VCC’, and the positive terminal is connected to node 2 while the negative terminal is grounded.

\[ \text{VB 2 0 DC=VCC} \]

Voltage source VH has a 2 volt DC bias, a 1 volt RMS AC bias with 90 degree phase offset, and the positive terminal is connected to node 3 while the negative terminal is connected to node 6.

\[ \text{VH 3 6 DC=2 AC=1, 90} \]

Current source IG has a time-varying response given by the piecewise-linear relationship with 1 milliamp at time=0 and 5 milliamps at 25 milliseconds, and the positive terminal is connected to node 8 while the negative terminal is connected to node 7.

\[ \text{IG 8 7 PL(1MA 0S 5MA 25MS)} \]

Voltage source VCC has a DC bias specified by the parameter ‘VCC’, and a time-varying response given by the piecewise-linear relationship with 0 volts at time=0, ‘VCC’ from 10 to 15 nanoseconds and back to 0 volts at 20 nanoseconds. The positive terminal is connected to node in while the negative terminal is connected to node out. The operating point for this source will be determined without the DC value (that is, it will be 0 volts).

\[ \text{VCC in out VCC PWL 0 0 10NS VCC 15NS VCC 20NS 0} \]

Voltage source VIN has a 0.001 volt DC bias, a 1 volt RMS AC bias, and a sinusoidal time-varying response from 0 to 1 volts with a frequency of 1 megahertz. The positive terminal is connected to node 13 while the negative terminal is connected to node 2.

\[ \text{VIN 13 2 0.001 AC 1 SIN (0 1 1MEG)} \]

Current source ISRC has a 1/3 amp RMS AC response with a 45-degree phase offset and a frequency modulated time-varying response with variation from 0 to 1 volts, a carrier frequency of 10 kHz, a signal frequency of 1 kHz and a modulation index of 5. The positive terminal is connected to node 23 while the negative terminal is connected to node 21.

\[ \text{ISRC 23 21 AC 0.333 45.0 SFFM (0 1 10K 5 1K)} \]
Voltage source VMEAS has a 0 volt DC bias, and the positive terminal is connected to node 12 while the negative terminal is connected to node 9.

VMEAS 12 9

**DC Sources**

For a DC source, you can specify the DC current or voltage in different ways:

V1 1 0 DC=5V  
V1 1 0 5V  
I1 1 0 DC=5mA  
I1 1 0 5mA

The first two examples specify a DC voltage source of 5 V connected between node 1 and ground. The third and fourth examples specify a 5 mA DC current source between node 1 and ground. The direction of current in both sources is from node 1 to ground.

**AC Sources**

AC current and voltage sources are impulse functions used for an AC analysis. Specify the magnitude and phase of the impulse with the AC keyword.

V1 1 0 AC=10V,90  
VIN 1 0 AC 10V 90

The above two examples specify an AC voltage source with a magnitude of 10 V and a phase of 90 degrees. Specify the frequency sweep range of the AC analysis in the .AC analysis statement. The AC or frequency domain analysis provides the impulse response of the circuit.

**Transient Sources**

For transient analysis, you can specify the source as a function of time. The functions available are pulse, exponential, damped sinusoidal, single frequency FM, and piecewise linear function.
Mixed Sources

Mixed sources specify source values for more than one type of analysis. For example, you can specify a DC source specified together with an AC source and transient source, all of which are connected to the same nodes. In this case, when specific analyses are run, Star-Hspice selects the appropriate DC, AC, or transient source. The exception is the zero-time value of a transient source, which overrides the DC value, and is selected for operating-point calculation for all analyses.

VIN 13 2 0.5 AC 1 SIN (0 1 1MEG)

The above example specifies a DC source of 0.5 V, an AC source of 1 V, and a transient damped sinusoidal source, each of which are connected between nodes 13 and 2. For DC analysis, the program uses zero source value since the sinusoidal source is zero at time zero.
Star-Hspice Independent Source Functions

Star-Hspice provides the following types of independent source functions:

- Pulse (PULSE function)
- Sinusoidal (SIN function)
- Exponential (EXP function)
- Piecewise linear (PWL function)
- Single-frequency FM (SFFM function)
- Single-frequency AM (AM function)

PWL also comes in a data driven version. The data driven PWL allows the results of an experiment or a previous simulation to provide one or more input sources for a transient simulation.

The independent sources supplied with Star-Hspice permit the designer to specify a variety of useful analog and digital test vectors for either steady state, time domain, or frequency domain analysis. For example, in the time domain, both current and voltage transient waveforms can be specified as exponential, sinusoidal, piecewise linear, single-sided FM functions, or AM functions.

Pulse Source Function

Star-Hspice has a trapezoidal pulse source function, which starts with an initial delay from the beginning of the transient simulation interval to an onset ramp. During the onset ramp, the voltage or current changes linearly from its initial value to the pulse plateau value. After the pulse plateau, the voltage or current moves linearly along a recovery ramp, back to its initial value. The entire pulse repeats with a period \( \text{per} \) from onset to onset.

The general syntax for including a pulse source in an independent voltage or current source is:

**General form:**

\[
\begin{align*}
Vxxx \ n+ \ n- \ \text{PU}<\text{LSE}> \ \langle(v1 \ v2 \ \text{td} \ \text{tr} \ \text{tf} \ \text{pw} \ \text{per}>>>> \ \rangle > \\
\text{or} \\
Ixxx \ n+ \ n- \ \text{PU}<\text{LSE}> \ \langle(v1 \ v2 \ \text{td} \ \text{tr} \ \text{tf} \ \text{pw} \ \text{per}>>>> \ \rangle >
\end{align*}
\]
The arguments are defined as:

\[ V_{xxx}, I_{xxx} \]

Independent voltage source that will exhibit the pulse response.

**PULSE**

Keyword for a pulsed time-varying source. The short form is ‘PU’.

\[ v1 \]

Initial value of the voltage or current, before the pulse onset (units of volts or amps).

\[ v2 \]

Pulse plateau value (units of volts or amps).

\[ td \]

Delay time in seconds from the beginning of transient interval to the first onset ramp. Default=0.0 and negative values are considered as zero.

\[ tr \]

Duration of the onset ramp in seconds, from the initial value to the pulse plateau value (reverse transit time). Default=TSTEP.

\[ tf \]

Duration of the recovery ramp in seconds, from the pulse plateau back to the initial value (forward transit time). Default=TSTEP.

\[ pw \]

Pulse width (the width of the plateau portion of the pulse) in seconds. Default=TSTEP.

\[ per \]

Pulse repetition period in seconds. Default=TSTEP.

Below is a table showing the time-value relationship for a PULSE source:

<table>
<thead>
<tr>
<th>Time</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[ v1 ]</td>
</tr>
<tr>
<td>td</td>
<td>[ v1 ]</td>
</tr>
<tr>
<td>td + tr</td>
<td>[ v2 ]</td>
</tr>
<tr>
<td>td + tr + pw</td>
<td>[ v2 ]</td>
</tr>
<tr>
<td>td + tr + pw + tf</td>
<td>[ v1 ]</td>
</tr>
<tr>
<td>tstop</td>
<td>[ v1 ]</td>
</tr>
</tbody>
</table>
Intermediate points are determined by linear interpolation.

**Note:** TSTEP is the printing increment, and TSTOP is the final time.

**Example**

The following example shows the pulse source connected between node 3 and node 0. The pulse has an output high voltage of 1 V, an output low voltage of -1 V, a delay of 2 ns, a rise and fall time of 2 ns, a high pulse width of 50 ns, and a period of 100 ns.

\[ V_{IN} 3 \ 0 \ \text{PULSE} \ (-1 \ 1 \ 2\text{NS} \ 2\text{NS} \ 2\text{NS} \ 50\text{NS} \ 100\text{NS}) \]

Pulse source connected between node 99 and node 0. The syntax shows parameterized values for all the specifications.

\[ V1 \ 99 \ 0 \ \text{PUL} \ \text{lv} \ \text{hv} \ \text{tdlay} \ \text{tris} \ \text{tfall} \ \text{tpw} \ \text{tper} \]

This example shows an entire Star-Hspice netlist, which contains a PULSE voltage source. The source has an initial voltage of 1 volt, a pulse voltage of 2 volts, a delay time, rise time and fall time each of 5 nanoseconds, a pulse width of 20 nanoseconds, and a pulse period of 50 nanoseconds. The result of the simulation of this netlist is shown in Figure 5-1.

```
File pulse.sp test of pulse
.option post
.tran .5ns 75ns
vpulse 1 0 pulse( v1 v2 td tr tf pw per )
r1 1 0 1
.param v1=1v v2=2v td=5ns tr=5ns tf=5ns pw=20ns +per=50ns
.end
```
Sinusoidal Source Function

Star-Hspice has a damped sinusoidal source that is the product of a dying exponential with a sine wave. Application of this waveform requires the specification of the sine wave frequency, the exponential decay constant, the beginning phase, and the beginning time of the waveform, as explained below.

The general syntax for including a sinusoidal source in an independent voltage or current source is:

**General Form:**

\[
\begin{align*}
V_{xxx} n+ n- & \text{ SIN } (< vo va < freq < td < \theta < \phi >>> < >) \\
\text{or} \\
I_{xxx} n+ n- & \text{ SIN } (< vo va < freq < td < \theta < \phi >>> < >)
\end{align*}
\]
Using Sources and Stimuli

Star-Hspice Independent Source Functions

The arguments are defined as:

- **Vxxx, Ixxx**: Independent voltage source that will exhibit the sinusoidal response.
- **SIN**: Keyword for a sinusoidal time-varying source
- **vo**: Voltage or current offset in volts or amps
- **va**: Voltage or current RMS amplitude in volts or amps
- **freq**: Source frequency in Hz. Default=1/TSTOP.
- **td**: Time delay before beginning the sinusoidal variation in seconds. Default=0.0, response will be 0 volts or amps until the delay value is reached, even with a non-zero DC voltage.
- **θ**: Damping factor in units of 1/seconds. Default=0.0.
- **φ**: Phase delay in units of degrees. Default=0.0.

The waveform shape is given by the following table of expressions:

<table>
<thead>
<tr>
<th>Time</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to td</td>
<td>$vo + va \cdot \sin\left(\frac{2 \cdot \pi \cdot \phi}{360}\right)$</td>
</tr>
<tr>
<td></td>
<td>$vo + va \cdot \exp\left[-\left(Time - td\right) \cdot \theta\right] \cdot$</td>
</tr>
<tr>
<td></td>
<td>$\sin\left{2 \cdot \pi \cdot \left[\text{freq} \cdot (time - td) + \frac{\phi}{360}\right]\right}$</td>
</tr>
</tbody>
</table>

where TSTOP is the final time; see the .TRAN statement for a detailed explanation.

**Example**

```
VIN 3 0 SIN (0 1 100MEG 1NS 1e10)
```
Damped sinusoidal source connected between nodes 3 and 0. The waveform has a peak value of 1 V, an offset of 0 V, a 100-MHz frequency, a time delay of 1 ns, a damping factor of 1e10, and a phase delay of zero degree. See Figure 5-2 for a plot of the source output.

**Figure 5-2: Sinusoidal Source Function**

*File: SIN.SP THE SINUSOIDAL WAVEFORM
*<decay envelope>
.OPTIONS POST
.PARAM V0=0 VA=1 FREQ=100MEG DELAY=2N THETA=5E7 +PHASE=0
V 1 0 SIN (V0 VA FREQ DELAY THETA PHASE)
R 1 0 1
.TRAN .05N 50N
.END

This example shows an entire Star-Hspice netlist that contains a SIN voltage source. The source has an initial voltage of 0 volts, a pulse voltage of 1 volt, a delay time of 2 nanoseconds, a frequency of 100 MHz, and a damping factor of 50 MHz.
Exponential Source Function

The general syntax for including an exponential source in an independent voltage or current source is:

**General Form:**

\[ \text{Vxxx n+ n- EXP (v1 v2 <td1 <td2 <τ1 <τ2>>>)} \]

or

\[ \text{Ixxx n+ n- EXP (v1 v2 <td1 <td2 <τ1 <τ2>>>)} \]

The arguments are defined as:

- **Vxxx, Ixxx**: Independent voltage source that will exhibit the exponential response.
- **EXP**: Keyword for a exponential time-varying source
- **v1**: Initial value of voltage or current in volts or amps
- **v2**: Pulsed value of voltage or current in volts or amps
- **td1**: Rise delay time in seconds. Default=0.0.
- **td2**: Fall delay time in seconds. Default=td1+TSTEP.
- **τ1**: Rise time constant in seconds. Default=TSTEP.
- **τ2**: Fall time constant in seconds. Default=TSTEP.

TSTEP is the printing increment, and TSTOP is the final time.

The waveform shape is given by the following table of expressions:

<table>
<thead>
<tr>
<th>Time</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to td1</td>
<td>v1</td>
</tr>
<tr>
<td>td1 to td2</td>
<td>v1 + (v2 - v1) \cdot \left[ 1 - \exp\left(-\frac{Time - td1}{τ1}\right) \right]</td>
</tr>
</tbody>
</table>
Example

VIN 3 0 EXP (-4 -1 2NS 30NS 60NS 40NS)

The above example describes an exponential transient source that is connected between nodes 3 and 0. It has an initial t=0 voltage of -4 V and a final voltage of -1 V. The waveform rises exponentially from -4 V to -1 V with a time constant of 30 ns. At 60 ns it starts dropping to -4 V again, with a time constant of 40 ns.

Figure 5-3: Exponential Source Function

\[
\text{td2 to tstop} \quad v_1 + (v_2 - v_1) \cdot \left[ 1 - \exp \left( \frac{td2 - td1}{\tau_1} \right) \right] \\
\exp \left[ -\left( \frac{Time - td2}{\tau_2} \right) \right]
\]
*FILE: EXP.SP THE EXPONENTIAL WAVEFORM
.OPTIONS POST
.PARAM V1=-4 V2=-1 TD1=5N TAU1=30N TAU2=40N TD2=80N
V 1 0 EXP (V1 V2 TD1 TAU1 TD2 TAU2)
R 1 0 1
.TRAN .05N 200N
.END

This example shows an entire Star-Hspice netlist that contains an EXP voltage source. It has an initial t=0 voltage of -4 V and a final voltage of -1 V. The waveform rises exponentially from -4 V to -1 V with a time constant of 30 ns. At 80 ns it starts dropping to -4 V again, with a time constant of 40 ns.

Piecewise Linear (PWL) Source Function

The general syntax for including a piecewise linear source in an independent voltage or current source is:

**General Form:**

```
Vxxx n+ n- PWL <(> t1 v1 <t2 v2 t3 v3...> <R <=repeat>>
+ <TD=delay> <})
```

Or

```
Ixxx n+ n- PWL <(> t1 v1 <t2 v2 t3 v3...> <R <=repeat>>
+ <TD=delay> <})
```

**MSINC and ASPEC form:**

```
Vxxx n+ n- PL <(> v1 t1 <v2 t2 v3 t3...> <R <=repeat>>
+ <TD=delay> <})
```

Or

```
Ixxx n+ n- PL <(> v1 t1 <v2 t2 v3 t3...> <R <=repeat>>
+ <TD=delay> <})
```
The arguments are defined as:

- **Vxxx, Ixxx**: Independent voltage source that will exhibit the piecewise linear response.
- **PWL**: Keyword for a piecewise linear time-varying source
- **v1 v2 ... vn**: Current or voltage values at corresponding timepoint
- **t1 t2 ... tn**: Timepoint values where the corresponding current or voltage value is valid.
- **R=repeat**: Keyword and time value to specify a repeating function. With no argument, the source repeats from the beginning of the function. “repeat” is time in units of seconds which specifies the start point of the waveform that is to be repeated. This time needs to be less than the greatest time point tn.
- **TD=delay**: Time in units of seconds that specifies the length of time to delay the piecewise linear function.

Each pair of values (t1, v1) specifies that the value of the source is v1 (in volts or amps) at time t1. The value of the source at intermediate values of time is determined by linear interpolation between the time points. ASPEC style formats are accommodated by the “PL” form of the function, which reverses the order of the time-voltage pairs to voltage-time pairs. Star-Hspice uses the DC value of the source as the time-zero source value if no time-zero point is given. Also, Star-Hspice does not force the source to terminate at the TSTOP value specified in the .TRAN statement.

If the slope of the piecewise linear function changes below a certain tolerance, the timestep algorithm may not choose the specified time points as simulation time points, thereby obtaining a value for the source voltage or current by extrapolation of neighboring values. In this situation, you may notice a small deviation of the simulated voltage from that specified in the PWL list. To force Star-Hspice to use the specified values, use the SLOPETOL option to reduce the
slope change tolerance (see “Specifying Simulation Output” on page 8-1 for more information about this option).

Specify “R” to cause the function to repeat. You can specify a value after this “R” to indicate the beginning of the function to be repeated: the repeat time must equal a breakpoint in the function. For example, if t1 = 1, t2 = 2, t3 = 3, and t4 = 4, “repeat” can be equal to 1, 2, or 3.

Specify TD=val to cause a delay at the beginning of the function. You can use TD with or without the repeat function.

Example

*FILE: PWL.SP THE REPEATED PIECEWISE LINEAR SOURCE
*ILLUSTRATION OF THE USE OF THE REPEAT FUNCTION “R”
*file pwl.sp REPEATED PIECEWISE LINEAR SOURCE
.OPTION POST
.TRAN 5N 500N
V1 1 0 PWL 60N 0V, 120N 0V, 130N 5V, 170N 5V, 180N 0V, R 0N
R1 1 0 1
V2 2 0 PL 0V 60N, 0V 120N, 5V 130N, 5V 170N, 0V 180N, R 60N
R2 2 0 1
.END

This example shows an entire Star-Hspice netlist that contains two piecewise linear voltage sources. The two sources have the same function (the first one is in normal format, and the second in ASPEC format). The first source has a repeat specified to start at the beginning of the function, whereas the second repeat starts at the first timepoint. See Figure 5-4 for the difference in responses.
Data Driven Piecewise Linear Source Function

The general syntax for including a data-driven piecewise linear source in an independent voltage or current source is:

**General Form:**

\[ Vxxx \ n+ \ n- \ PWL \ (TIME, \ PV) \]

or

\[ Ixxx \ n+ \ n- \ PWL \ (TIME, \ PV) \]

along with:

```
.DATA dataname
TIME PV
t1 v1
t2 v2
t3 v3
t4 v4
... .ENDATA
.TRAN DATA=datanam
```

**Figure 5-4: Results of Using the Repeat Function**
The arguments are defined as:

**TIME**  
Parameter name for time value provided in a .DATA statement.

**PV**  
Parameter name for amplitude value provided in a .DATA statement.

You must use this source with a .DATA statement that contains time-value pairs. For each \( t_n-v_n \) (time-value) pair given in the .DATA block, the data driven PWL function outputs a current or voltage of the given \( t_n \) duration and with the given \( v_n \) amplitude.

This source allows you to use the results of one simulation as an input source in another simulation. The transient analysis must be data driven.

**Example**

*DATA DRIVEN PIECEWISE LINEAR SOURCE

V1 1 0 PWL(TIME, pv1)  
R1 1 0 1  
V2 2 0 PWL(TIME, pv2)  
R2 2 0 1  
.DATA dsrc  
TIME pv1 pv2  
0n 5v 0v  
5n 0v 5v  
10n 0v 5v  
.ENDDATA  
.TRAN DATA=dsrc  
.END

This example shows an entire Star-Hspice netlist that contains two data-driven piecewise linear voltage sources. The .DATA statement contains the two sets of value data referenced in the sources, \( pv1 \) and \( pv2 \). The .TRAN statement references the data name.
Single-Frequency FM Source Function

The general syntax for including a single-frequency frequency-modulated source in an independent voltage or current source is:

**General Form:**

\[
\begin{align*}
Vxxx & \ n+ \ n- \ SFFM \ (<(> \ vo \ va \ <fc \ <mdi \ <fs>>> <>) > \\
or \\
Ixxx & \ n+ \ n- \ SFFM \ (<(> \ vo \ va \ <fc \ <mdi \ <fs>>> <>) > \\
\end{align*}
\]

The arguments are as follows:

- **Vxxx, Ixxx**: Independent voltage source that will exhibit the frequency-modulated response.
- **SFFM**: Keyword for a single-frequency frequency-modulated time-varying source.
- **vo**: Output voltage or current offset, in volts or amps.
- **va**: Output voltage or current amplitude, in volts or amps.
- **fc**: Carrier frequency in Hz. Default=1/TSTOP.
- **mdi**: Modulation index that determines the magnitude of deviation from the carrier frequency. Values normally lie between 1 and 10. Default=0.0.
- **fs**: Signal frequency in Hz. Default=1/TSTOP.

The waveform shape is given by the following expression:

\[
\text{sourcevalue} = vo + va \cdot \sin[2 \cdot \pi \cdot fc \cdot \text{Time} + mdi \cdot \sin(2 \cdot \pi \cdot fc \cdot \text{Time})]
\]

**Note:** TSTOP is discussed in the .TRAN statement description.
Example

*FILE: SFFM.SP THE SINGLE FREQUENCY FM SOURCE
.OPTIONS POST
V 1 0 SFFM (0, 1M, 20K, 10, 5K)
R 1 0 1
.TRAN .0005M .5MS
.END

This example shows an entire Star-Hspice netlist that contains a single-frequency frequency-modulated voltage source. The source has an offset voltage of 0 volts, and a maximum voltage of 1 millivolt. The carrier frequency is 20 kHz, and the signal is 5 kHz, with a modulation index of 10 (the maximum wavelength is roughly 10 times longer than the minimum).

Figure 5-5: Single Frequency FM Source
Amplitude Modulation Source Function

The general syntax for including a single-frequency frequency-modulated source in an independent voltage or current source is:

**General form:**

\[
V_{xxx} \ n+ \ n- \ AM \ (< sa \ oc \ fm \ fc \ <td> \ <>)
\]

or

\[
I_{xxx} \ n+ \ n- \ AM \ (< sa \ oc \ fm \ fc \ <td> \ <>)
\]

The arguments are as follows:

where

- **Vxxx, Ixxx**: Independent voltage source that will exhibit the amplitude-modulated response.
- **AM**: Keyword for an amplitude-modulated time-varying source
- **sa**: Signal amplitude in volts or amps. Default=0.0.
- **fc**: Carrier frequency in hertz. Default=0.0.
- **fm**: Modulation frequency in hertz. Default=1/TSTOP.
- **oc**: Offset constant, a unitless constant that determines the absolute magnitude of the modulation. Default=0.0.
- **td**: Delay time before start of signal in seconds. Default=0.0.

The waveform shape is given by the following expression:

\[
sourcevalue = sa \cdot \{oc + SIN[2 \cdot \pi \cdot fm \cdot (Time - td)]\} \cdot SIN[2 \cdot \pi \cdot fc \cdot (Time - td)]
\]
Example

.OPTION POST
.TRAN .01M 20M
V1 1 0 AM(10 1 100 1K 1M)
R1 1 0 1
V2 2 0 AM(2.5 4 100 1K 1M)
R2 2 0 1
V3 3 0 AM(10 1 1K 100 1M)
R3 3 0 1
.END

This example shows an entire Star-Hspice netlist that contains three amplitude-modulated voltage sources. The first has an amplitude of 10, an offset constant of 1, a carrier frequency of 1 kHz, a modulation frequency of 100 Hz, and a delay of 1 millisecond. The second source has the same frequencies and delay, but with an amplitude of 2.5 and an offset constant of 4. The third source is the same as the first but with the carrier and modulation frequencies exchanged.

Figure 5-6: Amplitude Modulation Plot
Using Voltage and Current Controlled Elements

Star-Hspice has four voltage and current controlled elements, known as E, G, H, and F Elements. You can use these controlled elements in Star-Hspice to model both MOS and bipolar transistors, tunnel diodes, SCRs, as well as analog functions such as operational amplifiers, summers, comparators, voltage controlled oscillators, modulators, and switched capacitor circuits. The controlled elements are either linear or nonlinear functions of controlling node voltages or branch currents, depending on whether you use the polynomial or piecewise linear functions. Each controlled element has different functions:

- The E Element is a voltage and/or current controlled voltage source, an ideal op-amp, an ideal transformer, an ideal delay element, or a piecewise linear voltage controlled multi-input AND, NAND, OR, and NOR gate.
- The G Element is a voltage and/or current controlled current source, a voltage controlled resistor, a piecewise linear voltage controlled capacitor, an ideal delay element, or a piecewise linear multi-input AND, NAND, OR, and NOR gate.
- The H Element is a current controlled voltage source, an ideal delay element, or a piecewise linear current controlled multi-input AND, NAND, OR, and NOR gate.
- The F Element is a current controlled current source, an ideal delay element, or a piecewise linear current controlled multi-input AND, NAND, OR, and NOR gate.

The following sections discuss the polynomial and piecewise linear functions and describe element statements for linear or nonlinear functions.

Polynomial Functions

The controlled element statement allows the definition of the controlled output variable (current, resistance, or voltage) as a polynomial function of one or more voltages or branch currents. You can select three polynomial equations through the POLY(NDIM) parameter.
POLY(1) One-dimensional equation
POLY(2) Two-dimensional equation
POLY(3) Three-dimensional equation

The POLY(1) polynomial equation specifies a polynomial equation as a function of one controlling variable, POLY(2) as a function of two controlling variables, and POLY(3) as a function of three controlling variables.

Along with each polynomial equation are polynomial coefficient parameters (P0, P1 … Pn) that can be set to explicitly define the equation.

One-Dimensional Function

If the function is one-dimensional (a function of one branch current or node voltage), the function value $FV$ is determined by the following expression:

$$ FV = P0 + (P1 \cdot FA) + (P2 \cdot FA^2) + (P3 \cdot FA^3) + (P4 \cdot FA^4) + (P5 \cdot FA^5) + \ldots $$

<table>
<thead>
<tr>
<th>$FV$</th>
<th>Controlled voltage or current from the controlled source</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P0$ to $PN$</td>
<td>Coefficients of polynomial equation</td>
</tr>
<tr>
<td>$FA$</td>
<td>Controlling branch current or nodal voltage</td>
</tr>
</tbody>
</table>

**Note:** If the polynomial is one-dimensional and exactly one coefficient is specified, Star-Hspice assumes it to be $P1$ ($P0 = 0.0$) to facilitate the input of linear controlled sources.
Example

The following controlled source statement is an example of a one-dimensional function:

\[ E1 \ 5 \ 0 \ \text{POLY}(1) \ 3 \ 2 \ 1 \ 2.5 \]

The above voltage-controlled voltage source is connected to nodes 5 and 0. The single-dimension polynomial function parameter, POLY(1), informs Star-Hspice that E1 is a function of the difference of one nodal voltage pair, in this case, the voltage difference between nodes 3 and 2, hence \( F_A = V(3,2) \). The dependent source statement then specifies that \( P_0 = 1 \) and \( P_1 = 2.5 \). From the one-dimensional polynomial equation above, the defining equation for \( V(5,0) \) is

\[ V(5, 0) = 1 + 2.5 \cdot V(3,2) \]

Two-Dimensional Function

Where the function is two-dimensional (a function of two node voltages or two branch currents), \( F_V \) is determined by the following expression:

\[
F_V = P_0 + (P_1 \cdot F_A) + (P_2 \cdot F_B) + (P_3 \cdot F_A^2) + (P_4 \cdot F_A \cdot F_B) + (P_5 \cdot F_B^2) \\
+ (P_6 \cdot F_A^3) + (P_7 \cdot F_A^2 \cdot F_B) + (P_8 \cdot F_A \cdot F_B^2) + (P_9 \cdot F_B^3) + \ldots
\]

For a two-dimensional polynomial, the controlled source is a function of two nodal voltages or currents. To specify a two-dimensional polynomial, set POLY(2) in the controlled source statement.

Example

For example, generate a voltage controlled source that gives the controlled voltage, \( V(1,0) \), as:

\[ V(1, 0) = 3 \cdot V(3,2) + 4 \cdot V(7,6)^2 \]

To implement this function, use the following controlled source element statement:

\[ E1 \ 1 \ 0 \ \text{POLY}(2) \ 3 \ 2 \ 7 \ 6 \ 0 \ 3 \ 0 \ 0 \ 0 \ 4 \]
This specifies a controlled voltage source connected between nodes 1 and 0 that is controlled by two differential voltages: the voltage difference between nodes 3 and 2 and the voltage difference between nodes 7 and 6, that is, \( FA = V(3, 2) \) and \( FB = V(7, 6) \). The polynomial coefficients are \( P_0 = 0, P_1 = 3, P_2 = 0, P_3 = 0, P_4 = 0, \) and \( P_5 = 4 \).

**Three-Dimensional Function**

For a three-dimensional polynomial function with arguments \( FA, FB, \) and \( FC \), the function value \( FV \) is determined by the following expression:

\[
FV = P_0 + (P_1 \cdot FA) + (P_2 \cdot FB) + (P_3 \cdot FC) + (P_4 \cdot FA^2) \\
+ (P_5 \cdot FA \cdot FB) + (P_6 \cdot FA \cdot FC) + (P_7 \cdot FB^2) + (P_8 \cdot FB \cdot FC) \\
+ (P_9 \cdot FC^2) + (P_{10} \cdot FA^3) + (P_{11} \cdot FA^2 \cdot FB) + (P_{12} \cdot FA^2 \cdot FC) \\
+ (P_{13} \cdot FA \cdot FB^2) + (P_{14} \cdot FA \cdot FB \cdot FC) + (P_{15} \cdot FA \cdot FC^2) \\
+ (P_{16} \cdot FB^3) + (P_{17} \cdot FB^2 \cdot FC) + (P_{18} \cdot FB \cdot FC^2) \\
+ (P_{19} \cdot FC^3) + (P_{20} \cdot FA^4) + \ldots
\]

**Example**

For example, generate a voltage controlled source that gives the voltage as:

\[
V(1, 0) = 3 \cdot V(3, 2) + 4 \cdot V(7, 6)^2 + 5 \cdot V(9, 8)^3
\]

from the above defining equation and the three-dimensional polynomial equation:

\[
FA = V(3, 2) \\
FB = V(7, 6) \\
FC = V(9, 8) \\
P_1 = 3 \\
P_7 = 4 \\
P_{19} = 5
\]
Substituting these values into the voltage controlled voltage source statement yields the following:

\[ V(1, 0) \text{ POLY}(3) \ 3 \ 2 \ 7 \ 6 \ 9 \ 8 \ 0 \ 3 \ 0 \ 0 \ 0 \ 0 \ 4 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 5 \]

The above specifies a controlled voltage source connected between nodes 1 and 0 that is controlled by three differential voltages: the voltage difference between nodes 3 and 2, the voltage difference between nodes 7 and 6, and the voltage difference between nodes 9 and 8, that is, \( FA=V(3,2) \), \( FB=V(7,6) \), and \( FC=V(9,8) \). The statement gives the polynomial coefficients as \( P1=3 \), \( P7=4 \), \( P19=5 \), and the rest are zero.

**Piecewise Linear Function**

The one-dimensional piecewise linear function allows you to model some special element characteristics, such as those of tunnel diodes, silicon-controlled rectifiers, and diode breakdown regions. The piecewise linear function can be described by specifying measured data points. Although the device characteristic is described by some data points, Star-Hspice automatically smooths the corners to ensure derivative continuity and, as a result, better convergence.

A parameter \( \text{DELTA} \) is provided to control the curvature of the characteristic at the corners. The smaller the \( \text{DELTA} \), the sharper the corners are. The maximum \( \text{DELTA} \) is limited to half of the smallest breakpoint distance. If the breakpoints are quite separated, specify the \( \text{DELTA} \) to a proper value. You can specify up to 100 point pairs. At least two point pairs (four coefficients) must be specified.

In order to model bidirectional switch or transfer gates, the functions NPWL and PPWL are provided for \( G \) Elements. The NPWL and PPWL function like NMOS and PMOS transistors.

The piecewise linear function also models multi-input AND, NAND, OR, and NOR gates. In this case, only one input determines the state of the output. In AND / NAND gates, the input with the smallest value is used in the piecewise linear function to determine the corresponding output of the gates. In the OR / NOR gates, the input with the largest value is used to determine the corresponding output of the gates.
Voltage Dependent Voltage Sources — E Elements

E Element syntax statements are described in the following paragraphs. The parameters are defined in the following section.

Voltage Controlled Voltage Source (VCVS)

Syntax

Linear

\[ E_{\text{xxx}} \text{ n+ n- } \text{VCVS} \text{ in+ in- gain } \text{MAX}=\text{val} \text{ } \text{MIN}=\text{val} \text{ } \text{SCALE}=\text{val} \text{ } \text{MAX}=\text{val} \text{ } \text{MIN}=\text{val} \text{ } \text{ABS}=\text{1} \text{ } \text{IC}=\text{val} \]

Polynomial

\[ E_{\text{xxx}} \text{ n+ n- } \text{VCVS} \text{ POLY(NDIM) in1+ in1- ... inndim+ inndim- } \text{TC1}=\text{val} \text{ } \text{TC2}=\text{val} \text{ } \text{SCALE}=\text{val} \text{ } \text{MAX}=\text{val} \text{ } \text{MIN}=\text{val} \text{ } \text{ABS}=\text{1} \text{ } P0 \text{ } P1... \]

Piecewise Linear

\[ E_{\text{xxx}} \text{ n+ n- } \text{VCVS} \text{ PWL(1) in+ in- } \text{DELTA}=\text{val} \text{ } \text{SCALE}=\text{val} \text{ } \text{TC1}=\text{val} \text{ } \text{TC2}=\text{val} \text{ } x1,y1 \text{ } x2,y2 \text{ } ... \text{ } x100,y100 \text{ } \text{IC}=\text{val} \]

Multi-Input Gates

\[ E_{\text{xxx}} \text{ n+ n- } \text{VCVS} \text{ gatetype(k) in1+ in1- ... ink+ ink- } \text{DELTA}=\text{val} \text{ } \text{TC1}=\text{val} \text{ } \text{TC2}=\text{val} \text{ } \text{SCALE}=\text{val} \text{ } x1,y1 \text{ } ... \text{ } x100,y100 \text{ } \text{IC}=\text{val} \]

Delay Element

\[ E_{\text{xxx}} \text{ n+ n- } \text{VCVS} \text{ DELAY in+ in- TD}=\text{val} \text{ } \text{SCALE}=\text{val} \text{ } \text{TC1}=\text{val} \text{ } \text{TC2}=\text{val} \text{ } \text{NPDELAY}=\text{val} \]

Behavioral Voltage Source

The syntax is:

\[ E_{\text{xxx}} \text{ n+ n- VOL='equation'} \text{ } \text{MAX}=\text{val} \text{ } \text{MIN}=\text{val} \]

Ideal Op-Amp

The syntax is:

\[ E_{\text{xxx}} \text{ n+ n- OPAMP in+ in-} \]
Ideal Transformer

The syntax is:

\[ \text{Exxx n+ n- TRANSFORMER in+ in- k} \]

Parameter Definitions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABS</strong></td>
<td>Output is absolute value if ABS=1.</td>
</tr>
<tr>
<td><strong>DELAY</strong></td>
<td>Keyword for the delay element. The delay element is the same as voltage controlled voltage source, except it is associated by a propagation delay TD. This element facilitates the adjustment of propagation delay in the macro-modelling process. <strong>Note:</strong> DELAY is a reserved word and should not be used as a node name.</td>
</tr>
<tr>
<td><strong>DELTA</strong></td>
<td>Used to control the curvature of the piecewise linear corners. The parameter defaults to one-fourth of the smallest breakpoint distances. The maximum is limited to one-half of the smallest breakpoint distances.</td>
</tr>
<tr>
<td><strong>Exxx</strong></td>
<td>Voltage controlled element name. The parameter must begin with an “E” followed by up to 1023 alphanumeric characters.</td>
</tr>
<tr>
<td><strong>gain</strong></td>
<td>Voltage gain</td>
</tr>
<tr>
<td><strong>gatetype(k)</strong></td>
<td>Can be one of AND, NAND, OR, or NOR. (k) represents the number of inputs of the gate. The x’s and y’s represent the piecewise linear variation of output as a function of input. In the multi-input gates only one input determines the state of the output.</td>
</tr>
<tr>
<td><strong>IC</strong></td>
<td>Initial condition: the initial estimate of the value(s) of the controlling voltage(s). If IC is not specified, the default=0.0.</td>
</tr>
<tr>
<td><strong>in +/-</strong></td>
<td>Positive or negative controlling nodes. Specify one pair for each dimension.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>k</strong></td>
<td>Ideal transformer turn ratio: &lt;br&gt;( V(\text{in}+,\text{in}-) = k \cdot V(\text{n}+,\text{n}-) ) &lt;br&gt;or, number of gates input</td>
</tr>
<tr>
<td><strong>MAX</strong></td>
<td>Maximum output voltage value. The default is undefined and sets no maximum value.</td>
</tr>
<tr>
<td><strong>MIN</strong></td>
<td>Minimum output voltage value. The default is undefined and sets no minimum value.</td>
</tr>
<tr>
<td><strong>n+/−</strong></td>
<td>Positive or negative node of controlled element</td>
</tr>
<tr>
<td><strong>NDIM</strong></td>
<td>Polynomial dimensions. If POLY(NDIM) is not specified, a one-dimensional polynomial is assumed. NDIM must be a positive number.</td>
</tr>
<tr>
<td><strong>NPDELAY</strong></td>
<td>Sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep &lt;br&gt;That is, &lt;br&gt;( NPDELAY_{\text{default}} = \max\left[\min\left(\frac{TD, tstop}{tstep}\right), 10\right] ) &lt;br&gt;The values of tstep and tstop are specified in the .TRAN statement.</td>
</tr>
<tr>
<td><strong>OPAMP</strong></td>
<td>The keyword for ideal op-amp element. OPAMP is a reserved word and should not be used as a node name.</td>
</tr>
<tr>
<td><strong>P0, P1 ...</strong></td>
<td>The polynomial coefficients. When one coefficient is specified, Star-Hspice assumes it to be P1 (P0=0.0), and the element is linear. When more than one polynomial coefficient is specified, the element is nonlinear, and P0, P1, P2 ... represent them (see “Polynomial Functions” on page 5-24).</td>
</tr>
<tr>
<td><strong>POLY</strong></td>
<td>Polynomial keyword function</td>
</tr>
<tr>
<td><strong>PWL</strong></td>
<td>Piecewise linear keyword function</td>
</tr>
<tr>
<td><strong>SCALE</strong></td>
<td>Element value multiplier</td>
</tr>
</tbody>
</table>
Example

**Ideal OpAmp**

A voltage amplifier with supply limits can be built with the voltage controlled voltage source. The output voltage across nodes 2,3 = v(14,1) * 2. The voltage gain parameter, 2, is also given. The MAX and MIN parameters specify a maximum E1 voltage of 5 V and a minimum E1 voltage output of -5 V. If, for instance, V(14,1) = -4V, E1 would be set to -5 V and not -8 V, as the equation would produce.

\[ E_{\text{opamp}} 2 3 14 1 \ \text{MAX}=+5 \ \text{MIN}=-5 \ 2.0 \]

A user-defined parameter can be used in the following format to specify a value for polynomial coefficient parameters:

\[
\text{.PARAM} \ \text{CU} = 2.0
\]

\[
E1 \ 2 \ 3 \ 14 \ 1 \ \text{MAX}=+5 \ \text{MIN}=-5 \ \text{CU}
\]

**Voltage Summer**

An ideal voltage summer specifies the source voltage as a function of three controlling voltage(s): V(13,0), V(15,0) and V(17,0). It describes a voltage source with the value:

\[
V(13,0) + V(15,0) + V(17,0)
\]
This example represents an ideal voltage summer. The three controlling voltages are initialized for a DC operating point analysis to 1.5, 2.0, and 17.25 V, respectively.

```
EX 17 0 POLY(3) 13 0 15 0 17 0 0 1 1 1
    + IC=1.5,2.0,17.25
```

**Polynomial Function**

The voltage controlled source also can output a nonlinear function using the one-dimensional polynomial. Since the POLY parameter is not specified, a one-dimensional polynomial is assumed—that is, a function of one controlling voltage. The equation corresponds to the element syntax. Behavioral equations replace this older method.

```
V (3,4) = 10.5 + 2.1 *V(21,17) + 1.75 *V(21,17)^2
E2 3 4 POLY 21 17 10.5 2.1 1.75
```

**Zero Delay Inverter Gate**

You can build a simple inverter with no delay with a piecewise linear transfer function.

```
Ein 0 PWL(1) in 0 .7v,5v 1v,0v
```

**Ideal Transformer**

With the turn ratio 10 to 1, the voltage relationship is V(out)=V(in)/10.

```
Etrans out 0 TRANSFORMER in 0 10
```

**Voltage Controlled Oscillator (VCO)**

Use the keyword VOL to define a single-ended input that controls the output of a VCO.

In the following example, the frequency of the sinusoidal output voltage at node “out” is controlled by the voltage at node “control”. Parameter “v0” is the DC offset voltage and “gain” is the amplitude. The output is a sinusoidal voltage with a frequency of “freq · control”.

```
Evco out 0 VOL=’v0+gain*SIN(6.28 freq*v(control)
  + *TIME)’
```
Voltage Dependent Current Sources — G Elements

G Element syntax statements are described in the following pages. The parameters are defined in the following section.

Voltage Controlled Current Source (VCCS)

Syntax

Linear

\[
gxxx \ n+ \ n- \ \text{<VCCS> in+ in- transconductance <MAX=val> <MIN=val> <SCALE=val> + \ <M=val> <TC1=val> <TC2=val> <ABS=1> <IC=val>}
\]

Polynomial

\[
gxxx \ n+ \ n- \ \text{<VCCS> POLY(NDIM) in1+ in1- ... inndim+ inndim-> MAX=val> + \ <MIN=val> <SCALE=val> <M=val> <TC1=val> <TC2=val> <ABS=1> + \ P0<P1...> <IC=vals>}
\]

Piecewise Linear

\[
gxxx \ n+ \ n- \ \text{<VCCS> PWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> + \ <TC1=val> <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> + \ <SMOOTH=val>}
\]

\[
gxxx \ n+ \ n- \ \text{<VCCS> NPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> + \ <TC1=val> <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>}
\]

\[
gxxx \ n+ \ n- \ \text{<VCCS> PPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> + \ <TC1=val> <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>}
\]

Multi-Input Gates

\[
gxxx \ n+ \ n- \ \text{<VCCS> gatetype(k) in1+ in1- ... ink+ ink- <DELTA=val> <TC1=val> + \ <TC2=val> <SCALE=val> <M=val> x1,y1 ... x100,y100<IC=val>}
\]

Delay Element

\[
gxxx \ n+ \ n- \ \text{<VCCS> DELAY in+ in- TD=val <SCALE=val> <TC1=val> <TC2=val> + \ NPDELAY=val}
\]

Behavioral Current Source

Syntax

\[
gxxx \ n+ \ n-\ \text{CUR='equation' <MAX=val> <MIN=val> <M=val> + \ <SCALE=val>}
\]
Voltage Controlled Resistor (VCR)

Syntax

**Linear**

\[ \text{Gxxx n+ n- VCR in+ in- transfactor <MAX=val> <MIN=val> <SCALE=val> <M=val> + \<TC1=val> <TC2=val> <IC=val> } \]

**Polynomial**

\[ \text{Gxxx n+ n- VCR POLY(NDIM) in1+ in1- ... <inndim+ inndim-> <MAX=val> + \<MIN=val><SCALE=val> <M=val> <TC1=val> <TC2=val> P0 <P1...> + \<IC=vals> } \]

**Piecewise Linear**

\[ \text{Gxxx n+ n- VCR PWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> <TC1=val> + \<TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val> } \]

\[ \text{Gxxx n+ n- VCR NPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> <TC1=val> + \<TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val> } \]

\[ \text{Gxxx n+ n- VCR PPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> <TC1=val> + \<TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val> } \]

**Multi-Input Gates**

\[ \text{Gxxx n+ n- VCR gatetype(k) in1+ in1- ... ink+ ink- <DELTA=val> + \<TC1=val> <TC2=val> <SCALE=val> <M=val> x1,y1 ... x100,y100 <IC=val> } \]

Voltage Controlled Capacitor (VCCAP)

Syntax (Piecewise Linear)

\[ \text{Gxxx n+ n- VCCAP PWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> + \<TC1=val><TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val> } \]

The two functions NPWL and PPWL allow the interchange of the “n+” and “n-” nodes while keeping the same transfer function. This action is summarized as follows:

**NPWL Function**

For node “in-” connected to “n-”:

If \(v(n+,n-) > 0\), then the controlling voltage would be \(v(in+,in-)\). Otherwise, the controlling voltage is \(v(in+,n+)\).
For node “in-” connected to “n+”:

If \(v(n+,n-) < 0\), then the controlling voltage would be \(v(in+,in-)\). Otherwise, the controlling voltage is \(v(in+,n+)\).

**PPWL Function**

For node “in-” connected to “n-”:

If \(v(n+,n-) < 0\), then the controlling voltage would be \(v(in+,in1-)\). Otherwise, the controlling voltage is \(v(in+,n+)\).

For node “in-” connected to “n+”:

If \(v(n+,n-) > 0\), then the controlling voltage would be \(v(in+,in-)\). Otherwise, the controlling voltage is \(v(in+,n+)\).

**Parameter Definitions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABS</strong></td>
<td>Output is absolute value if ABS=1.</td>
</tr>
<tr>
<td><strong>CUR, VALUE</strong></td>
<td>Current output that flows from n+ to n-. The equation that you define can be a function of node voltages, branch currents, TIME, temperature (TEMPER), and frequency (HERTZ).</td>
</tr>
<tr>
<td><strong>DELAY</strong></td>
<td>Keyword for the delay element. The delay element is the same as voltage controlled current source except it is associated by a propagation delay TD. This element facilitates the adjustment of propagation delay in the macromodel process. <strong>Note:</strong> Because DELAY is a Star-Hspice keyword, it should not be used as a node name.</td>
</tr>
<tr>
<td><strong>DELTA</strong></td>
<td>Used to control the curvature of the piecewise linear corners. The parameter defaults to 1/4 of the smallest breakpoint distances. The maximum is limited to 1/2 of the smallest breakpoint distances.</td>
</tr>
<tr>
<td><strong>Gxxx</strong></td>
<td>Voltage controlled element name. This parameter must begin with a “G” followed by up to 1023 alphanumeric characters.</td>
</tr>
<tr>
<td><strong>gatetype(k)</strong></td>
<td>Can be one of AND, NAND, OR, or NOR. The parameter (k) represents the number of inputs of the gate. The x’s and y’s represent the piecewise linear variation of output as a function of input. In the multi-input gates, only one input determines the state of the output.</td>
</tr>
<tr>
<td><strong>IC</strong></td>
<td>Initial condition. The initial estimate of the value(s) of the controlling voltage(s). If IC is not specified, the default=0.0.</td>
</tr>
<tr>
<td><strong>in +/-</strong></td>
<td>Positive or negative controlling nodes. Specify one pair for each dimension.</td>
</tr>
<tr>
<td><strong>M</strong></td>
<td>Number of element in parallel</td>
</tr>
<tr>
<td><strong>MAX</strong></td>
<td>Maximum current or resistance value. The default is undefined and sets no maximum value.</td>
</tr>
<tr>
<td><strong>MIN</strong></td>
<td>Minimum current or resistance value. The default is undefined and sets no minimum value.</td>
</tr>
<tr>
<td><strong>n +/-</strong></td>
<td>Positive or negative node of controlled element</td>
</tr>
<tr>
<td><strong>NDIM</strong></td>
<td>Polynomial dimensions. If POLY(NDIM) is not specified, a one-dimensional polynomial is assumed. NDIM must be a positive number.</td>
</tr>
</tbody>
</table>
| **NPDELAY** | Sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep. That is, 

\[
NPDELAY_{\text{default}} = \max \left[ \frac{\min(TD, tstop)}{tstep}, 10 \right]
\]

The values of tstep and tstop are specified in the .TRAN statement. |
| **NPWL** | Models the symmetrical bidirectional switch or transfer gate, NMOS |
### Voltage Dependent Current Sources — G Elements

#### Using Sources and Stimuli

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_0, P_1 \ldots )</td>
<td>The polynomial coefficients. When one coefficient is specified, Star-Hspice assumes it to be ( P_1 (P_0=0.0) ), and the element is linear. When more than one polynomial coefficient is specified, the element is nonlinear, and ( P_0, P_1, P_2 \ldots ) represent them (see “Polynomial Functions” on page 5-24).</td>
</tr>
<tr>
<td>POLY</td>
<td>Polynomial keyword function</td>
</tr>
<tr>
<td>PWL</td>
<td>Piecewise linear keyword function</td>
</tr>
<tr>
<td>PPWL</td>
<td>Models the symmetrical bidirectional switch or transfer gate, PMOS</td>
</tr>
<tr>
<td>SCALE</td>
<td>Element value multiplier</td>
</tr>
<tr>
<td>SMOOTH</td>
<td>For piecewise linear dependent source elements, SMOOTH selects the curve smoothing method. A curve smoothing method simulates exact data points you provide. This method can be used to make Star-Hspice simulate specific data points that correspond to measured data or data sheets, for example. Choices for SMOOTH are 1 or 2: 1 Selects the smoothing method used in Hspice releases prior to release H93A. Use this method to maintain compatibility with simulations done using releases older than H93A. 2 Selects the smoothing method that uses data points you provide. This is the default for Hspice releases starting with H93A.</td>
</tr>
</tbody>
</table>
| TC1, TC2 | First and second order temperature coefficients. The SCALE is updated by temperature: 
\[
\text{SCALE}_{\text{eff}} = \text{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)
\] |
| TD | Time delay keyword |
Example

Switch

A voltage-controlled resistor represents a basic switch characteristic. The resistance between nodes 2 and 0 varies linearly from 10 meg to 1 m ohms when voltage across nodes 1 and 0 varies between 0 and 1 volt. Beyond the voltage limits, the resistance remains at 10 meg and 1 m ohms, respectively.

\[ \text{Gswitch 2 0 VCR PWL(1) 1 0 0v,10meg 1v,1m} \]

Switch-Level MOSFET

Model a switch level n-channel MOSFET by the N-piecewise linear resistance switch. The resistance value does not change when the node d and s positions are switched.

\[ \text{Gnmos d s VCR NPWL(1) g s LEVEL=1 0.4v,150g + 1v,10meg 2v,50k 3v,4k 5v,2k} \]
**Voltage Controlled Capacitor**

The capacitance value across nodes (out,0) varies linearly from 1 p to 5 p when voltage across nodes (ctrl,0) varies between 2 v and 2.5 v. Beyond the voltage limits, the capacitance value remains constant at 1 picofarad and 5 picofarads respectively.

```
Gcap out 0 VCCAP PWL(1) ctrl 0 2v,1p 2.5v,5p
```

**Zero Delay Gate**

Implement a two-input AND gate using an expression and a piecewise linear table. The inputs are voltages at nodes a and b, and the output is the current flow from node out to 0. The current is multiplied by the SCALE value, which in this example is specified as the inverse of the load resistance connected across the nodes (out,0).

```
Gand out 0 AND(2) a 0 b 0 SCALE='1/rload' 0v,0a 1v,.5a
+ 4v,4.5a 5v,5a
```

**Delay Element**

A delay is a low-pass filter type delay similar to that of an opamp. A transmission line, on the other hand, has an infinite frequency response. A glitch input to a G delay is attenuated similarly to a buffer circuit. In this example, the output of the delay element is the current flow from node out to node 1 with a value equal to the voltage across nodes (in, 0) multiplied by SCALE value and delayed by TD value.

```
Gdel out 0 DELAY in 0 TD=5ns SCALE=2 NPDELAY=25
```

**Diode Equation**

Model forward bias diode characteristic from node 5 to ground with a runtime expression. The saturation current is 1e-14 amp, and the thermal voltage is 0.025 v.

```
Gdio 5 0 CUR='1e-14*(EXP(V(5)/0.025)-1.0)'
```
**Diode Breakdown**

Model a diode breakdown region to forward region using the following example. When voltage across the diode goes beyond the piecewise linear limit values (-2.2v, 2v), the diode current remains at the corresponding limit values (-1a, 1.2a).

```hspice
Gdiode 1 0 PWL(1) 1 0 -2.2v, -1a -2v, -1pa .3v, .15pa 
+ .6v, 10ua 1v, 1a 2v, 1.2a
```

**Triode**

Both the following voltage controlled current sources implement a basic triode. The first uses the poly(2) operator to multiply the anode and grid voltages together and scale by .02. The next example uses the explicit behavioral algebraic description.

```hspice
gt i_anode cathode poly(2) anode, cathode grid, cathode 0 0 
+ 0 .02 gt i_anode cathode 
+cur='20m*v( anode, cathode )*v( grid, cathode )'
```
Dependent Voltage Sources — H Elements

H Element syntax statements are described in the following paragraphs. The parameters are defined in the following section.

Current Controlled Voltage Source — (CCVS)

Syntax

**Linear**

Hxxx n+ n- <CCVS> vn1 transresistance <MAX=val> <MIN=val> <SCALE=val>
+ <TC1=val><TC2=val> <ABS=1> <IC=val>

**Polynomial**

Hxxx n+ n- <CCVS> POLY(NDIM) vn1 <... vnndim> <MAX=val>MIN=val>
+ <TC1=val><TC2=val> <SCALE=val> <ABS=1> P0 <P1…> <IC=vals>

**Piecewise Linear**

Hxxx n+ n- <CCVS> PWL(1) vn1 <DELTA=val> <SCALE=val> <TC1=val> <TC2=val>
+ x1,y1 ... x100,y100 <IC=val>

**Multi-Input Gates**

Hxxx n+ n- gatetype(k) vn1, ... vnk <DELTA=val> <SCALE=val> <TC1=val>
+ <TC2=val> x1,y1 ... x100,y100 <IC=val>

**Delay Element**

Hxxx n+ n- <CCVS> DELAY vn1 TD=val <SCALE=val> <TC1=val> <TC2=val>
+ <NPDELAY=val>

Parameter Definitions

<table>
<thead>
<tr>
<th>ABS</th>
<th>Output is absolute value if ABS=1.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCVS</td>
<td>Keyword for current controlled voltage source. CCVS is a reserved word and should not be used as a node name.</td>
</tr>
<tr>
<td><strong>DELAY</strong></td>
<td>Keyword for the delay element. The delay element is the same as a current controlled voltage source except it is associated by a propagation delay TD. This element facilitates the adjustment of propagation delay in the macromodel process. DELAY is a reserved word and should not be used as a node name.</td>
</tr>
<tr>
<td><strong>DELTA</strong></td>
<td>Used to control the curvature of the piecewise linear corners. The parameter defaults to 1/4 of the smallest breakpoint distances. The maximum is limited to 1/2 of the smallest breakpoint distances.</td>
</tr>
<tr>
<td><strong>gatetype(k)</strong></td>
<td>Can be one of AND, NAND, OR, NOR. (k) represents the number of inputs of the gate. The x's and y's represent the piecewise linear variation of output as a function of input. In the multi-input gates only one input determines the state of the output.</td>
</tr>
<tr>
<td><strong>Hxxx</strong></td>
<td>Current controlled voltage source element name. The parameter must begin with an “H” followed by up to 1023 alphanumeric characters.</td>
</tr>
<tr>
<td><strong>IC</strong></td>
<td>Initial condition. This is the initial estimate of the value(s) of the controlling current(s) in amps. If IC is not specified, the default=0.0.</td>
</tr>
<tr>
<td><strong>MAX</strong></td>
<td>Maximum voltage value. The default is undefined and sets no maximum value.</td>
</tr>
<tr>
<td><strong>MIN</strong></td>
<td>Minimum voltage value. The default is undefined and sets no minimum value.</td>
</tr>
<tr>
<td><strong>n+/-</strong></td>
<td>Positive or negative controlled source connecting nodes</td>
</tr>
<tr>
<td><strong>NDIM</strong></td>
<td>Polynomial dimensions. If POLY(NDIM) is not specified, a one-dimensional polynomial is assumed. NDIM must be a positive number.</td>
</tr>
</tbody>
</table>
### Dependental Voltage Sources — H Elements

#### NPDELAY

Sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep.

That is,

\[
NPDELAY_{\text{default}} = \max \left[ \min \left( \frac{TD}{tstep}, \frac{tstop}{tstep} \right), 10 \right]
\]

The values of tstep and tstop are specified in the TRAN statement.

#### P0, P1 . . .

When one polynomial coefficient is specified, the source is linear, and the polynomial is assumed to be P1 (P0=0.0). When more than one polynomial coefficient is specified, the source is nonlinear, with the polynomials assumed as P0, P1, P2 . . .

#### POLY

Polynomial keyword function

#### PWL

Piecewise linear keyword function

#### SCALE

Element value multiplier

#### TC1, TC2

First and second order temperature coefficients. The SCALE is updated by temperature:

\[
\text{SCALE}_{\text{eff}} = \text{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)
\]

#### TD

Time delay keyword

#### transresistance

Current to voltage conversion factor

#### vn1 . . .

Names of voltage sources through which the controlling current flows. One name must be specified for each dimension.

#### x1,...

Controlling current through vn1 source. The x values must be in increasing order.

#### y1,...

Corresponding output voltage values of x
Example

\[
HX \ 20 \ 10 \ VCUR \ MAX=+10 \ MIN=-10 \ 1000
\]

The example above selects a linear current controlled voltage source. The controlling current flows through the dependent voltage source called VCUR. The defining equation of the CCVS is:

\[
HX = 1000 \cdot VCUR
\]

The defining equation specifies that the voltage output of HX is 1000 times the value of current flowing through CUR. If the equation produces a value of HX greater than +10 V or less than -10 V, HX, because of the MAX= and MIN= parameters, would be set to either 10 V or -10 V, respectively. CUR is the name of the independent voltage source that the controlling current flows through. If the controlling current does not flow through an independent voltage source, a dummy independent voltage source must be inserted.

\[
-PARAM \ CT=1000
\]

\[
HX \ 20 \ 10 \ VCUR \ MAX=+10 \ MIN=-10 \ CT
\]

\[
HXY \ 13 \ 20 \ POLY(2) \ VIN1 \ VIN2 \ 0 \ 0 \ 0 \ 0 \ 1 \ IC=0.5, 1.3
\]

The example above describes a dependent voltage source with the value:

\[
V = I(VIN1) \cdot I(VIN2)
\]

This two-dimensional polynomial equation specifies FA1=VIN1, FA2=VIN2, P0=0, P1=0, P2=0, P3=0, and P4=1. The controlling current for flowing through VIN1 is initialized at .5 mA. For VIN2, the initial current is 1.3 mA.

The direction of positive controlling current flow is from the positive node, through the source, to the negative node of vnam (linear). The polynomial (nonlinear) specifies the source voltage as a function of the controlling current(s).
Current Dependent Current Sources — F Elements

F Element syntax statements are described in the following paragraphs. The parameter definitions follow.

Current Controlled Current Source (CCCS)

Syntax

**Linear**
Fxxx n+ n- <CCCS> vn1 gain <MAX=val> <MIN=val> <SCALE=val> <TC1=val> + <TC2=val> <M=val> <ABS=1> <IC=val>

**Polynomial**
Fxxx n+ n- <CCCS> POLY(NDIM) vn1 <... vnndim> <MAX=val> <MIN=val> + <TC1=val> <TC2=val> <SCALE=vals> <M=val> <ABS=1> <IC=vals>

**Piecewise Linear**
Fxxx n+ n- <CCCS> PWL(1) vn1 <DELTA=val> <SCALE=val><TC1=val> <TC2=val> + <M=val> x1,y1 ... x100,y100 <IC=val>

**Multi-Input Gates**
Fxxx n+ n- <CCCS> gatetype(k) vn1, ... vnk <DELTA=val> <SCALE=val> <TC1=val> + <TC2=val> <M=val> <ABS=1> x1,y1 ... x100,y100 <IC=val>

**Delay Element**
Fxxx n+ n- <CCCS> DELAY vn1 TD=val <SCALE=val> <TC1=val> <TC2=val> + NPDELAY=val

Parameter Definitions

<table>
<thead>
<tr>
<th>ABS</th>
<th>Output is absolute value if ABS=1.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CCCS</strong></td>
<td>Keyword for current controlled current source. Note that CCCS is a reserved word and should not be used as a node name.</td>
</tr>
</tbody>
</table>
### DELAY
Keyword for the delay element. The delay element is the same as a current controlled current source except it is associated by a propagation delay TD. This element facilitates the adjustment of propagation delay in the macromodel process. **Note:** DELAY is a reserved word and should not be used as a node name.

### DELTA
Used to control the curvature of the piecewise linear corners. The parameter defaults to 1/4 of the smallest breakpoint distances. The maximum is limited to 1/2 of the smallest breakpoint distances.

### Fxxx
Current controlled current source element name. The parameter must begin with an “F”, followed by up to 1023 alphanumeric characters.

### gain
Current gain

### gatetype(k)
Can be one of AND, NAND, OR, or NOR. (k) represents the number of inputs of the gate. The x's and y's represent the piecewise linear variation of output as a function of input. In the multi-input gates, only one input determines the state of the output. The above keyword names should not be used as a node name.

### IC
Initial condition: the initial estimate of the value(s) of the controlling current(s) in amps. If IC is not specified, the default=0.0.

### M
Number of element in parallel

### MAX
Maximum output current value. The default is undefined and sets no maximum value.

### MIN
Minimum output current value. The default is undefined and sets no minimum value.

### n+/-
Positive or negative controlled source connecting nodes

### NDIM
Polynomial dimensions. If POLY(NDIM) is not specified, a one-dimensional polynomial is assumed. NDIM must be a positive number.
### NPDELAY
Sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep.

That is,

\[
NPDELAY_{\text{default}} = \max\left[\frac{\min(TD, tstop)}{tstep}, 10\right]
\]

The values of tstep and tstop are specified in the `.TRAN` statement.

### P0, P1 ...
When one polynomial coefficient is specified, Star- Hspice assumes it to be P1 (P0=0.0) and the source is linear. When more than one polynomial coefficient is specified, the source is nonlinear, and P0, P1, P2 ... represent them.

### POLY
Polynomial keyword function

### PWL
Piecewise linear keyword function

### SCALE
Element value multiplier

### TC1, TC2
First and second order temperature coefficients. The SCALE is updated by temperature:

\[
\text{SCALE}_{\text{eff}} = \text{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)
\]

### TD
Time delay keyword

### vn1 ...
Names of voltage sources through which the controlling current flows. One name must be specified for each dimension.

### x1,...
Controlling current through vn1 source. The x values must be in increasing order.

### y1,...
Corresponding output current values of x

### Example

\[
F1 13 5 \text{ VSENS MAX=}+3 \text{ MIN=} -3 5
\]
The above example describes a current controlled current source connected between nodes 13 and 5. The current that controls the value of the controlled source flows through the voltage source named VSENS (to use a current controlled current source, a dummy independent voltage source is often placed into the path of the controlling current). The defining equation is:

\[ I(F1) = 5 \cdot I(VSENS) \]

The current gain is 5, the maximum current flow through F1 is 3 A, and the minimum current flow is -3 A. If \( I(VSENS) = 2 \) A, \( I(F1) \) would be set to 3 amps and not 10 amps as would be suggested by the equation. A user-defined parameter can be specified for the polynomial coefficient(s), as shown below.

\[
\text{PARAM VU = 5 } \\
F1 13 5 VSENS \text{ MAX=+3 MIN=-3 VU}
\]

The next example describes a current controlled current source with the value:

\[ I(F2)=1e^{-3} + 1.3e^{-3} \cdot I(VCC) \]

\[
F2 12 10 \text{ POLY VCC 1MA 1.3M}
\]

Current flow is from the positive node through the source to the negative node. The direction of positive controlling current flow is from the positive node through the source to the negative node of vnam (linear), or to the negative node of each voltage source (nonlinear).

\[
\text{Fd 1 0 DELAY vin TD=7ns SCALE=5}
\]

The above example is a delayed current controlled current source.

\[
\text{Filim 0 out PWL(1) vsrc -1a,-1a 1a,1a}
\]

The final example is a piecewise linear current controlled current source.
Digital and Mixed Mode Stimuli

There are two methods of using digital stimuli in a Star-Hspice input netlist: U Element digital input files and vector input files. They are both described in this section.

U Element Digital Input Elements and Models

The U Element can reference digital input and digital output models for mixed mode simulation. Viewlogic’s Viewsim mixed mode simulator uses Star-Hspice with digital input from Viewsim. The state information comes from a digital file if Star-Hspice is being run in standalone mode. Digital outputs are handled in a similar fashion. In digital input file mode, the input file is <design>.d2a and the output file is named <design>.a2d.

A2D and D2A functions accept the terminal “\” backslash character as a line-continuation character to allow more than 255 characters in a line. This is needed because the first line of a digital file, which contains the signal name list, is often longer than the maximum line length accepted by some text editors.

A digital D2A file must not have a blank first line. If the first line of a digital file is blank, Star-Hspice issues an error message.

The following example demonstrates the use of the “\” line continuation character to format an input file for text editing. The file contains a signal list for a 64-bit bus.

```plaintext
...  
a00 a01 a02 a03 a04 a05 a06 a07 \ 
a08 a09 a10 a11 a12 a13 a14 a15 \ 
...  * Continuation of signal names
a56 a57 a58 a59 a60 a61 a62 a63  * End of signal names
...  * Remainder of file
```
The general syntax for including a U Element digital source in a Star-Hspice netlist is:

**General form:**

\[ \text{Uxxx interface nlo nhi mname SIGNAME = sname IS = val} \]

The arguments are defined as:

- **Uxxx**: Digital input element name. Must begin with a “U”, which can be followed by up to 1023 alphanumeric characters.
- **interface**: Interface node in the circuit to which the digital input is attached.
- **nlo**: Node connected to low level reference.
- ** nhi**: Node connected to high level reference.
- **mname**: Digital input model reference (U model).
- **SIGNAME = sname**: Signal name as referenced in the digital output file header, can be a string of up to eight alphanumeric characters.
- **IS = val**: Initial state of the input element, must be a state defined in the model.

**Model Syntax:**

```
.MODEL mname U LEVEL=5 <parameters...>
```

### Digital-to-Analog Input Model Parameters

<table>
<thead>
<tr>
<th>Names (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLO</td>
<td>farad</td>
<td>0</td>
<td>Capacitance to low level node</td>
</tr>
<tr>
<td>CHI</td>
<td>farad</td>
<td>0</td>
<td>Capacitance to high level node</td>
</tr>
<tr>
<td>S0NAME</td>
<td></td>
<td></td>
<td>State “0” character abbreviation, can be a string of up to four alphanumeric characters.</td>
</tr>
<tr>
<td>S0TSW</td>
<td>sec</td>
<td></td>
<td>State “0” switching time</td>
</tr>
<tr>
<td>S0RLO</td>
<td>ohm</td>
<td></td>
<td>State “0” resistance to low level node</td>
</tr>
</tbody>
</table>
Up to 20 different states may be defined in the model definition by the SnNAME, SnTSW, SnRLO and SnRHI parameters, where n ranges from 0 to 19. The circuit representation of the element is shown in Figure 5-7.

![Figure 5-7: Digital-to-Analog Converter Element](image-url)
Example
The following example shows the usage of the U Element and model as a digital input for a Star-Hspice netlist.

* EXAMPLE OF U-ELEMENT DIGITAL INPUT
UC carry-in VLD2A VHD2A D2A SIGNAME=1 IS=0
VLO VLD2A GND DC 0
VHI VHD2A GND DC 1
.MODEL D2A U LEVEL=5 TIMESTEP=1NS,
  + S0NAME=x S0TSW=3NS S0RLO = 1K, S0RHI = 1K
  + S2NAME=x S2TSW=3NS S2RLO = 1K, S2RHI = 1K
  + S3NAME=x S3TSW=3NS S3RLO = 1MEG, S3RHI = 1MEG
  + S4NAME=x S4TSW=3NS S4RLO = 1MEG, S4RHI = 60
.PRINT V(carry-in)
.TRAN 1N 100N
.END

where the associated digital input file is:

1
00 1:1
09 z:1
10 0:1
11 z:1
20 1:1
30 0:1
39 x:1
40 1:1
41 x:1
50 0:1
60 1:1
70 0:1
80 1:1
Digital Outputs

The general syntax for including a digital output in a Star-Hspice output is:

**General form:**

\[ U<\text{name}> \text{ interface reference} \text{ mname SIGNAME} = \text{sname} \]

- **$Uxxx$** Digital output element name. Must begin with a “U”, which can be followed by up to 1023 alphanumeric characters.
- **interface** Interface node in the circuit at which the digital output is measured
- **reference** Node used as a reference for the output
- **mname** Digital output model reference (U model)
- **SIGNAME = sname** Signal name as referenced in the digital output file header, can be a string of up to eight alphanumeric characters.

**Model Syntax:**

\[ .MODEL \text{mname} \text{ U LEVEL}=4 <\text{parameters}...> \]

### Analog-to-Digital Output Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLOAD</td>
<td>ohm</td>
<td>1/gmin</td>
<td>Output resistance.</td>
</tr>
<tr>
<td>CLOAD</td>
<td>farad</td>
<td>0</td>
<td>Output capacitance.</td>
</tr>
<tr>
<td>S0NAME</td>
<td></td>
<td></td>
<td>State “0” character abbreviation, can be a string of up to four alphanumerical characters.</td>
</tr>
<tr>
<td>S0VLO</td>
<td>volt</td>
<td></td>
<td>State “0” low level voltage.</td>
</tr>
<tr>
<td>S0VHI</td>
<td>volt</td>
<td></td>
<td>State “0” high level voltage.</td>
</tr>
<tr>
<td>S1NAME</td>
<td></td>
<td></td>
<td>State “1” character abbreviation, can be a string of up to four alphanumerical characters.</td>
</tr>
</tbody>
</table>
Up to 20 different states may be defined in the model definition by the SnNAME, SnVLO and SnVHI parameters, where n ranges from 0 to 19. The circuit representation of the element is shown in Figure 5-8.

**Figure 5-8: Analog-to-Digital Converter Element**
Replacing Sources With Digital Inputs

Figure 5-9: Digital File Signal Correspondence

Traditional voltage sources...

V1 carry-in gnd PWL(0NS,lo 1NS,hi 7.5NS,hi 8.5NS,lo 15NS lo R
V2 A[0] gnd PWL (ONS,hi 1NS,lo 15.0NS,lo 16.0NS,hi 30NS hi R
V3 A[1] gnd PWL (ONS,hi 1NS,lo 15.0NS,lo 16.0NS,hi 30NS hi R
V4 B[0] gnd PWL (ONS,hi 1NS,lo 30.0NS,lo 31.0NS,hi 60NS hi
V5 B[1] gnd PWL (ONS,hi 1NS,lo 30.0NS,lo 31.0NS,hi 60NS hi

... become D2A drivers...

UC carry-in VLD2A VHD2A D2A SIGNAME=0 IS=0
UA[0] A[0] VLD2A VHD2A D2A SIGNAME=1 IS=1
UB[0] B[0] VLD2A VHD2A D2A SIGNAME=3 IS=1

... that get their input from the Digital stimulus file...
<designname>.d2a

Example

* EXAMPLE OF U-ELEMENT DIGITAL OUTPUT
VOUT carry_out GND PWL 0N 0V 1N 0V 11N 5V 19N 5V 2ON 0V
+ 30N 0V 31N 5V 39N 5V 40N 0V
VREF REF GND DC 0.0V
UCO carry-out REF A2D SIGNAME=12
* DEFAULT DIGITAL OUTPUT MODEL (no “X” value)
Using Sources and Stimuli

Digital and Mixed Mode Stimuli

```plaintext
.MODEL A2D U LEVEL=4 TIMESTEP=0.1NS TIMESCALE=1
  + S0NAME=0 S0VLO=-1 S0VHI= 2.7
  + S4NAME=1 S4VLO= 1.4 S4VHI=9.0
  + CLOAD=0.05pf
.TRAN 1N 50N
.END

and the digital output file should look like:

```
12
0 0:1
105 1:1
197 0:1
305 1:1
397 0:1
```

where the “12” represents the signal name, the first column is the time in units of 0.1 nanoseconds, and the second column has the signal value:name pairs. Subsequent outputs would be represented in the same file by more columns.

The following two-bit MOS adder uses the digital input file. In the following plot, nodes ‘A[0], A[1], B[0], B[1], and CARRY-IN’ all come from a digital file input (see Figure 5-9). SPICE outputs a digital file.

FILE: MOS2BIT.SP - ADDER - 2 BIT ALL-NAND-GATE BINARY ADDER

```plaintext
.OPTIONS ACCT NOMOD FAST scale=1u gmindc=100n post
.param lmin=1.25 hi=2.8v lo=.4v vdd=4.5
.global vdd

.TRAN .5NS 60NS
.MEAS PROP-DELAY TRIG V(carry-in) TD=10NS VAL=’vdd*.5’ RISE=1
  + TARG V(c[1]) TD=10NS VAL=’vdd*.5’ RISE=3

.MEAS PULSE-WIDTH TRIG V(carry-out_1) VAL=’vdd*.5’ RISE=1
  + TARG V(carry-out_1) VAL=’vdd*.5’ FALL=1

.MEAS FALL-TIME TRIG V(c[1]) TD=32NS VAL=’vdd*.9’ FALL=1
```
+ TARG V(c[1]) TD=32NS VAL='vdd*.1' FALL=1

* VDD vdd gnd DC vdd
X1 A[0] B[0] carry-in C[0] carry-out_1 ONEBIT

* Subcircuit Definitions
.subckt NAND in1 in2 out wp=10 wn=5
M1 out in1 vdd vdd P W=wp L=lmin ad=0
M2 out in2 vdd vdd P W=wp L=lmin ad=0
M3 out in1 mid gnd N W=wn L=lmin as=0
M4 mid in2 gnd gnd N W=wn L=lmin ad=0
CLOAD out gnd 'wp*5.7f'
.ends

* .subckt ONEBIT in1 in2 carry-in out carry-out
X1 in1 in2 #1_nand NAND
X2 in1 #1_nand 8 NAND
X3 in2 #1_nand 9 NAND
X4 8 9 10 NAND
X5 carry-in 10 half1 NAND
X6 carry-in half1 half2 NAND
X7 10 half1 13 NAND
X8 half2 13 out NAND
X9 half1 #1_nand carry-out NAND
.ends ONEBIT

* Stimulus
UC carry-in VLD2A VHD2A D2A SIGNAME=1 IS=0
UA[0] A[0] VLD2A VHD2A D2A SIGNAME=2 IS=1
UB[0] B[0] VLD2A VHD2A D2A SIGNAME=4 IS=1

Using Sources and Stimuli

Digital and Mixed Mode Stimuli

* uc0 c[0] vrefa2d a2d signame=10
  uc1 c[1] vrefa2d a2d signame=11
  uco carry-out_2 vrefa2d a2d signame=12
  uci carry-in vrefa2d a2d signame=13
*

* Models

*.MODEL N NMOS LEVEL=3 VTO=0.7 UO=500 KAPPA=.25 KP=30U
  + ETA=.01 THETA=.04 VMAX=2E5 NSUB=9E16 TOX=400 GAMMA=1.5
  + PB=0.6 JS=.1M XJ=0.5U LD=0.1U NFS=1E11 NSS=2E10
  + RSH=80 CJ=.3M MJ=0.5 CJSW=.1N MJSW=0.3
  + acm=2 capop=4
*

*.MODEL P PMOS LEVEL=3 VTO=-0.8 UO=150 KAPPA=.25 KP=15U
  + ETA=.015 THETA=.04 VMAX=5E4 NSUB=1.8E16 TOX=400 GAMMA=.672
  + PB=0.6 JS=.1M XJ=0.5U LD=0.15U NFS=1E11 NSS=2E10
  + RSH=80 CJ=.3M MJ=0.5 CJSW=.1N MJSW=0.3
  + acm=2 capop=4
*

* Default Digital Input Interface Model

*.MODEL D2A U LEVEL=5 TIMESTEP=0.1NS,
  + S0NAME=0 S0TSW=1NS S0RLO = 15, S0RHI = 10K,
  + S2NAME=x S2TSW=5NS S2RLO = 1K, S2RHI = 1K
  + S3NAME=z S3TSW=5NS S3RLO = 1MEG,S3RHI = 1MEG
  + S4NAME=1 S4TSW=1NS S4RLO = 10K, S4RHI = 60
  VLD2A VLD2A 0 DC lo
  VHD2A VHD2A 0 DC hi
*

* Default Digital Output Model (no “X” value)

*.MODEL A2D U LEVEL=4 TIMESTEP=0.1NS TIMESCALE=1
  + S0NAME=0 S0VLO=-1 S0VHI= 2.7
  + S4NAME=1 S4VLO= 1.4 S4VHI=6.0
  + CLOAD=0.05pf
Specifying a Digital Vector File

The digital vector file consists of three parts:

- **Vector Pattern Definition** section
- **Waveform Characteristics** section
- **Tabular Data** section.

To incorporate this information into your simulation, you need to include this line in your netlist:

```
.VEC 'digital_vector_file'
```
Defining Vector Patterns

The Vector Pattern Definition section defines the vectors—their names, sizes, signal direction, and so on—and must occur first in the digital vector file. A sample Vector Pattern Definition section follows:

\[
\text{radix } 1111 1111 \\
\text{vname } a \ b \ c \ d \ e \ f \ g \ h \\
\text{io } iiiiiiii \\
\text{tunit } ns
\]

Keywords such as \textit{radix}, \textit{vname} are explained in the “Defining Tabular Data” section later in this chapter.

Defining Waveform Characteristics

The Waveform Characteristics section defines various attributes for signals, such as the rise or fall time, thresholds for logic ‘high’ or ‘low’, and so on. A sample Waveform Characteristics section follows:

\[
\text{trise } 0.3 \ 137F \ 0000 \\
\text{tfall } 0.5 \ 137F \ 0000 \\
\text{vih } 5.0 \ 137F \ 0000 \\
\text{vil } 0.0 \ 137F \ 0000
\]

Using Tabular Data

The Tabular Data section defines the values of the input signals at specified times. The time is listed in the first column, followed by signal values, in the order specified by the \textit{vname} statement.

Example

An example of tabular data follows:

\[
11.0 \ 1000 \ 1000 \\
20.0 \ 1100 \ 1100 \\
33.0 \ 1010 \ 1001
\]
Comment Lines

A line beginning with a semi-colon “;” is considered a comment line. Comments may also start at any point along a line. Star-Hspice ignores characters following a semi-colon.

Example

An example of usage follows:

```
; This is a comment line
radix 1 1 4 1234 ; This is a radix line
```

Continuing a Line

Like netlists, a line beginning with a plus sign “+” is a continuation from the previous line.

Digital Vector File Example

An example of a vector pattern definition follows:

```
; specifies # of bits associated with each vector
radix 1 2 444
;****************************************************
; defines name for each vector. For multi-bit
; vectors, innermost [ ] provide the bit index range,
; MSB:LSB
vname v1 va[[1:0]] vb[12:1]
; actual signal names: v1, va[0], va[1], vb1 ... vb12
;****************************************************
; defines vector as input, output, or bi-direc
io i o bbb
; defines time unit
tunit ns
;****************************************************
; vb12-vb5 are output when ‘v1’ is ‘high’
enable v1 0 0 FF0
; vb4-vb1 are output when ‘v1’ is ‘low’
```
enable ~v1 0 0 00F

;*******************************************************
; all signals have delay of 1 ns
; Note: do not put unit (e.g., ns) again here because
; this value will be multiplied by the unit specified
; in the ‘tunit’ line.
tdelay 1.0
; signals va1 and va0 have delays of 1.5ns
tdelay 1.5 0 3 000
;*******************************************************
; specify input rise and fall times (if you want
; different rise and fall times, use trise/
tfallstmt.)
; Note: do not put unit (e.g., ns) again here because
; this value will be multiplied by the unit specified
; in the ‘tunit’ line.
slope 1.2
;*******************************************************
; specify the logic ‘high’ voltage for input signals
vih 3.3 1 0 000
vih 5.0 0 0 FFF
; likewise, may specify logic ‘low’ with ‘vil’
;*******************************************************
; va & vb switch from ‘lo’ to ‘hi’ at 1.75 volts
vth 1.75 0 1 FFF
;*******************************************************
; tabular data section
10.0 1 3 FFF
20.0 0 2 AFF
30.0 1 0 888
.
.
.
Defining Tabular Data

Although this section generally appears last in a digital vector file, following the Vector Pattern and Waveform Characteristics definitions, we describe it first to introduce the definitions of a vector.

The Tabular Data section defines (in tabular format) the values of the signals at specified times. Its general format is:

\[
\begin{align*}
time1 & \quad signal1\text{\_value1} & \quad signal2\text{\_value1} & \quad signal3\text{\_value1} \\
time2 & \quad signal1\text{\_value2} & \quad signal2\text{\_value2} & \quad signal3\text{\_value2} \\
time3 & \quad signal1\text{\_value3} & \quad signal2\text{\_value3} & \quad signal3\text{\_value3} \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ quad
Input Stimuli

Star-Hspice converts each input signal into a PWL (piecewise linear) voltage source and a series resistance. The legal states for an input signal are:

- 0: Drive to ZERO (gnd)
- 1: Drive to ONE (vdd)
- Z, z: Floating to HIGH IMPEDANCE
- X, x: Drive to ZERO (gnd)
- L: Resistive drive to ZERO (gnd)
- H: Resistive drive to ONE (vdd)
- U, u: Drive to ZERO (gnd)

For the 0, 1, X, x, U, u states, the resistance value is set to zero; for the L, H states, the resistance value is defined by the out (or outz) statement; and for the Z, z states, the resistance value is defined by the triz statement.

Expected Output

Star-Hspice converts each output signal into a .DOUT statement in the netlist. During simulation, Star-Hspice compares the actual results with the expected output vector(s), and if the states are different, an error message appears. The legal states for expected outputs include:

- 0: Expect ZERO
- 1: Expect ONE
- X, x: Don’t care
- U, u: Don’t care
- Z, z: Expect HIGH IMPEDANCE (don’t care)

Z, z are treated as “don’t care” because Star-Hspice cannot detect a high impedance state.
Example

An example of usage follows:

```verbatim
... ; start of tabular section data
11.0 1 0 0 1
20.0 1 1 0 0
30.0 1 0 0 0
35.0 x x 0 0
```

Verilog Value Format

Star-Hspice also accepts Verilog sized format for number specification:

`<size> '<base format> <number>`

The `<size>` specifies (in decimal) the number of bits, and `<base format>` indicates binary (’b or ’B), octal (’o or ’O), or hexadecimal (’h or ’H). Valid `<number>` fields are combinations of the characters 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Depending on the `<base format>` chosen, only a subset of these characters may be legal.

You may also use unknown values (X) and high impedance (Z) in the `<number>` field. An X or Z sets four bits in the hexadecimal base, three bits in the octal base, and one bit in the binary base.

If the most significant bit of a number is 0, X, or Z, the number is automatically extended (if necessary) to fill the remaining bits with (respectively) 0, X, or Z. If the most significant bit is 1, it is extended with 0.

Example

```verbatim
4'b1111
12'hABx
32'bZ
8'h1
```

Here we specify values for: a 4-bit signal in binary, a 12-bit signal in hexadecimal, a 32-bit signal in binary, and an 8-bit signal in hexadecimal.
Equivalents of these lines in non-Verilog format would be:

```
1111
AB xxxx
ZZZZ ZZZZ ZZZZ ZZZZ ZZZZ ZZZZ ZZZZ
1000 0000
```

**Periodic Tabular Data**

Very often tabular data is periodic, so it is unnecessary to specify the absolute time at every time point. When a user specifies the `period` statement, the tabular data section omits the absolute times (see “Using Tabular Data” on page 5-61 for details).

**Example**

```
radix 1111 1111
vname a b c d e f g h
io ii iii iii
unit ns
period 10
; start of vector data section
1000 1000
1100 1100
1010 1001
```

**Defining Vector Patterns**

The *Vector Pattern Definition* section defines the sequence or order for each vector stimulus, as well as any individual characteristics. The statements in this section (except the `radix` statement) might appear in any order, and all keywords are case-insensitive.
Radix Statement

The radix statement specifies the number of bits associated with each vector. Valid values for the number of bits range from 1 to 4.

<table>
<thead>
<tr>
<th># bits</th>
<th>Radix</th>
<th>Number System</th>
<th>Valid Digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>Binary</td>
<td>0, 1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>–</td>
<td>0 – 3</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>Octal</td>
<td>0 – 7</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>Hexadecimal</td>
<td>0 – F</td>
</tr>
</tbody>
</table>

Only one radix statement must appear in the file, and it must be the first noncomment line.

Example

This example illustrates two 1-bit signals followed by a 4-bit signal, followed by a 1-bit, 2-bit, 3-bit, 4-bit signals, and finally eight 1-bit signals.

```
; start of vector pattern definition section
radix 1 1 4 1234 1111 1111
```

Vname Statement

The vname statement defines the name of each vector. If not specified, a default name will be given to each signal: V1, V2, V3, and so on. If you define more than one vname statement, the last one overrules the previous one.

```
radix 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
vname V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12
```

Provide the range of the bit indices with a square bracket [ ] and a colon syntax:

```
[starting_index : ending index]
```

The vname name is required for each bit, and a single name may be associated with multiple bits (such as bus notation).
The bit order is MSB:LSB. This bus notation syntax may also be nested inside other grouping symbols such as <>, (), [], etc. The name of each bit will be *vname* with the index suffix appended.

**Example 1**

If you specify:

```
radix 2 4
vname VA[0:1] VB[4:1]
```

the resulting names of the voltage sources generated are:

```
VA0 VA1 VB4 VB3 VB2 VB1
```

where *VA0* and *VB4* are the MSBs and *VA1* and *VB1* are the LSBs.

**Example 2**

If you specify:

```
vname VA[[0:1]] VB<[4:1]>
```

the resulting names of the voltage sources are:

```
VA[0] VA[1] VB<4> VB<3> VB<2> VB<1>
```

**Example 3**

This example shows how to specify a single bit of a bus:

```
vname VA[[2:2]]
```

**Example 4**

This example generates signals A0, A1, A2, ... A23:

```
rادix 444444
vname A[0:23]
```
IO Statement

The io statement defines the type of each vector. The line starts with a keyword io and followed by a string of i, b, o, or u definitions indicating whether each corresponding vector is an input, bidirectional, output, or unused vector, respectively.

- i Input used to stimulate the circuit.
- o Expected output used to compare with the simulated outputs.
- b Star-Hspice ignores.

Example

If the io statement is not specified, all signals are assumed input signals. If you define more than one io statements, the last one overrules previous ones.

```
io i i i bbbb iiiioouu
```

Tunit Statement

The tunit statement defines the time unit in the digital vector file for period, tdelay, slope, trise, tfall, and absolute time. It must be:

- fs femto-second
- ps pico-second
- ns nano-second
- us micro-second
- ms milli-second

If you do not specify the tunit statement, the default time unit value is ns. If you define more than one tunit statement, the last one will overrule the previous one.
**Example**

The *tunit* statement in this example specifies that the absolute times in the *tabular data* section are 11.0ns, 20.0ns, and 33.0ns, respectively.

```
tunit ns
11.0 1000 1000
20.0 1100 1100
33.0 1010 1001
```

**Period and Tskip Statements**

The *period* statement defines the time interval for the *tabular data* section so that specifying the absolute time at every time point is not necessary. Thus, if a *period* statement is provided alone (without the *tskip* statement), the *tabular data* section contains only signal values, not absolute times. The time unit of *period* is defined by the *tunit* statement.

**Example**

In this example, the first row of the tabular data (1000 1000) is for time 0ns. The second row of the tabular data (1100 1100) is for time 10ns. The third row of the tabular data (1010 1001) is for time 20ns.

```
radius 1111 1111
period 10
1000 1000
1100 1100
1010 1001
```

The *tskip* statement specifies that the absolute time field in the tabular data is to be ignored. In this way, the absolute time field of each row may be kept in the tabular data (but ignored) when using the *period* statement.
**Example**

If you have:

```plaintext
radix 1111 1111
period 10
tskip
 11.0 1000 1000
 20.0 1100 1100
 33.0 1010 1001
```

the absolute times 11.0, 20.0 and 33.0 are ignored.

**Enable Statement**

The `enable` statement specifies the controlling signal(s) of bidirectional signals and is absolutely required for all bidirectional signals. If more than one `enable` statement exists, the last value will overrule the previous ones, and a warning message will be issued.

The syntax is a keyword `enable`, followed by the controlling signal name and the mask that defines the (bidirectional) signals to which `enable` applies.

The controlling signal of bi-directional signals must be an input signal with radix of 1. The bidirectional signals become output when the controlling signal is at state 1 (or high). If you wish to reverse this default control logic, you must start the control signal name with ‘~’.

**Example**

In this example, signals `x` and `y` are bidirectional, as defined by the ‘b’ in the `io` line. The first enable statement indicates that `x` (as defined by the position of ‘F’) becomes output when signal `a` is 1. The second enable specifies that bidirectional bus `y` becomes output when signal `a` is 0.

```plaintext
radix 144
io ibb
vname a x[3:0] y[3:0]
enable a 0 F 0
enable ~a 0 0 F
```
Modifying Waveform Characteristics

This section describes how to modify the waveform characteristics of your circuit.

Tdelay, Idelay, and Odelay Statements

The $tdelay$, $idelay$ and $odelay$ statements define the delay time of the signal relative to the absolute time of each row in the tabular data section; $idelay$ applies to the input signals, $odelay$ applies to the output signals, while $tdelay$ applies to both input and output signals.

The statement starts with a keyword $tdelay$ (or $idelay$, $odelay$) followed by a delay value, and then followed by a mask, which defines the signals to which the delay will be applied. If you do not provide a mask, the delay value will be applied to all the signals.

The time unit of $tdelay$, $idelay$ and $odelay$ is defined by the $tunit$ statement. Normally, you only need to use the $tdelay$ statement; only use the $idelay$ and $odelay$ statements to specify different input and output delay times for bi-directional signals. $idelay$ settings on output signals (or $odelay$ settings on input signals) are ignored with warning message issued.

More than one $tdelay$ ($idelay$, $odelay$) statement can be specified. If more than one $tdelay$ ($idelay$, $odelay$) statement is applied to a signal, the last value will overrule the previous ones, and a warning will be given. If you do not specify the signal delays by a $tdelay$ ($idelay$ or $odelay$) statement, Star-Hspice defaults to zero.

Example

The first $idelay$ statement indicates that all signals have the same delay time 1.0. The delay time of some signals are overruled by the subsequent $tdelay$ statements. The V2 and Vx signals have delay time -1.2, and V4 V5[0:1] V6[0:2] have a delay of 1.5. The V7[0:3] signals have an input delay time of 2.0 and an output delay time of 3.0.
Slope Statement

The *slope* statement specifies input signal rise/fall time, with the time unit defined by the *tunit* statement. You can specify the signals to which the *slope* applies using a mask. If the *slope* statement is not provided, the default slope value is 0.1 ns.

If you specify more than one *slope* statement, the last value will overrule the previous ones, and a warning message will be issued. The *slope* statement has no effect on the expected output signals. The rising time and falling time of a signal will be overruled if *trise* and *tfall* are specified.

Example

The first example indicates that the rising and falling times of all signals are 1.2 ns, whereas the second specifies a rising/falling time of 1.1 ns for the first, second, sixth, and seventh signal.

```
slope 1.2
slope 1.1 1100 0110
```
**Trise Statement**

The *trise* statement specifies the rise time of each input signal (for which the mask applies). The time unit of *trise* is defined by the *tunit* statement.

**Example**

If you do not specify the rising time of the signals by any *trise* statement, the value defined by the *slope* statement is used. If you apply more than one *trise* statements to a signal, the last value will overrule the previous ones, and a warning message will be issued.

```
trise 0.3
trise 0.5 0 1 1 137F 00000000
trise 0.8 0 0 0 0000 11110000
```

The *trise* statements have no effect on the expected output signals.

**Tfall Statement**

The *tfall* statement specifies the falling time of each input signal (for which the mask applies). The time unit of *tfall* is defined by the *tunit* statement.

**Example**

If you do not specify the falling time of the signals by a *tfall* statement, Star-Hspice uses the value defined by the *slope* statement. If you specify more than one *tfall* statement to a signal, the last value will overrule the previous ones, and a warning message will be issued.

```
tfall 0.5
tnfall 0.3 0 1 1 137F 00000000
tfall 0.9 0 0 0 0000 11110000
```

The *tfall* statements have no effect on the expected output signals.
Out /Outz Statements

The keywords *out* and *outz* are equivalent and specify the output resistance of each signal (for which the mask applies); *out* (or *outz*) applies to the input signals only.

Example

If you do not specify the output resistance of a signal by an *out* (or *outz*) statement, Star-Hspice uses the default (zero). If you specify more than one *out* (or *outz*) statement to a signal, Star-Hspice overrules the last value with the previous ones, and issues a warning message.

```
    out 15.1
    out 150 1 1 1 0000 00000000
    outz 50.5 0 0 0 137F 00000000
```

The *out* (or *outz*) statements have no effect on the expected output signals.

Triz Statement

The *triz* statement specifies the output impedance when the signal (for which the mask applies) is in *tristate*; *triz* applies to the input signals only.

Example

If you do not specify the *tristate* impedance of a signal by a *triz* statement, Star-Hspice assumes 1000M. If you apply more than one *triz* statement to a signal, the last value will override the previous ones, and Star-Hspice will issue a warning.

```
    triz 15.1M
    triz 150M 1 1 1 0000 00000000
    triz 50.5M 0 0 0 137F 00000000
```

The *triz* statements have no effect on the expected output signals.
Vih Statement

The vih statement specifies the logic high voltage of each input signal to which the mask applies.

Example

If you specify the logic high voltage of the signals by a vih statement, Star-Hspice assumes 3.3. If you apply more than one vih statements to a signal, the last value will overrule the previous ones, and Star-Hspice will issue a warning.

```
vih 5.0
vih 5.0 1 1 1 137F 00000000
vih 3.5 0 0 0 0000 11111111
```

The vih statements have no effect on the expected output signals.

Vil Statement

The vil statement specifies the logic low voltage of each input signal to which the mask applies.

Example

If you specify the logic low voltage of the signals by a vil statement, Star-Hspice assumes 0.0. If you apply more than one vil statement to a signal, the last value will overrule the previous ones, and Star-Hspice will issue a warning.

```
vil 0.0
vil 0.0 1 1 1 137F 11111111
```

The vil statements have no effect on the expected output signals.
Vref Statement

Similar to the tdelay statement, the vref statement specifies the name of the reference voltage for each input vector to which the mask applies; vref applies to the input signals only.

Example

If you have:

```plaintext
vname v1 v2 v3 v4 v5[1:0] v6[2:0] v7[0:3] v8 v9 v10
vref 0
vref 0 111 137F 000
vref vss 0 0 0 0000 111
```

When Star-Hspice implements it into the netlist, the voltage source realizes v1:

```plaintext
v1 V1 0 pwl(......)
```
as will v2, v3, v4, v5, v6, and v7. However, v8 will be realized by

```plaintext
V8 V8 vss pwl(......)
```
as will v9 and v10.

If you do not specify the reference voltage name of the signals by a vref statement, Star-Hspice assumes 0. If you apply more than one vref statement, the last value will overrule the previous ones, and Star-Hspice issues a warning. The vref statements have no effect on the output signals.

Vth Statement

Similar to the tdelay statement, the vth statement specifies the logic threshold voltage of each signals to which the mask applies; vth applies to the output signals only. The threshold voltage is used to decide the logic state of Star-Hspice’s output signals for comparison with the expected output signals.

Example

If you do not specify the threshold voltage of the signals by a vth statement, Star-Hspice assumes 1.65. If you apply more than one vth statements to a signal, the last value will overrule the previous ones, and Star-Hspice issues a warning.
vth 1.75
vth 2.5 1 1 1 137F 00000000
vth 1.75 0 0 0 0000 11111111

The \textit{vth} statements have no effect on the input signals.

**Voh Statement**

The \textit{voh} statement specifies the logic high voltage of each output signal to which the mask applies.

**Example**

If you do not specify the logic high voltage by a \textit{voh} statement, Star-Hspice assumes 3.3. If you apply more than one \textit{voh} statements to a signal, the last value will overrule the previous ones, and Star-Hspice issues a warning.

voh 4.75
voh 4.5 1 1 1 137F 00000000
voh 3.5 0 0 0 0000 11111111

The \textit{voh} statements have no effect on input signals.

\underline{Note:} If both \textit{voh} and \textit{vol} are not defined, Star-Hspice uses \textit{vth} (default or defined).

**Vol Statement**

The \textit{vol} statement specifies the logic low voltage of each output signal to which the mask applies.

**Example**

If you do not specify the logic low voltage by a \textit{vol} statement, Star-Hspice assumes 0.0. If you apply more than one \textit{vol} statements to a signal, the last value will overrule the previous ones, and Star-Hspice issues a warning.

vol 0.5
vol 0.5 1 1 1 137F 11111111
The `vol` statements have no effect on input signals.

**Note:** If both `voh` and `vol` are not defined, Star-Hpice uses `vth` (default or defined).
Chapter 6

Multi-Terminal Networks

This chapter covers various topics related to the S Element:

■ Using Scattering Parameter Element
Using Scattering Parameter Element

A new S Element, in conjunction with the generic frequency-domain model (.MODEL SP), provides a convenient way to describe a multi-terminal network. Currently, S (scattering) and Y parameters are supported. S Element can be used in AC and DC analyses.

In particular, the S parameter in S Element represents the generalized scattering parameter $S$ for a multi-terminal network, which is defined as:

$$v_{\text{ref}} = S \cdot v_{\text{inc}}.$$ 

where boldface lower-case and upper-case symbols denote vectors and matrices, respectively. $v_{\text{inc}}$ and $v_{\text{ref}}$ are the incident and reflected voltage wave vectors (see Figure 6-1). The S parameter can be converted to Y parameter using the following formula:

$$Y = Y_r (I - S)(I + S)^{-1}.$$ 

where $Y_r$ is the characteristic admittance matrix of the reference system, which is related to the characteristic impedance matrix $Z_r$ by:

$$Y_r = Z_r^{-1}.$$ 

Similarly, Y parameter can be converted to S parameter as follows:

$$S = (Y_r + Y)^{-1}(Y_r - Y).$$
Figure 6-1: Terminal Node Notation

Syntax

The syntax of the S Element is:

\[ Sxxx \text{ nd1 nd2 } \ldots \text{ ndN ndR FQMODEL=\text{name}} \ [\text{TYPE=\text{val}} \text{ Zo=\text{val}} \text{ Zof=\text{name}}] \]

- \text{nd1 nd2 } \ldots \text{ ndN} \quad \text{N signal nodes (see Figure 6-1).}
- \text{ndR} \quad \text{Reference node.}
- \text{FQMODEL} \quad \text{.MODEL statement of type sp, which defines the frequency behavior of S or Y parameter.}
- \text{TYPE} \quad \text{Parameter type:}
  - \text{S: scattering parameter (default)}
  - \text{Y: Y parameter}


**Zo**

Characteristic impedance value of the reference line (frequency-independent). For multi-terminal cases (\( N > 1 \)), the characteristic impedance matrix of the reference lines is assumed to be diagonal and its diagonal values are set to Zo. More general types of a reference line system can be specified using Zof. Default=50 \( \Omega \).


**Zof**

Name of the frequency-varying model that defines the frequency behavior of the reference system. If both Zo and Zof have been defined, then Zof has precedence.

---

**Frequency Table Model**

The Frequency Table Model is a generic model that can be used to describe frequency-varying behavior. Currently, it is used by S Element and .NOISENPT.

**Syntax**

The syntax of the .MODEL model card is:

```
.MOLDAL name sp [N=val FSTART=val FSTOP=val NI=val SPACING=val
 + MATRIX=val VALTYPE=val INFINITY=matrixval INTERPOLATION=val
 + EXTRAPOLATION=val] [DATA=(npts ...)] [DATAFILE=filename]
```

- **name**
  - Model name.

- **N**
  - Matrix dimension (number of signal terminals). Values other than 1 must be specified before setting INFINITY and DATA. Default=1.

- **FSTART**
  - Starting frequency point for data. Default=0.

- **FSTOP**
  - Final frequency point for data (used only for the LINEAR and LOG spacing formats).
NI

Number of frequency points per interval. Used only for the DEC and OCT spacing formats. Default=10.

npts

Number of data points.

SPACING

Data sample spacing format:

LIN (LINEAR): uniform spacing with the frequency step of \((FSTOP-FSTART)/(npts-1)\). Default.

OCT: octave variation with \(FSTART\) as the starting frequency and NI points per octave. \(npts\) determines the final frequency.

DEC: decade variation with \(FSTART\) as the starting frequency and NI points per decade. \(npts\) determines the final frequency.

LOG: logarithmic spacing with \(FSTART\) and \(FSTOP\) as the starting and final frequencies.

POI (NONUNIFORM): nonuniform spacing. Data points are paired with frequency points.
Matrix (data point) format:

**SYMMETRIC:** symmetric matrix. Only the lower-half triangle portion of a matrix is specified. Default.

**HERMITIAN:** similar to SYMMETRIC but off-diagonal terms are complex-conjugate of each other.

**NONSYMMETRIC:** nonsymmetric matrix. A full matrix is specified.

Data type of matrix elements:

**REAL:** real entry

**CARTESIAN:** complex number in real/imaginary format. Default.

**POLAR:** complex number in polar format. Angles are specified in radian.

Data point at infinity. Typically real-valued. Data format must be consistent with MATRIX and VALTYPE specifications. This point is not counted in npts.

Data port at DC. Typically real-valued. Data format must be consistent with MATRIX and VALTYPE specifications. This point is not counted in npts. It is required to specify DC point or data point at frequency=0.
INTERPOLATION

Interpolation scheme:

LINEAR: piecewise linear
SPLINE: b-spline curve fit

EXTRAPOLATION

Extrapolation scheme during simulation:

NONE: no extrapolation is allowed. Star-Hspice will terminate if data point is required outside of the specified range.
STEP: the last boundary point is used. Default.
LINEAR: linear extrapolation using the last two boundary points.

If the data point at the infinity is specified, then no extrapolation is performed and the infinity value is used.

DATA

Specifies data points. Syntax for LIN spacing:

```
.MMODEL name sp SPACING=LIN [N=dim] + FSTART=f0 DF=f1 DATA=npts d1 d2 ...
```

Syntax for OCT or DEC spacing:

```
.MMODEL name sp SPACING=DEC or OCT + [N=dim] FSTART=f0 + NI=n_per_intval DATA=npts d1 + d2 ...
```

Syntax for POI spacing:

```
.MMODEL name sp SPACING=NONUNIFORM + [N=dim] DATA=npts f1 d1 f2 d2 + ...
```
**Note:** The interpolation and extrapolation are performed after S parameter data are converted to Y parameter internally.

**Example**

The two outputs from the resistor and S parameter modeling should be matched exactly in this example. See Table 6-1 for the input file listing and Figure 6-2 for an illustration of a transmission line with a resistive termination.

**Figure 6-2: Transmission line with a resistive termination**
### Table 6-1: Input File Listing

| Header, options, and sources | *S parameter ex1: x-line with a resistive + termination .OPTIONS POST V1 i1 0 ac=1v |
| Analysis | .AC lin 500 OHz 30MegHz .DC v1 0v 5v 1v |
| Transmission line (W element) | W1 i1 i2 i3 0 o1 o2 o3 0 RLGCMODEL=wr1gc N=3 L=0.97 .MODEL wr1gc sp MODELTYPE=RLGC N=3 + Lo = 2.78310e-07 + 8.75304e-08 3.29391e-07 + 3.65709e-08 1.15459e-07 3.38629e-07 + Co = 1.41113e-10 + -2.13558e-11 9.26469e-11 + -8.92852e-13 -1.77245e-11 8.72553e-11 |
| Termination | x1 o1 o2 o3 0 terminator |
| Frequency model definition | .MODEL fmod sp N=3 FSTOP=30MegHz + DATA= 1 + 0.322825 0.0 -0.41488 0.0 + 0.17811 0.0 0.322825 0.0 -0.270166 0.0 |
| Resistor elements | .SUBCKT terminator n1 n2 n3 ref R1 n1 ref 75 R2 n2 ref 75 R3 n3 ref 75 R12 n1 n2 25 R23 n2 n3 25 .ends terminator |
| Equivalent S parameter element | .ALTER S parameter case .SUBCKT terminator n1 n2 n3 ref S1 n1 n2 n3 ref FQMODEL=fmod .ENDS terminator .END |
This is an example of a transmission line with a capacitive network termination.

### Table 6-2: Input File Listing

<table>
<thead>
<tr>
<th>Frequency model definition</th>
<th>.MODEL fmod sp N=3 FSTOP=30MegHz + DATA= 2 + 1.0 0.0 + 0.0 0.0 1.0 0.0 + 0.0 0.0 0.0 0.0 1.0 0.0 + 0.97409 -0.223096 + 0.00895303 0.0360171 0.964485 -0.25887 + -0.000651487 0.000242442 0.00895303 0.0360171 + 0.97409 -0.223096</th>
</tr>
</thead>
</table>

Using capacitive elements

<table>
<thead>
<tr>
<th>.SUBCKT terminator n1 n2 n3 ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1  n1 ref 10pF</td>
</tr>
<tr>
<td>C2  n2 ref 10pF</td>
</tr>
<tr>
<td>C3  n3 ref 10pF</td>
</tr>
<tr>
<td>C12 n1 n2  2pF</td>
</tr>
<tr>
<td>C23 n2 n3  2pF</td>
</tr>
<tr>
<td>.ENDS terminator</td>
</tr>
</tbody>
</table>

The two outputs from the resistor and S parameter modeling will differ slightly due to the linear frequency dependency related to the capacitor. This difference can be removed by using the linear interpolation scheme in .MODEL.

This is an example of a transmission line with S parameter.

### Figure 6-3: 3-Conductor Transmission line
## Table 6-3: Input File Listing

<table>
<thead>
<tr>
<th>Header, options and sources</th>
<th>*S parameter ex3: modeling x-line using + S parameter .OPTIONS POST vin in0 0 ac=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td>.AC lin 100 0 1000meg .DC vin 0 1v 0.2v</td>
</tr>
<tr>
<td>Transmission line</td>
<td>x1 in1 in2 out1 out2 0 x-line</td>
</tr>
<tr>
<td>Termination</td>
<td>R1 in0 in1 28</td>
</tr>
<tr>
<td></td>
<td>R2 in2 0 28</td>
</tr>
<tr>
<td></td>
<td>R3 out1 0 28</td>
</tr>
<tr>
<td></td>
<td>R4 out2 0 28</td>
</tr>
<tr>
<td>W Element RLGC model definition</td>
<td>.MODEL m2 W ModelType=RLGC, N=2 + Lo= 0.178e-6 + 0.0946e-7 0.178e-6 + Co= 0.23e-9 + -0.277e-11 0.23e-9 + Ro= 0.97 + 0 0.97 + Go= 0 + 0 0 + Rs= 0.138e-3 + 0 0.138e-3 + Gd= 0.29e-10 + 0 0.29e-10</td>
</tr>
<tr>
<td>Frequency model definition</td>
<td>.MODEL SM2 sp N=4 FSTART=0 FSTOP=1e+09 + SPACING=LINEAR + DATA= 60 + 0.00386491 0 + 0 0 0.00386491 0 + 0.996135 0 0 0 0.00386491 0 + 0.00492864 -0.15301 + 0.00188102 0.0063569 -0.0492864 -0.15301 + 0.926223 -0.307306 0.000630484 -0.00154619 + 0.0492864 -0.15301 + 0.000630484 -0.00154619 0.926223 -0.307306 + 0.00188102 0.0063569 -0.0492864 -0.15301 + -0.175236 -0.241602 + 0.00597 0.0103297 -0.175236 -0.241602 + 0.761485 -0.546979 0.00093508 -0.00508414 + -0.175236 -0.241602 + 0.00093508 -0.00508414 0.761485 -0.546979 + 0.00597 0.0103297 -0.175236 -0.241602 + ...</td>
</tr>
<tr>
<td>Equivalent S parameter element</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>.SUBCKT terminator n1 n2 n3 ref S1 n1 n2 n3 ref FQMODEL=SM2</td>
<td></td>
</tr>
<tr>
<td>.ENDS terminator</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 7
Parameters and Functions

Parameters are similar to variables found in most programming languages. They hold a value that is either assigned at design time or calculated during the simulation based on circuit solution values. Parameters are used for storage of static values for a variety of quantities (resistance, source voltage, rise time, and so on), or used in sweep or statistical analysis.

This chapter describes the basic usage of parameters within a Star-Hspice netlist:

- Using Parameters in Simulation (.PARAM)
- Using Algebraic Expressions
- Built-In Functions
- Parameter Scoping and Passing
Using Parameters in Simulation

Parameter Definition

Parameters in Star-Hspice are names that have associated numeric values. You can use any of the following methods to define parameters:

<table>
<thead>
<tr>
<th>Method</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple assignment</td>
<td>.PARAM &lt;SimpleParam&gt; = 1e-12</td>
</tr>
<tr>
<td>Algebraic definition</td>
<td>.PARAM &lt;AlgebraicParam&gt; = ‘SimpleParam*8.2’</td>
</tr>
<tr>
<td>User-defined function</td>
<td>.PARAM &lt;MyFunc( x, y )&gt; = ‘$sqrt((x<em>x)+(y</em>y))’</td>
</tr>
<tr>
<td>Subcircuit default</td>
<td>.SUBCKT &lt;SubName&gt; &lt;ParamDefName&gt; = + &lt;Value&gt;</td>
</tr>
<tr>
<td>Predefined analysis function</td>
<td>.PARAM &lt;mcVar&gt; = Agauss(1.0,0.1) (see “Statistical Analysis and Optimization” on page 13-1)</td>
</tr>
<tr>
<td>.MEASURE statement</td>
<td>.MEASURE &lt;DC</td>
</tr>
</tbody>
</table>

A parameter definition in Star-Hspice always takes the last value found in the Star-Hspice input (subject to local versus global parameter rules). Thus, the definitions below assign the value 3 to the parameter DupParam.

```
.PARAM DupParam = 1
...
.PARAM DupParam = 3
```

The value 3 will be substituted for all instances of DupParam, including instances that occur earlier in the input than the .PARAM DupParam = 3 statement.

All parameter values in Star-Hspice are IEEE double floating point numbers.
Parameter resolution order is as follows:

1. Resolve all literal assignments
2. Resolve all expressions
3. Resolve all function calls

Parameter passing order is shown in Table 7-1.

**Table 7-1: Parameter Passing Order**

<table>
<thead>
<tr>
<th>.OPTION PARHIER = GLOBAL</th>
<th>.OPTION PARHIER = LOCAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis sweep parameters</td>
<td>Analysis sweep parameters</td>
</tr>
<tr>
<td>.PARAM statement (library)</td>
<td>.SUBCKT call (instance)</td>
</tr>
<tr>
<td>.SUBCKT call (instance)</td>
<td>.SUBCKT definition (symbol)</td>
</tr>
<tr>
<td>.SUBCKT definition (symbol)</td>
<td>.PARAM statement (library)</td>
</tr>
</tbody>
</table>

**Parameter Assignment**

A constant real number or an algebraic expression of real values, predefined function, user-defined function, or circuit or model values can be assigned to parameters. A complex expression must be enclosed in single quotes in order to invoke the Star-Hspice algebraic processor. A simple expression consists of a single parameter name. The parameter keeps the assigned value unless there is a later definition that changes its value, or it is assigned a new value by an algebraic expression during simulation. There is no warning if a parameter is reassigned.

**Syntax**

```
.PARAM <ParamName> = <RealNumber>
.PARAM <ParamName> = ’<Expression>’ $ Quotes are mandatory
.PARAM <ParamName1> = <ParamName2> $ Cannot be recursive!
```

**Numerical Example**

```
.PARAM TermValue = lg
rTerm Bit0 0 TermValue
```
Expression Example

```
.PARAM Pi = '355/113'
.PARAM Pi2 = '2*Pi'

.PARAM npRatio = 2.1
.PARAM nWidth = 3u
.PARAM pWidth = 'nWidth * npRatio'
Mpl ... <pModelName> W = pWidth
Mnl ... <nModelName> W = nWidth
...
```

Inline Assignments

To define circuit values by a direct algebraic evaluation:
```
r1 n1 0 R = '1k/sqrt(HERTZ)' $ Resistance related to frequency.
```

Parameters in Output

To use an algebraic expression as an output variable in a .PRINT, .PLOT, or .PROBE statement, use the PAR keyword (see “Specifying Simulation Output” on page 8-1 for more information on simulation output). For example:
```
.PRINT DC v(3) gain = PAR('v(3)/v(2)') PAR('v(4)/v(2)')
```

User-Defined Function Parameters

A user-defined function can be defined similar to the parameter assignment except for the fact that it cannot be nested more than three deep.

Syntax

```
.PARAM <ParamName>(<pv1>[, <pv2>]) = '<Expression>'
```

Example

```
.PARAM CentToFar (c) = '(((c*9)/5)+32)'
.PARAM F(p1,p2) = 'Log(Cos(p1)*Sin(p2))'
.PARAM SqrdProd (a,b) = '(a*a)*(b*b)'
```
Subcircuit Default Definitions

The specification of hierarchical subcircuits allows you to pick default values for circuit elements. This is typically used in cell definitions so the circuit can be simulated with typical values (see “Using Subcircuits” on page 3-47 for more information on subcircuits).

Syntax

```
.SUBCKT <SubName> <PinList> [<SubDefaultsList>]
```

where `<SubDefaultsList>` is

```
<SubParam1> = <Expression> [<SubParam2> = <Expression> ...]
```

Subcircuit Parameter Example

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

```
.SUBCKT Inv a y Strength = 3
    Mp1 <MosPinList> pMosMod L = 1.2u W = 'Strength * 2u'
    Mn1 <MosPinList> nMosMod L = 1.2u W = 'Strength * 1u'
.ENDS
```

Parameter Scoping Example

The following example shows explicitly the difference between local and global scoping for parameter usage in subcircuits.

Given the input netlist fragment:

```
.PARAM DefPwid = 1u

.SUBCKT Inv a y DefPwid = 2u DefNwid = 1u
    Mp1 <MosPinList> pMosMod L = 1.2u W = DefPwid
    Mn1 <MosPinList> nMosMod L = 1.2u W = DefNwid
.ENDS
```

xInv0 a y0 Inv $ Default devices: p device = 6u, n device = 3u
xInv1 a y1 Inv Strength = 5$ p device = 10u, n device = 5u
xInv2 a y2 Inv Strength = 1$ p device =  2u, n device = 1u
...
with the global parameter scoping option .OPTION PARHIER = GLOBAL set, and the following input statements

```
... xInv0 a y0 Inv $ Xinv0.Mp1 width = 1u
xInv1 a y1 Inv DefPwid = 5u $ Xinv1.Mp1 width = 5u
.MEASURE TRAN Wid0 PARAM = ’lv2(xInv0.Mp1)’ $ lv2 is the
.MEASURE TRAN Wid1 PARAM = ’lv2(xInv1.Mp1)’ $ template for the
   $ channel width
   $’lv2(xInv1.Mp1)’
...
```

the following results are produced in the listing file:

```
wid0 = 1.0000E-06
wid1 = 1.0000E-06
```

With the local parameter scoping option .OPTION PARHIER = LOCAL set, and the following statements

```
... xInv0 a y0 Inv $ Xinv0.Mp1 width = 1u
xInv1 a y1 Inv DefPwid = 5u $ Xinv1.Mp1 width = 1u:
.MEASURE TRAN Wid0 PARAM = ’lv2(xInv0.Mp1)’
$ override the global .PARAM
.MEASURE TRAN Wid1 PARAM = ’lv2(xInv1.Mp1)’
...
```

the following results are produced in the listing file:

```
wid0 = 2.0000E-06
wid1 = 5.0000E-06
```

**Predefined Analysis Function**

Star-Hspice has specialized analysis types, primarily Optimization and Monte Carlo, that require a method of controlling the analysis. The parameter definitions related with these analysis types are described in “Statistical Analysis and Optimization” on page 13-1.

**Measurement Parameters**

.MEASURE statements in Star-Hspice produce a type of parameter called a measurement parameter. In general, the rules for measurement parameters are
the same as the rules for standard parameters, with one exception: measurement parameters are not defined in a .PARAM statement, but are defined directly in a .MEASURE statement. The detailed syntax and usage of the .MEASURE statement is described in “Specifying User-Defined Analysis (.MEASURE)” on page 8-37.

Multiply Parameter

The multiply parameter \( M \) is a special keyword common to all elements (except for voltage sources) and subcircuits. It multiplies the internal component values to give the effect of making parallel copies of the element or subcircuit. To simulate the effect of 32 output buffers switching simultaneously, only one subcircuit call needs to be placed, such as:

\[
X1 \text{ in out buffer } M = 32
\]

Multiply works hierarchically. A subcircuit within a subcircuit is multiplied by the product of the multiply parameters at both levels.

**Figure 7-1: Multiply Parameters Simplify Flip-Flop Initialization**
Using Algebraic Expressions

You can replace any parameter defined in the netlist by an algebraic expression with quoted strings. Then, these expressions may be used as output variables in the .PLOT, .PRINT, and .GRAPH statements. Using algebraic expressions can expand your options in the input netlist file. Some usages of algebraic expressions are:

- **Parameterization:**
  
  ```
  .PARAM x = 'y+3'
  ```

- **Functions:**
  
  ```
  .PARAM rho(leff,weff) = '2+*leff*weff-2u'
  ```

- **Algebra in elements:**
  
  ```
  R1 1 0 r = 'ABS(v(1)/i(m1))+10'
  ```

- **Algebra in .MEASURE statements:**
  
  ```
  .MEAS vmax MAX V(1)
  .MEAS imax MAX I(q2)
  .MEAS ivmax PARAM = 'vmax*imax'
  ```

- **Algebra in output statements:**
  
  ```
  .PRINT conductance = PAR('i(m1)/v(22)')
  ```

### Algebraic Expressions for Output

**Syntax**

```
PAR('algebraic expression')
```

In addition to using quotations, the expression must be defined inside the PAR( ) statement for output. The continuation character for quoted parameter strings is a double backslash, "\\". (Outside of quoted strings, the single backslash, "\", is the continuation character.)

---

1. Star-Hspice uses double-precision numbers (15 digits) for expressions, user-defined parameters, and sweep variables. For better precision, use parameters instead of constants in algebraic expressions, since constants are only single-precision numbers (7 digits).
Built-In Functions

In addition to simple arithmetic operations (+, -, *, /), Star-Hspice provides a number of built-in functions that you can use in expressions. The Star-Hspice built-in functions are listed in Table 7-2.

Table 7-2: Star-Hspice Built-in Functions

<table>
<thead>
<tr>
<th>HSPICE Form</th>
<th>Function</th>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sin(x)</td>
<td>sine</td>
<td>trig</td>
<td>Returns the sine of x (radians)</td>
</tr>
<tr>
<td>cos(x)</td>
<td>cosine</td>
<td>trig</td>
<td>Returns the cosine of x (radians)</td>
</tr>
<tr>
<td>tan(x)</td>
<td>tangent</td>
<td>trig</td>
<td>Returns the tangent of x (radians)</td>
</tr>
<tr>
<td>asin(x)</td>
<td>arc sine</td>
<td>trig</td>
<td>Returns the inverse sine of x (radians)</td>
</tr>
<tr>
<td>acos(x)</td>
<td>arc cosine</td>
<td>trig</td>
<td>Returns the inverse cosine of x (radians)</td>
</tr>
<tr>
<td>atan(x)</td>
<td>arc tangent</td>
<td>trig</td>
<td>Returns the inverse tangent of x (radians)</td>
</tr>
<tr>
<td>sinh(x)</td>
<td>hyperbolic sine</td>
<td>trig</td>
<td>Returns the hyperbolic sine of x (radians)</td>
</tr>
<tr>
<td>cosh(x)</td>
<td>hyperbolic cosine</td>
<td>trig</td>
<td>Returns the hyperbolic cosine of x (radians)</td>
</tr>
<tr>
<td>tanh(x)</td>
<td>hyperbolic tangent</td>
<td>trig</td>
<td>Returns the hyperbolic tangent of x (radians)</td>
</tr>
<tr>
<td>abs(x)</td>
<td>absolute value</td>
<td>math</td>
<td>Returns the absolute value of x:</td>
</tr>
<tr>
<td>sqrt(x)</td>
<td>square root</td>
<td>math</td>
<td>Returns the square root of the absolute value of x: \sqrt(-x) = -sqrt(</td>
</tr>
<tr>
<td>pow(x,y)</td>
<td>absolute power</td>
<td>math</td>
<td>Returns the value of x raised to the integer part of y: x^{(integer part of y)}</td>
</tr>
</tbody>
</table>
### Table 7-2: Star-Hspice Built-in Functions

<table>
<thead>
<tr>
<th>HSPICE Form</th>
<th>Function</th>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pwr(x,y)</td>
<td>signed power</td>
<td>math</td>
<td>Returns the absolute value of x raised to the y power, with the sign of x: (sign of x)</td>
</tr>
<tr>
<td>log(x)</td>
<td>natural logarithm</td>
<td>math</td>
<td>Returns the natural logarithm of the absolute value of x, with the sign of x: (sign of x)</td>
</tr>
<tr>
<td>log10(x)</td>
<td>base 10 logarithm</td>
<td>math</td>
<td>Returns the base 10 logarithm of the absolute value of x, with the sign of x: (sign of x)</td>
</tr>
<tr>
<td>exp(x)</td>
<td>exponential</td>
<td>math</td>
<td>Returns e raised to the power x: e^x</td>
</tr>
<tr>
<td>db(x)</td>
<td>decibels</td>
<td>math</td>
<td>Returns the base 10 logarithm of the absolute value of x, multiplied by 20, with the sign of x: (sign of x)</td>
</tr>
<tr>
<td>int(x)</td>
<td>integer</td>
<td>math</td>
<td>Returns the integer portion of x. The fractional portion of the number is lost.</td>
</tr>
<tr>
<td>sgn(x)</td>
<td>return sign</td>
<td>math</td>
<td>Returns -1 if x is less than 0, 0 if x is equal to 0, and 1 if x is greater than 0</td>
</tr>
<tr>
<td>sign(x,y)</td>
<td>transfer sign</td>
<td>math</td>
<td>Returns the absolute value of x, with the sign of y: (sign of y)</td>
</tr>
<tr>
<td>min(x,y)</td>
<td>smaller of two args</td>
<td>control</td>
<td>Returns the numeric minimum of x and y</td>
</tr>
<tr>
<td>max(x,y)</td>
<td>larger of two args</td>
<td>control</td>
<td>Returns the numeric maximum of x and y</td>
</tr>
<tr>
<td>lv(&lt;Element&gt;) or lx(&lt;Element&gt;)</td>
<td>element templates</td>
<td>various</td>
<td>Returns various element values during simulation. See “Element Template Output” in Chapter 7 for more information.</td>
</tr>
</tbody>
</table>
Star-Hspice reserves the variable names listed in Table 7-3 for use in elements such as E, G, R, C, and L. You cannot use them for any other purpose in your netlist (in .PARAM statements, for example).

### Table 7-3: Star-Hspice Special Variables

<table>
<thead>
<tr>
<th>HSPICE Form</th>
<th>Function</th>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>current simulation time</td>
<td>control</td>
<td>Parameterizes the current simulation time during transient analysis.</td>
</tr>
<tr>
<td>temper</td>
<td>current circuit temperature</td>
<td>control</td>
<td>Parameterizes the current simulation temperature during transient/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>temperature analysis.</td>
</tr>
<tr>
<td>hertz</td>
<td>current simulation frequency</td>
<td>control</td>
<td>Parameterizes the frequency during AC analysis.</td>
</tr>
</tbody>
</table>

### User-Defined Functions

An expression can contain parameters that have not yet been defined. A function must have at least one argument, and not more than two. Functions can be redefined.

**Syntax**

\[
\text{fname1 (arg1, arg2) = expr1 (fname2 (arg1, ...) = expr2) off}
\]
where:

\(.fname\) Specifies function name. This parameter must be distinct from array names and built-in functions. Subsequently defined functions must have all their embedded functions previously defined.

\(arg1, arg2\) Specifies variables used in the expression.

\(off\) Voids all user-defined functions.

**Example**

\[f(a,b) = \text{POW}(a,2)+a*b\] \[g(d) = \text{SQRT}(d)\] \[h(e) = e*f(1,2) - g(3)\]
Parameter Scoping and Passing

Parameterized subcircuits provide a method of reducing the number of similar cells that must be created to provide enough functionality within your library. Star-Hspice allows you to pass circuit parameters into hierarchical designs, allowing you to configure a cell at runtime.

For example, if you parameterize the initial state of a latch in its subcircuit definition, then you can override this initial default in the instance call. Only one cell needs to be created to handle both initial state versions of the latch.

You also can parameterize a cell to reflect its layout. Parameterize a MOS inverter to simulate a range of inverter sizes with only one cell definition. In addition, you can perform Monte Carlo analysis or optimization on a parameterized cell.

The way you choose to handle hierarchical parameters depends on how you construct and analyze your cells. You can choose to construct a design in which information flows from the top of the design down into the lowest hierarchical levels. Centralizing the control at the top of the design hierarchy involves setting global parameters. You can also choose to construct a library of small cells that are individually controlled from within by setting local parameters, and build upwards to the block level.

This section describes the scope of Star-Hspice parameter names, and how Star-Hspice resolves naming conflicts between levels of hierarchy.

Library Integrity

Integrity is a fundamental requirement for any symbol library. Library integrity can be as simple as a consistent, intuitive name scheme, or as complex as libraries with built-in range checking.

You can risk poor library integrity when using libraries from different vendors in a single design. Because there is no standardization between vendors on what circuit parameters are named, it is possible that two components can include the same parameter name with different functions. Suppose that the first vendor builds a library that uses the name $\text{Tau}$ as a parameter to control one or more
subcircuits in their library. Now suppose that the second vendor uses \textit{Tau} to control a different aspect of their library. Setting a global parameter named \textit{Tau} to control one library also modifies the behavior of the second library, which might not be the intent.

When the \textit{scope} of a higher level parameter is \textit{global} to all subcircuits at lower levels of the design hierarchy, lower level parameter values are overridden by the values from higher level definitions if the names are the same. The scope of a lower level parameter is \textit{local} to the subcircuit in which the parameter is defined (but global to all subcircuits that are even lower in the design hierarchy). The local scoping rules in Star-Hspice solve the problem of lower level parameters being overridden by higher level parameters of the same name when that is not desired.

**Reusing Cells**

Problems with parameter names also occur when different groups collaborate on a design. Because Star-Hspice global parameters prevail over local parameters, all designers are required to know the names of all parameters, even those used in sections of the design for which they are not responsible. This can lead to a large investment in standardized libraries. You can avoid this situation by using local parameter scoping that encapsulates all information about a section of a design within that section.

**Creating Parameters in a Library**

To ensure that critical, user-supplied parameters are present in a Star-Hspice netlist at simulation time, Star-Hspice allows the use of “illegal defaults”—that is, defaults that cause the simulator to abort if there are no overrides for the defaults.

Library cells that include illegal defaults require that the user provide a value for each and every instance of those cells. Failure to do so causes the Star-Hspice simulation to abort.

An example is the use of a default MOSFET width of 0.0. This causes Star-Hspice to abort because this parameter is required by the Star-Hspice MOSFET models.
Consider the following examples:

**Example**

...  
* Subcircuit default definition  
.SUBCKT Inv A Y Wid = 0 $ Inherit illegal values by default  
  mp1 <NodeList> <Model> L = 1u W = ’Wid*2’  
  mn1 <NodeList> <Model> L = 1u W = Wid  
.ENDS  
* Invocation of symbols in a design  
  x1 A Y1 Inv $ Bad! No widths specified  
  x2 A Y2 Inv Wid = 1u $ Overrides illegal value for Wid  
This simulation would abort on subcircuit instance *x1* because the required parameter *Wid* is never set on the subcircuit instance line. Subcircuit *x2* would simulate correctly. Additionally, the instances of the *Inv* cell are subject to accidental interference because the global parameter *Wid* is exposed outside the domain of the library. Anyone could have specified an alternative value for the parameter in another section of the library or the design, which could have prevented the simulation from catching the condition present on *x1*.

**Example**

Now consider the effect of a global parameter whose name conflicts with the library internal parameter *Wid*. Such a global parameter could be specified by the user or in a different library. In this example, the user of the library has specified a different meaning for the parameter *Wid* to be used in the definition of an independent source.

.Param Wid = 5u $ Default Pulse Width for source  
v1 Pulsed 0 Pulse ( 0v 5v 0u 0.1u 0.1u Wid 10u )  
...  
* Subcircuit default definition  
.SubCkt Inv A Y Wid = 0 $ Inherit illegals by default  
  mp1 <NodeList> <Model> L = 1u W = ’Wid*2’  
  mn1 <NodeList> <Model> L = 1u W = Wid  
.Ends  
* Invocation of symbols in a design  
  x1 A Y1 Inv $ Incorrect width!
Under global parameter scoping rules, the simulation succeeds, although incorrectly. There is no warning message that the inverter x1 has no widths assigned, because the global parameter definition for \textit{Wid} overrides the subcircuit default.

\textbf{Note:} Similarly, sweeping with different values of Wid dynamically changes both the Wid library internal parameter value and the pulse width value to the current sweep’s Wid value.

In global scoping, the highest level name prevails in name conflict resolution. In local scoping, the lowest level name is used.

The parameter inheritance method allows you to specify that local scoping rules be used. This feature can cause different results than you have obtained with Star-Hspice releases prior to 95.1 on existing circuits.

With local scoping rules, the Example 2 netlist correctly aborts in x1 for W = 0 (default Wid = 0 in the .SUBCKT definition has higher precedence than the .PARAM statement), and results in the correct device sizes for x2. This change might affect your simulation results if a circuit like the second one shown above is created intentionally or accidentally.

As an alternative to width testing in the Example 2 netlist, it is also possible to achieve a limited version of library integrity with the .OPTION DEFW. This option specifies the default width for all MOS devices during a simulation. Because part of the definition is still in the domain of the top-level circuit, this method still suffers from the possibility of making unwanted changes to library values without notification by the simulator.
Table 7-4 outlines and compares the three primary methods for configuring libraries to achieve required parameter checking in the case of default MOS transistor widths.

**Table 7-4: Methods for Configuring Libraries**

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter Location</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>On a .SUBCKT definition line</td>
<td>The library is protected from global circuit parameter definitions unless the user wishes to override. Single location for default values.</td>
<td>Cannot be used with releases of Star-Hspice prior to Release 95.1.</td>
</tr>
<tr>
<td>Global</td>
<td>At the global level and on .SUBCKT definition lines</td>
<td>Works with older Star-Hspice versions</td>
<td>The library can be changed by indiscrete user or other vendor assignment and by the intervening hierarchy. Cannot override a global value at a lower level.</td>
</tr>
<tr>
<td>Special</td>
<td>.OPTION DEFW statement</td>
<td>Simple to do</td>
<td>Third party libraries or other sections of the design might depend on the option DEFW.</td>
</tr>
</tbody>
</table>

**Parameter Defaults and Inheritance**

Use the .OPTION parameter PARHIER to specify scoping rules.

**Syntax**

\[ .OPTION\ PARHIER = < GLOBAL | LOCAL > \]

The default setting is GLOBAL, which gives the same scoping rules that Star-Hspice used prior to Release 95.1.
Figure 7-2 shows a flat representation of a hierarchical circuit that contains three resistors.

Each of the three resistors gets its simulation time resistance from the parameter named Val. The Val parameter is defined in four places in the netlist, with three different values.

**Figure 7-2: Hierarchical Parameter Passing Problem**

There are two possible solutions for the total resistance of the chain: 0.3333 Ω and 0.5455 Ω.

The PARHIER option allows you to specify which parameter value prevails when parameters with the same name are defined at different levels of the design hierarchy.

Under global scoping rules, in the case of name conflicts, the top-level assignment .PARAM Val = 1 overrides the subcircuit defaults, and the total is 0.3333 Ω. Under local scoping rules, the lower level assignments prevail, and the total is 0.5455 Ω (one, two and three ohms in parallel).
The example shown in Figure 7-2 produces the results in Table 7-5, based on the setting of the local/global PARHIER option:

**Table 7-5: PARHIER = LOCAL vs. PARHIER = GLOBAL Results**

<table>
<thead>
<tr>
<th>Element</th>
<th>PARHIER = Local</th>
<th>PARHIER = Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>r2</td>
<td>2.0</td>
<td>1.0</td>
</tr>
<tr>
<td>r3</td>
<td>3.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Parameter Passing Problems**

Changes in scoping rules can cause different simulation results for designs created prior to Star-Hspice Release 95.1 from designs created after that release. Use the following checklist to determine if you will see simulation differences with the new default scoping rules. These checks are especially important if your netlists contain devices from multiple vendors’ libraries.

- Check your subcircuits for parameter defaults on the .SUBCKT or .MACRO line.
- Check your subcircuits for a .PARAM statement within a .SUBCKT definition.
- Check your circuits for global parameter definitions using the .PARAM statement.
- If any of the names from the first three checks are identical, then set up two Star-Hspice jobs, one with .OPTION PARHIER = GLOBAL and one with .OPTION PARHIER = LOCAL, and look for differences in your output.
Chapter 8

Specifying Simulation Output

Use output format statements and variables to display steady state, frequency, and time domain simulation results. These variables also permit you to use behavioral circuit analysis, modeling, and simulation techniques. Display electrical specifications such as rise time, slew rate, amplifier gain, and current density using the output format features.

This chapter discusses the following topics:

- Using Output Statements
- Displaying Simulation Results
- Selecting Simulation Output Parameters
- Specifying User-Defined Analysis (.MEASURE)
- Element Template Listings
Using Output Statements

Output Commands

Star-Hspice output statements are contained in the input netlist file and include .PRINT, .PLOT, .GRAPH, .PROBE, and .MEASURE. Each statement specifies the output variables and type of simulation result to be displayed—for example, .DC, .AC, or .TRAN. With the use of .OPTION POST, all output variables referenced in .PRINT, .PLOT, .GRAPH, .PROBE, and .MEASURE statements are put into the interface files for AvanWaves. AvanWaves allows high resolution, post simulation, and interactive display of waveforms.

<table>
<thead>
<tr>
<th>Output Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.PRINT</td>
<td>Prints numeric analysis results in the output listing file (and post-processor data if .OPTION POST is used).</td>
</tr>
<tr>
<td>.PLOT</td>
<td>Generates low-resolution (ASCII) plots in the output listing file (and post-processor data if .OPTION POST is used).</td>
</tr>
<tr>
<td>.GRAPH</td>
<td>Generates high-resolution plots for specific printing devices (HP LaserJet for example) or in PostScript format, intended for hard-copy outputs without a using a post-processor.</td>
</tr>
<tr>
<td>.PROBE</td>
<td>Outputs data to post-processor output files but not to the output listing (used with .OPTION PROBE to limit output).</td>
</tr>
<tr>
<td>.MEASURE</td>
<td>Prints to output listing file the results of specific user-defined analyses (and post-processor data if .OPTION POST is used).</td>
</tr>
</tbody>
</table>
.Measure Performance

Star-Hspice will take a long time to run if too many .measure statements are specified. In release 2001.2 and above, this issue is resolved for cases created according to the following rule.

Similar variables, which are listed in the .measure statement, must be placed together to save run time.

Examples

Example 1 - Original Case (Slower)

.example tran val1 AVG v(1) FROM=0ms TO=50ms
.example tran val2 AVG v(2) FROM=0ms TO=50ms
.example tran val3 AVG v(1) FROM=50ms TO=100ms
.example tran val4 AVG v(2) FROM=50ms TO=100ms

Example 2 - Improved Case (Faster)

.example tran val1 AVG v(1) FROM=0ms TO=50ms
.example tran val3 AVG v(1) FROM=50ms TO=100ms
.example tran val2 AVG v(2) FROM=0ms TO=50ms
.example tran val4 AVG v(2) FROM=50ms TO=100ms

Output Variables

The output format statements require special output variables to print or plot analysis results for nodal voltages and branch currents. There are five groups of output variables: DC and transient analysis, AC analysis, element template, .MEASURE statement, and parametric analysis.

**DC and transient analysis** displays individual nodal voltages, branch currents, and element power dissipation.

**AC analysis** displays imaginary and real components of a nodal voltage or branch current, as well as the phase of a nodal voltage or branch current. AC analysis results also print impedance parameters and input and output noise.
Element template analysis displays element-specific nodal voltages, branch currents, element parameters, and the derivatives of the element’s node voltage, current, or charge.

The .MEASURE statement variables are user-defined. They represent the electrical specifications measured in a .MEASURE statement analysis.

Parametric analysis variables are mathematically defined expressions operating on user-specified nodal voltages, branch currents, element template variables, or other parameters. You can perform behavioral analysis of simulation results using these variables. See “Using Algebraic Expressions” on page 7-8 for information about parameters in Star-Hspice.
Displaying Simulation Results

The following section describes the statements used to display simulation results for your specific requirements.

.PRINT Statement

The .PRINT statement specifies output variables for which values are printed.

The maximum number of variables in a single .PRINT statement is 32. You can use additional .PRINT statements for more output variables.

To simplify parsing of the output listings, a single “x” printed in the first column indicates the beginning of the .PRINT output data, and a single “y” in the first column indicates the end of the .PRINT output data.

Syntax

```
.PRINT antype ov1 <ov2 ... ov32>
```

where:

- **antype**
  Specifies the type of analysis for outputs. Antype is one of the following types: DC, AC, TRAN, NOISE, or DISTO.

- **ov1 ...**
  Specifies output variables to be printed. These are voltage, current, or element template variables from a DC, AC, TRAN, NOISE, or DISTO analysis.

Statement Order

Star-Hspice produces different .sw0 and .tr0 files based on the order of the .print and .dc statements. If no analysis type is given for a .print command, then the analysis type will match the last analysis command found in the netlist before the .print. See examples below.
CASE 1
.print v(din) i(mxn18)
.dc v din 0 5.0 0.05
.tran 1ns 60ns

CASE 2
.dc v din 0 5.0 0.05
.tran 1ns 60ns
.print v(din) i(mxn18)

CASE 3
.dc v din 0 5.0 0.05
.print v(din) i(mxn18)
.tran 1ns 60ns

- If you are replacing the .print statement with
  .print TRAN v(din) i(mnx)
  then all 3 cases have identical .sw0 and .tr0 files.
- If you are replacing the .print statement with
  .print DC v(din) i(mnx)
  then the .sw0 and .tr0 files will be different.

Example
.PROPRINT TRAN V(4) I(VIN) PAR('V(OUT)/V(IN)')

This example prints out the results of a transient analysis for the nodal voltage named 4 and the current through the voltage source named VIN. The ratio of the nodal voltage at node “OUT” and node “IN” is also printed.

.PROPRINT AC VM(4,2) VR(7) VP(8,3) II(R1)

VM(4,2) specifies that the AC magnitude of the voltage difference (or the difference of the voltage magnitudes, depending on the value of the ACOUT option) between nodes 4 and 2 is printed. VR(7) specifies that the real part of the AC voltage between nodes 7 and ground is printed. VP(8,3) specifies that the phase of the voltage difference between nodes 8 and 3 (or the difference of the
phase of voltage at node 8 and voltage at node 3 depending on the value of ACOUT options) is printed. II(R1) specifies that the imaginary part of the current through R1 is printed.

```
.PRINT AC ZIN YOUT(P) S11(DB) S12(M) Z11(R)
```

The above example specifies that the magnitude of the input impedance, the phase of the output admittance, and several S and Z parameters are to be printed. This statement would accompany a network analysis using the .AC and .NET analysis statements.

```
.PRINT DC V(2) I(VSRC) V(23,17) I1(R1) I1(M1)
```

This example specifies that the DC analysis results are to be printed for several different nodal voltages and currents through the resistor named R1, the voltage source named VSRC, and the drain-to-source current of the MOSFET named M1.

```
.PRINT NOISE INOISE
```

In this example, the equivalent input noise is printed.

```
.PRINT DISTO HD3 SIM2(DB)
```

This example prints the magnitude of the third-order harmonic distortion and the decibel value of the intermodulation distortion sum through the load resistor specified in the .DISTO statement.

```
.PRINT AC INOISE ONOISE VM(OUT) HD3
```

In this statement, specifications of NOISE, DISTO, and AC output variables are included on the same .PRINT statement.

```
.PRINT pj1 = par('p(rd) +p(rs)')
```

This statement prints the value of pj1 with the specified function.

---

**Note:** Star-Hspice ignores .PRINT statement references to nonexistent netlist part names and prints those names in a warning message.
.PLOT Statement

The .PLOT statement plots output values of one or more variables in a selected analysis. Each .PLOT statement defines the contents of one plot, which can have 1 to 32 output variables.

When no plot limits are specified, Star-Hspice automatically determines the minimum and maximum values of each output variable being plotted and scales each plot to fit common limits. To cause Star-Hspice to set limits for certain variables, set the plot limits to (0,0) for those variables.

To make Star-Hspice find plot limits for each plot individually, select .OPTION PLIM to create a different axis for each plot variable. The PLIM option is similar to the plot limit algorithm in SPICE2G.6. In the latter case, each plot can have limits different from any other plot. The overlap of two or more traces on a plot is indicated by a number from 2 through 9.

When more than one output variable appears on the same plot, the first variable specified is printed as well as plotted. If a printout of more than one variable is desired, include another .PLOT statement.

The number of .PLOT statements you can specify for each type of analysis is unlimited. Plot width is set by the option CO (columns out). For a CO setting of 80, a 50-column plot is produced. If CO is 132, a 100-column plot is produced.

Syntax

.PLOT antype ov1 <(plo1,phi1)> … <ov32> + <(plo32,phi32)>

where:

antype Type of analysis for the specified plots. Analysis types are: DC, AC, TRAN, NOISE, or DISTO.
Specifying Simulation Output

Example

In the following example, PAR invokes the plot of the ratio of the collector current and the base current of the transistor Q1.

\[
\text{.PLOT DC V(4) V(5) V(1) PAR('I1(Q1)/I2(Q1)')}
\]
\[
\text{.PLOT TRAN V(17,5) (2,5) I(VIN) V(17) (1,9)}
\]
\[
\text{.PLOT AC VM(5) VM(31,24) VDB(5) VP(5) INOISE}
\]

The second of the two examples above uses the VDB output variable to plot the AC analysis results of the node named 5 in decibels. Also, NOISE results may be requested along with the other variables in the AC plot.

\[
\text{.PLOT AC ZIN YOUT(P) S11(DB) S12(M) Z11(R)}
\]
\[
\text{.PLOT DISTO HD2 HD3(R) SIM2}
\]
\[
\text{.PLOT TRAN V(5,3) V(4) (0,5) V(7) (0,10)}
\]
\[
\text{.PLOT DC V(1) V(2) (0,0) V(3) V(4) (0,5)}
\]

In the last example above, Star-Hspice sets the plot limits for V(1) and V(2), while 0 and 5 volts are specified as the plot limits for V(3) and V(4).

.PROBE Statement

The .PROBE statement saves output variables into the interface and graph data files. Star-Hspice usually saves all voltages and supply currents in addition to the output variables. Set .OPTION PROBE to save output variables only. Use the
.PROBE statement to specify which quantities are to be printed in the output listing.

If you are only interested in the output data file and do not want tabular or plot data in your listing file, set .OPTION PROBE and use the .PROBE statement to specify which values you want saved in the output listing.

**Syntax**

```
.PROBE antype ov1 ... <ov32>
```

where:

- **antype** Type of analysis for the specified plots. Analysis types are: DC, AC, TRAN, NOISE, or DISTO.
- **ov1 ...** Output variables to be plotted. These are voltage, current, or element template variables from a DC, AC, TRAN, NOISE, or DISTO analysis. The limit for the number of output variables in a single .PROBE statement is 32. Additional .PROBE statements may be used to deal with more output variables.

**Example**

```
.PROBE DC V(4) V(5) V(1) beta = PAR(`I1(Q1)/I2(Q1)')
```

**.GRAPH Statement**

The .GRAPH statement allows high resolution plotting of simulation results. This statement is similar to the .PLOT statement with the addition of an optional model. When a model is specified, you can add or change graphing properties for the graph. The .GRAPH statement generates a .gr# graph data file and sends this file directly to the default high resolution graphical device (specified by PRTDEFAULT in the meta.cfg configuration file).

Each .GRAPH statement creates a new .gr# file, where # ranges first from 0 to 9, and then from a to z. The maximum number of graph files that can exist is 36. If more than 36 .GRAPH statements are used, the graph files are overwritten.
starting with the .gr0 file. To overcome this limitation, the option ALT999 or ALT9999 should be used to extend the number of digits allowed in the file name extension to .gr### or .gr#### (in this case # ranges from 0 to 9).

*Note:* The .GRAPH statement is not provided in the PC version of Star-Hspice.

**Syntax**

```
.GRAPH antype <MODEL = mname> <unam1 = > ov1, + <unam2 = > ov2, ... <unam32 = > ov32 (plo, phi)
```

where:

- `antype` Type of analysis for the specified plots. Analysis types are: DC, AC, TRAN, NOISE, or DISTO.

- `mname` Plot model name referenced by the .GRAPH statement. .GRAPH and its plot name allow high resolution plots to be made from Star-Hspice directly.

- `unam1...` User-defined output names, which correspond to output variables ov1…ov32 (unam1 to unam32 respectively), are used as labels instead of output variables for a high resolution graphic output.

- `ov1 ...ov2` Output variables to be printed, 32 maximum. They can be voltage, current, or element template variables from a different type analysis. Algebraic expressions also are used as output variables, but they must be defined inside the PAR( ) statement.

- `plo, phi` Lower and upper plot limits. Set the plot limits only at the end of the .GRAPH statement.

**.MODEL Statement for .GRAPH**

This section describes the model statement for .GRAPH.
**Syntax**

```
.MODEL mname PLOT (pnam1 = val1 pnam2 = val2....)
```

- **mname**: Plot model name referenced by the .GRAPH statements
- **PLOT**: Keyword for a .GRAPH statement model
- **pnam1 = val1...**: Each .GRAPH statement model includes a variety of model parameters. If no model parameters are specified, Star-Hspice takes the default values of the model parameters described in the following table. Pnamn is one of the model parameters of a .GRAPH statement, and valn is the value of pnamn. Valn can be a number of parameter.

**Example**

```
.GRAPH DC cbg = lx18(m1) cgd = lx19(m1) cgs = lx20(m1)
.GRAPH DC MODEL = plotbjt
  + model_ib = i2(q1) meas_ib = par(ib)
  + model_ic = i1(q1) meas_ic = par(ic)
  + model_beta = par('i1(q1)/i2(q1)')
  + meas_beta = par('par(ic)/par(ib)')(1e-10,1e-1)
.MODEL plotbjt PLOT MONO = 1 YSCAL = 2 XSCAL = 2 XMIN = 1e-8
  + XMAX = 1e-1
```
## Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQ</td>
<td>0.0</td>
<td>Plots symbol frequency. Value 0 suppresses plot symbol generation; a value of n generates a plot symbol every n points.</td>
</tr>
<tr>
<td>MONO</td>
<td>0.0</td>
<td>Monotonic option. MONO = 1 automatically resets x-axis if any change in x direction.</td>
</tr>
<tr>
<td>TIC</td>
<td>0.0</td>
<td>Shows tick marks</td>
</tr>
<tr>
<td>XGRID, YGRID</td>
<td>0.0</td>
<td>Setting to 1.0 turns on the axis grid lines</td>
</tr>
<tr>
<td>XMIN, XMAX</td>
<td>0.0</td>
<td>If XMIN is not equal to XMAX, then XMIN and XMAX determines the x-axis plot limits. If XMIN equals XMAX, or if XMIN and XMAX are not set, then the limits are automatically set. These limits apply to the actual x-axis variable value regardless of the XSCAL type.</td>
</tr>
<tr>
<td>XSCAL</td>
<td>1.0</td>
<td>Scale for the x-axis. Two common axis scales are: Linear(LIN) (XSCAL = 1) Logarithm(LOG) (XSCAL = 2)</td>
</tr>
<tr>
<td>YMIN, YMAX</td>
<td>0.0</td>
<td>If YMIN is not equal to YMAX, then YMIN and YMAX determines the y-axis plot limits. The y-axis limits specified in the .GRAPH statement override YMIN and YMAX in the model. If limits are not specified then they are automatically set. These limits apply to the actual y-axis variable value regardless of the YSCAL type.</td>
</tr>
<tr>
<td>YSCAL</td>
<td>1.0</td>
<td>Scale for the y-axis. Two common axis scales are: Linear(LIN) (YSCAL = 1) Logarithm(LOG) (YSCAL = 2)</td>
</tr>
</tbody>
</table>
Print Control Options

.OPTION CO for Printout Width

The number of output variables printed on a single line of output is a function of the number of columns, set by the option CO. Typical values are CO = 80 for narrow printouts and CO = 132 for wide printouts. CO = 80 is the default. The maximum number of output variables allowed is 5 per 80-column output and 8 per 132-column output with twelve characters per column. Star-Hspice automatically creates additional print statements and tables for all output variables beyond the number specified by the CO option.

.WIDTH Statement

Syntax

.WIDTH OUT = {80 |132}

where OUT is the output print width

Example

.WIDTH OUT = 132 $ SPICE compatible style
.OPTION CO = 132 $ preferred style

Permissible values for OUT are 80 and 132. OUT can also be set with option CO.

.OPTION ALT999 or ALT9999 for Output File Name Extension

The output files for postprocessor (from .OPTION POST) or .GRAPH statements have unique extensions .xx#, where xx is a 2-character text string to denote the output type (see “Specifying Simulation Output” on page 8-1 for more information), and # is an alphanumeric character that denotes the .ALTER number of the current simulation. This limits the total number of .ALTER statements in a netlist to 36 before the outputs begin overwriting the current files.

The options ALT999 and ALT9999 extend the output file name extension syntax to .xx### and .xx####, respectively, where # now represents a numerical character only. This syntax allows for 1000 and 10,000 .ALTERs in the input netlist while maintaining a unique file name for the output files.
.OPTION INGOLD for Printout Numerical Format

Variable values are printed in engineering notation by default:

\[
\begin{align*}
F &= 1e-15 & M &= 1e-3 \\
P &= 1e-12 & K &= 1e3 \\
N &= 1e-9 & X &= 1e6 \\
U &= 1e-6 & G &= 1e9
\end{align*}
\]

In contrast to the exponential format, the engineering notation provides two to three extra significant digits and aligns columns to facilitate comparison. To obtain output in exponential format, specify \texttt{INGOLD = 1} or \texttt{2} with an .OPTION statement.

- \texttt{INGOLD = 0} (default) Engineering Format
  \[
  1.234K \\
  123M
  \]
- \texttt{INGOLD = 1} G Format (fixed and exponential)
  \[
  1.234e+03 \\
  .123
  \]
- \texttt{INGOLD = 2} E Format (exponential SPICE)
  \[
  1.234e+03 \\
  .123e-1
  \]

.OPTION POST for High Resolution Graphics

Use an .OPTION POST statement to use AvanWaves to display high resolution plots of simulation results on a graphics terminal or a high resolution laser printer. Use the .OPTION POST to provide output without specifying other parameters. POST has defaults that supply most parameters with usable data.

- \texttt{POST = 0,1,BINARY} Output format is binary
- \texttt{POST = 2,ASCII} Output format is ascii
.OPTION ACCT Summary of Job Statistics

A detailed accounting report is generated using the ACCT option, where:

.OPTION ACCT  Enables reporting
.OPTION ACCT = 1  Is the same as ACCT with no argument (default)
.OPTION ACCT = 2  Enables reporting plus matrix statistic reporting

Example

The following output appears at the end of the output listing.

******  job statistics summary  tnom =  25.000  temp =  25.000
# nodes = 15  # elements =  29  # real*8 mem avail/used = 333333/  13454
# diodes = 0  # bjts = 0  # jfets = 0  # mosfets = 24

           analysis          time    # points  tot. iter  conv.iter
           op point 0.24   1       11
           transient  5.45  161     265     103
           rev =     1
           pass1  0.08
           readin  0.12
           errchk  0.05
           setup   0.04
           output  0.00
the following time statistics are already included in the analysis time
           load   5.22
           solver  0.16
# external nodes =  15  # internal nodes =  0
# branch currents =  5  total matrix size =  20
pivot based and non pivotting solution times
       non pivotting:  ---- decompose 0.08  solve 0.08
       matrix size(  109) = initial size(  105) + fill(   4)
       words copied =  111124
       total cpu time       6.02 seconds
job started at 11:54:11  21-sep92
job ended   at 11:54:36  21-sep92
The definitions for the items in the previous listing follow:

- **# BJTS**: Number of bipolar transistors in the circuit
- **# ELEMENTS**: Total number of elements
- **# JFETS**: Number of JFETs in the circuit
- **# MOSFETS**: Number of MOSFETs in the circuit
- **# NODES**: Total number of nodes
- **# POINTS**: Number of transient points specified by the user on the .TRAN statement. JTRFLG is usually at least 50 unless the option DELMAX is set.

- **CONV.ITER**: Number of points that the simulator needed to take to preserve the accuracy specified by the tolerances
- **DC**: DC operating point analysis time and number of iterations required. The option ITL1 sets the maximum number of iterations.
- **ERRCHK**: Part of the input processing
- **MEM +**: Amount of workspace available and used for the simulation
- **AVAILUSED**: Measured in 64-bit (8-byte) words
- **OUTPUT**: Time required to process all prints and plots
- **LOAD**: Constructs the matrix equation
- **SOLVER**: Solves equations
- **PASS1**: Part of the input processing
- **READIN**: Specifies the input reader that takes the user data file and any additional library files, and generates an internal representation of the information
- **REV**: Number of times the simulator had to cut time (reversals). This is a measure of difficulty.
The ratio of TOT.ITER to CONV.ITER is the best measure of simulator efficiency. The theoretical ratio is 2:1. In this example the ratio was 2.57:1. SPICE generally has a ratio of 3:1 to 7:1.

In transient, the ratio of CONV.ITER to # POINTS is the measure of the number of points evaluated to the number of points printed. If this ratio is greater than about 4, the convergence and time step control tolerances might be too tight for the simulation.

Changing the File Descriptor Limit

A simulation that has a large number of .ALTER statements might fail due to the limit on the number of file descriptors. For example, for a Sun workstation, the default number of file descriptors is 64, and a design with more than 50 .ALTER statements is liable to fail with the following error message:

```
error could not open output spool file /tmp/tmp.nnn
a critical system resource is inaccessible or exhausted
```

To prevent this on a Sun workstation, enter the following operating system command before you start the simulation:

```
limit descriptors 128
```

For platforms other than Sun workstations, see your system administrator for help with increasing the number of files you can open concurrently.
Subcircuit Output Printing

The following examples demonstrate how to print or plot voltages of nodes in subcircuit definitions using .PRINT, .PLOT, .PROBE or .GRAPH.

Note: .PROBE, .PLOT, or .GRAPH may be substituted for .PRINT in the following example.

Example 1

.SUBCKT nor2 A B Y
    .PRINT v(B) v(N1) $ Print statement 1
    M1 N1 A vdd vdd pch w = 6u l = 0.8u
    M2 Y B N1 vdd pch w = 6u l = 0.8u
    M3 Y A vss vss nch w = 3u l = 0.8u
    M4 Y B vss vss nch w = 3u l = 0.8u
.ENDS

Print statement 1 invokes a printout of the voltage on input node B and internal node N1 for every instance of the nor2 subcircuit.

 Example 1

.SUBCKT nor2 A B Y
    .PRINT v(1) v(X1.A) $ Print statement 2
    M1 N1 A vdd vdd pch w = 6u l = 0.8u
    M2 Y B N1 vdd pch w = 6u l = 0.8u
    M3 Y A vss vss nch w = 3u l = 0.8u
    M4 Y B vss vss nch w = 3u l = 0.8u
.ENDS

The print statement above specifies two ways of printing the voltage on input A of instance X1.

 Example 1

.SUBCKT nor2 A B Y
    .PRINT v(3) v(X1.Y) v(X2.A) $ Print statement 3
    M1 N1 A vdd vdd pch w = 6u l = 0.8u
    M2 Y B N1 vdd pch w = 6u l = 0.8u
    M3 Y A vss vss nch w = 3u l = 0.8u
    M4 Y B vss vss nch w = 3u l = 0.8u
.ENDS

This print statement specifies three different ways of printing the voltage at output Y of instance X1. (input A of instance X2).

 Example 1

.SUBCKT nor2 A B Y
    .PRINT v(X2.N1) $ Print statement 4
    M1 N1 A vdd vdd pch w = 6u l = 0.8u
    M2 Y B N1 vdd pch w = 6u l = 0.8u
    M3 Y A vss vss nch w = 3u l = 0.8u
    M4 Y B vss vss nch w = 3u l = 0.8u
.ENDS

The print statement above prints out the voltage on the internal node N1 of instance X2.

 Example 1

.SUBCKT nor2 A B Y
    .PRINT i(X1.M1) $ Print statement 5
    M1 N1 A vdd vdd pch w = 6u l = 0.8u
    M2 Y B N1 vdd pch w = 6u l = 0.8u
    M3 Y A vss vss nch w = 3u l = 0.8u
    M4 Y B vss vss nch w = 3u l = 0.8u
.ENDS
The print statement above prints out the drain-to-source current through MOSFET M1 in instance X1.

**Example 2**

```plaintext
X1 5 6 YYY
  .SUBCKT YYY 15 16
  X2 16 36 ZZZ
  R1 15 25 1
  R2 25 16 1
  .ENDS
  .SUBCKT ZZZ 16 36
  C1 16 0 10P
  R3 36 56 10K
  C2 56 0 1P
  .ENDS
  .PRINT V(X1.25) V(X1.X2.56) V(6)
```

The .PRINT statement voltages are:

- \( V(X1.25) \) represents the voltage at node 25 in instance X1 of subcircuit YYY.
- \( V(X1.X2.56) \) represents the voltage at node 56 in subcircuit ZZZ, called by subcircuit X2, which was called by X1.
- \( V(6) \) represents the voltage of node 16 in instance X1 of subcircuit YYY.

This example prints analysis results for the voltage at node 56 within the subcircuits X2 and X1. The full path name X1.X2.56 specifies that node 56 is within subcircuit X2 that is within subcircuit X1.
Selecting Simulation Output Parameters

This section discusses how to define specific parameters so that the simulation provides the appropriate output. Define simulation parameters using the .OPTION and .MEASURE statements and specific variable element definitions.

DC and Transient Output Variables

Some types of output variables for DC and transient analyses are:

- Voltage differences between specified nodes (or one specified node and ground)
- Current output for an independent voltage source
- Current output for any element
- Element templates containing the values of user-input variables, state variables, element charges, capacitance currents, capacitances, and derivatives for the various types of devices

The codes that you can use to specify the element templates for output are summarized in “Print Control Options” on page 8-14.

Nodal Voltage

Syntax

\[ V \ (n1<,n2>) \]

\( n1, n2 \) Defines the nodes between which the voltage difference \((n1-n2)\) is to be printed or plotted. When \(n2\) is omitted, the voltage difference between \(n1\) and ground (node 0) is given.

Current: Voltage Sources

Syntax

\[ I \ (Vxxx) \]
where:

- \( V_{xxx} \): Voltage source element name. If an independent power supply is within a subcircuit, its current output is accessed by appending a dot and the subcircuit name to the element name, for example, \( I(X1.V_{xxx}) \).

**Example**

```plaintext
.PLOT TRAN I(VIN)
.PRINT DC I(X1.VSRC)
.PLOT DC I(XSUB.XSUBSUB.VY)
```

### Current: Element Branches

**Syntax**

\[ \text{In } (Wwww) \]

where:

- \( n \): Node position number in the element statement. For example, if the element contains four nodes, \( I3 \) denotes the branch current output for the third node; if \( n \) is not specified, the first node is assumed.

- \( Wwww \): Element name. If the element is within a subcircuit, its current output is accessed by appending a dot and the subcircuit name to the element name, for example, \( I3(X1.Wwww) \).

**Example**

- \( I1(R1) \)

  This example specifies the current through the first node of resistor \( R1 \).

- \( I4(X1.M1) \)
The above example specifies the current through the fourth node (the substrate node) of the MOSFET M1, which is defined in subcircuit X1.

I2 (Q1)

The last example specifies the current through the second node (the base node) of the bipolar transistor Q1.

Define each branch circuit by a single element statement. Star-Hspice evaluates branch currents by inserting a zero-volt power supply in series with branch elements.

If Star-Hspice cannot interpret a .PRINT or .PLOT statement containing a branch current, a warning is generated.

Branch current direction for the elements in Figures 8-1 through 8-6 is defined in terms of arrow notation (current direction) and node position number (terminal type).

---

**Figure 8-1: Resistor (node1, node2)**

\[
\begin{align*}
I1 & (R1) \\
I2 & (R1)
\end{align*}
\]

---

**Figure 8-2: Capacitor (node1, node2); Inductor (node1, node2)**

\[
\begin{align*}
I1(L1) \\
I2(L1)
\end{align*}
\]
Figure 8-3: Diode (node1, node2)

- Node1 (anode, P-type, + node)
- Node2 (cathode, N-type, -node)

Figure 8-4: JFET (node1, node2, node3) - n-channel

- Node2 (gate node)
- Node1 (drain node)
- Node3 (source node)

Figure 8-5: BJT (node1, node2, node3, node4) - npn

- Node1 (drain node)
- Node2 (gate node)
- Node3 (source node)
- Node4 (substrate node)
For power calculations, Star-Hspice computes dissipated or stored power in each passive element (R, L, C), and source (V, I, G, E, F, and H) by multiplying the voltage across an element and its corresponding branch current. However, for semiconductor devices, Star-Hspice calculates only the dissipated power. The power stored in the device junction or parasitic capacitances is excluded from the device power computation. Equations for calculating the power dissipated in different types of devices are shown in the following sections.

Star-Hspice also computes the total power dissipated in the circuit, which is the sum of the power dissipated in the devices, resistors, independent current sources, and all the dependent sources. For hierarchical designs, Star-Hspice computes the power dissipation for each subcircuit as well.

**Note:** For the total power (dissipated power + stored power), it is possible to add up the power of each independent source (voltage and current sources).

**Print or Plot Power**

Output the instantaneous element power and the total power dissipation using a .PRINT or .PLOT statement.

**Syntax**

```plaintext
.PRINT <DC | TRAN> P(element_or_subcircuit_name)POWER
```
Power calculation is associated only with transient and DC sweep analyses. The 
.MEASURE statement can be used to compute the average, rms, minimum, 
maximum, and peak-to-peak value of the power. The POWER keyword invokes 
the total power dissipation output.

**Example**

```
.PRINT TRAN P(M1) P(VIN) P(CLOAD) POWER
.PRINT TRAN P(Q1) P(DIO) P(J10) POWER
.PRINT TRAN POWER $ Total transient analysis power
  * dissipation
.PLOT DC POWER P(IIN) P(RLOAD) P(R1)
.PLOT DC POWER P(V1) P(RLOAD) P(VS)
.PRINT TRAN P(Xf1) P(Xf1.Xh1)
```

**Diode Power Dissipation**

\[ P_d = V_{p'n} \cdot (I_{do} + I_{cap}) + V_{p'n} \cdot I_{do} \]

- \( P_d \): Power dissipated in diode
- \( I_{do} \): DC component of the diode current
- \( I_{cap} \): Capacitive component of the diode current
- \( V_{p'n} \): Voltage across the junction
- \( V_{pp'} \): Voltage across the series resistance RS

**BJT Power Dissipation**

**Vertical**

\[ P_d = V_{ce'} \cdot I_{co} + V_{be'} \cdot I_{bo} + V_{cc'} \cdot I_{tot} + V_{ee'} \cdot I_{etot} + V_{sc'} \cdot I_{so} - V_{cc'} \cdot I_{stot} \]
**Lateral**

\[ P_d = V_{c'e'} \cdot I_{co} + V_{b'e'} \cdot I_{bo} + V_{cc'} \cdot I_{ctot} + V_{bb'} \cdot I_{btot} + V_{ee'} \cdot I_{etot} + V_{sb'} \cdot I_{sto} - V_{bb'} \cdot I_{stot} \]

- **Ibo:** DC component of the base current
- **Ico:** DC component of the collector current
- **Iso:** DC component of the substrate current
- **Pd:** Power dissipated in BJT
- **Ibtot:** Total base current (excluding the substrate current)
- **Ictot:** Total collector current (excluding the substrate current)
- **Ietot:** Total emitter current
- **Istot:** Total substrate current
- **Vb'e':** Voltage across the base-emitter junction
- **Vbb':** Voltage across the series base resistance \( R_B \)
- **Vc'e':** Voltage across the collector-emitter terminals
- **Vcc':** Voltage across the series collector resistance \( R_C \)
- **Vee':** Voltage across the series emitter resistance \( R_E \)
- **Vsb':** Voltage across the substrate-base junction
- **Vsc':** Voltage across the substrate-collector junction

**JFET Power Dissipation**

\[ P_d = V_{d's'} \cdot I_{do} + V_{gd'} \cdot I_{gdo} + V_{gs'} \cdot I_{gso} + V_{s's'} \cdot (I_{do} + I_{gso} + I_{cgs}) + V_{dd'} \cdot (I_{do} - I_{gdo} - I_{cgd}) \]
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{cgd}$</td>
<td>Capacitive component of the gate-drain junction current</td>
</tr>
<tr>
<td>$I_{cgs}$</td>
<td>Capacitive component of the gate-source junction current</td>
</tr>
<tr>
<td>$I_{do}$</td>
<td>DC component of the drain current</td>
</tr>
<tr>
<td>$I_{gdo}$</td>
<td>DC component of the gate-drain junction current</td>
</tr>
<tr>
<td>$I_{gso}$</td>
<td>DC component of the gate-source junction current</td>
</tr>
<tr>
<td>$P_d$</td>
<td>Power dissipated in JFET</td>
</tr>
<tr>
<td>$V_d's'$</td>
<td>Voltage across the internal drain-source terminals</td>
</tr>
<tr>
<td>$V_{dd}'$</td>
<td>Voltage across the series drain resistance $RD$</td>
</tr>
<tr>
<td>$V_{gd}'$</td>
<td>Voltage across the gate-drain junction</td>
</tr>
<tr>
<td>$V_{gs}'$</td>
<td>Voltage across the gate-source junction</td>
</tr>
<tr>
<td>$V_{s's}$</td>
<td>Voltage across the series source resistance $RS$</td>
</tr>
</tbody>
</table>

**MOSFET Power Dissipation**

$$P_d = V_d's' \cdot I_{do} + V_{bd}' \cdot I_{bdo} + V_{bs}' \cdot I_{bso} + V_{s's} \cdot (I_{do} + I_{bso} + I_{cbs} + I_{cgs}) + V_{dd}' \cdot (I_{do} - I_{bdo} - I_{cbd} - I_{cgd})$$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{bdo}$</td>
<td>DC component of the bulk-drain junction current</td>
</tr>
<tr>
<td>$I_{bso}$</td>
<td>DC component of the bulk-source junction current</td>
</tr>
<tr>
<td>$I_{cbd}$</td>
<td>Capacitive component of the bulk-drain junction current</td>
</tr>
<tr>
<td>$I_{cbs}$</td>
<td>Capacitive component of the bulk-source junction current</td>
</tr>
<tr>
<td>$I_{cgd}$</td>
<td>Capacitive component of the gate-drain current</td>
</tr>
</tbody>
</table>
AC Analysis Output Variables

Output variables for AC analysis include:

- Voltage differences between specified nodes (or one specified node and ground)
- Current output for an independent voltage source
- Element branch current
- Impedance (Z), admittance (Y), hybrid (H), and scattering (S) parameters
- Input and output impedance and admittance

AC output variable types are listed in Table 8-1. The type symbol is appended to the variable symbol to form the output variable name. For example, VI is the imaginary part of the voltage, or IM is the magnitude of the current.

<table>
<thead>
<tr>
<th>Type Symbol</th>
<th>Variable Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>decibel</td>
</tr>
<tr>
<td>I</td>
<td>imaginary part</td>
</tr>
<tr>
<td>M</td>
<td>magnitude</td>
</tr>
</tbody>
</table>

Icgs  
Ido  
Pd  
Vbd'  
Vbs'  
Vd's'  
Vdd'  
Vs's  

Capacitive component of the gate-source current
DC component of the drain current
Power dissipated in the MOSFET
Voltage across the bulk-drain junction
Voltage across the bulk-source junction
Voltage across the internal drain-source terminals
Voltage across the series drain resistance RD
Voltage across the series source resistance RS
Specify real or imaginary parts, magnitude, phase, decibels, and group delay for voltages and currents.

### Nodal Voltage

**Syntax**

\[ V_x(n_1, <, n_2> \]

where:

- \( x \) Specifies the voltage output type (see Table 8-1)
- \( n_1, n_2 \) Specifies node names. If \( n_2 \) is omitted, ground (node 0) is assumed.

**Example**

\[ \text{.PLOT AC VM(5) VDB(5) VP(5)} \]

The above example plots the magnitude of the AC voltage of node 5 using the output variable VM. The voltage at node 5 is plotted with the VDB output variable. The phase of the nodal voltage at node 5 is plotted with the VP output variable.

Since an AC analysis produces complex results, the values of real or imaginary parts of complex voltages of AC analysis and their magnitude, phase, decibel, and group delay values are calculated using either the SPICE or Star-Hspice method and the control option ACOUT. The default for Star-Hspice is ACOUT = 1. To use the SPICE method, set ACOUT = 0.

### Table 8-1: AC Output Variable Types

<table>
<thead>
<tr>
<th>Type Symbol</th>
<th>Variable Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>phase</td>
</tr>
<tr>
<td>R</td>
<td>real part</td>
</tr>
<tr>
<td>T</td>
<td>group delay</td>
</tr>
</tbody>
</table>

Specify real or imaginary parts, magnitude, phase, decibels, and group delay for voltages and currents.
The SPICE method is typically used to calculate the nodal vector difference in comparing adjacent nodes in a circuit. It is used to find phase or magnitude across a capacitor, inductor, or semiconductor device.

Use the Star-Hspice method to calculate an inter-stage gain in a circuit (such as an amplifier circuit) and to compare its gain, phase, and magnitude.

The following example defines the AC analysis output variables for the Star-Hspice method and then for the SPICE method.

**Example**

**Star-Hspice Method (ACOUT = 1, Default)**

Real and imaginary:

\[
VR(N1,N2) = \text{REAL} [V(N1,0)] - \text{REAL} [V(N2,0)]
\]

\[
VI(N1,N2) = \text{IMAG} [V(N1,0)] - \text{IMAG} [V(N2,0)]
\]

Magnitude:

\[
VM(N1,0) = [VR(N1,0)^2 + VI(N1,0)^2]^{0.5}
\]

\[
VM(N2,0) = [VR(N2,0)^2 + VI(N2,0)^2]^{0.5}
\]

\[
VM(N1,N2) = VM(N1,0) - VM(N2,0)
\]

Phase:

\[
VP(N1,0) = \text{ARCTAN}[VI(N1,0)/VR(N1,0)]
\]

\[
VP(N2,0) = \text{ARCTAN}[VI(N2,0)/VR(N2,0)]
\]

\[
VP(N1,N2) = VP(N1,0) - VP(N2,0)
\]

Decibel:

\[
VDB(N1,N2) = 20 \cdot \text{LOG10}(VM(N1,0)/VM(N2,0))
\]

**SPICE Method (ACOUT = 0)**

Real and imaginary:

\[
VR(N1,N2) = \text{REAL} [V(N1,0) - V(N2,0)]
\]

\[
VI(N1,N2) = \text{IMAG} [V(N1,0) - V(N2,0)]
\]

Magnitude:

\[
VM(N1,N2) = [VR(N1,N2)^2 + VI(N1,N2)^2]^{0.5}
\]
Phase:
\[ VP(N1,N2) = \text{ARCTAN}[VI(N1,N2)/VR(N1,N2)] \]

Decibel:
\[ VDB(N1,N2) = 20 \cdot \text{LOG10}[VM(N1,N2)] \]

**Current: Independent Voltage Sources**

**Syntax**
\[ I_z(Vxxx) \]

where:
- \( z \): Current output type (see Table 8-1)
- \( Vxxx \): Voltage source element name. If an independent power supply is within a subcircuit, its current output is accessed by appending a dot and the subcircuit name to the element name, for example, \( IM(X1.Vxxx) \).

**Example**
\[ .PLOT AC IR(V1) IM(VN2B) IP(X1.X2.VSRC) \]

**Current: Element Branches**

**Syntax**
\[ I_{zn}(www) \]

where:
- \( z \): Current output type (see Table 8-1)
- \( n \): Node position number in the element statement. For example, if the element contains four nodes, \( IM3 \) denotes the magnitude of the branch current output for the third node.
Example

.example

`.PRINT AC IP1(Q5) IM1(Q5) IDB4(X1.M1)`

If the form In(Xxxx) is used for AC analysis output, the magnitude IMn(Xxxx) is the value printed.

**Group Time Delay**

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

\[
TD = \frac{1}{360} \cdot \frac{(\text{phase}_2 - \text{phase}_1)}{(f_2 - f_1)}
\]

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

**Syntax**

.example

`.PRINT AC VT(10) VT(2.25) IT(RL)`

`.PLOT AC IT1(Q1) IT3(M15) IT(D1)`

**Note:** Since there is discontinuity in phase each 360°, the same discontinuity is seen in TD, even though TD is continuous.

**Example**

```
INTEG.SP ACTIVE INTEGRATOR
******  INPUT LISTING
******
V1       1   0  .5   AC   1
 R1       1   2      2K
```
Selecting Simulation Output Parameters

Specifying Simulation Output

C1   2   3      5NF
E3   3   0      2 0  -1000.0
.AC  DEC  15 1K  100K
.PLOT AC VT(3) (0,4U) VP(3)
.END

Network

Syntax

Xij (z), ZIN(z), ZOUT(z), YIN(z), YOUT(z)

where:

X          Specifies Z for impedance, Y for admittance, H for hybrid, or S for scattering parameters

ij         i and j can be 1 or 2. They identify which matrix parameter is printed.

z          Output type (see Table 8-1). If z is omitted, the magnitude of the output variable is printed.

ZIN         Input impedance. For a one port network ZIN, Z11, and H11 are the same

ZOUT        Output impedance

YIN         Input admittance. For a one-port network, YIN and Y11 are the same.

YOUT        Output admittance

Example

.PRINT AC Z11(R) Z12(R) Y21(I) Y22 S11 S11(DB)
.PRINT AC ZIN(R) ZIN(I) YOUT(M) YOUT(P) H11(M)
.PLOT AC S22(M) S22(P) S21(R) H21(P) H12(R)
Noise and Distortion

This section describes the variables used for noise and distortion analysis.

Syntax

\texttt{ovar \langle (z) \rangle}

where:

\texttt{ovar} \quad \text{Noise and distortion analysis parameter. It can be either \texttt{ONOISE} (output noise), or \texttt{INOISE} (equivalent input noise) or any of the distortion analysis parameters (HD2, HD3, SIM2, DIM2, DIM3).}

\texttt{z} \quad \text{Output type (only for distortion). If \texttt{z} is omitted, the magnitude of the output variable is output.}

Example

\texttt{.PRINT DISTO HD2 (M) HD2 (DB)}

Prints the magnitude and decibel values of the second harmonic distortion component through the load resistor specified in the .DISTO statement (not shown).

\texttt{.PLOT NOISE INOISE ONOISE}

\underline{Note:} The noise and distortion output variable may be specified along with other AC output variables in the .PRINT AC or .PLOT AC statements.

Element Template Output

Element templates are used in .PRINT, .PLOT, .PROBE, and .GRAPH statements for output of user-input parameters, state variables, stored charges, capacitor currents, capacitances, and derivatives of variables. The Star-Hspice element templates are listed at the end of this chapter.
**Syntax**

```
Elname:Property
```

<table>
<thead>
<tr>
<th><strong>Elname</strong></th>
<th>Name of the element</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property</strong></td>
<td>Property name of an element, such as a user-input parameter, state variable, stored charge, capacitance current, capacitance, or derivative of a variable</td>
</tr>
</tbody>
</table>

The alias is:

- `LVnn(Elname)`

or

- `LXnn(Elname)`

| **LV** | Form to obtain output of user-input parameters, and state variables |
| **LX** | Form to obtain output of stored charges, capacitor currents, capacitances, and derivatives of variables |
| **nn** | Code number for the desired parameter, given in the tables in this section |

| **Elname** | Name of the element |

**Example**

```
.PLOT TRAN V(1,12) I(X2.VSIN) I2(Q3) DI01:GD
```
Specifying User-Defined Analysis (.MEASURE)

Use the .MEASURE statement to modify information and define the results of successive simulations.

The .MEASURE statement prints user-defined electrical specifications of a circuit and is used extensively in optimization. The specifications include propagation, delay, rise time, fall time, peak-to-peak voltage, minimum and maximum voltage over a specified period, and a number of other user-defined variables. With either the error function or GOAL parameter, .MEASURE is also used extensively for optimization of circuit component values and curve fitting measured data to model parameters.

Measurement results are computed based on postprocessing output. Using the INTERP option to reduce the size of postprocessing output may lead to interpolation error in measurement results. See “Input and Output” on page 9-50 for more information on the INTERP option.

The .MEASURE statement has several different formats, depending on the application. You can use it for either DC sweep, AC, or transient analysis.

Fundamental measurement modes are:
- Rise, fall, and delay
- Find-when
- Equation evaluation
- Average, RMS, min, max, and peak-to-peak
- Integral evaluation
- Derivative evaluation
- Relative error

When a .MEASURE statement fails to execute, Star-Hspice writes 0.0e0 in the .mt# file as the .MEASURE result, and writes “FAILED” in the output listing file. Use the MEASFAIL option to write out results to the .mt#, .ms#, or .ma# files. See “Input and Output” on page 9-50 for additional information on the MEASFAIL option.
The user can also control the output variables which are listed in the .measure statement by using the .putmeas option. See “Input and Output Options” on page 9-50 for additional information.

**Measure Parameter Types**

Measurement parameter results produced by .PARAM statements in .SUBCKT blocks cannot be used outside the subcircuit. That means measurement parameters defined in .SUBCKT statements cannot be passed as bottom-up parameters in hierarchical designs.

Measurement parameter names cannot conflict with standard parameter names. Star-Hspice issues an error message if it encounters a measurement parameter with the same name as a standard parameter definition.

To prevent parameter values given in .MEASURE statements from overwriting parameter assignments in other statements, Star-Hspice keeps track of parameter types. If the same parameter name is used in both a .MEASURE statement and a .PARAM statement at the same hierarchical level, Star-Hspice terminates with an error. No error occurs if the parameter assignments are at different hierarchical levels. PRINT statements that occur at different levels do not print hierarchical information for the parameter name headings.

The following example illustrates how Star-Hspice handles .MEASURE statement parameters.

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

In the above listing, the ‘length’ in the resistor statement

```
rm1 in m1 m2mg w = 'width' l = 'length/6'
```

does not inherit its value from the length in the .MEASURE statement

```
.MEASURE tran length ...
```

since they are of different types. The correct value of l in rm1 should be
\[ l = \frac{\text{length}}{6} = 100\mu \]

instead of a value derived from the measured value in transient analysis.

**.MEASURE Statement: Rise, Fall, and Delay**

This format is used to measure independent-variable (time, frequency, or any parameter or temperature) differential measurements such as rise time, fall time, slew rate, and any measurement that requires the determination of independent variable values. The format specifies substatements TRIG and TARG. These two statements specify the beginning and ending of a voltage or current amplitude measurement.

The rise, fall, and delay measurement mode computes the time, voltage, or frequency between a trigger value and a target value. Examples for transient analysis include rise/fall time, propagation delay, and slew rate measurement. Applications for AC analysis are the measurement of the bandwidth of an amplifier or the frequency at which a certain gain is achieved.

**Syntax**

```
.MEASURE <DC|AC|TRAN> result TRIG … TARG …
+ <GOAL = val> <MINVAL = val> <WEIGHT = val>
```

where:

- **MEASURE** Specifies measurements. You can abbreviate to MEAS.
- **result** Name that is associated with the measured value in the Star-Hspice output. The item measured is the independent variable beginning at the trigger and ending at the target: for transient analysis it is time; for AC analysis it is frequency; for DC analysis it is the DC sweep variable. If the target is reached before the trigger is activated, the resulting value is negative.

**Note:** The terms “DC”, “TRAN”, and “AC” are illegal for result name.
TRIG..., TARG ... Identifies the beginning of trigger and target specifications, respectively.

<DC|AC|TRAN> Specifies the analysis type of the measurement. If omitted, the last analysis mode requested is assumed.

GOAL Specifies the desired measure value in optimization. The error is calculated by

\[ \text{ERRfun} = \frac{(\text{GOAL} - \text{result})}{\text{GOAL}}. \]

MINVAL If the absolute value of GOAL is less than MINVAL, the GOAL value is replaced by MINVAL in the denominator of the ERRfun expression. Default = 1.0e-12.

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

**Trigger**

TRIG trig_var VAL = trig_val <TD = time_delay> <CROSS = c> + <RISE = r> <FALL = f>

or

TRIG AT = val

**Target**

TARG targ_var VAL = targ_val <TD = time_delay> + <CROSS = c | LAST> <RISE = r | LAST> <FALL = f | LAST>

where:

TRIG Indicates the beginning of the trigger specification

trig_val Value of trig_var at which the counter for crossing, rises, or falls is incremented by one

trig_var Specifies the name of the output variable, which determines the logical beginning of measurement. If the target is reached before the trigger is activated, .MEASURE reports a negative value.
specifying simulation output

<table>
<thead>
<tr>
<th>TARG</th>
<th>Indicates the beginning of the target signal specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>targ_val</td>
<td>Specifies the value of the targ_var at which the counter for crossing, rises, or falls is incremented by one</td>
</tr>
<tr>
<td>targ_var</td>
<td>Name of the output variable whose propagation delay is determined with respect to the trig_var</td>
</tr>
<tr>
<td>time_delay</td>
<td>Amount of simulation time that must elapse before the measurement is enabled. The number of crossings, rises, or falls is counted only after time_delay value. The default trigger delay is zero.</td>
</tr>
<tr>
<td>CROSS = c</td>
<td>The numbers indicate which occurrence of a CROSS, FALL, or RISE event causes a measurement to be performed. For RISE = r, the WHEN condition is met and measurement is performed when the designated signal has risen r rise times. For FALL = f, measurement is performed when the designated signal has fallen f fall times. A crossing is either a rise or a fall, so for CROSS = c, measurement is performed when the designated signal has achieved a total of c crossing times, as a result of either rising or falling. For TARG, the last event is specified with the LAST keyword.</td>
</tr>
</tbody>
</table>


8-41
Specifying User-Defined Analysis (.MEASURE) Specifying Simulation Output

Example

.MEASURE TRAN tdlay TRIG V(1) VAL = 2.5 TD = 10n RISE = 2
+ TARG V(2) VAL = 2.5 FALL = 2

This example specifies that a propagation delay measurement is taken between nodes 1 and 2 for a transient analysis. The delay is measured from the second rising edge of the voltage at node 1 to the second falling edge of node 2. The measurement is specified to begin when the second rising voltage at node 1 is 2.5 V and to end when the second falling voltage at node 2 reaches 2.5 V. The TD = 10n parameter does not allow the crossings to be counted until after 10 ns has elapsed. The results are printed as tdlay = <value>.

.MEASURE TRAN riset TRIG I(Q1) VAL = 0.5m RISE = 3
+ TARG I(Q1) VAL = 4.5m RISE = 3

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

The last example uses the short form of TRIG. AT = 10n specifies that the time measurement is to begin at time t = 10 ns in the transient analysis. The TARG parameters specify that the time measurement is to end when V(IN) = 2.5 V on the third crossing. The variable pwidth is the printed output variable.

LAST

Measurement is performed when the last CROSS, FALL, or RISE event occurs. For CROSS = LAST, measurement is performed the last time the WHEN condition is true for either a rising or falling signal. For FALL = LAST, measurement is performed the last time the WHEN condition is true for a falling signal. For RISE = LAST, measurement is performed the last time the WHEN condition is true for a rising signal. LAST is a reserved word and cannot be chosen as a parameter name in the above .MEASURE statements.

AT = val

Special case for trigger specification. The “val” is the time for TRAN analysis, the frequency for AC analysis, or the parameter for DC analysis, at which measurement is to start.
**Note:** If the `.TRAN` statement is used in conjunction with a `.MEASURE` statement, using a nonzero START time in the `.TRAN` statement can result in incorrect `.MEASURE` results. Do not use nonzero START times in `.TRAN` statements when `.MEASURE` is also being used.

**FIND and WHEN Functions**

The FIND and WHEN functions allow any independent variables (time, frequency, parameter), any dependent variables (voltage or current, for example), or the derivative of any dependent variables to be measured when some specific event occurs. These measure statements are useful in unity gain frequency or phase measurements, as well as for measuring the time, frequency, or any parameter value when two signals cross each other, or when a signal crosses a constant value. The measurement starts after a specified time delay, TD. It is possible to find a specific event by setting RISE, FALL, or CROSS to a value (or parameter) or LAST for last event. LAST is a reserved word and cannot be chosen as a parameter name in the above measure statements. See “Displaying Simulation Results” on page 8-5 for the definitions of parameters on measure statement.

**Syntax**

```
.MEASURE <DC|TRAN|AC> result WHEN out_var = val <TD = val>
+ <RISE = r | LAST > <FALL = f | LAST > <CROSS = c | LAST >
+ <GOAL = val> <MINVAL = val> <WEIGHT = val>
```

or

```
.MEASURE <DC|TRAN|AC> result WHEN out_var1 = out_var2 <TD = val>
+ <RISE = r | LAST > <FALL = f | LAST > <CROSS = c | LAST >
+ <GOAL = val> <MINVAL = val> <WEIGHT = val>
```

or

```
.MEASURE <DC|TRAN|AC> result FIND out_var1 WHEN out_var2 = val <TD = val>
+ <RISE = r | LAST > <FALL = f | LAST >
+ <CROSS = c | LAST > <GOAL = val> <MINVAL = val> <WEIGHT = val>
```

or
.MEASURE <DC|TRAN|AC> result FIND out_var1 WHEN out_var2 = out_var3
+ <TD = val > < RISE = r | LAST > < FALL = f | LAST >
+ <CROSS = c | LAST> <GOAL = val> <MINVAL = val> <WEIGHT = val>

or

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

Parameter Definitions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CROSS = c</td>
<td>The numbers indicate which occurrence of a CROSS, FALL, or RISE event causes a measurement to be performed. For RISE = r, the WHEN condition is met and measurement is performed when the designated signal has risen r rise times. For FALL = f, measurement is performed when the designated signal has fallen f fall times. A crossing is either a rise or a fall, so for CROSS = c, measurement is performed when the designated signal has achieved a total of c crossing times, as a result of either rising or falling.</td>
</tr>
<tr>
<td>RISE = r</td>
<td></td>
</tr>
<tr>
<td>FALL = f</td>
<td></td>
</tr>
</tbody>
</table>

<DC|AC|TRAN> Specifies the analysis type of the measurement. If omitted, the last analysis type requested is assumed.

FIND Selects the FIND function

GOAL Specifies the desired .MEASURE value. It is used in optimization. The error is calculated by $\text{ERRfun} = (\text{GOAL} - \text{result})/\text{GOAL}$. 
### Equation Evaluation

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

#### Syntax

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

### Table

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAST</td>
<td>Measurement is performed when the last CROSS, FALL, or RISE event occurs. For CROSS = LAST, measurement is performed the last time the WHEN condition is true for either a rising or falling signal. For FALL = LAST, measurement is performed the last time the WHEN condition is true for a falling signal. For RISE = LAST, measurement is performed the last time the WHEN condition is true for a rising signal. LAST is a reserved word and cannot be chosen as a parameter name in the above .MEASURE statements.</td>
</tr>
<tr>
<td>MINVAL</td>
<td>If the absolute value of GOAL is less than MINVAL, the GOAL value is replaced by MINVAL in the denominator of the ERRfun expression. Default = 1.0e-12.</td>
</tr>
<tr>
<td>out_var(1,2,3)</td>
<td>Variables used to establish conditions at which measurement is to take place</td>
</tr>
<tr>
<td>result</td>
<td>Name associated with the measured value in the Star-Hspice output</td>
</tr>
<tr>
<td>TD</td>
<td>Identifies the time at which measurement is to start</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>Calculated error is multiplied by the weight value. Default = 1.0.</td>
</tr>
<tr>
<td>WHEN</td>
<td>Selects the WHEN function</td>
</tr>
</tbody>
</table>
+ <GOAL = val> <MINVAL = val>

**Average, RMS, MIN, MAX, INTEG, and Peak-To-Peak**

The average (AVG), RMS, MIN, MAX, and peak-to-peak (PP) measurement modes report statistical functions of the output variable rather than the analysis value. Average calculates the area under the output variable divided by the periods of interest. RMS takes the square root of the area under the output variable square divided by the period of interest. MIN reports the minimum value of the output function over the specified interval. MAX reports the maximum value of the output function over the specified interval. PP (peak-to-peak) reports the maximum value minus the minimum value over the specified interval.

**Syntax**

```plaintext
.MEASURE <DC|AC|TRAN> result func out_var <FROM = val> <TO = val> + <GOAL = val> <MINVAL = val> <WEIGHT = val>
```

where:

- `<DC|AC|TRAN>` Specifies the analysis type of the measurement. If omitted, the last analysis mode requested is assumed.
- `FROM` Specifies the initial value for the “func” calculation. For transient analysis, value is in units of time.
- `TO` Specifies the end of the “func” calculation.
- `GOAL` Specifies the desired .MEASURE value. It is used in optimization. The error is calculated by
  \[ \text{ERRfun} = \frac{(\text{GOAL} - \text{result})}{\text{GOAL}} \]
- `MINVAL` If the absolute value of GOAL is less than MINVAL, the GOAL value is replaced by MINVAL in the denominator of the ERRfun expression. Default = 1.0e-12.
func
Indicates the type of the measure statement, one of the following:

- **AVG (average):** Calculates the area under the `out_var` divided by the periods of interest
- **MAX (maximum):** Reports the maximum value of the `out_var` over the specified interval
- **MIN (minimum):** Reports the minimum value of the `out_var` over the specified interval
- **PP (peak-to-peak):** Reports the maximum value minus the minimum value of the `out_var` over the specified interval
- **RMS (root mean squared):** Calculates the square root of the area under the `out_var^2` curve divided by the period of interest

result
Name that is associated with the measured value in the Star-Hspice output. The value is a function of the variable specified (`out_var`) and func.

out_var
Name of any output variable whose function ("func") is to be measured in the simulation.

**WEIGHT**
The calculated error is multiplied by the weight value. Default = 1.0.

**Example**
```
.MEAS TRAN avgval AVG V(10) FROM = 10ns TO = 55ns
```
The example above calculates the average nodal voltage value for node 10 during the transient sweep from the time 10 ns to 55 ns and prints out the result as "avgval".
.MEAS TRAN MAXVAL MAX V(1,2)
FROM = 15ns TO = 100ns

The example above finds the maximum voltage difference between nodes 1 and 2 for the time period from 15 ns to 100 ns.

.MEAS TRAN MINVAL MIN V(1,2)
FROM = 15ns TO = 100ns

.MEAS TRAN P2PVAL PP I(M1)
FROM = 10ns TO = 100ns

**INTEGRAL Function**

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

**Syntax**

```
.MEASURE <DC|AC|TRAN> result INTEGRAL out_var <FROM = val>
+ <TO = val> <GOAL = val> <MINVAL = val> <WEIGHT = val>
```

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

**Example**

The following example calculates the integral of I(cload) from 10 ns to 100 ns.

```plaintext
.MEAS TRAN charge INTEG I(cload)
FROM = 10ns TO = 100ns
```

**DERIVATIVE Function**

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

**Syntax**

```
.MEASURE <DC|AC|TRAN> result DERIVATIVE out_var AT = val <GOAL = val>
+ <MINVAL = val> <WEIGHT = val>
```
or

```
.MEASURE <DC|AC|TRAN> result DERIVATIVE out_var WHEN var2 = val
+ <RISE = r | LAST> <FALL = f | LAST> <CROSS = c | LAST>
+ <TD = tdval> <GOAL = goalval> <MINVAL = minval> <WEIGHT = weightval>
```

or

```
.MEASURE <DC|AC|TRAN> result DERIVATIVE out_var WHEN var2 = var3
+ <RISE = r | LAST> <FALL = f | LAST> <CROSS = c | LAST>
+ <TD = tdval> <GOAL = goalval> <MINVAL = minval> <WEIGHT = weightval>
```

where:

- **AT = val**
  Value of `out_var` at which the derivative is to be found

- **CROSS = c**
  The numbers indicate which occurrence of a CROSS, FALL, or RISE event causes a measurement to be performed. For RISE = r, the WHEN condition is met and measurement is performed when the designated signal has risen `r` rise times. For FALL = f, measurement is performed when the designated signal has fallen `f` fall times. A crossing is either a rise or a fall, so for CROSS = c, measurement is performed when the designated signal has achieved a total of `c` crossing times, as a result of either rising or falling.

- **<DC|AC|TRAN>**
  Specifies the analysis type measured. If omitted, the last analysis mode requested is assumed.

- **DERIVATIVE**
  Selects the derivative function. May be abbreviated to DERIV.

- **GOAL**
  Specifies the desired .MEASURE value. It is used in optimization. The error is calculated by
  \[
  \text{ERRfun} = \frac{(\text{GOAL} - \text{result})}{\text{GOAL}}
  \]
LAST  Measurement is performed when the last CROSS, FALL, or RISE event occurs. For CROSS = LAST, measurement is performed the last time the WHEN condition is true for either a rising or falling signal. For FALL = LAST, measurement is performed that last time the WHEN condition is true for a falling signal. For RISE = LAST, measurement is performed the last time the WHEN condition is true for a rising signal. LAST is a reserved word and cannot be chosen as a parameter name in the above .MEASURE statements.

MINVAL  If the absolute value of GOAL is less than MINVAL, the GOAL value is replaced by MINVAL in the denominator of the ERRfun expression. Default = 1.0e-12.

out_var  Variable for which the derivative is to be found

result  Name associated with the measured value in the Star-Hspice output

TD  Identifies the time at which measurement is to start

var(2,3)  Variables used to establish conditions at which measurement is to take place

WEIGHT  The calculated error between result and GOAL is multiplied by the weight value. Default = 1.0.

WHEN  Selects the WHEN function

Example

The following example calculates the derivative of V(out) at 25 ns:

```
.MEAS TRAN slew rate DERIV V(out) AT = 25ns
```

The following example calculates the derivative of v(1) when v(1) is equal to 0.9*vdd:
The following example calculates the derivative of \( \frac{VP(output)}{360.0} \) when the frequency is 10 kHz:

\[ .MEAS AC \text{ delay} \text{ DERIV 'VP(output)/360.0' AT = 10khz} \]

**ERROR Function**

The relative error function reports the relative difference of two output variables. This format is often used in optimization and curve fitting of measured data. The relative error format specifies the variable to be measured and calculated from the .PARAM variables. The relative error between the two is calculated using the ERR, ERR1, ERR2, or ERR3 function. With this format, you can specify a group of parameters to vary to match the calculated value and the measured data.

**Syntax**

\[ .\text{MEASURE} <\text{DC}|\text{AC}|\text{TRAN}> \text{ result} \text{ ERRfun} \text{ meas_var} \text{ calc_var} \text{ <MINV AL = val> } \]
\[ + \text{ < IGNORE | YMIN = val> YMAX = val> WEIGHT = val> FROM = val> TO = val> } \]

where:

- \(<\text{DC}|\text{AC}|\text{TRAN}>\) Specifies the analysis type of the measurement. If omitted, the last analysis mode requested is assumed.

- \( \text{result} \) Name associated with the measured result in the output

- \( \text{ERRfun} \) ERRfun indicates which error function to use: ERR, ERR1, ERR2, or ERR3.

- \( \text{meas\_var} \) Name of any output variable or parameter in the data statement. M denotes the \( \text{meas\_var} \) in the error equation.

- \( \text{calc\_var} \) Name of the simulated output variable or parameter in the .MEASURE statement to be compared with \( \text{meas\_var} \). C denotes the \( \text{calc\_var} \) in the error equation.
Error Equations

ERR

ERR sums the squares of \((M-C)/\max(M, \text{MINVAL})\) for each point, divides by the number of points, and then takes the square root of the result. \(M\) (meas var) and \(C\) (calc var) are the measured and calculated values of the device or circuit response, respectively. \(NPTS\) is the number of data points.

\[
ERR = \left[ \frac{1}{NPTS} \sum_{i=1}^{NPTS} \frac{M_i - C_i}{\max(M, \text{MINVAL}, M_i)} \right]^{1/2}
\]
**ERR1**

ERR1 computes the relative error at each point. For NPTS points, there are NPTS ERR1 error function calculations. For device characterization, the ERR1 approach has been found to be more efficient than the other error functions (ERR, ERR2, ERR3).

\[
ERR1_i = \frac{M_i - C_i}{\max(MINVAL, M_i)} \quad i = 1, NPTS
\]

Star-Hspice does not print out each calculated ERR1 value. When the ERR1 option is set, it returns an ERR value calculated as follows:

\[
ERR = \left[ \frac{1}{NPTS} \cdot \sum_{i=1}^{NPTS} ERR1_i^2 \right]^{1/2}
\]

**ERR2**

This option computes the absolute relative error at each point. For NPTS points, there are NPTS error function calls.

\[
ERR2_i = \frac{M_i - C_i}{\max(MINVAL, M_i)} \quad i = 1, NPTS
\]

The returned value printed for ERR2 is

\[
ERR = \frac{1}{NPTS} \cdot \sum_{i=1}^{NPTS} ERR2_i
\]

**ERR3**

\[
ERR3_i = \pm \log \left( \frac{|M_i|}{C_i} \right) \quad i = 1, NPTS
\]

The + and - signs correspond to a positive and negative M/C ratio, respectively.
**Note:** If the measured value \( M \) is less than \( \text{MINVAL} \), the \( \text{MINVAL} \) is used instead. Also, if the absolute value of \( M \) is less than the \( \text{IGNOR} \) \( \text{YMIN} \) value or greater than the \( \text{YMAX} \) value, then this point is not considered in the error calculation.
## Element Template Listings

### Resistor

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>LV1</td>
<td>Conductance at analysis temperature</td>
</tr>
<tr>
<td>R</td>
<td>LV2</td>
<td>Resistance at reference temperature</td>
</tr>
<tr>
<td>TC1</td>
<td>LV3</td>
<td>First temperature coefficient</td>
</tr>
<tr>
<td>TC2</td>
<td>LV4</td>
<td>Second temperature coefficient</td>
</tr>
</tbody>
</table>

### Capacitor

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEFF</td>
<td>LV1</td>
<td>Computed effective capacitance</td>
</tr>
<tr>
<td>IC</td>
<td>LV2</td>
<td>Initial condition</td>
</tr>
<tr>
<td>Q</td>
<td>LX0</td>
<td>Charge stored in capacitor</td>
</tr>
<tr>
<td>CURR</td>
<td>LX1</td>
<td>Current flowing through capacitor</td>
</tr>
<tr>
<td>VOLT</td>
<td>LX2</td>
<td>Voltage across capacitor</td>
</tr>
<tr>
<td>–</td>
<td>LX3</td>
<td>Capacitance (not used in Star-Hspice releases after 95.3)</td>
</tr>
</tbody>
</table>

### Inductor

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEFF</td>
<td>LV1</td>
<td>Computed effective inductance</td>
</tr>
<tr>
<td>IC</td>
<td>LV2</td>
<td>Initial condition</td>
</tr>
<tr>
<td>FLUX</td>
<td>LX0</td>
<td>Flux in the inductor</td>
</tr>
</tbody>
</table>
### Inductor

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLT</td>
<td>LX1</td>
<td>Voltage across inductor</td>
</tr>
<tr>
<td>CURR</td>
<td>LX2</td>
<td>Current flowing through inductor</td>
</tr>
<tr>
<td></td>
<td>LX4</td>
<td>Inductance (not used in Star-Hspice releases after 95.3)</td>
</tr>
</tbody>
</table>

### Mutual Inductor

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>LV1</td>
<td>Mutual inductance</td>
</tr>
</tbody>
</table>

### Voltage-Controlled Current Source

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURR</td>
<td>LX0</td>
<td>Current through the source, if VCCS</td>
</tr>
<tr>
<td>R</td>
<td>LX0</td>
<td>Resistance value, if VCR</td>
</tr>
<tr>
<td>C</td>
<td>LX0</td>
<td>Capacitance value, if VCCAP</td>
</tr>
<tr>
<td>CV</td>
<td>LX1</td>
<td>Controlling voltage</td>
</tr>
<tr>
<td>CQ</td>
<td>LX1</td>
<td>Capacitance charge, if VCCAP</td>
</tr>
<tr>
<td>DI</td>
<td>LX2</td>
<td>Derivative of source current with respect to control voltage</td>
</tr>
<tr>
<td>ICAP</td>
<td>LX2</td>
<td>Capacitance current, if VCCAP</td>
</tr>
<tr>
<td>VCAP</td>
<td>LX3</td>
<td>Voltage across capacitance, if VCCAP</td>
</tr>
</tbody>
</table>

### Voltage-Controlled Voltage Source

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLT</td>
<td>LX0</td>
<td>Source voltage</td>
</tr>
</tbody>
</table>
## Voltage-Controlled Voltage Source

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURR</td>
<td>LX0</td>
<td>Current through source</td>
</tr>
<tr>
<td>CV</td>
<td>LX1</td>
<td>Controlling voltage</td>
</tr>
<tr>
<td>DV</td>
<td>LX2</td>
<td>Derivative of source voltage with respect to control current</td>
</tr>
</tbody>
</table>

## Current-Controlled Current Source

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURR</td>
<td>LX0</td>
<td>Current through source</td>
</tr>
<tr>
<td>CI</td>
<td>LX1</td>
<td>Controlling current</td>
</tr>
<tr>
<td>DI</td>
<td>LX2</td>
<td>Derivative of source current with respect to control current</td>
</tr>
</tbody>
</table>

## Current-Controlled Voltage Source

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLT</td>
<td>LX0</td>
<td>Source voltage</td>
</tr>
<tr>
<td>CURR</td>
<td>LX1</td>
<td>Source current</td>
</tr>
<tr>
<td>CI</td>
<td>LX2</td>
<td>Controlling current</td>
</tr>
<tr>
<td>DV</td>
<td>LX3</td>
<td>Derivative of source voltage with respect to control current</td>
</tr>
</tbody>
</table>

## Independent Voltage Source

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLT</td>
<td>LV1</td>
<td>DC/transient voltage</td>
</tr>
<tr>
<td>VOLTM</td>
<td>LV2</td>
<td>AC voltage magnitude</td>
</tr>
</tbody>
</table>
### Independent Voltage Source

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLTP</td>
<td>LV3</td>
<td>AC voltage phase</td>
</tr>
</tbody>
</table>

### Independent Current Source

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURR</td>
<td>LV1</td>
<td>DC/transient current</td>
</tr>
<tr>
<td>CURRM</td>
<td>LV2</td>
<td>AC current magnitude</td>
</tr>
<tr>
<td>CURRP</td>
<td>LV3</td>
<td>AC current phase</td>
</tr>
</tbody>
</table>

### Diode

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA</td>
<td>LV1</td>
<td>Diode area factor</td>
</tr>
<tr>
<td>AREAX</td>
<td>LV23</td>
<td>Area after scaling</td>
</tr>
<tr>
<td>IC</td>
<td>LV2</td>
<td>Initial voltage across diode</td>
</tr>
<tr>
<td>VD</td>
<td>LX0</td>
<td>Voltage across diode (VD), excluding RS (series resistance)</td>
</tr>
<tr>
<td>IDC</td>
<td>LX1</td>
<td>DC current through diode (ID), excluding RS. Total diode current is the sum of IDC and ICAP</td>
</tr>
<tr>
<td>GD</td>
<td>LX2</td>
<td>Equivalent conductance (GD)</td>
</tr>
<tr>
<td>QD</td>
<td>LX3</td>
<td>Charge of diode capacitor (QD)</td>
</tr>
<tr>
<td>ICAP</td>
<td>LX4</td>
<td>Current through diode capacitor. Total diode current is the sum of IDC and ICAP.</td>
</tr>
<tr>
<td>C</td>
<td>LX5</td>
<td>Total diode capacitance</td>
</tr>
<tr>
<td>PID</td>
<td>LX7</td>
<td>Photo current in diode</td>
</tr>
</tbody>
</table>
### BJT

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA</td>
<td>LV1</td>
<td>Area factor</td>
</tr>
<tr>
<td>ICVBE</td>
<td>LV2</td>
<td>Initial condition for base-emitter voltage (VBE)</td>
</tr>
<tr>
<td>ICVCE</td>
<td>LV3</td>
<td>Initial condition for collector-emitter voltage (VCE)</td>
</tr>
<tr>
<td>MULT</td>
<td>LV4</td>
<td>Number of multiple BJTs</td>
</tr>
<tr>
<td>FT</td>
<td>LV5</td>
<td>FT (Unity gain bandwidth)</td>
</tr>
<tr>
<td>ISUB</td>
<td>LV6</td>
<td>Substrate current</td>
</tr>
<tr>
<td>GSUB</td>
<td>LV7</td>
<td>Substrate conductance</td>
</tr>
<tr>
<td>LOGIC</td>
<td>LV8</td>
<td>LOG 10 (IC)</td>
</tr>
<tr>
<td>LOGIB</td>
<td>LV9</td>
<td>LOG 10 (IB)</td>
</tr>
<tr>
<td>BETA</td>
<td>LV10</td>
<td>BETA</td>
</tr>
<tr>
<td>LOGBETAI</td>
<td>LV11</td>
<td>LOG 10 (BETA) current</td>
</tr>
<tr>
<td>ICTOL</td>
<td>LV12</td>
<td>Collector current tolerance</td>
</tr>
<tr>
<td>IBTOL</td>
<td>LV13</td>
<td>Base current tolerance</td>
</tr>
<tr>
<td>RB</td>
<td>LV14</td>
<td>Base resistance</td>
</tr>
<tr>
<td>GRE</td>
<td>LV15</td>
<td>Emitter conductance, 1/RE</td>
</tr>
<tr>
<td>GRC</td>
<td>LV16</td>
<td>Collector conductance, 1/RC</td>
</tr>
<tr>
<td>PIBC</td>
<td>LV18</td>
<td>Photo current, base-collector</td>
</tr>
<tr>
<td>PIBE</td>
<td>LV19</td>
<td>Photo current, base-emitter</td>
</tr>
<tr>
<td>VBE</td>
<td>LX0</td>
<td>VBE</td>
</tr>
<tr>
<td>VBC</td>
<td>LX1</td>
<td>Base-collector voltage (VBC)</td>
</tr>
<tr>
<td>CCO</td>
<td>LX2</td>
<td>Collector current (CCO)</td>
</tr>
<tr>
<td>CBO</td>
<td>LX3</td>
<td>Base current (CBO)</td>
</tr>
<tr>
<td>GPI</td>
<td>LX4</td>
<td>( g_{\pi} = \frac{i_b}{v_{be}} \text{constant } v_{bc} )</td>
</tr>
</tbody>
</table>
### BJT

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GU</td>
<td>LX5</td>
<td>$g_n = \frac{ib}{vbc}$ constant $vbe$</td>
</tr>
<tr>
<td>GM</td>
<td>LX6</td>
<td>$g_m = \frac{ic}{vbe} + \frac{ic}{vbe}$ constant $vce$</td>
</tr>
<tr>
<td>G0</td>
<td>LX7</td>
<td>$g_0 = \frac{ic}{vce}$ constant $vbe$</td>
</tr>
<tr>
<td>QBE</td>
<td>LX8</td>
<td>Base-emitter charge (QBE)</td>
</tr>
<tr>
<td>CQBE</td>
<td>LX9</td>
<td>Base-emitter charge current (CQBE)</td>
</tr>
<tr>
<td>QBC</td>
<td>LX10</td>
<td>Base-collector charge (QBC)</td>
</tr>
<tr>
<td>CQBC</td>
<td>LX11</td>
<td>Base-collector charge current (CQBC)</td>
</tr>
<tr>
<td>QCS</td>
<td>LX12</td>
<td>Current-substrate charge (QCS)</td>
</tr>
<tr>
<td>CQCS</td>
<td>LX13</td>
<td>Current-substrate charge current (CQCS)</td>
</tr>
<tr>
<td>QBX</td>
<td>LX14</td>
<td>Base-internal base charge (QBX)</td>
</tr>
<tr>
<td>CQBX</td>
<td>LX15</td>
<td>Base-internal base charge current (CQBX)</td>
</tr>
<tr>
<td>GXO</td>
<td>LX16</td>
<td>$\frac{1}{R_{beff}}$ Internal conductance (GXO)</td>
</tr>
<tr>
<td>CEXBC</td>
<td>LX17</td>
<td>Base-collector equivalent current (CEXBC)</td>
</tr>
<tr>
<td>–</td>
<td>LX18</td>
<td>Base-collector conductance (GEQCBO) (not used in Star-Hspice releases after 95.3)</td>
</tr>
<tr>
<td>CAP_BE</td>
<td>LX19</td>
<td>$cbe$ capacitance ($C_{\Pi}$)</td>
</tr>
<tr>
<td>CAP_IBC</td>
<td>LX20</td>
<td>$cbc$ internal base-collector capacitance ($C_{\mu}$)</td>
</tr>
<tr>
<td>CAP_SCB</td>
<td>LX21</td>
<td>$csc$ substrate-collector capacitance for vertical transistors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$csb$ substrate-base capacitance for lateral transistors</td>
</tr>
<tr>
<td>CAP_XBC</td>
<td>LX22</td>
<td>$cbbc_x$ external base-collector capacitance</td>
</tr>
<tr>
<td>CMCMO</td>
<td>LX23</td>
<td>$(TF*IBE) / vbc$</td>
</tr>
<tr>
<td>VSUB</td>
<td>LX24</td>
<td>Substrate voltage</td>
</tr>
</tbody>
</table>
### JFET

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA</td>
<td>LV1</td>
<td>JFET area factor</td>
</tr>
<tr>
<td>VDS</td>
<td>LV2</td>
<td>Initial condition for drain-source voltage</td>
</tr>
<tr>
<td>VGS</td>
<td>LV3</td>
<td>Initial condition for gate-source voltage</td>
</tr>
<tr>
<td>PIGD</td>
<td>LV16</td>
<td>Photo current, gate-drain in JFET</td>
</tr>
<tr>
<td>PIGS</td>
<td>LV17</td>
<td>Photo current, gate-source in JFET</td>
</tr>
<tr>
<td>VGS</td>
<td>LX0</td>
<td>VGS</td>
</tr>
<tr>
<td>VGD</td>
<td>LX1</td>
<td>Gate-drain voltage (VGD)</td>
</tr>
<tr>
<td>CGSO</td>
<td>LX2</td>
<td>Gate-to-source (CGSO)</td>
</tr>
<tr>
<td>CDO</td>
<td>LX3</td>
<td>Drain current (CDO)</td>
</tr>
<tr>
<td>CGDO</td>
<td>LX4</td>
<td>Gate-to-drain current (CGDO)</td>
</tr>
<tr>
<td>GMO</td>
<td>LX5</td>
<td>Transconductance (GMO)</td>
</tr>
<tr>
<td>GDSO</td>
<td>LX6</td>
<td>Drain-source transconductance (GDSO)</td>
</tr>
<tr>
<td>GGSO</td>
<td>LX7</td>
<td>Gate-source transconductance (GGSO)</td>
</tr>
<tr>
<td>GGDO</td>
<td>LX8</td>
<td>Gate-drain transconductance (GGDO)</td>
</tr>
<tr>
<td>QGS</td>
<td>LX9</td>
<td>Gate-source charge (QGS)</td>
</tr>
<tr>
<td>CQGS</td>
<td>LX10</td>
<td>Gate-source charge current (CQGS)</td>
</tr>
<tr>
<td>QGD</td>
<td>LX11</td>
<td>Gate-drain charge (QGD)</td>
</tr>
<tr>
<td>CQGD</td>
<td>LX12</td>
<td>Gate-drain charge current (CQGD)</td>
</tr>
<tr>
<td>CAP_GS</td>
<td>LX13</td>
<td>Gate-source capacitance</td>
</tr>
<tr>
<td>CAP_GD</td>
<td>LX14</td>
<td>Gate-drain capacitance</td>
</tr>
<tr>
<td>–</td>
<td>LX15</td>
<td>Body-source voltage (not used in Star-Hspice releases after 95.3)</td>
</tr>
<tr>
<td>QDS</td>
<td>LX16</td>
<td>Drain-source charge (QDS)</td>
</tr>
</tbody>
</table>
### JFET

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQDS</td>
<td>LX17</td>
<td>Drain-source charge current (CQDS)</td>
</tr>
<tr>
<td>GMBS</td>
<td>LX18</td>
<td>Drain-body (backgate) transconductance (GMBS)</td>
</tr>
</tbody>
</table>

### MOSFET

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>LV1</td>
<td>Channel length (L)</td>
</tr>
<tr>
<td>W</td>
<td>LV2</td>
<td>Channel width (W)</td>
</tr>
<tr>
<td>AD</td>
<td>LV3</td>
<td>Area of the drain diode (AD)</td>
</tr>
<tr>
<td>AS</td>
<td>LV4</td>
<td>Area of the source diode (AS)</td>
</tr>
<tr>
<td>ICVDS</td>
<td>LV5</td>
<td>Initial condition for drain-source voltage (VDS)</td>
</tr>
<tr>
<td>ICVGS</td>
<td>LV6</td>
<td>Initial condition for gate-source voltage (VGS)</td>
</tr>
<tr>
<td>ICVBS</td>
<td>LV7</td>
<td>Initial condition for bulk-source voltage (VBS)</td>
</tr>
<tr>
<td>–</td>
<td>LV8</td>
<td>Device polarity: 1 = forward, -1 = reverse (not used in Star-Hspice releases after 95.3)</td>
</tr>
<tr>
<td>VTH</td>
<td>LV9</td>
<td>Threshold voltage (bias dependent)</td>
</tr>
<tr>
<td>VDSAT</td>
<td>LV10</td>
<td>Saturation voltage (VDSAT)</td>
</tr>
<tr>
<td>PD</td>
<td>LV11</td>
<td>Drain diode periphery (PD)</td>
</tr>
<tr>
<td>PS</td>
<td>LV12</td>
<td>Source diode periphery (PS)</td>
</tr>
<tr>
<td>RDS</td>
<td>LV13</td>
<td>Drain resistance (squares) (RDS)</td>
</tr>
<tr>
<td>RSS</td>
<td>LV14</td>
<td>Source resistance (squares) (RSS)</td>
</tr>
<tr>
<td>XQC</td>
<td>LV15</td>
<td>Charge sharing coefficient (XQC)</td>
</tr>
<tr>
<td>GDEFF</td>
<td>LV16</td>
<td>Effective drain conductance (1/RDeff)</td>
</tr>
<tr>
<td>GSEFF</td>
<td>LV17</td>
<td>Effective source conductance (1/RSeff)</td>
</tr>
</tbody>
</table>
### MOSFET

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDBS</td>
<td>LV18</td>
<td>Drain-bulk saturation current at -1 volt bias</td>
</tr>
<tr>
<td>ISBS</td>
<td>LV19</td>
<td>Source-bulk saturation current at -1 volt bias</td>
</tr>
<tr>
<td>VDBEFF</td>
<td>LV20</td>
<td>Effective drain bulk voltage</td>
</tr>
<tr>
<td>BETAFF</td>
<td>LV21</td>
<td>BETA effective</td>
</tr>
<tr>
<td>GAMMAFF</td>
<td>LV22</td>
<td>GAMMA effective</td>
</tr>
<tr>
<td>DELTAL</td>
<td>LV23</td>
<td>$\Delta$ (MOS6 amount of channel length modulation) (only valid for LEVELs 1, 2, 3 and 6)</td>
</tr>
<tr>
<td>UBEFF</td>
<td>LV24</td>
<td>UB effective (only valid for LEVELs 1, 2, 3 and 6)</td>
</tr>
<tr>
<td>VG</td>
<td>LV25</td>
<td>VG drive (only valid for LEVELs 1, 2, 3 and 6)</td>
</tr>
<tr>
<td>VFBFEFF</td>
<td>LV26</td>
<td>VFB effective</td>
</tr>
<tr>
<td>–</td>
<td>LV31</td>
<td>Drain current tolerance (not used in Star-Hspice releases after 95.3)</td>
</tr>
<tr>
<td>IDSTOL</td>
<td>LV32</td>
<td>Source diode current tolerance</td>
</tr>
<tr>
<td>IDDTOL</td>
<td>LV33</td>
<td>Drain diode current tolerance</td>
</tr>
<tr>
<td>COVLGS</td>
<td>LV36</td>
<td>Gate-source overlap capacitance</td>
</tr>
<tr>
<td>COVLGD</td>
<td>LV37</td>
<td>Gate-drain overlap capacitance</td>
</tr>
<tr>
<td>COVLGB</td>
<td>LV38</td>
<td>Gate-bulk overlap capacitance</td>
</tr>
<tr>
<td>VBS</td>
<td>LX1</td>
<td>Bulk-source voltage (VBS)</td>
</tr>
<tr>
<td>VGS</td>
<td>LX2</td>
<td>Gate-source voltage (VGS)</td>
</tr>
<tr>
<td>VDS</td>
<td>LX3</td>
<td>Drain-source voltage (VDS)</td>
</tr>
<tr>
<td>CDO</td>
<td>LX4</td>
<td>DC drain current (CDO)</td>
</tr>
<tr>
<td>CBSO</td>
<td>LX5</td>
<td>DC source-bulk diode current (CBSO)</td>
</tr>
<tr>
<td>CBDO</td>
<td>LX6</td>
<td>DC drain-bulk diode current (CBDO)</td>
</tr>
</tbody>
</table>
**MOSFET**

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMO</td>
<td>LX7</td>
<td>DC gate transconductance (GMO)</td>
</tr>
<tr>
<td>GDSO</td>
<td>LX8</td>
<td>DC drain-source conductance (GDSO)</td>
</tr>
<tr>
<td>GMBSO</td>
<td>LX9</td>
<td>DC substrate transconductance (GMBSO)</td>
</tr>
<tr>
<td>GBDO</td>
<td>LX10</td>
<td>Conductance of the drain diode (GBDO)</td>
</tr>
<tr>
<td>GBSO</td>
<td>LX11</td>
<td>Conductance of the source diode (GBSO)</td>
</tr>
</tbody>
</table>

*Meyer and Charge Conservation Model Parameters*

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QB</td>
<td>LX12</td>
<td>Bulk charge (QB)</td>
</tr>
<tr>
<td>CQB</td>
<td>LX13</td>
<td>Bulk charge current (CQB)</td>
</tr>
<tr>
<td>QG</td>
<td>LX14</td>
<td>Gate charge (QG)</td>
</tr>
<tr>
<td>CQG</td>
<td>LX15</td>
<td>Gate charge current (CQG)</td>
</tr>
<tr>
<td>QD</td>
<td>LX16</td>
<td>Channel charge (QD)</td>
</tr>
<tr>
<td>CQD</td>
<td>LX17</td>
<td>Channel charge current (CQD)</td>
</tr>
<tr>
<td>CGGBO</td>
<td>LX18</td>
<td>(CGGBO = \frac{\partial Q_g}{\partial V_{gb}})</td>
</tr>
<tr>
<td>CGDBO</td>
<td>LX19</td>
<td>(CGDBO = \frac{\partial Q_g}{\partial V_{db}}, (\text{for Meyer } CGD = -CGDBO))</td>
</tr>
<tr>
<td>CGSBO</td>
<td>LX20</td>
<td>(CGSBO = \frac{\partial Q_g}{\partial V_{sb}}, (\text{for Meyer } CGS = -CGSBO))</td>
</tr>
<tr>
<td>CBGBO</td>
<td>LX21</td>
<td>(CBGBO = \frac{\partial Q_b}{\partial V_{gb}}, (\text{for Meyer } CGB = -CBGBO))</td>
</tr>
<tr>
<td>CBDBO</td>
<td>LX22</td>
<td>(CBDBO = \frac{\partial Q_b}{\partial V_{db}})</td>
</tr>
<tr>
<td>CBSBO</td>
<td>LX23</td>
<td>(CBSBO = \frac{\partial Q_b}{\partial V_{sb}})</td>
</tr>
<tr>
<td>QBD</td>
<td>LX24</td>
<td>Drain-bulk charge (QBD)</td>
</tr>
<tr>
<td>–</td>
<td>LX25</td>
<td>Drain-bulk charge current (CQBD) (not used in Star-Hspice releases after 95.3)</td>
</tr>
</tbody>
</table>
### MOSFET

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QBS</td>
<td>LX26</td>
<td>Source-bulk charge (QBS)</td>
</tr>
<tr>
<td>–</td>
<td>LX27</td>
<td>Source-bulk charge current (CQBS) (not used in Star-Hspice releases after 95.3)</td>
</tr>
<tr>
<td>CAP_BS</td>
<td>LX28</td>
<td>Bulk-source capacitance</td>
</tr>
<tr>
<td>CAP_BD</td>
<td>LX29</td>
<td>Bulk-drain capacitance</td>
</tr>
<tr>
<td>CQS</td>
<td>LX31</td>
<td>Channel charge current (CQS)</td>
</tr>
<tr>
<td>CDGBO</td>
<td>LX32</td>
<td>( \frac{\partial Q_d}{\partial V_{gb}} )</td>
</tr>
<tr>
<td>CDDBO</td>
<td>LX33</td>
<td>( \frac{\partial Q_d}{\partial V_{db}} )</td>
</tr>
<tr>
<td>CDSBO</td>
<td>LX34</td>
<td>( \frac{\partial Q_d}{\partial V_{sb}} )</td>
</tr>
</tbody>
</table>

### Saturable Core Element

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MU</td>
<td>LX0</td>
<td>Dynamic permeability (mu) ( \text{Weber/(amp-turn-meter)} )</td>
</tr>
<tr>
<td>H</td>
<td>LX1</td>
<td>Magnetizing force (H) ( \text{Ampere-turns/meter} )</td>
</tr>
<tr>
<td>B</td>
<td>LX2</td>
<td>Magnetic flux density (B) ( \text{Webers/meter}^2 )</td>
</tr>
</tbody>
</table>

### Saturable Core Winding

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEFF</td>
<td>LV1</td>
<td>Effective winding inductance (Henry)</td>
</tr>
<tr>
<td>IC</td>
<td>LV2</td>
<td>Initial condition</td>
</tr>
<tr>
<td>FLUX</td>
<td>LX0</td>
<td>Flux through winding (Weber-turn)</td>
</tr>
<tr>
<td>VOLT</td>
<td>LX1</td>
<td>Voltage across winding (Volt)</td>
</tr>
</tbody>
</table>
This chapter describes the options available for changing the Star-Hspice simulation. These options can modify various aspects of the simulation, including output types, accuracy, speed, and convergence. This chapter provides a complete reference of all options available in Star-Hspice from the .OPTION statement.

This chapter covers the following topics:

- Setting Control Options
- General Control Options
- Model Analysis Options
- DC Operating Point, DC Sweep, and Pole/Zero
- Transient and AC Small Signal Analysis
Setting Control Options

This section describes how to set control options.

.OPTIONS Statement

Control options are set in .OPTIONS statements. You can set any number of options in one .OPTIONS statement, and include any number of .OPTIONS statements in a Star-Hspice input netlist file. All the Star-Hspice control options are listed in Table 9-1. Descriptions of the options follow the table. Options that are relevant to a specific simulation type are also described in the appropriate DC, transient, and AC analysis chapters.

Generally, options default to 0 (OFF) when not assigned a value, either using .OPTIONS <opt> = <val> or by simply stating the option with no assignment: .OPTIONS <opt>. Option defaults are stated in the option descriptions in this section.

Syntax

.OPTIONS opt1 <opt2 opt3 ...>

Example

You can reset options by setting them to zero (.OPTIONS <opt> = 0). You can redefine an option by entering a new .OPTIONS statement for it; the last definition will be used. For example, set the BRIEF option to 1 to suppress printout, and reset BRIEF to 0 later in the input file to resume printout.

.OPTIONS BRIEF $ Sets BRIEF to 1 (turns it on)
* Netlist, models,
...
.OPTIONS BRIEF = 0 $ Turns BRIEF off
Options Keyword Summary

Table 9-1 lists the keywords for the .OPTIONS statement, grouped by their typical application.

The sections that follow the table provide a description of the options listed under each type of analysis.

### Table 9-1: .OPTION Keyword Application Table

<table>
<thead>
<tr>
<th>GENERAL CONTROL OPTIONS</th>
<th>MODEL ANALYSIS</th>
<th>DC OPERATING POINT, DC SWEEP, and POLE/ZERO</th>
<th>TRANSIENT and AC SMALL SIGNAL ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input, Output</td>
<td>Interfaces</td>
<td>General Accuracy Convergence</td>
<td>Accuracy Timestep</td>
</tr>
<tr>
<td>ACCT</td>
<td>ARTIST</td>
<td>DCAP ABSH CONVERGE</td>
<td>ABSH ABSVAR</td>
</tr>
<tr>
<td>Acout</td>
<td>CDS</td>
<td>SCALE ABSI CSHDC</td>
<td>ABSV, VNTOL DELMAX</td>
</tr>
<tr>
<td>ALT999</td>
<td>CSDF</td>
<td>TNOM ABSMOS DCFOR</td>
<td>ACCURATE DVDT</td>
</tr>
<tr>
<td>ALT9999</td>
<td>MEASOUT</td>
<td>ABSTOL DCHOLD</td>
<td>ACOUT FS</td>
</tr>
<tr>
<td>BRIEF</td>
<td>MOSFETS</td>
<td>ABSVDC DCON</td>
<td>CHGTOL FT</td>
</tr>
<tr>
<td>CO</td>
<td>POST</td>
<td>CVTOL DI DCFEP</td>
<td>CSHUNT, GSHUNT IMIN, ITL3</td>
</tr>
<tr>
<td>INGOLD</td>
<td>PROBE</td>
<td>DEFAD KCLTEST DCTRAN</td>
<td>IMAX, ITL4</td>
</tr>
<tr>
<td>LENNAM</td>
<td>PSF</td>
<td>DEFAS MAXAMP DV</td>
<td>DI ITL5</td>
</tr>
<tr>
<td>LIST</td>
<td>SDA</td>
<td>DEFL RELH GMAX</td>
<td>GMIN RELVAR</td>
</tr>
<tr>
<td>MEASDGt</td>
<td>ZUKEN</td>
<td>DEFNRD RELI GMINDC</td>
<td>GSHUNT, RMAX</td>
</tr>
<tr>
<td>NODE</td>
<td>DRFTS</td>
<td>RELMOS GRAMP</td>
<td>RMAX</td>
</tr>
<tr>
<td>NOELCK</td>
<td>Analysis</td>
<td>DEFPD RELV GSHUNT</td>
<td>MAXAMP SLOPETOL</td>
</tr>
<tr>
<td>NOMOD</td>
<td>ASPEC</td>
<td>DEFPS RELVDC ICSWEEP</td>
<td>RELH TIMERES</td>
</tr>
<tr>
<td>NOPAGE</td>
<td>LIMPTS</td>
<td>DEFW NEWTOL RELI</td>
<td></td>
</tr>
<tr>
<td>NOTOP</td>
<td>PARHIER</td>
<td>SCALM Matrix OFF</td>
<td>RELQ Algorithm</td>
</tr>
<tr>
<td>NUMDGt</td>
<td>SPICE</td>
<td>WL ITL1 RESMIN RELTOL DVTR</td>
<td></td>
</tr>
<tr>
<td>NXX</td>
<td>SEED</td>
<td>ITL2 RISETIME IMAX</td>
<td></td>
</tr>
<tr>
<td>OPTLST</td>
<td>Inductors</td>
<td>NOPIV Pole/Zero TRTOL IMIN</td>
<td></td>
</tr>
<tr>
<td>OPTS</td>
<td>Error</td>
<td>GENK PIVOT, SPARSE CSCAL VNTOL, ABSV</td>
<td></td>
</tr>
<tr>
<td>PATHNUM</td>
<td>BADCHR</td>
<td>KLIM FMAX Speed MAXORD</td>
<td></td>
</tr>
<tr>
<td>PLIM</td>
<td>DIAGNOSTIC</td>
<td>PIVREF FSCAL AUTOSTOP METHOD</td>
<td></td>
</tr>
</tbody>
</table>
### Table 9-1: .OPTION Keyword Application Table

<table>
<thead>
<tr>
<th>GENERAL CONTROL OPTIONS</th>
<th>MODEL ANALYSIS</th>
<th>DC OPERATING POINT, DC SWEEP, and POLE/ ZERO</th>
<th>TRANSIENT and AC SMALL SIGNAL ANALYSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST_VERSION</td>
<td>NOWARN</td>
<td>BJT (s )</td>
<td>PIVREL, GSCAL, BKPSIZ, MU, XMU</td>
</tr>
<tr>
<td>SEARCH</td>
<td>WARNLIMIT</td>
<td>EXPLI</td>
<td>PIVTOL, LSCAL, BYPASS</td>
</tr>
<tr>
<td>VERIFY</td>
<td></td>
<td>SPARSE, PIVOT</td>
<td>PZABS, BYTOL, Input, Output</td>
</tr>
<tr>
<td>CPU</td>
<td>Version</td>
<td>Diodes</td>
<td>PZTOL, FAST, INTERP</td>
</tr>
<tr>
<td>CPTIME</td>
<td>H9007</td>
<td>EXPLI</td>
<td>RITOL, ITLPZ, ITRPRT</td>
</tr>
<tr>
<td>EPSMIN</td>
<td></td>
<td>Input, Output</td>
<td>XnR, XnI, MBYPASS, UNWRAP</td>
</tr>
<tr>
<td>EXPMAX</td>
<td>CAPTAB</td>
<td>DCCAP</td>
<td>NEWTOL</td>
</tr>
<tr>
<td>LIMTIM</td>
<td></td>
<td>VFLOOR</td>
<td></td>
</tr>
</tbody>
</table>
General Control Options

Descriptions of the general control options follow. The descriptions are alphabetical by keyword under the sections presented in the table.

Input and Output Options

**ACCT**

Reports job accounting and runtime statistics at the end of the output listing. Simulation efficiency is determined by the ratio of output points to total iterations. Reporting is automatic unless you disable it.

Choices for ACCT are:

0  disables reporting
1  enables reporting
2  enables reporting of MATRIX statistics

**ACOUT**

AC output calculation method for the difference in values of magnitude, phase and decibels for prints and plots. The default value equals 1.

The default value, ACOUT = 1, selects the Star-Hspice method, which calculates the difference of the magnitudes of the values. The SPICE method, ACOUT = 0, calculates the magnitude of the differences.

**ALT999, ALT9999**

This option generates up to 1000 (ALT999) or 10,000 (ALT9999) unique output files from .ALTER runs. Star-Hspice appends a number from 0-999 (ALT999) or 0-9999 (ALT9999) to the extension of the output file. For example, for a .TRAN analysis with 50 .ALTER statements, the filenames would be filename.tr0, filename.tr1, ..., filename.tr50. Without this option, the files would be overwritten after the 36th .ALTER.
**BRIEF, NXX**

Stops printback of the data file until an .OPTIONS BRIEF = 0 or the .END statement is encountered. It also resets the options LIST, NODE and OPTS while setting NOMOD. BRIEF = 0 enables printback. NXX is the same as BRIEF.

**CO = x**

Sets the number of columns for printout: x can be either 80 (for narrow printout) or 132 (for wide carriage printouts). You also can set the output width by using the .WIDTH statement. The default value equals 80.

**INGOLD = x**

Specifies the printout data format. Use INGOLD = 2 for SPICE compatibility. The default value equals 0. Numeric output from Star-Hspice can be printed in one of three ways:

- **INGOLD = 0**
  Engineering format, exponents are expressed as a single character:
  
  \[1G = 1e9\] \[1X = 1e6\] \[1K = 1e3\] \[1M = 1e-3\]  
  \[1U = 1e-6\] \[1N = 1e-9\] \[1P = 1e-12\] \[1F = 1e-15\]

- **INGOLD = 1**
  Combined fixed and exponential format (G Format).
  Fixed format for numbers between 0.1 and 999. Exponential format for numbers greater than 999 or less than 0.1.

- **INGOLD = 2**
  Exclusively exponential format (SPICE2G style).
  Exponential format generates constant number sizes suitable for post-analysis tools.

Use .OPTIONS MEASDGT in conjunction with INGOLD to control the output data format of .MEASURE results.

**LENNAM = x**

Specifies the maximum length of names in the operating point analysis results printout. The default value equals 8. The maximum value of x is 16.
**LIST, VERIFY**

Produces an element summary listing of the input data to be printed. Calculates effective sizes of elements and the key values. LIST is suppressed by BRIEF. VERIFY is an alias for LIST.

**MEASDGT = x**

Used for formatting of the .MEASURE statement output in both the listing file and the .MEASURE output files (.ma0, .mt0, .ms0, and so on).

The value of x is typically between 1 and 7, although it can be set as high as 10. The default value equals 4.0. For example, if MEASDGT = 5, numbers displayed by .MEASURE are displayed as:

- Five decimal digits for numbers in scientific notation
- Five digits to the right of the decimal for numbers between 0.1 and 999

In the listing (.lis) file, all .MEASURE output values are in scientific notation, so .OPTIONS MEASDGT = 5 results in five decimal digits.

Use MEASDGT in conjunction with .OPTIONS INGOLD = x to control the output data format.

**NODE**

Causes a node cross reference table to be printed. NODE is suppressed by BRIEF. The table lists each node and all the elements connected to it. The terminal of each element is indicated by a code, separated from the element name with a colon (:). The codes are as follows:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Diode anode</td>
</tr>
<tr>
<td>-</td>
<td>Diode cathode</td>
</tr>
<tr>
<td>B</td>
<td>BJT base</td>
</tr>
<tr>
<td>B</td>
<td>MOSFET or JFET bulk</td>
</tr>
<tr>
<td>C</td>
<td>BJT collector</td>
</tr>
<tr>
<td>D</td>
<td>MOSFET or JFET drain</td>
</tr>
<tr>
<td>E</td>
<td>BJT emitter</td>
</tr>
</tbody>
</table>
G MOSFET or JFET gate
S BJT substrate
S MOSFET or JFET source

For example, part of a cross reference might look like:
1 M1:B D2:+ Q4:B

This line indicates that the bulk of M1, the anode of D2, and the base of Q4 are all connected to node 1.

NOELCK No element check; bypasses element checking to reduce preprocessing time for very large files.

NOMOD Suppresses the printout of model parameters

NOPAGE Suppresses page ejects for title headings

NOTOP Suppresses topology check resulting in increased speed for preprocessing very large files

NUMDGT = x Sets the number of significant digits printed for output variable values. The value of x is typically between 1 and 7, although it can be set as high as 10. The default value equals 4.0. This option does not affect the accuracy of the simulation.

NXX Stops printback of the data file until an .OPTIONS BRIEF = 0 or the .END statement is encountered. It also resets the options LIST, NODE and OPTS while setting NOMOD. BRIEF = 0 enables printback. NXX is the same as BRIEF.

OPTLST = x Outputs additional optimization information:
0 No information (default)
1 Prints parameter, Broyden update, and bisection results information
2 Prints gradient, error, Hessian, and iteration information
3 Prints all of the above and Jacobian
**OPTS**
Prints the current settings of all control options. If any of the default values of the options have been changed, the OPTS option prints the values actually used for the simulation. Suppressed by the BRIEF option.

**PATHNUM**
Prints subcircuit path numbers instead of path names.

**PLIM = x**
Specifies plot size limits for printer plots of current and voltage:

1. Finds a common plot limit and plots all variables on one graph at the same scale
2. Enables SPICE-type plots, in which a separate scale and axis are created for each plot variable

This option has no effect on graph data POST processing.

**POST_VERSION = x**
Sets the post-processing output version with values $x = 9601$ or $9007$. $x = 9007$ truncates the node name in the post-processor output file to be no longer than 16 characters. $x = 9601$ sets the node name length for the output file to be consistent with the input restrictions (1024 characters).

**POST_VERSION=2001**
Sets the post-processing output version with value of 2001. When using this option, the user will see the new output file header which includes the right number of output variables rather than **** when the number exceeds 9999. If .option post_version=2001 post=2 is set in the netlist, then the user will receive more accurate ascii results.

The syntax is:

`.option post_version=2001`

**SEARCH**
Sets the search path for libraries and included files. Star-Hspice automatically looks in the directory specified with .OPTIONS SEARCH for libraries referenced in the simulation.
**VERIFY**

Produces an element summary listing of the input data to be printed. Calculates effective sizes of elements and the key values. LIST is suppressed by BRIEF. VERIFY is an alias for LIST.
CPU Options

**CPTIME = x**

Sets the maximum CPU time, in seconds, allotted for this job. When the time allowed for the job exceeds CPTIME, the results up to that point are printed or plotted and the job is concluded. Use this option when uncertain about how long the simulation will take, especially when debugging new data files. Also see LIMTIM. The default value equals 1e7 (400 days).

**EPSMIN = x**

Specifies the smallest number that can be added or subtracted on a computer, a constant value. The default value equals 1e-28.

**EXPMAX = x**

 Specifies the largest exponent you can use for an exponential before overflow occurs. Typical value for an IBM platform is 350.

**LIMTIM = x**

Sets the amount of CPU time reserved for generating prints and plots in case a CPU time limit (CPTIME = x) causes termination. The default value equals 2 (seconds). This default is normally sufficient time for short printouts and plots.

Interface Options

**ARTIST = x**

ARTIST = 2 enables the Cadence Analog Artist interface. This option requires a specific license.

**CDS, SDA**

CDS = 2 produces a Cadence WSF ASCII format post-analysis file for Opus™. This option requires a specific license. SDA is the same as CDS.

**CSDF**

Selects Common Simulation Data Format (Viewlogic-compatible graph data file format)
**MEASOUT**  Outputs .MEASURE statement values and sweep parameters into an ASCII file for post-analysis processing by AvanWaves or other analysis tools. The output file is named \(<design>.mt#\), where # is incremented for each .TEMP or .ALTER block. For example, for a parameter sweep of an output load, measuring the delay, the .mt# file contains data for a delay versus fanout plot. The default value equals 1. You can set this option to 0 (off) in the *hspice.ini* file.

**MENTOR = x**  MENTOR = 2 enables the Mentor MSPICE-compatible ASCII interface. Requires a specific license.

**POST = x**  Enables storing of simulation results for analysis using the AvanWaves graphical interface or other methods. POST = 1 saves the results in binary. POST = 2 saves the results in ASCII format. POST = 3 saves the results in New Wave binary format. Set the POST option, and use the .PROBE statement to specify which data you want saved. The default value equals 1.

**PROBE**  Limits the post-analysis output to just the variables designated in .PROBE, .PRINT, .PLOT, and .GRAPH statements. By default, Star-Hspice outputs all voltages and power supply currents in addition to variables listed in .PROBE/.PRINT/.PLOT/.GRAPH statements. Use of PROBE significantly decreases the size of simulation output files.

**PSF = x**  Specifies whether Star-Hspice outputs binary or ASCII when Star-Hspice is run from Cadence Analog Artist. The value of \(x\) can be 1 or 2. If \(x\) is 2, Star-Hspice produces ASCII output. If .OPTIONS ARTIST PSF = 1, Star-Hspice produces binary output.

**SDA**  CDS = 2 produces a Cadence WSF ASCII format post-analysis file for Opus. This option requires a specific license. SDA is the same as CDS.
ZUKEN = x
If x is 2, enables the Zuken interactive interface. If x is 1, disables it. The default value equals 1.

Analysis Options

ASPEC
Sets Star-Hspice into ASPEC compatibility mode. With this option set, Star-Hspice can read ASPEC models and netlists and the results are compatible. The default value equals 0 (Star-Hspice mode).

Note: When the ASPEC option is set, the following model parameters default to ASPEC values:

ACM = 1: Default values for CJ, IS, NSUB, TOX, U0, UTRA are changed.
Diode Model:
TLEV = 1 affects temperature compensation of PB.
MOSFET Model:
TLEV = 1 affects PB, PHB, VTO, and PHI.

SCALM, SCALE:
Sets model scale factor to microns for length dimensions.
WL:
Reverses implicit order on MOSFET element of width and length.

LIMPTS = x
Sets the total number of points that you can print or plot in AC analysis. It is not necessary to set LIMPTS for DC or transient analysis, as Star-Hspice spools the output file to disk. The default value equals 2001.
PARHIER

Selects the parameter passing rules that control the evaluation order of subcircuit parameters. They only apply to parameters with the same name at different levels of subcircuit hierarchy. The options are:

LOCAL During analysis of a subcircuit, a parameter name specified in the subcircuit prevails over the same parameter name specified at a higher hierarchical level.

GLOBAL A parameter name specified at a higher hierarchical level prevails over the same parameter name specified at a lower level.

SPICE

Makes Star-Hspice compatible with Berkeley SPICE. When the option SPICE is set, the following options and model parameters are used:

Example of general parameters used with .OPTIONS SPICE:
TNOM = 27 DEFNRD = 1 DEFNRS = 1 INGOLD = 2
ACOUT = 0 DC
PIVOT PIVTOL = 1E-13 PIVREL = 1E-3 RELTOL = 1E-3
ITL1 = 100
ABSMOS = 1E-6 RELMOS = 1E-3 ABSTOL = 1E-12
VNTOL = 1E-6
ABSVDC = 1E-6 RELVDC = 1E-3 RELI = 1E-3

Example of transient parameters used with .OPTIONS SPICE:
DCAP = 1 RELQ = 1E-3 CHGTOL = 1E-14 ITL3 = 4 ITL4 = 10
ITL5 = 5000
FS = 0.125 FT = 0.125

Example of model parameters used with .OPTIONS SPICE:
For BJT: MJS = 0
For MOSFET, CAPOP = 0
LD = 0 if not user-specified
UTRA = 0 not used by SPICE for LEVEL = 2
NSUB must be specified
NLEV = 0 for SPICE noise equation
**SEED**
User-specified random number generator starting seed for Monte Carlo analysis. The minimum value is 1 and maximum value is 259200.

**Error Options**

**BADCHR**
Generates a warning when a nonprintable character is found in an input file.

**DIAGNOSTIC**
Logs the occurrence of negative model conductances.

**NOWARN**
Suppresses all warning messages except those generated from statements in .ALTER blocks.

**WARNLIMIT = x**
Limits the number of times that certain warnings appear in the output listing, thus reducing output listing file size. The value of x is the total number of warnings allowed for each warning type. The types of warning messages this limit applies to are:
- MOSFET has negative conductance
- Node conductance is zero
- Saturation current is too small
- Inductance or capacitance is too large
The default value equals 1.

**Version Options**

**H9007**
Sets general control option default values to correspond to the values for Star-Hspice Release H9007D. The EXPLI model parameter is not used when this option is set.
Model Analysis Options

General Options

DCAP
The DCAP option selects the equations used in calculating the depletion capacitance for Level 1 and 3 diodes and BJTs. See the individual device model chapters for information concerning the equations used.

MODSRH
If MODSRH=1, Hspice will not load the model and will consider it not referenced when the model is described by .model but does not appear in the netlist. This option will shorten the run time when many models are referenced but not called by an element in the netlist.

The default value is MODSRH=0. If MODSRH=1, then the read-in time will increase slightly.

eexample.sp:
  * modsrh used incorrectly
  .option post modsrh=1
  x1 net8 b c t6
  x10 a b net8 t6
  v1 a 0 pulse 3.3 0.0 1E-6 1E-9 1E-9 25E-6 50E-6
  v2 b 0 2
  v3 c 0 3
  .model nch nmos level=49 version=3.2
  .end

This input file will search for t6.inc automatically. If the model nch is used in t6.inc and MODSRH is set to 1, then Hspice will not load nch. Do not set MODSRH=1 in this type of included file call. Use this option in front of the .model card definition. This will be improved in the next release.
**SCALE**

Element scaling factor. This option will scale parameters used in element cards by its value. The default value equals 1.

**TNOM**

The reference temperature for the simulation. This is the temperature at which component derating is zero. The default is 25 degrees Celsius, or if .OPTION SPICE is enabled the default is 27 degrees Celsius.
MOSFET Control Options

CVTOL
Changes the number of numerical integration steps in the calculation of the gate capacitor charge for a MOSFET using CAPOP = 3. See the discussion of CAPOP = 3 in the chapter 18 for explicit equations and discussion.

DEFAD
Default value for MOSFET drain diode area. The default value equals 0.

DEFAS
Default value for MOSFET source diode area. The default value equals 0.

DEFL
Default value for MOSFET channel length. The default value equals $1 \times 10^{-4}$ m.

DEFNRD
Default value for the number of squares for the drain resistor on a MOSFET. The default value equals 0.

DEFNRS
Default value for the number of squares for the source resistor on a MOSFET. The default value equals 0.

DEFPD
Default value for MOSFET drain diode perimeter. The default value equals 0.

DEFPS
Default value for MOSFET source diode perimeter. The default value equals 0.

DEFW
Default value for MOSFET channel width. The default value equals $1 \times 10^{-4}$ m.

SCALM
Model scaling factor. This option will scale parameters defined in device model cards by its value. The default value equals 1. See the individual device model chapters for information about which parameters are scaled.

WL
This option changes the order of specifying MOS element VSIZE from the default order length-width to width-length. The default value equals 0.
Inductors

**GENK**  
Option for enabling automatic computation of second-order mutual inductance for several coupled inductors, where a value of 1 enables the calculation. The default value equals 1.

**KLIM**  
Minimum mutual inductance below which automatic second-order mutual inductance calculation will no longer proceed. KLIM is unitless (analogous to coupling strength specified in the K Element), and typical values for klim are between .5 and 0.0. The default value equals 0.01.

BJTs

**EXPLI**  
Current explosion model parameter. The PN junction characteristics above the explosion current are linearized, with the slope determined at the explosion point. This speeds up simulation and improves convergence. The default value equals 1e15 amp/AREAeff.

Diodes

**EXPLI**  
Current explosion model parameter. The PN junction characteristics above the explosion current are linearized, with the slope determined at the explosion point. This speeds up simulation and improves convergence. The default value equals 1e15 amp/AREAeff.
**DC Operating Point, DC Sweep, and Pole/Zero**

**Accuracy**

- **ABSH = x**  
  Sets the absolute current change through voltage defined branches (voltage sources and inductors). In conjunction with DI and RELH, ABSH is used to check for current convergence. The default value equals 0.0.

- **ABSI = x**  
  Sets the absolute branch current error tolerance in diodes, BJTs, and JFETs during DC and transient analysis. Decrease ABSI if accuracy is more important than convergence time.

  If you want an analysis with currents less than 1 nanoamp, change ABSI to a value at least two orders of magnitude smaller than the minimum expected current.

  The default value equals 1e-9 for KCLTEST = 0, 1e-6 for KCLTEST = 1.

- **ABSMOS = x**  
  Current error tolerance used for MOSFET devices in both DC and transient analysis. Star-Hspice uses the ABSMOS setting to determine if the drain-to-source current solution has converged. If the difference between the last and the present iteration’s drain-to-source current is less than ABSMOS, or if it is greater than ABSMOS, but the percent change is less than RELMOS, the drain-to-source current is considered converged. Star-Hspice then checks the other accuracy tolerances and, if all indicate convergence, the circuit solution at that timepoint is considered solved, and the next timepoint solution is calculated. For low power circuits, optimization, and single transistor simulations, set ABSMOS = 1e-12. The default value equals 1e-6 (amperes).
ABSTOL = x  
Sets the absolute error tolerance for branch currents. Decrease ABSTOL if accuracy is more important than convergence time.

ABSVDC = x  
Sets the absolute minimum voltage for DC and transient analysis. Decrease ABSVDC if accuracy is of more concern than convergence. If voltages less than 50 microvolts are required, ABSVDC can be reduced to two orders of magnitude less than the smallest desired voltage. This ensures at least two digits of significance. Typically ABSVDC need not be changed unless the circuit is a high voltage circuit. For 1000-volt circuits, a reasonable value can be 5 to 50 millivolts. The default value equals VNTOL (VNTOL default = 50 µV).

DI = x  
Sets the maximum iteration-to-iteration current change through voltage defined branches (voltage sources and inductors). This option is only applicable when the value of the ABSH control option is greater than 0. The default value equals 0.0.

KCLTEST  
Activates the KCL test (Kirchhoff’s Current Law) function. This test results in a longer simulation time, especially for large circuits, but provides a very accurate check of the solution. The default value equals 0.
When set to 1, Star-Hspice sets the following options:

- RELMOS and ABSMOS options are set to 0 (off).
- ABSI is set to 1e-16 A
- RELI is set to 1e-6

To satisfy the KCL test, the following condition must be satisfied for each node:

$$|\sum i_b| < RELI \cdot \sum |i_b| + ABSI$$

where the $i_b$s are the node currents.
**MAXAMP** = $x$  
Sets the maximum current through voltage defined branches (voltage sources and inductors). If the current exceeds the MAXAMP value, an error is issued. The default value equals 0.0.

**RELH** = $x$  
Sets relative current tolerance through voltage defined branches (voltage sources and inductors). It is used to check current convergence. This option is applicable only if the value of the ABSH control option is greater than zero. The default value equals 0.05.

**RELI** = $x$  
Sets the relative error/tolerance change from iteration to iteration to determine convergence for all currents in diode, BJT, and JFET devices. (RELMOS sets the tolerance for MOSFETs). This is the change in current from the value calculated at the previous timepoint. The default value equals 0.01 for KCLTEST = 0, 1e-6 for KCLTEST = 1.

**RELMOS** = $x$  
Sets the relative drain-to-source current error tolerance percent from iteration to iteration to determine convergence for currents in MOSFET devices. (RELI sets the tolerance for other active devices.) This is the change in current from the value calculated at the previous timepoint. RELMOS is only considered when the current is greater than the floor value, ABSMOS. The default value equals 0.05.

**RELV** = $x$  
Sets the relative error tolerance for voltages. When voltages or currents exceed their absolute tolerances, the RELV test is used to determine convergence. Increasing RELV increases the relative error. In general, RELV should be left at its default value. RELV controls simulator charge conservation. For voltages, RELV is the same as RELTOL. The default value equals 1e-3.
Relvdc = x
Sets the relative error tolerance for voltages. When voltages or currents exceed their absolute tolerances, the RELVDC test is used to determine convergence. Increasing RELVDC increases the relative error. In general, RELVDC should be left at its default value. RELVDC controls simulator charge conservation. The default value equals RELTOL (RELTOL default = 1e-3).

Matrix-Related

Itl1 = x
Sets the maximum DC iteration limit. Increasing this value is unlikely to improve convergence for small circuits. Values as high as 400 have resulted in convergence for certain large circuits with feedback, such as operational amplifiers and sense amplifiers. Something is usually wrong with a model if more than 100 iterations are required for convergence. Set .OPTION ACCT to obtain a listing of how many iterations are required for an operating point. The default value equals 200.

Itl2 = x
Sets the DC transfer curve iteration limit. Increasing the iteration limit can be effective in improving convergence only on very large circuits. The default value equals 50.

NoPiv
Prevents Star-Hspice from switching automatically to pivoting matrix factorization when a nodal conductance is less than PIVTOL. NOPIV inhibits pivoting. Also see PIVOT.
PIVOT = x

Provides different pivoting algorithm selections. These can be used effectively to reduce simulation time and achieve convergence in circuits that produce hard-to-solve matrix equations. The pivot algorithm is selected by setting PIVOT to one of the following values:

- 0: Original nonpivoting algorithm
- 1: Original pivoting algorithm
- 2: Pick largest pivot in row algorithm
- 3: Pick best in row algorithm
- 10: Fast nonpivoting algorithm, more memory required
- 11: Fast pivoting algorithm, more memory required than PIVOT values less than 11
- 12: Pick largest pivot in row algorithm, more memory required than for PIVOT values less than 12
- 13: Fast best pivot: faster, more memory required than for PIVOT values less than 13

The default value equals 10.

The fastest algorithm is PIVOT = 13, which can improve simulation time by up to ten times on very large circuits. However, the PIVOT = 13 option requires substantially more memory for the simulation. Some circuits with large conductance ratios, such as switching regulator circuits, might need pivoting. If PIVTOL = 0, Star-Hspice automatically changes from nonpivoting to a row pivot strategy upon detection of any diagonal matrix entry less than PIVTOL. This strategy provides the time and memory advantages of nonpivoting inversion, while avoiding unstable simulations and incorrect results. Use .OPTION NOPIV to prevent pivoting from being used under any circumstances.
For very large circuits, PIVOT = 10, 11, 12, or 13 can require excessive memory.

If Star-Hspice switches to pivoting during a simulation, the message “pivot change on the fly” is printed, followed by the node numbers causing the problem. Use .OPTION NODE to obtain a node-to-element cross reference.

SPARSE is the same as PIVOT.

**PIVREF**

Pivot reference. Used in PIVOT = 11, 12, 13 to limit the size of the matrix. The default value equals 1e+8.

**PIVREL = x**

Sets the maximum/minimum row/matrix ratio. Use only for PIVOT = 1. Large values for PIVREL can result in very long matrix pivot times. If the value is too small, however, no pivoting occurs. It is best to start with small values of PIVREL, using an adequate but not excessive value for convergence and accuracy. The default value equals 1E-20 (max = 1e-20, min = 1).

**PIVTOL = x**

Sets the absolute minimum value for which a matrix entry is accepted as a pivot. PIVTOL is used as the minimum conductance in the matrix when PIVOT = 0. The default value equals 1.0e-15.

**Note:** PIVTOL should always be less than GMIN or GMINDC. Values approaching 1 yield increased pivot.
**SPARSE = x**

Provides different pivoting algorithm selections. These can be used effectively to reduce simulation time and achieve convergence in circuits that produce hard-to-solve matrix equations. The pivot algorithm is selected by setting PIVOT to one of the following values:

- **0**: Original nonpivoting algorithm
- **1**: Original pivoting algorithm
- **2**: Pick largest pivot in row algorithm
- **3**: Pick best in row algorithm
- **10**: Fast nonpivoting algorithm, more memory required
- **11**: Fast pivoting algorithm, more memory required than PIVOT values less than 11
- **12**: Pick largest pivot in row algorithm, more memory required than for PIVOT values less than 12
- **13**: Fast best pivot: faster, more memory required than for PIVOT values less than 13

The default value equals 10.

The fastest algorithm is PIVOT = 13, which can improve simulation time by up to ten times on very large circuits. However, the PIVOT = 13 option requires substantially more memory for the simulation. Some circuits with large conductance ratios, such as switching regulator circuits, might need pivoting. If PIVTOL = 0, Star-Hspice automatically changes from nonpivoting to a row pivot strategy upon detection of any diagonal matrix entry less than PIVTOL. This strategy provides the time and memory advantages of nonpivoting inversion, while avoiding unstable simulations and incorrect results. Use .OPTION NOPIV to prevent pivoting from being used under any circumstances.
For very large circuits, PIVOT = 10, 11, 12, or 13 can require excessive memory.

If Star-Hspice switches to pivoting during a simulation, the message “pivot change on the fly” is printed, followed by the node numbers causing the problem. Use .OPTION NODE to obtain a node-to-element cross reference.

SPARSE is the same as PIVOT.
### Input and Output

**CAPTAB**
- Prints table of single plate nodal capacitance for diodes, BJTs, MOSFETs, JFETs and passive capacitors at each operating point.

**DCCAP**
- Used to generate C-V plots and to print out the capacitance values of a circuit (both model and element) during a DC analysis. C-V plots are often generated using a DC sweep of the capacitor. The default value equals 0 (off).

**VFLOOR = x**
- Sets a lower limit for the voltages that are printed in the output listing. All voltages lower than VFLOOR are printed as 0. This only affects the output listing; the minimum voltage used in a simulation is set by VNTOL (ABSV).
Convergence

CONVERGE Invokes different methods for solving nonconvergence problems:

■ CONVERGE = -1
  together with DCON = -1, disables autoconvergence

■ CONVERGE = 1
  uses the Damped Pseudo Transient Algorithm. If the simulation fails to converge within the amount of CPU time set by the CPTIME control option, the simulation halts.

■ CONVERGE = 2
  uses a combination of DCSTEP and GMINDC ramping

■ CONVERGE = 3
  invokes the source stepping method

Even if it is not set in an .OPTIONS statement, the CONVERGE option is activated in the event of a matrix floating point overflow, or a timestep too small error. The default value equals 0.

In the event of a matrix floating point overflow,
Star-Hspice sets CONVERGE = 1.

CSHDC The same option as CSHUNT, but is used only with option CONVERGE.

DCFOR = x Used in conjunction with the DCHOLD option and the .NODESET statement to enhance the DC convergence properties of a simulation. DCFOR sets the number of iterations that are to be calculated after a circuit converges in the steady state. Since the number of iterations after convergence is usually zero, DCFOR adds iterations (and computational time) to the calculation of the DC circuit solution. DCFOR helps ensure that a circuit has actually, not falsely, converged. The default value equals 0.
**DCHOLD = x**

DFOR and DCHOLD are used together for the initialization process of a DC analysis. They enhance the convergence properties of a DC simulation. DCFOR and DCHOLD work together with the .NODESET statement.

The DCHOLD option specifies the number of iterations a node is to be held at the voltage values specified by the .NODESET statement. The effects of DCHOLD on convergence differ according to the DCHOLD value and the number of iterations needed to obtain DC convergence. If a circuit converges in the steady state in fewer than DCHOLD iterations, the DC solution includes the values set by the .NODESET statement. However, if the circuit requires more than DCHOLD iterations to converge, the values set in the .NODESET statement are ignored and the DC solution is calculated with the .NODESET fixed source voltages open circuited. The default value equals 1.

**DCON = X**

In the case of convergence problems, Star-Hspice automatically sets DCON = 1 and the following calculations are made:

\[
DV = max\left(0.1, \frac{V_{max}}{50}\right), \text{ if } DV = 1000
\]

\[
GRAMP = max\left(6, \log_{10}\left(\frac{I_{max}}{GMINDC}\right)\right)
\]

\[
ITL1 = ITL1 + 20 \cdot GRAMP
\]

where \(V_{max}\) is the maximum voltage and \(I_{max}\) is the maximum current.
If convergence problems still exist, Star-Hspice sets DCON = 2, which is the same as the above except DV = 1e6. The above calculations are used for DCON = 1 or 2. DCON = 1 is automatically invoked if the circuit fails to converge. DCON = 2 is invoked if DCON = 1 fails.

If the circuit contains uninitialized flip-flops or discontinuous models, the simulation might be unable to converge. Setting DCON = -1 and CONVERGE = -1 disables the autoconvergence algorithm and provides a list of nonconvergent nodes and devices.

\[DCSTEP = x\]

Used to convert DC model and element capacitors to a conductance to enhance DC convergence properties. The value of the element capacitors are all divided by DCSTEP to obtain a DC conductance model. The default value equals 0 (seconds).
**DCTRAN**

Invokes different methods for solving nonconvergence problems:

- **CONVERGE = -1**
  
  together with DCON = -1, disables autoconvergence

- **CONVERGE = 1**
  
  uses the Damped Pseudo Transient Algorithm. If the simulation fails to converge within the amount of CPU time set by the CPTIME control option, the simulation halts.

- **CONVERGE = 2**
  
  uses a combination of DCSTEP and GMINDC ramping

- **CONVERGE = 3**
  
  invokes the source stepping method

Even if it is not set in an .OPTIONS statement, the CONVERGE option is activated in the event of a matrix floating point overflow, or a timestep too small error. The default value equals 0.

In the event of a matrix floating point overflow, Star-Hspice sets CONVERGE = 1.

**DCTRAN** is an alias for **CONVERGE**.

**DV = x**

The maximum iteration-to-iteration voltage change for all circuit nodes in both DC and transient analysis. Values of 0.5 to 5.0 can be necessary for some high-gain bipolar amplifiers to achieve a stable DC operating point. CMOS circuits frequently require a value of about 1 volt for large digital circuits. The default value equals 1000 (or 1e6 if DCON = 2).
$GMAX = x$

The conductance in parallel with the current source used for .IC and .NODESET initialization conditions circuitry. Some large bipolar circuits can require $GMAX$ set to 1 for convergence. The default value equals 100 (mho).

$GMINDC = x$

A conductance that is placed in parallel with all pn junctions and all MOSFET nodes for DC analysis. $GMINDC$ helps overcome DC convergence problems caused by low values of off conductance for pn junctions and MOSFET devices. $GRAMP$ can be used to reduce $GMINDC$ by one order of magnitude for each step. $GMINDC$ can be set between $1e^{-4}$ and PIVTOL. The default value equals $1e^{-12}$.

Large values of $GMINDC$ can cause unreasonable circuit response. If large values are required for convergence, a bad model or circuit is suspect. In the event of a matrix floating point overflow, if $GMINDC$ is $1.0e^{-12}$ or less, Star-Hspice sets it to $1.0e^{-11}$.

$GMINDC$ is manipulated by Star-Hspice in autoconverge mode.

$GRAMP = x$

Value is set by Star-Hspice during the autoconvergence procedure. $GRAMP$ is used in conjunction with the $GMINDC$ convergence control option to find the smallest value of $GMINDC$ that results in DC convergence.

$GRAMP$ specifies the conductance range over which $GMINDC$ is to be swept during a DC operating point analysis. Star-Hspice substitutes values of $GMINDC$ over this range and simulates at each value. It then picks the lowest value of $GMINDC$ that resulted in the circuit converging in the steady state.
DC Operating Point, DC Sweep, and Pole/Zero

If GMINDC is swept between 1e-12 mhos (the default) and 1e-6 mhos, GRAMP is set to 6 (the value of the exponent difference between the default and the maximum conductance limit). In this case, GMINDC is first set to 1e-6 mhos, and the circuit is simulated. If convergence is achieved, GMINDC is next set to 1e-7 mhos, and the circuit simulated again. The sweep continues until a simulation has been performed at all values on the GRAMP ramp. If the combined conductance of GMINDC and GRAMP is greater than 1e-3 mho, a false convergence can occur. The default value equals 0.

**GSHUNT**

Conductance added from each node to ground. The default value is zero. Add a small GSHUNT to each node to possibly solve “Timestep too small” problems caused by high frequency oscillations or by numerical noise.

**ICSWEEP**

For a parameter or temperature sweep, saves the results of the current analysis for use as the starting point in the next analysis in the sweep. When ICSWEEP = 1, the current results are used in the next analysis. When ICSWEEP = 0, the results of the current analysis are not used in the next analysis. The default value equals 1.

**NEWTOL**

Calculates one more iterations past convergence for every DC solution and timepoint circuit solution calculated. When NEWTOL is not set, once convergence is determined, the convergence routine is ended and the next program step begun. The default value equals 0.
Specifying Simulation Options

OFF

Initializes the terminal voltages of all active devices to zero if they are not initialized to other values. For example, if the drain and source nodes of a transistor are not both initialized using .NODESET or .IC statements or by connecting them to sources, then the OFF option initializes all of the nodes of the transistor to zero. The OFF option is checked before element IC parameters, so if an element IC parameter assignment exists for a particular node, the node is initialized to the element IC parameter value even if it was previously set to zero by the OFF option. (The element parameter OFF can be used to initialize the terminal voltages to zero for particular active devices).

The OFF option is used to help find exact DC operating point solutions for large circuits.

RESMIN = x

Specifies the minimum resistance value for all resistors, including parasitic and inductive resistances. The default value equals 1e-5 (ohm). Range: 1e-15 to 10 ohm.

Pole/Zero Control Options

CSCAL

Sets the capacitance scale. Capacitances are multiplied by CSCAL. The default value equals 1e+12 (thus, by default, all capacitances are entered in units of pF).

FMAX

Sets the limit for maximum pole and zero angular frequency value. The default value equals 1.0e+12 rad/sec.

FSCAL

Sets the frequency scale. Frequency is multiplied by FSCAL. The default value equals 1e-9 (thus, by default, all frequencies are entered in units of GHz).

GSCAL

Sets the conductance scale. Conductances are multiplied by GSCAL. Resistances are divided by GSCAL. The default value equals 1e+3 (thus, by default, all resistances are entered in units of kΩ).

LSCAL

Sets inductance scale. Inductances are multiplied by LSCAL. The default value equals 1e+6 (thus, by default, all inductances are entered in units of µH).

The scale factors must satisfy the following relations:

\[ GSCAL = CSCAL \cdot FSCAL \]

\[ GSCAL = \frac{1}{LSCAL} \cdot FSCAL \]

If scale factors are changed, it might be necessary to modify the initial Muller points \((X0R, X0I), (X1R, X1I)\) and \((X2R, X2I)\), even though Star-Hspice multiplies initial values by \((1e-9/GSCAL)\).

PZABS

Sets absolute tolerances for poles and zeros. This option affects the low frequency poles or zeros. It is used as follows:

If \(|X_{real}| + |X_{imag}| < PZABS\),

then \(X_{real} = 0\) and \(X_{imag} = 0\).

It is also used for convergence tests. The default value equals 1e-2.

PZTOL

Sets the relative error tolerance for poles or zeros. The default value equals 1.0e-6.

RITOL

Sets the minimum ratio value for (real/imaginary) or (imaginary/real) parts of the poles or zeros. RITOL is used as follows:

If \(|X_{imag}| \leq RITOL \cdot |X_{real}|\), then \(X_{imag} = 0\)

If \(|X_{real}| \leq RITOL \cdot |X_{imag}|\), then \(X_{real} = 0\)

The default value equals 1.0e-6.
(X0R,X0I), The three complex starting points in the Muller pole/zero analysis algorithm are:

(X1R,X1I), X0R = -1.23456e6  X0I = 0.0
(X2R,X2I)  X1R = -1.23456e5  X1I = 0.0
        X2R = +.23456e6  X2I = 0.0

These initial points are multiplied by FSCAL.
Transient and AC Small Signal Analysis

Accuracy

$ABSH = x$
Sets the absolute current change through voltage defined branches (voltage sources and inductors). In conjunction with DI and RELH, ABSH is used to check for current convergence. The default value equals 0.0.

$ABSV = x$
Sets the absolute minimum voltage for DC and transient analysis. Decrease VNTOL if accuracy is of more concern than convergence. If voltages less than 50 microvolts are required, VNTOL can be reduced to two orders of magnitude less than the smallest desired voltage, ensuring at least two digits of significance. Typically, VNTOL need not be changed unless the circuit is a high voltage circuit. For 1000 volt circuits, a reasonable value can be 5 to 50 millivolts. ABSV is the same as VNTOL. The default value equals 50 (microvolts).

$ACCURATE$
Selects a time algorithm that uses $LVLTIM = 3$ and $DVDT = 2$ for circuits such as high-gain comparators. Circuits that combine high gain with large dynamic range should use this option to guarantee solution accuracy. When ACCURATE is set to 1, it sets the following control options:
- $LVLTIM = 3$
- $DVDT = 2$
- $RELVAR = 0.2$
- $ABSVAR = 0.2$
- $FT = 0.2$
- $RELMOS = 0.01$
The default value equals 0.
ACOUT

AC output calculation method for the difference in values of magnitude, phase and decibels for prints and plots. The default value equals 1.

The default value, ACOUT = 1, selects the Star-Hspice method, which calculates the difference of the magnitudes of the values. The SPICE method, ACOUT = 0, calculates the magnitude of the differences.

CHGTOL = x

Sets the charge error tolerance when LVLTIM = 2 is set. CHGTOL, along with RELQ, sets the absolute and relative charge tolerance for all Star-Hspice capacitances. The default value equals 1e-15 (coulomb).

CSHUNT

Capacitance added from each node to ground. Adding a small CSHUNT to each node can solve some “internal timestep too small” problems caused by high-frequency oscillations or numerical noise. The default value equals 0.

DI = x

Sets the maximum iteration-to-iteration current change through voltage defined branches (voltage sources and inductors). This option is only applicable when the value of the DI control option is greater than 0. The default value equals 0.0.

GMIN = x

Sets the minimum conductance allowed for in a transient analysis time sweep. The default value equals 1e-12.

GSHUNT

Conductance added from each node to ground. The default value is zero. Adding a small GSHUNT to each node can solve some “internal timestep too small” problems caused by high frequency oscillations or by numerical noise.

MAXAMP = x

Sets the maximum current through voltage defined branches (voltage sources and inductors). If the current exceeds the MAXAMP value, an error is issued. The default value equals 0.0.
**RELH = x**
Sets relative current tolerance through voltage defined branches (voltage sources and inductors). RELH is used to check current convergence. This option is applicable only if the value of the ABSH control option is greater than zero. The default value equals 0.05.

**RELI = x**
Sets the relative error/tolerance change from iteration to iteration to determine convergence for all currents in diode, BJT, and JFET devices. (RELMOS sets the tolerance for MOSFETs). This is the change in current from the value calculated at the previous timepoint. The default value equals 0.01 for KCLTEST = 0, 1e-4 for KCLTEST = 1.

**RELQ = x**
Used in the local truncation error timestep algorithm (LVLTIM = 2). RELQ changes the size of the timestep. If the capacitor charge calculation of the present iteration exceeds that of the past iteration by a percentage greater than the value of RELQ, the internal timestep (Delta) is reduced. The default value equals 0.01.

**RELTOL, RELV**
Sets the relative error tolerance for voltages. RELV is used in conjunction with the ABSV control option to determine voltage convergence. Increasing RELV increases the relative error. RELV is the same as RELTOL. Options RELI and RELVDC default to the RELTOL value. The default value equals 1e-3.

**RISETIME**
Specifies the smallest risetime of the signal. .OPTION RISETIME = x. Currently, used only in transmission line models. In the U Element, the number of lumps is determined by:

\[
\text{MIN}\left[20, 1 + \left(\frac{T_{Deff}}{\text{RISETIME}}\right) \cdot 20\right]
\]

where \(T_{Deff}\) is the total end-to-end delay in a transmission line. In the W Element, RISETIME is used only if \(R_s\) or \(G_d\) is nonzero. In such cases, RISETIME is used to determine the maximum frequency content of signals.
Specifying Simulation Options

**TRTOL = x**
Used in the local truncation error timestep algorithm (LVLTIM = 2). TRTOL is a multiplier of the internal timestep generated by the local truncation error timestep algorithm. TRTOL reduces simulation time, while maintaining accuracy. It is a factor that estimates the amount of error introduced by truncating the Taylor series expansion used in the algorithm. This error is a reflection of what the minimum value of the timestep should be to reduce simulation time and maintain accuracy. The range of TRTOL is 0.01 to 100, with typical values being in the 1 to 10 range. If TRTOL is set to 1, the minimum value, a very small timestep is used. As the setting of TRTOL increases, the timestep size increases. The default value equals 7.0.

**VNTOL = x, ABSV**
Sets the absolute minimum voltage for DC and transient analysis. Decrease VNTOL if accuracy is of more concern than convergence. If voltages less than 50 microvolts are required, VNTOL can be reduced to two orders of magnitude less than the smallest desired voltage, ensuring at least two digits of significance. Typically, VNTOL need not be changed unless the circuit is a high voltage circuit. For 1000 volt circuits, a reasonable value can be 5 to 50 millivolts. ABSV is the same as VNTOL. The default value equals 50 (microvolts).

**Speed**

**AUTOSTOP**
Stops the transient analysis when all TRIG-TARG and FIND-WHEN measure functions are calculated. This option can result in a substantial CPU time reduction. If the data file contains measure functions such as AVG, RMS, MIN, MAX, PP, ERR, ERR1,2,3, and PARAM, then AUTOSTOP is disabled.

**BKPSIZ = x**
Sets the size of the breakpoint table. The default value equals 5000.
**BYPASS**

Speeds up simulation by not updating the status of latent devices. Setting `.OPTION BYPASS = 1` enables bypassing. **BYPASS** applies to MOSFETs, MESFETs, JFETs, BJTs, and diodes. The default value equals 1.

**Note:** Use the **BYPASS** algorithm cautiously. For some types of circuits it can result in nonconvergence problems and loss of accuracy in transient analysis and operating point calculations.

**BYTOL = x**

Specifies the tolerance for the voltage at which a MOSFET, MESFET, JFET, BJT, or diode is considered latent. Star-Hspice does not update the status of latent devices. The default value equals MBYPASSxVNTOL.

**FAST**

Speeds up simulation by not updating the status of latent devices. This option is applicable for MOSFETs, MESFETs, JFETs, BJTs, and diodes. The default value equals 0.

A device is considered to be latent when its node voltage variation from one iteration to the next is less than the value of either the **BYTOL** control option or the **BYPASSTOL** element parameter. (When **FAST** is on, Star-Hspice sets **BYTOL** to different values for different types of device models.)

In addition to the **FAST** option, the input preprocessing time can be reduced by the options **NOTOP** and **NOELCK**. Increasing the value of the **MBYPASS** option or the **BYTOL** option setting also helps simulations run faster, but can reduce accuracy.

**ITLPZ**

Sets the pole/zero analysis iteration limit. The default value equals 100.
**MBYPASS** = x  
Used to compute the default value for the BYTOL control option:

\[ \text{BYTOL} = \text{MBYPASS} \times \text{VNTOL} \]

Also multiplies voltage tolerance RELV. MBYPASS should be set to about 0.1 for precision analog circuits. The default value equals 1 for DVDT = 0, 1, 2, or 3. The default value equals 2 for DVDT = 4.

**Timestep**

**ABSVAR** = x  
Sets the limit on the maximum voltage change from one time point to the next. Used with the DVDT algorithm. If the simulator produces a convergent solution that is greater than ABSVAR, the solution is discarded, the timestep is set to a smaller value, and the solution is recalculated. This is called a timestep reversal. The default value equals 0.5 (volts).

**DELMAX** = x  
Sets the maximum value for the internal timestep Delta. Star-Hspice automatically sets the DELMAX value based on various factors, which are listed in “Timestep Control for Accuracy” on page 11-26. This means that the initial DELMAX value shown in the Star-Hspice output listing is generally not the value used for simulation.
**DVDT**

Allow the timestep to be adjusted based on node voltage rates of change. Choices are:

- 0 - original algorithm
- 1 - fast
- 2 - accurate
- 3, 4 - balance speed and accuracy

The default value equals 4.

The ACCURATE option also increases the accuracy of the results.

**FS = x**

Sets the fraction of a timestep (TSTEP) that Delta (the internal timestep) is decreased for the first time point of a transient. Decreasing the FS value helps circuits that have timestep convergence difficulties. It also is used in the DVDT = 3 method to control the timestep.

\[
Delta = FS \times [\text{MIN}(TSTEP, \text{DELMAX}, \text{BKPT})]
\]

where DELMAX is specified and BKPT is related to the breakpoint of the source. TSTEP is set in the .TRAN statement. The default value equals 0.25.

**FT = x**

Sets the fraction of a timestep (TSTEP) by which Delta (the internal timestep) is decreased for an iteration set that does not converge. It is also used in DVDT = 2 and DVDT = 4 to control the timestep. The default value equals 0.25.
**Specifying Simulation Options**

**Transient and AC Small Signal Analysis**

\[ IMIN = x, \]
\[ ITL3 = x \]

Determines the timestep in the algorithms used for transient analysis simulations. IMIN sets a lower limit on the number of iterations required to obtain convergence. If the number of iterations is less than IMIN, the internal timestep, Delta, is doubled. This option is useful for decreasing simulation times in circuits where the nodes are stable most of the time, such as digital circuits. If the number of iterations is greater than IMIN, the timestep is kept the same unless the option IMAX is exceeded (see IMAX). ITL3 is the same as IMIN. The default value equals 3.0.

\[ IMAX = x, \]
\[ ITL4 = x \]

Determines the maximum timestep in the timestep algorithms used for transient analysis simulations. IMAX sets an upper limit on the number of iterations allowed to obtain a convergent solution at a timepoint. If the number of iterations needed is greater than IMAX, the internal timestep Delta is decreased by a factor equal to the transient control option FT, and a new solution is calculated using the new timestep. IMAX also works in conjunction with the transient control option IMIN. ITL4 is the same as IMAX. The default value equals 8.0.

\[ ITL3 = x \]

Determines the timestep in the algorithms used for transient analysis simulations. IMIN sets a lower limit on the number of iterations required to obtain convergence. If the number of iterations is less than IMIN, the internal timestep, Delta, is doubled. This option is useful for decreasing simulation times in circuits where the nodes are stable most of the time, such as digital circuits. If the number of iterations is greater than IMIN, the timestep is kept the same unless the option IMAX is exceeded (see IMAX). ITL3 is the same as IMIN. The default value equals 3.0.
**ITL4 = x**

Determines the maximum timestep in the timestep algorithms used for transient analysis simulations. IMAX sets an upper limit on the number of iterations allowed to obtain a convergent solution at a timepoint. If the number of iterations needed is greater than IMAX, the internal timestep Delta is decreased by a factor equal to the transient control option FT, and a new solution is calculated using the new timestep. IMAX also works in conjunction with the transient control option IMIN. ITL4 is the same as IMAX. The default value equals 8.0.

**ITL5 = x**

Sets the transient analysis total iteration limit. If a circuit uses more than ITL5 iterations, the program prints all results to that point. The default allows an infinite number of iterations. The default value equals 0.0.

**RELVAR = x**

Used with ABSVAR and the timestep algorithm option DVDT. RELVAR sets the relative voltage change for LVLTIM = 1 or 3. If the nodal voltage at the current time point exceeds the nodal voltage at the previous time point by RELVAR, the timestep is reduced and a new solution at a new time point is calculated. The default value equals 0.30 (30%).

**RMAX = x**

Sets the TSTEP multiplier, which determines the maximum value, DELMAX, that can be used for the internal timestep Delta:

\[ \text{DELMAX} = \text{TSTEP} \times \text{RMAX} \]

The default value equals 5 when dvdt = 4 and lvlim = 1, otherwise, default = 2.

**RMIN = x**

Sets the minimum value of Delta (internal timestep). An internal timestep smaller than RMINxTSTEP results in termination of the transient analysis with the error message “internal timestep too small”. Delta is decreased by the amount set by the FT option if the circuit has not converged in IMAX iterations. The default value equals 1.0e-9.
**Algorithm**

**SLOPETOL = x**  
Sets a lower limit for breakpoint table entries in a piecewise linear (PWL) analysis. If the difference in the slopes of two consecutive PWL segment is less than the SLOPETOL value, the breakpoint table entry for the point between the segments is ignored. The default value equals 0.5

**TIMERES = x**  
Sets a minimum separation between breakpoint values for the breakpoint table. If two breakpoints are closer together in time than the TIMERES value, only one of them is entered in the breakpoint table. The default value equals 1 ps.

**DVTR**  
Allows the use of voltage limiting in transient analysis. The default value equals 1000.

**IMAX = x, ITL4 = x**  
Determines the maximum timestep in the timestep algorithms used for transient analysis simulations. IMAX sets an upper limit on the number of iterations allowed to obtain a convergent solution at a timepoint. If the number of iterations needed is greater than IMAX, the internal timestep Delta is decreased by a factor equal to the transient control option FT, and a new solution is calculated using the new timestep. IMAX also works in conjunction with the transient control option IMIN. ITL4 is the same as IMAX. The default value equals 8.0.
**IMIN = x,**
**ITL3 = x**
Determines the timestep in the algorithms used for transient analysis simulations. IMIN sets a lower limit on the number of iterations required to obtain convergence. If the number of iterations is less than IMIN, the internal timestep, Delta, is doubled. This option is useful for decreasing simulation times in circuits where the nodes are stable most of the time, such as digital circuits. If the number of iterations is greater than IMIN, the timestep is kept the same unless the option IMAX is exceeded (see IMAX). ITL3 is the same as IMIN. The default value equals 3.0.

**LVLTIM = x**
Selects the timestep algorithm used for transient analysis. If LVLTIM = 1, the DVDT timestep algorithm is used. If LVLTIM = 2, the local truncation error timestep algorithm is used. If LVLTIM = 3, the DVDT timestep algorithm with timestep reversal is used.

If the GEAR method of numerical integration and linearization is used, LVLTIM = 2 is selected. If the TRAP linearization algorithm is used, LVLTIM 1 or 3 can be selected. Using LVLTIM = 1 (the DVDT option) helps avoid the “internal timestep too small” convergence problem. Using LVLTIM = 1 (the DVDT option) helps avoid the “internal timestep too small” nonconvergence problem. The local truncation algorithm (LVLTIM = 2), however, provides a higher degree of accuracy and prevents errors propagating from time point to time point, which can sometimes result in an unstable solution. The default value equals 1.

**MAXORD = x**
Sets the maximum order of integration when the GEAR method is used (see METHOD). The value of x can be either 1 or 2. If MAXORD = 1, the backward Euler method of integration is used. MAXORD = 2, however, is more stable, accurate, and practical. The default value equals 2.0.
METHOD = name
Sets the numerical integration method used for a transient analysis to either GEAR or TRAP. To use GEAR, set METHOD = GEAR. This automatically sets LVLTIM = 2.

(You can change LVLTIM from 2 to 1 or 3 by setting LVLTIM = 1 or 3 after the METHOD = GEAR option. This overrides the LVLTIM = 2 setting made by METHOD = GEAR.)

TRAP (trapezoidal) integration generally results in reduced program execution time, with more accurate results. However, trapezoidal integration can introduce an apparent oscillation on printed or plotted nodes that might not be caused by circuit behavior. To test if this is the case, run a transient analysis with a small timestep. If the oscillation disappears, it was due to the trapezoidal method.

The GEAR method acts as a filter, removing the oscillations found in the trapezoidal method. Highly nonlinear circuits such as operational amplifiers can require very long execution times with the GEAR method. Circuits that are not convergent with trapezoidal integration often converge with GEAR. The default value equals TRAP (trapezoidal).

MU = x,
XMU = x
The coefficient for trapezoidal integration. The range for MU is 0.0 to 0.5. XMU is the same as MU. The default value equals 0.5.
Input and Output

INTERP

Limits output to post-analysis tools, such as Cadence or Zuken, to only the .TRAN timestep intervals. By default, Star-Hspice outputs all convergent iterations. INTERP typically produces a much smaller design.tr# file.

You should use INTERP = 1 with caution when the .MEASURE statement is present. Star-Hspice computes measure statements using the postprocessing output. Reducing postprocessing output may lead to interpolation errors in measure results.

Note that when running a data driven transient analysis (.TRAN DATA statement) within optimization routines, INTERP is forced to 1. As a result, all measurement results are made at the time points of the data in the data driven sweep. If the measurement needs to use all converged internal timesteps, e.g. AVG or RMS calculations, you should set ITRPRT = 1.

ITRPRT

Prints output variables at their internal timepoint values. Using this option can generate a long output list.

MEASFAIL

Produces “0” into .mt#, .ms# or .ma# and prints “failed” to the listing file when measfail=0. Prints “failed” into .mt#, .ms# or .ma# file and the listing file when measfail=1.

Note that the default value is 1.

Use the following syntax:
.option measfail=1 | 0
**PUTMEAS**

Allows the user to control the output variables listed in the .measure statement.

The syntax is:

```
.option putmeas=0 or (1)
```

The default value is 1.

0: will not produce the values of the variables which are listed in the .measure statement into the corresponding output file such as .tr#, .ac# or .sw#. This option will decrease the size of the output file.

1: will produce the values of the variables which are listed in the .measure statement into the corresponding output file such as .tr#, .ac# or .sw#. This option is similar to the output produced by Hspice 2000.4.

**UNWRAP**

Displays phase results in AC analysis in unwrapped form (with a continuous phase plot). This allows accurate calculation of group delay. Note that group delay is always computed based on unwrapped phase results, even if the UNWRAP option is not set.
Chapter 10

DC Initialization and Operating Point Analysis

This chapter describes DC initialization and operating point analysis. It covers the following topics:

- Understanding the Simulation Flow
- Performing Initialization and Analysis
- Using DC Initialization and Operating Point Statements
- .DC Statement—DC Sweeps
- Using Other DC Analysis Statements
- Setting DC Initialization Control Options
- Specifying Accuracy and Convergence
- Reducing DC Errors
- Diagnosing Convergence Problems
Understanding the Simulation Flow

Figure 10-1 illustrates the simulation flow for Star-Hspice.

**Figure 10-1: DC Initialization and Operating Point Analysis**

Simulation Flow

```
Simulation Experiment

  DC

  Transient

  AC

Op. point simulation

  Sweep analysis

  DC-related AC small-signal analysis

  Monte Carlo analysis

Options:

  Tolerance
  ABSI
  ABSMOS
  ABSTOL
  ABSVDC
  KCLTEST
  RELI
  RELMOS
  RELV
  RELVDC

  Matrix
  ITL1
  NOPIV
  PIVOT
  PIVREF
  PIVREL
  PIVTOL
  SPARSE
  NOTOP

  Convergence
  CONVERGE
  CSHDC
  DCFOR
  DCHOLD
  DCON
  DCSTEP
  DCTRAN
  DV

  Limit
  GMAX
  GMINDC
  GRAMP
  GSHUNT
  ICSWEEP
  NEWTOL
  OFF
  RESMIN

.SENS
.TF
.PZ
```

.SENS
.TF
.PZ

Performing Initialization and Analysis

The first task Star-Hspice performs for .OP, .DC sweep, .AC, and .TRAN analyses is to set the DC operating point values for all nodes and sources. It does this either by calculating all of the values or by applying values specified in .NODESET and .IC statements or stored in an initial conditions file. The .OPTIONS OFF statement and the element parameters OFF and IC = val also control initialization.

Initialization is fundamental to the operation of simulation. Star-Hspice starts any analysis with known nodal voltages or initial estimates for unknown voltages and some branch currents, and then iteratively finds the exact solution. Initial estimates close to the exact solution increase the likelihood of a convergent solution and a lower simulation time.

A transient analysis first calculates a DC operating point using the DC equivalent model of the circuit (unless the UIC parameter is specified in the .TRAN statement). The resulting DC operating point is then used as an initial estimate to solve the next timepoint in the transient analysis.

If you do not provide an initial guess, or provide only partial information, Star-Hspice provides a default estimate of each of the nodes in the circuit and then uses this estimate to iteratively find the exact solution. The .NODESET and .IC statements are two methods that supply an initial guess for the exact DC solution of nodes within a circuit. Set any circuit node to any value by using the .NODESET statement. Star-Hspice then connects a voltage source equivalent to each initialized node (a current source with a parallel conductance GMAX set with a .OPTION statement). Next, a DC operating point is calculated with the .NODESET voltage source equivalent connected. Then Star-Hspice disconnects the equivalent voltage sources set with the .NODESET statement and recalculates the DC operating point. This is considered the DC operating point solution.
Use the .IC statement to provide both an initial guess and final solution to selected nodes within the circuit. Nodes initialized with the .IC statement become part of the solution of the DC operating point.

You can also use the OFF option to initialize active devices. The OFF option works in conjunction with .IC and .NODESET voltages as follows:

1. If any .IC or .NODESET statements exist, node voltages are set according to those statements.

2. If the OFF option is set, the terminal voltages of all active devices (BJTs, diodes, MOSFETs, JFETs, MESFETs) that are not set by .IC or .NODESET statements or by sources are set to zero.

3. If any IC parameters are specified in element statements, those initial conditions are set.

4. The resulting voltage settings are used as the initial guess at the operating point.

Use OFF to find an exact solution when performing an operating point analysis in a large circuit, where the majority of device terminals are at zero volts for the operating point solution. You can initialize the terminal voltages for selected active devices to zero by setting the OFF parameter in the element statements for those devices.

After a DC operating point has been found, use the .SAVE statement to store the operating point node voltages in a <design>.ic file. Then use the .LOAD statement to restore the operating point values from the ic file for subsequent analyses.
Setting Initial Conditions for Transient Analysis

If UIC is included in the .TRAN statement, a transient analysis is started using node voltages specified in an .IC statement.

Use the .OP statement to store an estimate of the DC operating point during a transient analysis.

An “internal timestep too small” error message indicates that the circuit failed to converge. The failure can be due to stated initial conditions that make it impossible to calculate the actual DC operating point.
Using DC Initialization and Operating Point Statements

.OP Statement — Operating Point

When an .OP statement is included in an input file, the DC operating point of the circuit is calculated. You can also use the .OP statement to produce an operating point during a transient analysis. Only one .OP statement can appear in a Star-Hspice simulation.

If an analysis is being used which requires an operating point to be calculated, the .OP statement is not required; an operating point calculation will be performed. If a .OP statement is specified and the keyword UIC exists in a .TRAN analysis statement, the time = 0 operating point analysis will be omitted and a warning issued in the output listing.

Syntax

.OP <format> <time> <format> <time>

format Any of the following keywords. (Only the first letter is required. Default = ALL.)

- ALL
  Full operating point, including voltage, currents, conductances, and capacitances. This parameter causes voltage/current output for time specified.

- BRIEF
  Produces a one line summary of each element’s voltage, current, and power. Current is stated in milliamperes and power in milliwatts.

- CURRENT
  Voltage table with element currents and power, a brief summary
Example

The following example calculates operating point voltages and currents for the DC solution, as well as currents at 10 ns, and voltages at 17.5 ns, 20 ns and 25 ns for the transient analysis.

```plaintext
.OP .5NS CUR 10NS VOL 17.5NS 20NS 25NS
```

The following example calculates the complete DC operating point solution. A printout of the solution is shown below.

```plaintext
.OP
```
Using DC Initialization and Operating Point Statements

Output

***** OPERATING POINT INFORMATION  TNOM = 25.000
TEMP = 25.000
***** OPERATING POINT STATUS IS ALL  SIMULATION TIME IS 0.

<table>
<thead>
<tr>
<th>NODE</th>
<th>VOLTAGE</th>
<th>NODE</th>
<th>VOLTAGE</th>
<th>NODE</th>
<th>VOLTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ 0:2</td>
<td>0.0000</td>
<td>0:3</td>
<td>437.3258M</td>
<td>0:4</td>
<td>455.1343M</td>
</tr>
<tr>
<td>+ 0:5</td>
<td>478.6763M</td>
<td>0:6</td>
<td>496.4858M</td>
<td>0:7</td>
<td>537.8452M</td>
</tr>
<tr>
<td>+ 0:8</td>
<td>555.6659M</td>
<td>0:10</td>
<td>5.0000</td>
<td>0:11</td>
<td>234.3306M</td>
</tr>
</tbody>
</table>

**** VOLTAGE SOURCES

SUBCKT

ELEMENT 0:VNCE 0:VN7 0:VPCE 0:VP7
VOLTS 0.0000 5.00000 0.0000 -5.00000
AMPS -2.07407U -405.41294P 2.07407U 405.41294P
POWER 0.00000 0.0000 0.02706N 0.02706N
TOTAL VOLTAGE SOURCE POWER DISSIPATION = 4.0541 N WATTS

**** BIPOLAR JUNCTION TRANSISTORS

SUBCKT

ELEMENT 0:QN1 0:QN2 0:QN3 0:QN4
MODEL 0:N1 0:N1 0:N1 0:N1
IB 999.99912N 2.00000U 5.00000U 10.00000U
IC -987.65345N -1.97530U -4.93827U -9.87654U
VBE 437.32588M 455.13437M 478.67632M 496.48580M
VCE 437.32588M 17.80849M 23.54195M 17.80948M
VBC 437.32588M 455.13437M 478.67632M 496.48580M
VS 0.00000 0.0000 0.0000 0.0000
POWER 5.39908N 875.09107N 2.27712U 4.78896U
BETAD -987.65432M -987.65432M -987.65432M -987.65432M
GM 0.00000 0.0000 0.0000 0.0000
RPI 2.0810E+06 1.0405E+06 416.20796K 208.10396K
RX 250.00000M 250.00000M 250.00000M 250.00000M
RO 2.0810E+06 1.0405E+06 416.20796K 208.10396K
CPI 1.43092N 1.44033N 1.45279N 1.46225N
CMU 954.16927P 960.66843P 969.64689P 977.06866P
CBX 0.00000 0.0000 0.0000 0.0000
CCS 800.00000P 800.00000P 800.00000P 800.00000P
BETAAC 0.00000 0.0000 0.0000 0.0000
FT 0.00000 0.0000 0.0000 0.0000
Element Statement IC Parameter

Use the element statement parameter, IC = \(<val>\), to set DC terminal voltages for selected active devices. The value set by IC = \(<val>\) is used as the DC operating point value, as in the DC solution.

Example

HXCC 13 20 VIN1 VIN2 IC = 0.5, 1.3

The example above describes an H element dependent voltage source with the current through VIN1 initialized to 0.5 mA and the current through VIN2 initialized to 1.3 mA.

.IC and .DCVOLT Initial Condition Statements

The .IC statement or the .DCVOLT statement is used to set transient initial conditions. How it initializes depends upon whether the UIC parameter is included in the .TRAN analysis statement.

When the UIC parameter is specified in the .TRAN statement, Star-Hspice does not calculate the initial DC operating point. In this case, the transient analysis is entered directly. The transient analysis uses the .IC initialization values as part of the solution for timepoint zero (a fixed equivalent voltage source is applied during the calculation of the timepoint zero). The .IC statement is equivalent to specifying the IC parameter on each element statement, but is more convenient. You can still specify the IC parameter, but it does not take precedence over values set in the .IC statement.

When the UIC parameter is not specified in the .TRAN statement, the DC operating point solution is computed before the transient analysis. In this case, the node voltages specified in the .IC statement are fixed for the determination of the DC operating point. For the transient analysis, the initialized nodes are released for the calculation of the second timepoint and later.
DC Initialization and Operating Point Statements

Syntax

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

or

```
.DCVOLT V(node1) = val1 V(node2) = val2 ...
```

where:

- **val1 ...** Specifies voltages. The significance of these specified voltages depends on whether the UIC parameter is specified in the .TRAN statement.
- **node1 ...** Node numbers or node names can include full path names or circuit numbers.

Example

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

**.NODESET Statement**

.NODESET initializes specified nodal voltages for a DC operating point analysis. The .NODESET statement often is used to correct convergence problems in DC analysis. Setting the nodes in the circuit to values that are close to the actual DC operating point solution enhances the convergence of the simulation. The simulator uses the NODESET voltages for the first iteration only.

Syntax

```
.NODESET V(node1) = val1 <V(node2) = val2 ...>
```

or

```
.NODESET node1 val1 <node2 val2>
```

- **node1 ...** Node numbers or node names can include full path names or circuit numbers.
Example

```
.NODESET V(5:SETX) = 3.5V V(X1.X2.VINT) = 1V
.NODESET V(12) = 4.5 V(4) = 2.23
.NODESET 12 4.5 4 2.23 1 1
```

Using .SAVE and .LOAD Statements

Star-Hspice always saves the operating point unless the .SAVE LEVEL = NONE statement is used. The saved operating-point file is restored only if the Star-Hspice input file contains a .LOAD statement.

Any node initialization commands, such as .NODESET and .IC, overwrite the initialization done through a .LOAD command if they appear in the netlist after the .LOAD command. This feature helps you to set particular states for multistate circuits such as flip-flops and still take advantage of the .SAVE command to speed up the DC convergence.

.SAVE and .LOAD continues to work even on changed circuit topologies. Adding or deleting nodes results in a new circuit topology. The new nodes are initialized as if no operating point were saved. References to deleted nodes are ignored. The coincidental nodes are initialized to the values saved from the previous run.

When nodes are initialized to voltages, Star-Hspice inserts Norton equivalent circuits at each initialized node. The conductance value of a Norton equivalent circuit is GMAX = 100. This conductance value might be too large for some circuits.

If using .SAVE and .LOAD does not speed up the simulation or causes problems with the simulation, you can use .OPTION GMAX = 1e-12 to minimize the effect of the Norton equivalent circuits on matrix conductances. Star-Hspice still uses the initialized node voltages for device initialization.
.SAVE Statement

The .SAVE statement stores the operating point of a circuit in a user-specified file. Then you can use the .LOAD statement to input the contents of this file for subsequent simulations to obtain quick DC convergence. The operating point is always saved by default, even if the Star-Hspice input file does not contain a .SAVE statement. To not save the operating point, specify .SAVE LEVEL = NONE.

You can specify that the operating point data be saved as an .IC statement or a .NODESET statement.

Syntax:

```
.SAVE <TYPE = type_keyword> <FILE = save_file> + <LEVEL = level_keyword> <TIME = save_time>
```

where:

- **type_keyword**
  Type of operating point storage desired. The type can be one of the following. Default: NODESET.
  - .NODESET
    Stores the operating point as a .NODESET statement. In subsequent simulations, all node voltages are initialized to these values if the .LOAD statement is used. Assuming incremental changes in circuit conditions, DC convergence should be achieved in a few iterations.
  - .IC
    Causes the operating point to be stored as a .IC statement. In subsequent simulations, node voltages are initialized to these values if .LOAD is included in the netlist file.

- **save_file**
  Name of the file in which the DC operating point data is stored. The default is `<design>.ic`. 
DC Initialization and Operating Point Analysis

For a parameter or temperature sweep, only the first operating point is saved. For example, if the Star-Hspice input netlist file contains the statement

```
.TEMP -25 0 25
```

the operating point corresponding to .TEMP -25 is saved.

**.LOAD Statement**

Use the .LOAD statement to input the contents of a file stored with the .SAVE statement. Files stored with the .SAVE statement contain operating point information for the point in the analysis at which the .SAVE was executed.

Do not use the .LOAD command for concatenated netlist files.

**Syntax**

```
.LOAD <FILE = load_file>
```

`load_file` Name of the file in which an operating point for the circuit under simulation was saved using .SAVE. The default is `<design>.ic`, where `design` is the root name of the design.
The .DC statement is used in DC analysis to:
- Sweep any parameter value
- Sweep any source value
- Sweep temperature range
- Perform a DC Monte Carlo analysis (random sweep)
- Perform a DC circuit optimization
- Perform a DC model characterization

The format for the .DC statement depends on the application in which it is used, as shown in the following examples:

**Syntax**

**Sweep or parameterized sweep:**

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

or

```
.DC var1 START = <param_expr1> STOP = <param_expr2>
+ STEP = <param_expr3>
```

or

```
.DC var1 start1 stop1 incr1 <SWEEP var2 type np start2 stop2>
```

or

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

**Data driven sweep:**

```
.DC var1 type np start1 stop1 <SWEEP DATA = datanm>
```

or

```
.DC DATA = datanm<SWEEP var2 start2 stop2 incr2>
```

or

```
.DC DATA = datanm
```
Monte Carlo:

```
.DC var1 type np start1 stop1 <Sweep MONTE = val>
```

or

```
.DC MONTE = val
```

Optimization:

```
.DC DATA = datanm OPTIMIZE = opt_par_fun
   + RESULTS = measnames MODEL = optmod
```

or

```
.DC var1 start1 stop1 SWEEP OPTIMIZE = OPTxxx
   + RESULTS = measname MODEL = optmod
```

The .DC statement keywords and parameters have the following descriptions:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA = datanm</td>
<td>Datanm is the reference name of a .DATA statement.</td>
</tr>
<tr>
<td>incr1 …</td>
<td>Voltage, current, element, model parameters, or temperature increment values</td>
</tr>
<tr>
<td>MODEL</td>
<td>Specifies the optimization reference name used in the .MODEL OPT statement used in an optimization analysis</td>
</tr>
<tr>
<td>MONTE = val</td>
<td>Produces a number val of randomly generated values, which are used to select parameters from a distribution. The distribution can be Gaussian, Uniform, or Random Limit.</td>
</tr>
<tr>
<td>np</td>
<td>Number of points per decade or per octave, or just number of points depending on the preceding keyword.</td>
</tr>
<tr>
<td>OPTIMIZE</td>
<td>Specifies the parameter reference name used for optimization in the .PARAM statement</td>
</tr>
<tr>
<td>RESULTS</td>
<td>Specifies the measure name used for optimization in the .MEASURE statement</td>
</tr>
</tbody>
</table>
### .DC Statement—DC Sweeps

**Start**1 …

Starting voltage, current, element, model parameters, or temperature values

If type variation “POI” (list of points) is used, a list of parameter values, instead of “start stop” is specified.

**Stop**1 …

Final voltage, current, any element, model parameter, or temperature values

**Sweep**

Keyword to indicate a second sweep has different type of variation (DEC, OCT, LIN, POI, DATA statement, or MONTE = val)

**Temp**

Keyword to indicate a temperature sweep

**Type**

Can be any of the following keywords:

- DEC — decade variation
- OCT — octave variation
- LIN — linear variation
- POI — list of points

**Var**1 …

Name of an independent voltage or current source, any element or model parameter, or the keyword TEMP (indicating a temperature sweep). Star-Hspice supports source value sweep, referring to the source name (SPICE style). However, if parameter sweep, a .DATA statement, and temperature sweep are selected, a parameter name must be chosen for the source value and subsequently referred to in the .DC statement. The parameter name cannot start with V or I.
Example

The following example causes the value of the voltage source VIN to be swept from 0.25 volts to 5.0 volts in increments of 0.25 volts.

```
.DC VIN 0.25 5.0 0.25
```

The following example invokes a sweep of the drain-to-source voltage from 0 to 10 V in 0.5 V increments at VGS values of 0, 1, 2, 3, 4, and 5 V.

```
.DC VDS 0 10 0.5 VGS 0 5 1
```

The following example asks for a DC analysis of the circuit from -55°C to 125°C in 10°C increments.

```
.DC TEMP -55 125 10
```

As a result of the following script, a DC analysis is conducted at five temperatures: 0, 30, 50, 100° and 125°C.

```
.DC TEMP POI 5 0 30 50 100 125
```

In the following example, a DC analysis is performed on the circuit at each temperature value, which results from a linear temperature sweep from 25°C to 125°C (five points), sweeping a resistor value called xval from 1 k to 10 k in 0.5 k increments.

```
.DC xval 1k 10k .5k SWEEP TEMP LIN 5 25 125
```

The example below specifies a sweep of the value par1 from 1 k to 100 k by 10 points per decade.

```
.DC DATA = datanm SWEEP par1 DEC 10 1k 100k
```

The next example also requests a DC analysis at specified parameters in the .DATA statement referenced by the .DATA statement reference name datanm. Parameter par1 also is swept from 1k to 100k by 10 points per decade.

```
.DC par1 DEC 10 1k 100k SWEEP DATA = datanm
```

The final example invokes a DC sweep of the parameter par1 from 1k to 100k by 10 points per decade, using 30 randomly generated (Monte Carlo) values.

```
.DC par1 DEC 10 1k 100k SWEEP MONTE = 30
```
Schmitt Trigger Example

*file: bjtschmt.sp    bipolar schmitt trigger
options post = 2
vcc 6 0 dc 12
vin 1 0 dc 0 pw1(0,0 2.5u,12 5u,0)
cbl 2 4 .1pf
rc1 6 2 1k
rc2 6 5 1k
rb1 2 4 5.6k
rb2 4 0 4.7k
re 3 0 .47k
*
diode 0 1 dmod
q1 2 1 3 bmod 1 ic = 0,8
q2 5 4 3 bmod 1 ic = .5,0.2
*
.dc vin 0,12,.1
*
.model dmod d is = 1e-15 rs = 10
.model bmod npn is = 1e-15 bf = 80 tf = 1n
+ cjc = 2pf cje = 1pf rc = 50 rb = 100 vaf = 200
.plot  v(1) v(5)
.graph dc model = schmittplot input = v(1) output = v(5) 4.0 5.0
.model schmittplot plot xscal = 1 yscal = 1 xmin = .5u xmax = 1.2u
.end
Using Other DC Analysis Statements

Star-Hspice provides the following additional DC analysis statements. Each of these statements uses the DC equivalent model of the circuit for its analysis functions. For .PZ, capacitors and inductors are included in the equivalent circuit.

- **.PZ**: Performs pole/zero analysis (.OP specification is not required)
- **.SENS**: Obtains the DC small-signal sensitivities of specified output variables with respect to circuit parameters (.OP specification is not required)
- **.TF**: Calculates the DC small-signal value of a transfer function (the ratio of an output variable to an input source). An .OP specification is not required.

Star-Hspice also provides a set of DC control options and DC initialization statements that allow for the modeling of resistive parasitics and the initialization of nodes. These enhance the convergence properties and the accuracy of the simulation. This section describes how to perform DC-related small signal analysis.

**.SENS Statement — DC Sensitivity Analysis**

If a .SENS statement is included in the input file, Star-Hspice determines the DC small-signal sensitivities of each specified output variable relative to every circuit parameter. The sensitivity measurement is the partial derivative of each output variable with respect to the value of a given circuit element, taken at the operating point and normalized to the total change in output magnitude. Therefore, the sum of the sensitivities of all elements is 100%. Sensitivities are calculated for resistors, voltage sources, current sources, diodes, and BJTs.

You can only perform one .SENS analysis per simulation. If more than one .SENS statement is present, only the last one is run.
Using Other DC Analysis Statements

DC Initialization and Operating Point Analysis

Syntax

`.SENS ov1 <ov2 ...>`

`ov1 ov2 ...` Branch currents or nodal voltage for DC component sensitivity analysis

Example

`.SENS V(9) V(4,3) V(17) I(VCC)`

Note: The .SENS statement can generate very large amounts of output for large circuits.

.TF Statement — DC Small-Signal Transfer Function Analysis

The transfer function statement (.TF) defines the small-signal output and input for DC small-signal analysis. When the .TF statement is included, Star-Hspice computes the DC small-signal value of the transfer function (output/input), input resistance, and output resistance.

Syntax

`.TF ov srcnam`

where:

`ov` Small-signal output variable

`srcnam` Small-signal input source

Example

`.TF V(5,3) VIN`

`.TF I(VLOAD) VIN`

For the first example, Star-Hspice computes the ratio of V(5,3) to VIN, the small-signal input resistance at VIN, to the small-signal output resistance
measured across nodes 5 and 3. Only one .TF statement can be used per simulation. If more than one .TF statement is present, only the last is performed.

**.PZ Statement— Pole/Zero Analysis**

**Syntax**

```
.PZ ov srcnam
```

where:

- `ov` Output variable: a node voltage V(n), or branch current I(element)
- `srcnam` Input source: an independent voltage or current source name

**Example**

```
.PZ V(10) VIN
.PZ I(RL) ISORC
```

See “Performing Pole/Zero Analysis” on page 27-1 for complete information about pole/zero analysis.
Setting DC Initialization Control Options

The DC operating point analysis control options control the DC convergence properties, as well as simulation algorithms. Many of these options also affect transient analysis because DC convergence is an integral part of transient convergence. The absolute and relative voltages, the current tolerances, and the matrix options should be considered for both DC and transient convergence.

Options are specified in .OPTIONS statements. The .OPTIONS statement is discussed in “Specifying Simulation Options” on page 9-1.

The following options are associated with controlling DC analysis. They are described in this section.

- ABSTOL
- CAPTAB
- CSHDC
- DCCAP
- DCFOR
- DCHOLD
- DCSTEP
- DV
- GRAMP
- GSHUNT
- ICSWEEP
- ITL1
- ITL2
- KCLTEST
- MAXAMP
- NEWTOL
- NOPIV
- OFF
- PIVOT
- PIVREF
- PIVREL
- PIVTOL
- RESMIN
- SPARSE

Some of these options also are used in DC and AC analysis. Many of these options also affect the transient analysis, because DC convergence is an integral part of transient convergence. Transient analysis is discussed in “Performing Transient Analysis” on page 11-1.

Option Descriptions

- **ABSTOL** = x
  - Sets the absolute node voltage error tolerance for DC and transient analysis. Decrease ABSTOL if accuracy is more important than convergence time.

- **CAPTAB**
  - Prints table of single plate nodal capacitance for diodes, BJTs, MOSFETs, JFETs and passive capacitors at each operating point.
**DC Initialization and Operating Point Analysis**

**Setting DC Initialization Control Options**

**CSHDC**

The same option as CSHUNT, but is used only with option CONVERGE.

**DCCAP**

Used to generate C-V plots and to print out the capacitance values of a circuit (both model and element) during a DC analysis. C-V plots are often generated using a DC sweep of the capacitor. Default = 0 (off).

**DCFOR** = \( x \)

Used in conjunction with the DCHOLD option and the .NODESET statement to enhance the DC convergence properties of a simulation. DCFOR sets the number of iterations that are to be calculated after a circuit converges in the steady state. Since the number of iterations after convergence is usually zero, DCFOR adds iterations (and computational time) to the calculation of the DC circuit solution. DCFOR helps ensure that a circuit has actually, not falsely, converged. Default = 0.

**DCHOLD** = \( x \)

DCFOR and DCHOLD are used together for the initialization process of a DC analysis. They enhance the convergence properties of a DC simulation. DCFOR and DCHOLD work together with the .NODESET statement. The DCHOLD option specifies the number of iterations a node is to be held at the voltage values specified by the .NODESET statement. The effects of DCHOLD on convergence differ according to the DCHOLD value and the number of iterations needed to obtain DC convergence.
If a circuit converges in the steady state in fewer than DCHOLD iterations, the DC solution includes the values set by the .NODESET statement. However, if the circuit requires more than DCHOLD iterations to converge, the values set in the .NODESET statement are ignored and the DC solution is calculated with the .NODESET fixed source voltages open circuited. Default = 1.

\[ DCSTEP = x \]

Used to convert DC model and element capacitors to a conductance to enhance DC convergence properties. The value of the element capacitors are all divided by DCSTEP to obtain a DC conductance model. Default = 0 (seconds).

\[ DV = x \]

The maximum iteration-to-iteration voltage change for all circuit nodes in both DC and transient analysis. Values of 0.5 to 5.0 can be necessary for some high-gain bipolar amplifiers to achieve a stable DC operating point. CMOS circuits frequently require a value of about 1 volt for large digital circuits. Default = 1000 (or 1e6 if DCON = 2).

\[ GRAMP = x \]

Value is set by Star-Hspice during the autoconvergence procedure. GRAMP is used in conjunction with the GMINDC convergence control option to find the smallest value of GMINDC that results in DC convergence. GMINDC is described in “Convergence Control Option Descriptions” on page 10-36.

GRAMP specifies the conductance range over which GMINDC is to be swept during a DC operating point analysis. Star-Hspice substitutes values of GMINDC over this range and simulates at each value. It then picks the lowest value of GMINDC that resulted in the circuit converging in the steady state.
If GMINDC is swept between 1e-12 mhos (the default) and 1e-6 mhos, GRAMP is set to 6 (the value of the exponent difference between the default and the maximum conductance limit). In this case, GMINDC is first set to 1e-6 mhos, and the circuit is simulated. If convergence is achieved, GMINDC is next set to 1e-7 mhos, and the circuit is simulated again. The sweep continues until a simulation has been performed at all values on the GRAMP ramp. If the combined conductance of GMINDC and GRAMP is greater than 1e-3 mho, a false convergence can occur. Default = 0.

**GSHUNT**

Conductance added from each node to ground. The default value is zero. Add a small GSHUNT to each node to possibly solve “Timestep too small” problems caused by high frequency oscillations or by numerical noise.

**ICSWEEP**

For a parameter or temperature sweep, saves the results of the current analysis for use as the starting point in the next analysis in the sweep. When ICSWEEP = 1, the current results are used in the next analysis. When ICSWEEP = 0, the results of the current analysis are not used in the next analysis. Default = 1.

**ITL1 = x**

Sets the maximum DC iteration limit. Increasing this value is unlikely to improve convergence for small circuits. Values as high as 400 have resulted in convergence for certain large circuits with feedback, such as operational amplifiers and sense amplifiers. Something is usually wrong with a model if more than 100 iterations are required for convergence. Set .OPTION ACCT to obtain a listing of how many iterations are required for an operating point. Default = 200.
**ITL2 = val**

Sets the DC transfer curve iteration limit. Increasing the iteration limit can be effective in improving convergence only on very large circuits.

Default = 50.

**KCLTEST**

Activates the KCL test (Kirchhoff’s Current Law) function. This test results in a longer simulation time, especially for large circuits, but provides a very accurate check of the solution. Default = 0.

When set to 1, Star-Hspice sets the following options:

- RELMOS and ABSMOS options are set to 0 (off).
- ABSI is set to 1e-16 A.
- RELI is set to 1e-6.

To satisfy the KCL test, the following condition must be satisfied for each node:

\[ |\sum i_b| < RELI \cdot \Sigma |i_b| + ABSI \]

where the \( i_b \)s are the node currents.

**MAXAMP = x**

Sets the maximum current through voltage defined branches (voltage sources and inductors). If the current exceeds the MAXAMP value, an error is issued. Default = 0.0.

**NEWTOL**

Calculates one more iterations past convergence for every DC solution and timepoint circuit solution calculated. When NEWTOL is not set, once convergence is determined, the convergence routine is ended and the next program step begun.

Default = 0.

**NOPIV**

Prevents Star-Hspice from switching automatically to pivoting matrix factorization when a nodal conductance is less than PIVTOL. NOPIV inhibits pivoting. Also see PIVOT.
**OFF**

Initializes the terminal voltages of all active devices to zero if they are not initialized to other values. For example, if the drain and source nodes of a transistor are not both initialized using `.NODESET` or `.IC` statements or by connecting them to sources, then the OFF option initializes all of the nodes of the transistor to zero. The OFF option is checked before element IC parameters, so if an element IC parameter assignment exists for a particular node, the node is initialized to the element IC parameter value even if it was previously set to zero by the OFF option. (The element parameter OFF can be used to initialize the terminal voltages to zero for particular active devices).

The OFF option is used to help find exact DC operating point solutions for large circuits.

**PIVOT = x**

Provides different pivoting algorithm selections. These can be used effectively to reduce simulation time and achieve convergence in circuits that produce hard-to-solve matrix equations. The pivot algorithm is selected by setting PIVOT to one of the following values:

- **0** Original nonpivoting algorithm
- **1** Original pivoting algorithm
- **2** Pick largest pivot in row algorithm
- **3** Pick best in row algorithm
- **10** Fast nonpivoting algorithm, more memory required
- **11** Fast pivoting algorithm, more memory required than for PIVOT values less than 11
12 Pick largest pivot in row algorithm, more memory required than for PIVOT values less than 12

13 Fast best pivot: faster, more memory required than for PIVOT values less than 13

Default = 10.

The fastest algorithm is PIVOT = 13, which can improve simulation time by up to ten times on very large circuits. However, the PIVOT = 13 option requires substantially more memory for the simulation. Some circuits with large conductance ratios, such as switching regulator circuits, might need pivoting. If PIVTOL = 0, Star-Hspice automatically changes from nonpivoting to a row pivot strategy upon detection of any diagonal matrix entry less than PIVTOL. This strategy provides the time and memory advantages of nonpivoting inversion, while avoiding unstable simulations and incorrect results. Use .OPTION NOPIV to prevent pivoting from being used under any circumstances. For very large circuits, PIVOT = 10, 11, 12, or 13 can require excessive memory.

If Star-Hspice switches to pivoting during a simulation, the message “pivot change on the fly” is printed, followed by the node numbers causing the problem. Use .OPTION NODE to obtain a node-to-element cross reference.

SPARSE is the same as PIVOT.

**PIVREF**

Pivot reference. Used in PIVOT = 11, 12, 13 to limit the size of the matrix. Default = 1e+8.
**PIVREL = x**
Sets the maximum/minimum row/matrix ratio. Use only for PIVOT = 1. Large values for PIVREL can result in very long matrix pivot times. If the value is too small, however, no pivoting occurs. It is best to start with small values of PIVREL, using an adequate but not excessive value for convergence and accuracy. Default = 1E-20 (max = 1e-20, min = 1).

**PIVTOL = x**
Sets the absolute minimum value for which a matrix entry is accepted as a pivot. PIVTOL is used as the minimum conductance in the matrix when PIVOT = 0. Default = 1.0e-15.

---
**Note:** PIVTOL should always be less than GMIN or GMINDC. Values approaching 1 yield increased pivot.

**RESMIN = x**
Specifies the minimum resistance value for all resistors, including parasitic and inductive resistances. Default = 1e-5 (ohm). Range: 1e-15 to 10 ohm.
SPARSE = x

Provides different pivoting algorithm selections. These can be used effectively to reduce simulation time and achieve convergence in circuits that produce hard-to-solve matrix equations. The pivot algorithm is selected by setting PIVOT to one of the following values:

- 0 Original nonpivoting algorithm
- 1 Original pivoting algorithm
- 2 Pick largest pivot in row algorithm
- 3 Pick best in row algorithm
- 10 Fast nonpivoting algorithm, more memory required
- 11 Fast pivoting algorithm, more memory required than for PIVOT values less than 11
- 12 Pick largest pivot in row algorithm, more memory required than for PIVOT values less than 12
- 13 Fast best pivot: faster, more memory required than for PIVOT values less than 13

Default = 10.

The fastest algorithm is PIVOT = 13, which can improve simulation time by up to ten times on very large circuits. However, the PIVOT = 13 option requires substantially more memory for the simulation. Some circuits with large conductance ratios, such as switching regulator circuits, might need pivoting. If PIVTOT = 0, Star-Hspice automatically changes from nonpivoting to a row pivot strategy upon detection of any diagonal matrix entry less than PIVTOL. This strategy provides the time and memory advantages of nonpivoting inversion, while avoiding unstable simulations and incorrect results. Use .OPTION NOPIV to prevent pivoting from being used under any circumstances.
For very large circuits, PIVOT = 10, 11, 12, or 13 can require excessive memory.

If Star-Hspice switches to pivoting during a simulation, the message “pivot change on the fly” is printed, followed by the node numbers causing the problem. Use .OPTION NODE to obtain a node-to-element cross reference.

SPARSE is the same as PIVOT.

### Pole/Zero Analysis Options

Control options are set using the .OPTIONS statement. The following are the control options used in pole/zero analysis.

- **CSCAL**: Sets the capacitance scale. Capacitances are multiplied by CSCAL. Default = 1e+12.

- **FMAX**: Sets the limit for maximum pole and zero frequency value. Default = 1.0e+12 \cdot FSCAL.

- **FSCAL**: Sets the frequency scale. Frequency is multiplied by FSCAL. Default = 1e-9.

- **GSCAL**: Sets the conductance scale. Conductances are multiplied by GSCAL. Resistances are divided by GSCAL. Default = 1e+3.

- **ITLPZ**: Sets the pole/zero analysis iteration limit. Default = 100.
**LSCAL**

Sets inductance scale. Inductances are multiplied by LSCAL. Default = 1e+6.

The scale factors must satisfy the following relations:

\[ GSCA = CSCAL \cdot FSCAL \]

\[ GSCAL = \frac{1}{LSCAL} \cdot FSCAL \]

If scale factors are changed, it might be necessary to modify the initial Muller points (X0R, X0I), (X1R, X1I) and (X2R, X2I), even though Star-Hspice multiplies initial values by (1e-9/GSCAL).

**PZABS**

Sets absolute tolerances for poles and zeros. This option affects the low frequency poles or zeros. It is used as follows:

If \(|X_{real}| + |X_{imag}| < PZABS\),

then \(X_{real} = 0\) and \(X_{imag} = 0\).

It is also used for convergence tests. Default = 1e-2.

**PZTOL**

Sets the relative error tolerance for poles or zeros. Default = 1.0e-6.

**RITOL**

Sets the minimum ratio value for (real/imaginary) or (imaginary/real) parts of the poles or zeros. RITOL is used as follows:

If \(|X_{imag}| \leq RITOL \cdot |X_{real}|\), then \(X_{imag} = 0\)

If \(|X_{real}| \leq RITOL \cdot |X_{imag}|\), then \(X_{real} = 0\)

Default = 1.0e-6.
The three complex starting points in the Muller pole/zero analysis algorithm are:

\[(X0R, X0I), \quad (X1R, X1I), \quad (X2R, X2I)\]

- \(X0R = -1.23456e6, \quad X0I = 0.0\)
- \(X1R = -1.23456e5, \quad X1I = 0.0\)
- \(X2R = +0.23456e6, \quad X2I = 0.0\)

These initial points are multiplied by FSCAL.
Specifying Accuracy and Convergence

Convergence is defined as the ability to obtain a solution to a set of circuit equations within a given tolerance criteria. In numerical circuit simulation, the designer specifies a relative and absolute accuracy for the circuit solution and the simulator iteration algorithm attempts to converge onto a solution that is within these set tolerances.

Accuracy Tolerances

Star-Hspice uses accuracy tolerance specifications to help assure convergence by determining whether or not to exit the convergence loop. For each iteration of the convergence loop, Star-Hspice takes the value of the previously calculated solution and subtracts it from the present solution, then compares this result with the accuracy tolerances.

\[ |V_{n^k}^k - V_{n^{k-1}}^{k-1}| \leq \text{accuracy tolerance} \]

where

- \(V_{n^k}^k\) is the solution at timepoint \(n\) and iteration \(k\)
- \(V_{n^{k-1}}^{k-1}\) is the solution at timepoint \(n\) and iteration \(k - 1\)

Absolute and Relative Accuracy Tolerances

As shown in Table 10-1, Star-Hspice defaults to specific absolute and relative values. You can change these tolerance levels so that simulation time is not excessive and accuracy is not compromised. The options in the table are described in the following section.

<table>
<thead>
<tr>
<th>Type</th>
<th>Option</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodal Voltage Tolerances</td>
<td>ABSVDC</td>
<td>50 µv</td>
</tr>
<tr>
<td></td>
<td>RELVDC</td>
<td>0.001</td>
</tr>
</tbody>
</table>
Nodal voltages and element currents are compared to the values from the previous iteration. If the absolute value of the difference is less than ABSVDC or ABSI, the node or element is considered to be convergent. ABSV and ABSI set the floor value below which values are ignored. Values above the floor use the relative tolerances of RELVDC and RELI. If the iteration-to-iteration absolute difference is less than these tolerances, then it is considered to be convergent. ABSMOS and RELMOS are the tolerances for MOSFET drain currents.

The number of iterations required is directly affected by the value of the accuracy settings. If the accuracy tolerances are tight, a longer time is required to converge. If the accuracy setting is too loose, the resulting solution can be inaccurate and unstable.

Table 10-2 shows an example of the relationship between the RELVDC value and the number of iterations.

**Table 10-1: Absolute and Relative Accuracy Tolerances**

<table>
<thead>
<tr>
<th>Type</th>
<th>Option</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Element Tolerances</td>
<td>ABSI</td>
<td>1 nA</td>
</tr>
<tr>
<td></td>
<td>RELI</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>ABSMOS</td>
<td>1 uA</td>
</tr>
<tr>
<td></td>
<td>RELMOS</td>
<td>.05</td>
</tr>
</tbody>
</table>

**Table 10-2: RELV vs. Accuracy and Simulation Time for 2 Bit Adder**

<table>
<thead>
<tr>
<th>RELVDC</th>
<th>Iteration</th>
<th>Delay (ns)</th>
<th>Period (ns)</th>
<th>Fall time (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.001</td>
<td>540</td>
<td>31.746</td>
<td>14.336</td>
<td>1.2797</td>
</tr>
<tr>
<td>.005</td>
<td>434</td>
<td>31.202</td>
<td>14.366</td>
<td>1.2743</td>
</tr>
<tr>
<td>.01</td>
<td>426</td>
<td>31.202</td>
<td>14.366</td>
<td>1.2724</td>
</tr>
<tr>
<td>.02</td>
<td>413</td>
<td>31.202</td>
<td>14.365</td>
<td>1.3433</td>
</tr>
</tbody>
</table>
Accuracy Control Options

Star-Hspice is shipped with control option settings designed to maximize accuracy without significantly degrading performance. The options and their settings are discussed in “Controlling Simulation Speed and Accuracy” on page 11-25.

Convergence Control Option Descriptions

The options listed below are described in this section.

- \( ABSH = x \)
- \( DCON \)
- \( RELH \)
- \( ABI \)
- \( DCTRAN \)
- \( RELI \)
- \( ABMOS \)
- \( DI \)
- \( RELMOS \)
- \( ABSVDC \)
- \( GMAX \)
- \( RELV \)
- \( CONVERGE \)
- \( GMINDC \)
- \( RELVDC \)

\( ABSH = x \) sets the absolute current change through voltage defined branches (voltage sources and inductors). In conjunction with DI and RELH, ABSH is used to check for current convergence. Default = 0.0.

### Table 10-2: RELV vs. Accuracy and Simulation Time for 2 Bit Adder

<table>
<thead>
<tr>
<th>RELVDC</th>
<th>Iteration</th>
<th>Delay (ns)</th>
<th>Period (ns)</th>
<th>Fall time (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.05</td>
<td>386</td>
<td>31.203</td>
<td>14.365</td>
<td>1.3315</td>
</tr>
<tr>
<td>.1</td>
<td>365</td>
<td>31.203</td>
<td>14.363</td>
<td>1.3805</td>
</tr>
<tr>
<td>.2</td>
<td>354</td>
<td>31.203</td>
<td>14.363</td>
<td>1.3908</td>
</tr>
<tr>
<td>.3</td>
<td>354</td>
<td>31.203</td>
<td>14.363</td>
<td>1.3909</td>
</tr>
<tr>
<td>.4</td>
<td>341</td>
<td>31.202</td>
<td>14.363</td>
<td>1.3916</td>
</tr>
<tr>
<td>.4</td>
<td>344</td>
<td>31.202</td>
<td>14.362</td>
<td>1.3904</td>
</tr>
</tbody>
</table>
ABSI = x

Sets the absolute branch current error tolerance in diodes, BJTs, and JFETs during DC and transient analysis. Decrease ABSI if accuracy is more important than convergence time.

If you want an analysis with currents less than 1 nanoamp, change ABSI to a value at least two orders of magnitude smaller than the minimum expected current.

Default: 1e-9 for KCLTEST = 0, 1e-16 for KCLTEST = 1

ABSMOS = x

Current error tolerance used for MOSFET devices in both DC and transient analysis. Star-Hspice uses the ABSMOS setting to determine if the drain-to-source current solution has converged. If the difference between the last and the present iteration’s drain-to-source current is less than ABSMOS, or if it is greater than ABSMOS, but the percent change is less than RELMOS, the drain-to-source current is considered converged. Star-Hspice then checks the other accuracy tolerances and, if all indicate convergence, the circuit solution at that timepoint is considered solved, and the next timepoint solution is calculated. For low power circuits, optimization, and single transistor simulations, set ABSMOS = 1e-12. Default = 1e-6 (amperes).
**ABSVDC** = $x$

Sets the absolute minimum voltage for DC and transient analysis. Decrease ABSVDC if accuracy is of more concern than convergence. If voltages less than 50 microvolts are required, ABSVDC can be reduced to two orders of magnitude less than the smallest desired voltage. This ensures at least two digits of significance. Typically ABSVDC need not be changed unless the circuit is a high voltage circuit. For 1000-volt circuits, a reasonable value can be 5 to 50 millivolts. Default = VNTOL (VNTOL default = 50 $\mu$V).

**CONVERGE**

Invokes different methods for solving nonconvergence problems

- **CONVERGE = -1**
  
  together with DCON = -1, disables autoconvergence

- **CONVERGE = 1**
  
  uses the Damped Pseudo Transient Algorithm. If the simulation fails to converge within the amount of CPU time set by the CPTIME control option, the simulation halts.

- **CONVERGE = 2**
  
  uses a combination of DCSTEP and GMINDC ramping

- **CONVERGE = 3**
  
  invokes the source stepping method

Even if it is not set in an .OPTIONS statement, the CONVERGE option is activated in the event of a matrix floating point overflow, or a timestep too small error. Default = 0.

In the event of a matrix floating point overflow, Star-Hspice sets CONVERGE = 1.
In the case of convergence problems, Star-Hspice automatically sets DCON = 1 and the following calculations are made:

\[ DV = \max\left(0.1, \frac{V_{\text{max}}}{50}\right), \text{ if } DV = 1000 \]

\[ GRAMP = \max\left(6, \log_{10}\left(\frac{I_{\text{max}}}{GM\text{INDC}}\right)\right) \]

\[ ITL1 = ITL1 + 20 \cdot GRAMP \]

where \( V_{\text{max}} \) is the maximum voltage and \( I_{\text{max}} \) is the maximum current.

If convergence problems still exist, Star-Hspice sets DCON = 2, which is the same as the above except \( DV = 1e6 \). The above calculations are used for DCON = 1 or 2. DCON = 1 is automatically invoked if the circuit fails to converge. DCON = 2 is invoked if DCON = 1 fails.

If the circuit contains uninitialized flip-flops or discontinuous models, the simulation might be unable to converge. Setting DCON = -1 and CONVERGE = -1 disables the autoconvergence algorithm and provides a list of nonconvergent nodes and devices.

**DCTRAN**

DCTRAN is an alias for CONVERGE. See CONVERGE.

**DI = x**

Sets the maximum iteration to iteration current change through voltage defined branches (voltage sources and inductors). This option is only applicable when the value of the ABSH control option is greater than 0. Default = 0.0.
**GMAX = x**
The conductance in parallel with the current source used for .IC and .NODESET initialization conditions circuitry. Some large bipolar circuits can require GMAX set to 1 for convergence. Default = 100 (mho).

**GMINDC = x**
A conductance that is placed in parallel with all pn junctions and all MOSFET nodes except gate (see Figure 6-4 for details) for DC analysis. GMINDC helps overcome DC convergence problems caused by low values of off conductance for pn junctions and MOSFET devices. GRAMP can be used to reduce GMINDC by one order of magnitude for each step. GMINDC can be set between 1e-4 and PIVTOL. Default = 1e-12.

Large values of GMINDC can cause unreasonable circuit response. If large values are required for convergence, a bad model or circuit is suspect. In the event of a matrix floating point overflow, if GMINDC is 1.0e-12 or less, Star-Hspice sets it to 1.0e-11.

GMINDC is manipulated by Star-Hspice in autoconverge mode, as described in the “Autoconverge Process” on page 10-42.

**RELH = x**
Sets relative current tolerance through voltage defined branches (voltage sources and inductors). It is used to check current convergence. This option is applicable only if the value of the ABSH control option is greater than zero. Default = 0.05.
**RELI = x**
Sets the relative error tolerance for current from iteration-to-iteration to determine convergence for all currents in diode, BJT, and JFET devices. (RELMOS sets the tolerance for MOSFETs). This is the change in current from the value calculated at the previous timepoint. Default = 0.01 for KCLTEST = 0, 1e-6 for KCLTEST = 1.

**RELMOS = x**
Sets the relative drain-to-source current error tolerance from iteration-to-iteration to determine convergence for currents in MOSFET devices. (RELI sets the tolerance for other active devices.) This is the change in current from the value calculated at the previous timepoint. RELMOS is only considered when the current is greater than the floor value, ABSPORT. Default = 0.05.

**RELV = x**
Sets the relative error tolerance for voltages. When voltages or currents exceed their absolute tolerances, the RELV test is used to determine convergence. Increasing RELV increases the relative error. In general, RELV should be left at its default value. RELV controls simulator charge conservation. For voltages, RELV is the same as RELTOL. Default = 1e-3.

**RELVDC = x**
Sets the relative error tolerance for voltages. When voltages or currents exceed their absolute tolerances, the RELVDC test is used to determine convergence. Increasing RELVDC increases the relative error. In general, RELVDC should be left at its default value. RELVDC controls simulator charge conservation. Default = RELTOL (RELTOL default = 1e-3).
Autoconverge Process

If convergence is not achieved in the number of iterations set by ITL1, Star-Hspice initiates an autoconvergence process, in which it manipulates DCON, GRAMP, and GMINDC, as well as CONVERGE in some cases. The autoconverge process is illustrated in Figure 10-3.

Notes:

1. Setting .OPTIONS DCON = -1 disables steps 2 and 3.
3. Setting .OPTIONS DCON = -1 CONVERGE = -1 disables steps 2, 3, and 4.
4. If you set the DV option to a value different from the default value, the value you set for DV is used in step 2, but DV is changed to 1e6 in step 3.
5. Setting GRAMP in an .OPTIONS statement has no effect on the autoconverge process. The autoconverge process sets GRAMP independently.
6. If you specify a value for GMINDC in an .OPTIONS statement, GMINDC is ramped to the value you set instead of to 1e-12 in steps 2 and 3.

DCON and GMINDC

GMINDC is important in stabilizing the circuit during DC operating point analysis. For MOSFETs, GMINDC helps stabilize the device in the vicinity of the threshold region. GMINDC is inserted between drain and bulk, source and bulk, and drain and source. The drain to source GMINDC helps linearize the transition from cutoff to weakly on, helps smooth out model discontinuities, and compensates for the effects of negative conductances.

The pn junction insertion of GMINDC in junction diodes linearizes the low conductance region so that the device behaves like a resistor in the low conductance region. This prevents the occurrence of zero conductance and improves the convergence of the circuit.
Figure 10-3: Autoconvergence Process Flow Diagram

**STEP 1**
Iterates up to ITL1 limit

**STEP 2**
Sets DCON = 1
If DV = 1000, sets DV from 1000 to max(0.1, Vmax/50)
Sets GRAMP = (Imax/GMINDC)
Ramps GMINDC from GMINDC ⋅ 10^{GRAMP} to 1e-12

**STEP 3**
Sets DCON = 2
Relaxes DV to 1e6
Sets GRAMP = (Imax/GMINDC)
Ramps GMINDC from GMINDC ⋅ 10^{GRAMP} to 1e-12

**STEP 4**
Adds CSHDC and GSHUNT from each node to ground
Ramps supplies from zero to set values
Removes CSHDC and GSHUNT after DC convergence and iterates further to a stable DC bias point

Nonconvergence report
DCON is an option that Star-Hspice sets automatically in case of nonconvergence. It invokes the GMINDC ramping process in steps 2 and 3 in Figure 10-3. GMINDC is shown for various elements in Figure 10-4.

**Figure 10-4: GMINDC Insertion**

- **diode element**
- **BJT element**
- **MOSFET element**
- **JFET or MESFET element**
Reducing DC Errors

You can reduce DC errors by performing the following steps.

1. Check topology, set .OPTION NODE to get a nodal cross reference listing if you are in doubt.
   
   Are all MOS p-channel substrates connected to VCC or positive supplies?
   Are all MOS n-channel substrates connected to GND or negative supplies?
   Are all vertical NPN substrates connected to GND or negative supplies?
   Are all lateral PNP substrates connected to negative supplies?
   Do all latches have either an OFF transistor or a .NODESET or an .IC on one side?
   Do all series capacitors have a parallel resistance, or is .OPTION DCSTEP set?

2. Check your .MODEL statements.
   
   Be sure to check your model parameter units. Use model printouts to verify actual values and units, since some model parameters are multiplied by scaling options.
   
   Do MOS models have subthreshold parameters set with reasonable value (such as NFS = 1e11 for SPICE models 1, 2, and 3 and N0 = 1.0 for Star-Hspice models BSIM1, BSIM2, and Level 28)?
   
   Avoid setting UTRA in MOS Level 2 models.
   
   Are JS and JSW set in MOS model for DC portion of diode model? A typical JS value is 1e-4A/M^2.
   
   Are CJ and CJSW set in MOS diode model?
   
   Do JFET and MESFET models have weak inversion NG and ND set?
   
   If MOS Level 6 LGAMMA equation is used, is UPDATE = 1?
   
   DIODE models should have nonzero values for saturation current, junction capacitance, and series resistance.
   
   Use MOS ACM = 1, ACM = 2, or ACM = 3 source and drain diode calculations to automatically generate parasitics.
3. General remarks:

   Ideal current sources require large values of .OPTION GRAMP, especially for BJT and MESFET circuits because they do not ramp up with the supply voltages and can force reverse bias conditions, leading to excessive nodal voltages.

   Schmitt triggers are unpredictable for DC sweep and sometimes for operating points for the same reasons oscillators and flip-flops are. Use slow transient.

   Large circuits tend to have more convergence problems because they have a higher probability of uncovering a modeling problem.

   Circuits that converge individually and fail when combined are almost guaranteed to have a modeling problem.

   Open loop op-amps have high gain, which can lead to difficulties in converging. Start op-amps in unity gain configuration and open them up in transient analysis with a voltage-variable resistor or a resistor with a large AC value for AC analysis.

4. Check your options:

   Remove all convergence-related options and try first with no special options settings.

   Check nonconvergence diagnostic tables for nonconvergent nodes. Look up nonconvergent nodes in the circuit schematic. They are generally latches, Schmitt triggers, or oscillating nodes.

   For stubborn convergence failures, bypass DC altogether with .TRAN with UIC set. Continue transient analysis until transients settle out, then specify .OP time to obtain an operating point during the transient analysis. An AC analysis also can be specified during the transient analysis by adding an .AC statement to the .OP time statement.

   SCALE and SCALM scaling options have a significant effect on the element and model parameter values. Be careful with units.
**Shorted Element Nodes**

Star-Hspice disregards any capacitor, resistor, inductor, diode, BJT, or MOSFET that has all its leads connected together. The component is not counted in the component tally Star-Hspice produces. Star-Hspice issues the following warning:

** warning ** all nodes of element x:<name> are connected together

**Conductance Insertion Using DCSTEP**

In a DC operating point analysis, failure to include conductances in a capacitor model results in broken circuit loops (since a DC analysis opens all capacitors), which might not be solvable. By including a small conductance in the capacitor model, the circuit loops are complete and can be solved.

Modeling capacitors as complete opens often results in the following error message:

“No DC Path to Ground”

For a DC analysis, .OPTION DCSTEP is used to give a conductance value to all capacitors in the circuit. DCSTEP calculates the value as follows:

\[
\text{conductance} = \frac{\text{capacitance}}{\text{DCSTEP}}
\]

Figure 10-5 illustrates how Star-Hspice inserts conductance G in parallel with capacitance Cg to provide current paths around capacitances in DC analysis.
Floating Point Overflow

Negative or zero MOS conductance sometimes results in Star-Hspice having difficulty converging. An indication of this type of problem is a floating point overflow during matrix solutions. Star-Hspice detects floating point overflow and invokes the Damped Pseudo Transient algorithm (CONVERGE = 1) to try to achieve DC convergence without requiring user intervention. If GMINDC is 1.0e-12 or less when a floating point overflow occurs, Star-Hspice sets it to 1.0e-11.
**Diagnosing Convergence Problems**

Before simulation, Star-Hspice diagnoses potential convergence problems in the input circuit, and provides an early warning to help debugging. When a circuit condition that indicates possible convergence problems is detected, Star-Hspice prints the following message into the output file:

"Warning: Zero diagonal value detected at node ( ) in equation solver, which might cause convergence problems. If your simulation fails, try adding a large resistor between node ( ) and ground."

**Nonconvergence Diagnostic Table**

Two automatic printouts are generated when nonconvergence is encountered: the nodal voltage printout and the element printout (the diagnostic tables). The nodal voltage printout prints all nonconvergent node voltage names and the associated voltage error tolerances (tol). The element printout lists all nonconvergent elements, along with their associated element currents, element voltages, model parameters, and current error tolerances (tol).

To locate the branch current or nodal voltage resulting in nonconvergence, analyze the diagnostic tables for unusually large values of branch currents, nodal voltages or tolerances. Once located, initialize the node or branch using the .NODESET or .IC statements. The nonconvergence diagnostic table is automatically generated when a circuit simulation has not converged, indicating the quantity of recorded voltage failures and the quantity of recorded branch element failures. A voltage failure can be generated by any node in the circuit, including “hidden” nodes, such as the extra nodes created by parasitic resistors.

The element printout lists the subcircuit, model name, and element name of all parts of the circuit having nonconvergent nodal voltages or currents. Table 10-3 identifies the inverters, xinv21, xinv22, xinv23, and xinv24 as problem subcircuits of a ring oscillator. It also indicates that the p-channel transistor of subcircuits xinv21, xinv22, xinv24 are nonconvergent elements. The n-channel transistor of xinv23 is also a nonconvergent element. The table gives the voltages and currents of the transistors, so the designer can quickly check to see
if they are of a reasonable value. The tolds, tolbd, and tolbs error tolerances indicate how close the element currents (drain to source, bulk to drain, and bulk to source) were to a convergent solution. For tol variables, a value close to or below 1.0 indicates a convergent solution. As shown in Table 10-3, the tol values in the order of 100 indicate the currents were far from convergence. The element current and voltage values are also given (id, ibs, ibd, vgs, vds, and vbs). These values can be examined for realistic values and determination of the transistor regions of operation.

### Table 10-3: Voltages, Currents, and Tolerances for Subcircuits

<table>
<thead>
<tr>
<th>subckt</th>
<th>element</th>
<th>model</th>
<th>subckt</th>
<th>element</th>
<th>model</th>
<th>subckt</th>
<th>element</th>
<th>model</th>
<th>subckt</th>
<th>element</th>
<th>model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>xinv21</td>
<td>21:mphc1 0:p1</td>
<td></td>
<td>xinv22</td>
<td>22:mphc1 0:p1</td>
<td></td>
<td>xinv23</td>
<td>23:mphc1 0:p1</td>
<td></td>
<td>xinv23</td>
<td>23:mnch1 0:n1</td>
</tr>
<tr>
<td>id</td>
<td>27.5809f</td>
<td>140.5646u</td>
<td>1.8123p</td>
<td>1.7017m</td>
<td>5.5132u</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ibs</td>
<td>205.9804f</td>
<td>3.1881f</td>
<td>31.2989f</td>
<td>0.</td>
<td>200.0000f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ibd</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>-168.7011f</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vgs</td>
<td>4.9994</td>
<td>-4.9992</td>
<td>69.9223</td>
<td>4.9998</td>
<td>-67.8955</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vds</td>
<td>4.9994</td>
<td>206.6633u</td>
<td>69.9225</td>
<td>-64.9225</td>
<td>2.0269</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vbs</td>
<td>4.9994</td>
<td>206.6633u</td>
<td>69.9225</td>
<td>0.</td>
<td>2.0269</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vth</td>
<td>-653.8030m</td>
<td>-745.5860m</td>
<td>-732.8632m</td>
<td>549.4114m</td>
<td>-656.5097m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tolds</td>
<td>114.8609</td>
<td>82.5624</td>
<td>155.9508</td>
<td>104.5004</td>
<td>5.3653</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tolbd</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tolbs</td>
<td>3.534e-19</td>
<td>107.1528m</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Traceback of Nonconvergence Source

To locate a nonconvergence source, trace the circuit path for error tolerance. In an inverter chain, for example, the last inverter can have a very high error tolerance. If this is the case, the error tolerance of the elements driving the inverter should be examined. If the driving tolerance is high, the driving element could be the source of nonconvergence. However, if the tolerance is low, the driven element should be checked as the source of nonconvergence.

By examining the voltages and current levels of a nonconvergent MOSFET, you can discover the operating region of the MOSFET. This information can flow to the location of the discontinuity in the model, for example, subthreshold-to-linear or linear-to-saturation.

When considering error tolerances, check the current and nodal voltage values. If the current or nodal voltage values are extremely low, nonconvergence errors can be induced because a relatively large number is being divided by a very small number. This results in nonconvergence because the calculation produces a large result. A solution is to increase the value of the absolute accuracy options.

Use the diagnostic table in conjunction with the DC iteration limit (ITL1 statement) to find the sources of nonconvergence. By increasing or decreasing ITL1, output for the problem nodes and elements for a new iteration is printed—that is, the last iteration of the analysis set by ITL1.

Solutions for Nonconvergent Circuits

Nonconvergent circuits generally result from:

- Poor Initial Conditions
- Inappropriate Model Parameters
- PN Junctions (Diodes, MOSFETs, BJTs)

These conditions are discussed in the following sections.
Poor Initial Conditions

Multistable circuits need state information to guide the DC solution. You must initialize ring oscillators and flip-flops. These multistable circuits either give the intermediate forbidden state or cause a DC convergence problem. Initialize a circuit using the .IC statement to force a node to the requested voltage. Ring oscillators usually need only one stage set.

**Figure 10-6: Ring Oscillator**

It is best to set up the flip-flop with an .IC statement inside the subcircuit definition. In the following example, a local parameter “Qset” is set to 0. It is used as the value for the .IC statement to initialize the latch output node “Q”. This results in all latches having a default state of “Q” low. This state is overridden by the call to a latch by setting “Qset” to vdd.

**Example**

```
.subckt latch in Q Q/ d Qset = 0
.ic Q = Qset
...
.ends
```
Inappropriate Model Parameters

It is possible to create a discontinuous IDS or capacitance model by imposing nonphysical model parameters. This can cause an “internal timestep too small” error during the transient simulation. The demonstration file mosivcv.sp shows IDS, VGS, GM, GDS, GMB, and CV plots for MOS devices. A sweep near threshold from Vth-0.5 V to Vth+0.5 V using a delta of 0.01 V sometimes discloses a possible discontinuity in the curves.

Figure 10-7: Discontinuous I-V Characteristics

If the simulation no longer converges when a component is added or a component value is changed, the model parameters are inappropriate or do not correspond to the physical values they represent. Check the Star-Hspice input netlist file for nonconvergent elements. Devices with a “TOL” greater than 1 are nonconvergent. Find the devices at the beginning of the combined logic string of gates that seem to start the nonconvergent string. Check the operating point of these devices very closely to see what region they operate in. The model parameters associated with this region are most likely inappropriate.
Circuit simulation is based on using single-transistor characterization to simulate a large collection of devices. If a circuit fails to converge, it can be caused by a single transistor somewhere in the circuit.

**PN Junctions (Diodes, MOSFETs, BJT)**

PN junctions found in diode, BJT, and MOSFET models can exhibit nonconvergent behavior in both DC and transient analysis. For example, PN junctions often have a high off resistance, resulting in an ill-conditioned matrix. To overcome this, the options GMINDC and GMIN automatically parallel every PN junction in a design with a conductance. Nonconvergence can occur by overdriving the PN junction. This happens when a current-limiting resistor is omitted or has a very small value. In transient analysis, protection diodes often are temporarily forward biased (due to the inductive switching effect), overdriving the diode and resulting in nonconvergence if a current-limiting resistor is omitted.
Chapter 11

Performing Transient Analysis

Star-Hspice transient analysis computes the circuit solution as a function of time over a time range specified in the .TRAN statement.

This chapter covers the following topics:

- Understanding the Simulation Flow
- Understanding Transient Analysis
- Using the .TRAN Statement
- Using the .BIASCHK Statement
- Understanding the Control Options
- Controlling Simulation Speed and Accuracy
- Numerical Integration Algorithm Controls
- Selecting Timestep Control Algorithms
- Performing Fourier Analysis
Understanding the Simulation Flow

Figure 11-1 illustrates the transient analysis simulation flow for Star-Hspice.

Figure 11-1: Transient Analysis Simulation Flow

Options:

Method

BYPASS
CSHUNT
DVDT
GSHUNT
LVLTIM = x
MAXORD = x
METHOD

Tolerance

ABSV = x
ABSVAR = x
ACCURATE
BYTOL = x
CHGTOL = x
DELMAX = x
FAST
MBYPASS
MU
RELQ = x
RELVAR = x
SLOPETOL = x
TIMERES
TRTOL = x
VNTOL

Limit

AUTOSTOP
BKPSIZ
DVTR = x
FS = x
FT = x
GMIN = x
IMAX = x
IMIN = x
ITL3 = x
ITL4 = x
ITL5 = x
RMAX = x
RMIN = x
VFLOOR
Understanding Transient Analysis

Since transient analysis is dependent on time, it uses different analysis algorithms, control options with different convergence-related issues and different initialization parameters than DC analysis. However, since a transient analysis first performs a DC operating point analysis (unless the UIC option is specified in the .TRAN statement), most of the DC analysis algorithms, control options, and initialization and convergence issues apply to transient analysis.

Initial Conditions for Transient Analysis

Some circuits, such as oscillators or circuits with feedback, do not have stable operating point solutions. For these circuits, either the feedback loop must be broken so that a DC operating point can be calculated or the initial conditions must be provided in the simulation input. The DC operating point analysis is bypassed if the UIC parameter is included in the .TRAN statement. If UIC is included in the .TRAN statement, a transient analysis is started using node voltages specified in an .IC statement. If a node is set to 5 V in a .IC statement, the value at that node for the first time point (time 0) is 5 V.

You can use the .OP statement to store an estimate of the DC operating point during a transient analysis.

Example

```
.TRAN 1ns 100ns UIC
.OP 20ns
```

The .TRAN statement UIC parameter in the above example bypasses the initial DC operating point analysis. The .OP statement calculates transient operating point at t = 20 ns during the transient analysis.

Although a transient analysis might provide a convergent DC solution, the transient analysis itself can still fail to converge. In a transient analysis, the error message “internal timestep too small” indicates that the circuit failed to converge. The convergence failure might be due to stated initial conditions that are not close enough to the actual DC operating point values. See the later part of this chapter for a discussion of transient analysis convergence aids.
Using the .TRAN Statement

Syntax

**Single-point analysis:**

```
.TRAN var1 START = start1 STOP = stop1 STEP = incr1
```

or

```
.TRAN var1 START = <param_expr1> STOP = <param_expr2> + STEP = <param_expr3>
```

**Double-point analysis:**

```
.TRAN var1 START = start1 STOP = stop1 STEP = incr1 + <SWEEP var2 type np start2 stop2>
```

or

```
.TRAN tincr1 tstop1 <tincr2 tstop2 ...tincrN tstopN> + <START = val> <UIC> +  <SWEEP var pstart + pstop pincr>
```

**Parameterized sweep:**

```
.TRAN tincr1 tstop1 <tincr2 tstop2 ...tincrN tstopN> + <START = val> <UIC>
```

**Data driven sweep:**

```
.TRAN DATA = datanm
```

or

```
.TRAN var1 START = start1 STOP = stop1 STEP = incr1 + <SWEEP DATA = datanm>
```

or

```
.TRAN DATA = datanm<SWEEP var pstart pstop pincr>
```

**Monte Carlo:**

```
.TRAN tincr1 tstop1 <tincr2 tstop2 ...tincrN tstopN> + <START = val> <UIC><SWEEP MONTE = val>
```
Performing Transient Analysis

Using the .TRAN Statement

**Optimization:**
```
.TRAN DATA = datanm OPTIMIZE = opt_par_fun
+ RESULTS = measnames MODEL = optmod
```

Transient sweep specifications can include the following keywords and parameters:

<table>
<thead>
<tr>
<th><strong>DATA = datanm</strong></th>
<th>Data name referred to in the .TRAN statement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MONTE = val</strong></td>
<td>Produces a number val of randomly generated values that are used to select parameters from a distribution. The distribution can be <em>Gaussian</em>, <em>Uniform</em>, or <em>Random Limit.</em></td>
</tr>
<tr>
<td><strong>np</strong></td>
<td>Number of points or number of points per decade or octave, depending on the preceding keyword</td>
</tr>
<tr>
<td><strong>param_expr...</strong></td>
<td>User-specified expressions—for example, param_expr1...param_exprN</td>
</tr>
<tr>
<td><strong>pincr</strong></td>
<td>Voltage, current, element or model parameter, or temperature increment value</td>
</tr>
</tbody>
</table>

**Note:** If “type” variation is used, the “np” (number of points) is specified instead of “pincr”.

| **pstart**        | Starting voltage, current, temperature, any element or model parameter value |

**Note:** If type variation “POI” is used (list of points), a list of parameter values is specified instead of “pstart pstop”.

| **pstop**        | Final voltage, current, temperature, any element or model parameter value |
### .TRAN Statement

- **START**
  - Time at which printing or plotting is to begin. The START keyword is optional: you can specify start time without preceding it with “START = ”

  **Note:** If the .TRAN statement is used in conjunction with a .MEASURE statement, using a nonzero START time can result in incorrect .MEASURE results. Nonzero START times should not be used in .TRAN statements when .MEASURE also is being used.

- **SWEEP**
  - Keyword to indicate a second sweep is specified on the .TRAN statement

- **tincr1**…
  - Printing or plotting increment for printer output, and the suggested computing increment for the postprocessor.

- **tstop1**…
  - Time at which the transient analysis stops incrementing by tincr1. If another tincr-tstop pair follows, the analysis continues with the new increment.

- **type**
  - Specifies any of the following keywords:
    - DEC – decade variation
    - OCT – octave variation (the value of the designated variable is eight times its previous value)
    - LIN – linear variation
    - POI – list of points
Performing Transient Analysis

Using the .TRAN Statement

**Example**

1. The following example performs and prints the transient analysis every 1 ns for 100 ns.
   ```
   .TRAN 1NS 100NS
   ```

2. The following example performs the calculation every 0.1 ns for the first 25 ns, and then every 1 ns until 40 ns; the printing and plotting begin at 10 ns.
   ```
   .TRAN .1NS 25NS 1NS 40NS START = 10NS
   ```

3. The following example performs the calculation every 10 ns for 1 µs; the initial DC operating point calculation is bypassed, and the nodal voltages specified in the .IC statement (or by IC parameters in element statements) are used to calculate initial conditions.
   ```
   .TRAN 10NS 1US UIC
   ```

4. The following example increases the temperature by 10 °C through the range -55 °C to 75 °C and performs transient analysis for each temperature.
   ```
   .TRAN 10NS 1US UIC SWEEP TEMP -55 75 10
   ```

---

**UIC**

Causes Star-Hspice to use the nodal voltages specified in the .IC statement (or by the “IC = ” parameters in the various element statements) to calculate the initial transient conditions, rather than solving for the quiescent operating point.

**var**

Name of an independent voltage or current source, any element or model parameter, or the keyword TEMP (indicating a temperature sweep). Star-Hspice supports source value sweep, referring to the source name (SPICE style). However, if a parameter sweep, a .DATA statement, and a temperature sweep are specified, a parameter name must be chosen for the source value and subsequently referred to in the .TRAN statement. The parameter name must not start with V or I.
5. The following performs an analysis for each load parameter value at 1 pF, 5 pF, and 10 pF.
   `.TRAN 10NS 1US SWEEP load POI 3 1pf 5pf 10pf`

6. The following example is a data driven time sweep and allows a data file to be used as sweep input. If the parameters in the data statement are controlling sources, they must be referenced by a piecewise linear specification.
   `.TRAN data = dataname`
Using the .BIASCHK Statement

Breakdown will occur when the voltage bias between some terminals of an element is too large. The .BIASCHK statement monitors the voltage bias with user defined limits and noise. Bias monitoring can check the bias that you want to monitor during transient analysis and report the following:

■ element name
■ time
■ terminals
■ bias exceeding the limit
■ number of times the bias exceeds the limit for an appointed element

The information will be saved as a warning and biaschk summary in the *.lis file.

This command is for MOS only in Star-Hspice release 2001.2. BIASCHK cannot detect the bias exceeding the limit if the bias is always the same value during transient analysis.

Syntax

`.biaschk type terminal1=t1 terminal2=t2 limit=lim
+ <noise=ns><name=devname1><name=devname2>...
+ <mname=modelname1><mname=modelname2> ...

where:

type Element type the user wants to check
MOS (R, C . . .)
Type is for MOS only in 2001.2

terminal 1, 2 Terminals the user wants to check between:
For MOS level 57: nd, ng, ns, ne, np, n6
For MOS level 58: nd, ngf, ns, ngb
For MOS level 59: nd, ng, ns, ne, np
For other MOS level: nd, ng, ns, nb

limit Biaschk limit defined by user
Using the .BIASCHK Statement
Performing Transient Analysis

noise
Biaschk noise defined by user
Default = 0.1v

name
Element name the user wants to check

mname
Model name. Elements of this model will be checked for bias

Example
.biaschk MOS terminal1=ng terminal2=nb limit=2v
+ noise=0.01v name=x1.x3.ml mname=nch.1 name=m3
Understanding the Control Options

The options in this section modify the behavior of the transient analysis integration routines. Delta refers to the internal timestep. TSTEP and TSTOP refer to the step and stop values entered with the .TRAN statement. The options are grouped into three categories: method, tolerance, and limit:

<table>
<thead>
<tr>
<th>Method</th>
<th>Tolerance</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYPASS</td>
<td>ABSH</td>
<td>RELH</td>
</tr>
<tr>
<td>CSHUNT</td>
<td>ABSV</td>
<td>RELI</td>
</tr>
<tr>
<td>DVDT</td>
<td>ABSVAR</td>
<td>RELQ</td>
</tr>
<tr>
<td>GSHUNT</td>
<td>ACCURATE</td>
<td>RELTOL</td>
</tr>
<tr>
<td>INTERP</td>
<td>BYTOL</td>
<td>RELV</td>
</tr>
<tr>
<td>ITRPRT</td>
<td>CHGTOL</td>
<td>RELVAR</td>
</tr>
<tr>
<td>LVLTIM</td>
<td>DI</td>
<td>SLOPETOL</td>
</tr>
<tr>
<td>MAXORD</td>
<td>FAST</td>
<td>TIMERES</td>
</tr>
<tr>
<td>METHOD</td>
<td>MBYPASS</td>
<td>TRTOL</td>
</tr>
<tr>
<td>MAXAMP</td>
<td>VNTOL</td>
<td></td>
</tr>
<tr>
<td>MU</td>
<td>XMU</td>
<td></td>
</tr>
</tbody>
</table>

**Method Options**

**BYPASS**

Speeds up simulation by not updating the status of latent devices. Setting `.OPTION BYPASS = 1` enables bypassing. BYPASS applies to MOSFETs, MESFETs, JFETs, BJTs, and diodes. Default = 1.

*Note:* Use the BYPASS algorithm cautiously. For some types of circuits it can result in nonconvergence problems and loss of accuracy in transient analysis and operating point calculations.
### Understanding the Control Options

#### Performing Transient Analysis

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSHUNT</td>
<td>Capacitance added from each node to ground. Adding a small CSHUNT to each node can solve some “internal timestep too small” problems caused by high-frequency oscillations or numerical noise. Default = 0.</td>
</tr>
</tbody>
</table>
| DVDT | Allows the timestep to be adjusted based on node voltage rates of change. Choices are:  
  - 0 - original algorithm  
  - 1 - fast  
  - 2 - accurate  
  - 3,4 - balance speed and accuracy  
  
  Default = 4.  
  
  The ACCURATE option also increases the accuracy of the results. |
| GSHUNT | Conductance added from each node to ground. The default value is zero. Adding a small GSHUNT to each node can solve some “internal timestep too small” problems caused by high frequency oscillations or by numerical noise. |
| INTERP | Limits output to post-analysis tools, such as Cadence or Zuken, to only the .TRAN timestep intervals. By default, Star-Hspice outputs all convergent iterations in design.tr# file. INTERP typically produces a much smaller design.tr# file.  
  
  You should use INTERP = 1 with caution when the .MEASURE statement is present. Star-Hspice computes measure statements using the postprocessing output. Reducing postprocessing output may lead to interpolation errors in measure results. |
### Performing Transient Analysis

**Understanding the Control Options**

<table>
<thead>
<tr>
<th><strong>Note:</strong> When running a data driven transient analysis (.TRAN DATA statement) within optimization routines, INTERP is forced to 1. As a result, all measurement results are made at the time points of the data in the data driven sweep. If the measurement needs to use all converged internal timesteps, e.g. AVG or RMS calculations, you should set <code>ITRPRT = 1</code>.</th>
</tr>
</thead>
</table>

- **ITRPRT**
  - Prints output variables at their internal timepoint values. Using this option can generate a long output list.

- **LVLTIM = x**
  - Selects the timestep algorithm used for transient analysis. If `LVLTIM = 1`, the DVDT timestep algorithm is used. If `LVLTIM = 2`, the local truncation error timestep algorithm is used. If `LVLTIM = 3`, the DVDT timestep algorithm with timestep reversal is used.

  If the GEAR method of numerical integration and linearization is used, `LVLTIM = 2` is selected. If the TRAP linearization algorithm is used, `LVLTIM 1` or `3` can be selected. Using `LVLTIM = 1` (the DVDT option) helps avoid the “internal timestep too small” nonconvergence problem. The local truncation algorithm (LVLTIM = 2), however, provides a higher degree of accuracy and prevents errors propagating from time point to time point, which can sometimes result in an unstable solution. Default = 1.
**Understanding the Control Options**

### Performing Transient Analysis

**MAXORD = x**
- Sets the maximum order of integration when the GEAR method is used (see METHOD). The value of x can be either 1 or 2. If MAXORD = 1, the backward Euler method of integration is used. MAXORD = 2, however, is more stable, accurate, and practical. Default = 2.0.

**METHOD = name**
- Sets the numerical integration method used for a transient analysis to either GEAR or TRAP. To use GEAR, set METHOD = GEAR. This automatically sets LVLTIM = 2.

  (You can change LVLTIM from 2 to 1 or 3 by setting LVLTIM = 1 or 3 after the METHOD = GEAR option. This overrides the LVLTIM = 2 setting made by METHOD = GEAR.)

  TRAP (trapezoidal) integration generally results in reduced program execution time, with more accurate results. However, trapezoidal integration can introduce an apparent oscillation on printed or plotted nodes that might not be caused by circuit behavior. To test if this is the case, run a transient analysis with a small timestep. If the oscillation disappears, it was due to the trapezoidal method.

  The GEAR method acts as a filter, removing the oscillations found in the trapezoidal method. Highly nonlinear circuits such as operational amplifiers can require very long execution times with the GEAR method. Circuits that are not convergent with trapezoidal integration often converge with GEAR. Default = TRAP (trapezoidal).
### Tolerance Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( ABSH = x )</td>
<td>Sets the absolute current change through voltage defined branches (voltage sources and inductors). In conjunction with DI and RELH, ( ABSH ) is used to check for current convergence. Default = 0.0.</td>
</tr>
<tr>
<td>( ABSV = x )</td>
<td>Sets the absolute minimum voltage for DC and transient analysis. Decrease VNTOL if accuracy is of more concern than convergence. If voltages less than 50 microvolts are required, VNTOL can be reduced to two orders of magnitude less than the smallest desired voltage, ensuring at least two digits of significance. Typically, VNTOL need not be changed unless the circuit is a high voltage circuit. For 1000 volt circuits, a reasonable value can be 5 to 50 millivolts. ( ABSV ) is the same as VNTOL. Default = 50 (microvolts).</td>
</tr>
<tr>
<td>( ABSVAR = x )</td>
<td>Sets the limit on the maximum voltage change from one time point to the next. Used with the DVDT algorithm. If the simulator produces a convergent solution that is greater than ( ABSVAR ), the solution is discarded, the timestep is set to a smaller value, and the solution is recalculated. This is called a timestep reversal. Default = 0.5 (volts).</td>
</tr>
</tbody>
</table>
### ACCURATE

Selects a time algorithm that uses LVLTIM = 3 and DVDT = 2 for circuits such as high-gain comparators. Circuits that combine high gain with large dynamic range should use this option to guarantee solution accuracy. When ACCURATE is set to 1, it sets the following control options:

- LVLTIM = 3
- DVDT = 2
- RELVAR = 0.2
- ABSVAR = 0.2
- FT = 0.2
- RELMOS = 0.01

Default = 0.

### BYTOL = x

Specifies the tolerance for the voltage at which a MOSFET, MESFET, JFET, BJT, or diode is considered latent. Star-Hspice does not update the status of latent devices. Default = MBYPASS x VNTOL.

### CHGTOL = x

Sets the charge error tolerance when LVLTIM = 2 is set. CHGTOL, along with RELQ, sets the absolute and relative charge tolerance for all Star-Hspice capacitances. Default = 1e-15 (coulomb).

### DI = x

Sets the maximum iteration-to-iteration current change through voltage defined branches (voltage sources and inductors). This option is only applicable when the value of the DI control option is greater than 0. Default = 0.0.
### FAST

Speeds up simulation by not updating the status of latent devices. This option is applicable for MOSFETs, MESFETs, JFETs, BJTs, and diodes. Default = 0.

A device is considered to be latent when its node voltage variation from one iteration to the next is less than the value of either the BYTOL control option or the BYPASSTOL element parameter. (When FAST is on, Star-Hspice sets BYTOL to different values for different types of device models.)

In addition to the FAST option, the input preprocessing time can be reduced by the options NOTOP and NOELCK. Increasing the value of the MBYPASS option or the BYTOL option setting also helps simulations run faster, but can reduce accuracy.

### MAXAMP = x

Sets the maximum current through voltage defined branches (voltage sources and inductors). If the current exceeds the MAXAMP value, an error is issued. Default = 0.0.

### MBYPASS = x

Used to compute the default value for the BYTOL control option:

$$\text{BYTOL} = \text{MBYPASS} \times \text{VNTOL}$$

Also multiplies voltage tolerance RELV. MBYPASS should be set to about 0.1 for precision analog circuits. Default = 1 for DVDT = 0, 1, 2, or 3. Default = 2 for DVDT = 4.

### MU = x, XMU = x

The coefficient for trapezoidal integration. The range for MU is 0.0 to 0.5. XMU is the same as MU. Default = 0.5.
### Understanding the Control Options

#### Performing Transient Analysis

<table>
<thead>
<tr>
<th>Control Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RELH = x</strong></td>
<td>Sets relative current tolerance through voltage defined branches (voltage sources and inductors). RELH is used to check current convergence. This option is applicable only if the value of the ABSH control option is greater than zero. Default = 0.05.</td>
</tr>
<tr>
<td><strong>RELI = x</strong></td>
<td>Sets the relative error/tolerance change from iteration to iteration to determine convergence for all currents in diode, BJT, and JFET devices. (RELMOS sets the tolerance for MOSFETs). This is the percent change in current from the value calculated at the previous timepoint. Default = 0.01 for KCLTEST = 0, 1e-4 for KCLTEST = 1.</td>
</tr>
<tr>
<td><strong>RELQ = x</strong></td>
<td>Used in the local truncation error timestep algorithm (LVLTIM = 2). RELQ changes the size of the timestep. If the capacitor charge calculation of the present iteration exceeds that of the past iteration by a percentage greater than the value of RELQ, the internal timestep (Delta) is reduced. Default = 0.01.</td>
</tr>
<tr>
<td><strong>RELTOL, RELV</strong></td>
<td>Sets the relative error tolerance for voltages. RELV is used in conjunction with the ABSV control option to determine voltage convergence. Increasing RELV increases the relative error. RELV is the same as RELTOL. Options RELI and RELVDC default to the RELTOL value. Default = 1e-3.</td>
</tr>
<tr>
<td><strong>RELVAR = x</strong></td>
<td>Used with ABSVAR and the timestep algorithm option DVDT. RELVAR sets the relative voltage change for LVLTIM = 1 or 3. If the nodal voltage at the current time point exceeds the nodal voltage at the previous time point by RELVAR, the timestep is reduced and a new solution at a new time point is calculated. Default = 0.30 (30%).</td>
</tr>
</tbody>
</table>
### Performing Transient Analysis

#### Understanding the Control Options

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SLOPETOL = x</strong></td>
<td>Sets a lower limit for breakpoint table entries in a piecewise linear (PWL) analysis. If the difference in the slopes of two consecutive PWL segment is less than the SLOPETOL value, the breakpoint table entry for the point between the segments is ignored. Default = 0.5</td>
</tr>
<tr>
<td><strong>TIMERES = x</strong></td>
<td>Sets a minimum separation between breakpoint values for the breakpoint table. If two breakpoints are closer together in time than the TIMERES value, only one of them is entered in the breakpoint table. Default = 1 ps.</td>
</tr>
<tr>
<td><strong>TRTOL = x</strong></td>
<td>Used in the local truncation error timestep algorithm (LVLTIM = 2). TRTOL is a multiplier of the internal timestep generated by the local truncation error timestep algorithm. TRTOL reduces simulation time, while maintaining accuracy. It is a factor that estimates the amount of error introduced by truncating the Taylor series expansion used in the algorithm. This error is a reflection of what the minimum value of the timestep should be to reduce simulation time and maintain accuracy. The range of TRTOL is 0.01 to 100, with typical values being in the 1 to 10 range. If TRTOL is set to 1, the minimum value, a very small timestep is used. As the setting of TRTOL increases, the timestep size increases. Default = 7.0.</td>
</tr>
</tbody>
</table>
### Limit Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VNTOL</strong> = x, <strong>ABSV</strong></td>
<td>Sets the absolute minimum voltage for DC and transient analysis. Decrease VNTOL if accuracy is of more concern than convergence. If voltages less than 50 microvolts are required, VNTOL can be reduced to two orders of magnitude less than the smallest desired voltage, ensuring at least two digits of significance. Typically, VNTOL need not be changed unless the circuit is a high voltage circuit. For 1000 volt circuits, a reasonable value can be 5 to 50 millivolts. ABSV is the same as VNTOL. Default = 50 (microvolts).</td>
</tr>
<tr>
<td><strong>XMU</strong> = x</td>
<td>The coefficient for trapezoidal integration. The range for MU is 0.0 to 0.5. XMU is the same as MU. Default = 0.5.</td>
</tr>
<tr>
<td><strong>AUTOSTOP</strong></td>
<td>Stops the transient analysis when all TRIG-TARG and FIND-WHEN measure functions are calculated. This option can result in a substantial CPU time reduction. If the data file contains measure functions such as AVG, RMS, MIN, MAX, PP, ERR, ERR1,2,3, and PARAM, then AUTOSTOP is disabled.</td>
</tr>
<tr>
<td><strong>BKPSIZ</strong> = x</td>
<td>Sets the size of the breakpoint table. Default = 5000.</td>
</tr>
<tr>
<td><strong>DELMAX</strong> = x</td>
<td>Sets the maximum value for the internal timestep Delta. Star-Hspice automatically sets the DELMAX value based on various factors, which are listed in “Timestep Control for Accuracy” on page 11-26. This means that the initial DELMAX value shown in the Star-Hspice output listing is generally not the value used for simulation.</td>
</tr>
</tbody>
</table>
### DVTR

Allows the use of voltage limiting in transient analysis. Default = 1000.

### FS = x

Sets the fraction of a timestep (TSTEP) that Delta (the internal timestep) is decreased for the first time point of a transient. Decreasing the FS value helps circuits that have timestep convergence difficulties. It also is used in the DVDT = 3 method to control the timestep.

\[
Delta = FS \times [MIN(TSTEP, DELMAX, BKPT)]
\]

where DELMAX is specified and BKPT is related to the breakpoint of the source. TSTEP is set in the .TRAN statement. Default = 0.25.

### FT = x

Sets the fraction of a timestep (TSTEP) by which Delta (the internal timestep) is decreased for an iteration set that does not converge. It is also used in DVDT = 2 and DVDT = 4 to control the timestep. Default = 0.25.

### GMIN = x

Sets the minimum conductance allowed for in a transient analysis time sweep. Default = 1e-12.

### IMAX = x, ITL4 = x

Determines the maximum timestep in the timestep algorithms used for transient analysis simulations. IMAX sets an upper limit on the number of iterations allowed to obtain a convergent solution at a timepoint. If the number of iterations needed is greater than IMAX, the internal timestep Delta is decreased by a factor equal to the transient control option FT, and a new solution is calculated using the new timestep. IMAX also works in conjunction with the transient control option IMIN. ITL4 is the same as IMAX. Default = 8.0.
**IMIN** = \(x\),
**ITL3** = \(x\)  
Determines the timestep in the algorithms used for transient analysis simulations. **IMIN** sets a lower limit on the number of iterations required to obtain convergence. If the number of iterations is less than **IMIN**, the internal timestep, Delta, is doubled. This option is useful for decreasing simulation times in circuits where the nodes are stable most of the time, such as digital circuits. If the number of iterations is greater than **IMIN**, the timestep is kept the same unless the option **IMAX** is exceeded (see **IMAX**). **ITL3** is the same as **IMIN**. Default = 3.0.

**ITL4** = \(x\)  
Determines the maximum timestep in the timestep algorithms used for transient analysis simulations. **IMAX** sets an upper limit on the number of iterations allowed to obtain a convergent solution at a timepoint. If the number of iterations needed is greater than **IMAX**, the internal timestep Delta is decreased by a factor equal to the transient control option **FT**, and a new solution is calculated using the new timestep. **IMAX** also works in conjunction with the transient control option **IMIN**. **ITL4** is the same as **IMAX**. Default = 8.0.
Performing Transient Analysis

Understanding the Control Options

| **ITL5 = x** | Sets the transient analysis total iteration limit. If a circuit uses more than ITL5 iterations, the program prints all results to that point. The default allows an infinite number of iterations. Default = 0.0. |
| **RMAX = x** | Sets the TSTEP multiplier, which determines the maximum value, DELMAX, that can be used for the internal timestep Delta: delmax = TSTEP x RMAX. Default = 5 when dvdt = 4 and lvltim = 1, otherwise, default = 2. |
| **RMIN = x** | Sets the minimum value of Delta (internal timestep). An internal timestep smaller than RMIN x TSTEP results in termination of the transient analysis with the error message “internal timestep too small”. Delta is decreased by the amount set by the FT option if the circuit has not converged in IMAX iterations. Default = 1.0e-9. |
| **VFLOOR = x** | Sets a lower limit for the voltages that are printed in the output listing. All voltages lower than VFLOOR are printed as 0. This only affects the output listing: the minimum voltage used in a simulation is set by VNTOL (ABSV). |

Matrix Manipulation Options

After linearization of the individual elements within a Star-Hspice input netlist file, the linear equations are constructed for the matrix. User-controlled variables affecting the construction and solution of the matrix equation include options PIVOT and GMIN. GMIN places a variable into the matrix that prevents the matrix becoming ill-conditioned.
Pivot Option

Select the PIVOT option for a number of different pivoting methods to reduce simulation time and assist in both DC and transient convergence. Pivoting reduces the error resulting from elements in the matrix that are widely different in magnitude. The use of PIVOT results in a search of the matrix for the largest element value. This element value then is used as the pivot.
Controlling Simulation Speed and Accuracy

Convergence is defined as the ability to obtain a solution to a set of circuit equations within a given tolerance criteria and number of iterations. In numerical circuit simulation, the designer specifies a relative and absolute accuracy for the circuit solution and the simulator iteration algorithm attempts to converge to a solution that is within these set tolerances. In many cases, the speed of reaching a solution also is of primary interest to the designer, particularly for preliminary design trials, and some accuracy is willingly sacrificed.

Simulation Speed

Star-Hspice can substantially reduce the computer time needed to solve complex problems. The following user options alter internal algorithms to increase simulation efficiency.

- .OPTIONS FAST – sets additional options that increase simulation speed with little loss of accuracy
- .OPTIONS AUTOSTOP – terminates the simulation when all .MEASURE statements have completed. This is of special interest when testing corners.

The FAST and AUTOSTOP options are described in “Understanding the Control Options” on page 11-11.

Simulation Accuracy

Star-Hspice is shipped with control option default values that aim for superior accuracy while delivering good performance in simulation time. The control options and their default settings to maximize accuracy are:

- DVT = 4   LVLTIM = 1   RMAX = 5   SLOPETOL = 0.75
- FT = FS = 0.25BYPASS = 1   BYTOL = MBYPASSxVNTOL = 0.100m
Controlling Simulation Speed and Accuracy

Performing Transient Analysis

---

**Note:** BYPASS is only turned on (set to 1) when DVDT = 4. For other DVDT settings, BYPASS is off (0). SLOPETOL is set to 0.75 when DVDT = 4 and LVLTIM = 1. For all other values of DVDT or LVLTIM, SLOPETOL defaults to 0.5.

---

**Timestep Control for Accuracy**

The DVDT control option selects the timestep control algorithm. Relationships between DVDT and other control options are discussed in “Selecting Timestep Control Algorithms” on page 11-32.

The DELMAX control option also affects simulation accuracy. DELMAX specifies the maximum allowed timestep size. If DELMAX is not set in an .OPTIONS statement, Star-Hspice computes a DELMAX value. Factors that determine the computed DELMAX value are:

- .OPTIONS RMAX and FS
- Breakpoint locations for a PWL source
- Breakpoint locations for a PULSE source
- Smallest period for a SIN source
- Smallest delay for a transmission line component
- Smallest ideal delay for a transmission line component
- TSTEP value in a .TRAN analysis
- Number of points in an FFT analysis

The FS and RMAX control options provide some user control over the DELMAX value. The FS option, which defaults to 0.25, scales the breakpoint interval in the DELMAX calculation. The RMAX option, which defaults to 5 if DVDT = 4 and LVLTIM = 1, scales the TSTEP (timestep) size in the DELMAX calculation.

For circuits that contain oscillators or ideal delay elements, an .OPTIONS statement should be used to set DELMAX to one-hundredth of the period or less.
Performing Transient Analysis  

Controlling Simulation Speed and Accuracy

The ACCURATE control option tightens the simulation options to give the most accurate set of simulation algorithms and tolerances. When ACCURATE is set to 1, it sets the following control options:

- DVDT = 2
- LVLTIM = 3
- FT = FS = 0.2
- SLOPETOL = 0.5
- BYTOL = 0
- BYPASS = 0
- RMAX = 2
- RELVAR = 0.2
- ABSVAR = 0.2
- RELMOS = 0.01

Models and Accuracy

Simulation accuracy relies heavily on the sophistication and accuracy of the models used. More advanced MOS, BJT, and GaAs models give superior results for critical applications. Simulation accuracy is increased by:

- Algebraic models that describe parasitic interconnect capacitances as a function of the width of the transistor. The wire model extension of the resistor can model the metal, diffusion, or poly interconnects to preserve the relationship between the physical layout and electrical property.
- MOS model parameter ACM that calculates defaults for source and drain junction parasitics. Star-Hspice uses ACM equations to calculate the size of the bottom wall, the length of the sidewall diodes, and the length of a lightly doped structure. SPICE defaults with no calculation of the junction diode. Specify AD, AS, PD, PS, NRD, NRS to override the default calculations.
- MOS model parameter CAPOP = 4 that models the most advanced charge conservation, non-reciprocal gate capacitances. The gate capacitors and overlaps are calculated from the IDS model for LEVEL 49 or 53, however, the CAPOP parameter is ignored, model parameter CAPMOD with reasonable value should be used instead.
Guidelines for Choosing Accuracy Options

Use the ACCURATE option for
- Analog or mixed signal circuits
- Circuits with long time constants, such as RC networks
- Circuits with ground bounce

Use the default options (DVDT = 4) for
- Digital CMOS
- CMOS cell characterization
- Circuits with fast moving edges (short rise and fall times)

For ideal delay elements, use one of the following:
- ACCURATE
- DVDT = 3
- DVDT = 4, and, if the minimum pulse width of any signal is less than the minimum ideal delay, set DELMAX to a value smaller than the minimum pulse width.
Numerical Integration Algorithm Controls

When using Star-Hspice for transient analysis, you can select one of three options, Gear, Backward-Euler or Trapezoidal, to convert differential terms into algebraic terms.

Syntax

Gear algorithm:

.OPTION METHOD = GEAR

Backward-Euler:

.OPTION METHOD = GEAR MU = 0

Trapezoidal algorithm (default):

.OPTION METHOD = TRAP

Each of these algorithms has advantages and disadvantages, but the trapezoidal is the preferred algorithm overall because of its highest accuracy level and lowest simulation time.

The selection of the algorithm is not, however, an elementary task. The appropriate algorithm for convergence depends to a large degree on the type of circuit and its associated behavior for different input stimuli.

Gear and Trapezoidal Algorithms

The timestep control algorithm is automatically set by the choice of algorithm. In Star-Hspice, if the GEAR algorithm is selected (including Backward-Euler), the timestep control algorithm defaults to the truncation timestep algorithm. On the other hand, if the trapezoidal algorithm is selected, the DVDT algorithm is the default. You can change these Star-Hspice default by using the timestep control options.
Figure 11-2: Time Domain Algorithm

- Initialization
- Iteration Solution
- Converged
- Reversal
- Time Step Algorithm
- Advancement \( t_{\text{new}} = t_{\text{old}} + \Delta t \)
- Time Step Limit Check
- Extrapolated Solution for timepoint \( n \)
- Timestep too small error
- FAIL
One limitation of the trapezoidal algorithm is that it can result in computational oscillation—that is, an oscillation caused by the trapezoidal algorithm and not by the circuit design. This also produces an unusually long simulation time. When this occurs in circuits that are inductive in nature, such as switching regulators, use the GEAR algorithm.
Selecting Timestep Control Algorithms

Star-Hspice allows the selection of three dynamic timestep control algorithms:

- Iteration Count Dynamic Timestep Algorithm
- Local Truncation Error (LTE) Dynamic Timestep Algorithm
- DVDT Dynamic Timestep Algorithm

Each of these algorithms uses a dynamically changing timestep. A dynamically changing timestep increases the accuracy of simulation and reduces the simulation time by varying the value of the timestep over the transient analysis sweep depending upon the stability of the output. Dynamic timestep algorithms increase the timestep value when internal nodal voltages are stable and decrease the timestep value when nodal voltages are changing quickly.

Figure 11-4: Internal Variable Timestep
In Star-Hspice, the timestep algorithm is selected by the LVLTIM option:

- LVLTIM = 0 selects the iteration count algorithm.
- LVLTIM = 1 selects the DVDT timestep algorithm, along with the iteration count algorithm. Operation of the timestep control algorithm is controlled by the setting of the DVDT control option. For LVLTIM = 1 and DVDT = 0, 1, 2, or 3, the algorithm does not use timestep reversal. For DVDT = 4, the algorithm uses timestep reversal.

The DVDT algorithm is discussed further in “DVDT Dynamic Timestep Algorithm” on page 11-34.

- LVLTIM = 2 selects the truncation timestep algorithm, along with the iteration count algorithm with reversal.
- LVLTIM = 3 selects the DVDT timestep algorithm with timestep reversal, along with the iteration count algorithm. For LVLTIM = 3 and DVDT = 0, 1, 2, 3, or 4, the algorithm uses timestep reversal.

**Iteration Count Dynamic Timestep Algorithm**

The simplest dynamic timestep algorithm used is the iteration count algorithm. The iteration count algorithm is controlled by the following options:

<table>
<thead>
<tr>
<th><strong>IMAX</strong></th>
<th>Controls the internal timestep size based on the number of iterations required for a timepoint solution. If the number of iterations per timepoint exceeds the IMAX value, the internal timestep is decreased. Default = 8.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IMIN</strong></td>
<td>Controls the internal timestep size based on the number of iterations required for the previous timepoint solution. If the last timepoint solution took fewer than IMIN iterations, the internal timestep is increased. Default = 3.</td>
</tr>
</tbody>
</table>
Local Truncation Error (LTE) Dynamic Timestep Algorithm

The local truncation error timestep method uses a Taylor series approximation to calculate the next timestep for a transient analysis. This method calculates an internal timestep using the allowed local truncation error. If the calculated timestep is smaller than the current timestep, then the timepoint is set back (timestep reversal) and the calculated timestep is used to increment the time. If the calculated timestep is larger than the current one, then there is no need for a reversal. A new timestep is used for the next timepoint.

The local truncation error timestep algorithm is selected by setting LVLTIM = 2. The control options available with the local truncation error algorithm are:

- TRTOL (default = 7)
- CHGTOL (default = 1e-15)
- RELQ (default = 0.01)

DVDT Dynamic Timestep Algorithm

Select this algorithm by setting the option LVLTIM to 1 or 3. If you set LVLTIM = 1, the DVDT algorithm does not use timestep reversal. The results for the current timepoint are saved, and a new timestep is used for the next timepoint. If you set LVLTIM = 3, the algorithm uses timestep reversal. If the results are not converging at a given iteration, the results of current timepoint are ignored, time is set back by the old timestep, and a new timestep is used.

Therefore, LVLTIM = 3 is more accurate and more time consuming than LVLTIM = 1.

The test the algorithm uses for reversing the timestep depends on the DVDT control option setting. For DVDT = 0, 1, 2, or 3, the decision is based on the SLOPETOL control option value. For DVDT = 4, the decision is based on the settings of the SLOPETOL, RELVAR, and ABSVAR control options.

The DVDT algorithm calculates the internal timestep based on the rate of nodal voltage changes. For circuits with rapidly changing nodal voltages, the DVDT algorithm uses a small timestep. For circuits with slowly changing nodal voltages, the DVDT algorithm uses larger timesteps.
The DVDT = 4 setting selects a timestep control algorithm that is based on nonlinearity of node voltages, and employs timestep reversals if the LVLTIM option is set to either 1 or 3. The nonlinearity of node voltages is measured through changes in slopes of the voltages. If the change in slope is larger than the setting of the SLOPETOL control option, the timestep is reduced by a factor equal to the setting of the FT control option. The FT option defaults to 0.25. Star-Hspice sets the SLOPETOL value to 0.75 for LVLTIM = 1, and to 0.50 for LVLTIM = 3. Reducing the value of SLOPETOL increases simulation accuracy, but also increases simulation time. For LVLTIM = 1, the simulation accuracy can be controlled by SLOPETOL and FT. For LVLTIM = 3, the RELVAR and ABSVAR control options also affect the timestep, and therefore affect the simulation accuracy.

You can use options RELVAR and ABSVAR in conjunction with the DVDT option to improve simulation time or accuracy. For faster simulation time, RELVAR and ABSVAR should be increased (although this might decrease accuracy).

---

**Note:** If you need backward compatibility with Star-Hspice Release 95.3, use the following option values. Setting .OPTIONS DVDT = 3 sets all of these values automatically.

- LVLTIM = 1
- RMAX = 2
- SLOPETOL = 0.5
- FT = FS = 0.25
- BYPASS = 0
- BYTOL = 0.050

---

**User Timestep Controls**

The RMIN, RMAX, FS, FT, and DELMAX control options allow you to control the minimum and maximum internal timestep allowed for the DVDT algorithm. If the timestep falls below the minimum timestep default, the execution of the program halts. For example, an “internal timestep too small” error results when the timestep becomes less than the minimum internal timestep found by TSTEPxRMIN.
Note: RMIN is the minimum timestep coefficient and has a default value of 1e-9. TSTEP is the time increment, and is set in the .TRAN statement.

If DELMAX is set in an .OPTIONS statement, then DVDT = 0 is used. If DELMAX is not specified in an .OPTIONS statement, Star-Hspice computes a DELMAX value. For DVDT = 0, 1, or 2, the maximum internal timestep is:

\[
\text{min}\left(\frac{TSTOP}{50}, \text{DELMAX}, (TSTEP \times RMAX)\right)
\]

The TSTOP time is the transient sweep range set in the .TRAN statement.

In circuits with piecewise linear (PWL) transient sources, the SLOPETOL option also affects the internal timestep. A PWL source with a large number of voltage or current segments contributes a correspondingly large number of entries to the internal breakpoint table. The number of breakpoint table entries that must be considered contributes to the internal timestep control.

If the difference in the slope of consecutive segments of a PWL source is less than the SLOPETOL value, the breakpoint table entry for the point between the segments is ignored. For a PWL source with a signal that changes value slowly, ignoring its breakpoint table entries can help reduce the simulation time. Since the data in the breakpoint table is a factor in the internal timestep control, reducing the number of usable breakpoint table entries by setting a high SLOPETOL reduces the simulation time.
Performing Fourier Analysis

This section describes the flow for Fourier and FFT Analysis.

Figure 11-5: Fourier and FFT Analysis

Performing Transient Analysis

Performing Fourier Analysis
There are two different Fourier analyses available in Star-Hspice: .FOUR and .FFT. The former is the same as is available in SPICE 2G6, a standard, fixed-window analysis tool. The latter is a much more flexible Fourier analysis tool, and is recommended for analysis tasks requiring more detail and precision.

**.FOUR Statement**

This statement performs a Fourier analysis as a part of the transient analysis. The Fourier analysis is performed over the interval (tstop-fperiod, tstop), where tstop is the final time specified for the transient analysis (see .TRAN statement), and fperiod is one period of the fundamental frequency (parameter “freq”). Fourier analysis is performed on 101 points of transient analysis data on the last 1/f time period, where f is the fundamental Fourier frequency. Transient data is interpolated to fit on 101 points running from (tstop-1/f) to tstop. The phase, the normalized component, and the Fourier component are calculated using 10 frequency bins. The Fourier analysis determines the DC component and the first nine AC components.

**Syntax**

```
.FOUR freq ov1 <ov2 ov3 ...>
```

- `freq` : the fundamental frequency
- `ov1` ... : the output variables for which the analysis is desired

**Example**

```
.FOUR 100K V(5)
```

**Accuracy and DELMAX**

For maximum accuracy, .OPTION DELMAX should be set to (period/100). For circuits with very high resonance factors (high Q circuits such as crystal oscillators, tank circuits, and active filters) DELMAX should be set to less than (period/100).
Fourier Equation

The total harmonic distortion is the square root of the sum of the squares of the second through the ninth normalized harmonic, times 100, expressed as a percent:

\[
THD = \frac{1}{R_1} \cdot \left( \sum_{m=2}^{9} R_m^2 \right)^{1/2} \cdot 100\%
\]

This interpolation can result in various inaccuracies. For example, if the transient analysis runs at intervals longer than 1/(101*f), the frequency response of the interpolation dominates the power spectrum. Furthermore, there is no error range derived for the output.

The Fourier coefficients are calculated from:

\[
g(t) = \sum_{m=0}^{9} C_m \cdot \cos(mt) + \sum_{m=0}^{9} D_m \cdot \sin(mt)
\]

where

\[
C_m = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} g(t) \cdot \cos(mt) \cdot dt
\]

\[
D_m = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} g(t) \cdot \sin(mt) \cdot dt
\]

\[
g(t) = \sum_{m=0}^{9} C_m \cdot \cos(mt) + \sum_{m=0}^{9} D_m \cdot \sin(mt)
\]
Performing Fourier Analysis

C and D are approximated by:

\[ C_m = \sum_{n=0}^{101} g(n \cdot \Delta t) \cdot \cos\left(\frac{2 \cdot \pi \cdot m \cdot n}{101}\right) \]

\[ D_m = \sum_{n=0}^{101} g(n \cdot \Delta t) \cdot \sin\left(\frac{2 \cdot \pi \cdot m \cdot n}{101}\right) \]

The magnitude and phase are calculated by:

\[ R_m = (C_m^2 + D_m^2)^{1/2} \]

\[ \Phi_m = \arctan\left(\frac{C_m}{D_m}\right) \]

Example

The following is Star-Hspice input for an .OP, .TRAN, and .FOUR analysis.

CMOS INVERTER
*
M1 2 1 0 0 NMOS W = 20U L = 5U
M2 2 1 3 3 PMOS W = 40U L = 5U
VDD 3 0 5
VIN 1 0 SIN 2.5 2.5 20MEG
.MODEL NMOS NMOS LEVEL = 3 CGDO = .2N CGSO = .2N CGBO = 2N
.MODEL PMOS PMOS LEVEL = 3 CGDO = .2N CGSO = .2N CGBO = 2N
.OP
.TRAN 1N 100N
.FOUR 20MEG V(2)
.PRINT TRAN V(2) V(1)
.END

Output for the Fourier analysis is shown below.

******
cmos inverter
****** fourier analysis

| tnom | 25.000 |
| temp | 25.000 |

performing transient analysis  performing fourier analysis

star-hspice manual, release 2001.2 11-41

fourier components of transient response v(2)
dc component = 2.430d+00
harmonic  frequency  fourier  normalized  phase
normalized
no (hz) component component (deg)  phase (deg)
1 20.0000x  3.0462 1.0000 176.5386 0.
2 40.0000x  115.7006m 37.9817m -106.2672 -282.8057
3 60.0000x  753.0446m 247.2061m 170.7288 -5.8098
4 80.0000x  77.8910m 25.5697m -125.9511 -302.4897
5 100.0000x  296.5549m 97.3517m 164.5430 -11.9956
6 120.0000x  50.0994m 16.4464m -148.1115 -324.6501
7 140.0000x  25.6916m 8.4339m 172.9579 -3.5807
8 160.0000x  47.7347m 15.6701m 154.1858 -22.3528
9 180.0000x  50.0994m 16.4464m -148.1115 -324.6501
total harmonic distortion = 27.3791 percent

for further information on fourier analysis, see “performing fourier analysis” on page 11-37.

.fft statement

the syntax of the .fft statement is shown below. the parameters are described in table 11-1.

syntax

.fft <output_var> <start = value> <stop = value> <np = value>
+ <format = keyword> <window = keyword> <alfa = value>
+ <freq = value> <fmin = value> <fmax = value>

example

below are four examples of valid .fft statements.

.fft v(1)
.fft v(1,2) np = 1024 start = 0.3m stop = 0.5m freq = 5.0k
+ window = kaiser alfa = 2.5
.fft i(rload) start = 0m to = 2.0m fmin = 100k fmax = 120k
+ format = unorm
### Table 11-1: .FFT Statement Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>output_var</td>
<td>—</td>
<td>Can be any valid output variable, such as voltage, current, or power</td>
</tr>
<tr>
<td>START</td>
<td>see Description</td>
<td>Specifies the beginning of the output variable waveform to be analyzed. Defaults to the START value in the .TRAN statement, which defaults to 0.</td>
</tr>
<tr>
<td>FROM</td>
<td>see START</td>
<td>In .FFT statements, FROM is an alias for START.</td>
</tr>
<tr>
<td>STOP</td>
<td>see Description</td>
<td>Specifies the end of the output variable waveform to be analyzed. Defaults to the TSTOP value in the .TRAN statement.</td>
</tr>
<tr>
<td>TO</td>
<td>see STOP</td>
<td>In .FFT statements, TO is an alias for STOP.</td>
</tr>
<tr>
<td>NP</td>
<td>1024</td>
<td>Specifies the number of points used in the FFT analysis. NP must be a power of 2. If NP is not a power of 2, Star-Hspice automatically adjusts it to the closest higher number that is a power of 2.</td>
</tr>
<tr>
<td>FORMAT</td>
<td>NORM</td>
<td>Specifies the output format: NORM = normalized magnitude UNORM = unnormalized magnitude</td>
</tr>
<tr>
<td>WINDOW</td>
<td>RECT</td>
<td>Specifies the window type to be used: RECT = simple rectangular truncation window BART = Bartlett (triangular) window HANN = Hanning window HAMM = Hamming window BLACK = Blackman window HARRIS = Blackman-Harris window GAUSS = Gaussian window KAISER = Kaiser-Bessel window</td>
</tr>
<tr>
<td>ALFA</td>
<td>3.0</td>
<td>Specifies the parameter used in GAUSS and KAISER windows to control the highest side-lobe level, bandwidth, and so on. 1.0 ≤ ALFA ≤ 20.0</td>
</tr>
</tbody>
</table>
.fft par('v(1) + v(2)') from = 0.2u stop = 1.2u window = harris

Only one output variable is allowed in an .FFT command. The following is an incorrect use of the command.

```
.fft v(1) v(2) np = 1024
```

The correct use of the command is shown in the example below. In this case, a .flt0 and a .flt1 file are generated for the FFT of v(1) and v(2), respectively.

```
.fft v(1) np = 1024
.fft v(2) np = 1024
```
FFT Analysis Output

The results of the FFT analysis are printed in a tabular format in the .lis file, based on the parameters in the .FFT statement. The normalized magnitude values are printed unless you specify FORMAT = UNORM, in which case unnormalized magnitude values are printed. The number of printed frequencies is half the number of points (NP) specified in the .FFT statement. If you specify a minimum or a maximum frequency, using FMIN or FMAX, the printed information is limited to the specified frequency range. Moreover, if you specify a frequency of interest using FREQ, then the output is limited to the harmonics of this frequency, along with the percent of total harmonic distortion.

A .ft# file is generated, in addition to the listing file, for each FFT output variable, which contains the graphic data needed to display the FFT analysis waveforms. The magnitude in dB and the phase in degrees are available for display.

In the sample FFT analysis .lis file output below, notice that all the parameters used in the FFT analysis are defined in the header.

```
****** Sample FFT output extracted from the .lis file
fft test ... sine
****** fft analysis tnom = 25.000 temp = 25.000
****** fft components of transient response v(1)
Window: Rectangular
First Harmonic: 1.0000k
Start Freq: 1.0000k
Stop Freq: 10.0000k
dc component: mag(db) = -1.132D+02 mag = 2.191D-06 phase = 1.800D+02
```

<table>
<thead>
<tr>
<th>frequency index</th>
<th>frequency (hz)</th>
<th>fft_mag (db)</th>
<th>fft_mag (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.0000k</td>
<td>0</td>
<td>1.0000</td>
</tr>
<tr>
<td>4</td>
<td>2.0000k</td>
<td>-125.5914</td>
<td>525.3264n</td>
</tr>
<tr>
<td>6</td>
<td>3.0000k</td>
<td>-106.3740</td>
<td>4.8007u</td>
</tr>
<tr>
<td>8</td>
<td>4.0000k</td>
<td>-113.5753</td>
<td>2.0952u</td>
</tr>
<tr>
<td>10</td>
<td>5.0000k</td>
<td>-112.6689</td>
<td>2.3257u</td>
</tr>
<tr>
<td>12</td>
<td>6.0000k</td>
<td>-118.3365</td>
<td>1.2111u</td>
</tr>
</tbody>
</table>
Performing Transient Analysis

Performing Fourier Analysis

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>7.0000k</td>
<td>-109.8888</td>
<td>3.2030u</td>
</tr>
<tr>
<td>16</td>
<td>8.0000k</td>
<td>-117.4413</td>
<td>1.3426u</td>
</tr>
<tr>
<td>18</td>
<td>9.0000k</td>
<td>-97.5293</td>
<td>13.2903u</td>
</tr>
<tr>
<td>20</td>
<td>10.0000k</td>
<td>-114.3693</td>
<td>1.9122u</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total harmonic distortion = 1.5065m percent</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The preceding example specifies a frequency of 1 kHz and the THD up to 10 kHz, which corresponds to the first ten harmonics.

**Note:** The highest frequency shown in the Star-Hspice FFT output might not be identical to the specified FMAX, due to Star-Hspice adjustments.

Table 11-2 describes the output of the Star-Hspice FFT analysis.

**Table 11-2: .FFT Statement Output Description**

<table>
<thead>
<tr>
<th>Column Heading</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>frequency index</td>
<td>Runs from 1 to NP/2, or the corresponding index for FMIN and FMAX. The DC component corresponding to index 0 is displayed independently.</td>
</tr>
<tr>
<td>frequency</td>
<td>Actual frequency associated with the index</td>
</tr>
<tr>
<td>fft_mag (db), fft_mag</td>
<td>There are two FFT magnitude columns: the first in dB and the second in the units of the output variable. The magnitude is normalized unless UNORM format is specified.</td>
</tr>
<tr>
<td>fft_phase</td>
<td>Associated phase, in degrees</td>
</tr>
</tbody>
</table>

**Notes:**

1. The following formula should be used as a guideline when specifying a frequency range for FFT output:

   \[
   \text{frequency increment} = \frac{1.0}{\text{STOP} - \text{START}}
   \]

   Each frequency index corresponds to a multiple of this increment. To obtain a finer frequency resolution, maximize the duration of the time window.
2. FMIN and FMAX have no effect on the .ft0, .ft1, ..., .ftn files.

For further information on the .FFT statement, see “Using the .FFT Statement” on page 28-7.
Chapter 12

AC Sweep and Small Signal Analysis

This chapter describes performing an AC sweep and small signal analysis. It covers the following topics:

- Understanding AC Small Signal Analysis
- Using the .AC Statement
- Using Other AC Analysis Statements
Understanding AC Small Signal Analysis

The AC small signal analysis portion of Star-Hspice computes (see Figure 12-1) AC output variables as a function of frequency. Star-Hspice first solves for the DC operating point conditions, which are used to develop linearized, small-signal models for all nonlinear devices in the circuit.

**Figure 12-1: AC Small Signal Analysis Flow**
Capacitor and inductor values are converted to their corresponding admittances:
\[ \text{for capacitors} \]
\[ Y_C = j\omega C \]
and
\[ \text{for inductors} \]
\[ Y_L = \frac{1}{j\omega L} \]
Star-Hspice allows resistors to have different DC and AC values. If AC = <value> is specified in a resistor statement, the operating point is calculated using the DC value of resistance, but the AC resistance value is used in the AC analysis. This is convenient when analyzing operational amplifiers, since the operating point computation can be performed on the unity gain configuration using a low value for the feedback resistance. The AC analysis then can be performed on the open loop configuration by using a very large value for the AC resistance.

AC analysis of bipolar transistors is based on the small-signal equivalent circuit, as described in “Using BJT Model Equations (NPN and PNP)” on page 16-28. MOSFET AC equivalent circuit models are described in “Introducing MOSFETs” on page 20-1.

The AC analysis statement permits sweeping values for:

- Frequency
- Element
- Temperature
- Model parameter
- Randomized distribution (Monte Carlo)
- Optimization and AC design analysis

Additionally, as part of the small signal analysis tools, Star-Hspice provides:

- Noise analysis
- Distortion analysis
- Network analysis
- Sampling noise
Using the .AC Statement

You can use the .AC statement in several different formats, depending on the application, as shown in the examples below. The parameters are described below.

Syntax

Single/double sweep:

.AC type np fstart fstop

or

.AC type np fstart fstop <SWEEP var start stop incr>

or

.AC type np fstart fstop <SWEEP var type np start stop>

or

.AC var1 START = <param_expr1> STOP = <param_expr2> + STEP = <param_expr3>

or

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

Parameterized sweep:

.AC type np fstart fstop <SWEEP DATA = datanm>

or

.AC DATA = datanm

Optimization:

.AC DATA = datanm OPTIMIZE = opt_par_fun RESULTS = measnames + MODEL = optmod
**Random/Monte Carlo:**

```
.AC type np fstart fstop <SWEEP MONTE = val>
```

The `.AC` statement keywords and parameters have the following descriptions:

<table>
<thead>
<tr>
<th><strong>DATA = datann</strong></th>
<th>Data name referred to in the <code>.AC</code> statement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>incr</strong></td>
<td>voltage, current, element or model parameter increment value</td>
</tr>
</tbody>
</table>

*Note: If “type” variation is used, the “np” (number of points) is specified instead of “incr”.*

<table>
<thead>
<tr>
<th><strong>fstart</strong></th>
<th>Starting frequency</th>
</tr>
</thead>
</table>

*Note: If type variation “POI” (list of points) is used, a list of frequency values is specified instead of “fstart fstop”.*

<table>
<thead>
<tr>
<th><strong>fstop</strong></th>
<th>Final frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MONTE = val</strong></td>
<td>Produces a number <code>val</code> of randomly-generated values that are used to select parameters from a distribution. The distribution can be Gaussian, Uniform, or Random Limit. See “Performing Monte Carlo Analysis” on page 13-14 for more information.</td>
</tr>
<tr>
<td><strong>np</strong></td>
<td>Number of points per decade or per octave, or just number of points, depending on the preceding keyword</td>
</tr>
<tr>
<td><strong>start</strong></td>
<td>Starting voltage, current, any element or model parameter value</td>
</tr>
<tr>
<td><strong>stop</strong></td>
<td>Final voltage, current, any element or model parameter value</td>
</tr>
</tbody>
</table>
### Using the .AC Statement

**AC Sweep and Small Signal Analysis**

| SWEEP | Keyword to indicate a second sweep is specified in the .AC statement |
| TEMP  | Keyword to indicate a temperature sweep |
| **type** | Can be any of the following keywords:  |
| ■ DEC – decade variation  |
| ■ OCT – octave variation  |
| ■ LIN – linear variation  |
| ■ POI – list of points  |
| **var** | Name of an independent voltage or current source, any element or model parameter, or the keyword TEMP (indicating a temperature sweep). Star-Hspice supports source value sweep, referring to the source name (SPICE style). However, if parameter sweep, a .DATA statement, and temperature sweep are selected, a parameter name must be chosen for the source value and subsequently referred to in the .AC statement. The parameter name can not start with V or I. |

### Example

The following example performs a frequency sweep by 10 points per decade from 1 kHz to 100 MHz.

```
.AC DEC 10 1K 100MEG
```

The next line calls for a 100 point frequency sweep from 1 Hz to 100 Hz.

```
.AC LIN 100 1 100HZ
```

The following example performs an AC analysis for each value of cload, which results from a linear sweep of cload between 1 pF and 10 pF (20 points), sweeping frequency by 10 points per decade from 1 Hz to 10 kHz.

```
.AC DEC 10 1 10K SWEEP cload LIN 20 1pf 10pf
```
The following example performs an AC analysis for each value of \( rx \), 5 k and 15 k, sweeping frequency by 10 points per decade from 1 Hz to 10 kHz.

\[
\text{.AC DEC 10 1 10K SWEEP rx n POI 2 5k 15k}
\]

The next example uses the DATA statement to perform a series of AC analyses modifying more than one parameter. The parameters are contained in the file \( \text{datanm} \).

\[
\text{.AC DEC 10 1 10K SWEEP DATA = datanm}
\]

The following example illustrates a frequency sweep along with a Monte Carlo analysis with 30 trials.

\[
\text{.AC DEC 10 1 10K SWEEP MONTE = 30}
\]

When an .AC statement is included in the input file, Star-Hspice performs an AC analysis of the circuit over the specified frequency range for each parameter value specified in the second sweep.

For an AC analysis, at least one independent AC source element statement must be in the data file (for example, VI INPUT GND AC 1V). Star-Hspice checks for this condition and reports a fatal error if no such AC sources have been specified (see “Using Sources and Stimuli” on page 5-1).

### AC Control Options

<table>
<thead>
<tr>
<th><strong>ABSH</strong> = ( x )</th>
<th>Sets the absolute current change through voltage defined branches (voltage sources and inductors). In conjunction with DI and RELH, ABSH is used to check for current convergence. Default = 0.0.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACOUT</strong></td>
<td>AC output calculation method for the difference in values of magnitude, phase and decibels for prints and plots. Default = 1.</td>
</tr>
</tbody>
</table>

The default value, ACOUT = 1, selects the Star-Hspice method, which calculates the difference of the magnitudes of the values. The SPICE method, ACOUT = 0, calculates the magnitude of the differences.
### DI = x
- Sets the maximum iteration-to-iteration current change through voltage defined branches (voltage sources and inductors). This option is only applicable when the value of the DI control option is greater than 0. Default = 0.0.

### MAXAMP = x
- Sets the maximum current through voltage defined branches (voltage sources and inductors). If the current exceeds the MAXAMP value, an error message is issued. Default = 0.0.

### RELH = x
- Sets relative current tolerance through voltage defined branches (voltage sources and inductors). It is used to check current convergence. This option is applicable only if the value of the ABSH control option is greater than zero. Default = 0.05.

### UNWRAP
- Displays phase results in AC analysis in unwrapped form (with a continuous phase plot). This allows accurate calculation of group delay. Note that group delay is always computed based on unwrapped phase results, even if the UNWRAP option is not set.
Using Other AC Analysis Statements

This section describes how to use other AC analysis statements.

**.DISTO Statement — AC Small-Signal Distortion Analysis**

The .DISTO statement causes Star-Hspice to compute the distortion characteristics of the circuit in an AC small-signal, sinusoidal, steady-state analysis. The program computes and reports five distortion measures at the specified load resistor. The analysis is performed assuming that one or two signal frequencies are imposed at the input. The first frequency, F1 (used to calculate harmonic distortion), is the nominal analysis frequency set by the .AC statement frequency sweep. The optional second input frequency, F2 (used to calculate intermodulation distortion), is set implicitly by specifying the parameter skw2, which is the ratio F2/F1.

<table>
<thead>
<tr>
<th>DIM2</th>
<th>Intermodulation distortion, difference. The relative magnitude and phase of the frequency component (F1 - F2).</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIM3</td>
<td>Intermodulation distortion, second difference. The relative magnitude and phase of the frequency component (2 ⋅ F1 - F2).</td>
</tr>
<tr>
<td>HD2</td>
<td>Second-order harmonic distortion. The relative magnitude and phase of the frequency component 2 ⋅ F1 (ignoring F2).</td>
</tr>
<tr>
<td>HD3</td>
<td>Third-order harmonic distortion. The relative magnitude and phase of the frequency component 3 ⋅ F1 (ignoring F2).</td>
</tr>
<tr>
<td>SIM2</td>
<td>Intermodulation distortion, sum. The relative magnitude and phase of the frequency component (F1 + F2).</td>
</tr>
</tbody>
</table>
The .DISTO summary report includes a set of distortion measures for each contributing component of every element, a summary set for each element, and a set of distortion measures representing a sum over all the elements in the circuit.

**Syntax**

```
.DISTO Rload <inter <skw2 <refpwr <spwf>>>>
```

where:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rload</strong></td>
<td>The resistor element name of the output load resistor into which the output power is fed</td>
</tr>
<tr>
<td><strong>inter</strong></td>
<td>Interval at which a distortion-measure summary is to be printed. Specifies a number of frequency points in the AC sweep (see the np parameter in “Using the .AC Statement”). If inter is omitted or set to zero, no summary printout is made. In this case, the distortion measures can be printed or plotted with the .PRINT or .PLOT statement. If inter is set to 1 or higher, a summary printout is made for the first frequency, and once for each inter frequency increment thereafter. To obtain a summary printout for only the first and last frequencies, set inter equal to the total number of increments needed to reach fstop in the .AC statement. For a summary printout of only the first frequency, set inter to greater than the total number of increments required to reach fstop.</td>
</tr>
<tr>
<td><strong>skw2</strong></td>
<td>Ratio of the second frequency F2 to the nominal analysis frequency F1. The acceptable range is 1e-3 &lt; skw2 ≤ 0.999. If skw2 is omitted, a value of 0.9 is assumed.</td>
</tr>
</tbody>
</table>
AC Sweep and Small Signal Analysis

Using Other AC Analysis Statements

Example

```
.DISTO RL 2 0.95 1.0E-3 0.75
```

Only one distortion analysis is performed per simulation. If more than one .DISTO statement is found, only the last is performed.

---

**Note:** The summary printout from the distortion analysis for each frequency listed is extensive. Use the “`inter`” parameter in the .DISTO statement to limit the amount of output generated.

---

**.NOISE Statement — AC Noise Analysis**

**Syntax**

```
.NOISE ovv srcnam inter
```

where:

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ovv</code></td>
<td>Nodal voltage output variable defining the node at which the noise is summed</td>
</tr>
<tr>
<td><code>srcnam</code></td>
<td>Name of the independent voltage or current source to be used as the noise input reference</td>
</tr>
<tr>
<td><code>inter</code></td>
<td>Interval at which a noise analysis summary is to be printed, inter specifies a number of frequency points summary in the AC sweep. If inter is omitted or set to zero, no summary printout is made. If inter is equal to or greater than one, a summary printout is made for the first frequency, and once for each inter frequency increment thereafter.</td>
</tr>
</tbody>
</table>

![Image](https://via.placeholder.com/150)
Example

.NOISE V(5) VIN 10

The .NOISE statement, used in conjunction with the AC statement, controls the noise analysis of the circuit.

Noise Calculations

The noise calculations in Star-Hspice are based on the complex AC nodal voltages, which in turn are based on the DC operating point. Noise models are described for each device type in the appropriate chapter in Volume II. A noise source is not assumed to be statistically correlated to the other noise sources in the circuit; each noise source is calculated independently. The total output noise voltage is the RMS sum of the individual noise contributions:

\[ \text{onoise} = \sum_{n=1}^{n} |Z_n \cdot I_n|^2 \]

where:

- \( \text{onoise} \) Total output noise
- \( I \) Equivalent current due to thermal noise, shot or flicker noise
- \( Z \) Equivalent transimpedance between noise source and the output
- \( n \) Number of noise sources associated with all resistors, MOSFETs, diodes, JFETs, and BJTs

The equivalent input noise voltage is the total output noise divided by the gain or transfer function of the circuit. The contribution of each noise generator in the circuit is printed for each inter frequency point. The output and input noise levels are normalized with respect to the square root of the noise bandwidth, and have the units volts/Hz\(^{1/2}\) or amps/Hz\(^{1/2}\).
You can simulate flicker noise sources in the noise analysis by including values for the parameters $K_F$ and $A_F$ on the appropriate device model statements.

Use the .PRINT or .PLOT statement to print or plot the output noise and the equivalent input noise.

You can only perform one noise analysis per simulation. If more than one NOISE statement is present, only the last one is performed.

**.SAMPLE Statement — Noise Folding Analysis**

For data acquisition of analog signals, data sampling noise often needs to be analyzed. This is accomplished with the .SAMPLE statement used in conjunction with the .NOISE and .AC statements.

The SAMPLE analysis causes Star-Hspice to perform a simple noise folding analysis at the output node.

**Syntax**

```
SAMPLE FS = freq <TOL = val> <NUMF = val> <MAXFLD = val>
+ <BETA = val>
```

where:

- $FS = freq$ Sample frequency, in Hertz
- $TOL$ Sampling error tolerance: the ratio of the noise power in the highest folding interval to the noise power in baseband. Default = 1.0e-3.
- $NUMF$ Maximum allowed number of user-specified frequencies. The algorithm requires approximately ten times this number of internally generated frequencies, so it should be kept small. Default = 100.
**.NET Statement - AC Network Analysis**

The .NET statement computes the parameters for the impedance matrix $Z$, the admittance matrix $Y$, the hybrid matrix $H$, and the scattering matrix $S$. The input impedance, output impedance, and admittance are also computed. This analysis is a part of the AC small-signal analysis. Therefore, network analysis requires the specification of the AC statement frequency sweep.

**Syntax**

One-port network:

```
.NET input <RIN = val>
```

or

```
.NET input <val >
```

Two-port network:

```
.NET output input <ROUT = val> <RIN = val>
```
AC Sweep and Small Signal Analysis

where:

\[\begin{align*}
\text{input} & \quad \text{AC input voltage or current source name} \\
\text{output} & \quad \text{Output port. It can be an output voltage, } V(n1,n2), \text{ or an output current, } I(\text{source}), \text{ or } I(\text{element}). \\
RIN & \quad \text{Input or source resistance keyword. The RIN value is used to calculate the output impedance and admittance, and also the scattering parameters. The RIN value defaults to 1 ohm.} \\
ROUT & \quad \text{Output or load resistance keyword. The ROUT value is used to calculate the input impedance and admittance, and also the scattering parameters. The ROUT value defaults to 1 ohm.}
\end{align*}\]

Example

One-port network:

\[\begin{align*}
\.NET & \quad \text{VINAC} \quad \text{RIN} = 50 \\
\.NET & \quad \text{IIN} \quad \text{RIN} = 50
\end{align*}\]

Two-port network:

\[\begin{align*}
\.NET & \quad V(10,30) \quad \text{VINAC} \quad \text{ROUT} = 75 \quad \text{RIN} = 50 \\
\.NET & \quad I(RX) \quad \text{VINAC} \quad \text{ROUT} = 75 \quad \text{RIN} = 50
\end{align*}\]

AC Network Analysis - Output Specification

Syntax

\[\begin{align*}
X_{ij}(z), \quad ZIN(z), \quad ZOUT(z), \quad YIN(z), \quad YOUT(z)
\end{align*}\]

where:

\[\begin{align*}
X & \quad \text{Specifies Z for impedance, Y for admittance, H for hybrid, and S for scattering} \\
ij & \quad i \text{ and } j \text{ can be 1 or 2. They identify which matrix parameter is to be printed.}
\end{align*}\]
Using Other AC Analysis Statements

AC Sweep and Small Signal Analysis

\[ z \]

Output type:
- **R**: real part
- **I**: imaginary part
- **M**: magnitude
- **P**: phase
- **DB**: decibel
- **T**: group time delay

**ZIN**
Input impedance. For the one port network, ZIN, Z11 and H11 are the same.

**ZOUT**
Output impedance

**YIN**
Input admittance. For the one port network, YIN and Y11 are the same.

**YOUT**
Output admittance

If “z” is omitted, output includes the magnitude of the output variable.

**Example**

```plaintext
.PRINT AC Z11(R) Z12(R) Y21(I) Y22 S11 S11(DB) Z11(T)
.PRINT AC ZIN(R) ZIN(I) YOUT(M) YOUT(P) H11(M) H11(T)
.PLOT AC S22(M) S22(P) S21(R) H21(P) H12(R) S22(T)
```

**Bandpass Netlist:**

Star-Hspice Network Analysis Results

```plaintext
*FILE: FBP_1.SP
.OPTIONS DCSTEP = 1 POST
*BAND PASS FILTER
C1 IN 2 3.166PF
L1 2 3 203NH
C2 3 0 3.76PF
C3 3 4 1.75PF
C4 4 0 9.1PF
L2 4 0 36.81NH
C5 4 5 1.07PF
C6 5 0 3.13PF
```

AC Sweep and Small Signal Analysis Using Other AC Analysis Statements

L3 5 6 233.17NH
C7 6 7 5.92PF
C8 7 0 4.51PF
C9 7 8 1.568PF
C10 8 0 8.866PF
L4 8 0 35.71NH
C11 8 9 2.06PF
C12 9 0 4.3PF
L5 9 10 200.97NH
C13 10 OUT 2.97PF
RX OUT 0 1E14
VIN IN 0 AC 1
.AC LIN 41 200MEG 300MEG
.NET V(OUT) VIN ROUT = 50 RIN = 50
.PLOT AC S11(DB) (-50,10) S11(P) (-180,180)
.PLOT AC ZIN(M) (5,130) ZIN(P) (-90,90)
.END

Figure 12-2: S11 Magnitude and Phase Plots
NETWORK Variable Specification

Star-Hspice uses the results of AC analysis to perform network analysis. The .NET statement defines the Z, Y, H, and S parameters to be calculated.

The following list shows various combinations of the .NET statement for network matrices that are initially calculated in Star-Hspice:

1) \[ \text{.NET Vout Isrc } V = [Z] [I] \]
2) \[ \text{.NET Iout Vsrc } I = [Y] [V] \]
3) \[ \text{.NET Iout Isrc } [V1 V2]^T = [H] [I1 V2]^T \]
4) \[ \text{.NET Vout Vsrc } [I1 V2]^T = [S] [V1 I2]^T \]

( \([M]^T\) represents the transpose of matrix M )
Note: The preceding list does not mean that combination (1) must be used for calculating the Z parameters. However, if .NET Vout Isrc is specified, Star-HSpice initially evaluates the Z matrix parameters and then uses standard conversion equations to determine the S parameters or any other requested parameters.

The example in Figure 12-4 shows the importance of the variables used in the .NET statement. Here, Isrc and Vce are the DC biases applied to the BJT.

Figure 12-4: Parameters with .NET V(2) Isrc

This .NET statement provides an incorrect result for the Z parameters calculation:

.NET V(2) Isrc

When Star-HSpice performs AC analysis, all the DC voltage sources are shorted and all the DC current sources are open-circuited. As a result, V(2) is shorted to ground, and its value is zero for AC analysis, directly affecting the results of the network analysis. When Star-HSpice attempts to calculate the Z parameters Z11 and Z21, defined as Z11 = V1/I1 and Z21 = V2/I1 with I2=0, the requirement...
that $I_2$ must be zero is not satisfied in the circuit above. Instead, $V_2$ is zero, which results in incorrect values for $Z_{11}$ and $Z_{21}$.

The correct biasing configurations for performing network analysis for $Z$, $Y$, $H$, and $S$ parameters are shown in Figure 12-5.

**Figure 12-5: Network Parameter Configurations**

As an example, the $H$ parameters are calculated by using the .NET statement.

.NET  $I(V_C)$  $I_B$
Here, \( V_C \) denotes the voltage at node C, the collector of the BJT. With this statement, Star-Hspice calculates the H parameters immediately after AC analysis. The H parameters are calculated by:

\[
V_1 = H_{11} \cdot I_1 + H_{12} \cdot V_2
\]
\[
I_2 = H_{21} \cdot I_1 + H_{22} \cdot V_2
\]

For Hybrid parameter calculations of \( H_{11} \) and \( H_{21} \), \( V_2 \) is set to zero (due to the DC voltage source \( V_{CE} \)), while for \( H_{12} \) and \( H_{22} \) calculations, \( I_1 \) is set to zero (due to the DC current source \( I_B \)). Setting \( I_1 \) and \( V_2 \) equal to zero precisely meets the conditions of the circuit under examination; namely, that the input current source is open circuited and the output voltage source is shorted to ground.

External DC biases applied to a BJT can be driven by a data file of measured results. In some cases, not all of the DC currents and voltages at input and output ports are available. When performing network analysis, examine the circuit and select suitable input and output variables to obtain correctly calculated results. The following examples demonstrate the network analysis of a BJT using Star-Hspice.

### Network Analysis Example: Bipolar Transistor

BJT network analysis

\[\text{option nopage list}\]
\[+\text{newtol reli = 1e-5 absi = 1e-10 relv = 1e-5 relvdc = 1e-7}\]
\[+\text{nomod post gmindc = 1e-12}\]
\[.op\]
\[.param vbe = 0 ib = 0 ic = 0 vce = 0\]

\$ H-parameter

\[.\text{NET i(vc) ibb rin = 50 rout = 50}\]
\[ve e 0 0\]
\[ibb 0 b dc = 'ib' ac = 0.1\]
\[vc c 0 'vce'\]
\[q1 c b e 0 bjt\]

\[.\text{model bjt npn subs = 1}\]
Using Other AC Analysis Statements

Using Other AC Analysis Statements

AC Sweep and Small Signal Analysis

+ \textbf{bf} = 1.292755e+02 \quad \textbf{br} = 8.379600e+00
+ \textbf{is} = 8.753000e-18 \quad \textbf{nf} = 9.710631e-01 \quad \textbf{nr} = 9.643484e-01
+ \textbf{ise} = 3.428000e-16 \quad \textbf{isc} = 1.855000e-17 \quad \textbf{iss} = 0.000000e+00
+ \textbf{ne} = 2.000000e+00 \quad \textbf{nc} = 9.460594e-01 \quad \textbf{ns} = 1.000000e+00
+ \textbf{vaf} = 4.942130e+01 \quad \textbf{var} = 4.589800e+00
+ \textbf{ikf} = 5.763400e-03 \quad \textbf{ikr} = 5.000000e-03 \quad \textbf{irb} = 8.002451e-07
+ \textbf{rc} = 1.216835e+02 \quad \textbf{rb} = 1.786930e+04 \quad \textbf{rbm} = 8.123460e+01
+ \textbf{re} = 2.136400e+00
+ \textbf{cje} = 9.894950e-14 \quad \textbf{mje} = 4.567345e-01 \quad \textbf{vje} = 1.090217e+00
+ \textbf{cjc} = 5.248670e-14 \quad \textbf{mjc} = 1.318637e-01 \quad \textbf{vjc} = 5.184017e-01
+ \textbf{xcjc} = 6.720303e-01
+ \textbf{cjs} = 9.671580e-14 \quad \textbf{mjs} = 2.395731e-01 \quad \textbf{vjs} = 5.000000e-01
+ \textbf{tf} = 3.319200e-11 \quad \textbf{itf} = 1.457110e-02 \quad \textbf{xtf} = 2.778660e+01
+ \textbf{vtf} = 1.157990e+00 \quad \textbf{ptf} = 6.000000e-05
+ \textbf{xti} = 4.460500e+00 \quad \textbf{xtb} = 1.456600e+00 \quad \textbf{eg} = 1.153300e+00
+ \textbf{tikf1} = -5.397800e-03 \quad \textbf{tirb1} = -1.071400e-03
+ \textbf{tre1} = -1.121900e-02 \quad \textbf{trb1} = 3.039900e-03
+ \textbf{trc1} = -4.020700e-03 \quad \textbf{trm1} = 0.000000e+00

.data bias

.vbe \hspace{1cm} \textbf{vce} \hspace{1cm} \textbf{ib} \hspace{1cm} \textbf{ic}
771.5648m \hspace{1cm} 292.5047m \hspace{1cm} 1.2330u \hspace{1cm} 126.9400u
797.2571m \hspace{1cm} 323.9037m \hspace{1cm} 2.6525u \hspace{1cm} 265.0100u
821.3907m \hspace{1cm} 848.7848m \hspace{1cm} 5.0275u \hspace{1cm} 486.9900u
843.5569m \hspace{1cm} 1.6596 \hspace{1cm} 8.4783u \hspace{1cm} 789.9700u
864.2217m \hspace{1cm} 2.4031 \hspace{1cm} 13.0750u \hspace{1cm} 1.1616m
884.3707m \hspace{1cm} 2.0850 \hspace{1cm} 19.0950u \hspace{1cm} 1.5675m

.enddata

.\textbf{end}

Other possible biasing configurations for the network analysis follow.
$S$-parameter

```
.NET v(c) vbb rin = 50 rout = 50
ve e 0  0
vbb b 0  dc = 'vbe' ac = 0.1
icc 0 c  'ic'
ql c b e 0 bjt
```

$Z$-parameter

```
.NET v(c) ibb rin = 50 rout = 50
ve e 0  0
ibb 0 b  dc = 'ib' ac = 0.1
icc 0 c  'ic'
ql c b e 0 bjt
```

$Y$-parameter

```
.NET i(vc) vbb rin = 50 rout = 50
ve e 0  0
vbb b 0  'vbe' ac = 0.1
vc c 0  'vce'
ql c b e 0 bjt
```

References

Chapter 13

Statistical Analysis and Optimization

An electrical circuit must be designed to the tolerances associated with the specific manufacturing process. The electrical yield refers to the number of parts that meet the electrical test specifications. Maximizing the yield is important for the overall process efficiency. Star-Hspice analyzes and optimizes the yield by using statistical techniques and observing the effects of element and model parameter variation.

- Specifying Analytical Model Types
- Simulating Circuit and Model Temperatures
- Performing Worst Case Analysis
- Performing Monte Carlo Analysis
- Worst Case and Monte Carlo Sweep Example
- Optimization
- Optimization Examples
Specifying Analytical Model Types

You can model parametric and statistical variation in circuit behavior in Star Hspice by using:

- The .PARAM statement. Use to investigate the performance of a circuit as specified changes in circuit parameters are made. See “Specifying Simulation Input and Controls” on page 3-1 for details of the .PARAM statement.

- Temperature Variation Analysis. The circuit and component temperatures are varied and the circuit responses compared. The temperature-dependent effects of the circuit can be studied in detail.

- Monte Carlo Analysis. The statistical standard deviations of component values are known. Use for centering a design for maximum process yield, or for determining component tolerances.

- Worst Case Corners Analysis. The component value limits are known. Automates quality assurance for basic circuit function for process extremes, quick estimation of speed and power trade-offs, and best case and worst case model selection through parameter corners library files.

- Data Driven Analysis. Use for cell characterization, response surface, or Taguchi analysis. See “Performing Cell Characterization” on page 24-1. Automates cell characterization, including timing simulator polynomial delay coefficient calculation. There is no limit on the number of parameters simultaneously varied or number of analyses to be performed. Convenient ASCII file format for automated parameter value generation. Can replace hundreds or thousands of Star-Hspice runs.

Yield analyses are used to modify DC operating points, DC sweeps, AC sweeps, and transient analysis. They can generate scatter plots for operating point analysis and family of curves plots for DC, AC, and transient analysis.

The .MEASURE statement in Star-Hspice is used with yield analyses, allowing you to see distributions of delay times, power, or any other characteristic described with a .MEASURE statement. Often, this is more useful than viewing a family of curves generated by a Monte Carlo analysis. When the .MEASURE statement is used, a table of results is generated as an .mt# file, which is in
readable ASCII format, and can be displayed in AvanWaves. Also, when .MEASURE statements are used in a Monte Carlo or data driven analysis, the Star-Hspice output file includes calculations for standard statistical descriptors:

\[
\text{Mean} \quad = \quad \frac{x_1 + x_2 + \ldots + x_n}{N}
\]

\[
\text{Variance} \quad = \quad \frac{(x_1 - \text{Mean})^2 + \ldots + (x_n - \text{Mean})}{N - 1}
\]

\[
\text{Sigma} \quad = \quad \sqrt{\text{Variance}}
\]

\[
\text{Average Deviation} \quad = \quad \frac{|x_1 - \text{Mean}| + \ldots + |x_n - \text{Mean}|}{N - 1}
\]
Simulating Circuit and Model Temperatures

Temperature affects all electrical circuits. The key temperature parameters associated with circuit simulation are (see Figure 13-1):

- Model reference temperature – each model might be measured at a different temperature and each model has a TREF parameter
- Element junction temperature – each resistor, transistor, or other element generates heat and will be hotter than the ambient temperature.
- Part temperature – at the system level, each part has its own temperature
- System temperature – a collection of parts form a system that has a local temperature.
- Ambient temperature – the ambient temperature is the air temperature of the system.

**Figure 13-1: Part Junction Temperature Sets System Performance**
Temperatures in Star-Hspice are calculated as differences from ambient temperature:

\[ T_{\text{ambient}} + \Delta_{\text{system}} + \Delta_{\text{part}} + \Delta_{\text{junction}} = T_{\text{junction}} \]

\[ I_{ds} = f(T_{\text{junction}}, T_{\text{model}}) \]

Every element in Star-Hspice has a keyword DTEMP. This is the difference between junction and ambient temperature. An example of using DTEMP in a MOSFET element statement is shown below.

```
M1 drain gate source bulk Model_name W=10u L=1u DTEMP=+20
```

**Temperature Analysis**

Star-Hspice allows you to specify three temperatures:

- Model reference temperature, specified in a .MODEL statement using the TREF parameter (or TEMP or TNOM, for some models). This is the temperature, in °C, at which the model parameters are measured and extracted. The value of TNOM can be set in a .OPTION statement. Its default value is 25 °C.

- Circuit temperature, specified using a .TEMP statement or the TEMP parameter. This is the temperature, in °C, at which all elements are simulated. To modify the temperature for a particular element, you can use the DTEMP parameter. The default circuit temperature is the value of TNOM.

- Individual element temperature, specified as the circuit temperature plus an optional amount specified using the DTEMP parameter

You can specify the temperature of a circuit for a Star-Hspice run with either the .TEMP statement or the TEMP parameter in the .DC, .AC, or .TRAN statements. The circuit simulation temperature set by any of these statements is compared against the reference temperature set by the TNOM option. TNOM defaults to 25 °C unless the option SPICE is used, in which case it defaults to 27 °C. The derating of component values and model parameters is calculated by using the difference of the circuit simulation temperature and the reference temperature, TNOM.
Since elements and models within a circuit can be operating at different temperatures (for example, a high-speed input/output buffer switching at 50 MHz will be much hotter than a low-drive NAND gate switching at 1 MHz), use an element temperature parameter, DTEMP, and a model reference parameter, TREF. Specifying DTEMP in an element statement causes the element temperature for the simulation to be:

\[
\text{element temperature} = \text{circuit temperature} + \text{DTEMP}
\]

Specify the DTEMP value in the element statement (resistor, capacitor, inductor, diode, BJT, JFET, or MOSFET statement). You can assign a parameter to DTEMP, then sweep the parameter using the .DC statement. The DTEMP value defaults to zero.

By specifying TREF in the model statement, the model reference temperature is changed (TREF overrides TNOM). The derating of the model parameters is based on the difference of the circuit simulator’s temperature and TREF, instead of TNOM.

**.TEMP Statement**

The syntax is:

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

\[t1 \ t2 \ ...\] The temperatures, in °C, at which the circuit is simulated

**Example**

```
.TEMP -55.0 25.0 125.0
```

The .TEMP statement sets the circuit temperatures for the entire circuit simulation. Star-Hspice uses the temperature set in the .TEMP statement along with the TNOM option setting (or the TREF model parameter) and the DTEMP element temperature, and simulates the circuit with individual elements or model temperatures.

```
.TEMP 100
D1 N1 N2 DMOD DTEMP=30
D2 NA NC DMOD
```
R1 NP NN 100 TC1=1 DTEMP=-30
.MODEL DMOD D IS=1E-15 VJ=0.6 CJA=1.2E-13 CJP=1.3E-14
+ TREF=60.0

From the .TEMP statement, the circuit simulation temperature is given as 100°C. Since TNOM is not specified, it defaults to 25°C. The temperature of the diode is given as 30°C above the circuit temperature by the DTEMP parameter. That is, D1temp = 100°C + 30°C = 130°C. The diode, D2, is simulated at 100°C. R1 is simulated at 70°C. Since TREF is specified at 60°C in the diode model statement, the diode model parameters given are derated by 70°C (130°C - 60°C) for diode D1 and by 40°C (100°C - 60°C) for diode D2. The value of R1 is derated by 45°C (70°C - TNOM).
Performing Worst Case Analysis

Worst case analysis often is used for design and analysis of MOS and BJT IC circuits. The worst case is simulated by taking all variables to their 2-sigma or 3-sigma worst case values. Since it is unlikely that several independent variables will attain their worst case values simultaneously, this technique tends to be overly pessimistic, and can lead to over-designing the circuit. However, it is useful as a fast check.

Model Skew Parameters

Avant! has extended the models in Star-Hspice to include physically measurable model parameters. The parameter variations allow the circuit simulator to predict the actual circuit response to the extremes of the manufacturing process. The physically measurable model parameters are called “skew” parameters because they are skewed from a statistical mean to obtain the predicted performance variations.

Examples of skew parameters are the difference between the drawn and physical dimension of metal, polysilicon, or active layers of an integrated circuit.

Generally, skew parameters are chosen independent of each other, so that combinations of skew parameters can be used to represent worst cases. Typical skew parameters for CMOS technology include:

- XL – polysilicon CD (critical dimension of poly layer representing the difference between drawn and actual size)
- XW_n, XW_p – active CD (critical dimension of active layer representing the difference between drawn and actual size)
- TOX – gate oxide thickness
- RSH_n, RSH_p – active layer resistivity
- DELVTO_n, DELVTO_p – threshold voltage variation

These parameters are allowed in any level MOS model in Star-Hspice. The DELVTO parameter simply shifts the threshold value. It is added to VTO for the Level 3 model and is added to or subtracted from VFB0 for the BSIM model.
Table 13-2 shows whether deviations are added to or subtracted from the average.

**Table 13-2: Sigma Deviations**

<table>
<thead>
<tr>
<th>Type</th>
<th>Param</th>
<th>Slow</th>
<th>Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMOS</td>
<td>XL</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>RSH</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DELVTO</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>TOX</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>XW</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>PMOS</td>
<td>XL</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>RSH</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DELVTO</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>TOX</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>XW</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

Skew parameters are chosen based on the available historical data collected either during fabrication or electrical test. For example, the poly CD skew parameter XL is collected during fabrication. This parameter is usually the most important skew parameter for a MOS process. Historical records produce data as shown in Figure 13-3.
Using Skew Parameters in Star-Hspice

The following example shows how to create a worst case corners library file for a CMOS process model. The physically measured parameter variations must be chosen so that their proper minimum and maximum values are consistent with measured current (IDS) variations. For example, a 3-sigma variation in IDS can be generated from a 2-sigma variation in the physically measured parameters.
The simulator accesses the models and skew through the .LIB library statement and the .INCLUDE include file statement. The library contains parameters that modify .MODEL statements. The following example of .LIB of model skew parameters features both worst case and statistical distribution data. The statistical distribution median value is the default for all non-Monte Carlo analysis.

**Example**

```
.LIB TT
$TYPICAL P-CHANNEL AND N-CHANNEL CMOS LIBRARY DATE:3/4/91
$ PROCESS: 1.0U CMOS, FAB22, STATISTICS COLLECTED 3/90-2/91
$ following distributions are 3 sigma ABSOLUTE GAUSSIAN

.PARAM
$ polysilicon Critical Dimensions
+ polycd=a gauss(0,0.06u,1) x1='polycd-sigma*0.06u'
$ Active layer Critical Dimensions
+ nactcd=a gauss(0,0.3u,1) xwn='nactcd+sigma*0.3u'
+ pactcd=a gauss(0,0.3u,1) xwp='pactcd+sigma*0.3u'
```
Performing Worst Case Analysis Statistical Analysis and Optimization

$ Gate Oxide Critical Dimensions (200 angstrom +/- 10a at 1 $ sigma)
$ toxcd=agauss(200,10,1) tox='toxcd-sigma*10'
$ Threshold voltage variation
+ vtoncd=agauss(0,0.05v,1) delvton='vtoncd-sigma*0.05'
+ vtopcd=agauss(0,0.05v,1) delvtop='vtopcd+sigma*0.05'

.INC '/usr/meta/lib/cmos1_mod.dat'$ model include file

.ENDL TT
.LIB FF
$HIGH GAIN P-CH AND N-CH CMOS LIBRARY 3SIGMA VALUES

.PARAM TOX=230 XL=-0.18u DELVTON=-.15V DELVTOP= 0.15V
.INC '/usr/meta/lib/cmos1_mod.dat'$ model include file

.ENDL FF

The model would be contained in the include file /usr/meta/lib/cmos1_mod.dat.

.MODEL NCH NMOS LEVEL=2 XL=XL TOX=TOX DELVTO=DELVTON ..... 
.MODEL PCH PMOS LEVEL=2 XL=XL TOX=TOX DELVTO=DELVTOP ..... 

Note: The model keyname (left-hand side) is being equated to the skew parameter (right-hand side). Model keynames and skew parameters can have the same names.

Skew File Interface to Device Models

The skew parameters are model parameters. They are used most often for transistor models, but they also apply to passive components. A typical device model set includes:

- MOSFET models for all device sizes using automatic model selector
- RC wire models for polysilicon, metal1, and metal2 layers (These models include temperature coefficients and fringing capacitance. They apply to drawn dimension)
- Single and distributed diode models for N+,P+, and well (includes temperature, leakage, and capacitance based on drawn dimension)
- BJT models for parasitic bipolar transistors, as well as any special BJTs such as a BiCMOS for ECL BJT process (includes current and capacitance as a function of temperature)
- Metal1 and metal2 transmission line models for long metal lines
- Models must be able to accept elements with sizes based on drawn dimension. A cell might be drawn at 2 µ dimension, shrink to 1 µ, with a physical size of 0.9 µ and an effective electrical size of 0.8 µ. The following dimension levels must be accounted for:
  - drawn size
  - shrunken size
  - physical size
  - electrical size

**Note:** Most simulator models go directly from drawn size to the electrical size. Star-Hspice is designed to support all four size levels for MOS models. Figure 13-5 shows the importance of the four size levels.

**Figure 13-5: Device Model from Drawn to Electrical Size**

![Diagram showing device model from drawn size to electrical size with labels for drawn, shrunken, physical, and electrical sizes.]
Performing Monte Carlo Analysis

Monte Carlo analysis uses a random number generator to create the following types of functions:

**Gaussian Parameter Distribution**
- Relative variation – variation is a ratio of average
- Absolute variation – variation is added to average
- Bimodal – nominal parameters are statistically reduced by multiplication of distribution

**Uniform Parameter Distribution**
- Relative variation – variation is a ratio of average
- Absolute variation – variation is added to average
- Bimodal – nominal parameters are statistically reduced by multiplication of distribution

**Random Limit Parameter Distribution**
- Absolute variation – variation is added to average
- Min or max variation is randomly selected

The number of times the operating point, DC sweep, AC sweep, or transient analysis is performed is determined by the value of the analysis keyword MONTE.

**Monte Carlo Setup**

To set up a Monte Carlo analysis, use the following Star-Hspice statements:
- `.PARAM` statement – sets a model or element parameter to a Gaussian, Uniform, or Limit function distribution.
- `.DC`, `.AC`, or `.TRAN` analysis – enable MONTE.
- `.MEASURE` statement – calculates output mean, variance, sigma, and standard deviation.
Analysis Syntax

Select the type of analysis desired, such as operating point, DC sweep, AC sweep, or TRAN sweep.

**Operating point:**

```
.DC MONTE=val
```

**DC sweep:**

```
.DC vin 1 5 .25 SWEEP MONTE=val
```

**AC sweep:**

```
.AC dec 10 100 10meg SWEEP MONTE=val
```

**TRAN sweep:**

```
.TRAN 1n 10n SWEEP MONTE=val
```

The value “val” represents the number of Monte Carlo iterations to be performed. A reasonable number is 30. The statistical significance of 30 iterations is quite high. If the circuit operates correctly for all 30 iterations, there is a 99% probability that over 80% of all possible component values operate correctly. The relative error of a quantity determined through Monte Carlo analysis is proportional to \( \text{val}^{-1/2} \).

Monte Carlo Output

Use .MEASURE statements as the most convenient way to summarize the results.

The .PRINT statement generates tabular results and prints all Monte Carlo parameter usage values. If one iteration is out of specification, obtain the component values from the tabular listing. A detailed resimulation of that iteration might help identify the problem.

.GRAPH generates a high-resolution plot for each iteration. In contrast, AvanWaves superimposes all iterations as a single plot and allows you to analyze each iteration individually.
.PARAM Distribution Function Syntax

A .PARAM parameter is assigned to the keywords of Star-Hspice elements and models. A distribution function is assigned to each .PARAM parameter. The distribution function is recalculated for each element or model keyword use of a parameter. This feature allows a parameterized schematic netlist to be used for Monte Carlo analysis with no additional modifications.

The syntax is:

```
.PARAM xx=UNIF(nominal_val, rel_variation <, multiplier>)
```
or

```
.PARAM xx=AUNIF(nominal_val, abs_variation <, multiplier>)
```
or

```
.PARAM xx=GAUSS(nominal_val, rel_variation, sigma <, + multiplier>)
```
or

```
.PARAM xx=AGAUSS(nominal_val, abs_variation, sigma <, + multiplier>)
```
or

```
.PARAM xx=LIMIT(nominal_val, abs_variation)
```

where:

- `xx` The parameter whose value is calculated by distribution function

- `UNIF` Uniform distribution function using relative variation

- `AUNIF` Uniform distribution function using absolute variation

- `GAUSS` Gaussian distribution function using relative variation

- `AGAUSS` Gaussian distribution function using absolute variation
**LIMIT**
Random limit distribution function using absolute variation,
+/- abs\_variation is added to nominal\_val based on whether the random outcome of a -1 to 1 distribution is greater or less than 0.

**nominal\_val**
Nominal value for Monte Carlo analysis and default value for all other analyses

**abs\_variation**
The AUNIF and AGAUSS vary the nominal\_val by +/- abs\_variation

**rel\_variation**
The UNIF and GAUSS vary the nominal\_val by +/- (nominal\_val \cdot rel\_variation)

**sigma**
The abs\_variation or rel\_variation is specified at the sigma level. For example, if sigma=3, then the standard deviation is abs\_variation divided by 3.

**multiplier**
If not specified, the default is 1. The calculation is repeated this many times and the largest deviation is saved. The resulting parameter value might be greater or less than nominal\_val. The resulting distribution is bimodal.

**Figure 13-6: Monte Carlo Distribution**

![Diagram showing Gaussian and Uniform distributions with nominal value, abs\_variation, and rel\_variation](image-url)
Monte Carlo Parameter Distribution Summary

A new random variable is calculated each time a parameter is used. If no Monte Carlo distribution is specified, then the nominal value is assumed. When a Monte Carlo distribution is specified for only one analysis, the nominal value is used for all other analyses.

You can assign a Monte Carlo distribution to all elements that share a common model. The actual element value will vary by the element distribution. A Monte Carlo distribution also can be assigned to a model keyword, and all elements that share that model use the same keyword value. This allows double element and model distributions to be created.

For example, the MOSFET channel length varies from transistor to transistor by a small amount corresponding to the die distribution. The die distribution is responsible for offset voltages in operational amplifiers and for the tendency of flip-flops to settle into random states. However, all transistors on a given die site will vary by the wafer or fabrication run distribution, which is much larger than the die distribution, but affects all transistors the same. The wafer distribution is assigned to the MOSFET model; it sets the speed and power dissipation characteristics.

Monte Carlo Examples

Gaussian, Uniform, and Limit Functions

Test of monte carlo gaussian, uniform, and limit functions

```
.options post
.dc monte=60
* setup plots
.model histo plot ymin=80 ymax=120 freq=1

.graph model=HISTO aunif_1=v(au1)
.graph model=HISTO aunif_10=v(au10)
.graph model=HISTO agauss_1=v(ag1)
.graph model=HISTO agauss_10=v(ag10)
.graph model=HISTO limit=v(L1)

* uniform distribution relative variation +/- .2
```
Performing Monte Carlo Analysis

.statistical analysis and optimization

.param ru_1 = unif(100, .2)
Iu1 u1 0 -1
ru1 u1 0 ru_1

* absolute uniform distribution absolute variation +/- 20
* single throw and 10 throw maximum
.param rau_1 = aunif(100, 20)
.param rau_10 = aunif(100, 20, 10)
Iau1 aul 0 -1
raul aul 0 rau_1
Iau10 aul0 0 -1
raul0 aul0 0 rau_10

* gaussian distribution relative variation +/- .2 at 3 sigma
.param rg_1 = gauss(100, .2, 3)
Ig1 g1 0 -1
rg1 g1 0 rg_1

* absolute gaussian distribution absolute variation +/- .2 at 3 sigma
* single throw and 10 throw maximum
.param rag_1 = agauss(100, 20, 3)
.param rag_10 = agauss(100, 20, 3, 10)
Iag1 ag1 0 -1
rag1 ag1 0 rag_1
Iag10 ag10 0 -1
rag10 ag10 0 rag_10

* random limit distribution absolute variation +/- 20
.param RL = limit(100, 20)
IL1 L1 0 -1
rL1 L1 0 RL
.end
**Figure 13-7: Gaussian Functions**

![Gaussian Functions](image1.png)

**Figure 13-8: Uniform Functions**

![Uniform Functions](image2.png)
Major and Minor Distribution

MOS IC processes have both a major and a minor statistical distribution of manufacturing tolerance parameters. The major distribution is the wafer-to-wafer and run-to-run variation. The minor distribution is the transistor-to-transistor process variation. The major distribution determines electrical yield. The minor distribution is responsible for critical second-order effects, such as amplifier offset voltage and flip-flop preference.
The example below is a Monte Carlo analysis of a DC sweep of the supply voltage VDD from 4.5 volts to 5.5 volts. Transistors M1 through M4 form two inverters. The channel lengths for the MOSFETs are set by the nominal value of the parameter LENGTH, which is set to 1u. Since all of the transistors are on the same integrated circuit die, the distribution is given by the parameter LEFF, which is a ±5% distribution in the variation of the channel lengths at the ±3-sigma level. Each MOSFET gets an independent random Gaussian value.

The parameter PHOTO controls the difference between the physical gate length and drawn gate length. Because both n-channel and p-channel transistors use the same layer for the gates, the Monte Carlo distribution XPHOTO is set to the local parameter PHOTO.

PHOTO lithography for both NMOS and PMOS devices is controlled by XPHOTO, which is consistent with the physics of manufacturing.

File: MONDC_A.SP

```
.TC VDD 4.5 5.5 .1 SWEEP MONTE=30
.PARAM LENGTH=1U LPHOTO=.1U
.PARAM LEFF=GAUSS (LENGTH, .05, 3)
+ XPHOTO=GAUSS (LPHOTO, .3, 3)
.PARAM PHOTO=XPHOTO

M1 1 2 GND GND  NCH  W=10U L=LEFF
M2 1 2 VDD VDD  PCH  W=20U L=LEFF
M3 2 3 GND GND  NCH  W=10U L=LEFF
M4 2 3 VDD VDD  PCH  W=20U L=LEFF

.MODEL NCH NMOS  LEVEL=2 UO=500 TOX=100 GAMMA=.7 VTO=.8
+ XL=PHOTO
.MODEL PCH PMOS  LEVEL=2 UO=250 TOX=100 GAMMA=.5 VTO=-.8
+ XL=PHOTO
.INC Model.dat
.END
```
RC Time Constant

This simple example demonstrates the uniform distribution for resistance and capacitance and the resulting transient waveforms for 10 different random values.

*FILE: MONI.SP WITH UNIFORM DISTRIBUTION
.OPTION LIST POST=2
.PARAM RX=UNIF(1, .5) CX=UNIF(1, .5)
.TRAN .1 1 SWEEP MONTE=10
.IC 1 1
R1 1 0 RX
C1 1 0 CX
.END

Figure 13-11: Monte Carlo Analysis of RC Time Constant
Switched Capacitor Filter Design

The capacitors used in switched capacitor filter applications are composed of parallel connections of a basic cell. Use Monte Carlo techniques to estimate the variation in total capacitance. There are two distributions involved in the capacitance calculation:

- Minor distribution of cell capacitance from cell-to-cell on a single die
- Major distribution of the capacitance from wafer-to-wafer or manufacturing run-to-run

The minor distribution is the element distribution, and the major distribution is the model distribution.

You can approach this problem from either physical or electrical levels. The physical level relies on physical distributions such as oxide thickness and polysilicon linewidth control. The electrical level relies on actual capacitor measurements.

Physical Approach

Assume that the variation in capacitance for adjacent cells is controlled by the local variation of polysilicon, since the oxide thickness control is excellent for small areas on a single wafer.

Next, define a local poly linewidth variation and a global or model level poly linewidth variation.
The local polysilicon linewidth control for a line 10 μ wide, manufactured with process A, is assumed to be ±0.02 μ for a 1-sigma distribution. The global or model level polysilicon linewidth control is much wider; use 0.1 μ for this example. The global oxide thickness is assumed to be 200 angstroms with a ±5 angstrom variation at 1 sigma.

The cap element is assumed to be square, with local poly variation in both directions. The cap model has two distributions, the poly linewidth distribution and the oxide thickness distribution. Since the effective length is

\[
L_{\text{eff}} = L_{\text{drawn}} - 2 \cdot \text{DEL}
\]

the model poly distribution is half the physical per-side values.

```
C1a 1 0 CMOD W=ELPOLY L=ELPOLY
C1b 1 0 CMOD W=ELPOLY L=ELPOLY
C1c 1 0 CMOD W=ELPOLY L=ELPOLY
C1d 1 0 CMOD W=ELPOLY L=ELPOLY
$ 10U \text{ POLYWIDTH, 0.05U}=1\text{SIGMA}
$ \text{ CAP MODEL USES 2*MODPOLY } .05u= 1 \text{ sigma}
$ 5\text{angstrom oxide thickness AT }1\text{SIGMA}
.\text{PARAM ELPOLY}=\text{AGAUSS}(10U,0.02U,1)
+ MODPOLY=\text{AGAUSS}(0,.05U,1)
+ POLYCAP=\text{AGAUSS}(200e-10,5e-10,1)
.\text{MODEL CMOD C THICK=POLYCAP DEL=MODPOLY}
```

**Electrical Approach**

The electrical approach assumes no physical interpretation, but requires a local or element distribution and a global or model distribution.

Assume that the capacitors can be matched to ±1% for the 2-sigma population. The process can maintain a ±10% variation from run to run for a 2-sigma distribution.

```
C1a 1 0 CMOD SCALE=ELCAP
C1b 1 0 CMOD SCALE=ELCAP
C1c 1 0 CMOD SCALE=ELCAP
C1d 1 0 CMOD SCALE=ELCAP
.\text{PARAM ELCAP=Gauss(1,.01,2) } $1\% \text{ at 2 sigma}
+ MODCAP=Gauss(.25p,.1,2) $10\% \text{ at 2 sigma}
.\text{MODEL CMOD C CAP=MODCAP}
```
Worst Case and Monte Carlo Sweep Example

The following example measures the delay of a pair of inverters. The input is buffered by an inverter, and the output is loaded by another inverter. The model was prepared according to the scheme described in the previous sections. The first .TRAN analysis statement sweeps from the worst case 3-sigma slow to 3-sigma fast. The second .TRAN does 100 Monte Carlo sweeps.

HSPICE Input File

The Star-Hspice input file can contain the following sections.

Analysis Setup Section

The simulation is accelerated by the use of the AUTOSTOP option, which automatically stops the simulation when the .MEASURE statements have achieved their target values.

```
$ inv.sp sweep mosfet -3 sigma to +3 sigma, then Monte Carlo
.option nopage nomod acct + autostop post=2
.tran 20p 1.0n sweep sigma -3 3 .5
.tran 20p 1.0n sweep monte=20
.option post co=132
.param vref=2.5
.meas m_delay trig v(2) val=vref fall=1 + targ v(out) val=vref fall=1
.meas m_power rms power to=m_delay
.param sigma=0
```

Circuit Netlist Section

```
.global 1
vcc 1 0 5.0
vin in 0 pwl 0,0 0.2n,5
x1 in 2 inv
x2 2 3 inv
x3 3 out inv
x4 out 5 inv
.macro inv in out
  mn out in 0 0 nch W=10u L=1u
  mp out in 1 1 pch W=10u L=1u
```

Skew Parameter Overlay for Model Section
* overlay of gaussian and algebraic for best case worst case and + monte carlo
* +/- 3 sigma is the maximum value for parameter sweep
.param
+ mult1=1
+ polycd=agauss(0,0.06u,1) xl='polycd-sigma*0.06u'
+ nactcd=agauss(0,0.3u,1) xwn='nactcd+sigma*0.3u'
+ pactcd=agauss(0,0.3u,1) xwp='pactcd+sigma*0.3u'
+ toxcd=agauss(200,10,1) tox='toxcd-sigma*10'
+ vtoncd=agauss(0,0.05v,1) delvton='vtoncd-sigma*0.05'
+ vtopcd=agauss(0,0.05v,1) delvtop='vtopcd+sigma*0.05'
+ rshncd=agauss(50,8,1) rshn='rshncd-sigma*8'
+ rsphcd=agauss(150,20,1) rsph='rsphcd-sigma*20'

MOS Model for N-Channel and P-Channel Transistors Section
* LEVEL=28 example model for high accuracy model
.model nch nmos
+ LEVEL=28
+ lmlt=mult1 wmlt=mult1 wref=22u lref=4.4u
+ xl=xl xw=xwn tox=tox delvto=delvton rsh=rshn
+ ld=0.06u wd=0.2u
+ acm=2 ldif=0 hdif=2.5u
+ rs=0 rd=0 rdc=0 rsc=0
+ js=3e-04 jsw=9e-10
+ cj=3e-04 mj=.5 pb=.8 cjsw=3e-10 mjsw=.3 php=.8 fc=.5
+ capop=4 xqc=.4 meto=0.08u
+ tlev=1 cta=0 ctp=0 tlevc=0 nlev=0
+ trs=1.6e-03 bex=-1.5 tcv=1.4e-03
* dc model
+ x2e=0 x3e=0 x2u1=0 x2ms=0 x2u0=0 x2m=0
+ vfb0=-.5 phi0=0.65 k1=.9 k2=.1 eta0=0
+ muz=500 u00=.075
+ x3ms=15 ul=.02 x3u1=0
+ b1=.28 b2=.22 x3m=0.00000e+00
+ alpha=1.5 vcr=20
+ n0=1.6 wfac=15 wfacu=0.25
+ lvfb=0 lk1=.025 lk2=.05
+ lalpha=5
.model pch pmos
+ LEVEL=28
+ lmlt=mult1 wmlt=mult1 wref=22u lref=4.4u
+ xl=xl xw=xwp tox=tox delvto=delvtop rsh=rshp
+ ld=0.08u wd=0.2u
+ acm=2 ldif=0 hdif=2.5u
+ rs=0 rd=0 rdc=0 rsc=0 rsh=rshp
+ js=3e-04 jsw=9e-10
+ cj=3e-04 mj=.5 pb=.8 cjsw=3e-10 mjsw=.3 php=.8 fc=.5
+ capop=4 xqc=.4 meto=0.08u
+ tlev=1 cta=0 ctp=0 tlevc=0 nlev=0
+ trs=1.6e-03 bex=-1.5 tcv=-1.7e-03
* dc model
+ x2e=0 x3e=0 x2u=0 x2ms=0 x2u=0 x2m=5
+ vfb0=-.1 phi0=0.65 k1=3.5 k2=0 eta0=0
+ muz=200 u00=.175
+ x3ms=8 ul=0 x3ul=0.0
+ b1=.25 b2=.25 x3m=0.0
+ alpha=0 vcr=20
+ n0=1.3 wfac=12.5 wfacu=.2
+ lvfb=0 lkl=-.05
.end

**Transient Sigma Sweep Results**

The plot in Figure 13-13 shows the family of transient analysis curves from the transient sweep of the sigma parameter from -3 to +3. Sigma is then algebraically coupled into the skew parameters and the resulting parameters modify the actual NMOS and PMOS models.
You can view the transient family of curves by plotting the .MEASURE output file. The plot in Figure 13-14 shows the measured pair delay and the total dissipative power against the parameter SIGMA.

**Figure 13-14: Sweep MOS Inverter, Pair Delay and Power: -3 Sigma to 3 Sigma**
Monte Carlo Results

The output of the Monte Carlo analysis is evaluated in this section. The plot in Figure 13-15 is a quality control step that plots TOX against XL (polysilicon critical dimension). The cloud of points was obtained in Avant!’s graphing software by setting XL as the X-axis independent variable and plotting TOX with a symbol frequency of 1. This has the effect of showing the points without any connecting lines. The resulting graph demonstrates that the TOX model parameter is randomly independent of XL.

**Figure 13-15: Scatter Plot, XL and TOX**

![Figure 13-15: Scatter Plot, XL and TOX]

The next graph (see Figure 13-16) is a standard scatter plot of the measured inverter pair delay against the Monte Carlo index number. If a particular result looks interesting, such as if the very smallest delay was obtained in simulation 68 (“monte carlo index = 68”), you can read the output listing file and obtain the actual Monte Carlo parameters associated with that simulation.

*** monte carlo index = 68 ***

MONTE CARLO PARAMETER DEFINITIONS
polycd: xl = -1.6245E-07
nactcd: xwn = 3.4997E-08
pactcd: xwp = 3.6255E-08
toxcd: tox = 191.0
vtoncd: delvton = -2.2821E-02
From the preceding listing, you can see that the \textit{m\_delay} value of 1.79e-10 seconds is the fastest pair delay. In addition, the Monte Carlo parameters can be examined.

\textbf{Figure 13-16: Scatter Plot of Inverter Pair Delay}

Plotting against the Monte Carlo index number does not help in centering the design. The first step in centering a design is to determine the most sensitive process variables. We can do this by graphing the various process parameters against the pair delay. Select the pair delay as the X-axis independent variable, and also set the symbol frequency to 1 to obtain the scatter plot. The graph in Figure 13-17 demonstrates the expected sensitivity of output pair delay to channel length variation (polysilicon variation).
Now, the parameter TOX is plotted against pair delay (Figure 13-18). Note that there is no clear tilt to the scatter plot. This indicates that TOX is a secondary process parameter compared to XL. To explore this in more detail, set the skew parameter XL to a constant and simulate.
The plot in Figure 13-19 shows the overlay of a 3-sigma worst case corners response and the 100 point Monte Carlo. Notice that the actual (Monte Carlo) distribution for power/delay is very different than the +3 sigma to -3 sigma plot. The worst case was simulated in 0.5 sigma steps. The actual response is closer to ± 1.5 sigma instead of ± 3 sigma. This produces a predicted delay variation of 100 ps instead of 200 ps. Therefore, the advantage of using Monte Carlo over traditional 3-sigma worst case corners is a 100% improvement in accuracy of simulated-to-actual distribution. This is an example of how the worst-case procedure is overly pessimistic.

**Figure 13-19: Superimpose Sigma Sweep over Monte Carlo**
Now take the Monte Carlo plot and superimpose the assumed part grades from marketing studies (Figure 13-20). In this case we have used a 250 ps delay and 7.5 mW power dissipation to determine the 4 binning grades. A manual count gives: Bin1 - 13%, Bin2 - 37%, Bin3 - 27%, Bin4 - 23%. If this circuit were representative of the entire chip, we would predict a present yield of 13% for the premium Bin 1 parts, assuming the design and process variations.

**Figure 13-20: Speed/Power Yield Estimation**
Optimization

Optimization, the automatic generation of model parameters and component values from a given set of electrical specifications or measured data, is available in Star-Hspice. With a user-defined optimization program and a known circuit topology, Star-Hspice automatically selects the design components and model parameters to meet DC, AC, and transient electrical specifications.

Star-Hspice optimization is the result of more than ten years of research in both the optimizing algorithms and user interface. The optimizing function is integrated into the core of Star-Hspice, resulting in optimum efficiency. The circuit result targets are part of the .MEASURE command structure, and the parameters to be optimized are Star-Hspice-defined parameter functions. A .MODEL statement sets up the optimization.

**Note:** Star-Hspice computes the .MEASURE statements using the postprocessing output. Reducing postprocessing output by setting option INTERP=1 may lead to interpolation errors in measurement results. See “Input and Output” on page 9-50 for more information on using these options.

The most powerful feature of the Star-Hspice approach is its incremental optimization technique. Incremental optimization allows you to solve the DC parameters first, then the AC parameters, and finally the transient parameters. A set of optimizer measurement functions not only makes the task of transistor optimization easy, but significantly improves cell and whole circuit optimization.

To perform optimization, create an input netlist file specifying:
- Minimum and maximum parameter and component limits
- The variable parameters and components
- An initial estimate of the selected parameter and component values
- The circuit performance goals or model-versus-data error function
Given the input netlist file, optimization specifications, component limits, and initial guess, the optimizer reiterates the circuit simulation until the target electrical specification is met or an optimized solution is found.

For improved optimization and simulation time and to increase the likelihood of a convergent solution, the initial estimate of the component values should produce a circuit with specifications near those of the original target. This reduces the number of times the optimizer reselects component values and resimulates the circuit.

**Optimization Control**

The length of time to complete an optimization is a function of the number of iterations allowed, the relative input tolerance, the output tolerance, and the gradient tolerance. The default values are satisfactory for most applications. Generally, 10 to 30 iterations are sufficient to get accurate optimizations.

**Simulation Accuracy**

Set the simulator with tighter convergence options than normal for optimization. The following options are suggested:

For DC MOS model optimizations:
- absmos=1e-8
- relmos=1e-5
- relv=1e-4

For DC JFET, BJT, and diode model optimizations:
- absi=1e-10
- reli=1e-5
- relv=1e-4

For transient optimizations:
- relv=1e-4
- relvar=1e-2
Curve Fit Optimization

Use optimization to curve fit user-defined DC, AC, or transient data. In a curve fit optimization, the desired numeric data for curves is stored in the data file as in-line data using the .DATA statement. The variable circuit components and parameter values of the netlist are specified in the .PARAM xxx=OPTxxx statement. The optimization analysis statements call the in-line data using the DATA= keyword. The .MEASURE statement uses the simulation result and compares it with the values given in the data file. The .MEASURE statement controls the comparison of simulation results to the values given in the data file. This is usually done with the ERR1 keyword. If the calculated value is not within the error tolerances specified in the optimization model, a new set of component values are selected and the circuit is resimulated. This is repeated until the closest fit to the curve is obtained, or the error tolerances set is satisfied.

Goal Optimization

Goal optimization differs from curve fit optimization in that it usually only applies to the optimization of a particular electrical specification, such as rise time or power dissipation.

Goal optimizations are specified using the GOAL keyword with a choice of relational operator in the .MEASURE statement, where GOAL is the target electrical specification being measured. This choice of relational operator is useful in multiple-constraint optimizations, when the absolute accuracy of some criteria is less important than for others.

Performing Timing Analysis

To analyze circuit timing violation, Star-Hspice uses a binary search algorithm to generate a set of operational parameters that produce a failure in the required behavior of the circuit. When a circuit timing failure occurs, you can identify a timing constraint that can lead to a design guideline. Typical types of timing constraint violations include:

- Data setup time before clock
- Data hold time after clock
Minimum pulse width required to allow a signal to propagate to the output

- Maximum toggle frequency of the component(s)

Bisection is a method of optimization that finds the value of an input variable (target value) associated with a goal value of an output variable. The input and output variables can be of various types (for example, voltage, current, delay time, or gain) related by some transfer function. You can use the bisection feature in a pass-fail mode or a bisection mode. The process is largely the same in each case.

**Understanding the Optimization Syntax**

Several Star-Hspice statements are required for optimization.

1. **.MODEL** modname OPT ...
2. **.PARAM** parameter=OPTxxx (init, min, max)
3. A .DC, .AC, or .TRAN analysis statement with MODEL=modname, OPTIMIZE=OPTxxx, and RESULTS=measurename
4. **.MEASURE** measurename ... <GOAL = | < | > val> – note that a space is required on either side of the relational operator =, <, or >

The .PARAM statement lets you specify initial, lower, and upper bound values. The types of .MEASURE statements available for optimization are described in “Specifying Simulation Output” on page 8-1.

Output statements .PRINT, .PLOT, and .GRAPH must be associated with the analysis statements .DC, .AC, or .TRAN. An analysis statement with the keyword OPTIMIZE is used for optimization only. To generate output for the optimized circuit, another analysis statement (.DC, .AC, or .TRAN) must be specified, along with the output statements. The proper specification order is:

1. Analysis statement with OPTIMIZE
2. .MEASURE statements specifying optimization goals or error functions
3. Ordinary analysis statement
4. Output statements
Analysis Statement (.DC, .TRAN, .AC)

The syntax is:

```
.DC <DATA=filename> SWEEP OPTIMIZE=OPTxxx RESULTS=ierr1 ... + ierrn MODEL=optmod
```
or

```
.AC <DATA=filename> SWEEP OPTIMIZE=OPTxxx RESULTS=ierr1 ... + ierrn MODEL=optmod
```
or

```
.TRAN <DATA=filename> SWEEP OPTIMIZE=OPTxxx RESULTS=ierr1 + ... ierrn MODEL=optmod
```

where:

- **DATA** Specifies the in-line file of parameter data to use in the optimization
- **OPTIMIZE** Indicates the analysis is for optimization. Specifies the parameter reference name used in the .PARAM optimization statement. All .PARAM optimization statements with the parameter reference name selected by OPTIMIZE will have their associated parameters varied during an optimization analysis.
- **MODEL** The optimization reference name that is also specified in the .MODEL optimization statement.
- **RESULTS** The measurement reference name that is also specified in the .MEASURE optimization statement. RESULTS is used to pass analysis data to the .MEASURE optimization statement.

.PARAM Statement

The syntax is:

```
.PARAM parameter=OPTxxx (initial_guess, low_limit, upper_limit)
```
or

```
.PARAM parameter=OPTxxx (initial_guess, low_limit, upper_limit,
```
+ \delta)

where:

\textit{OPTxxx} Specifies the optimization parameter reference name, referenced by the associated optimization analysis. This must agree with the OPTxxx name given in the analysis command associated with the keyname OPTIMIZE.

\textit{parameter} Specifies the parameter to be varied, the initial value estimate, the lower limit, and the upper limit allowed for the parameter. If the best solution does not exist within these constraints, the optimizer attempts to find the best solution.

\textit{delta} The final parameter value is the initial guess ± (n \cdot \delta). If delta is not specified, the final parameter value can be anything between \textit{low_limit} and \textit{upper_limit}. This is useful for optimizing transistor drawn widths and lengths, which must be quantized.

\textbf{Example}

\texttt{.PARAM vtx=OPT1(.7,.3,1.0) uox=OPT1(650,400,900)}

In the above example, the parameters \textit{uox} and \textit{vtx} are the variable model parameters to optimize a model for a given set of electrical specifications. The parameter \textit{vtx} is given an initial value estimate of 0.7 volts and can be varied within the limits of 0.3 and 1.0 volts for the optimization procedure. The optimization parameter reference name, OPT1, is used to reference the associated optimization analysis statement (not shown).

\textbf{.MODEL Statement}

For each optimization within a data file, specify a .MODEL statement to allow for more than one optimization per simulation run to be executed. The optimization .MODEL statement defines the convergence criteria, number of iterations, and derivative methods.
The syntax is:

```
.MODEL mname OPT <parameter=val ...>
```

You can specify the following OPT parameters in the .MODEL statement:

- **mname**
  - Model name. Elements refer to the model by this name.

- **CENDIF**
  - Represents the point at which more accurate derivatives are required. When the gradient of the RESULTS functions are less than CENDIF, the more time-consuming derivative methods are used. Values of 0.1 to 0.01 are suitable for most applications. If too large a value is used, the optimizer requires more CPU time. If too small a value is used, it might not find as accurate an answer. Default=1.0e-9.

- **CLOSE**
  - The initial estimate of how close the parameter initial value estimates are to the final solution. CLOSE multiplies the changes in the new parameter estimates. A large value for CLOSE causes the optimizer to take large steps toward the solution and a small value causes the optimizer to take smaller steps toward the solution. CLOSE should range from 0.01 for very close parameter estimates to 10 for rough initial guesses. Default=1.0.

  If CLOSE is greater than 100, the steepest descent part of the Levenburg-Marquardt algorithm dominates. For CLOSE less than 1, the Gauss-Newton method dominates. For further details, see L. Spruiell, “Optimization Error Surfaces,” *Meta-Software Journal*, Vol. 1, No. 4, December 1994.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CUT</strong></td>
<td>Modifies CLOSE, depending on how successful the iterations toward the solution become. If the last iteration was successful, descent toward the solution CLOSE is decreased by CUT. If the last iteration was not a successful descent to the solution, CLOSE is increased by CUT squared. CUT drives CLOSE up or down depending on the relative success in finding the solution. The CUT value must be greater than 1. Default = 2.0.</td>
</tr>
<tr>
<td><strong>DIFSIZ</strong></td>
<td>Determines the increment change in a parameter value for gradient calculations ($\Delta x = DIFSIZ \cdot \max(x, 0.1)$). If delta is specified in a .PARAM statement, then $\Delta x = \text{delta}$. Default = 1e-3.</td>
</tr>
<tr>
<td><strong>GRAD</strong></td>
<td>Represents a possible convergence when the gradient of the RESULTS function is less than GRAD. Values of 1e-6 to 1e-5 are suitable for most applications. If too large a value is used, the optimizer could stop before the best solution is found. Too small a value requires more iterations. Default=1.0e-6.</td>
</tr>
<tr>
<td><strong>ITROPT</strong></td>
<td>Sets the maximum number of iterations. Typically no more than 20-40 iterations are needed to find a solution. Too many iterations can imply the values for RELIN, GRAD, or RELOUT are too small. Default=20.</td>
</tr>
<tr>
<td><strong>LEVEL</strong></td>
<td>Selects the optimizing algorithm to use. Currently, the only option is LEVEL=1, a modified Levenburg-Marquardt algorithm.</td>
</tr>
<tr>
<td><strong>MAX</strong></td>
<td>Sets the upper limit on CLOSE. Values greater than 100 are recommended. Default=6000.</td>
</tr>
</tbody>
</table>
PARMIN

Allows better control of incremental parameter changes during error calculations. This produces more control over the trade-off between simulation time and optimization result accuracy. Star-Hspice calculates parameter increments using the relationship:

\[ \Delta \text{par}_\text{val} = \text{DIFSIZ} \cdot \text{MAX(\text{par}_\text{val}, \text{PARMIN})} \]

Default=0.1.

RELIN

Specifies the relative input parameter variation for convergence. If all the optimizing input parameters vary by no more than RELIN from one iteration to the next, then the solution is declared convergent. Since RELIN is a relative variance test, a value of 0.001 implies that the optimizing parameters are varying by less than 0.1% from one iteration to the next. Default=0.001.

RELOUT

Represents the relative output RESULTS function variance for convergence. For RELOUT=0.001, the difference in the RMS error of the RESULTS functions should vary less than 0.001. Default=0.001.
Optimization Examples

This section provides examples of the following types of Star-Hspice optimizations:

- MOS Level 3 Model DC Optimization
- MOS Level 13 Model DC Optimization
- RC Network Optimization
- CMOS Tristate Buffer Optimization
- BJT S Parameters Optimization
- BJT Model DC Optimization
- GaAsFET Model DC Optimization
- MOS Op-amp Optimization

MOS Level 3 Model DC Optimization

This example shows an optimization of I-V data to a Level 3 MOS model. The data consists of gate curves (\(i ds\) versus \(v gs\)) and drain curves (\(i ds\) versus \(v ds\)). The Level 3 parameters \(V TO\), \(G AMMA\), \(U O\), \(V MAX\), \(TH ETA\), and \(K APPA\) are optimized. After optimization, the model is compared separately to the data for the gate and drain curves. The option POST generates AvanWaves files for comparing the model to the data.

Level 3 Model DC Optimization Input Netlist File

```
$LEVEL 3 mosfet optimization
$.tighten the simulator convergence properties
.OPTION nomod post=2 newtol relmos=1e-5 absmos=1e-8
.MODEL optmod OPT itropt=30

Circuit Input
    vds  30 0  vds
    vgs  20 0  vgs
    vbs  40 0  vbs
    m1 30 20 0 40 nch w=50u l=4u
$.process skew parameters for this data
```
.PARAM xwn=-0.3u xln=-0.1u toxn=196.6 rshn=67
$.the model and initial guess
.MODEL nch NMOS LEVEL=3
  + acm=2 ldif=0 hdif=4u tlev=1 n=2
  + capop=4 meto=0.08u xqc=0.4
$.note capop=4 is ok for H8907 and later, otherwise use
$.Capop=2
$.fixed parameters
  + wd=0.15u ld=0.07u
  + js=1.5e-04 jsw=1.8e-09
  + cj=1.7e-04 cjsw=3.8e-10
  + nfs=2e11 xj=0.1u delta=0 eta=0
$.process skew parameters
  + tox=toxn rsh=rshn
  + xw=xwn xl=xln

Optimized Parameters
  + vto=vto gamma=gamma
  + uo=uo vmax=vmax theta=theta kappa=kappa
.PARAM
  + vto  = opt1(1,0.5,2)
  + gamma = opt1(0.8,0.1,2)
  + uo   = opt1(480,400,1000)
  + vmax = opt1(2e5,5e4,5e7)
  + theta = opt1(0.05,1e-3,1)
  + kappa = opt1(2,1e-2,5)

Optimization Sweeps
  .DC DATA=all optimize=opt1 results=comp1 model=optmod
  .MEAS DC comp1 ERR1 par(ids) i(m1) minval=1e-04 ignor=1e-05

DC Sweeps
  .DC DATA=gate
  .DC DATA=drain

Print Sweeps
  .PRINT DC vds=par(vds) vgs=par(vgs) im=i(m1) id=par(ids)
  .PRINT DC vds=par(vds) vgs=par(vgs) im=i(m1) id=par(ids)
**DC Sweep Data**

$.data
.PARAM vds=0 vgs=0 vbs=0 ids=0
.DATA all vds vgs vbs ids
1.000000e-01 1.000000e+00 0.000000e+00 1.655500e-05
5.000000e+00 5.000000e+00 0.000000e+00 4.861000e-03
.ENDATA
.DATA gate vds vgs vbs ids
1.000000e-01 1.000000e+00 0.000000e+00 1.655500e-05
1.000000e-01 5.000000e+00 -2.000000e+00 3.149500e-04
.ENDDATA
.DATA drain vds vgs vbs ids
2.500000e-01 2.000000e+00 0.000000e+00 2.302000e-04
5.000000e+00 5.000000e+00 0.000000e+00 4.861000e-03
.ENDDATA
.END

The Star-Hspice input netlist shows:

- Using .OPTIONS to tighten tolerances increases the accuracy of Star-Hspice. This is recommended for I-V optimization.
- “.MODEL optmod OPT itropt=30” limits the number of iterations to 30.
- The circuit is simply one transistor. VDS, VGS, VBS are parameter names that match names used in the data statements.
- The process variation parameters, XL, XW, TOX, RSH are specified as constants in a .PARAM statement. These are measured parameters for the device being characterized.
- The model contains references to parameters. In “GAMMA= GAMMA”, the left-hand side is a Level 3 model parameter name, while the right-hand side is a “.PARAM” parameter name.
- The long .PARAM statement specifies initial, min and max values for the optimized parameters. UO is initialized at 480 and kept within the range 400 to 1000 during optimization.
- The first .DC statement indicates that the data is in the in-line “.DATA all” block (which contains merged gate and drain curve data), optimization of parameters that were declared as OPT1 (in this case all of the optimized parameters), error function COMPI (matches the name of a .MEASURE statement), and model OPTMOD (sets iteration limit).
The .MEASURE statement specifies least-squares relative error. The difference between data par(ids) and model i(m1) is divided by either the absolute value of par(ids), or by minval=1e-6, whichever is larger. Using minval keeps low current data from dominating the error.

The remaining .DC and .PRINT statements are for printback after optimization. You can place them anywhere in the netlist input file because they will be correctly assigned when the file is parsed.

The “.PARAM VDS=0 VGS=0 VBS=0 IDS=0” statements simply declare these data column names as parameters.

The .DATA statements give data for IDS versus VDS, VGS, VBS. The selection of data should match the choice of model parameters to optimize. To optimize GAMMA, data with back bias must be provided (VBS= -2 in this case). To optimize KAPPA, the saturation region must have data. In this example, the data set “all” contain:

- gate curves: vds=0.1 vbs=0,-2 vgs=1 to 5 in steps 0.25
- drain curves: vbs=0 vgs=2,3,4,5 vds=0.25 to 5 in steps 0.25

The results are shown in Figure 13-21.
This example shows an optimization of I-V data to a Level 13 MOS model. The data consists of gate curves ($i_d$ versus $v_{gs}$) and drain curves ($i_d$ versus $v_{ds}$). This example demonstrates two-stage optimization. The Level 13 parameters $v_{fb0}$, $k_1$, $\mu_z$, $x_2m$, and $u_{00}$ are optimized to the gate data. Then the Level 13 parameters MUS, X3MS, $U_1$, and the impact ionization parameter $\alpha$ are optimized to the drain data. After optimization, the model is compared to the data. The option POST generates AvanWaves files for comparing the model to the data.

The results are shown in Figure 13-22.

**MOS Level 13 Model DC Optimization Input Netlist File**

```
$LEVEL 13 mosfet optimization
$..tighten the simulator convergence properties
.OPTION nomod post=2
+ newtol relmos=1e-5 absmos=1e-8
.MODEL optmod OPT itropt=30
```
Circuit Input

vds  30 0  vds
vgs  20 0  vgs
vbs  40 0  vbs
m1 30 20 0 40 nch w=50u l=4u

$. process skew parameters for this data
.PARAM xwn=-0.3u xln=-0.1u toxn=196.6 rshn=67
$. the model and initial guess
.MODEL nch NMOS LEVEL=13
+ acm=2 ldif=0 hdif=4u tlev=1 n=2 capop=4 meto=0.08u xqc=0.4
$. parameters obtained from measurements
+ wd=0.15u ld=0.07u js=1.5e-04 jsw=1.8e-09
+ cj=1.7e-04 cjsw=3.8e-10
$. parameters not used for this data
+ k2=0 eta0=0 x2e=0 x3e=0 x2u1=0 x2ms=0 x2u0=0 x3u1=0
$. process skew parameters
+ toxm=toxn rsh=rshn
+ xw=xwn xl=xln
$. optimized parameters
+ vfb0=vfb0 k1=k1 x2m=x2m muz=muz u00=u00
+ mus=mus x3ms=x3ms ul=ul
$. impact ionization parameters
+ alpha=alpha vcr=15

.PARAM
+ vfb0    = opt1(-0.5, -2, 1)
+ k1      = opt1(0.6,0.3,1)
+ muz     = opt1(600,300,1500)
+ x2m     = opt1(0,-10,10)
+ u00     = opt1(0.1,0,0.5)
+ mus     = opt2(700,300,1500)
+ x3ms    = opt2(5,0,50)
+ ul      = opt2(0.1,0,1)
+ alpha   = opt2(1,1,3,10)

Optimization Sweeps

.DC DATA=gate optimize=opt1 results=comp1 model=optmod
.MEAS DC comp1 ERR1 par(ids) i(m1) minval=1e-04 ignor=1e-05
.DC DATA=drain optimize=opt2 results=comp2 model=optmod
.MEAS DC comp2 ERR1 par(ids) i(m1) minval=1e-04 ignor=1e-05
DC Data Sweeps

```plaintext
.DC DATA=gate
.DC DATA=drain
```

Print Sweeps

```plaintext
.PRINT DC vds=par(vds) vgs=par(vgs) im=i(m1) id=par(ids)
.PRINT DC vds=par(vds) vgs=par(vgs) im=i(m1) id=par(ids)
```

DC Sweep Data

```plaintext
$.data
.PARAM vds=0 vgs=0 vbs=0 ids=0
.DATA gate vds vgs vbs ids
1.000000e-01 1.000000e+00 0.000000e+00 1.655500e-05
1.000000e-01 5.000000e+00 -2.000000e+00 3.149500e-04
.ENDDATA
.DATA drain vds vgs vbs ids
2.500000e-01 2.000000e+00 0.000000e+00 2.809000e-04
5.000000e+00 5.000000e+00 0.000000e+00 4.861000e-03
.ENDDATA
.END
```

**Figure 13-22: Level 13 MOSFET Optimization**
RC Network Optimization

Following is an example of optimizing the power dissipation and time constant of an RC network. The circuit is a parallel resistor and capacitor. The following design targets are specified.

- 1 s time constant
- 50 mW rms power dissipation through the resistor

The Star-Hspice strategy is as follows:

- .MEASURE statement RC1 calculates RC time constant (GOAL of .3679 V corresponds to 1 s time constant $e^{-\tau_c}$).
- .MEASURE statement RC2 calculates the rms power where the GOAL is 50 mW.
- OPTrc identifies RX and CX as optimization parameters and sets their starting, minimum, and maximum values.

Star-Hspice features used:

- Measure voltages and report times subject to goal
- Measure device power dissipation subject to goal
- Measure statements replace tabular or plot output
- Element value parameterization
- Parameter optimization function
- Transient with SWEEP optimize

RC Network Optimization Input Netlist File

```markdown
.title RCOPT.sp
.option post

.PARAM RX=OPTRC(.5, 1E-2, 1E+2)
.PARAM CX=OPTRC(.5, 1E-2, 1E+2)

.MEASURE TRAN RC1 TRIG AT=0 TARG V(1) VAL=.3679 FALL=1 + GOAL=1sec
.MEASURE TRAN RC2 RMS P(R1) GOAL=50mwatts
```
.MODEL OPT1 OPT

.tran .1 2 $ initial values
.tran .1 2 SWEEP OPTIMIZE=OPTrc RESULTS=RC1,RC2 MODEL=OPT1
.tran .1 2 $ analysis using final optimized values

.ic 1 1
R1 1 0 RX
c1 1 0 CX

**Optimization Results**

RESIDUAL SUM OF SQUARES = 1.323651E-06  
NORM OF THE GRADIENT = 6.343728E-03  
MARQUARDT SCALING PARAMETER = 2.799235E-06  
NO. OF FUNCTION EVALUATIONS = 24  
NO. OF ITERATIONS = 12

**Residual Sum of Squares**

The residual sum of squares is a measure of the total error. The smaller this value is, the more accurate the optimization results are.

\[
\text{residual sum of squares} = \sum_{i=1}^{ne} E_i^2
\]

where \(E\) is the error function and \(ne\) is the number of error functions.
**Norm of the Gradient**

The norm of the gradient is another measure of the total error. The smaller this value is, the more accurate the optimization results are.

The gradient $G$ is found by

$$G_j = \sum_{i=1}^{ne} E_i \cdot (\Delta E_i / \Delta P_j)$$

and

$$\text{norm of the gradient} = 2 \cdot \sqrt{\sum_{i=1}^{np} G_j^2}$$

where $P$ is the parameter and $np$ is the number of parameters to be optimized.

**Marquardt Scaling Parameter**

This parameter is used in the Levenburg-Marquardt algorithm to find the actual solution of the optimizing parameters. The search direction is a combination of the Steepest Descent method and the Gauss-Newton method.

The Steepest Descent method is used initially to approach the solution because it is fast, and then the Gauss-Newton method is used to find the solution. During this process, the Marquardt Scaling Parameter becomes very small, but starts to increase again if the solution starts to deviate. If this happens, the optimizer chooses between the two methods to work toward the solution again.

If the optimal solution is not attained, an error message is printed and a large Marquardt Scaling Parameter value is printed.

**Number of Function Evaluations**

This is the number of analyses (for example, finite difference or central difference) that were needed to find a minimum of the function.
Number of Iterations
This is the number of iterations needed to find the optimized or actual solution.

Optimized Parameters OPTRC

| * | PARAM RX | 6.7937 $ 54.5260 50.2976M |
|   | PARAM CX | 147.3697M $ 45.4740 33.7653M |

Figure 13-23: Power Dissipation and Time Constant
(VOLT) RCOPT.TR0 = Before Optimization, RCOPT.TR1 = Optimized Result
CMOS Tristate Buffer Optimization

The example circuit is an inverting CMOS tristate buffer. The following design targets are specified:

1. Rising edge delay of 5 ns (input 50% voltage to output 50% voltage)
2. Falling edge delay of 5 ns (input 50% voltage to output 50% voltage)
3. RMS power dissipation should be as low as possible
4. Output load consists of
   - pad capacitance
   - leadframe inductance
   - 50 pF capacitive load
The Star-Hspice strategy is as follows:

- Simultaneously optimize rising delay buffer and falling delay buffer.
- Set up internal power supplies and tristate enable as global nodes.
- Optimize all device widths except:
  - Initial inverter (this is assumed to be standard size)
  - Tristate inverter and part of tristate control (the optimization is not sensitive to this path)
- Perform initial transient analysis for plotting purposes, then optimize and perform a final transient for plotting purposes.
- Use a weighted RMS power measure by specifying an unrealistically low power goal and using MINVAL to attenuate the error.

**CMOS Tristate Buffer Optimization Input Netlist File**

```plaintext
*Tri-State input/output Optimization
.options defl=1.2u nomod post=2
+ relv=1e-3 absvar=.5 relvar=.01

Circuit Input
.global lgnd lvcc enb
.macro buff data out
    mp1 DATAN DATA LVCC LVCC p w=35u
    mn1 DATAN DATA LGND LGND n w=17u
    mp2 BUS DATAN LVCC LVCC p w=wp2
    mn2 BUS DATAN LGND LGND n w=wn2
    mp3 PEN PENN LVCC LVCC p w=wp3
    mn3 PEN PENN LGND LGND n w=wn3
    mp4 NEN NENN LVCC LVCC p w=wp4
    mn4 NEN NENN LGND LGND n w=wn4
    mp5 OUT PEN LVCC LVCC p w=wp5 l=1.8u
    mn5 OUT NEN LGND LGND n w=wn5 l=1.8u
    mp10 NENN BUS LVCC LVCC p w=wp10
    mn12 PENN ENB NENN LGND n w=wn10
```
mn10 PENN BUS LGND LGND n w=wn10
mp11 NENN ENB LVCC LVCC p w=wp11
mp12 NENN ENBN PENN LVCC p w=wp11
mn11 PENN ENBN LGND LGND n w=80u

mp13 ENBN ENB LVCC LVCC p w=35u
mn13 ENBN ENB LGND LGND n w=17u
cbus BUS LGND 1.5pf
cpad OUT LGND 5.0pf
.ends
* * input signals *
.vcc VCC GND 5V
.lvcc vcc lvcc 6nh
.lgnd lgnd gnd 6nh
.vin DATA LGND pl (0v 0n, 5v 0.7n)
vinnb DATAbar LGND pl (5v 0n, 0v 0.7n)
.ven ENB GND 5V

** circuit **
x1 data out buff
cext1 out GND 50pf
x2 databar outbar buff
cext2 outbar GND 50pf

Optimization Parameters
.param
+ wp2=opt1(70u,30u,330u)
+ wn2=opt1(22u,15u,400u)
+ wp3=opt1(400u,100u,500u)
+ wn3=opt1(190u,80u,580u)
+ wp4=opt1(670u,150u,800u)
+ wn4=opt1(370u,50u,500u)
+ wp5=opt1(1200u,1000u,5000u)
+ wn5=opt1(600u,400u,2500u)
+ wp10=opt1(240u,150u,450u)
+ wn10=opt1(140u,30u,280u)
+ wp11=opt1(240u,150u,450u)
**Control Section**

```
.tran 1ns 15ns
.tran .5ns 15ns sweep optimize=opt1 results=tfopt,tropt,rmspowo
model=optmod
** put soft limit for power with minval setting (i.e. values
** less than 1000mw are less important)
.measure rmspowo rms power goal=100mw minval=1000mw
.meas tran tfopt trig v(data) val=2.5 rise=1 targ v(out)
 + val=2.5 fall=1 goal 5.0n
.meas tran tropt trig v(databar) val=2.5 fall=1 targ
 + v(outbar) val=2.5 rise=1 goal 5.0n

.model optmod opt itropt=30  max=1e+5
```

```
.tran 1ns 15ns
* output section *
*.plot tran v(data) v(out)
*.plot tran v(databar) v(outbar)
```

**Model Section**

```
.MODEL N NMOS LEVEL=3 VTO=0.7 UO=500 KAPPA=.25 KP=30U ETA=.03
 + THETA=.04 VMAX=2E5 NSUB=9E16 TOX=500E-10 GAMMA=1.5 PB=0.6
 + JS=.1M XJ=0.5U LD=0.0 NFS=1E11 NSS=2E10 CGSO=200P CGDO=200P
 + CGBO=300P
.MODEL P PMOS LEVEL=3 VTO=-0.8 UO=150 KAPPA=.25 KP=15U
 + ETA=.03 THETA=.04 VMAX=5E4 NSUB=1.8E16 TOX=500E-10 NFS=1E11
 + GAMMA=.672 PB=0.6 JS=.1M XJ=0.5U LD=0.0
 + NSS=2E10 CGSO=200P CGDO=200P CGBO=300P
.end
```

**Optimization Results**

```
residual sum of squares     = 2.388803E-02
norm of the gradient        = 0.769765
marquardt scaling parameter = 12624.2
no. of function evaluations = 175
no. of iterations           = 23
```

**Optimization Completed**

Parameters relin= 1.0000E-03 on last iterations
Optimized Parameters OPT1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>%norm</th>
<th>%change</th>
</tr>
</thead>
<tbody>
<tr>
<td>.param wp2           = 84.4981u</td>
<td>$22.5877</td>
<td>-989.3733u</td>
<td></td>
</tr>
<tr>
<td>.param wn2           = 34.1401u</td>
<td>$7.6568</td>
<td>-659.2874u</td>
<td></td>
</tr>
<tr>
<td>.param wp3           = 161.7354u</td>
<td>$730.7865m</td>
<td>-351.7833u</td>
<td></td>
</tr>
<tr>
<td>.param wn3           = 248.6829u</td>
<td>$8.1362</td>
<td>-351.7833u</td>
<td></td>
</tr>
<tr>
<td>.param wp4           = 238.9825u</td>
<td>$1.2798</td>
<td>43.5213m</td>
<td></td>
</tr>
<tr>
<td>.param wn4           = 61.3509u</td>
<td>$315.4656m</td>
<td>43.5213m</td>
<td></td>
</tr>
<tr>
<td>.param wp5           = 1.7753m</td>
<td>$4.1713</td>
<td>2.1652m</td>
<td></td>
</tr>
<tr>
<td>.param wn5           = 1.0238m</td>
<td>$5.8506</td>
<td>413.9667u</td>
<td></td>
</tr>
<tr>
<td>.param wp10          = 268.3125u</td>
<td>$8.1917</td>
<td>-2.0266m</td>
<td></td>
</tr>
<tr>
<td>.param wn10          = 115.6907u</td>
<td>$40.5975</td>
<td>-422.8411u</td>
<td></td>
</tr>
<tr>
<td>.param wp11          = 153.1344u</td>
<td>$482.0655m</td>
<td>-30.6813m</td>
<td></td>
</tr>
</tbody>
</table>

*** optimize results measure names and values
* tfopt           = 5.2056n
* tropt           = 5.5513n
* rmspowo         = 200.1808m

Figure 13-25: Tristate Buffer Optimization Circuit
BJT S Parameters Optimization

In the following example, the S parameters are optimized to match those given for a set of measurements. These measured S parameters, as a function of frequency, are in the “.DATA MEASURED” in-line data statement. The model parameters of the microwave transistor (LBB, LCC, LEE, TF, CBE, CBC, RB, RE, RC, and IS) are varied so that measured S parameters in the .DATA statement matches the calculated S parameters from the simulation results.

This optimization uses a 2n6604 microwave transistor and an equivalent circuit consisting of a BJT with parasitic resistances and inductances. The BJT is biased at a 10 mA collector current (0.1 mA base current at DC bias and bf=100).

Key Star-Hspice Features Used

- NET command to simulate network analyzer action
- AC optimization
- Optimization of element and model parameters
- Optimization comparing measured S parameters versus calculated parameters
- S parameters used in magnitude and phase (real and imaginary available)
- Data-driven frequency versus S parameter table weighting used for phase domain

**BJT S Parameters Optimization Input Netlist File**

* BJTOPT.SP BJT S PARAMETER OPTIMIZATION
.OPTION ACCT NOMOD POST=2

**BJT Equivalent Circuit Input**

* THE NET COMMAND IS AUTOMATICALLY REVERSING THE SIGN OF
* THE POWER SUPPLY CURRENT FOR THE NETWORK CALCULATIONS
.NET I(VCE) IBASE ROUT=50 RIN=50
VCE VCE 0 10V
IBASE 0 IIN AC=1 DC=.1MA
LBB IIN BASE LBB
LCC VCE COLLECT LCC
LEE EMIT 0 LEE
Q1 COLLECT BASE EMIT T2N6604
.MODEL T2N6604 NPN RB=RB BF=100 TF=TF CJE=CBE CJC=CBC
+ RE=RE RC=RC IS=IS
.PARAM
+ LBB= OPT1(100P, 1P, 10N)
+ LCC= OPT1(100P, 1P, 10N)
+ LEE= OPT1(100P, 1P, 10N)
+ TF = OPT1(1N, 5P, 5N)
+ CBE= OPT1(.5P, .1P, 5P)
+ CBC= OPT1(.4P, .1P, 5P)
+ RB= OPT1(10, 1, 300)
+ RE= OPT1(.4, .01, 5)
+ RC= OPT1(10, .1, 100)
+ IS= OPT1(1E-15, 1E-16, 1E-10)
.AC DATA=MEASURED OPTIMIZE=OPT1
+ RESULTS=COMP1,COMP3,COMP5,COMP6,COMP7
+ MODEL=CONVERGE
.MODEL CONVERGE OPT RELIN=1E-4 RELOUT=1E-4 CLOSE=100 ITROPT=25
.MEASURE AC COMP1 ERR1 PAR(S11M) S11(M)
.MEASURE AC COMP2 ERR1 PAR(S11P) S11(P) MINVAL=10
Optimization Examples

Optimization Results

RESIDUAL SUM OF SQUARES = 5.142639e-02
NORM OF THE GRADIENT = 6.068882e-02
MARQUARDT SCALING PARAMETER = 0.340303
CO. OF FUNCTION EVALUATIONS = 170
NO. OF ITERATIONS = 35

The maximum number of iterations (25) was exceeded. However, the results probably are accurate. Increase ITROPT accordingly.

Optimized Parameters OPT1– Final Values

***OPTIMIZED PARAMETERS OPT1 SENS %NORM-SEN
.PARAM LBB = 1.5834N $ 27.3566X 2.4368
.PARAM LCC = 2.1334N $ 12.5835X 1.5138
.PARAM LEE =723.0995P $254.2312X 12.3262
.PARAM TF = 12.7611P $ 7.4344G 10.0532
.PARAM CBE =620.5195F $ 23.0855G 1.5300
.PARAM CBC = 1.0263P $346.0167G 44.5016
BJT Model DC Optimization

The goal is to match the forward and reverse Gummel plots obtained from a HP4145 semiconductor analyzer with the Star-Hspice LEVEL=1 Gummel-Poon BJT model. Since the Gummel plots are at low base currents, the base resistance is not optimized. The forward and reverse Early voltages (VAF and VAR) are not optimized, since no VCE data was measured.

The key feature used in this optimization is incremental optimization. First the forward Gummel data points are optimized. The forward optimized parameters are updated into the model and not allowed to change. Then the reverse Gummel data points are optimized.
BJT Model DC Optimization Input Netlist File

* FILE OPT_BJT.SP BJT OPTIMIZATION T2N3947
* OPTIMIZE THE DC FORWARD AND REVERSE CHARACTERISTICS FROM A
* GUMMEL PLOT
* ALL DC GUMMEL-POON DC PARAMETERS EXCEPT BASE RESISTANCE AND
* EARLY VOLTAGES OPTIMIZED
*
$.TIGHTEN THE SIMULATOR CONVERGENCE PROPERTIES
.OPTION NOMOD INSGOLD=2 NOPAGE VTOL=1E-10 POST
+ NUMDGT=5 RELI=1E-4 RELV=1E-4
$.OPTIMIZATION CONVERGENCE CONTROLS
.MODEL OPTMOD OPT RELIN=1E-4 ITROPT=30 GRAD=1E-5 CLOSE=10
+ CUT=2 CENDIF=1E-6 RELOUT=1E-4 MAX=1E6

Room Temp Device
VBER BASE 0 VBE
VBCR BASE COL VBC
Q1 COL BASE 0 BJTMOD

Model and Initial Estimates

.MODEL BJTMOD NPN
+ ISS = 0. XTF = 1. NS = 1.
+ CJS = 0. VJS = 0.50000 PTF = 0.
+ MJS = 0. EG = 1.10000 AF = 1.
+ ITF = 0.50000 VTF = 1.00000
+ FC = 0.95000 XCJC = 0.94836
+ SUBS = 1
+ TF=0.0 TR=0.0 CJE=0.0 CJC=0.0 MJJE=0.5 MJJC=0.5 VJE=0.6
+ VJC=0.6 RB=0.3 RC=10 VAF=550 VAR=300
$.THESE ARE THE OPTIMIZED PARAMETERS
+ BF=BF IS=IS IKF=IKF ISE=ISE RE=SE
+ NF=NF NE=NE
$.THESE ARE FOR REVERSE BASE OPT
+ BR=BR IKR=IKR ISC=ISC
+ NR=NR NC=NC

.PARAM VBE=0 IB=0 IC=0 VCE_EMIT=0 VBC=0 IB_EMIT=0 IC_EMIT=0
+ BF= OPT1(100, 50, 350)
+ IS= OPT1(5E-15, 5E-16, 1E-13)
+ NF= OPT1(1.0, 0.9, 1.1)
+ IKF=OPT1(50E-3, 1E-3, 1)
+ RE= OPT1( 10, 0.1, 50)
+ ISE=OPT1( 1E-16, 1E-18, 1E-11)
+ NE= OPT1( 1.5, 1.2, 2.0)
+ BR= OPT2( 2, 1, 10)
+ NR= OPT2( 1.0, 0.9, 1.1)
+ IKR=OPT2( 50E-3, 1E-3, 1)
+ ISC=OPT2( 1E-12, 1E-15, 1E-10)
+ NC= OPT2( 1.5, 1.2, 2.0)

.DC DATA=BASEF SWEEP OPTIMIZE=OPT1 RESULTS=IBVBE,ICVBE
+ MODEL=OPTMOD
.MEAS DC IBVBE ERR1 PAR(IB) I2(Q1) MINVAL=1E-14 IGNORE=1E-16
.MEAS DC ICVBE ERR1 PAR(IC) I1(Q1) MINVAL=1E-14 IGNORE=1E-16

.DC DATA=BASER SWEEP OPTIMIZE=OPT2 RESULTS=IBVBER,ICVBER
+ MODEL=OPTMOD
.MEAS DC IBVBER ERR1 PAR(IB) I2(Q1) MINVAL=1E-14 IGNORE=1E-16
.MEAS DC ICVBER ERR1 PAR(IC) I1(Q1) MINVAL=1E-14 IGNORE=1E-16

.DC DATA=BASEF
.PRINT DC PAR(IC) I1(Q1) PAR(IB) I2(Q1)
.DC DATA=BASER
.PRINT DC PAR(IC) I1(Q1) PAR(IB) I2(Q1)

**Optimization Results**

RESIDUAL SUM OF SQUARES = 2.196240E-02

**Optimized Parameters OPT1**

<table>
<thead>
<tr>
<th>%NORM-SEN</th>
<th>%CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>.PARAM BF = 1.4603E+02</td>
<td>2.7540E+00</td>
</tr>
<tr>
<td>.PARAM IS = 2.8814E-15</td>
<td>3.7307E+00</td>
</tr>
<tr>
<td>.PARAM NF = 9.9490E-01</td>
<td>9.1532E+01</td>
</tr>
<tr>
<td>.PARAM IKF = 8.4949E-02</td>
<td>1.3782E-02</td>
</tr>
<tr>
<td>.PARAM RE = 6.2358E-01</td>
<td>8.6337E-02</td>
</tr>
<tr>
<td>.PARAM ISE = 5.0569E-16</td>
<td>1.0221E-01</td>
</tr>
<tr>
<td>.PARAM NE = 1.3489E+00</td>
<td>1.7806E+00</td>
</tr>
</tbody>
</table>

**Optimization Results**

RESIDUAL SUM OF SQUARES = 1.82776

**Optimized Parameters OPT2**

<table>
<thead>
<tr>
<th>%NORM-SEN</th>
<th>%CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>.PARAM BF = 1.4603E+02</td>
<td>2.7540E+00</td>
</tr>
<tr>
<td>.PARAM IS = 2.8814E-15</td>
<td>3.7307E+00</td>
</tr>
<tr>
<td>.PARAM NF = 9.9490E-01</td>
<td>9.1532E+01</td>
</tr>
<tr>
<td>.PARAM IKF = 8.4949E-02</td>
<td>1.3782E-02</td>
</tr>
<tr>
<td>.PARAM RE = 6.2358E-01</td>
<td>8.6337E-02</td>
</tr>
<tr>
<td>.PARAM ISE = 5.0569E-16</td>
<td>1.0221E-01</td>
</tr>
<tr>
<td>.PARAM NE = 1.3489E+00</td>
<td>1.7806E+00</td>
</tr>
</tbody>
</table>
.PARAM BR = 1.0000E+01 $ 1.1939E-01 1.7678E+00
.PARAM NR = 9.8185E-01 $ 1.4880E+01 -1.1685E-03
.PARAM IKR = 7.3896E-01 $ 1.2111E-03 -3.5325E+01
.PARAM ISC = 1.8639E-12 $ 6.6144E+00 -5.2159E-03
.PARAM NC = 1.2800E+00 $ 7.8385E+01 1.6202E-03

Figure 13-28: BJT Optimization Forward Gummel Plots
GaAsFET Model DC Optimization

This example circuit is a high-performance GaAsFET transistor. The design target is to match HP4145 DC measured data to the Star-Hspice LEVEL=3 JFET model.

The Star-Hspice strategy is as follows:

■ MEASURE IDSERR is an ERR1 type function providing linear attenuation of the error results, starting at 20 mA and ignoring all currents below 1 mA. The high current fit is the most important for this model.

■ The OPT1 optimization function allows all DC parameters to be simultaneously optimized.

■ The .DATA statement merges raw data files TD1.dat and TD2.dat together.

■ The graph plot model sets the parameter MONO=1 to remove the retrace lines from the family of curves.
GaAsFET Model DC Optimization Input Netlist File

*FILE JOPT.SP JFET OPTIMIZATION
.OPTIONS ACCT NOMOD POST
+ RELI=2E-4 RELV=2E-4

VG GATE 0 XVGS
VD DRAIN 0 XVDS
J1 DRAIN GATE 0 JFETN1

.MODEL JFETN1 NJF LEVEL=3 CAPOP=1 SAT=3
+ NG=1
+ CGS=1P CGD=1P RG=1
+ VTO=VTO BETA=BETA LAMBDA=LAMBDA
+ RS=RDS RD=RDS IS=1E-15 ALPHA=ALPHA
+ UCRIT=UCRIT SATEXP=SATEXP
+ GAMDS=GAMDS VGEXP=VGEXP

.PARAM
+ VTO=OPT1(-.8,-4,0)
+ VGEXP=OPT1(2,1,3.5)
+ GAMDS=OPT1(0,-.5,0)
+ BETA= OPT1(6E-3, 1E-5,9E-2)
+ LAMBDA=OPT1(30M,1E-7,5E-1)
+ RDS=OPT1(1,.001,40)
+ ALPHA=OPT1(2,1,3)
+ UCRIT=OPT1(.1,.001,1)
+ SATEXP=OPT1(1,.5,3)

.DC DATA=DESIRED OPTIMIZE=OPT1 RESULTS=IDSERR MODEL=CONV
.MODEL CONV OPT RELIN=1E-4 RELOUT=1E-4 CLOSE=100 ITROPT=25
.MEASURE DC IDSERR ERR1 PAR(XIDS) I(J1) MINVAL=20M IGNORE=1M

.DC DATA=DESIRED
.GRAPH PAR(XIDS) I(J1)
.MODEL GRAPH PLOT MONO=1
.PRINT PAR(XVGS) PAR(XIDS) I(J1)

.DATA DESIRED MERGE
+ FILE=JDC.DAT XVDS=1 XVGS=2 XIDS=3
.ENDDATA
.END

Optimization Results
RESIDUAL SUM OF SQUARES = 7.582202E-02
## Optimized Parameters Opt1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>%NORM-SEN</th>
<th>%CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTO</td>
<td>-1.1067</td>
<td>64.6110</td>
<td>43.9224M</td>
</tr>
<tr>
<td>VGEXP</td>
<td>2.9475</td>
<td>13.2024</td>
<td>219.4709M</td>
</tr>
<tr>
<td>GAMDS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>BETA</td>
<td>11.8701M</td>
<td>17.2347</td>
<td>136.8216M</td>
</tr>
<tr>
<td>LAMBDA</td>
<td>138.9821M</td>
<td>2.2766</td>
<td>-1.5754</td>
</tr>
<tr>
<td>RDS</td>
<td>928.3216M</td>
<td>704.3204M</td>
<td>464.0863M</td>
</tr>
<tr>
<td>ALPHA</td>
<td>2.2914</td>
<td>728.7492M</td>
<td>168.4004M</td>
</tr>
<tr>
<td>UCRIT</td>
<td>1.0000M</td>
<td>18.2438M</td>
<td>-125.0856</td>
</tr>
<tr>
<td>SATEXP</td>
<td>1.4211</td>
<td>1.2241</td>
<td>2.2218</td>
</tr>
</tbody>
</table>

---

**Figure 13-30: JFET Optimization**

![Graph showing JFET Optimization](image)
MOS Op-amp Optimization

The design goals for the MOS operational amplifier are:

- Minimize the gate area (and hence the total cell area).
- Minimize the power dissipation.
- Open-loop transient step response of 100 ns for rising and falling edges

The Star-Hspice strategy is:

- Simultaneous optimization of two amplifier cells for rising and falling edges.
- Total power is power for two cells.
- The optimization transient analysis must be longer to allow for a range of values in intermediate results.
- All transistor widths and lengths are optimized.
- Transistor area is calculated algebraically, a voltage value is used, and the resulting voltage is minimized.
- The transistor area measure statement uses MINVAL to give less weighting to the area minimization.
- Bias voltage is optimized.

MOS Op-amp Optimization Input Netlist File

AMPOPT.SP  MOS OPERATIONAL AMPLIFIER OPTIMIZATION

.OPTION RELV=1E-3 RELVAR=.01 NOMOD ACCT POST
.PARAM VDD=5  VREF='VDD/2'
VDD VSUPPLY 0 VDD
VIN+ VIN+ 0 PWL(0 , 'VREF-10M' 10NS 'VREF+10M')
VINBAR+ VINBAR+ 0 PWL(0 , 'VREF+10M' 10NS 'VREF-10M')
VIN- VIN- 0 VREF
VBIAS VBIAS 0 BIAS
.GLOBAL VSUPPLY VBIAS

XRISE VIN+ VIN- VOUTR AMP
CLOADR VOUTR 0 .4P
XFALL VINBAR+ VIN- VOUTF AMP
CLOADF VOUTF 0 .4P
.MACRO AMP VIN+ VIN− VOUT
M1 2 VIN− 3 3 MOSN W=WM1 L=LM
M2 4 VIN+ 3 3 MOSN W=WM1 L=LM
M3 2 2 VSUPPLY VSUPPLY MOSP W=WM1 L=LM
M4 4 2 VSUPPLY VSUPPLY MOSP W=WM1 L=LM
M5 VOUT VBIAS 0 0 MOSN W=WM5 L=LM
M6 VOUT 4 VSUPPLY VSUPPLY MOSP W=WM6 L=LM
M7 3 VBIAS 0 0 MOSN W=WM7 L=LM
.ENDS

.PARAM AREA='4*WM1*LM + WM5*LM + WM6*LM + WM7*LM'
VX 1000 0 AREA
RX 1000 0 1K

.MODEL MOSP PMOS (VTO=-1 KP=2.4E-5 LAMBDA=.004 + GAMMA =.37 TOX=3E-8 LEVEL=3)
.MODEL MOSN NMOS (VTO=1.2 KP=6.0E-5 LAMBDA=.0004 + GAMMA =.37 TOX=3E-8 LEVEL=3)

.PARAM WM1=OPT1(60U,20U,100U) + WM5=OPT1(40U,20U,100U) + WM6=OPT1(300U,20U,500U) + WM7=OPT1(70U,40U,200U) + LM=OPT1(10U,2U,100U) + BIAS=OPT1(2.2,1.2,3.0)

.TRAN 2.5N 300N SWEEP OPTIMIZE=OPT 1 + RESULTS=DELAYR,DELAYF,TOT_POWER,AREA MODEL=OPT
.MODEL OPT OPT CLOSE=100

.TRAN 2N 150N
.MEASURE DELAYR TRIG AT=0 TARG V(VOUTR) VAL=2.5 RISE=1 + GOAL=100NS
.MEASURE DELAYF TRIG AT=0 TARG V(VOUTF) VAL=2.5 FALL=1 + GOAL=100NS
.MEASURE TOT_POWER AVG POWER GOAL=10MW
.MEASURE AREA MIN PAR(AREA) GOAL=1E-9 MINVAL=100N
.PRINT V(VIN+) V(VOUTR) V(VOUTF)
.END
Optimization Results

RESIDUAL SUM OF SQUARES = 4.654377E-04
NORM OF THE GRADIENT = 6.782920E-02

Optimized Parameters Opt1

* %NORM-SEN %CHANGE

.PARAM WM1 = 47.9629U $ 1.6524 -762.3661M
.PARAM WM5 = 66.8831U $ 10.1048 23.4480M
.PARAM WM6 = 127.1928U $ 12.7991 22.7612M
.PARAM WM7 = 115.8941U $ 9.6104 -246.4540M
.PARAM LM = 6.2588U $ 20.3279 -101.4044M
.PARAM BIAS = 2.7180 $ 45.5053 5.6001M

*** OPTIMIZE RESULTS MEASURE NAMES AND VALUES
* DELAYR = 100.4231N
* DELAYF = 99.5059N
* TOT_POWER = 10.0131M
* AREA = 3.1408N

Figure 13-31: CMOS Op-amp
Figure 13-32: Operational Amplifier Optimization
Chapter 14
Using Passive Device Models

This chapter describes model statements for passive devices. It includes statements for resistors, inductors, and capacitors.

Use the set of passive model statements in conjunction with element definitions to construct a wide range of board and integrated circuit-level designs. Passive device models let you include transformers, PC board trace interconnects, coaxial cables and transmission lines in an analysis. The wire element model is specifically designed to model the RC delay and RC transmission line effects of interconnects at both the IC level and the PC board level.

To aid in designing power supplies, a mutual-inductor model includes switching regulators and a number of other magnetic circuits, including a magnetic-core model and element. You can specify precision modeling of passive elements using geometric, temperature, and parasitic model parameters.

This chapter describes:
- Resistor Device Model and Equations
- Capacitor Device Model and Equations
- Inductor Device Model and Equations
Resistor Device Model and Equations

Wire RC Model

The Star-HSpice wire element RC model is a CRC (pi) model. Use the CRATIO wire model parameter to allocate the parasitic capacitance of the wire element between the model’s input capacitor and output capacitor. This allows for symmetric node impedance for bidirectional circuits such as buses.

Syntax

```
.MODEL mname R keyword=value <CRATIO=val>
```

- **mname**: Model name. Elements reference the model with this name.
- **R**: Specifies a wire model.
- **keyword**: Any model parameter name.
- **CRATIO**: Ratio to allocate the total wire element parasitic capacitance between the capacitor connected to the input node and the capacitor connected to the output node of the wire element pi model.

You can assign CRATIO any value between 0 and 1:

- 0 — Assigns all of the parasitic capacitance (CAPeff) to the output node.
- 0.5 — Assigns half of the parasitic capacitance to the input node and half to the output node.
Using Passive Device Models

Resistor Device Model and Equations

CRATIO

1 — Assigns all of the parasitic capacitance to the input node
The default is 0.5. CRATIO values smaller than 0.5 assign more of the capacitance to the output node than to the input node. Values greater than 0.5 assign more of the capacitance to the input node than to the output node.
If you specify a value outside the range of 0 to 1.0 CRATIO, Star-Hspice displays a warning, sets CRATIO to 0.5, and continues the analysis.

A resistor referred to as a wire model behaves like an elementary transmission line if you specify an optional capacitor from node n2 to a bulk or ground node in the model statement. The bulk node functions as a ground plane for the wire capacitance.

A wire is described by a drawn length and a drawn width. The resistance of the wire is the effective length multiplied by RSH divided by the effective width.

To avoid syntactic conflicts, if a resistor model exists using the same name as a parameter for rval in the element statement, Star-Hspice uses the model name. In the following example, R1 assumes that REXX refers to the model and not the parameter.

```plaintext
.PARAMETER REXX=1
R1 1 2 REXX
.MODEL REXX R RES=1
```
### Wire Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BULK</td>
<td>gnd</td>
<td>Default</td>
<td>Reference node for capacitance</td>
</tr>
<tr>
<td>CAP</td>
<td>F</td>
<td>0</td>
<td>Default capacitance</td>
</tr>
<tr>
<td>CAPSW</td>
<td>F/m</td>
<td>0</td>
<td>Sidewall fringing capacitance</td>
</tr>
<tr>
<td>COX</td>
<td>F/m²</td>
<td>0</td>
<td>Bottomwall capacitance</td>
</tr>
<tr>
<td>DI</td>
<td></td>
<td>0</td>
<td>Relative dielectric constant</td>
</tr>
<tr>
<td>DLR</td>
<td>m</td>
<td>0</td>
<td>Difference between drawn length and actual length (for resistance calculation only). For capacitance calculation, DW is used DLeff=DLR⋅SCALM</td>
</tr>
<tr>
<td>DW</td>
<td>m</td>
<td>0</td>
<td>Difference between drawn width and actual width DWeff=DW⋅SCALM</td>
</tr>
<tr>
<td>L</td>
<td>m</td>
<td>0</td>
<td>Default length of wire</td>
</tr>
<tr>
<td>LEVEL</td>
<td></td>
<td></td>
<td>Model selector (not used)</td>
</tr>
<tr>
<td>RAC</td>
<td>ohm</td>
<td></td>
<td>Default AC resistance (RACeff default is Reff)</td>
</tr>
<tr>
<td>RES</td>
<td>ohm</td>
<td>0</td>
<td>Default resistance</td>
</tr>
<tr>
<td>RSH</td>
<td></td>
<td>0</td>
<td>Sheet resistance/square</td>
</tr>
<tr>
<td>SHRINK</td>
<td></td>
<td>1</td>
<td>Shrink factor</td>
</tr>
<tr>
<td>TC1C</td>
<td>1/deg</td>
<td>0</td>
<td>First-order temperature coefficient for capacitance</td>
</tr>
<tr>
<td>TC2C</td>
<td>1/deg²</td>
<td>0</td>
<td>Second-order temperature coefficient for capacitance</td>
</tr>
<tr>
<td>TC1R</td>
<td>1/deg</td>
<td>0</td>
<td>First-order temperature coefficient for resistance</td>
</tr>
<tr>
<td>TC2R</td>
<td>1/deg²</td>
<td>0</td>
<td>Second-order temperature coefficient for resistance</td>
</tr>
</tbody>
</table>
**Resistor Device Model and Equations**

**Wire Resistance Calculation**

You can specify the wire width and length in both the element and model statements. The element values override the model values. The element width and length are scaled by the option SCALE and the model parameter SHRINK. The model width and length are scaled by the option SCALM and the model parameter SHRINK.

The effective width and length are calculated as follows:

\[
W_{\text{eff}} = W_{\text{scaled}} - 2 \cdot D_{\text{Weff}}
\]

\[
L_{\text{eff}} = L_{\text{scaled}} - 2 \cdot D_{\text{LReff}}
\]

If element resistance is specified:

\[
R_{\text{eff}} = \frac{R \cdot \text{SCALE(element)}}{M}
\]

Otherwise, if \((W_{\text{eff}} \cdot L_{\text{eff}} \cdot R_{\text{SH}})\) is greater than zero, then:

\[
R_{\text{eff}} = \frac{L_{\text{eff}} \cdot R_{\text{SH}} \cdot \text{SCALE(element)}}{M \cdot W_{\text{eff}}}
\]

If \((W_{\text{eff}} \cdot L_{\text{eff}} \cdot R_{\text{SH}})\) is zero, then:

\[
R_{\text{eff}} = \frac{R_{\text{ES}} \cdot \text{SCALE(element)}}{M}
\]

---

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICK</td>
<td>m</td>
<td>0</td>
<td>Dielectric thickness</td>
</tr>
<tr>
<td>TREF</td>
<td>deg C</td>
<td>TNOM</td>
<td>Temperature reference for model parameters</td>
</tr>
<tr>
<td>W</td>
<td>m</td>
<td>0</td>
<td>Default width of wire</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(W_{\text{scaled}} = W \cdot \text{SHRINK} \cdot \text{SCALM})</td>
</tr>
</tbody>
</table>

---

If AC resistance is specified in the element, then:

\[ \text{RAC}_{\text{eff}} = \frac{AC \cdot \text{SCALE}(\text{element})}{M} \]

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

\[ \text{RAC}_{\text{eff}} = \frac{RAC \cdot \text{SCALE}(\text{element})}{M} \]

If neither are specified, it defaults to:

\[ \text{RAC}_{\text{eff}} = \text{Reff} \]

If the resistance is less than option RESMIN, it is reset it to RESMIN and a warning message is displayed.

\[ \text{RESMIN} = \frac{1}{G\text{MAX} \cdot 1000 \cdot M} \]

**Wire Capacitance Calculation**

The effective length is the scaled drawn length less 2 \( \cdot \) DLeff. Leff represents the effective length of the resistor from physical edge to physical edge. DWeff is the distance from the drawn edge of the resistor to the physical edge of the resistor. The effective width is the same as the width used in the resistor calculation.

\[ \text{Leff} = L_{\text{scaled}} - 2 \cdot D\text{Leff} \]

\[ \text{Weff} = W_{\text{scaled}} - 2 \cdot D\text{Weff} \]

If the element capacitance C is specified:

\[ \text{CAP}_{\text{eff}} = C \cdot \text{SCALE}(\text{element}) \cdot M \]

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

\[
\text{CAP}_{\text{eff}} = M \cdot \text{SCALE}(\text{element}) \cdot (\text{Leff} \cdot \text{Weff} \cdot \text{COX} + 2 \cdot \text{Leff} + \text{Weff}) \cdot \text{CAPSW}]
\]
The computation of the bottom wall capacitance \( COX \) is based upon a hierarchy of defaults and specified values involving the dielectric thickness \( \text{THICK} \), the relative dielectric constant \( DI \), and two absolute dielectric constants \( \varepsilon_o \) and \( \varepsilon_{ox} \), as follows:

1. If \( COX=\text{value} \) is given, the value is used.

2. If \( COX \) is not given specifically but \( \text{THICK} \) (the dielectric thickness) is given and nonzero:
   a. If \( DI=\text{value} \) is given and nonzero then:
      \[
      COX = \frac{DI \cdot \varepsilon_o}{\text{THICK}}
      \]
   b. If \( DI \) is not given, or is zero, then:
      \[
      COX = \frac{\varepsilon_{ox}}{\text{THICK}}
      \]
      where
      \[
      \varepsilon_o = 8.8542149 \times 10^{-12} \text{ F/meter}
      \]
      \[
      \varepsilon_{ox} = 3.453148 \times 10^{-11} \text{ F/meter}
      \]

3. If \( COX \) is not given and \( \text{THICK}=0 \) is an error.

   If only the model capacitance \( \text{CAP} \) is specified, then:
   \[
   \text{CAP}_{\text{eff}} = \text{CAP} \cdot \text{SCALE(element)} \cdot M
   \]

   If the capacitance is specified and the bulk node is not specified, then capacitance is not evaluated and a warning message is issued.

**Resistor Noise Equation**

The thermal noise of a resistor is modeled by:

\[
inr = \left( \text{NOISE} \cdot \frac{4kT}{R_{\text{val}}} \right)^{1/2}
\]

where \( \text{NOISE} \) is a model parameter that defaults to 1. To eliminate the contribution of resistor noise, use the \( \text{NOISE} \) parameter. To specify the \( \text{NOISE} \) parameter, use a model for the resistor.
Noise Summary Print Out Definitions

<table>
<thead>
<tr>
<th>RX</th>
<th>Transfer function of thermal noise to the output. This is not noise, but is a transfer coefficient, reflecting the contribution of thermal noise to the output.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOT, V²/Hz</td>
<td>Total output noise: ( \dot{\text{OT}} = RX^2 \cdot \text{in}r^2 )</td>
</tr>
</tbody>
</table>

Resistor Temperature Equations

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

\[
R(t) = R \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2) \\
RAC(t) = RAC \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2) \\
C(t) = C \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)
\]

<table>
<thead>
<tr>
<th>( \Delta t )</th>
<th>( t - t_{\text{nom}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t )</td>
<td>Element temperature in °K: ( t = \text{circuit temp} + \text{DTEMP} + 273.15 )</td>
</tr>
<tr>
<td>( t_{\text{nom}} )</td>
<td>Nominal temperature in °K: ( t_{\text{nom}} = 273.15 + \text{TNOM} )</td>
</tr>
</tbody>
</table>
### Capacitor Device Model and Equations

#### Capacitance Model

**Syntax**

```
.MODEL mname C parameter=value
```

<table>
<thead>
<tr>
<th>mname</th>
<th>Model name</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Specifies a capacitance model</td>
</tr>
<tr>
<td>parameter</td>
<td>Any model parameter name</td>
</tr>
</tbody>
</table>

#### Capacitance Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAP</td>
<td>F</td>
<td>0</td>
<td>Default capacitance value</td>
</tr>
<tr>
<td>CAPSW</td>
<td>F/m</td>
<td>0</td>
<td>Sidewall fringing capacitance</td>
</tr>
<tr>
<td>COX</td>
<td>F/m²</td>
<td>0</td>
<td>Bottomwall capacitance</td>
</tr>
<tr>
<td>DEL</td>
<td>m</td>
<td>0</td>
<td>Difference between drawn width and actual width or length</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DELeff = DEL ⋅ SCALM</td>
</tr>
<tr>
<td>DI</td>
<td></td>
<td>0</td>
<td>Relative dielectric constant</td>
</tr>
<tr>
<td>L</td>
<td>m</td>
<td>0</td>
<td>Default length of capacitor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lscaled = L ⋅ SHRINK ⋅ SCALM</td>
</tr>
<tr>
<td>SHRINK</td>
<td></td>
<td>1</td>
<td>Shrink factor</td>
</tr>
<tr>
<td>TC1</td>
<td>1/deg</td>
<td>0</td>
<td>First temperature coefficient for capacitance</td>
</tr>
<tr>
<td>TC2</td>
<td>1/deg²</td>
<td>0</td>
<td>Second temperature coefficient for capacitance</td>
</tr>
<tr>
<td>THICK</td>
<td>m</td>
<td>0</td>
<td>Insulator thickness</td>
</tr>
</tbody>
</table>
Parameter Limit Checking

Star-Hspice writes a warning message to the output listing file if a capacitive element value exceeds 0.1 Farad. This feature helps you identify elements with missing units or wrong values, particularly those in automatically produced netlists.

Capacitor Device Equations

Effective Capacitance Calculation

A model can be associated with a capacitor in Star-Hspice. You can specify some of the parameters in both the element and model descriptions. The element values override the model values. The option SCALE and the model parameter SHRINK scale the element width and length. The option SCALM and the model parameter SHRINK scale the model width and length.

The effective width and length are calculated as follows:

- \[ W_{eff} = W_{scaled} - 2 \cdot DELeff \]
- \[ L_{eff} = L_{scaled} - 2 \cdot DELeff \]

If the element capacitance \( C \) is specified:

- \( CAP_{eff} = C \cdot SCALE(\text{element}) \cdot M \)

Otherwise, the capacitance is calculated from the \( L_{eff} \), \( W_{eff} \) and \( COX \).

- \( CAP_{eff} = M \cdot SCALE(\text{element}) \cdot \left[ L_{eff} \cdot W_{eff} \cdot COX + 2 \cdot (L_{eff} + W_{eff}) \cdot CAPSW \right] \)
If COX is not specified, but THICK is not zero, then:

\[
COX = \frac{DI \cdot \varepsilon_o}{THICK} \quad \text{if } DI \text{ not zero}
\]

Or

\[
COX = \frac{\varepsilon_{ox}}{THICK} \quad \text{if } DI=0
\]

Where

\[
\varepsilon_o = 8.8542149 \times 10^{-12} \frac{F}{meter}
\]

\[
\varepsilon_{ox} = 3.453148 \times 10^{-11} \frac{F}{meter}
\]

If only model capacitance CAP is specified, then:

\[
CAP_{eff} = CAP \cdot SCALE(element) \cdot M
\]

**Capacitance Temperature Equation**

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

\[
C(t) = C \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)
\]

<table>
<thead>
<tr>
<th>(\Delta t)</th>
<th>(t - \text{tnom})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t)</td>
<td>Element temperature in degrees Kelvin</td>
</tr>
<tr>
<td>(t_{nom})</td>
<td>Nominal temperature in degrees Kelvin</td>
</tr>
<tr>
<td></td>
<td>(t_{nom}+273.15 + \text{TNOM})</td>
</tr>
</tbody>
</table>
Inductor Device Model and Equations

You can use several elements and models to analyze switching regulators, transformers, and mutual inductive circuits. These elements include magnetic winding elements, mutual cores, and magnetic core models.

You can use the Star-Hspice saturable core model for chokes, saturable transformers, and linear transformers. To use the model, you must provide a mutual core statement, specify the core parameters with a .MODEL statement, and provide specification of the windings around each core element with a magnetic winding element statement.

Inductor Core Models

Magnetic Core Syntax

```
.MODEL mname L (<pname1 = val1>…)
```

Jiles-Atherton Ferromagnetic Core Syntax

```
.MODEL mname CORE (LEVEL=1 <pname1 = val1>…)
```

<table>
<thead>
<tr>
<th>mname</th>
<th>Model name. Elements refer to the model by this name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>Identifies a saturable core model</td>
</tr>
<tr>
<td>CORE</td>
<td>Identifies a Jiles-Atherton Ferromagnetic Core model</td>
</tr>
<tr>
<td>level=x</td>
<td>Equation selection for Jiles-Atherton model</td>
</tr>
<tr>
<td>pname1=val1</td>
<td>Value of the model parameter. Each core model can include several model parameters.</td>
</tr>
</tbody>
</table>

Example

```
.MODEL CHOKE L(BS=12K BR=10K HS=1 HCR=.2 HC=.3 AC=.1. LC=3.)
```
Obtain the core model parameters from the manufacturer’s data. Figure 14-1 illustrates the required $b$-$h$ loop parameters for the model. The model includes core area, length, and gap size, as well as the core growth time constant.

**Example**

*file: bhloop.sp b-h loop nonlinear magnetic core transformer
* plot in avanwaves i(l1) versus 22 to get b-h loop
.option acct method=gear post rmax=.05
.tran 1m 25m
.probe mu=lx0(k1) h=lx1(k1) b=lx2(k1) L1=lv1(l1) L2=lv1(l2)
+ i(l1)
k1 11 12 mag2
l1 1 0 nt=20
l2 2 0 nt=20
r11 1 11 1
v11 11 0 sin (0 5 60
r22 2 22 1
c22 22 0 1
.model mag2 1 bs=6k br=3k hs=1 hcr=.1 hc=.8 ac=1 lc=16
.end
### Magnetic Core Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>cm · 2</td>
<td>1.0</td>
<td>Core area</td>
</tr>
<tr>
<td>BS</td>
<td>Gauss</td>
<td>13000</td>
<td>Magnetic flux density at saturation</td>
</tr>
<tr>
<td>BR</td>
<td>Gauss</td>
<td>12000</td>
<td>Residual magnetization</td>
</tr>
<tr>
<td>HC</td>
<td>Oersted</td>
<td>0.8</td>
<td>Coercive magnetizing force</td>
</tr>
<tr>
<td>HCR</td>
<td>Oersted</td>
<td>0.6</td>
<td>Critical magnetizing force</td>
</tr>
<tr>
<td>HS</td>
<td>Oersted</td>
<td>1.5</td>
<td>Magnetizing force at saturation</td>
</tr>
<tr>
<td>LC</td>
<td>cm</td>
<td>3.0</td>
<td>Core length</td>
</tr>
<tr>
<td>LG</td>
<td>cm</td>
<td>0.0</td>
<td>Gap length</td>
</tr>
<tr>
<td>TC</td>
<td>s</td>
<td>0.0</td>
<td>Core growth time constant</td>
</tr>
</tbody>
</table>
Figure 14-1: Magnetic Saturable Core Model
Jiles-Atherton Core Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>2</td>
<td>Model selector. For the Jiles-Atherton model, set LEVEL=1. LEVEL=2, the default, selects the Pheno model, which is the original Star-Hspice model</td>
</tr>
<tr>
<td>AREA, (AC)</td>
<td>cm²</td>
<td>1</td>
<td>Mean of magnetic core cross section. AC is an alias of AREA</td>
</tr>
<tr>
<td>PATH, (LC)</td>
<td>cm</td>
<td>3</td>
<td>Mean of magnetic core path length. LC is an alias of PATH</td>
</tr>
<tr>
<td>MS</td>
<td>amp/meter</td>
<td>1e6</td>
<td>Magnetization saturation</td>
</tr>
<tr>
<td>A</td>
<td>amp/meter</td>
<td>1e3</td>
<td>Characterizes the shape of the anhysteretic magnetization</td>
</tr>
<tr>
<td>ALPHA</td>
<td></td>
<td>1e-3</td>
<td>Represents the coupling between the magnetic domains</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>0.2</td>
<td>Domain flexing parameter</td>
</tr>
<tr>
<td>K</td>
<td>amp/meter</td>
<td>500</td>
<td>Domain an isotropy parameter</td>
</tr>
</tbody>
</table>

Magnetic Core Element Outputs

<table>
<thead>
<tr>
<th>Output Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LX1</td>
<td>magnetic field, ( h ) (oersted)</td>
</tr>
<tr>
<td>LX2</td>
<td>magnetic flux density, ( b ) (gauss)</td>
</tr>
<tr>
<td>LX3</td>
<td>slope of the magnetization curve, ( \frac{dm}{dh} )</td>
</tr>
</tbody>
</table>
Parameter Limit Checking

Star-Hspice writes a warning message to the output listing file if an inductive element value exceeds 0.1 Henry. This feature helps you identify elements with missing units or wrong values, particularly those in automatically produced netlists.

Inductor Temperature Equation

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

\[ L(t) = L \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2) \]

<table>
<thead>
<tr>
<th>Output Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LX4</td>
<td>bulk magnetization, m (amp/meter)</td>
</tr>
<tr>
<td>LX5</td>
<td>slope of the anhysteretic magnetization curve, ( \frac{dm_{an}}{dh} )</td>
</tr>
<tr>
<td>LX6</td>
<td>anhysteretic magnetization, ( m_{an} ) (amp/meter)</td>
</tr>
<tr>
<td>LX7</td>
<td>effective magnetic field, ( h_e ) (amp/meter)</td>
</tr>
</tbody>
</table>

\( \Delta t \): t - tnom

\( t \): Element temperature in degrees Kelvin

\( t = \text{circuit temp} + \text{DTEMP} + 273.15 \)

\( t_{nom} \): Nominal temperature in degrees Kelvin

\( t_{nom} = 273.15 + TNOM \)
Create coupling between inductors with a separate coupling element. Specify mutual inductance between two inductors by the coefficient of coupling, \( k_{value} \), defined by the equation:

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

| \( L_1, L_2 \) | the inductances of the two coupled inductors |
| \( M \) | the mutual inductance between the inductors |

Linear branch relation for transient analysis:

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

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

Linear branch relation for AC analysis:

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

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

\[\text{Note: You must define an inductor reference by a mutual inductor statement; otherwise Star-Hspice displays an error message and terminates.}\]
Jiles-Atherton Ferromagnetic Core Model

The Jiles-Atherton ferromagnetic core model is based on domain wall motion, including both bending and translation. The hysteresis-free (anhysteretic) magnetization curve is described by a modified Langevin expression. This leads to:

\[ m_{an} = M \dot{S} \cdot \left( \coth \left( \frac{h_e}{A} \right) - \frac{A}{h_e} \right) \]

\[ h_e = h + ALPHA \cdot m_{an} \]

where

- \( m_{an} \) is Magnetization level, if domain walls could move freely.
- \( h_e \) is Effective magnetic field.
- \( h \) is Magnetic field.
- \( MS \) is Model parameter that represents the saturation magnetization.
- \( A \) is Model parameter that characterizes the shape of the anhysteretic magnetization.
- \( ALPHA \) is Model parameter that represents the coupling between the magnetic domains.

The above equation generates anhysteretic curves when the model parameter \( ALPHA \) has a small value. Otherwise, it generates some elementary forms of hysteresis loops, which is not a desirable result. The slope of the curve at zero (0) can be calculated by:

\[ \frac{dm_{an}}{dh} = \frac{1}{3 \cdot \frac{A}{MS} - ALPHA} \]

The slope must be positive, therefore the denominator of the above equation must be positive. Star-Hspice generates an error message if the slope becomes negative.
The anhysteretic magnetization represents the global energy state of the material if the domain walls could move freely. But the walls are displaced and bent in the material. If the bulk magnetization $m$ is expressed as the sum of an irreversible component due to wall displacement and a reversible component due to domain wall bending, then:

$$\frac{dm}{dh} = \frac{(m_{an} - m)}{K} + C \cdot \left(\frac{dm_{an}}{dh} - \frac{dm}{dh}\right)$$

or

$$\frac{dm}{dh} = \frac{(m_{an} - m)}{(1 + C) \cdot K} + C \cdot \frac{dm_{an}}{dh}$$

By solving the above differential equation, the bulk magnetization $m$ is obtained. The flux density $b$ is computed from $m$:

$$b = \mu_0 \cdot (h + m)$$

where $\mu_0$, the permeability of free space, is $4\pi \cdot 10^{-7}$, and the units of $h$ and $m$ are in amp/meter. Then the units of $b$ would be in Tesla (Wb/meter$^2$).

**Jiles-Atherton Model Examples**

**Effects of Varying the ALPHA, A, and K Parameters**

This example demonstrates the effects of the ALPHA, A, and K model parameters on the $b$-$h$ curve.

Figure 14-2 shows the $b$-$h$ curves for three values of ALPHA.

Figure 14-3 shows the $b$-$h$ curves for three values of A.

Figure 14-4 shows the $b$-$h$ curves for three values of K.
**Star-Hspice Input File**

* Test the Jiles-Atherton model

```plaintext
.options post
* the following analysis studies the effect of parameter ALPHA.
.param palpha=0.0 pk=0.0 pc=0.0 pa=26
.tran 0.01 1 sweep palpha poi 3 0.0 5.0e-5 1.0e-4
* the following analysis studies the effects of parameter A.
.param palpha=0.0 pk=0.0 pc=0.0 pa=26
.tran 0.01 1 sweep pa poi 3 10 26 50
* the following analysis studies the effects of parameter K.
.param palpha=0.0 pk=5 pc=1.05 pa=26
.tran 0.01 1.25 $ sweep pk poi 2 5 50
rl 1 2 1
l1 2 0 nt=50
k1 l1 ct
igen 0 1 sin(0 0.1a 1hz 0 )
.model ct core LEVEL=1 ms=420k k=pk c=pc a=pa
+ alpha=palpha area=1.17 path=8.49
.probe b=lx2(k1) h=lx1(k1) i(r1) v(1)
.probe dmdh=lx3(k1) m=lx4(k1) man=lx6(k1)
.probe l=lv1(l1)
.alter
.param pk=50
.end
```
Plots of the b-h Curve

Figure 14-2: Variation of Anhysteretic b-h Curve: the Slope Increases as ALPHA Increases

Figure 14-3: Variation of Anhysteretic b-h Curve: the Slope Decreases as A Increases
Discontinuities in Inductance Due to Hysteresis

This example creates multiloop hysteresis b-h curves for a magnetic core. Discontinuities in the inductance, which is proportional to the slope of the b-h curve, can cause convergence problems. Figure 14-5 demonstrates the effects of hysteresis on the inductance of the core.

Star-Hspice Input File

```
*file tj2b.sp Multiloop hysteresis test using Jiles-Atherton model.
.options post
.tran 0.01 5
rl 1 2 1
li 2 0 nt=50
k1 li ct
igen 0 10 sin(0 0.1a 1hz 0 )
ipls 0 20 pwl(0,0 1m,0.5 1s,0.5 + 1.001,1.0 2.000,1.0 + 2.001,1.5 3.000,1.5 + 3.001,2.0 4.000,2.0 + 4.001,2.5 5.000,2.5) gigen 0 1 cur='v(10)*v(20)'
rpls 0 20 1
rsin 0 10 1
```
.model ct core LEVEL=1 ms=420k k=18 c=1.05 a=26
+ alpha=2e-5 area=1.17 path=8.49

.probe b=lx2(k1) h=lx1(k1) i(rl) v(l)
.probe dmdh=lx3(k1) m=lx4(k1) dmandh=lx5(k1)
+ man=lx6(k1)
.probe l=lv1(l1) heff=lx7(k1)
.end

Plots of the Hysteresis Curve and Inductance

Figure 14-5: Hysteresis Curve and Inductance of a Magnetic Core

Optimization of Parameter Extraction

This example demonstrates the usage of optimization in the parameter extraction
of the Jiles-Atherton model. Figure 14-6 shows the plots of the core output
before and after optimization.

Star-Hspice Input File

*file tj_opt.sp for Jiles-Atherton model parameter optimization.
.options post
+ delmax=5m
.param palpha=0.0
.param pms= opt1(150k,100k,500k)
+ pa =opt1(10,5,50)
+ pk=opt1(5,1,50)
\begin{verbatim}
+ pc= opt1(1,0,3)
.tran 0.01 1.0
.tran 0.01 1.0 sweep
+ optimize=opt1 results=bsat,br,hc model=optmod
.model optmod opt itropt=40
+ relin=1e-4 relout=1e-6
.meas bsat find par('abs(lx2(k1))') when lx1(k1)=5.0 goal=3.1k
.meas br find par('abs(lx2(k1))') when lx1(k1)=0 td=.25 goal=1k
.meas hc find par('abs(lx1(k1))') when lx2(k1)=0 td=.25 goal=.4
rl 1 2 0.01
ll 2 0 nt=20
k1 ll ct
igen 0 1 sin(0 2a 1hz 0 )
.model ct core LEVEL=1 ms=pms k=pk c=pc a=pa
+ alpha=palpha area=1.17 path=8.49
.probe b=lx2(k1) h=lx1(k1) i(rl) v(1)
.probe dmdh=lx3(k1) m=lx4(k1) dmandh=lx5(k1)
+ man=lx6(k1)
.probe l=lv1(ll) heff=lx7(k1)
.end

Analysis Results Listing
****** transient analysis tnom= 25.000 temp= 25.000
optimization results
residual sum of squares = 1.043893E-12
norm of the gradient = 1.411088E-06
marquardt scaling parameter = 1.267004E-04
no. of function evaluations = 30
no. of iterations = 11
optimization completed

norm of gradient < grad= 1.0000E-06 on last iterations
**** optimized parameters opt1

.param pms = 267.5975k
.param pa = 27.8196
.param pk = 37.2947
.param pc = 316.4197m
\end{verbatim}
*** Measure results
bsat = 3.1000E+03
br = 9.9999E+02
hc = 3.9880E-01

Figure 14-6: Output Curves Before Optimization (top), and After Optimization (bottom)
Chapter 15
Using Diodes

Use diode models to describe pn junction diodes within MOS and bipolar integrated circuit environments and discrete devices. You can use four types of models and a wide range of parameters to model standard junction diodes:

- Zener diodes
- Silicon diffused junction diodes
- Schottky barrier diodes
- Nonvolatile memory diodes (tunneling current)

Note: See “Introducing MOSFETs” on page 20-1; “Selecting MOSFET Models: LEVEL 1-40” on page 21-1; and “Preparing for Simulation” on page 25-2 for other MOSFET and standard discrete diodes.

Diode model types include the junction diode model and the Fowler-Nordheim model. The junction diode model has two variations: geometric and nongeometric.

This chapter provides an overview of model parameters and scaling effects for the geometric and nongeometric junction diodes. It describes:

- Understanding the Diode Types
- Using Diode Model Statements
- Specifying Junction Diode Models
- Determining Temperature Effects on Junction Diodes
- Using Junction Diode Equations
- Using the Junction Cap Model
- Using the Fowler-Nordheim Diode
Using Diodes

- Converting National Semiconductor Models
Understanding the Diode Types

Use the geometric junction diode to model IC-based standard silicon diffused diodes, Schottky barrier diodes, and Zener diodes. The geometric parameter lets you specify pn junction poly and metal capacitance dimensions for a particular IC process technology.

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

The Fowler-Nordheim diode defines tunneling current flow through insulators. Use it to model diode effects in nonvolatile EEPROM memory.
Using Diode Model Statements

Use model and element statements to select the diode models. The model statement’s LEVEL parameter selects the type of diode model used:

- LEVEL=1 selects the nongeometric junction diode model
- LEVEL=2 selects the Fowler-Nordheim diode model
- LEVEL=3 selects the geometric junction diode model

You can design Zener, Schottky barrier, and silicon diffused diodes by altering model parameters for both LEVEL 1 and LEVEL 3. LEVEL 2 does not permit modeling of these effects. For Zener diodes, the BV parameter is set for an appropriate Zener breakdown voltage.

If you do not specify the LEVEL parameter in the .MODEL statement, the model defaults to the nongeometric junction diode model, LEVEL 1.

Use control options with the diode model to scale model units, select diffusion capacitance equations, and change model parameters.

Setting Control Options

Control options related to the analysis of diode circuits, as well as other models, include DCAP, DCCAP, GMIN, GMINDC, SCALE, and SCALM. Specify these models using the .OPTIONS statement.

Setting Scaling Options

Use the scale element option, SCALE, to scale LEVELs 2 and 3 diode element parameters. Use the scale model option, SCALM, to scale LEVELs 2 and 3 diode model parameters. LEVEL 1 does not use SCALE or SCALM.

Include SCALM=<val> in the .MODEL statement to override global scaling that uses the .OPTION SCALM=<val> statement in a diode model.
Using the Capacitor Equation Selector Option — DCAP

The DCAP option selects the equations used in calculating the depletion capacitance (LEVEL 1 and LEVEL 3). The option DCCAP invokes calculation of capacitances in DC analysis.

Include the DCAP=<val> in the diode’s .MODEL statement to override the global depletion capacitance equation selection with the .OPTIONS DCAP=<val> statement.

Using Control Options for Convergence

Diode convergence problems often occur at the breakdown voltage region when the diode is overdriven or in the OFF condition. To achieve convergence in such cases, include a nonzero value in the model for the series resistor parameter RS, or increase GMIN (the parallel conductance Star-Hspice automatically places in the circuit). You can specify GMIN and GMINDC in the .OPTIONS statement.

The diode control options follow:

<table>
<thead>
<tr>
<th>Function</th>
<th>Control Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacitance</td>
<td>DCAP, DCCAP</td>
</tr>
<tr>
<td>Conductance</td>
<td>GMIN, GMINDC</td>
</tr>
<tr>
<td>Geometry</td>
<td>SCALM, SCALE</td>
</tr>
</tbody>
</table>
Specifying Junction Diode Models

Use the diode element statement to specify the two types of junction diodes, geometric and nongeometric. Use a different element type format for the Fowler-Nordheim model.

The diode element statement parameter fields define the connecting nodes, initialization, temperature, geometric junction, and capacitance parameters of the diode model selected in the diode .MODEL statement. Both LEVEL 1 and LEVEL 3 junction diode models share the same element parameter set. Poly and metal capacitor parameters of LM, LP, WM and WP do not share the same element parameter.

Element parameters take precedence over model parameters, if repeated in the .MODEL statement as model parameters.

Parameters common to both element and model statements are: AREA, PJ, M, LM, LP, WM, WP, W, and L.

Table 15-1: Junction Diode Element Parameters

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netlist</td>
<td>Dxxx, n+, n-, mname</td>
</tr>
<tr>
<td>Initialization</td>
<td>IC, OFF</td>
</tr>
<tr>
<td>Temperature</td>
<td>DTEMP</td>
</tr>
<tr>
<td>Geometric junction</td>
<td>AREA, L, M, PJ, W</td>
</tr>
<tr>
<td>Geometric capacitance (LEVEL=3</td>
<td>LM, LP, WM, WP</td>
</tr>
<tr>
<td>only)</td>
<td></td>
</tr>
</tbody>
</table>
Using the Junction Model Statement

This section describes how to use the junction model statement.

Syntax

The syntax of the junction model statement is:

```
.MODEL mnameD <LEVEL = val> <keyword = val> ...
```

- **mname**: Model name. The diode element refers to the model by this name.
- **D**: Symbol that identifies a diode model
- **LEVEL**: Symbol that identifies a diode model
  - LEVEL=1 = junction diode
  - LEVEL=2 = Fowler-Nordheim
  - LEVEL=3 = geometric processing for junction diode
- **keyword**: Model parameter keyword such as CJO or IS

Example

```
.MODEL D D (CO=2PF, RS=1, IS=1P)
.MODEL DFOWLER D (LEVEL=2, TOX=100, JF=1E-10, EF=1E8)
.MODEL DGEO D (LEVEL=3, JS=1E-4, JSW=1E-8)
.MODEL d1n750a D
   + LEVEL=1 XP =0.0 EG =1.1
   + XOI =0.0 XOM =0.0 XM =0.0
   + WP =0.0 WM =0.0 LP =0.0
   + LM =0.0 AF =1.0 JSW =0.0
   + PB =0.65 PHP =0.8 M =0.2994
   + FC =0.95 FCS =0.4 MJSW=0.5
   + TT =2.446e-9 BV =4.65 RS =19
   + IS =1.485e-11 CJO =1.09e-9 CJP =0.0
   + PJ =0.0 N =1.615 IK =0.0
   + IKR =1.100e-2 IBV =2.00e-2
```
Using Junction Model Parameters

The .MODEL statement is referenced by the diode element statement. The .MODEL statement contains parameters that specify the type of diode model used (LEVEL 1, 2, or 3), as well as DC, capacitance, temperature, resistance, geometric, and noise parameters.

Table 15-2: Junction Diode Model Parameters (LEVEL 1 and LEVEL 3)

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>model type</td>
<td>LEVEL</td>
</tr>
<tr>
<td>DC parameters</td>
<td>IBV, IK, IKR, IS, ISW, N, RS, VB, RS</td>
</tr>
<tr>
<td>geometric junction</td>
<td>AREA, M, PJ</td>
</tr>
<tr>
<td>geometric capacitance</td>
<td>L, LM, LP, SHRINK, W, WM, WP, XM, XOJ, XOM, XP, XW</td>
</tr>
<tr>
<td>capacitance</td>
<td>CJ, CJP, FC, FCS, M, MJSW, PB, PHP, TT</td>
</tr>
<tr>
<td>noise</td>
<td>AK, KF</td>
</tr>
</tbody>
</table>
## Setting Junction DC Parameters in LEVEL 1 and 3

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA</td>
<td></td>
<td>1.0</td>
<td>Junction area&lt;br&gt;For LEVEL=1&lt;br&gt;(\text{AREA}<em>{\text{eff}} = \text{AREA} \cdot \text{M}, \text{unitless})&lt;br&gt;For LEVEL=3&lt;br&gt;(\text{AREA}</em>{\text{eff}} = \text{AREA} \cdot \text{SCALM}^2 \cdot \text{SHRINK}^2 \cdot \text{M})&lt;br&gt;unit = meter(^2)&lt;br&gt;If you specify W and L:&lt;br&gt;(\text{AREA}_{\text{eff}} = \text{Weff} \cdot \text{Leff} \cdot \text{M}) unit = meter(^2)</td>
</tr>
<tr>
<td>EXPLI</td>
<td>amp/AREAeff</td>
<td>1e15</td>
<td>Current explosion model parameter. The PN junction characteristics above the explosion current are linear, with the slope at the explosion point, which increases simulation speed and improves convergence.&lt;br&gt;(\text{EXPLI}<em>{\text{eff}} = \text{EXPLI} \cdot \text{AREA}</em>{\text{eff}})</td>
</tr>
<tr>
<td>IB</td>
<td>amp</td>
<td>1.0e-3</td>
<td>Current at breakdown voltage&lt;br&gt;For LEVEL=3&lt;br&gt;(\text{IBV}<em>{\text{eff}} = \text{IBV} \cdot \text{AREA}</em>{\text{eff}} / \text{SCALM}^2)</td>
</tr>
<tr>
<td>IBV</td>
<td>amp</td>
<td>1.0e-3</td>
<td>Current at breakdown voltage&lt;br&gt;For LEVEL=3&lt;br&gt;(\text{IBV}<em>{\text{eff}} = \text{IBV} \cdot \text{AREA}</em>{\text{eff}} / \text{SCALM}^2)</td>
</tr>
<tr>
<td>IK (IKF, JBF)</td>
<td>amp/AREAeff</td>
<td>0.0</td>
<td>Forward knee current (intersection of the high- and low-current asymptotes)&lt;br&gt;(\text{IK}<em>{\text{eff}} = \text{IK} \cdot \text{AREA}</em>{\text{eff}}).</td>
</tr>
<tr>
<td>IKR (JBR)</td>
<td>amp/AREAeff</td>
<td>0.0</td>
<td>Reverse knee current (intersection of the high- and low-current asymptotes)&lt;br&gt;(\text{IKR}<em>{\text{eff}} = \text{IKR} \cdot \text{AREA}</em>{\text{eff}}).</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------</td>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IS (JS)</td>
<td>amp/AREAeff</td>
<td>LEVEL 1=1.0e-14</td>
<td>If you use an IS value less than EPSMIN, the program resets the value of IS to EPSMIN and displays a warning. EPSMIN default=1.0e-28 If the value of IS is too large, the program displays a warning. For LEVEL=1 ISeff = AREAeff · IS For LEVEL=3 ISeff = AREAeff · IS/SCALM²</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LEVEL 3=0.0</td>
<td></td>
</tr>
<tr>
<td>JSW (ISP)</td>
<td>amp/PJeff</td>
<td>0.0</td>
<td>Sidewall saturation current per unit junction periphery For LEVEL=1 JSWeff = PJeff · JSW For LEVEL=3 JSWeff = PJeff · JSW/SCALM</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td>Default length of diode Leff = L SHRINK SCALM+ XWeff</td>
</tr>
<tr>
<td>LEVEL</td>
<td></td>
<td>1</td>
<td>Diode model selector LEVEL=1 or LEVEL=3 selects junction diode model LEVEL=2 selects Fowler-Nordheim model</td>
</tr>
<tr>
<td>N</td>
<td></td>
<td>1.0</td>
<td>Emission coefficient</td>
</tr>
<tr>
<td>PJ</td>
<td></td>
<td>0.0</td>
<td>Junction periphery For LEVEL=1 PJeff = PJ · M, unitless For LEVEL=3 PJeff = PJ · SCALM M SHRINK, meter If W and L are specified PJeff = (2 · Weff + 2 · Leff) · M, meter</td>
</tr>
</tbody>
</table>
**Using Diodes**

### Specifying Junction Diode Models

**Note:** If you use a diode model for which the AREA is not specified, AREA defaults to 1; then RS has units of ohms. If AREA is specified in the netlist in $m^2$, then the units of RS are ohms/$m^2$.

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| RS           | ohms or ohms/$m^2$ (see note below) | 0.0 | Ohmic series resistance  
For LEVEL=1  
$R_{Seff} = RS/AREA_{eff}$  
For LEVEL=3  
$R_{Seff} = RS \cdot SCALM^2/AREA_{eff}$ |
| SHRINK       |       | 1.0     | Shrink factor |
| VB (BV, VAR, VRB) | V     | 0.0     | Reverse breakdown voltage. 0.0 indicates an infinite breakdown voltage |
| XW           |       |         | Accounts for masking and etching effects  
$X_{Weff} = XW \cdot SCALM$ |
## Setting Junction Capacitance Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJ (CJA, CJO)</td>
<td>F/ AREAeff</td>
<td>0.0</td>
<td>Zero-bias junction capacitance per unit junction bottomwall area</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For LEVEL=1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( CJO_{\text{eff}} = CJO \cdot \text{AREAeff} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For LEVEL=3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( CJ_{\text{eff}} = CJ \cdot \text{AREAeff/SCALM}^2 )</td>
</tr>
<tr>
<td>CJP (CJSW)</td>
<td>F/PJeff</td>
<td>0.0</td>
<td>Zero-bias junction capacitance per unit junction periphery (PJ)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For LEVEL=1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( CJP_{\text{eff}} = CJP \cdot PJeff )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For LEVEL=3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( CJP_{\text{eff}} = CJP \cdot PJeff/SCALM )</td>
</tr>
<tr>
<td>FC</td>
<td></td>
<td>0.5</td>
<td>Coefficient for forward-bias depletion area capacitance formula</td>
</tr>
<tr>
<td>FCS</td>
<td></td>
<td>0.5</td>
<td>Coefficient for the forward-bias depletion periphery capacitance formula</td>
</tr>
<tr>
<td>M (EXA, MJ)</td>
<td></td>
<td>0.5</td>
<td>Area junction grading coefficient</td>
</tr>
<tr>
<td>MJSW (EXP)</td>
<td></td>
<td>0.33</td>
<td>Periphery junction grading coefficient</td>
</tr>
<tr>
<td>PB (PHI, VJ, PHA)</td>
<td>V</td>
<td>0.8</td>
<td>Area junction contact potential</td>
</tr>
<tr>
<td>PHP</td>
<td>V</td>
<td>PB</td>
<td>Periphery junction contact potential</td>
</tr>
<tr>
<td>TT</td>
<td>s</td>
<td>0.0</td>
<td>Transit time</td>
</tr>
</tbody>
</table>
### Setting Metal and Poly Capacitor Parameters for LEVEL=3

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM</td>
<td>m</td>
<td>0.0</td>
<td>Use this parameter when LM is not specified in the element statement. ( L_{\text{Meff}} = L_{M} \cdot SCALM \cdot SHRINK )</td>
</tr>
<tr>
<td>LP</td>
<td>m</td>
<td>0.0</td>
<td>Use this parameter if LP is not specified in the element statement. ( L_{\text{Peff}} = L_{P} \cdot SCALM \cdot SHRINK )</td>
</tr>
<tr>
<td>WM</td>
<td>m</td>
<td>0.0</td>
<td>Use this parameter if WM is not specified in the element statement. ( W_{\text{Meff}} = W_{M} \cdot SCALM \cdot SHRINK )</td>
</tr>
<tr>
<td>WP</td>
<td>m</td>
<td>0.0</td>
<td>Use this parameter if WP is not specified in the element statement. ( W_{\text{Peff}} = W_{P} \cdot SCALM \cdot SHRINK )</td>
</tr>
<tr>
<td>XM</td>
<td>m</td>
<td>0.0</td>
<td>XM accounts for masking and etching effects: ( X_{\text{Meff}} = X_{M} \cdot SCALM ).</td>
</tr>
<tr>
<td>XOI</td>
<td></td>
<td>10k</td>
<td>Thickness of the poly to bulk oxide</td>
</tr>
<tr>
<td>XOM</td>
<td>Å</td>
<td>10k</td>
<td>Thickness of the metal to bulk oxide</td>
</tr>
<tr>
<td>XP</td>
<td>m</td>
<td>0.0</td>
<td>Accounts for masking and etching effects ( X_{\text{Peff}} = X_{P} \cdot SCALM )</td>
</tr>
</tbody>
</table>

### Setting Noise Parameters for LEVEL=1 and 3

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td></td>
<td>1.0</td>
<td>Flicker noise exponent</td>
</tr>
<tr>
<td>KF</td>
<td></td>
<td>0.0</td>
<td>Flicker noise coefficient</td>
</tr>
</tbody>
</table>
Providing Geometric Scaling for Diode Models

**LEVEL=1 Scaling**

Scaling for LEVEL 1 involves the use of the AREA and M Element parameters. The element and model parameters scaled with AREA and M include: IK, IKR, JS, CJO, and RS. For AREA and M, default=1

This element is not a geometric model because both the area (AREA) and periphery (PJ) are measured in dimensionless values. These parameters are not affected by the SCALE and SCALM options.

The periphery junction parameter is multiplied by M, the multiplier parameter, to scale the dimensionless periphery junction.

\[ P_{Jeff} = PJ \cdot M \]

PJ\_eff is then used to scale CJP, the zero-bias junction capacitance, and the sidewall saturation current, JSW.

\[ C_{JPeff} = P_{Jeff} \cdot CJP \]
\[ J_{SWeff} = P_{Jeff} \cdot JSW \]

AREA and M are used to obtain AREA\_eff.

\[ A_{REAeff} = AREA \cdot M \]

CJO, IK, IKR, IBV, and IS are multiplied by AREA\_eff to obtain their effective scaled values. RS, however, is divided by AREA\_eff.

\[ I_{Keff} = A_{REAeff} \cdot IK \]
\[ I_{KReff} = A_{REAeff} \cdot IKR \]
\[ I_{BVeff} = A_{REAeff} \cdot IBV \]
\[ I_{Seff} = A_{REAeff} \cdot IS \]
\[ R_{Seff} = RS / A_{REAeff} \]
\[ C_{JOeff} = CJO \cdot A_{REAeff} \]

**LEVEL=3 Scaling**

LEVEL 3 scaling is affected by SCALM, SCALE, SHRINK, and M.

The LEVEL 3 element parameters affected by SCALE include:

AREA, LM, LP, PJ, WM, WP, W, L
The model parameters affected by SCALM include:
AREA, IBV, IK, IKR, IS, PJ, JSW, RS, CJO, CJP, LM, LP, WP, XM, XP, W, L, XW

If you include the AREA as either an element parameter or a model parameter, the program uses SCALE or SCALM. The following equations use the AREA element parameter, instead of the AREA model parameter.

If the AREA and PJ model parameters are specified and the element is not, use SCALM as the scaling factor instead of SCALE. The scaled effective area and periphery junction element parameters are determined by:

\[
\text{AREAeff} = \text{AREA} \cdot M \cdot \text{SCALE}^2 \cdot \text{SHRINK}^2 \\
\text{PJeff} = \text{PJ} \cdot \text{SCALE} \cdot M \cdot \text{SHRINK}
\]

or, if W and L are specified:

\[
\text{AREAeff} = \text{Weff} \cdot \text{Leff} \cdot M \\
\text{PJeff} = (2 \cdot \text{Weff} + 2 \cdot \text{Leff}) \cdot M
\]

where

\[
\text{Weff} = W \cdot \text{SCALE} \cdot \text{SHRINK} + \text{XWeff} \\
\text{Leff} = L \cdot \text{SCALE} \cdot \text{SHRINK} + \text{XWeff}
\]

To find the value of JSWeff and CJPeff use the formula:

\[
\text{JSWeff} = \text{PJeff} \cdot (\text{JSW} / \text{SCALM}) \\
\text{CJPeff} = \text{PJeff} \cdot (\text{CJP} / \text{SCALM})
\]

To determine the polysilicon and metal capacitor dimensions, multiply each by SCALE or by SCALM if specified as model parameters.

\[
\text{LMeff} = \text{LM} \cdot \text{SCALE} \cdot \text{SHRINK} \\
\text{WMeff} = \text{WM} \cdot \text{SCALE} \cdot \text{SHRINK} \\
\text{LPeff} = \text{LP} \cdot \text{SCALE} \cdot \text{SHRINK} \\
\text{WPeff} = \text{WP} \cdot \text{SCALE} \cdot \text{SHRINK} \\
\text{XPeff} = \text{XP} \cdot \text{SCALM} \\
\text{XMeff} = \text{XM} \cdot \text{SCALM}
\]

You can determine the effective scaled model parameters, IBeff, IKeff, IKReff, IBVeff, RSeff, and CJO as follows:

\[
\text{IKeff} = \text{AREAeff} \cdot \text{IK}
\]
Defining Diode Models

Diode Current

Figure 15-1 shows the direction of current flow through the diode. Use either I(D1) or I1(D1) syntax to print the diode current.

If the voltage on node1 is 0.6V greater than the voltage on node2, the diode is \textit{forward biased} or turned on. The anode is the p-doped side of a diode, and the cathode is the n-doped side.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{diode_current_convention}
\caption{Diode Current Convention}
\end{figure}
Using Diode Equivalent Circuits

Star-Hspice uses three equivalent circuits in diode analysis: transient, AC, and noise circuits. Components of these circuits form the basis for all element and model equations.

The fundamental component in the DC equivalent circuit is the DC diode current ($i_d$). For noise and AC analyses, the actual $i_d$ current is not used. The partial derivative of $i_d$ with respect to the terminal voltage $v_d$ is used instead. The name for this partial derivative is:

Conductance

$$ g_d = \frac{\partial i_d}{\partial v_d} $$

The drain current ($i_d$) equation accounts for all basic DC effects of the diodes. Star-Hspice assumes capacitance effects to be separate from the $i_d$ equations.

**Figure 15-2: Equivalent Circuit, Diode Transient Analysis**
Figure 15-3: Equivalent Circuit, Diode AC Analysis

Figure 15-4: Equivalent Circuit, Diode AC Noise Analysis
Determining Temperature Effects on Junction Diodes

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

Table 15-3: Junction Diode Temperature Parameters (LEVEL 1 and 3)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistance coefficient</td>
<td>TRS</td>
</tr>
<tr>
<td>Capacitance coefficient</td>
<td>CTA, CTP</td>
</tr>
<tr>
<td>Energy gap</td>
<td>EG, GAP1, GAP2</td>
</tr>
<tr>
<td>Transit time coefficient</td>
<td>TTT1, TTT2</td>
</tr>
<tr>
<td>Reference temperature</td>
<td>TREF</td>
</tr>
<tr>
<td>Temperature selectors</td>
<td>TLEV, TLEVC</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>TM1, TM2, TPB, TPHP</td>
</tr>
<tr>
<td>Saturation current</td>
<td>XT1</td>
</tr>
</tbody>
</table>

Setting Temperature Effect Parameters LEVEL=1 and 3

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTA (CTC)</td>
<td>1/°</td>
<td>0.0</td>
<td>Temperature coefficient for area junction capacitance (CJ). Set parameter TLEVC to 1 to let CTA override default temperature coefficient.</td>
</tr>
<tr>
<td>CTP</td>
<td>1/°</td>
<td>0.0</td>
<td>Temperature coefficient for periphery junction capacitance (CJP). Set TLEVC to 1 to let CTP override default temperature coefficient.</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>EG</td>
<td>eV</td>
<td>1.11</td>
<td>Energy gap for pn junction diode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.16</td>
<td>For TLEV=0, 1, default=1.11, for TLEV=2, default=1.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.17</td>
<td>1.17 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.69</td>
<td>0.69 - Schottky barrier diode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.67</td>
<td>0.67 - germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.52</td>
<td>1.52 - gallium arsenide</td>
</tr>
<tr>
<td>GAP1</td>
<td>eV/°</td>
<td>7.02e-4</td>
<td>7.02e-4 - silicon (old value)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.73e-4</td>
<td>4.73e-4 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.56e-4</td>
<td>4.56e-4 - germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.41e-4</td>
<td>5.41e-4 - gallium arsenide</td>
</tr>
<tr>
<td>GAP2</td>
<td>°</td>
<td>1108</td>
<td>1108 - silicon (old value)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>636</td>
<td>636 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>210</td>
<td>210 - germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>204</td>
<td>204 - gallium arsenide</td>
</tr>
<tr>
<td>TCV</td>
<td>1/°</td>
<td>0.0</td>
<td>Breakdown voltage temperature coefficient</td>
</tr>
<tr>
<td>TLEV</td>
<td></td>
<td>0.0</td>
<td>Temperature equation selector for diode; interacts with TLEVC</td>
</tr>
<tr>
<td>TLEVC</td>
<td></td>
<td>0.0</td>
<td>Level selector for diode temperature, junction capacitances and contact potentials; interacts with TLEV</td>
</tr>
<tr>
<td>TM1</td>
<td>1/°</td>
<td>0.0</td>
<td>First-order temperature coefficient for MJ</td>
</tr>
<tr>
<td>TM2</td>
<td>1/°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for MJ</td>
</tr>
<tr>
<td>TPB (TVJ)</td>
<td>V/°</td>
<td>0.0</td>
<td>Temperature coefficient for PB. Set parameter TLEVC to 1 or 2 to enable TPB to override default temperature compensation.</td>
</tr>
<tr>
<td>TPHP</td>
<td>V/°</td>
<td>0.0</td>
<td>Temperature coefficient for PHP. Set parameter TLEVC to 1 or 2 to enable TPHP to override default temperature compensation.</td>
</tr>
</tbody>
</table>
### Using Diodes

#### Specifying Junction Diode Models

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREF</td>
<td></td>
<td>25.0</td>
<td>Model reference temperature (LEVEL 1 or 3 only)</td>
</tr>
<tr>
<td>TRS</td>
<td>1/°C</td>
<td>0.0</td>
<td>Resistance temperature coefficient</td>
</tr>
<tr>
<td>TTT1</td>
<td>1/°C</td>
<td>0.0</td>
<td>First order temperature coefficient for TT</td>
</tr>
<tr>
<td>TTT2</td>
<td>1/°²</td>
<td>0.0</td>
<td>Second order temperature coefficient for TT</td>
</tr>
<tr>
<td>XTI</td>
<td></td>
<td>3.0</td>
<td>Saturation current temperature exponent. Set XTI=3.0 for silicon-diffused junction. Set XTI=2.0 for Schottky barrier diode.</td>
</tr>
</tbody>
</table>
Using Junction Diode Equations

Table 15-4 shows the diode equation variable definition.

**Table 15-4: Equation Variable Definitions**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>cd</td>
<td>total diode capacitance</td>
</tr>
<tr>
<td>f</td>
<td>frequency</td>
</tr>
<tr>
<td>gd</td>
<td>diode conductance</td>
</tr>
<tr>
<td>id</td>
<td>diode DC current</td>
</tr>
<tr>
<td>id1</td>
<td>current without high level injection</td>
</tr>
<tr>
<td>ind</td>
<td>diode equivalent noise current</td>
</tr>
<tr>
<td>inrs</td>
<td>series resistor equivalent noise current</td>
</tr>
<tr>
<td>vd</td>
<td>voltage across the diode</td>
</tr>
</tbody>
</table>

Table 15-5 shows the equation quantity definition:

**Table 15-5: Equation Quantity Definition**

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>tox</td>
<td>3.453143e-11 F/m</td>
</tr>
<tr>
<td>k</td>
<td>1.38062e-23 (Boltzmann’s constant)</td>
</tr>
<tr>
<td>q</td>
<td>1.60212e-19 (electron charge)</td>
</tr>
<tr>
<td>t</td>
<td>temperature in °Kelvin</td>
</tr>
<tr>
<td>Δt</td>
<td>t - tnom</td>
</tr>
<tr>
<td>tnom</td>
<td>nominal temperature of parameter measurements in °Kelvin</td>
</tr>
<tr>
<td>vt(t)</td>
<td>k · t/q: thermal voltage</td>
</tr>
</tbody>
</table>
Using Junction DC Equations

The basic diode is modeled in three regions:
- Forward bias
- Reverse bias
- Breakdown regions

For a forward bias diode, the anode is more positive than the cathode. The diode is turned on and conducts above 0.6 volts. Set the model parameter RS to limit conduction current. As the forward bias voltage increases past 0.6 volts, the limiting resistor prevents the value of the diode current from becoming too high and the solution from converging.

Forward Bias: \( v_d > -10 \cdot v_t \)

\[
 id = I_{Seff} \cdot \left( e^{\frac{v_d}{N \cdot v_t}} - 1 \right)
\]

\( v_d = v_{node1} - v_{node2} \)

For reverse bias, the anode (node1) is more negative than the cathode. The diode is turned off, and conducts a small leakage current.

Reverse Bias: \( BV_{eff} < v_d < -10 \cdot v_t \)

\( id = -I_{Seff} \)

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

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_t(t_{nom}) )</td>
<td>( k \cdot t_{nom}/q ): thermal voltage</td>
</tr>
</tbody>
</table>
Note: BV is always described as a positive number.

Breakdown: \( v_d < - BV_{eff} \)

\[ id = -I_{Seff} \cdot e^{\frac{v_d + BV_{eff}}{N \cdot vt}} \]

The BV parameter is adjusted as follows to obtain \( BV_{eff} \):

\[ i_{break} = -I_{Seff} \cdot \left( e^{\frac{-BV}{N \cdot vt}} - 1 \right) \]

If \( IBV_{eff} > i_{break} \), then,

\[ BV_{eff} = BV - N \cdot vt \cdot \ln\left( \frac{IBV_{eff}}{i_{break}} \right) \]

Otherwise,

\[ IBV_{eff} = i_{break} \]

Most diodes do not behave as ideal diodes. The parameters \( IK \) and \( IKR \) are called high-level injection parameters. They tend to limit the exponential current increase.

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

Forward Bias

\[ id = \frac{id_1}{1 + \left( \frac{id_1}{IK_{eff}} \right)^{1/2}} \]
Reverse Bias

\[ id = \frac{id_1}{1 + \left(\frac{id_1}{IKReff}\right)^{1/2}} \]

where \( id_1 \) is

For \( vd \geq -BV_{eff} \):

\[ id_1 = IS_{eff} \cdot \left(e^{\frac{vd}{Nvt}} - 1\right) \]

Otherwise:

\[ id_1 = IS_{eff} \cdot \left(e^{\frac{vd}{Nvt}} - 1\right) - IS_{eff} \cdot \left[e^{\frac{vd + BV_{eff}}{Nvt}} - 1\right] \]

You can estimate the reverse saturation current \( IS \), emission coefficient \( N \), and model parameter \( RS \) from DC measurements of the forward biased diode characteristics. You can determine \( N \) from the slope of the diode characteristic in the ideal region. In most cases, the emission coefficient is the value of unit, but is closer to 2 for MOS diodes.

In practice, at higher levels of bias, the diode current deviates from the ideal exponential characteristic. This deviation is due to the presence of ohmic resistance in the diode as well as high-level injection effects. The deviation of the actual diode voltage from the ideal exponential characteristic at a specific current determines the value of \( RS \). In practice, \( RS \) is estimated at several values of \( id \) and averaged, since the value of \( RS \) depends upon diode current.

Using Diode Capacitance Equations

The diode capacitance is modeled by \( cd \) in Figure 15-2. The capacitance, \( cd \), is a combination of diffusion capacitance, \( (cdiff) \), depletion capacitance, \( (cdep) \), metal, \( (cmetal) \), and poly capacitances, \( (cpoly) \).

\[ cd = cdiff + cdep + cmetal + cpoly \]
Using Diffusion Capacitance Equations

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

\[ \text{cdiff} = TT \cdot \frac{\partial id}{\partial vd} \]

Using Depletion Capacitance Equations

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

**DCAP=1**

The junction bottom area capacitance formula is:

\[ \text{vd} < FC \cdot PB \]

\[ cdepa = CJeff \cdot \left(1 - \frac{vd}{PB}\right)^{-MJ} \]

\[ \text{vd} \geq FC \cdot PB \]

\[ cdepa = CJeff \cdot \frac{1 - FC \cdot (1 + MJ) + MJ \cdot \frac{vd}{PB}}{(1 - FC)^{(1 + MJ)}} \]

The junction periphery capacitance formula is:

\[ \text{vd} < FCS \cdot PHP \]

\[ cdepp = CJPeff \cdot \left(1 - \frac{vd}{PHP}\right)^{-MJSW} \]
Using Diodes

\[ \text{vd} \quad \text{FCS} \cdot \text{PHP} \]
\[
c\text{depp} = C\text{JPeff} \cdot \frac{1 - \text{FCS} \cdot (1 + \text{MJSW}) + \text{MJSW} \cdot \frac{\text{vd}}{\text{PHP}}}{(1 - \text{FCS})^{\left(1 + \text{MJSW}\right)}}
\]

then,
\[ c\text{dep} = c\text{depa} + c\text{depp} \]

**DCAP=2 (default)**

The total depletion capacitance formula is:

\[ \text{vd} < 0 \]
\[
c\text{dep} = C\text{Jeff} \cdot \left(1 - \frac{\text{vd}}{\text{PB}}\right)^{-\text{MJ}} + C\text{JPeff} \cdot \left(1 - \frac{\text{vd}}{\text{PHP}}\right)^{-\text{MJSW}}
\]

\[ \text{vd} = 0 \]
\[
c\text{dep} = C\text{Jeff} \cdot \left(1 + \text{MJ} \cdot \frac{\text{vd}}{\text{PB}}\right) + C\text{JPeff} \cdot \left(1 + \text{MJSW} \cdot \frac{\text{vd}}{\text{PHP}}\right)
\]

**DCAP=3**

Limits peak depletion capacitance to \( \text{FC} \cdot \text{CGDeff} \) or \( \text{FC} \cdot \text{CGSeff} \), with proper fall-off when forward bias exceeds \( \text{PB} \) (\( \text{FC} \geq 1 \)).

**Metal and Poly Capacitance Equations (LEVEL=3 Only)**

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

\[
c\text{metal} = \left(\frac{\varepsilon_{\text{ox}}}{\text{XOI}}\right) \cdot (\text{WPeff} + \text{XPeff}) \cdot (\text{LPeff} + \text{XPeff}) \cdot M
\]

\[
c\text{poly} = \left(\frac{\varepsilon_{\text{ox}}}{\text{XOM}}\right) \cdot (\text{WMeff} + \text{XMeff}) \cdot (\text{LMeff} + \text{XMeff}) \cdot M
\]
Using Noise Equations

Figure 15-4 shows the noise model for a diode. An independent current source, $inrs$, in parallel with the resistor models the thermal noise generated by a resistor. To determine the value of $inrs$, use the equation:

$$inrs = \left( \frac{4 \cdot k \cdot t}{R_{\text{eff}}} \right)^{1/2}$$

The unit of $inrs$ is Amp/(Hz)$^{1/2}$.

The shot and flicker noise of the diode are modeled by the current source $ind$, which is defined by:

$$ind = \left( 2 \cdot q \cdot id + \frac{KF \cdot idAF}{f} \right)^{1/2}$$

Temperature Compensation Equations

This section describes the temperature compensation equations.

Energy Gap Temperature Equations

Use the following equations to determine energy gap for temperature compensation.

$TLEV=0$ or 1

$$egnom = 1.16 - 7.02e^{-4} \cdot \frac{tnom^2}{tnom + 1108.0}$$

$$eg(t) = 1.16 - 7.02e^{-4} \cdot \frac{t^2}{t + 1108.0}$$
Using Diodes

Using Junction Diode Equations

**TLEV=2**

\[ e_{gnom} = EG - GAP1 \cdot \frac{tnom^2}{tnom + GAP2} \]

\[ e_g(t) = EG - GAP1 \cdot \frac{t^2}{t + GAP2} \]

**Leakage Current Temperature Equations**

\[ JS(t) = JS \cdot e^{\frac{facln}{N}} \]

\[ JSW(t) = JSW \cdot e^{\frac{facln}{N}} \]

**TLEV=0 or 1**

\[ facln = \frac{EG}{vt(tnom)} - \frac{EG}{vt(t)} + XTI \cdot \ln \left( \frac{t}{tnom} \right) \]

**TLEV=2**

\[ facln = \frac{e_{gnom}}{vt(tnom)} - \frac{e_g(t)}{vt(t)} + XTI \cdot \ln \left( \frac{t}{tnom} \right) \]

**Breakdown Voltage Temperature Equations**

**TLEV=0**

\[ BV(t) = BV - TCV \cdot \Delta t \]

**TLEV=1 or 2**

\[ BV(t) = BV \cdot (1 - TCV \cdot \Delta t) \]

**Transit Time Temperature Equations**

\[ TT(t) = TT \cdot (1 + TTT1 \cdot \Delta t + TTT2 \cdot \Delta t^2) \]
Contact Potential Temperature Equations

**TLEVC=0**

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

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

**TLEVC=1 or 2**

\[
P_B(t) = PB - TPB \cdot \Delta t
\]

\[
PHP(t) = PHP - TPHP \cdot \Delta t
\]

**TLEVC=3**

\[
P_B(t) = PB + dpbdt \cdot \Delta t
\]

\[
PHP(t) = PHP + dphpdt \cdot \Delta t
\]

where TLEV=0 or 1

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

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

and TLEV=2

\[
dpbdt = \frac{-\left[ egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left( 2 - \frac{tnom}{tnom + GAP2} \right) - PB \right]}{tnom}
\]

\[
dphpdt = \frac{-\left[ egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left( 2 - \frac{tnom}{tnom + GAP2} \right) - PHP \right]}{tnom}
\]
Junction Capacitance Temperature Equations

**TLEVC=0**

\[ CJ(t) = CJ \cdot \left[ 1 + MJ \cdot \left( 4.0e-4 \cdot \Delta t - \frac{PB(t)}{PB} + 1 \right) \right] \]

\[ CJSW(t) = CJSW \cdot \left[ 1 + MJSW \cdot \left( 4.0e-4 \cdot \Delta t - \frac{PHP(t)}{PHP} + 1 \right) \right] \]

**TLEVC=1**

\[ CJ(t) = CJ \cdot (1 + CTA \cdot \Delta t) \]

\[ CJSW(t) = CJSW \cdot (1 + CTP \cdot \Delta t) \]

**TLEVC=2**

\[ CJ(t) = CJ \cdot \left( \frac{PB}{PB(t)} \right)^{MJ} \]

**TLEVC=3**

\[ CJ(t) = CJ \cdot \left( 1 - 0.5 \cdot dpbdt \cdot \frac{\Delta t}{PB} \right) \]

\[ CJSW(t) = CJSW \cdot \left( 1 - 0.5 \cdot dphpdt \cdot \frac{\Delta t}{PHP} \right) \]

*Note: In the above equation MJ is not MJ(t).*
Grading Coefficient Temperature Equation

\[ MJ(t) = MJ \cdot (1 + TM1 \cdot \Delta t + TM2 \cdot \Delta t^2) \]

Resistance Temperature Equations

\[ RS(t) = RS \cdot (1 + TRS \cdot \Delta t) \]
Using the Junction Cap Model

This section describes how to use the junction cap model statement.

General Syntax

The general syntax for including a diode element in a Star-Hspice netlist is:

```
Dxxx nodeplus nodeminus modelname <<area=>val>
+ <<peri=>val> <<pgate>=val> <<dtemp>=val>
+ <<off>=val> <<IC=>val> <<m=>val>
```

- **Dxxx**: Diode element name. Must begin with “D”
- **nodeplus**: Positive terminal (anode) node name. The series resistor of the equivalent circuit is attached to this terminal
- **nodeminus**: Negative terminal (cathode) node name
- **modelname**: Diode model name reference
- **area**: Diode area. In the model card, it can be used by AB
- **peri**: Length of the side-wall of the diffusion area AB which is not under the gate. In the model card, it is used by LS
- **pgate**: Length of the side-wall of the diffusion area AB which is under the gate. In the model card, it is used by LG
- **off**: Sets initial condition to OFF for this element in DC analysis. The default is ON
- **M**: Multiplier to simulate multiple diodes in parallel. All currents, capacitances and resistances are affected by setting M. Default=1
Juncap Model Syntax

The juncap model statement syntax is:

```
.MODEL modelname D level=4 <keyword=val>
```

- **ic**: Initial voltage across the diode element. This value is used when the UIC option is present in the .tran statement and is overridden by the .ic statement.

- **Dtemp**: The difference between the element temperature and circuit temperature in celsius. Default=0.0

- **.option list**: Prints the updated temperature parameters for juncap diode model

Examples

```
.model MD D level=4
+AB=2E-12 LS=2E-6 LG=1.3E-6 DTA=0 TR=30 VR=0.3
+JSGBR=1.2e-3 JSDBR=1.3e-3 JSGSR=1.1e-3
+JSDSR=1.3e-3 JSGGR=1.4e-3 JSDGR=1.4e-3 NB=1.6
+NS=1.3
+NG=1.3 VB=0.9 CJBR=1.2e-12 CJSR=1.2e-12
+CJGR=1.3e-12 VDBR=1.6 VDSR=1.3 VGDR=1.2 PB=0.5
+PS=0.6 PG=0.4
```
Setting Juncap Model Parameters

Table 15-6: Juncap Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Units</th>
<th>Default</th>
<th>Clip Low</th>
<th>High</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>M²</td>
<td>1e-12</td>
<td>0.0</td>
<td></td>
<td>Diffusion area</td>
</tr>
<tr>
<td>LS</td>
<td>M</td>
<td>1.0e-6</td>
<td>0.0</td>
<td></td>
<td>Length of side-wall of diffusion area AB which is not under gate</td>
</tr>
<tr>
<td>LG</td>
<td>M</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td>Length of side-wall of diffusion area AB which is under gate</td>
</tr>
<tr>
<td>DTA</td>
<td>C</td>
<td>0.0</td>
<td></td>
<td></td>
<td>Temperature offset of Juncap element with respect to TA</td>
</tr>
<tr>
<td>TR</td>
<td>C</td>
<td>25</td>
<td>-273.15</td>
<td></td>
<td>Temperature at which parameters have been determined</td>
</tr>
<tr>
<td>VR</td>
<td>V</td>
<td>0.0</td>
<td></td>
<td></td>
<td>Voltage at which parameters have been determined</td>
</tr>
<tr>
<td>JSGBR</td>
<td>Am²</td>
<td>1.0E-3</td>
<td>0.0</td>
<td></td>
<td>Bottom saturation-current density due to electron-hole generation at V=VR</td>
</tr>
<tr>
<td>JSDBR</td>
<td>Am²</td>
<td>1.0E-3</td>
<td>0.0</td>
<td></td>
<td>Bottom saturation-current density due to diffusion from back contact</td>
</tr>
<tr>
<td>JSGSR</td>
<td>Am²</td>
<td>1.0E-3</td>
<td>0.0</td>
<td></td>
<td>Sidewall saturation-current density due to electron-hole generation at V=VR</td>
</tr>
<tr>
<td>JSDSR</td>
<td>Am²</td>
<td>1.0E-3</td>
<td>0.0</td>
<td></td>
<td>Sidewall saturation-current density due to diffusion from back contact</td>
</tr>
<tr>
<td>JSGGR</td>
<td>Am²</td>
<td>1.0E-3</td>
<td>0.0</td>
<td></td>
<td>Gate edge saturation current density due to electron-hole generation at V=VR</td>
</tr>
</tbody>
</table>
### Table 15-6: Juncap Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Units</th>
<th>Default</th>
<th>Clip Low</th>
<th>High</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSDGR</td>
<td>Am⁻²</td>
<td>1.0E-3</td>
<td>0.0</td>
<td></td>
<td>Gate edge saturation current density due to diffusion from back contact</td>
</tr>
<tr>
<td>JSGGR</td>
<td>Am⁻²</td>
<td>1.0E-3</td>
<td>0.0</td>
<td></td>
<td>Gate edge saturation current density due to electron-hole generation at V=VR</td>
</tr>
<tr>
<td>JSDGR</td>
<td>Am⁻²</td>
<td>1.0E-3</td>
<td>0.0</td>
<td></td>
<td>Gate edge saturation current density due to diffusion from back contact</td>
</tr>
<tr>
<td>NB</td>
<td></td>
<td>1.0</td>
<td>0.1</td>
<td></td>
<td>Emission coefficient of the bottom forward current</td>
</tr>
<tr>
<td>NS</td>
<td></td>
<td>1.0</td>
<td>0.1</td>
<td></td>
<td>Emission coefficient of the sidewall forward current</td>
</tr>
<tr>
<td>NG</td>
<td></td>
<td>1.0</td>
<td>0.1</td>
<td></td>
<td>Emission coefficient of the gate edge forward current</td>
</tr>
<tr>
<td>VB</td>
<td>V</td>
<td>0.9</td>
<td></td>
<td></td>
<td>Reverse breakdown voltage</td>
</tr>
<tr>
<td>CJBR</td>
<td>Fm⁻²</td>
<td>1.0E-12</td>
<td>0.0</td>
<td></td>
<td>Bottom junction capacitance at V=VR</td>
</tr>
<tr>
<td>CJSR</td>
<td>Fm⁻²</td>
<td>1.0E-12</td>
<td>0.0</td>
<td></td>
<td>Sidewall junction capacitance at V=VR</td>
</tr>
<tr>
<td>CJGR</td>
<td>Fm⁻²</td>
<td>1.0E-12</td>
<td>0.0</td>
<td></td>
<td>Gate edge junction capacitance at V=VR</td>
</tr>
<tr>
<td>VDBR</td>
<td>V</td>
<td>1.00</td>
<td>0.05</td>
<td></td>
<td>Diffusion voltage of the bottom junction at T=TR</td>
</tr>
<tr>
<td>VDSR</td>
<td>V</td>
<td>1.00</td>
<td>0.05</td>
<td></td>
<td>Diffusion voltage of the sidewall junction at T=TR</td>
</tr>
<tr>
<td>VDGR</td>
<td>V</td>
<td>1.00</td>
<td>0.05</td>
<td></td>
<td>Diffusion voltage of the gate edge junction</td>
</tr>
<tr>
<td>PB</td>
<td></td>
<td>0.40</td>
<td>0.05</td>
<td></td>
<td>Bottom junction grading coefficient</td>
</tr>
</tbody>
</table>
This section summarizes the elementary physics of a junction diode. Refer to semiconductor textbooks for additional information.

Generally, the current voltage characteristics can be represented as follows:

\[
J = \left\{ J_d \left( n_i^2 + (J_g(n_i, V)) \right) \right\} \cdot \left[ \exp \left( \frac{qV}{kT} \right) - 1 \right]
\]

\[
n_i \sim T^{\frac{3}{2}} \cdot \exp \left( \frac{-E_g}{2kT} \right)
\]

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(J)</td>
<td>Am(^{-2})</td>
<td>Total reverse current density</td>
</tr>
<tr>
<td>(J_d)</td>
<td>Am(^{-2})</td>
<td>Diffusion saturation current density</td>
</tr>
<tr>
<td>(J_g)</td>
<td>Am(^{-2})</td>
<td>Generation current density</td>
</tr>
<tr>
<td>(n_i)</td>
<td>m(^{-3})</td>
<td>Intrinsic carrier concentration</td>
</tr>
<tr>
<td>(V)</td>
<td>V</td>
<td>Voltage across the diode</td>
</tr>
<tr>
<td>(E_g)</td>
<td>J</td>
<td>Energy gap</td>
</tr>
</tbody>
</table>

**Table 15-6: Juncap Model Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Units</th>
<th>Default</th>
<th>Clip Low</th>
<th>High</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS</td>
<td></td>
<td>0.40</td>
<td>0.05</td>
<td></td>
<td>Sidewall junction grading coefficient</td>
</tr>
<tr>
<td>PG</td>
<td></td>
<td>0.40</td>
<td>0.05</td>
<td></td>
<td>Gate edge junction grading coefficient</td>
</tr>
</tbody>
</table>

**Table 15-7: Current Voltage Characteristics**
For $V < V_D$, the charge of the junction capacitance is described by:

$$Q = Q_j \left[ 1 - \left( 1 - \frac{V}{V_D} \right)^{1-P} \right]$$

### Table 15-7: Current Voltage Characteristics

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>JK$^{-1}$</td>
<td>Boltzmann constant</td>
</tr>
<tr>
<td>$T$</td>
<td>K</td>
<td>Temperature</td>
</tr>
</tbody>
</table>

### Table 15-8: Junction Capacitance Charge

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q$</td>
<td>C</td>
<td>Total diode junction charge</td>
</tr>
<tr>
<td>$Q_j$</td>
<td>C</td>
<td>Junction charge at built-in voltage</td>
</tr>
<tr>
<td>$V$</td>
<td>V</td>
<td>Voltage across the diode</td>
</tr>
<tr>
<td>$V_d$</td>
<td>V</td>
<td>Junction diffusion voltage</td>
</tr>
<tr>
<td>$P$</td>
<td></td>
<td>Junction grading coefficient</td>
</tr>
</tbody>
</table>

### JUNCAP Model Equations

#### JUNCAP Model

The JUNCAP model is intended to describe formed by the source, drain or well-to-bulk junction devices, limited to the case of reverse biasing of these junctions. Similar to the MOS model, the current equations are formulated and AC effects are modeled via charge equations using the quasi-static approximation.
In order to include the effects from differences in the sidewall, bottom, and gate-edge junction profiles, these three contributions are calculated separately in the JUNCAP model.

Both the diffusion and the generation currents are treated in the model, each with individual temperature and voltage dependence.

In the JUNCAP model, a part of the total charge comes from the gate-edge junction very close to the surface. This charge is also included in the MOS model charge equations and is counted twice. However, this results in only a very minor error.

In the next section, the model equations are presented. Correct operation of the model in a circuit simulator environment requires some numerical additions, which are described in the section on implementation. Any fixed capacitance that is present on a node (e.g., metal-1-to-substrate capacitance) must appear in a fixed capacitor statement or must be included in INTCAP. They no longer form the JUNCAP model in contrast to the old NODCAP model.

**Nomenclature**

The following table lists the electrical variable parameters:

**Table 15-9: Electrical Variable Parameters**

<table>
<thead>
<tr>
<th>No</th>
<th>Variable</th>
<th>Programming Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( V_a )</td>
<td>VA</td>
<td>V</td>
<td>Potential applied to the anode</td>
</tr>
<tr>
<td>2</td>
<td>( V_k )</td>
<td>VK</td>
<td>V</td>
<td>Potential applied to the cathode</td>
</tr>
<tr>
<td>3</td>
<td>( I_a )</td>
<td>IA</td>
<td>A</td>
<td>DC current into the anode</td>
</tr>
<tr>
<td>4</td>
<td>( I_k )</td>
<td>IK</td>
<td>A</td>
<td>DC current into the cathode</td>
</tr>
<tr>
<td>5</td>
<td>( Q_a )</td>
<td>QA</td>
<td>C</td>
<td>Charge in the device attributed to the anode</td>
</tr>
<tr>
<td>6</td>
<td>( Q_k )</td>
<td>QK</td>
<td>C</td>
<td>Charge in the device attributed to the cathode</td>
</tr>
</tbody>
</table>
Note: The parameters are listed above in the model card. See “Setting Juncap Model Parameters” on page 15-35.

The following table lists internal variables and parameters:

<table>
<thead>
<tr>
<th>No</th>
<th>Parameter</th>
<th>Programming Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V_{db}</td>
<td>VDB</td>
<td>V</td>
<td>Diffusion voltage of bottom area AB</td>
</tr>
<tr>
<td>2</td>
<td>V_{ds}</td>
<td>VDS</td>
<td>V</td>
<td>Diffusion voltage of Locos-edge LS</td>
</tr>
<tr>
<td>3</td>
<td>V_{dg}</td>
<td>VDG</td>
<td>V</td>
<td>Diffusion voltage of gate-edge LG</td>
</tr>
<tr>
<td>4</td>
<td>C_{jb}</td>
<td>CJB</td>
<td>F</td>
<td>Capacitance of bottom area AB</td>
</tr>
<tr>
<td>5</td>
<td>C_{js}</td>
<td>CJS</td>
<td>F</td>
<td>Capacitance of Locos-edge LS</td>
</tr>
<tr>
<td>6</td>
<td>C_{jg}</td>
<td>CJG</td>
<td>F</td>
<td>Capacitance of gate-edge LG</td>
</tr>
<tr>
<td>7</td>
<td>I_{sdb}</td>
<td>ISDB</td>
<td>A</td>
<td>Diffusion saturation current of bottom area AB</td>
</tr>
<tr>
<td>8</td>
<td>I_{sds}</td>
<td>ISDS</td>
<td>A</td>
<td>Diffusion saturation current of Locos-edge LS</td>
</tr>
<tr>
<td>9</td>
<td>I_{sdg}</td>
<td>ISDG</td>
<td>A</td>
<td>Diffusion saturation current of gate-edge LG</td>
</tr>
<tr>
<td>10</td>
<td>I_{sgb}</td>
<td>ISGB</td>
<td>A</td>
<td>Generation saturation current of bottom area AB</td>
</tr>
<tr>
<td>11</td>
<td>I_{sgs}</td>
<td>ISGS</td>
<td>A</td>
<td>Generation saturation current of Locos-edge LS</td>
</tr>
<tr>
<td>12</td>
<td>I_{sgg}</td>
<td>ISGG</td>
<td>A</td>
<td>Generation saturation current of gate-edge LG</td>
</tr>
<tr>
<td>13</td>
<td>T_{a}</td>
<td>TA</td>
<td>C</td>
<td>Ambient circuit temperature</td>
</tr>
<tr>
<td>14</td>
<td>T_{kd}</td>
<td>TKD</td>
<td>K</td>
<td>Absolute temperature of the junction/device</td>
</tr>
<tr>
<td>15</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>Diode bias voltage (V=VA - VK)</td>
</tr>
</tbody>
</table>
ON/OFF Condition

Circuit solution involves a process of successive calculations. The calculations are started from a set of “initial guesses” for the electrical quantities of the non-linear elements. The devices start in the default state.

Example

<table>
<thead>
<tr>
<th>JUNCAP</th>
<th>Default</th>
<th>ON</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_D$</td>
<td>-0.1</td>
<td>0.7</td>
<td>-0.1</td>
</tr>
</tbody>
</table>

DC Operating Point Output

The DC operating point output facility gives information on the state of a device at its operation point.

*Note:* The conductance $G_{\text{min}}$ is connected in parallel to the conductance $G$. This conductance influences the DC operating output.

Temperature, Geometry and Voltage Dependence

The general scaling rules, which apply to all three components of the JUNCAP model, are:

---

Table 15-10: Internal Variables and Parameters

<table>
<thead>
<tr>
<th>No</th>
<th>Parameter</th>
<th>Programming Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>$I$</td>
<td>$I$</td>
<td>$A$</td>
<td>Total DC current from anode to cathode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>($I = I_A = -I_K$)</td>
</tr>
<tr>
<td>17</td>
<td>$Q$</td>
<td>$Q$</td>
<td>$C$</td>
<td>Total junction charge</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>($Q = Q_A = -Q_K$)</td>
</tr>
</tbody>
</table>
Using the Junction Cap Model

\[ T_{KR} = T_0 + T_R \]
\[ T_{KD} = T_0 + T_A + DT_A \]
\[ V_{TR} = k \cdot \frac{T_{KR}}{q} \]
\[ V_{TD} = K \cdot \frac{T_{KD}}{q} \]

\[ V_{gR} = 1.16 - \frac{(7.02 \cdot 10^4 - 4 \cdot T_{KR} \cdot T_{KR})}{(1108.0 + T_{KR})} \]
\[ V_{gD} = 1.16 - \frac{(7.02 \cdot 10^4 - 4 \cdot T_{KR} \cdot T_{KD})}{(1108.0 + T_{KD})} \]
\[ F_{TD} = \left( \frac{T_{KD}}{T_{KR}} \right)^{1.5} \cdot \exp\left( \frac{V_{gR}}{2 \cdot V_{TR}} - \frac{V_{gD}}{2 \cdot V_{TD}} \right) \]

**Internal Reference**

The internal reference parameters for the bottom component are specified by:

\[ V_{DB} = \frac{V_{DBR} \cdot T_{KD}}{T_{KR} - 2 \cdot V_{TD} \cdot \ln F_{TD}} \]
\[ C_{JB} = C_{JBR} \cdot A_B \cdot \left( \frac{V_{DBR} - V_R}{V_{DB}} \right)^{P_B} \]
\[ I_{SGB} = J_{SGB} \cdot F_{TD} \cdot A_B \cdot \left( \frac{V_{DB}}{V_{DBR} - V_R} \right)^{P_B} \]
\[ I_{SDB} = J_{SDB} \cdot F_{TD} \cdot F_{TD} \cdot A_B \]
Similar formulations hold for the locos-edge and the gate-edge components. Replace the index $B$ by $S$ and $G$, and the area $AB$ by $LS$ and $LG$.

**For the locos-edge:**

\[
V_{DS} = \frac{V_{DSR} \cdot T_{KR}}{T_{KR} - \frac{1}{2} \cdot V_{TD} \cdot T_{KR}}
\]

\[
C_{JS} = C_{JSR} \cdot L_S \cdot \left( \frac{V_{DSR} - V_R}{V_{DS}} \right)^p
\]

\[
I_{SGS} = J_{SGSR} \cdot F_{TD} \cdot L_S \cdot \left( \frac{V_{DS}}{V_{DSR} - V_R} \right)^p
\]

\[
I_{SDS} = J_{SDSR} \cdot F_{TD} \cdot F_{TD} \cdot L_S
\]

**For the gate-edge:**

\[
V_{DG} = \frac{V_{DGR} \cdot T_{KD}}{T_{KR} - \frac{1}{2} \cdot V_{TD} \cdot T_{KR}}
\]

\[
C_{JG} = C_{JGR} \cdot L_G \cdot \left( \frac{V_{DGR} - V_R}{V_{DG}} \right)^p
\]

\[
I_{SGS} = J_{SGGR} \cdot F_{TD} \cdot L_G \cdot \left( \frac{V_{DG}}{V_{DGR} - V_R} \right)^p
\]

\[
I_{SDS} = J_{SDGR} \cdot F_{TD} \cdot F_{TD} \cdot L_G
\]

**Note:** In subsequent sections, we will show the equations only for the bottom component.
JUNCAP Capacitor and Leakage Current Model

In the charge description, the following internal parameter is defined:

\[ Q_{JDB} = C_JB \cdot V_{DB}(1 - P_B) \]

In order to prevent an unlimited increase of the voltage derivative of the charge, the charge description is in two parts: the original power function and a supplemented quadratic function. At the cross-over point between these regions, indicated by \( V_l \), the following parameters are defined:

\[ F_{CB} = 1 - \left( \frac{(1 + P_B)}{3} \right)^{P_a} \]

\[ V_{LB} = F_{CB} \cdot V_{DB} \]

\[ C_{LB} = C_JB(1 - F_{CB})^{-P_a} \]

\[ Q_{LB} = Q_{JDB} \cdot (1 - (1 - F_{CB})^{(1 - P_a)}) \]

\[ Q_{JBV} = Q_{JC B} \cdot \left( \frac{(1 - (1 - V))}{V_{DB}} \right)^{(1 - P_a)} \quad V < V_{LB} \]

\[ Q_{LB} + C_{LB}(V - V_{LB}) \cdot \left( \frac{1 + (P_B(V - V_{LB}))}{2 \cdot V_{DB} \cdot (1 - F_{CB})} \right) \quad V \geq V_{LB} \quad (12.63) \]

Similar expressions exist for the locos-edge and gate-edge charges, \( Q_{jsv} \) and \( Q_{jgv} \).
The total charge characteristic can be described by:

\[ Q = Q_{JBV} + Q_{JSV} + Q_{JGV} \]

Using elementary mathematics, we can derive from Equation 12.63 (above) simple equations for the capacitance of the bottom area:

\[ C_{JBV} = C_{JB} \left( \frac{1}{\left( \frac{1 - \nu}{V_{DB}} \right)} \right)^{p_s} \quad V < V_{LB} \]

\[ C_{LB} + C_{LB} \cdot P_B \left( \frac{V - V_{LB}}{V_{DB} \cdot (1 - F_{CB})} \right) \quad V \geq V_{LB} \]

Similar expressions exist for \( C_{JSV} \) and \( C_{JGV} \).

**Total Capacitance**

The total capacitance can be described by:

\[ C = C_{JBV} + C_{JSV} + C_{JGV} \]

*Bulk to source or bulk to drain diode current.

**Diffusion and Generation Currents**

With the scaled parameters of the preceding section, the diffusion and generation current components can be expressed as:

\[ I_{DB} = I_{SDB} \cdot \exp \left( \frac{V}{(N_B \cdot V_{TD})} - 1 \right) \]

\[ I_{GB} = I_{SGB} \left( \frac{(V_{DB} - V)}{V_{DB}} \right)^{p_s} \cdot \exp \left( \frac{V}{N_B \cdot V_{TD}} - 1 \right) \quad V \leq V_{DB} \]

\[ 0 \quad V > V_{DB} \]
The first relation concerning the diffusion component is valid over the whole operating range. The second relation, describing the generation current, shows an unlimited increase in the derivative of this function at \( V = V_{DB} \). Therefore, the power function is merged at \( V = 0.0 \) with a hyperbolic function in the forward bias range. The exponential part is divided by \( \exp(V/(N_B \cdot V_{TD})) \). This enables a gradual decrease in the generation current component.

The hyperbolic function \( I_HYP = F_{SB}(V + V_{AB})^B \) is used. The parameter \( B \) controls the decrease of the current for voltages \( V > 0.0 \) for all generation components. The value of \( B \) is fixed and set to 2 in the model. The continuity constraints of function and derivative in the merge point lead to the following relations for \( F_{SB} \) and \( V_{AB} \):

\[
V_{AB} = B \cdot \frac{V_{DB}}{P_B}
\]

\[
F_{SB} = I_{SGB} \cdot V_{AB}^B
\]

The generation current voltage characteristic in the forward region becomes:

\[
I_{GB} = F_{SB}/((V + V_{AB})^B) \cdot (1 - \exp(-v)/(N_B \cdot V_{TD}))
\]

**Final Model Equations**

The final model equations for the currents of the bottom area are:

\[
I_{DB} = I_{SDB} \cdot (\exp(V/(N_B \cdot V_{TD})) - 1)
\]

\[
I_{GB} = I_{SGB} \cdot \left(\frac{V_{DB} - V}{V_{DB}}\right)^B \cdot \exp\left(\frac{V}{N_B \cdot V_{TD}}\right) - 1 \quad V \leq 0
\]

\[
I_{SGB} \cdot \left(\frac{V_{ab}}{(V + V_{ab})}\right)^B \cdot \left(1 - \exp\left(-\frac{V}{(N_B \cdot V_{td})}\right)\right) \quad V > 0.0
\]

Similar expressions exist for the locos-edge and gate-edge components.

The total junction current can be expressed as:

\[
I = (I_{DB} + I_{GB}) + (I_{DS} + I_{GS}) + (I_{DG} + I_{GG})
\]
Using the Fowler-Nordheim Diode

The diode model parameter LEVEL=2 selects the Fowler-Nordheim model. Fowler-Nordheim diodes are formed as a metal-insulator-semiconductor or as a semiconductor-insulator-semiconductor layer device. The insulator is sufficiently thin (100 Angstroms) to permit tunneling of carriers. It models electrically alterable memory cells, air-gap switches, and other insulation breakdown devices.

Fowler-Nordheim Diode Model Parameters LEVEL=2

Table 15-11 shows the Fowler-Nordheim diode model parameters for LEVEL 2.

<table>
<thead>
<tr>
<th>Name (alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EF</td>
<td>V/cm</td>
<td>1.0e8</td>
<td>Forward critical electric field</td>
</tr>
<tr>
<td>ER</td>
<td>V/cm</td>
<td>EF</td>
<td>Reverse critical electric field</td>
</tr>
<tr>
<td>JF</td>
<td>amp/V²</td>
<td>1.0e-10</td>
<td>Forward Fowler-Nordheim current coefficient</td>
</tr>
<tr>
<td>JR</td>
<td>amp/V²</td>
<td>JF</td>
<td>Reverse Fowler-Nordheim current coefficient</td>
</tr>
<tr>
<td>L</td>
<td>m</td>
<td>0.0</td>
<td>Length of diode for calculation of Fowler-Nordheim current</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( L_{eff} = L \cdot \text{SCALM} \cdot \text{SHRINK} + X_{Weff} )</td>
</tr>
<tr>
<td>TOX</td>
<td>Å</td>
<td>100.0</td>
<td>Thickness of oxide layer</td>
</tr>
<tr>
<td>W</td>
<td>m</td>
<td>0.0</td>
<td>Width of diode for calculation of Fowler-Nordheim current</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( W_{eff} = W \cdot \text{SCALM} \cdot \text{SHRINK} + X_{Weff} )</td>
</tr>
<tr>
<td>XW</td>
<td>m</td>
<td>0.0</td>
<td>( X_{Weff} = X_{W} \cdot \text{SCALM} )</td>
</tr>
</tbody>
</table>
Using Fowler-Nordheim Diode Equations

The DC characteristics of the Fowler-Nordheim diode are modeled by the following forward and reverse nonlinear current source equations. In the following equations:

\[ \text{AREA}_\text{eff} = \text{Weff} \cdot \text{Leff} \cdot M \]

**Forward Bias**: \( v_d \geq 0 \)

\[ i_d = \text{AREA}_\text{eff} \cdot J_F \cdot \left( \frac{v_d}{\text{TOX}} \right)^2 \cdot e^{- \frac{E_F \cdot \text{TOX}}{v_d}} \]

**Reverse Bias**: \( v_d < 0 \)

\[ i_d = -\text{AREA}_\text{eff} \cdot J_R \cdot \left( \frac{v_d}{\text{TOX}} \right)^2 \cdot e^{\frac{E_R \cdot \text{TOX}}{v_d}} \]

**Fowler-Nordheim Diode Capacitances**

The Fowler-Nordheim diode capacitance is a constant derived from:

\[ c_d = \text{AREA}_\text{eff} \cdot \frac{\varepsilon_{ox}}{\text{TOX}} \]
Converting National Semiconductor Models

National Semiconductor's circuit simulator has a scaled diode model that is not the same as that used by Star-Hspice. To use National Semiconductor circuit models, do the following:

For a subcircuit that consists of the scaled diode model, the subcircuit name must be the same as the name of the model.

The .PARAM statement inside the subcircuit specifies the scaled diode model parameter values. Add a scaled diode model inside the subcircuit, then change the .MODEL mname mtype statement to a .PARAM statement.

Ensure that all the scaled diode elements are preceded by the character X.

Check that every parameter used in the .MODEL statement inside the subcircuit has a value in the .PARAM statement.

Using the Scaled Diode Subcircuit Definition

The scaled diode subcircuit definition converts the National Semiconductor scaled diode model to a form a model usable in Star-Hspice. The .PARAM parameter inside the .SUBCKT represents the .MODEL parameter in the National circuit simulator. Replace the .MODEL mname statement by a .PARAM statement. Change the model name to SDIODE.

Example

An example of scaled diode subcircuit definition is:

```
.SUBCKT SDIODE NP NN SF=1 SCJA=1 SCJP=0 SIS=1 SICS=1 + SRS=1
D NP NN SDIODE
.PARAM IS=1.10E-18 N=1.03 EG=0.8 RS=20.7E3 + CJA=0.19E-15 PHI=0.25 CJP=0.318E-15 + EXA=0.5 EXP=0.325 CTC=6E-4 + TRS=2.15M M=2
*```
.MODEL SDIODE D
  + IS='IS*SIS*SF'  CJA='CJA*SF*SCJA'  CJP='CJP*SF*SCJP'
  + RS='RS*SRS/SF'  EXA=EXA  EXP=EXP
  + N=N  CTA=CTC  CTP=CTC
  + TRS=TRS  TLEV=1  TLEVC=1  xti='m*n'
.ENDS SDIODE

Note that all the parameters used in the following model must have a value that comes from either a .PARAM statement or the .SUBCKT call. The diode statements are then replaced by the call to the subcircuit SDIODE:

XDS 14 1048 SDIODE SIS=67.32 SCJA=67.32 SRS=1.2285E-2
Chapter 16
Using BJT Models

The bipolar-junction transistor (BJT) model in Star-Hspice is an adaptation of the integral charge control model of Gummel and Poon.

The Star-Hspice model extends the original Gummel-Poon model to include several effects at high bias levels. This model automatically simplifies to the Ebers-Moll model when certain parameters (VAF, VAR, IKF, and IKR) are not specified.

This chapter covers the following topics:

- Using BJT Models
- Understanding the BJT Model Statement
- Using BJT Device Equivalent Circuits
- Using BJT Model Equations (NPN and PNP)
- Using BJT Capacitance Equations
- Defining BJT Noise Equations
- Using BJT Temperature Compensation Equations
- Using the BJT Quasi-Saturation Model
- Converting National Semiconductor Models
- Using the VBIC Bipolar Transistor Model
- LEVEL 6 Philips Bipolar Model (MEXTRAM LEVEL 503)
- LEVEL 6 Philips Bipolar Model (MEXTRAM LEVEL 504)
- LEVEL 8 HiCUM Model
Using BJT Models

The BJT model is used to develop BiCMOS, TTL, and ECL circuits. For BiCMOS devices, use the high-current Beta degradation parameters, IKF and IKR, to modify high injection effects. The model parameter SUBS facilitates the modeling of both vertical and lateral geometrics.

Selecting Models

To select a BJT device, use a BJT element and model statement. The element statement references the model statement by the reference model name. The following example uses the reference name MOD1. In this case an NPN model type is used to describe an NPN transistor.

Example

Q3 3 2 5 MOD1 <parameters>
.MODEL MOD1 NPN <parameters>

Parameters can be specified in both element and model statements. The element parameter always overrides the model parameter when a parameter is specified as both. The model statement specifies the type of BJT, for example, NPN or PNP.

Using Control Options

Control options affecting the BJT model are: DCAP, GRAMP, GMIN, and GMINDC. DCAP selects the equation that determines the BJT capacitances. GRAMP, GMIN, and GMINDC place a conductance in parallel with both the base-emitter and base-collector pn junctions. DCCAP invokes capacitance calculations in DC analysis. The BJT control options follow:

<table>
<thead>
<tr>
<th>Function</th>
<th>Control Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacitance</td>
<td>DCAP, DCCAP</td>
</tr>
<tr>
<td>Conductance</td>
<td>GMIN, GMINDC</td>
</tr>
</tbody>
</table>
You can override global depletion capacitance equation selection that uses the .OPTION DCAP=<val> statement in a BJT model by including DCAP=<val> in the BJTs .MODEL statement.

**Convergence**

Adding a base, collector, and emitter resistance to the BJT model improves its convergence. The resistors limit the current in the device so that the forward-biased pn junctions are not overdriven.
Understanding the BJT Model Statement

Syntax

```
.Model mname NPN (<pname1 = val1> ...) ...
```

or

```
.Model mname PNP <pname1 = val1> ...
```

- `mname`: Model name. Elements refer to the model by this name.
- `NPN`: Identifies an NPN transistor model.
- `PNP`: Identifies a PNP transistor model.
- `pname1`: Each BJT model can include several model parameters.

Example

```
.Model t2n2222a NPN
+ ISS= 0.
  XTF= 1.  NS = 1.00000
+ CJS= 0.
  VJS= 0.50000  PTF= 0.
+ MJS= 0.
  EG = 1.10000  AF = 1.
+ ITF= 0.50000
  VTF= 1.00000
+ BR = 40.00000
  IS = 1.6339e-14  VAF= 103.40529
+ VAR= 17.77498
  IKF= 1.00000
+ NE = 1.31919
  IKR= 1.00000  ISC= 3.6856e-13
+ NC = 1.10024
  IRB= 4.3646e-05  NF = 1.00531
+ NR = 1.00688
  RBM= 1.0000e-02  RB = 71.82988
+ RC = 0.42753
  RE = 3.0503e-03  MJE= 0.32339
+ MJC= 0.34700
  VJE= 0.67373  VJC= 0.47372
+ TF = 9.693e-10
  TR = 380.00e-9  CJE= 2.6734e-11
+ CJC= 1.4040e-11
  FC = 0.95000  XCJC= 0.94518
```
Using BJT Basic Model Parameters

To permit the use of model parameters from earlier versions of Star-Hspice, many of the model parameters have aliases, which are included in the model parameter list in “Using BJT Basic DC Model Parameters” on page 16-6. The new name is always used on printouts, even if an alias is used in the model statement.

BJT model parameters are divided into several groups. The first group of DC model parameters includes the most basic Ebers-Moll parameters. This model is effective for modeling low-frequency large-signal characteristics.

Low-current Beta degradation effect parameters ISC, ISE, NC, and NE aid in modeling the drop in the observed Beta, caused by the following mechanisms:

- Recombination of carriers in the emitter-base space charge layer
- Recombination of carriers at the surface
- Formation of emitter-base channels

Low base and emitter dopant concentrations, found in some BIMOS type technologies, use the high-current Beta degradation parameters, IKF and IKR.

Use the base-width modulation parameters, that is, early effect parameters VAF and VAR, to model high-gain, narrow-base devices. The model calculates the slope of the I-V curve for the model in the active region with VAF and VAR. If VAF and VAR are not specified, the slope in the active region is zero.

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

Transient model parameters for BJTs are composed of two groups: junction capacitor parameters and transit time parameters. The base-emitter junction is modeled by CJE, VJE, and MJE. The base-collector junction capacitance is modeled by CJC, VJC, and MJC. The collector-substrate junction capacitance is modeled by CJS, VJS, and MJS.
TF is the forward transit time for base charge storage. TF can be modified to account for bias, current, and phase, by XTF, VTF, ITF, and PTF. The base charge storage reverse transit time is set by TR. There are several sets of temperature equations for the BJT model parameters that you can select by setting TLEV and TLEVC.

**Table 16-1: BJT Model Parameters**

<table>
<thead>
<tr>
<th>DC</th>
<th>BF, BR, IBC, IBE, IS, ISS, NF, NR, NS, VAF, VAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>beta degradation</td>
<td>ISC, ISE, NC, NE, IKF, IKR</td>
</tr>
<tr>
<td>geometric</td>
<td>SUBS, BULK</td>
</tr>
<tr>
<td>resistor</td>
<td>RB, RBM, RE, RC, IRB</td>
</tr>
<tr>
<td>junction capacitor</td>
<td>CJC, CJE, CJS, FC, MJC, MJE, MJS, VJC, VJE, VJS, XCJC</td>
</tr>
<tr>
<td>parasitic capacitance</td>
<td>CBCP, CBEP, CCSP</td>
</tr>
<tr>
<td>transit time</td>
<td>ITF, PTF, TF, VT, VTF, XTF</td>
</tr>
<tr>
<td>noise</td>
<td>KF, AF</td>
</tr>
</tbody>
</table>

**Using BJT Basic DC Model Parameters**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BF (BFM)</td>
<td></td>
<td>100.0</td>
<td>Ideal maximum forward Beta</td>
</tr>
<tr>
<td>BR (BRM)</td>
<td></td>
<td>1.0</td>
<td>Ideal maximum reverse Beta</td>
</tr>
<tr>
<td>BULK (NSUB)</td>
<td></td>
<td>0.0</td>
<td>Sets the bulk node to a global node name. A substrate terminal node name (ns) in the element statement overrides BULK.</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| IBC         | amp  | 0.0     | Reverse saturation current between base and collector. If both IBE and IBC are specified, Star-Hspice uses them in place of IS to calculate DC current and conductance; otherwise it uses IS.

\[ I_{BC_{eff}} = I_{BC} \cdot A_{REAB} \cdot M \]

\[ A_{REAC} \text{ replaces } A_{REAB}, \text{ depending on vertical or lateral geometry.} \]

| EXPLI       | amp  | 1e15    | Current explosion model parameter. The PN junction characteristics above the explosion current area linear, with the slope at the explosion point. This speeds up simulation and improves convergence.

\[ I_{EXPLI_{eff}} = I_{EXPLI} \cdot A_{REAreff} \]

| IBE         | amp  | 0.0     | Reverse saturation current between base and emitter. If both IBE and IBC are specified, Star-Hspice uses them in place of IS to calculate DC current and conductance; otherwise it uses IS.

\[ I_{BE_{eff}} = I_{BE} \cdot A_{REA} \cdot M \]

| IS          | amp  | 1.0e-16 | Transport saturation current. If both IBE and IBC are specified, Star-Hspice uses them in place of IS to calculate DC current and conductance; otherwise it uses IS.

\[ I_{SEff} = I_{S} \cdot A_{REA} \cdot M \]

| ISS         | amp  | 0.0     | Reverse saturation current bulk-to-collector or bulk-to-base, depending on vertical or lateral geometry selection

\[ S_{SEff} = I_{SS} \cdot A_{REA} \cdot M \]
## Understanding the BJT Model Statement Using BJT Models

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>Model selector</td>
</tr>
<tr>
<td>NF</td>
<td></td>
<td>1.0</td>
<td>Forward current emission coefficient</td>
</tr>
<tr>
<td>NR</td>
<td></td>
<td>1.0</td>
<td>Reverse current emission coefficient</td>
</tr>
<tr>
<td>NS</td>
<td></td>
<td>1.0</td>
<td>Substrate current emission coefficient</td>
</tr>
<tr>
<td>SUBS</td>
<td></td>
<td></td>
<td>Substrate connection selector: +1 for vertical geometry, -1 for lateral geometry default=1 for NPN, default=-1 for PNP</td>
</tr>
<tr>
<td>UPDATE</td>
<td></td>
<td>0</td>
<td>UPDATE = 1 selects alternate base charge equation</td>
</tr>
</tbody>
</table>
Using Low-Current Beta Degradation Effect Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| ISC (C4, JLC) | amp | 0.0 | Base-collector leakage saturation current. If ISC is greater than 1e-4, then:  
\[ ISC = IS \cdot ISC \]
\[ IS_{eff} = ISC \cdot AREAB \cdot M \]
AREAC replaces AREAB, depending on vertical or lateral geometry. |
| ISE (C2, JLE) | amp | 0.0 | Base-emitter leakage saturation current. If ISE is greater than 1e-4, then:  
\[ ISE = IS \cdot ISE \]
\[ ISE_{eff} = ISE \cdot AREA \cdot M \] |
| NC (NLC) | 2.0 | Base-collector leakage emission coefficient |
| NE (NLE) | 1.5 | Base-emitter leakage emission coefficient |

Using Base Width Modulation Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAF (VA, VBF)</td>
<td>V</td>
<td>0.0</td>
<td>Forward early voltage. Use zero to indicate an infinite value.</td>
</tr>
<tr>
<td>VAR (VB, VRB, BV)</td>
<td>V</td>
<td>0.0</td>
<td>Reverse early voltage. Use zero to indicate an infinite value.</td>
</tr>
</tbody>
</table>
Using High-Current Beta Degradation Effect Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IKF (IK, JBF)</td>
<td>amp</td>
<td>0.0</td>
<td>Corner for forward Beta high-current roll-off. Use zero to indicate an infinite value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\text{IKFeff} = \text{IKF} \cdot \text{AREA} \cdot \text{M}$</td>
</tr>
<tr>
<td>IKR (JBR)</td>
<td>amp</td>
<td>0.0</td>
<td>Corner for reverse Beta high-current roll-off. Use zero to indicate an infinite value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\text{IKReff} = \text{IKR} \cdot \text{AREA} \cdot \text{M}$</td>
</tr>
<tr>
<td>NKF</td>
<td></td>
<td>0.5</td>
<td>Exponent for high-current Beta roll-off</td>
</tr>
</tbody>
</table>

Using Parasitic Resistance Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRB (JRB, IOB)</td>
<td>amp</td>
<td>0.0</td>
<td>Base current, where base resistance falls halfway to RBM. Use zero to indicate an infinite value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\text{IRBeff} = \text{IRB} \cdot \text{AREA} \cdot \text{M}$</td>
</tr>
<tr>
<td>RB</td>
<td>ohm</td>
<td>0.0</td>
<td>Base resistance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\text{RBeff} = \text{RB} / (\text{AREA} \cdot \text{M})$</td>
</tr>
<tr>
<td>RBM</td>
<td>ohm</td>
<td>RB</td>
<td>Minimum high-current base resistance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\text{RBMeff} = \text{RBM} / (\text{AREA} \cdot \text{M})$</td>
</tr>
<tr>
<td>RE</td>
<td>ohm</td>
<td>0.0</td>
<td>Emitter resistance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\text{REeff} = \text{RE} / (\text{AREA} \cdot \text{M})$</td>
</tr>
<tr>
<td>RC</td>
<td>ohm</td>
<td>0.0</td>
<td>Collector resistance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\text{RCEff} = \text{RC} / (\text{AREA} \cdot \text{M})$</td>
</tr>
</tbody>
</table>
## Using Junction Capacitor Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJC</td>
<td>F</td>
<td>0.0</td>
<td>Base-collector zero-bias depletion capacitance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Vertical:</strong> $C_{JC_{eff}} = C_{JC} \cdot A_{REAB} \cdot M$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Lateral:</strong> $C_{JC_{eff}} = C_{JC} \cdot A_{REAC} \cdot M$</td>
</tr>
<tr>
<td>CJE</td>
<td>F</td>
<td>0.0</td>
<td>Base-emitter zero-bias depletion capacitance (vertical and lateral):</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$C_{JE_{eff}} = C_{JE} \cdot A_{AREA} \cdot M$</td>
</tr>
<tr>
<td>CJS (CCS, CSUB)</td>
<td>F</td>
<td>0.0</td>
<td>Zero-bias collector substrate capacitance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Vertical:</strong> $C_{JS_{eff}} = C_{JS} \cdot A_{REAC} \cdot M$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Lateral:</strong> $C_{JS_{eff}} = C_{JS} \cdot A_{REAB} \cdot M$</td>
</tr>
<tr>
<td>FC</td>
<td></td>
<td>0.5</td>
<td>Coefficient for forward bias depletion capacitance formula for $DCAP=1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DCAP Default=2 and FC is ignored</td>
</tr>
<tr>
<td>MJC (MC)</td>
<td></td>
<td>0.33</td>
<td>Base-collector junction exponent (grading factor)</td>
</tr>
<tr>
<td>MJE (ME)</td>
<td></td>
<td>0.33</td>
<td>Base-emitter junction exponent (grading factor)</td>
</tr>
<tr>
<td>MJS (ESUB)</td>
<td></td>
<td>0.5</td>
<td>Substrate junction exponent (grading factor)</td>
</tr>
<tr>
<td>VJC (PC)</td>
<td>V</td>
<td>0.75</td>
<td>Base-collector built-in potential</td>
</tr>
<tr>
<td>VJE (PE)</td>
<td>V</td>
<td>0.75</td>
<td>Base-emitter built-in potential</td>
</tr>
<tr>
<td>VJS (PSUB)</td>
<td>V</td>
<td>0.75</td>
<td>Substrate junction built in potential</td>
</tr>
<tr>
<td>XCJC (CDIS)</td>
<td></td>
<td>1.0</td>
<td>Internal base fraction of base-collector depletion capacitance</td>
</tr>
</tbody>
</table>
Using Parasitic Capacitances Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBCP</td>
<td>F</td>
<td>0.0</td>
<td>External base-collector constant capacitance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CBCP_{eff} = CBCP \cdot \text{AREA} \cdot \text{M}</td>
</tr>
<tr>
<td>CBEP</td>
<td>F</td>
<td>0.0</td>
<td>External base-emitter constant capacitance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CBEP_{eff} = CBEP \cdot \text{AREA} \cdot \text{M}</td>
</tr>
<tr>
<td>CCSP</td>
<td>F</td>
<td>0.0</td>
<td>External collector substrate constant capacitance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(vertical) or base substrate (lateral)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CCSP_{eff} = CCSP \cdot \text{AREA} \cdot \text{M}</td>
</tr>
</tbody>
</table>

Using Transit Time Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITF (JTF)</td>
<td>amp</td>
<td>0.0</td>
<td>TF high-current parameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ITF_{eff} = ITF \cdot \text{AREA} \cdot \text{M}</td>
</tr>
<tr>
<td>PTF</td>
<td>x</td>
<td>0.0</td>
<td>Frequency multiplier to determine excess phase</td>
</tr>
<tr>
<td>TF</td>
<td>s</td>
<td>0.0</td>
<td>Base forward transit time</td>
</tr>
<tr>
<td>TR</td>
<td>s</td>
<td>0.0</td>
<td>Base reverse transit time</td>
</tr>
<tr>
<td>VTF</td>
<td>V</td>
<td>0.0</td>
<td>TF base-collector voltage dependence coefficient</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Zero indicates an infinite value.</td>
</tr>
<tr>
<td>XTF</td>
<td></td>
<td>0.0</td>
<td>TF bias dependence coefficient</td>
</tr>
</tbody>
</table>
### Using Noise Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>1.0</td>
<td></td>
<td>Flicker-noise exponent</td>
</tr>
<tr>
<td>KF</td>
<td>0.0</td>
<td></td>
<td>Flicker-noise coefficient</td>
</tr>
</tbody>
</table>

### Using BJT LEVEL=2 Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRS</td>
<td>1.0</td>
<td></td>
<td>Reverse beta for substrate BJT.</td>
</tr>
</tbody>
</table>
| GAMMA| 0.0   |         | Epitaxial doping factor, 

\[
\text{GAMMA} = (2 \cdot n_i / n)^2
\]

where \( n \) is epitaxial impurity concentration |
| NEPI | 1.0   |         | Emission coefficient                                                        |
| QCO  | Coul  | 0.0     | Epitaxial charge factor

\[
\text{Vertical: } QCO_{\text{eff}} = QCO \cdot \text{AREAB} \cdot M
\]

\[
\text{Lateral: } QCO_{\text{eff}} = QCO \cdot \text{AREAC} \cdot M
\]
| RC   | ohm   | 0.0     | Resistance of the epitaxial region under equilibrium conditions

\[
\text{RCEff} = RC / (\text{AREA} \cdot M)
\]
| VO   | V     | 0.0     | Carrier velocity saturation voltage. Use zero to indicate an infinite value. |
Handling BJT Model Temperature Effects

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

### Table 16-2: BJT Temperature Parameters

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>base modulation</td>
<td>TVAF1, TVAF2, TVAR1, TVAR2</td>
</tr>
<tr>
<td>capacitor</td>
<td>CTC, CTE, CTS</td>
</tr>
<tr>
<td>capacitor potentials</td>
<td>TVJC, TVJE, TVJS</td>
</tr>
<tr>
<td>DC</td>
<td>TBF1, TBF2, TBR1, TBR2, TIKF1, TIKF2, TIKR1, TIKR2, TIRB1, TIRB2, TISC1, TISC2, TIS1, TIS2, TISE1, TISE2, TISS1, TISS2, XTB, XTI</td>
</tr>
<tr>
<td>emission coefficients</td>
<td>TNC1, TNC2, TNE1, TNE2, TNF1, TNF2, TNR1, TNR2, TNS1, TNS2</td>
</tr>
<tr>
<td>energy gap</td>
<td>EG, GAP1, GAP2</td>
</tr>
<tr>
<td>equation selectors</td>
<td>TLEV, TLEVC</td>
</tr>
<tr>
<td>grading</td>
<td>MJC, MJE, MJS, TMJC1, TMJC2, TMJE1, TMJE2, TMJS1, TMJS2</td>
</tr>
<tr>
<td>resistors</td>
<td>TRB1, TRB2, TRC1, TRC2, TRE1, TRE2, TRM1, TRM2</td>
</tr>
<tr>
<td>transit time</td>
<td>TTF1, TTF2, TTR1, TTR2</td>
</tr>
</tbody>
</table>

Using Temperature Effect Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEX</td>
<td></td>
<td>2.42</td>
<td>VO temperature exponent (LEVEL 2 only)</td>
</tr>
<tr>
<td>BEXV</td>
<td></td>
<td>1.90</td>
<td>RC temperature exponent (LEVEL 2 only)</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>CTC</td>
<td>1/°C</td>
<td>0.0</td>
<td>Temperature coefficient for zero-bias base collector capacitance. TLEV=1 enables CTC to override the default Star-Hspice temperature compensation.</td>
</tr>
<tr>
<td>CTE</td>
<td>1/°C</td>
<td>0.0</td>
<td>Temperature coefficient for zero-bias base emitter capacitance. TLEV=1 enables CTE to override the default Star-Hspice temperature compensation.</td>
</tr>
<tr>
<td>CTS</td>
<td>1/°C</td>
<td>0.0</td>
<td>Temperature coefficient for zero-bias substrate capacitance. TLEV=1 enables CTS to override the default Star-Hspice temperature compensation.</td>
</tr>
<tr>
<td>EG</td>
<td>eV</td>
<td></td>
<td>Energy gap for pn junction for TLEV=0 or 1, default=1.11; for TLEV=2, default=1.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.17 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.69 - Schottky barrier diode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.67 - germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.52 - gallium arsenide</td>
</tr>
<tr>
<td>GAP1</td>
<td>eV/°C</td>
<td>7.02e-4</td>
<td>First bandgap correction factor (from Sze, alpha term)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7.02e-4 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.73e-4 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.56e-4 - germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.41e-4 - gallium arsenide</td>
</tr>
</tbody>
</table>
### Understanding the BJT Model Statement

#### Using BJT Models

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAP2</td>
<td>x</td>
<td>1108</td>
<td>Second bandgap correction factor (from Sze, beta term)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1108 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>636 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>210 - germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>204 - gallium arsenide</td>
</tr>
<tr>
<td>MJC (MC)</td>
<td></td>
<td>0.33</td>
<td>Base-collector junction exponent (grading factor)</td>
</tr>
<tr>
<td>MJE (ME)</td>
<td></td>
<td>0.33</td>
<td>Base-emitter junction exponent (grading factor)</td>
</tr>
<tr>
<td>MJS (ESUB)</td>
<td></td>
<td>0.5</td>
<td>Substrate junction exponent (grading factor)</td>
</tr>
<tr>
<td>TBF1</td>
<td>1/°</td>
<td>0.0</td>
<td>First-order temperature coefficient for BF</td>
</tr>
<tr>
<td>TBF2</td>
<td>1/°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for BF</td>
</tr>
<tr>
<td>TBR1</td>
<td>1/°</td>
<td>0.0</td>
<td>First-order temperature coefficient for BR</td>
</tr>
<tr>
<td>TBR2</td>
<td>1/°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for BR</td>
</tr>
<tr>
<td>TIKF1</td>
<td>1/°</td>
<td>0.0</td>
<td>First-order temperature coefficient for IKF</td>
</tr>
<tr>
<td>TIKF2</td>
<td>1/°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for IKF</td>
</tr>
<tr>
<td>TIKR1</td>
<td>1/°</td>
<td>0.0</td>
<td>First-order temperature coefficient for IKR</td>
</tr>
<tr>
<td>TIKR2</td>
<td>1/°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for IKR</td>
</tr>
<tr>
<td>TIRB1</td>
<td>1/°</td>
<td>0.0</td>
<td>First-order temperature coefficient for IRB</td>
</tr>
<tr>
<td>TIRB2</td>
<td>1/°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for IRB</td>
</tr>
<tr>
<td>TISC1</td>
<td>1/°</td>
<td>0.0</td>
<td>First-order temperature coefficient for ISC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TLEV=3 enables TISC1.</td>
</tr>
<tr>
<td>TISC2</td>
<td>1/°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for ISC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TLEV=3 enables TISC2.</td>
</tr>
</tbody>
</table>
Using BJT Models

Understanding the BJT Model Statement

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIS1</td>
<td>1/o</td>
<td>0.0</td>
<td>First-order temperature coefficient for IS or IBE and IBC TLEV=3 enables TIS1.</td>
</tr>
<tr>
<td>TIS2</td>
<td>1/o2</td>
<td>0.0</td>
<td>Second-order temperature coefficient for IS or IBE and IBC TLEV=3 enables TIS2.</td>
</tr>
<tr>
<td>TISE1</td>
<td>1/o</td>
<td>0.0</td>
<td>First-order temperature coefficient for ISE TLEV=3 enables TISE1.</td>
</tr>
<tr>
<td>TISE2</td>
<td>1/o2</td>
<td>0.0</td>
<td>Second-order temperature coefficient for ISE TLEV=3 enables TISE2.</td>
</tr>
<tr>
<td>TISS1</td>
<td>1/o</td>
<td>0.0</td>
<td>First-order temperature coefficient for ISS TLEV=3 enables TISS1.</td>
</tr>
<tr>
<td>TISS2</td>
<td>1/o2</td>
<td>0.0</td>
<td>Second-order temperature coefficient for ISS TLEV=3 enables TISS2.</td>
</tr>
<tr>
<td>TITF1</td>
<td></td>
<td></td>
<td>First-order temperature coefficient for ITF</td>
</tr>
<tr>
<td>TITF2</td>
<td></td>
<td></td>
<td>Second-order temperature coefficient for ITF</td>
</tr>
<tr>
<td>TLEV</td>
<td></td>
<td>1</td>
<td>Temperature equation level selector for BJTs (interacts with TLEVC)</td>
</tr>
<tr>
<td>TLEVC</td>
<td></td>
<td>1</td>
<td>Temperature equation level selector for BJTs, junction capacitances and potentials (interacts with TLEV)</td>
</tr>
<tr>
<td>TMJC1</td>
<td>1/o</td>
<td>0.0</td>
<td>First-order temperature coefficient for MJC</td>
</tr>
<tr>
<td>TMJC2</td>
<td>1/o2</td>
<td>0.0</td>
<td>Second-order temperature coefficient for MJC</td>
</tr>
<tr>
<td>TMJE1</td>
<td>1/o</td>
<td>0.0</td>
<td>First order temperature coefficient for MJE</td>
</tr>
<tr>
<td>TMJE2</td>
<td>1/o2</td>
<td>0.0</td>
<td>Second-order temperature coefficient for MJE</td>
</tr>
<tr>
<td>TMJS1</td>
<td>1/o</td>
<td>0.0</td>
<td>First-order temperature coefficient for MJS</td>
</tr>
<tr>
<td>TMJS2</td>
<td>1/o2</td>
<td>0.0</td>
<td>Second-order temperature coefficient for MJS</td>
</tr>
<tr>
<td>TNC1</td>
<td>1/o</td>
<td>0.0</td>
<td>First-order temperature coefficient for NC</td>
</tr>
<tr>
<td>TNC2</td>
<td></td>
<td>0.0</td>
<td>Second-order temperature coefficient for NC</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>---------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>TNE1</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for NE</td>
</tr>
<tr>
<td>TNE2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for NE</td>
</tr>
<tr>
<td>TNF1</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for NF</td>
</tr>
<tr>
<td>TNF2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for NF</td>
</tr>
<tr>
<td>TNR1</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for NR</td>
</tr>
<tr>
<td>TNR2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for NR</td>
</tr>
<tr>
<td>TNS1</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for NS</td>
</tr>
<tr>
<td>TNS2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for NS</td>
</tr>
<tr>
<td>TRB1 (TRB)</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for RB</td>
</tr>
<tr>
<td>TRB2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for RB</td>
</tr>
<tr>
<td>TRC1 (TRC)</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for RC</td>
</tr>
<tr>
<td>TRC2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for RC</td>
</tr>
<tr>
<td>TRE1 (TRE)</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for RE</td>
</tr>
<tr>
<td>TRE2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for RE</td>
</tr>
<tr>
<td>TRM1</td>
<td>1°</td>
<td></td>
<td>TRB1 First-order temperature coefficient for RBM</td>
</tr>
<tr>
<td>TRM2</td>
<td>1°²</td>
<td></td>
<td>TRB2 Second-order temperature coefficient for RBM</td>
</tr>
<tr>
<td>TTF1</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for TF</td>
</tr>
<tr>
<td>TTF2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for TF</td>
</tr>
<tr>
<td>TTR1</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for TR</td>
</tr>
<tr>
<td>TTR2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for TR</td>
</tr>
<tr>
<td>TVAF1</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for VAF</td>
</tr>
<tr>
<td>TVAF2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for VAF</td>
</tr>
<tr>
<td>TVAR1</td>
<td>1°</td>
<td>0.0</td>
<td>First-order temperature coefficient for VAR</td>
</tr>
<tr>
<td>TVAR2</td>
<td>1°²</td>
<td>0.0</td>
<td>Second-order temperature coefficient for VAR</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>TVJC</td>
<td>V/°</td>
<td>0.0</td>
<td>Temperature coefficient for VJC. TLEV=1 or 2 enables TVJC to override the default Star Hspice temperature compensation.</td>
</tr>
<tr>
<td>TVJE</td>
<td>V/°</td>
<td>0.0</td>
<td>Temperature coefficient for VJE. TLEV=1 or 2 enables TVJE to override the default Star Hspice temperature compensation.</td>
</tr>
<tr>
<td>TVJS</td>
<td>V/°</td>
<td>0.0</td>
<td>Temperature coefficient for VJS. TLEV=1 or 2 enables TVJS to override the default Star Hspice temperature compensation.</td>
</tr>
<tr>
<td>XTB (TB, TCB)</td>
<td></td>
<td>0.0</td>
<td>Forward and reverse Beta temperature exponent (used with TLEV=0, 1 or 2)</td>
</tr>
<tr>
<td>XTI</td>
<td></td>
<td>3.0</td>
<td>Saturation current temperature exponent. Use XTI = 3.0 for silicon diffused junction. Set XTI = 2.0 for Schottky barrier diode.</td>
</tr>
</tbody>
</table>
Using BJT Device Equivalent Circuits

Scaling

Scaling is controlled by the element parameters AREA, AREAB, AREAC, and M. The AREA parameter, the normalized emitter area, divides all resistors and multiplies all currents and capacitors. AREAB and AREAC scale the size of the base area and collector area. Either AREAB or AREAC is used for scaling, depending on whether vertical or lateral geometry is selected (using the SUBS model parameter). For vertical geometry, AREAB is the scaling factor for IBC, ISC, and CJC. For lateral geometry, AREAC is the scaling factor. The scaling factor is AREA for all other parameters.

The scaling of the DC model parameters (IBE, IS, ISE, IKF, IKR, and IRB) for both vertical and lateral BJT transistors, is determined by the following formula:

\[ I_{\text{eff}} = \text{AREA} \cdot M \cdot I \]

where I is either IBE, IS, ISE, IKF, IKR, or IRB.

For both the vertical and lateral, the resistor model parameters, RB, RBM, RE, and RC are scaled by the following equation.

\[ R_{\text{eff}} = \frac{R}{\text{AREA} \cdot M} \]

where R is either RB, RBM, RE, or RC.

Understanding the BJT Current Convention

The direction of current flow through the BJT is assumed in the example Figure 16-1. Use either I(Q1) or I1(Q1) syntax to print the collector current. I2(Q1) refers to the base current, I3(Q1) refers to the emitter current, and I4(Q1) refers to the substrate current.
Using BJT Equivalent Circuits

Star-Hspice uses four equivalent circuits in the analysis of BJTs: DC, transient, AC, and AC noise circuits. The components of these circuits form the basis for all element and model equations. Since these circuits represent the entire BJT in Star-Hspice, every effort has been made to demonstrate the relationship between the equivalent circuit and the element/model parameters.

The fundamental components in the equivalent circuit are the base current (ib) and the collector current (ic). For noise and AC analyses, the actual ib and ic currents are not used. Instead, the partial derivatives of ib and ic with respect to the terminal voltages vbe and vbc are used. The names for these partial derivatives are:

**Reverse Base Conductance**

\[
g_{\mu} = \left. \frac{\partial ib}{\partial vbc} \right|_{vbe = \text{const.}}
\]

**Forward Base Conductance**

\[
g_{\pi} = \left. \frac{\partial ib}{\partial vbe} \right|_{vbc = \text{const.}}
\]
Collector Conductance

\[ g_o = \frac{\partial i_c}{\partial v_{ce}} \bigg|_{v_{be} = \text{const.}} = \frac{\partial i_c}{\partial v_{bc}} \bigg|_{v_{be} = \text{const.}} \]

Transconductance

\[ g_m = \frac{\partial i_c}{\partial v_{be}} \bigg|_{v_{ce} = \text{con.}} = \frac{\partial i_c}{\partial v_{be}} + \frac{\partial i_c}{\partial v_{bc}} = \frac{\partial i_c}{\partial v_{be}} - g_o \]

The ib and ic equations account for all DC effects of the BJT.

**Figure 16-2: Lateral Transistor, BJT Transient Analysis**
Figure 16-3: Vertical Transistor, BJT Transient Analysis

Figure 16-4: Lateral Transistor, BJT AC Analysis
Figure 16-5: Vertical Transistor, BJT AC Analysis

Figure 16-6: Lateral Transistor, BJT AC Noise Analysis
Figure 16-7: Vertical Transistor, BJT AC Noise Analysis

Table 16-3: Equation Variable Names

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>cbc</td>
<td>Internal base to collector capacitance</td>
</tr>
<tr>
<td>cbcx</td>
<td>External base to collector capacitance</td>
</tr>
<tr>
<td>cbe</td>
<td>Internal base to emitter capacitance</td>
</tr>
<tr>
<td>csc</td>
<td>Substrate to collector capacitance (vertical transistor only)</td>
</tr>
<tr>
<td>cbs</td>
<td>Base to substrate capacitance (lateral transistor only)</td>
</tr>
<tr>
<td>f</td>
<td>Frequency</td>
</tr>
<tr>
<td>gbc</td>
<td>Reverse base conductance</td>
</tr>
<tr>
<td>gbe</td>
<td>Forward base conductance</td>
</tr>
<tr>
<td>gm</td>
<td>Transconductance</td>
</tr>
</tbody>
</table>
Table 16-3: Equation Variable Names

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>gsc</td>
<td>Substrate to collector conductance (vertical transistor only)</td>
</tr>
<tr>
<td>go</td>
<td>Collector conductance</td>
</tr>
<tr>
<td>gbs</td>
<td>Base to substrate conductance (lateral transistor only)</td>
</tr>
<tr>
<td>ib</td>
<td>External base terminal current</td>
</tr>
<tr>
<td>ibc</td>
<td>DC current base to collector</td>
</tr>
<tr>
<td>ibe</td>
<td>DC current base to emitter</td>
</tr>
<tr>
<td>ic</td>
<td>External collector terminal current</td>
</tr>
<tr>
<td>ice</td>
<td>DC current collector to emitter</td>
</tr>
<tr>
<td>inb</td>
<td>Base current equivalent noise</td>
</tr>
<tr>
<td>inc</td>
<td>Collector current equivalent noise</td>
</tr>
<tr>
<td>inrb</td>
<td>Base resistor current equivalent noise</td>
</tr>
<tr>
<td>inrc</td>
<td>Collector resistor equivalent noise</td>
</tr>
<tr>
<td>inre</td>
<td>Emitter resistor current equivalent noise</td>
</tr>
<tr>
<td>ibs</td>
<td>DC current base to substrate (lateral transistor only)</td>
</tr>
<tr>
<td>isc</td>
<td>DC current substrate to collector (vertical transistor only)</td>
</tr>
<tr>
<td>qb</td>
<td>Normalized base charge</td>
</tr>
<tr>
<td>rb</td>
<td>Base resistance</td>
</tr>
<tr>
<td>rbb</td>
<td>Short-circuit base resistance</td>
</tr>
<tr>
<td>vbs</td>
<td>Internal base substrate voltage</td>
</tr>
<tr>
<td>vsc</td>
<td>Internal substrate collector voltage</td>
</tr>
</tbody>
</table>
Table 16-4: Equation Constants

<table>
<thead>
<tr>
<th>Quantities</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>$1.38062e-23$ (Boltzmann’s constant)</td>
</tr>
<tr>
<td>q</td>
<td>$1.60212e-19$ (electron charge)</td>
</tr>
<tr>
<td>t</td>
<td>Temperature in °Kelvin</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>$t - t_{\text{nom}}$</td>
</tr>
<tr>
<td>$t_{\text{nom}}$</td>
<td>$t_{\text{nom}} = 273.15 + T_{\text{NOM}}$ in °Kelvin</td>
</tr>
<tr>
<td>$v_t(t)$</td>
<td>$k \cdot t/q$</td>
</tr>
<tr>
<td>$v_t(t_{\text{nom}})$</td>
<td>$k \cdot t_{\text{nom}}/q$</td>
</tr>
</tbody>
</table>
Using BJT Model Equations (NPN and PNP)

This section describes the NPN and PNP BJT models.

Understanding Transistor Geometry in Substrate Diodes

The substrate diode is connected to either the collector or the base depending on whether the transistor has a lateral or vertical geometry. Lateral geometry is implied when the model parameter SUBS=-1, and vertical geometry when SUBS=+1. The lateral transistor substrate diode is connected to the internal base and the vertical transistor substrate diode is connected to the internal collector. Figure 16-8 and Figure 16-9 show vertical and lateral transistor geometries.
In Figure 16-10, the views from the top demonstrate how IBE is multiplied by either base area, AREAB, or collector area, AREAC.

**Figure 16-10: Base, AREAB, Collector, AREAC**

Using DC Model Equations

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

**Current Equations - IS Only**

If only IS is specified, without IBE and IBC:

\[
\begin{align*}
\text{ic} &= \frac{ISeff}{q_b} \cdot \left( \frac{v_{be}}{e^{NF \cdot v_t} - e^{NR \cdot v_t}} \right) - \frac{ISeff}{BR} \cdot \left( e^{NR \cdot v_t - 1} \right) - ISCeff \cdot \left( e^{NC \cdot v_t - 1} \right) \\
\text{ib} &= \frac{ISeff}{BF} \cdot \left( e^{NF \cdot v_t - 1} \right) + \frac{ISeff}{BR} \cdot \left( e^{NR \cdot v_t - 1} \right) + ISCeff \cdot \left( e^{NC \cdot v_t - 1} \right)
\end{align*}
\]
Current Equations - IBE and IBC

If IBE and IBC are specified, instead of IS:

\[
\begin{align*}
  i_c &= \frac{IBEff}{q_b} \cdot \left( e^{v_{be}/N_F \cdot v_t} - 1 \right) - \frac{IBCEff}{q_b} \cdot \left( e^{v_{bc}/N_R \cdot v_t} - 1 \right) - \frac{IBCEff}{BR} \cdot \left( e^{v_{bc}/N_R \cdot v_t} - 1 \right) \\
  &\quad - ISCeff \cdot \left( e^{v_{bc}/N_C \cdot v_t} - 1 \right) \\
  ib &= \frac{IBEff}{BF} \cdot \left( e^{v_{be}/N_F \cdot v_t} - 1 \right) + \frac{IBCEff}{BF} \cdot \left( e^{v_{bc}/N_R \cdot v_t} - 1 \right) + ISEff \cdot \left( e^{v_{bc}/N_E \cdot v_t} - 1 \right) \\
  &\quad + ISCeff \cdot \left( e^{v_{bc}/N_C \cdot v_t} - 1 \right)
\end{align*}
\]

\[
\begin{align*}
  IBCeff &= IBC \cdot AREAB \cdot M \quad \text{Vertical} \\
  IBCeff &= IBC \cdot AREAC \cdot M \quad \text{Lateral} \\
  IBEeff &= IBE \cdot AREA \cdot M \quad \text{Vertical or Lateral} \\
  ISCEff &= ISC \cdot AREAB \cdot M \quad \text{Vertical} \\
  ISCEff &= ISC \cdot AREAC \cdot M \quad \text{Lateral} \\
  ISEEff &= ISE \cdot AREA \cdot M \quad \text{Vertical or Lateral}
\end{align*}
\]

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

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

**Vertical Transistors**

\[
isc = ISS_{eff} \cdot \left( e^{\frac{vsc}{NS \cdot vt}} - 1 \right) \quad vsc > -10 \cdot NS \cdot vt
\]

\[
isc = -ISS_{eff} \quad vsc \leq -10 \cdot NS \cdot vt
\]

**Lateral Transistors**

\[
ibs = ISS_{eff} \cdot \left( e^{\frac{vbs}{NS \cdot vt}} - 1 \right) \quad vbs > -10 \cdot NS \cdot vt
\]

\[
ibs = -ISS_{eff} \quad vbs \leq -10 \cdot NS \cdot vt
\]

If both IBE and IBC are not specified:

\[
ISS_{eff} = ISS \cdot AREA \cdot M
\]

If both IBE and IBC are specified:

\[
ISS_{eff} = ISS \cdot AREAC \cdot M \quad \text{vertical}
\]

\[
ISS_{eff} = ISS \cdot AREAB \cdot M \quad \text{lateral}
\]
Using Base Charge Equations

VAF and VAR are, respectively, forward and reverse early voltages. IKF and IKR determine the high-current Beta roll-off. ISE, ISC, NE, and NC determine the low-current Beta roll-off with ic.

If UPDATE = 0 or \( \frac{v_{bc}}{VAF} + \frac{v_{be}}{VAR} < 0 \), then

\[
q_1 = \frac{1}{1 - \frac{v_{bc}}{VAF} - \frac{v_{be}}{VAR}}
\]

Otherwise, if UPDATE = 1 and \( \frac{v_{bc}}{VAF} + \frac{v_{be}}{VAR} \geq 0 \), then

\[
q_1 = 1 + \frac{v_{bc}}{VAF} + \frac{v_{be}}{VAR}
\]

\[
q_2 = \frac{I_{SEeff}}{I_{KFeff}} \cdot \left( e^{\frac{v_{be}}{V_{NF} \cdot v_t}} - 1 \right) + \frac{I_{SCeff}}{I_{KReff}} \cdot \left( e^{\frac{v_{bc}}{V_{NR} \cdot v_t}} - 1 \right)
\]

\[
q_b = \frac{q_1}{2} \cdot \left[ 1 + (1 + 4 \cdot q_2)^{NK_F} \right]
\]

Using Variable Base Resistance Equations

Star-Hspice provides a variable base resistance model consisting of a low-current maximum resistance set by RB and a high-current minimum resistance set by RBM. IRB is the current when the base resistance is halfway to its minimum value. If RBM is not specified, it is set to RB.

If IRB is not specified:

\[
r_{bb} = R_{Meff} + \frac{R_{Beff} - R_{Meff}}{q_b}
\]
If IRB is specified:

\[
rbb = RB_{\text{Eff}} + 3 \cdot (R_{\text{Beff}} - RB_{\text{Eff}}) \cdot \frac{\tan(z) - z}{z \cdot \tan(z) \cdot \tan(z)}
\]

\[
z = \frac{-1 + \left[1 + 144 \cdot \frac{ib}{I(R_{\text{Beff}})}\right]^{1/2}}{24 \cdot \frac{ib}{\pi^2} \cdot \left(\frac{ib}{R_{\text{Beff}}}\right)^{1/2}}
\]
Using BJT Capacitance Equations

This section describes BJT capacitances.

Using Base-Emitter Capacitance Equations

The base-emitter capacitance contains a complex diffusion term with the standard depletion capacitance formula. The diffusion capacitance is modified by model parameters TF, XTF, ITF, and VTF.

Determine the base-emitter capacitance \( c_{be} \) by the following formula:

\[
  c_{be} = c_{bediff} + c_{bedep}
\]

where \( c_{bediff} \) and \( c_{bedep} \) are the base-emitter diffusion and depletion capacitances, respectively.

**Note:** When you run a DC sweep on a BJT, use .OPTIONS DCCAP to force the evaluation of the voltage-variable capacitances during the DC sweep.

Determining Base-Emitter Diffusion Capacitance

Determine diffusion capacitance as follows:

\( \text{ibe} \leq 0 \)

\[
  c_{bediff} = \frac{\partial}{\partial v_{be}} \left( TF \cdot \frac{\text{ibe}}{q b} \right)
\]

\( \text{ibe} > 0 \)

\[
  c_{bediff} = \frac{\partial}{\partial v_{be}} \left[ TF \cdot (1 + \text{argtf}) \cdot \frac{\text{ibe}}{q b} \right]
\]
where:

\[
\arg_{tf} = XTF \cdot \left( \frac{ibe}{ibe + ITF} \right)^2 \cdot e^{\frac{vbc}{1.44 \cdot VTF}}
\]

The forward part of the collector-emitter branch current is determined as follows:

\[
ibe = IS_{eff} \cdot \left( e^{\frac{vbe}{NF \cdot \nu_l} - 1} \right)
\]

**Determining Base-Emitter Depletion Capacitance**

There are two different equations for modeling the depletion capacitance. Select the proper equation by specifying option DCAP in an OPTIONS statement.

**DCAP=1**

The base-emitter depletion capacitance is determined as follows:

\[
vbe < FC \cdot VJE
\]

\[
cbedep = CJE_{eff} \cdot \left( 1 - \frac{vbe}{VJE} \right)^{-MJE}
\]

\[
vbe \geq FC \cdot VJE
\]

\[
cbedep = CJE_{eff} \cdot \frac{1 - FC \cdot (1 + MJE) + MJE \cdot \frac{vbe}{VJE}}{(1 - FC)(1 + MJE)}
\]
DCAP=2
The base-emitter depletion capacitance is determined as follows:

\( v_{be} < 0 \)
\[ c_{bedep} = C_{JEeff} \cdot \left(1 - \frac{v_{be}}{V_{JE}}\right)^{-M_{JE}} \]

\( v_{be} \geq 0 \)
\[ c_{bedep} = C_{JEeff} \cdot \left(1 + M_{JE} \cdot \frac{v_{be}}{V_{JE}}\right) \]

DCAP=3
Limits peak depletion capacitance to \( F_C \cdot C_{JCEff} \) or \( F_C \cdot C_{JJEff} \), with proper fall-off when forward bias exceeds \( P_B \) (\( F_C \geq 1 \)).

Determining Base Collector Capacitance
Determine the base collector capacitance \( c_{bc} \) as follows:
\[ c_{bc} = c_{bcdiff} + c_{bcdep} \]

where \( c_{bcdiff} \) and \( c_{bcdep} \) are the base-collector diffusion and depletion capacitances, respectively.

Determining Base Collector Diffusion Capacitance
\[ c_{bcdiff} = \frac{\partial}{\partial v_{bc}}(T_R \cdot i_{bc}) \]

where the internal base-collector current \( i_{bc} \) is:
\[ i_{bc} = I_{Seff} \cdot \left(\frac{v_{bc}}{e^{N_R \cdot v_t}} - 1\right) \]
Determining Base Collector Depletion Capacitance

There are two different equations for modeling the depletion capacitance. Select the proper equation by specifying option DCAP in an .OPTIONS statement.

**DCAP=1**

Specify DCAP=1 to select one of the following equations:

\[ \text{vbc} < \text{FC} \cdot \text{VJC} \]

\[ cbcdep = XCJC \cdot CJCeff \cdot \left(1 - \frac{\text{vbc}}{\text{VJC}}\right)^{-MJC} \]

\[ \text{vbc} \geq \text{FC} \cdot \text{VJC} \]

\[ cbcdep = XCJC \cdot CJCeff \cdot \frac{1 - \text{FC} \cdot (1 + MJC) + MJC \cdot \frac{\text{vbc}}{\text{VJC}}}{(1 - \text{FC})^{(1 + MJC)}} \]

**DCAP=2**

Specify DCAP=2 to select one of the following equations:

\[ \text{vbc} < 0 \]

\[ cbcdep = XCJC \cdot CJCeff \cdot \left(1 - \frac{\text{vbc}}{\text{VJC}}\right)^{-MJC} \]

\[ \text{vbc} \geq 0 \]

\[ cbcdep = XCJC \cdot CJCeff \cdot \left(1 + MJC \cdot \frac{\text{vbc}}{\text{VJC}}\right) \]

External Base — Internal Collector Junction Capacitance

The base-collector capacitance is modeled as a distributed capacitance when the model parameter XCJC is set. Since the default setting of XCJC is one, the entire base-collector capacitance is on the internal base node cbc.
Using BJT Capacitance Equations

**DCAP=1**

Specify DCAP=1 to select one of the following equations:

\[ \text{vbcx} < \text{FC} \cdot \text{VJC} \]

\[ cbcx = CJC_{\text{eff}} \cdot (1 - XCJC) \cdot \left(1 - \frac{vbcx}{VJC}\right)^{-MJC} \]

\[ \text{vbcx} \geq \text{FC} \cdot \text{VJC} \]

\[ cbcx = CJC_{\text{eff}} \cdot (1 - XCJC) \cdot \frac{1 - FC \cdot (1 + MJC) + MJC \cdot \frac{vbcx}{VJC}}{(1 - FC)^{(1 + MJC)}} \]

**DCAP=2**

Specify DCAP=2 to select one of the following equations:

\[ \text{vbcx} < 0 \]

\[ cbcx = CJC_{\text{eff}} \cdot (1 - XCJC) \cdot \left(1 - \frac{vbcx}{VJC}\right)^{-MJC} \]

\[ \text{vbcx} \geq 0 \]

\[ cbcx = CJC_{\text{eff}} \cdot (1 - XCJC) \cdot \left(1 + MJC \cdot \frac{vbcx}{VJC}\right) \]

where \( vbcx \) is the voltage between the external base node and the internal collector node.

**Using Substrate Capacitance**

The function of substrate capacitance is similar to that of the substrate diode. Switch it from the collector to the base by setting the model parameter, SUBS.
Using Substrate Capacitance Equation — Lateral

Base to Substrate Diode
Reverse Bias \( v_{bs} < 0 \)

\[
c_{bs} = C_{J, eff} \cdot \left( 1 - \frac{v_{bs}}{V_{JS}} \right)^{-M_{JS}}
\]

Forward Bias \( v_{bs} \geq 0 \)

\[
c_{bs} = C_{J, eff} \cdot \left( 1 + M_{JS} \cdot \frac{v_{bs}}{V_{JS}} \right)
\]

Using Substrate Capacitance Equation — Vertical

Substrate to Collector Diode
Reverse Bias \( v_{sc} < 0 \)

\[
c_{sc} = C_{J, eff} \cdot \left( 1 - \frac{v_{sc}}{V_{JS}} \right)^{-M_{JS}}
\]

Forward Bias \( v_{sc} \geq 0 \)

\[
c_{sc} = C_{J, eff} \cdot \left( 1 + M_{JS} \cdot \frac{v_{sc}}{V_{JS}} \right)
\]

Using Excess Phase Equation

The model parameter, \( PTF \), models excess phase. It is defined as extra degrees of phase delay (introduced by the BJT) at any frequency and is determined by the equation:

\[
\text{excess phase} = \left( 2 \cdot \pi \cdot PTF \cdot \frac{TF}{360} \right) \cdot (2 \cdot \pi \cdot f)
\]

where \( f \) is in Hertz, and you can set \( PTF \) and \( TF \). The excess phase is a delay (linear phase) in the transconductance generator for AC analysis. Use it also in transient analysis.
Defining BJT Noise Equations

Equations for modeling BJT thermal, shot, and flicker noise are as follows.

**Defining Noise Equations**

The mean square short-circuit base resistance noise current equation is:

\[
inrb = \left( \frac{4 \cdot k \cdot t}{rbb} \right)^{1/2}
\]

The mean square short-circuit collector resistance noise current equation is:

\[
inrc = \left( \frac{4 \cdot k \cdot t}{RC_{\text{eff}}} \right)^{1/2}
\]

The mean square short-circuit emitter resistance noise current equation is:

\[
inre = \left( \frac{4 \cdot k \cdot t}{RE_{\text{eff}}} \right)^{1/2}
\]

The noise associated with the base current is composed of two parts: shot noise and flicker noise. Typical values for the flicker noise coefficient, KF, are $1e^{-17}$ to $1e^{-12}$. They are calculated as:

\[
2 \cdot q \cdot f_{\text{knee}}
\]

where \(f_{\text{knee}}\) is noise knee frequency (typically 100 Hz to 10 MHz) and \(q\) is electron charge.

\[
inb^2 = (2 \cdot q \cdot ib) + \left( \frac{KF \cdot ib^{AF}}{f} \right)
\]

\[
inb^2 = \text{shot noise}^2 + \text{flicker noise}^2
\]

\[
\text{shot noise} = (2 \cdot q \cdot ib)^{1/2}
\]

\[
\text{flicker noise} = \left( \frac{KF \cdot ib^{AF}}{f} \right)^{1/2}
\]
The noise associated with the collector current is modeled as shot noise only.

\[ inc = (2 \cdot q \cdot ic)^{1/2} \]

**Noise Summary Printout Definitions**

- \( RB, V^2/Hz \) output thermal noise due to base resistor
- \( RC, V^2/Hz \) output thermal noise due to collector resistor
- \( RE, V^2/Hz \) output thermal noise due to emitter resistor
- \( IB, V^2/Hz \) output shot noise due to base current
- \( FN, V^2/Hz \) output flicker noise due to base current
- \( IC, V^2/Hz \) output shot noise due to collector current
- \( TOT, V^2/Hz \) total output noise: \( TOT = RB + RC + RE + IB + IC + FN \)
Using BJT Temperature Compensation Equations

This section describes how to use temperature compensation equations.

Using Energy Gap Temperature Equations

To determine energy gap for temperature compensation, use the equations:

\[ T_{LEV} = 0, 1 \text{ or } 3 \]

\[ e_{gnom} = 1.16 - 7.02 \times 10^{-4} \cdot \frac{t_{nom}^2}{t_{nom} + 1108.0} \]

\[ e_g(t) = 1.16 - 7.02 \times 10^{-4} \cdot \frac{t^2}{t + 1108.0} \]

\[ T_{LEV} = 2 \]

\[ e_{gnom} = E_G - GAP1 \cdot \frac{t_{nom}^2}{t_{nom} + GAP2} \]

\[ e_g(t) = E_G - GAP1 \cdot \frac{t^2}{t + GAP2} \]

Using Saturation and Beta Temperature Equations, TLEV=0 or 2

The basic BJT temperature compensation equations for beta and the saturation currents when TLEV=0 or 2 (default is SPICE style TLEV=0):

\[ BF(t) = BF \cdot \left( \frac{t}{t_{nom}} \right)^{XTB} \]

\[ BR(t) = BR \cdot \left( \frac{t}{t_{nom}} \right)^{XTB} \]
Using BJT Models

Using BJT Temperature Compensation Equations

\[ ISE(t) = \frac{ISE}{
\left(\frac{t}{t_{\text{nom}}}\right)^{XTB} \cdot e^{\frac{\text{facln}}{NE}}}
\]

\[ ISC(t) = \frac{ISC}{
\left(\frac{t}{t_{\text{nom}}}\right)^{XTB} \cdot e^{\frac{\text{facln}}{NC}}}
\]

\[ ISS(t) = \frac{ISS}{
\left(\frac{t}{t_{\text{nom}}}\right)^{XTB} \cdot e^{\frac{\text{facln}}{NS}}}
\]

The parameter XTB usually should be set to zero for TLEV=2.

\[ IS(t) = IS \cdot e^{\frac{\text{facln}}{}} \]

\[ IBE(t) = IBE \cdot e^{\frac{\text{facln}}{NF}} \]

\[ IBC(t) = IBC \cdot e^{\frac{\text{facln}}{NR}} \]

**TLEV=0, 1 or 3**

\[ \text{facln} = \frac{EG}{vt(t_{\text{nom}})} - \frac{EG}{vt(t)} + XTI \cdot \ln\left(\frac{t}{t_{\text{nom}}}\right) \]

**TLEV=2**

\[ \text{facln} = \frac{eg_{\text{nom}}}{vt(t_{\text{nom}})} - \frac{eg(t)}{vt(t)} + XTI \cdot \ln\left(\frac{t}{t_{\text{nom}}}\right) \]
Using Saturation and Temperature Equations, TLEV=1

The basic BJT temperature compensation equations for beta and the saturation currents when TLEV=1:

\[
BF(t) = BF \cdot (1 + XTB \cdot \Delta t)
\]

\[
BR(t) = BR \cdot (1 + XTB \cdot \Delta t)
\]

\[
ISE(t) = \frac{ISE}{1 + XTB \cdot \Delta t} \cdot e^{\frac{facln}{NE}}
\]

\[
ISC(t) = \frac{ISC}{1 + XTB \cdot \Delta t} \cdot e^{\frac{facln}{NC}}
\]

\[
ISS(t) = \frac{ISS}{1 + XTB \cdot \Delta t} \cdot e^{\frac{facln}{NS}}
\]

\[
IS(t) = IS \cdot e^{\frac{facln}{}}
\]

\[
IBE(t) = IBE \cdot e^{\frac{facln}{NF}}
\]

\[
IBC(t) = IBC \cdot e^{\frac{facln}{NR}}
\]

where:

\[
facln = \frac{EG}{vt(t_{nom})} - \frac{EG}{vt(t)} + XTI \cdot \ln\left(\frac{t}{t_{nom}}\right)
\]

TLEV=0, 1, 2

The parameters IKF, IKR, and IRB are also modified as:

\[
IKF(t) = IKF \cdot (1 + TIKF1 \cdot \Delta t + TIKF2 \cdot \Delta t^2)
\]

\[
IKR(t) = IKR \cdot (1 + TIKR1 \cdot \Delta t + TIKR2 \cdot \Delta t^2)
\]

\[
IRB(t) = IRB \cdot (1 + TIRB1 \cdot \Delta t + TIRB2 \cdot \Delta t^2)
\]
Using Saturation Temperature Equations, TLEV=3

The basic BJT temperature compensation equations for the saturation currents when TLEV=3

\[ IS(t) = IS(1 + TIS1 \cdot \Delta t + TIS2 \cdot \Delta t^2) \]
\[ IBE(t) = IBE(1 + TIS1 \cdot \Delta t + TIS2 \cdot \Delta t^2) \]
\[ IBC(t) = IBC(1 + TIS1 \cdot \Delta t + TIS2 \cdot \Delta t^2) \]
\[ ISE(t) = ISE(1 + TISE1 \cdot \Delta t + TISE2 \cdot \Delta t^2) \]
\[ ISC(t) = ISC(1 + TISC1 \cdot \Delta t + TISC2 \cdot \Delta t^2) \]
\[ ISS(t) = ISS(1 + TISS1 \cdot \Delta t + TISS2 \cdot \Delta t^2) \]

The parameters IKF, IKR, and IRB are also modified as:

\[ IKF(t) = IKF(1 + TIKF1 \cdot \Delta t + TIKF2 \cdot \Delta t^2) \]
\[ IKR(t) = IKR(1 + TIKR1 \cdot \Delta t + TIKR2 \cdot \Delta t^2) \]
\[ IRB(t) = IRB(1 + TIRB1 \cdot \Delta t + TIRB2 \cdot \Delta t^2) \]

The following parameters are also modified when corresponding temperature coefficients are specified, regardless of the TLEV value.

\[ BF(t) = BF \cdot (1 + TBF1 \cdot \Delta t + TBF2 \cdot \Delta t^2) \]
\[ BR(t) = BR \cdot (1 + TBR1 \cdot \Delta t + TBR2 \cdot \Delta t^2) \]
\[ VAF(t) = VAF \cdot (1 + TVAF1 \cdot \Delta t + TVAF2 \cdot \Delta t^2) \]
\[ VAR(t) = VAR \cdot (1 + TVAR1 \cdot \Delta t + TVAR2 \cdot \Delta t^2) \]
\[ ITF(t) = ITF \cdot (1 + TITF1 \cdot \Delta t + TITF2 \cdot \Delta t^2) \]
\[ TF(t) = TF \cdot (1 + TTF1 \cdot \Delta t + TTF2 \cdot \Delta t^2) \]
\[ TR(t) = TR \cdot (1 + TTR1 \cdot \Delta t + TTR2 \cdot \Delta t^2) \]
\[ NF(t) = NF \cdot (1 + TNF1 \cdot \Delta t + TNF2 \cdot \Delta t^2) \]
\[ NR(t) = NR \cdot (1 + TNR1 \cdot \Delta t + TNR2 \cdot \Delta t^2) \]
\[ NE(t) = NE \cdot (1 + TNE1 \cdot \Delta t + TNE2 \cdot \Delta t^2) \]
\[ NC(t) = NC \cdot (1 + TNC1 \cdot \Delta t + TNC2 \cdot \Delta t^2) \]
\[ NS(t) = NS \cdot (1 + TNS1 \cdot \Delta t + TNS2 \cdot \Delta t^2) \]
\[ MJE(t) = MJE \cdot (1 + TMJE1 \cdot \Delta t + TMJE2 \cdot \Delta t^2) \]
\[ MJC(t) = MJC \cdot (1 + TMJC1 \cdot \Delta t + TMJC2 \cdot \Delta t^2) \]
\[ MJS(t) = MJS \cdot (1 + TMJS1 \cdot \Delta t + TMJS2 \cdot \Delta t^2) \]

**Using Capacitance Temperature Equations**

\[ TLEVC=0 \]

\[ CJE(t) = CJE \cdot \left[ 1 + MJE \cdot \left( 4.0e^{-4} \cdot \Delta t - \frac{VJE(t)}{VJE} + 1 \right) \right] \]
\[ CJC(t) = CJC \cdot \left[ 1 + MJC \cdot \left( 4.0e^{-4} \cdot \Delta t - \frac{VJC(t)}{VJC} + 1 \right) \right] \]
\[ CJS(t) = CJS \cdot \left[ 1 + MJS \cdot \left( 4.0e^{-4} \cdot \Delta t - \frac{VJS(t)}{VJS} + 1 \right) \right] \]

where:

\[ VJE(t) = VJE \cdot \frac{t}{tnom} - vt(t) \cdot \left[ 3 \cdot \ln \left( \frac{t}{tnom} \right) + \frac{egnom}{vt(tnom)} - \frac{eg(t)}{vt(t)} \right] \]
\[ VJC(t) = VJC \cdot \frac{t}{tnom} - vt(t) \cdot \left[ 3 \cdot \ln \left( \frac{t}{tnom} \right) + \frac{egnom}{vt(tnom)} - \frac{eg(t)}{vt(t)} \right] \]
\[ VJS(t) = VJS \cdot \frac{t}{tnom} - vt(t) \cdot \left[ 3 \cdot \ln \left( \frac{t}{tnom} \right) + \frac{egnom}{vt(tnom)} - \frac{eg(t)}{vt(t)} \right] \]
TLEVC=1

\[ CJE(t) = CJE \cdot (1 + CTE \cdot \Delta t) \]
\[ CJC(t) = CJC \cdot (1 + CTC \cdot \Delta t) \]
\[ CJS(t) = CJS \cdot (1 + CTS \cdot \Delta t) \]

and contact potentials determined as:
\[ VJE(t) = VJE - TVJE \cdot \Delta t \]
\[ VJC(t) = VJC - TVJC \cdot \Delta t \]
\[ VJS(t) = VJS - TVJS \cdot \Delta t \]

TLEVC=2

\[ CJE(t) = CJE \cdot \left( \frac{VJE}{VJE(t)} \right)^{MJE} \]
\[ CJC(t) = CJC \cdot \left( \frac{VJC}{VJC(t)} \right)^{MJC} \]
\[ CJS(t) = CJS \cdot \left( \frac{VJS}{VJS(t)} \right)^{MJS} \]

where:
\[ VJE(t) = VJE - TVJE \cdot \Delta t \]
\[ VJC(t) = VJC - TVJC \cdot \Delta t \]
\[ VJS(t) = VJS - TVJS \cdot \Delta t \]

TLEVC=3

\[ CJE(t) = CJE \cdot \left( 1 - 0.5 \cdot \frac{dvjedt}{VJE} \cdot \Delta t \right) \]
Using BJT Temperature Compensation Equations

\[ CJC(t) = CJC \cdot \left(1 - 0.5 \cdot dvjcdt \cdot \frac{\Delta t}{VJC}\right) \]

\[ CJS(t) = CJS \cdot \left(1 - 0.5 \cdot dvjsdt \cdot \frac{\Delta t}{VJS}\right) \]

\[ VJE(t) = VJE + dvjedt \cdot \Delta t \]

\[ VJC(t) = VJC + dvjcdt \cdot \Delta t \]

\[ VJS(t) = VJS + dvjsdt \cdot \Delta t \]

where TLEV = 0, 1, or 3

\[ dvjedt = \frac{egnom + 3 \cdot vt(t_{nom}) + (1.16 - egnom) \cdot \left(2 - \frac{t_{nom}}{t_{nom} + 1108}\right) - VJE}{tnom} \]

\[ dvjcdt = \frac{egnom + 3 \cdot vt(t_{nom}) + (1.16 - egnom) \cdot \left(2 - \frac{t_{nom}}{t_{nom} + 1108}\right) - VJC}{tnom} \]

\[ dvjsdt = \frac{egnom + 3 \cdot vt(t_{nom}) + (1.16 - egnom) \cdot \left(2 - \frac{t_{nom}}{t_{nom} + 1108}\right) - VJS}{tnom} \]

and TLEV = 2

\[ dvjedt = \frac{egnom + 3 \cdot vt(t_{nom}) + (EG - egnom) \cdot \left(2 - \frac{t_{nom}}{t_{nom} + GAP2}\right) - VJE}{tnom} \]

\[ dvjcdt = \frac{egnom + 3 \cdot vt(t_{nom}) + (EG - egnom) \cdot \left(2 - \frac{t_{nom}}{t_{nom} + GAP2}\right) - VJC}{tnom} \]

\[ dvjsdt = \frac{egnom + 3 \cdot vt(t_{nom}) + (EG - egnom) \cdot \left(2 - \frac{t_{nom}}{t_{nom} + GAP2}\right) - VJS}{tnom} \]
Using Parasitic Resistor Temperature Equations

The parasitic resistors, as a function of temperature regardless of TLEV value, are determined as:

\[ RE(t) = RE \cdot (1 + TRE_1 \cdot \Delta t + TRE_2 \cdot \Delta t^2) \]
\[ RB(t) = RB \cdot (1 + TRB_1 \cdot \Delta t + TRB_2 \cdot \Delta t^2) \]
\[ RBM(t) = RBM \cdot (1 + TRM_1 \cdot \Delta t + TRM_2 \cdot \Delta t^2) \]
\[ RC(t) = RC \cdot (1 + TRC_1 \cdot \Delta t + TRC_2 \cdot \Delta t^2) \]

Using BJT LEVEL=2 Temperature Equations

The model parameters of BJT LEVEL 2 model are modified for temperature compensation as:

\[ GAMMA(t) = GAMMA \cdot e^{fac\cdot ln} \]
\[ RC(t) = RC \cdot \left(\frac{t}{tnom}\right)^{BEX} \]
\[ VO(t) = VO \cdot \left(\frac{t}{tnom}\right)^{BEXV} \]
Using the BJT Quasi-Saturation Model

Use the BJT quasi-saturation model (LEVEL=2), an extension of the Gummel-Poon model (LEVEL 1 model), to model bipolar junction transistors that exhibit quasi-saturation or base push-out effects. When a device with lightly doped collector regions operates at high injection levels, the internal base-collector junction is forward biased, while the external base-collector junction is reversed biased; DC current gain and the unity gain frequency $f_T$ falls sharply. Such an operation regime is referred to as quasi-saturation, and its effects have been included in this model.

Figure 16-11 show the additional elements of the LEVEL 2 model. The current source $I_{epli}$ and charge storage elements $C_i$ and $C_x$ model the quasi-saturation effects. The parasitic substrate bipolar transistor is also included in the vertical transistor by the diode D and current source $I_{bs}$.

**Figure 16-11: Vertical npn Bipolar Transistor (SUBS=+1)**

![Diagram](image)
Using Epitaxial Current Source \( I_{epi} \)

The epitaxial current value, \( I_{epi} \), is determined by the equation:

\[
I_{epi} = \frac{ki-kx - \ln\left(\frac{1+ki}{1+kx}\right) + \frac{v_{bc} - v_{bcx}}{NEPI \cdot vt}}{\left(\frac{RC_{eff}}{NEPI \cdot vt}\right) \cdot \left(1 + \frac{v_{bc} - v_{bcx}}{VO}\right)}
\]
where:

\[ k_i = \left[ 1 + \text{GAMMA} \cdot e^{v_{bc} / (\text{NEPI} \cdot vt)} \right]^{1/2} \]

\[ k_x = \left[ 1 + \text{GAMMA} \cdot e^{v_{bcx} / (\text{NEPI} \cdot vt)} \right]^{1/2} \]

In special cases when the model parameter GAMMA is set to zero, \( k_i \) and \( k_x \) become one and,

\[ I_{\text{epi}} = \frac{v_{bc} - v_{bcx}}{R \text{C} \text{eff} \cdot \left( 1 + \frac{v_{bc} - v_{bcx}}{V_O} \right)} \]

**Using Epitaxial Charge Storage Elements \( \text{Ci} \) and \( \text{Cx} \)**

The epitaxial charges are determined by:

\[ q_i = Q \text{C} \text{Oeff} \cdot \left( k_i - 1 - \frac{\text{GAMMA}}{2} \right) \]

and:

\[ q_x = Q \text{C} \text{Oeff} \cdot \left( k_x - 1 - \frac{\text{GAMMA}}{2} \right) \]

The corresponding capacitances are calculated as:

\[ C_i = \frac{\partial}{\partial v_{bc}} (q_i) = \left( \frac{\text{GAMMA} \cdot Q \text{C} \text{Oeff}}{2 \cdot \text{NEPI} \cdot vt \cdot k_x} \right) \cdot e^{v_{bc} / (\text{NEPI} \cdot vt)} \]

and:

\[ C_x = \frac{\partial}{\partial v_{bcx}} (q_x) = \left( \frac{\text{GAMMA} \cdot Q \text{C} \text{Oeff}}{2 \cdot \text{NEPI} \cdot vt \cdot k_x} \right) \cdot e^{v_{bcx} / (\text{NEPI} \cdot vt)} \]

In the special case where GAMMA=0 the \( \text{Ci} \) and \( \text{Cx} \) become zero.
Example

*quasisat.sp comparison of bjt LEVEL1 and LEVEL2
*model
'options nomod relv=.001 reli=.001 absv=.1u absi=1p
'options post
q11 10 11 0 mod1
q12 10 12 0 mod2
q21 10 21 0 mod1
q22 10 22 0 mod2
q31 10 31 0 mod1
q32 10 32 0 mod2
vcc 10 0 .7
i11 0 11 15u
i12 0 12 15u
i21 0 21 30u
i22 0 22 30u
i31 0 31 50u
i32 0 32 50u
/dc vcc 0 3 .1
/print dc vcc=par('v(10)') i(q11) i(q12) i(q21)
+ i(q22) i(q31) i(q32)
/*.graph dc i(q11) i(q12) i(q21) i(q22)
/*.graph dc i(q11) i(q12)
/model MOD1 NPn IS=4.0E-16 BF=75 VAF=75
+ LEVEL=1 rc=500 SUBS=+1
/model MOD2 NPn IS=4.0E-16 BF=75 VAF=75
+ LEVEL=2 rc=500 vo=1 qco=1e-10
+ gamma=1e-9 SUBS=+1
/end
Figure 16-13: Comparing BJT LEVEL 1 and LEVEL 2 Models
Converting National Semiconductor Models

National Semiconductor’s SNAP circuit simulator has a scaled BJT model that is not the same as that used by Star-Hspice. To use this model with Star-Hspice, make the following changes.

For a subcircuit that consists of the scaled BJT model, the subcircuit name must be the same as the name of the model. Inside the subcircuit there is a .PARAM statement that specifies the scaled BJT model parameter values. Put a scaled BJT model inside the subcircuit, then change the “.MODEL mname mtype” statement to a .PARAM statement. Ensure that each parameter in the .MODEL statement within the subcircuit has a value in the .PARAM statement.

Defining Scaled BJT Subcircuits

The following subcircuit definition converts the National Semiconductor scaled BJT model to a form usable in Star-Hspice. The .PARAM parameter inside the .SUBCKT represents the .MODEL parameter in the National circuit simulator. Therefore, replace the “.MODEL mname” statement with a .PARAM statement. Change the model name to SBJT.

**Note:** All the parameters used in the following model must have a value that comes either from a .PARAM statement or the subcircuit call.

Example

```
.SUBCKT SBJT NC NB NE SF=1 SCBC=1 SCBE=1 SCCS=1 SIES=1 SICS=1 + SRB=1 SRC=1 SRE=1 SIC=0 SVCE=0 SBET=1 Q NC NB NE SBJT IC=SIC VCE=SVCE
.PARAM IES=1 10E-18 ICS=5.77E-18NE=1.02 NC=1.03 + ME=3.61 MC=1.24 EG=1.12 NSUB=0 + CJE=1E-15 CJC=1E-15 CSUB=1E-15 EXE=0.501 + EXC=0.222 ESUB=0.709 FE=1.16 PC=0.37 + PSUB=0.698 RE=75 RC=0.0 RB=1.0 + TRE=2E-3 TRC=6E-3 TRB=1.9E-3 VA=25 + FTF=2.8E9 FTR=40E6 BR=1.5 TCB=5.3E-3 + TCB2=1.6E-6 BF1=9.93 BF2=45.7 BF3=55.1
```
Converting National Semiconductor Models Using BJT Models

+ BF4=56.5  BF5=53.5  BF6=33.8
+ IBF1=4.8P  IBF2=1.57N  IBF3=74N
+ IBF4=3.13U  IBF5=64.2U  IBF6=516U
*

.CODE

.MODEL SBJT NPN
+ IBE='IES*SF*SIES' IBC='ICS*SF*SICS'
+ CJ1='CJE*SF*SCBE' CJ2='CJC*SF*SCBC'
+ CJS='CSUB*SF*SCCS' RB='RB*SRB/SF'
+ RC='RC*SRC/SF' RE='RE*SRE/SF'
+ TF='1/(6.28*FTF)' TR='1/(6.28*FTR)'
+ MJE=EXE MJC=EXC
+ MJS=ESUB VJE=PE
+ VJC=PC VJS=PSUB
+ NF=NE NR=NC
+ EG=EG BR=BR VAF=VA
+ TRE1=TRE TRC1=TRC TRB1=TRB
+ TBF1=TCB TBF2=TCB2
+ BF0=BF1 IB0=IBF1
+ BF1=BF2 IB1=IBF2
+ BF2=BF3 IB2=IBF3
+ BF3=BF4 IB3=IBF4
+ BF4=BF5 IB4=IBF5
+ BF5=BF6 IB5=IBF6
+ NSUB=0 sbet=sbet
+ TLEV=1 TLEVC=1
+ XTIR='MC*NC' XTI='ME*NE'
.ENDS SBJT

The BJT statement is replaced by:

XQ1 1046 1047 8 SBJT SIES=25.5 SICS=25.5 SRC=3.92157E-2
+ SRE=3.92157E-2 SBET=3.92157E-2 SRB=4.8823E+2 SCBE=94.5234
+ SCBC=41.3745 SCCS=75.1679 SVCE=1

Using the VBIC Bipolar Transistor Model

The VBIC (Vertical Bipolar Inter-Company) model is a new bipolar transistor model for Star-Hspice. You can use VBIC by specifying parameter LEVEL=4 for the bipolar transistor model.

VBIC addresses many problems of the SPICE Gummel-Poon model:
- More accurate modeling of Early effect
- Parasitic substrate transistor
- Modulation of collector resistance
- Avalanche multiplication in collector junction, parasitic capacitances of base-emitter overlap in double poly BJTs, and self heating.

Understanding the History of VBIC

VBIC was developed by engineers at several companies. The detailed equations\(^1\) for all elements are given in the referenced publication. Recent information and source code can be found on the web site:

http://www-sm.rz.fht-esslingen.de/institute/iafgp/neu/VBIC/index.html

Our implementation is compliant to standard VBIC. Self-heating and excess phases have been implemented or enabled in this version 99.4.

The large signal equivalent circuit for VBIC is shown in Figure 16-14. Capacitors CBCO, CBEO and resistors RCX, RBX, RE, and RS are linear elements, all other elements of the equivalent circuit are nonlinear.

VBIC Parameters

Table 16-5 lists the parameters for the model that you can set. Table 16-5 also contains the default values for the parameters. The same parameter names are used in the table and the previous referenced publication.

If values of parameters given by the user are beyond their ranges, those parameters will be reset to new values and warnings will be printed unless the option NOWARN is set.

**Noise Analysis**

The following sources of noise are taken into account:

- The thermal noise of resistors RBX, RCX, RE, RS, RBP, RCI, RBI
- Shot noise of currents IBE, IBEP, ICC, ICCP
- Flicker noise due to currents IBE, IBEP
The noise due to IBEX and IGC is not included in this preliminary version (nor in the standard VBIC), but will be included in the next release.

**Accounting for Self-heating and excess phase**

After self-heating effect is accounted for, the device element syntax becomes:

\[ Qxxx \text{ nc nb ne } <\text{ns}> <\text{nT}> \text{ mname } <\text{regular parameters}> <\text{tnodeout}> \]

where \(nT\) is the node for temperature. If this node is given, but \(ns\) is not given, the flag “\(\text{tnodeout}\)” must be specified to indicate the fourth node is temperature node instead of substrate node. To turn on self-heating, in addition to giving the \(T\) node, the model parameter \(Rth\) must be not zero in the model card.

Excess phase has only effects on ac and transient characteristics analysis. To turn on this effect, the model parameter \(TD\) must be non-zero. But for transient analysis, to turn on excess phase is not recommended due to model’s convergence very sensitive to \(TD\) value.

**Example**

Example with no self-heating effect.

Usage:

```
Q1 21 22 22 22 VBIC <parameters>
.MODEL VBIC NPN <parameters>
```

Complete netlist:

```
*VBIC example, DC analysis
.OPTIONS NODE POST NOPAGE
.WIDTH OUT=80
.DC QVcolem 0 5 0.1 SWEEP QVbasem 0.7 0.86 0.05
.TEMP -20.0 +25. +100.
.PRINT DC I1(Q1) I2(Q1) I3(Q1) I4(Q1)
.PRINT DC V(102) V(202)
Vbas 101 0 QVbasem
Vcol 102 0 QVcolem
```
Example with self-heating effects.

*# VERSION: 99.4

.option absmos=1e-12 relmos=1e-6 relv=1e-6 absv=1e-9
vc c 0 0
vb b 0 0
ve e 0 0
vs s 0 0

vc1 c1 c 0
vb1 b1 b 0
ve1 e1 e 0
vs1 s1 s 0
*vt t 0 1meg

.temp 27

Q1 c1 b1 e1 s1 t mod1 area=1 tnodeout

.model mod1 npn LEVEL=4
+ Tnom=27 RCX=10 RCI=60 VO=2 GAMM=2.e-11
+ HRCF=2 RBX=10 RBI=40 RE=2
+ RS=20 RBP=40 IS=1e-16 NF=1.00000e+00
+ NR=1.00000e+00 FC=9.00000e-01 CBEO=0
+ CJE=1.e-13 PE=0.75 ME=0.33
+ AJE=-5.00000e-01 CBCO=0 CJC=2e-14
+ QCO=1e-12 CJEP=1e-13 PC=7.50000e-01
+ MC=3.30000e-01 AJC=-5.00000e-01 CJCP=4e-13
+ PS=7.50000e-01 MS=3.30000e-01 AJS=-5.00000e-01
+ IBEI=1e-18 WBE=1.0000 NEI=1.00000e+00
+ IBEN=5e-15 NEN=2.00000e+00 IBCI=2e-17
+ NCI=1.00000e+00 IBCN=5e-15 NCN=2.00000e+00
+ AVC1=2 AVC2=15 ISP=1e-15
+ WSP=1.00000e+00 NFP=1.00000e+00 IBEIP=0
+ IBENP=0 IBCIP=0 NCIP=1.00000e+00
Using the VBIC Bipolar Transistor Model

+ IBCNP=0 NCNP=2.00000e+00 VEF=10
+ VER=4 IKF=0.002 IKR=0.0002 IKP=0.0002
+ TF=1.e-11 QTF=0 XTF=20
+ VTF=0 ITF=0.08 TR=1e-10
+ KFN=0 AFN=1.0e+00
+ BFN=1.0000e+00 XRE=0 XRB=0
+ XRC=0 XRS=0 XVO=0
+ EA=1.12000e+00 EAIE=1.12000e+00
+ EANE=1.12000e+00 EANC=1.12000e+00
+ EANS=1.12000e+00 XIS=3.00000e+00
+ XII=3.00000e+00 XIN=3.00000e+00
+ TNF=0 TAVC=0
+ RTH=300 CTH=0
+ TD=0
++ TD=2.e-11

.dc vc 0.0 5.0001 0.05 vb 0.7 1.0001 0.05
.print i(vc) i(vb) v(t)
.end

where v(t) print out the device temperature using T node.
### Table 16-5: Default Model Parameters for BJT, LEVEL 4

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFN</td>
<td>1</td>
<td>Flicker noise exponent for current</td>
<td></td>
</tr>
<tr>
<td>AJC</td>
<td>-0.5</td>
<td>Base-collector capacitance switching parameter</td>
<td></td>
</tr>
<tr>
<td>AJE</td>
<td>-0.5</td>
<td>Base-emitter capacitance switching parameter</td>
<td></td>
</tr>
<tr>
<td>AJS</td>
<td>-0.5</td>
<td>Substrate-collector capacitance switching parameter</td>
<td></td>
</tr>
<tr>
<td>AVC1</td>
<td>V(^{-1})</td>
<td>0</td>
<td>Base-collector weak avalanche parameter 1</td>
</tr>
<tr>
<td>AVC2</td>
<td>V(^{-1})</td>
<td>0</td>
<td>Base-collector weak avalanche parameter 2</td>
</tr>
<tr>
<td>BFN</td>
<td></td>
<td>1</td>
<td>Flicker noise exponent for 1/f dependence</td>
</tr>
<tr>
<td>CBCO (CBC0)</td>
<td>F</td>
<td>0</td>
<td>Extrinsic base-collector overlap capacitance</td>
</tr>
<tr>
<td>CBEO (CBE0)</td>
<td>F</td>
<td>0</td>
<td>Extrinsic base-emitter overlap capacitance</td>
</tr>
<tr>
<td>CJC</td>
<td>F</td>
<td>0</td>
<td>Base-collector intrinsic zero bias capacitance</td>
</tr>
<tr>
<td>CJCP</td>
<td>F</td>
<td>0</td>
<td>Substrate-collector zero bias capacitance</td>
</tr>
<tr>
<td>CJE</td>
<td>F</td>
<td>0</td>
<td>Base-emitter zero bias capacitance</td>
</tr>
<tr>
<td>CJEP</td>
<td>F</td>
<td>0</td>
<td>Base-collector extrinsic zero bias capacitance</td>
</tr>
<tr>
<td>CTH</td>
<td>J/K</td>
<td>0</td>
<td>Thermal capacitance</td>
</tr>
<tr>
<td>EA</td>
<td>eV</td>
<td>1.12</td>
<td>Activation energy for IS</td>
</tr>
<tr>
<td>EAIC</td>
<td>eV</td>
<td>1.12</td>
<td>Activation energy for IBCI/IBEIP</td>
</tr>
<tr>
<td>EAIE</td>
<td>eV</td>
<td>1.12</td>
<td>Activation energy for IBEI</td>
</tr>
<tr>
<td>EAIS</td>
<td>eV</td>
<td>1.12</td>
<td>Activation energy for IBCIP</td>
</tr>
<tr>
<td>EANC</td>
<td>eV</td>
<td>1.12</td>
<td>Activation energy for IBCN/IBENP</td>
</tr>
<tr>
<td>EANE</td>
<td>eV</td>
<td>1.12</td>
<td>Activation energy for IBEN</td>
</tr>
<tr>
<td>EANS</td>
<td>eV</td>
<td>1.12</td>
<td>Activation energy for IBCNP</td>
</tr>
<tr>
<td>FC</td>
<td></td>
<td>0.9</td>
<td>Forward bias depletion capacitance limit</td>
</tr>
</tbody>
</table>
## Table 16-5: Default Model Parameters for BJT, LEVEL 4

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAMM</td>
<td></td>
<td>0</td>
<td>Epi doping parameter</td>
</tr>
<tr>
<td>HRCF</td>
<td></td>
<td>1</td>
<td>High-current RC factor</td>
</tr>
<tr>
<td>IBCI</td>
<td>A</td>
<td>1e-16</td>
<td>Ideal base-collector saturation current</td>
</tr>
<tr>
<td>IBCIP</td>
<td>A</td>
<td>0</td>
<td>Ideal parasitic base-collector saturation current</td>
</tr>
<tr>
<td>IBCN</td>
<td>A</td>
<td>1e-15</td>
<td>Non-ideal base-collector saturation current</td>
</tr>
<tr>
<td>IBCNP</td>
<td>A</td>
<td>0</td>
<td>Non-ideal parasitic base-collector saturation current</td>
</tr>
<tr>
<td>IBEI</td>
<td>A</td>
<td>1e-18</td>
<td>Ideal base-emitter saturation current</td>
</tr>
<tr>
<td>IBEIP</td>
<td>A</td>
<td>0</td>
<td>Ideal parasitic base-emitter saturation current</td>
</tr>
<tr>
<td>IBEN</td>
<td>A</td>
<td>1e-15</td>
<td>Non-ideal base-emitter saturation current</td>
</tr>
<tr>
<td>IBENP</td>
<td>A</td>
<td>0</td>
<td>Non-ideal parasitic base-emitter saturation current</td>
</tr>
<tr>
<td>IKF</td>
<td>A</td>
<td>2e-3</td>
<td>Forward knee current</td>
</tr>
<tr>
<td>IKP</td>
<td>A</td>
<td>2e-4</td>
<td>Parasitic knee current</td>
</tr>
<tr>
<td>IKR</td>
<td>A</td>
<td>2e-4</td>
<td>Reverse knee current</td>
</tr>
<tr>
<td>IS</td>
<td>A</td>
<td>1e-16</td>
<td>Transport saturation current</td>
</tr>
<tr>
<td>ISP</td>
<td>A</td>
<td>1e-16</td>
<td>Parasitic transport saturation current</td>
</tr>
<tr>
<td>ITF</td>
<td>A</td>
<td>1e-3</td>
<td>Coefficient of TF dependence in Ic</td>
</tr>
<tr>
<td>KFN</td>
<td></td>
<td>0</td>
<td>Base-emitter flicker noise constant</td>
</tr>
<tr>
<td>MC</td>
<td></td>
<td>0.33</td>
<td>Base-collector grading coefficient</td>
</tr>
<tr>
<td>ME</td>
<td></td>
<td>0.33</td>
<td>Base-emitter grading coefficient</td>
</tr>
<tr>
<td>MS</td>
<td></td>
<td>0.33</td>
<td>Substrate-collector grading coefficient</td>
</tr>
<tr>
<td>NCI</td>
<td></td>
<td>1</td>
<td>Ideal base-collector emission coefficient</td>
</tr>
<tr>
<td>NCIP</td>
<td></td>
<td>1</td>
<td>Ideal parasitic base-collector emission coefficient</td>
</tr>
<tr>
<td>NCN</td>
<td></td>
<td>2</td>
<td>Non-ideal base-collector emission coefficient</td>
</tr>
</tbody>
</table>
Table 16-5: Default Model Parameters for BJT, LEVEL 4

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCNP</td>
<td></td>
<td>2</td>
<td>Non-ideal parasitic base-collector emission coefficient</td>
</tr>
<tr>
<td>NEI</td>
<td></td>
<td>1</td>
<td>Ideal base-emitter emission coefficient</td>
</tr>
<tr>
<td>NEN</td>
<td></td>
<td>2</td>
<td>Non-ideal base-emitter emission coefficient</td>
</tr>
<tr>
<td>NF</td>
<td></td>
<td>1</td>
<td>Forward emission coefficient</td>
</tr>
<tr>
<td>NFP</td>
<td></td>
<td>1</td>
<td>Parasitic forward emission coefficient</td>
</tr>
<tr>
<td>NR</td>
<td></td>
<td>1</td>
<td>Reverse emission coefficient</td>
</tr>
<tr>
<td>PC</td>
<td>V</td>
<td>0.75</td>
<td>Base-collector built-in potential</td>
</tr>
<tr>
<td>PE</td>
<td>V</td>
<td>0.75</td>
<td>Base-emitter built-in potential</td>
</tr>
<tr>
<td>PS</td>
<td>V</td>
<td>0.75</td>
<td>Substrate-collector built-in potential</td>
</tr>
<tr>
<td>QCO (QC0)</td>
<td>C</td>
<td>0</td>
<td>Epi charge parameter</td>
</tr>
<tr>
<td>QTF</td>
<td></td>
<td>0</td>
<td>Variation of TF with base-width modulation</td>
</tr>
<tr>
<td>RBI</td>
<td>Ohm</td>
<td>1e-1</td>
<td>Intrinsic base resistance</td>
</tr>
<tr>
<td>RBP</td>
<td>Ohm</td>
<td>1e-1</td>
<td>Parasitic base resistance</td>
</tr>
<tr>
<td>RBX</td>
<td>Ohm</td>
<td>1e-1</td>
<td>Extrinsic base resistance</td>
</tr>
<tr>
<td>RCI</td>
<td>Ohm</td>
<td>1e-1</td>
<td>Intrinsic collector resistance</td>
</tr>
<tr>
<td>RCX</td>
<td>Ohm</td>
<td>1e-1</td>
<td>Extrinsic collector resistance</td>
</tr>
<tr>
<td>RE</td>
<td>Ohm</td>
<td>1e-1</td>
<td>Emitter resistance</td>
</tr>
<tr>
<td>RS</td>
<td>Ohm</td>
<td>1e-1</td>
<td>Substrate resistance</td>
</tr>
<tr>
<td>RTH</td>
<td>K/W</td>
<td>0</td>
<td>Thermal resistance</td>
</tr>
<tr>
<td>TAVC</td>
<td>1/K</td>
<td>0</td>
<td>Temperature coefficient of AVC2</td>
</tr>
<tr>
<td>TD</td>
<td>s</td>
<td>0</td>
<td>Forward excess-phase delay time</td>
</tr>
<tr>
<td>TF</td>
<td>s</td>
<td>1e-11</td>
<td>Forward transit time</td>
</tr>
</tbody>
</table>
### Table 16-5: Default Model Parameters for BJT, LEVEL 4

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNF</td>
<td>1/K</td>
<td>0</td>
<td>Temperature coefficient of NF</td>
</tr>
<tr>
<td>TR</td>
<td>s</td>
<td>1e-11</td>
<td>Reverse transit time</td>
</tr>
<tr>
<td>TREF (TNOM)</td>
<td>°C</td>
<td>27</td>
<td>Nominal measurement temperature of parameters (please do not use TNOM alias, though it is allowed)</td>
</tr>
<tr>
<td>VEF</td>
<td>V</td>
<td>0</td>
<td>Forward Early voltage</td>
</tr>
<tr>
<td>VER</td>
<td>V</td>
<td>0</td>
<td>Reverse Early voltage</td>
</tr>
<tr>
<td>VO (V0)</td>
<td>V</td>
<td>0</td>
<td>Epi drift saturation voltage</td>
</tr>
<tr>
<td>VTF</td>
<td>V</td>
<td>0</td>
<td>Coefficient of TF dependence on Vbc</td>
</tr>
<tr>
<td>WBE</td>
<td>1</td>
<td></td>
<td>Portion of IBEI from Vbei, 1-WBE from Vbex</td>
</tr>
<tr>
<td>WSP</td>
<td>1</td>
<td></td>
<td>Portion of ICCP from Vbep, 1-WSP from Vbci</td>
</tr>
<tr>
<td>XII</td>
<td>3</td>
<td></td>
<td>Temperature exponent of IBEI/IBCI/IBEIP/IBCIP</td>
</tr>
<tr>
<td>XIN</td>
<td>3</td>
<td></td>
<td>Temperature exponent of IBEN/IBCN/IBENP/IBCNP</td>
</tr>
<tr>
<td>XIS</td>
<td>3</td>
<td></td>
<td>Temperature exponent of IS</td>
</tr>
<tr>
<td>XRB</td>
<td>1</td>
<td></td>
<td>Temperature exponent of base resistance</td>
</tr>
<tr>
<td>XRC</td>
<td>1</td>
<td></td>
<td>Temperature exponent of collector resistance</td>
</tr>
<tr>
<td>XRE</td>
<td>1</td>
<td></td>
<td>Temperature exponent of emitter resistance</td>
</tr>
<tr>
<td>XRS</td>
<td>1</td>
<td></td>
<td>Temperature exponent of substrate resistance</td>
</tr>
<tr>
<td>XTF</td>
<td>0</td>
<td></td>
<td>Coefficient of TF bias dependence</td>
</tr>
<tr>
<td>XVO (XV0)</td>
<td>0</td>
<td></td>
<td>Temperature exponent of VO</td>
</tr>
</tbody>
</table>
Notes on Using VBIC

1. Set LEVEL to 4 to identify the model as a VBIC bipolar junction transistor model.

2. The LEVEL 4 model does not scale with any area terms, and does not yet scale with M.

3. Setting these parameters to zero infers a value of infinity: HRCF, IKF, IKP, IKR, ITF, VEF, VER, VO, VTF.

4. Parameters CBC0, CBE0, QC0, TNOM, V0, and XV0 are aliases for CBCO, CBEO, QCO, TREF, VO, and XVO, respectively. Avant! discourages use of TNOM as a model parameter name as it is used as the name of the default room temperature in Star-Hspice.

5. The default room temperature is 25 degrees in Star-Hspice, but is 27 in some other simulators. If the VBIC bipolar junction transistor model parameters are specified at 27 degrees, TREF=27 should be added to the model, so that the model parameters will be interpreted correctly. It is a matter of choice whether or not to set the nominal simulation temperature to 27, by adding .OPTION TNOM=27 to the netlist. This should be done when testing Star-Hspice versus other simulators that use 27 as the default room temperature.

6. Pole-zero simulation of this model is not supported.

7. For this version of implementation, all seven internal resistors should have values greater than or equal to $1.0 \times 10^{-3}$. Values smaller than this will be reassigned a value of $1.0 \times 10^{-3}$. 
LEVEL 6 Philips Bipolar Model (MEXTRAM LEVEL 503)

The Philips bipolar model (MEXTRAM LEVEL 503) is now installed in Star-Hspice as BJT LEVEL 6. The MEXTRAM covers several effects that are not included in, e.g., the original Gummel-Poon model. These effects include:

- Temperature
- Charge storage
- Substrate
- Parasitic PNP
- High-injection
- Built-in electric field in base region
- Bias-dependent Early effect
- Low-level, non-ideal base currents
- Hard- and quasi-saturation
- Weak avalanche
- Hot carrier effects in the collector epilayer
- Explicit modeling of inactive regions
- Split base-collector depletion capacitance
- Current crowding and conductivity modulation for base resistance
- First order approximation of distributed high frequency effects in the intrinsic base (high frequency current crowding and excess phase shift)

The description for this model can be found at “http://www-us.semiconductors.com/Philips_Models/”.

LEVEL 6 Philips Bipolar Model (MEXTRAM LEVEL 503) Using BJT Models
LEVEL 6 Element Syntax

**General Form**

Qxxx nc nb ne <ns> mname <AREA=val>
+ <OFF<VBE=val><VCE=val> <M=val>
+ <DTEMP=val>

where the angle brackets indicate optional parameters.

The arguments are as follows:

- **Qxxx**: BJT element name. Must begin with Q, which can be followed by up to 1023 alphanumeric characters.
- **nc**: Collector terminal node name or number.
- **nb**: Base node name or number.
- **ne**: Emitter terminal node name or number.
- **ns**: Substrate node name or number.
- **mname**: BJT model name reference.
- **AREA**: The normalized emitter area.
- **OFF**: Sets initial condition to OFF for this element in DC analysis.
- **VBE**: Initial internal base to emitter voltage.
- **VCE**: Initial internal collector to emitter voltage.
- **M**: Multiplier to simulate multiple BJTs in parallel.
- **DTEMP**: The difference between element and circuit temperature.

**LEVEL 6 Model Parameters**

The following tables describe MEXTRAM as LEVEL 6 model parameters including parameters name, descriptions, units, default values and notes.
### Flags

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td>-</td>
<td>-</td>
<td>LEVEL 6 for MEXTRAM</td>
</tr>
<tr>
<td>EXAVL</td>
<td>-</td>
<td>0</td>
<td>Flag for extended modeling of avalanche currents</td>
</tr>
<tr>
<td>EXMOD</td>
<td>-</td>
<td>0</td>
<td>Flag for extended modeling of the reverse current gain</td>
</tr>
<tr>
<td>EXPHI</td>
<td>-</td>
<td>1</td>
<td>Flag for distributed high frequency effects</td>
</tr>
</tbody>
</table>

### Basic Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREF</td>
<td>°C</td>
<td>0.0</td>
<td>Model nominal temperature</td>
</tr>
<tr>
<td>IS</td>
<td>A</td>
<td>5.E-17</td>
<td>Collector-emitter saturation current</td>
</tr>
<tr>
<td>BF</td>
<td>A</td>
<td>140.0</td>
<td>Ideal forward current gain</td>
</tr>
<tr>
<td>XIBI</td>
<td>-</td>
<td>0.0</td>
<td>Fraction of ideal base current that belongs to the sidewall</td>
</tr>
<tr>
<td>IBF</td>
<td>A</td>
<td>2.0E-14</td>
<td>Saturation current of the non-ideal forward base current</td>
</tr>
<tr>
<td>VLF</td>
<td>V</td>
<td>0.5</td>
<td>Cross-over voltage of the non-ideal forward base current</td>
</tr>
<tr>
<td>IK</td>
<td>A</td>
<td>15.E-3</td>
<td>High-injection knee current</td>
</tr>
<tr>
<td>BRI</td>
<td>-</td>
<td>16.0</td>
<td>Ideal reverse current gain</td>
</tr>
<tr>
<td>IBR</td>
<td>A</td>
<td>8.0e-15</td>
<td>Saturation current of the non-ideal reverse base current</td>
</tr>
<tr>
<td>VLR</td>
<td>V</td>
<td>0.5</td>
<td>Cross-over voltage of the non-ideal reverse base current</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>XEX</td>
<td></td>
<td>0.5</td>
<td>Part of I EX ,Q EX ,Q TEX and I SUB that depends on the base-collector voltage VBC1</td>
</tr>
<tr>
<td>QBO</td>
<td>C</td>
<td>1.2e-12</td>
<td>Base charge at zero bias</td>
</tr>
<tr>
<td>ETA</td>
<td></td>
<td>4.0</td>
<td>Factor of the built-in field of the base</td>
</tr>
<tr>
<td>AVL</td>
<td></td>
<td>50.</td>
<td>Weak avalanche parameter</td>
</tr>
<tr>
<td>EFI</td>
<td></td>
<td>0.7</td>
<td>Electric field intercept (with EXAVL=1)</td>
</tr>
<tr>
<td>IHC</td>
<td>A</td>
<td>3.e-3</td>
<td>Critical current for hot carriers</td>
</tr>
<tr>
<td>RCC</td>
<td>ohm</td>
<td>25.</td>
<td>Constant part of the collector resistance</td>
</tr>
<tr>
<td>RCV</td>
<td>ohm</td>
<td>750.</td>
<td>Resistance of the unmodulated epilayer</td>
</tr>
<tr>
<td>SCRCV</td>
<td>ohm</td>
<td>1000.0</td>
<td>Space charge resistance of the epilayer</td>
</tr>
<tr>
<td>SFH</td>
<td></td>
<td>0.6</td>
<td>Current spreading factor epilayer</td>
</tr>
<tr>
<td>RBC</td>
<td>ohm</td>
<td>50.</td>
<td>Constant part of the base resistance</td>
</tr>
<tr>
<td>RBV</td>
<td>ohm</td>
<td>100.</td>
<td>Variable part of the base resistance at zero bias</td>
</tr>
<tr>
<td>RE</td>
<td>ohm</td>
<td>2.0</td>
<td>Emitter series resistance</td>
</tr>
<tr>
<td>TAUNE</td>
<td>s</td>
<td>3.e-10</td>
<td>Minimum delay time of neutral and emitter charge</td>
</tr>
<tr>
<td>MTAU</td>
<td></td>
<td>1.18</td>
<td>Non-ideality factor of the neutral and emitter charge</td>
</tr>
<tr>
<td>CJE</td>
<td>F</td>
<td>2.5e-13</td>
<td>Zero bias collector-base depletion capacitance</td>
</tr>
<tr>
<td>VDE</td>
<td>V</td>
<td>0.9</td>
<td>Emitter-base diffusion voltage</td>
</tr>
<tr>
<td>PE</td>
<td></td>
<td>0.33</td>
<td>Emitter-base grading coefficient</td>
</tr>
<tr>
<td>XCJE</td>
<td>F</td>
<td>0.5</td>
<td>Fraction of the emitter-base depletion capacitance that belongs to the sidewall</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>CJC</td>
<td>F</td>
<td>1.3e-13</td>
<td>Zero bias collector-base depletion capacitance</td>
</tr>
<tr>
<td>VDC</td>
<td>V</td>
<td>0.6</td>
<td>Collector-base diffusion voltage</td>
</tr>
<tr>
<td>PC</td>
<td>-</td>
<td>0.4</td>
<td>Collector-base grading coefficient variable part</td>
</tr>
<tr>
<td>XP</td>
<td>F</td>
<td>0.2</td>
<td>Constant part of CJC</td>
</tr>
<tr>
<td>MC</td>
<td>-</td>
<td>0.5</td>
<td>Collector current modulation coefficient</td>
</tr>
<tr>
<td>XCJC</td>
<td>-</td>
<td>0.1</td>
<td>Fraction of the collector-base depletion capacitance under the emitter area</td>
</tr>
<tr>
<td>VGE</td>
<td>V</td>
<td>1.01</td>
<td>Band-gap voltage of the emitter</td>
</tr>
<tr>
<td>VGB</td>
<td>V</td>
<td>1.18</td>
<td>Band-gap voltage of the base</td>
</tr>
<tr>
<td>VGC</td>
<td>V</td>
<td>1.205</td>
<td>Band-gap voltage of the collector</td>
</tr>
<tr>
<td>VGJ</td>
<td>V</td>
<td>1.1</td>
<td>Band-gap voltage recombination emitter-base junction</td>
</tr>
<tr>
<td>VI</td>
<td>V</td>
<td>0.040</td>
<td>Ionization voltage base dope</td>
</tr>
<tr>
<td>NA</td>
<td>cm^-3</td>
<td>3.0E17</td>
<td>Maximum base dope concentration</td>
</tr>
<tr>
<td>ER</td>
<td>-</td>
<td>2.E-3</td>
<td>Temperature coefficient of VLF and VLR</td>
</tr>
<tr>
<td>AB</td>
<td>-</td>
<td>1.35</td>
<td>Temperature coefficient resistivity of the base</td>
</tr>
<tr>
<td>AEPI</td>
<td>-</td>
<td>2.15</td>
<td>Temperature coefficient resistivity of the epilayer</td>
</tr>
<tr>
<td>AEX</td>
<td>-</td>
<td>1.</td>
<td>Temperature coefficient resistivity of the extrinsic base</td>
</tr>
<tr>
<td>AC</td>
<td>-</td>
<td>0.4</td>
<td>Temperature coefficient resistivity of the buried layer</td>
</tr>
<tr>
<td>KF</td>
<td>-</td>
<td>2.E-16</td>
<td>Flicker noise coefficient ideal base current</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>KFN</td>
<td>-</td>
<td>2.0E-16</td>
<td>Flicker noise coefficient non-ideal base current</td>
</tr>
<tr>
<td>AF</td>
<td>-</td>
<td>1.0</td>
<td>Flicker noise exponent</td>
</tr>
<tr>
<td>ISS</td>
<td>A</td>
<td>6.0E-16</td>
<td>Base-substrate saturation current</td>
</tr>
<tr>
<td>IKS</td>
<td>A</td>
<td>5.0E-6</td>
<td>Knee current of the substrate</td>
</tr>
<tr>
<td>CJS</td>
<td>F</td>
<td>1.0E-12</td>
<td>Zero bias collector-substrate depletion capacitance</td>
</tr>
<tr>
<td>VDS</td>
<td>V</td>
<td>0.5</td>
<td>Collector-substrate diffusion voltage</td>
</tr>
<tr>
<td>PS</td>
<td>-</td>
<td>0.33</td>
<td>Collector-substrate grading coefficient</td>
</tr>
<tr>
<td>VGS</td>
<td>V</td>
<td>1.15</td>
<td>Band-gap voltage of the substrate</td>
</tr>
<tr>
<td>AS</td>
<td>-</td>
<td>2.15</td>
<td>For a closed buried layer: AS=AC For an open buried layer: AS=AEPI</td>
</tr>
</tbody>
</table>
Example

*Simulation Input File
.options gmin=1e-13 gmindc=1e-13 POST=1 converge=1
QCKT 1 2 3 4 mextram area=1.0 m=1
* START SOURCES
VE 3 0 DC 0
VB 2 0 DC 0
VC 1 0 DC 0.1
VS 4 0 DC 0
.DC Vb 0.1 0.90001 0.1
.op
.PRINT DC I(VC) I(VB) I(VE) I(VS)
.temp 22
.model mextram npn LEVEL=6
+ TREF = 2.200000e+01
+ EXMOD= 1.000000e+00 EXPHI= 0.000000e+00
+ EXAVL= 1.000000e+00 IS = 9.602000e-18 BF = 1.381000e+02
+ XIBI = 0.000000e+00 IBF = 2.614800e-15 VLF = 6.164000e-01
+ IK = 1.500000e-02 BRI = 5.951000e+00 IBR = 4.606600e-14
+ VLR = 5.473000e-01 XEXT = 6.016000e-01 QBO = 9.439600e-14
+ ETA = 4.800000e+00 AVL = 6.329000e+01 EFI = 7.306000e-01
+ IHC = 4.541900e-04 RCV = 9.819000e+02 RCC=1.91e+01
+ SCRCV= 1.899000e+03 SFH = 3.556000e-01 RBC = 1.165000e+02
+ RBV = 3.077000e+02 RE = 2.525000e+00 TAUNE= 4.126600e-12
+ MTAU = 1.000000e+00 CJE = 4.909400e-14 VDE = 8.764000e-01
+ PE = 3.242000e-01 XCJE = 2.600000e-01 CJC = 8.539400e-14
+ VDC = 6.390000e-01 PC = 5.237000e-01 XP = 6.561000e-01
+ MC = 5.000000e-01 XCJC= 2.759700e-02 VGE = 1.129000e+00
+ VGB = 1.206000e+00 VGC = 1.120000e+00 VGJ = 1.129000e+00
+ VI = 2.100000e-02 NA = 4.400000e+17 ER = 2.000000e-03
+ AB = 1.000000e+00 AEPI =1.900000e+00 AEX= 3.100000e-01
+ AC = 2.600000e-01 KF =2.000000e-16 KFN = 2.000000e-16
+ AF = 1.000000e+00 ISS = 5.860200e-17 IKS = 6.481200e-06
+ CJS = 2.219600e-13 VDS = 5.156000e-01 PS = 3.299000e-01
+ VGS = 1.120000e+00 AS = 1.900000e+00
.END
LEVEL 6 Philips Bipolar Model (MEXTRAM LEVEL 504)

The new version of the MEXTRAM model has been added to Star-Hspice as BJT LEVEL 6. You can now specify the VERS parameter to choose MEXTRAM level 503 or 504. The default value of the VERS parameter is 504.

MEXTRAM 504 gives better results for the description of first and higher-order characteristic derivatives than MEXTRAM 503. This effect is noticeable in the output-conductance, the cut-off frequency, and the low-frequency third order distortion.

The MEXTRAM LEVEL 504 covers several effects that are not included in the original Gummel-Poon model. These effects include:

- Temperature
- Charge storage
- Substrate
- Parasitic PNP
- High-injection
- Bias-dependent early effect
- Low-level, non-ideal base currents
- Hard- and quasi-saturation (including Kirk Effect)
- Weak avalanche (optionally including snap-back behavior)
- Explicit modeling of inactive regions
- Split base-collector and base-emitter depletion capacitance
- Current crowding and conductivity modulation of the base resistance
- First order approximation of distributed high frequency effects in the intrinsic base (high frequency current crowding and excess phase shift)
- Ohmic resistance of epilayer
- Velocity saturation effects on the resistance of the epilayer
- Recombination in the base (meant for SiGe transistors)
- Early effects in the case of a graded bandgap (SiGe)
LEVEL 6 Philips Bipolar Model (MEXTRAM LEVEL 504)

- Thermal noise, shot noise, and 1/f-noise
- Self-heating

The description for this model can be found at:
http://www.semiconductors.philips.com/Philips_Models/newsflash/mextram504

Notes for HSPICE Users

The following information should be considered when using MEXTRAM 504 with Hspice:

- Set Level 6 to identify the model as a MEXTRAM bipolar junction transistor model
- Set VERS parameter to 503 to use MEXTRAM 503
- Set VERS parameter to 504 to use MEXTRAM 504
- Consider that all internal resistors are limited to greater than or equal to 1.0e-6
- Reference temperature, TREF, is equal to 25 degrees
- MEXTRAM does not contain extensive geometrical or process scaling rules (it has a multiplication factor to put transistors in parallel)
- MEXTRAM does not contain a substrate resistance
- Constant overlap capacitances are not modelled within MEXTRAM
- MEXTRAM 504 has better convergence than 503
- MEXTRAM is more complex than Gummel-Poon (the computation time will be longer and the convergence will be less)
- No reverse emitter-base breakdown mechanism
- Forward current of the parasitic PNP transistor is modelled
- Output conductance dIc/dVce at the point where hard saturation starts seems to be too abrupt for high current levels, compared to measurements
- Clarity of extrinsic current model describing Xiex and Xisub could be improved by adding an extra node and an extra contact base resistance. In this case, parameter extraction would be more difficult
Self-heating is not enabled for this model, so model parameters RTH and CTH have no influence.

LEVEL 6 Model Parameters (504)

The following tables describe MEXTRAM 504 as LEVEL 6 model parameters including parameter name, unit, default, description and notes.

TAUNE in MEXTRAM 503 acts as TAUE in the 504 model.

Parameters noted with a ‘*’ are not used in the DC model.

The following nine parameters are deleted from MEXTRAM 503:

- QBO
- VLF
- AVL
- ETA
- EFI
- VGE
- VI
- NA
- ER

The following 18 parameters have been added to MEXTRAM 504:

- VEF
- VER
- MLF
- WAVL
- VAVL
- AXI
- TAUE
- TAUB
- TEPI
- TAUR
- DEG
- XREC
- AE
- DVGBF
- DVGBR
- DVGTE
- RTH
- CTH
### Flags

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td>-</td>
<td>6</td>
<td>Model level</td>
</tr>
<tr>
<td>VERS</td>
<td>-</td>
<td>504</td>
<td>Flag for choosing MEXTRAM model (level 503 or 504)</td>
</tr>
<tr>
<td>EXMOD</td>
<td>-</td>
<td>1</td>
<td>Flag for extended modeling of the reverse current gain</td>
</tr>
<tr>
<td>EXPHI</td>
<td>-</td>
<td>1</td>
<td>*Flag for distributed high frequency effects in transient</td>
</tr>
<tr>
<td>EXAVL</td>
<td>-</td>
<td>0</td>
<td>Flag for extended modeling of avalanche currents</td>
</tr>
<tr>
<td>TREF</td>
<td>^°C</td>
<td>25.0</td>
<td>Reference temperature</td>
</tr>
</tbody>
</table>

### Basic Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS</td>
<td>A</td>
<td>2.2e-17</td>
<td>Collector-emitter saturation current</td>
</tr>
<tr>
<td>VER</td>
<td>-</td>
<td>2.5</td>
<td>Reverse early voltage</td>
</tr>
<tr>
<td>VEF</td>
<td>-</td>
<td>44.0</td>
<td>Forward early voltage</td>
</tr>
<tr>
<td>BF</td>
<td>-</td>
<td>215.0</td>
<td>Ideal forward current gain</td>
</tr>
<tr>
<td>XIBI</td>
<td>-</td>
<td>0.0</td>
<td>Fraction of ideal base current that belongs to the sidewall</td>
</tr>
<tr>
<td>IBF</td>
<td>A</td>
<td>2.7e-15</td>
<td>Saturation current of the non-ideal forward base current</td>
</tr>
<tr>
<td>MLF</td>
<td>V</td>
<td>2.0</td>
<td>Non-ideality factor of the non-ideal forward base current</td>
</tr>
</tbody>
</table>
### Basic Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IK</td>
<td>A</td>
<td>0.1</td>
<td>Collector-emitter high injection knee current</td>
</tr>
<tr>
<td>BRI</td>
<td>-</td>
<td>7.0</td>
<td>Ideal reverse current gain</td>
</tr>
<tr>
<td>IBR</td>
<td>A</td>
<td>1.0e-15</td>
<td>Saturation current of the non-ideal reverse base current</td>
</tr>
<tr>
<td>VLR</td>
<td>V</td>
<td>0.2</td>
<td>Cross-over voltage of the non-ideal reverse base current</td>
</tr>
<tr>
<td>XEXT</td>
<td>-</td>
<td>0.63</td>
<td>Part of Iex, Qex, Qtex, and Isub that depends on the base-collector voltage Vbc1</td>
</tr>
</tbody>
</table>

### Avalanche Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAVL</td>
<td>m</td>
<td>1.1e-6</td>
<td>Epilayer thickness used in weak-avalanche model</td>
</tr>
<tr>
<td>VAVL</td>
<td>V</td>
<td>3.0</td>
<td>Voltage determining the curvature of avalanche current</td>
</tr>
<tr>
<td>SFH</td>
<td>-</td>
<td>0.3</td>
<td>Current spreading factor of avalanche model (when EXAVL=1)</td>
</tr>
</tbody>
</table>

### Resistance and Epilayer Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RE</td>
<td>Ohm</td>
<td>5.0</td>
<td>Emitter resistance</td>
</tr>
<tr>
<td>RBC</td>
<td>Ohm</td>
<td>23.0</td>
<td>Constant part of the base resistance</td>
</tr>
</tbody>
</table>
### Resistance and Epilayer Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBV</td>
<td>Ohm</td>
<td>18.0</td>
<td>Zero-bias value of the variable part of the base resistance</td>
</tr>
<tr>
<td>RCC</td>
<td>Ohm</td>
<td>12.0</td>
<td>Constant part of the collector resistance</td>
</tr>
<tr>
<td>RCV</td>
<td>Ohm</td>
<td>150.0</td>
<td>Resistance of the un-modulated epilayer</td>
</tr>
<tr>
<td>SCRCV</td>
<td>Ohm</td>
<td>1250.0</td>
<td>Space charge resistance of the epilayer</td>
</tr>
<tr>
<td>IHC</td>
<td>A</td>
<td>4.0e-3</td>
<td>Critical current for velocity saturation in the epilayer</td>
</tr>
<tr>
<td>AXI</td>
<td>-</td>
<td>0.3</td>
<td>Smoothness parameter for the onset of quasi-saturation</td>
</tr>
</tbody>
</table>

### Base-Emitter Capacitances

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJE</td>
<td>F</td>
<td>7.3e-14</td>
<td>*Zero bias emitter-base depletion capacitance</td>
</tr>
<tr>
<td>VDE</td>
<td>V</td>
<td>0.95</td>
<td>Emitter-base diffusion voltage</td>
</tr>
<tr>
<td>PE</td>
<td>-</td>
<td>0.4</td>
<td>Emitter-base grading coefficient</td>
</tr>
<tr>
<td>XCJE</td>
<td>-</td>
<td>0.4</td>
<td>*Fraction of the emitter-base depletion capacitance that belongs to the sidewall</td>
</tr>
</tbody>
</table>

### Base-Collector Capacitances

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJC</td>
<td>F</td>
<td>7.8e-14</td>
<td>*Zero bias collector-base depletion capacitance</td>
</tr>
<tr>
<td>VDC</td>
<td>V</td>
<td>0.68</td>
<td>Collector-base diffusion voltage</td>
</tr>
</tbody>
</table>
### Base-Collector Capacitances

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>-</td>
<td>0.5</td>
<td>Collector-base grading coefficient</td>
</tr>
<tr>
<td>XP</td>
<td>-</td>
<td>0.35</td>
<td>Constant part of CJC</td>
</tr>
<tr>
<td>MC</td>
<td>-</td>
<td>0.5</td>
<td>Coefficient for the current modulation of the collector-base depletion capacitance</td>
</tr>
<tr>
<td>XCJC</td>
<td>-</td>
<td>3.2e-2</td>
<td>*Fraction of the collector-base depletion capacitance under the emitter</td>
</tr>
</tbody>
</table>

### Transit Time Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTAU</td>
<td>-</td>
<td>1.0</td>
<td>*Non-ideality of the emitter stored charge</td>
</tr>
<tr>
<td>TAUE</td>
<td>S</td>
<td>2.0e-12</td>
<td>*Minimum transit time of stored emitter charge</td>
</tr>
<tr>
<td>TAUB</td>
<td>S</td>
<td>4.2e-12</td>
<td>*Transit time of stored base charge</td>
</tr>
<tr>
<td>TEPI</td>
<td>S</td>
<td>4.1e-11</td>
<td>*Transit time of stored epilayer charge</td>
</tr>
<tr>
<td>TAUR</td>
<td>S</td>
<td>5.2e-10</td>
<td>*Transit time of reverse extrinsic stored base charge</td>
</tr>
<tr>
<td>DEG</td>
<td>EV</td>
<td>0.0</td>
<td>Bandgap difference over the base</td>
</tr>
<tr>
<td>XREC</td>
<td>-</td>
<td>0.0</td>
<td>Pre-factor of the recombination part of Ib1</td>
</tr>
</tbody>
</table>
## Temperature Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AQBO</td>
<td>-</td>
<td>0.3</td>
<td>Temperature coefficient of the zero-bias base charge</td>
</tr>
<tr>
<td>AE</td>
<td>-</td>
<td>0.0</td>
<td>Temperature coefficient of the resistivity of the emitter</td>
</tr>
<tr>
<td>AB</td>
<td>-</td>
<td>1.0</td>
<td>Temperature coefficient of the resistivity of the base</td>
</tr>
<tr>
<td>AEPI</td>
<td>-</td>
<td>2.5</td>
<td>Temperature coefficient of the resistivity of the epilayer</td>
</tr>
<tr>
<td>AEX</td>
<td>-</td>
<td>0.62</td>
<td>Temperature coefficient of the resistivity of the extrinsic base</td>
</tr>
<tr>
<td>AC</td>
<td>-</td>
<td>2.0</td>
<td>Temperature coefficient of the resistivity of the buried layer</td>
</tr>
<tr>
<td>DVGBF</td>
<td>V</td>
<td>5.0e-2</td>
<td>Bandgap voltage difference of forward current gain</td>
</tr>
<tr>
<td>CVGBR</td>
<td>V</td>
<td>4.5e-2</td>
<td>Bandgap voltage difference of reverse current gain</td>
</tr>
<tr>
<td>VGB</td>
<td>V</td>
<td>1.17</td>
<td>Bandgap voltage of the base</td>
</tr>
<tr>
<td>VGC</td>
<td>V</td>
<td>1.18</td>
<td>Bandgap voltage of the collector</td>
</tr>
<tr>
<td>VGJ</td>
<td>V</td>
<td>1.15</td>
<td>Bandgap voltage recombination emitter-base junction</td>
</tr>
<tr>
<td>DVGTE</td>
<td>V</td>
<td>0.05</td>
<td>*Bandgap voltage difference of emitter stored charge</td>
</tr>
</tbody>
</table>
### Noise Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>-</td>
<td>2.0</td>
<td>Exponent of the flicker-noise</td>
</tr>
<tr>
<td>KF</td>
<td>-</td>
<td>2.0e-11</td>
<td>Flicker-noise coefficient of the ideal base current</td>
</tr>
<tr>
<td>KFN</td>
<td>-</td>
<td>2.0e-11</td>
<td>Flicker-noise coefficient of the non-ideal base current</td>
</tr>
</tbody>
</table>

### Substrate Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISS</td>
<td>A</td>
<td>4.8e-17</td>
<td>Base-substrate saturation current</td>
</tr>
<tr>
<td>IKS</td>
<td>A</td>
<td>2.5e-4</td>
<td>Base-substrate high injection knee current</td>
</tr>
<tr>
<td>CJS</td>
<td>F</td>
<td>3.15e-13</td>
<td>*Zero bias collector-substrate depletion capacitance</td>
</tr>
<tr>
<td>VDS</td>
<td>V</td>
<td>0.62</td>
<td>*Collector-substrate diffusion voltage</td>
</tr>
<tr>
<td>PS</td>
<td>-</td>
<td>0.34</td>
<td>*Collector-substrate grading coefficient</td>
</tr>
<tr>
<td>VGS</td>
<td>V</td>
<td>1.2</td>
<td>Bandgap voltage of the substrate</td>
</tr>
<tr>
<td>AS</td>
<td>-</td>
<td>1.58</td>
<td>For a closed buried layer: AS=AC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For an open buried layer: AS=AEPI</td>
</tr>
</tbody>
</table>

### Self-Heating Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTH</td>
<td>^C/W</td>
<td>300.0</td>
<td>Thermal resistance</td>
</tr>
<tr>
<td>CTH</td>
<td>J/^C</td>
<td>3.0e-9</td>
<td>*Thermal capacitance</td>
</tr>
</tbody>
</table>
MEXTRAM Level 504 DC OP Analysis Example

**** DC OP analysis of mextram 504 model ****

.OPTIONs GMIN=1.0e-13

Q1 1 2 3 4 m504 area=1 m=1

* Start sources
VB  2 0 DC 1.2
VC  1 0 DC 2.2
VE  3 0 DC 0.0
VS  4 0 DC 0.0

.DC VB 0.4 1.2 0.1
.DC VC 1.4 2.2 0.1

.op
.PRINT DC I(VC)  I(VB)  I(VE)  I(VS)
.TEMP 25

.Model m504 npn level=6
     + VERS    =  504
     + TREF    =  25.0
     + EXMOD   =  1.0
     + EXPHI   =  1.0
     + EXAVL   =  0.0
     + IS      =  22.0E-18
     + IK      =  0.1
     + VER     =  2.5
     + VEF     =  44.0
     + BF      =  215.0
     + IBF     =  2.7E-15
     + MLF     =  2.0
     + XIBI    =  0.0
+ BRI  =  7.0
+ IBR  =  1.0e-15
+ VLR  =  0.2
+ XEXT =  0.63
+ WAVL =  1.1E-6
+ VAVL =  3.0
+ SFH  =  0.3
+ RE   =  5.0
+ RBC  =  23.
+ RBV  =  18.
+ RCC  =  12.
+ RCV  =  150.
+ SCRCV = 1250.0
+ IHC  =  4.e-3
+ AXI  =  0.3
+ CJE  =  73.0e-15
+ VDE  =  0.95
+ PE   =  0.4
+ XCJE =  0.4
+ CJC  =  78.0E-15
+ VDC  =  0.68
+ PC   =  0.5
+ XP   =  0.35
+ MC   =  0.5
+ XCJC =  32.E-3
+ MTAU =  1.0
+ TAUUE = 2.0E-12
+ TAUB =  4.2E-12
+ TEPI =  41.E-12
+ TAUR =  520.E-12
+ DEG  =  0.01
+ XREC =  0.1
+ AQBO =  0.3
MEXTRAM Level 504 Transient Analysis Example

*** Transient analysis of Mextram 504 model ***

.options gmin=1e-13 dccap POST=1

QCKT 1 2 0 0 m504 area=1.0 m=1

* START SOURCES

VC 3 0 DC 2
VB 2 0 DC 0 PULSE (0 0.8 0 1n 1n 10n 25n)
R 1 3 0.1
.temp 100
.TRAN 10p 50n
.op
.PRINT tran I(VC) I(VB)
.Model m504 npn level=6
  + VERS  =  504
  + TREF  =  25.0
  + EXMOD =  1.0
  + EXPHI =  1.0
  + EXAVL =  0.0
  + IS    =  22.0E-18
  + IK    =  0.1
  + VER   =  2.5
  + VEF   =  44.0
  + BF    =  215.0
  + IBF   =  2.7E-15
  + MLF   =  2.0
  + XIBI  =  0.0
  + BRI   =  7.0
  + IBR   =  1.0e-15
  + VLR   =  0.2
  + XEXT  =  0.63
  + WAVL  =  1.1E-6
  + VAVL  =  3.0
  + SFH   =  0.3
  + RE    =  5.0
  + RBC   =  23.
  + RBV   =  18.
+ RCC       =  12.
+ RCV       =  150.
+ SCRCV     =  1250.0
+ IHC       =  4.e-3
+ AXI       =  0.3
+ CJE       =  73.0e-15
+ VDE       =  0.95
+ PE        =  0.4
+ XCJE      =  0.4
+ CJC       =  78.0E-15
+ VDC       =  0.68
+ PC        =  0.5
+ XP        =  0.35
+ MC        =  0.5
+ XCJC      =  32.E-3
+ MTAU      =  1.0
+ TAUE      =  2.0E-12
+ TAUB      =  4.2E-12
+ TEPI      =  41.E-12
+ TAU R      =  520.E-12
+ DEG       =  0.01
+ XREC      =  0.1
+ AQBO      =  0.3
+ AE        =  0.0
+ AB        =  1.0
+ AEPI      =  2.5
+ AEX       =  0.62
+ AC        =  2.0
+ DVGBF     =  0.05
+ DVGBR     =  0.045
+ VGB       =  1.17
+ VGC       =  1.18
+ VGJ       =  1.15
Using BJT Models

LEVEL 6 Philips Bipolar Model (MEXTRAM LEVEL 504)

+ DVGTE = 0.05
+ AF = 2.0
+ KF = 2.0E-11
+ KFN = 2.0E-11
+ ISS = 48.0E-18
+ IKS = 250.0E-6
+ CJS = 315.0E-15
+ VDS = 0.62
+ PS = 0.34
+ VGS = 1.20
+ AS = 1.58

.END

MEXTRAM Level 504 AC Analysis Example

*** AC analysis of Mextram 504 model ***

.options gmin=1e-13 POST=1 converge=1 relv=1.e-6
         absv=1.e-9

QCKT 11 22 33 44 m504 area=1.0 m=1

* START SOURCES
VE 3 0 DC 0
VB 2 0 DC 0.7
VC 1 0 DC 1.0
VS 4 0 DC 0

.DC VB 0.7 1.0 0.02
.ac dec 1 1.e4 1.e11

vcl 11 1 0
vb1 22 2 0 ac=0.001
ve1 33 3 0
vs1 44 4 0

.op
.print ac ir(vcl) ii(vcl) ir(vbl) ii(vbl)

.temp 25

.Model m504 npn level=6
   + VERS   =  504
   + TREF   =  25.0
   + EXMOD  =  1.0
   + EXPHI  =  1.0
   + EXAVL  =  0.0
   + IS     =  22.0E-18
   + IK     =  0.1
   + VER    =  2.5
   + VEF    =  44.0
   + BF     =  215.0
   + IBF    =  2.7E-15
   + MLF    =  2.0
   + XIBI   =  0.0
   + BRI    =  7.0
   + IBR    =  1.0e-15
   + VLR    =  0.2
   + XEXT   =  0.63
   + WAVL   =  1.1E-6
   + VAVL   =  3.0
   + SFH    =  0.3
   + RE     =  5.0
   + RBC    =  23.
   + RBV    =  18.
   + RCC    =  12.
   + RCV    =  150.
+ SCRCV = 1250.0
+ IHC = 4.e-3
+ AXI = 0.3
+ CJE = 73.0e-15
+ VDE = 0.95
+ PE = 0.4
+ XCJE = 0.4
+ CJC = 78.0E-15
+ VDC = 0.68
+ PC = 0.5
+ XP = 0.35
+ MC = 0.5
+ XCJC = 32.E-3
+ MTAU = 1.0
+ TAUE = 2.0E-12
+ TAUB = 4.2E-12
+ TEPI = 41.E-12
+ Taur = 520.E-12
+ DEG = 0.01
+ XREC = 0.1
+ AQBO = 0.3
+ AE = 0.0
+ AB = 1.0
+ AEPI = 2.5
+ AEX = 0.62
+ AC = 2.0
+ DVGBF = 0.05
+ DVGBR = 0.045
+ VGB = 1.17
+ VGC = 1.18
+ VGJ = 1.15
+ DVGTE = 0.05
+ AF = 2.0
LEVEL 6 Philips Bipolar Model (MEXTRAM LEVEL 504)  Using BJT Models

+ KF    = 2.E-11
+ KFN   = 2.E-11
+ ISS   = 48.E-18
+ IKS   = 250.E-6
+ CJS   = 315.E-15
+ VDS   = 0.62
+ PS    = 0.34
+ VGS   = 1.20
+ AS    = 1.58

.END
LEVEL 8 HiCUM Model

What is the HiCUM Model?

HiCUM is an advanced transistor model for bipolar transistors, with a primary emphasis on circuit design for high-speed/high-frequency applications. HiCUM development was spurred by the SPICE Gummel-Poon model’s (SGPM) inadequate level of accuracy for high-speed large-signal transient applications and the required high-collector current densities. Other major disadvantages of the SGPM are:

- A lack of sufficient physical background
- Poor descriptions of base resistance and junction capacitances in the regions of interest
- Inadequate description of both Si- and III-V material-based HBTs

The HiCUM model is implemented with LEVEL 8 in Hspice.

HiCUM Model Advantages

Major features of HiCUM are:

- Accurate description of the high-current operating region (including quasi-saturation and saturation).
- Distributed modelling of external base-collector region.
- Proper handling of emitter periphery injection and charge storage.
- Internal base resistance as a function of operating point (conductivity modulation and emitter current crowding), and emitter geometry.
- Sufficiently physical model equations allowing predictions of temperature and process variations, as well as scalability, even at high current densities.
- Parasitic capacitances, independent on operating point, are available in the equivalent circuit, representing base-emitter and base-collector oxide overlaps, that become significant for small-size transistors.
- Weak avalanche breakdown is available.
Self-heating effects are included. Non-quasi-static effects, resulting in a delay of collector current AND stored minority charge, are modelled as function of bias.

Collector current spreading is included in minority charge and collector current formulation.

Extensions for graded-base SiGe HBTs have been derived using the Generalized Integral Charge-Control Relation (GICCR); the GICCR also permits modelling of HBTs with (graded) bandgap differences within the junctions.

Base-emitter tunneling model is available (e.g., for simulation of varactor leakage).

Simple parasitic substrate transistor is included in the equivalent circuit.

Simple parallel RC network taking into account the frequency dependent coupling between buried layer and substrate terminal.

Parameter extraction is closely related to the process enabling parametric yield simulation; parameter extraction procedure and list of test structures are available; HiCUM parameters can be determined using standard measurement equipment and mostly simple, decoupled extraction procedures.

Simple equivalent circuit and numerical formulation of model equations result in easy implementation and relatively fast execution time.

These features together with the choice of easily measurable basic variables such as junction capacitances and transit time provide - compared to the SGPM - high accuracy for digital circuit, small-signal high-frequency and, in particular, high-speed large-signal transient simulation. Also, HiCUM is laterally scaleable over a wide range of emitter widths and lengths up to high collector current densities; the scaling algorithm is generic and has been applied to the SGPM (within its validity limits).

In summary, HiCUM’s major advantages over other bipolar compact models are:

- Scalability
- Process-based and relatively simple parameter extraction
■ Predictive capability in terms of process and layout variations
■ Fairly simple numerical formulation facilitating easy implementation and resulting in still reasonable simulation time compared to the (too) simple SGPM at high current densities

Star-Hspice HiCUM Model vs. Public HiCUM Model

Difference Highlights
To maintain flexibility, the Star-Hspice LEVEL 8 HiCUM model uses FBCS, IS, KRBI, MCF, MSR, and ZETACX as additional model parameters. See “Other Parameters” on page 16-103.

Note: Self-heating is not supported in the 2000.4 Star-Hspice release.

Model Implementation

Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>9</td>
<td>HiCUM BJT level in Hspice</td>
</tr>
<tr>
<td>TREF</td>
<td>C</td>
<td>26.85</td>
<td>Temperature in simulation</td>
</tr>
</tbody>
</table>

Internal Transistors

Transfer Current Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C10</td>
<td>A^2s</td>
<td>3.76e-32</td>
<td>M^2</td>
<td>Constant(IS*QP0)</td>
</tr>
<tr>
<td>Qp0</td>
<td>As</td>
<td>2.78e-14</td>
<td></td>
<td>Zero-bias hole charge</td>
</tr>
<tr>
<td>ICH</td>
<td>A</td>
<td>2.09e-02</td>
<td></td>
<td>High-current correction for 2D/3D</td>
</tr>
</tbody>
</table>
**BE Depletion Capacitance Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDEI</td>
<td>V</td>
<td>0.95</td>
<td></td>
<td>Built-in voltage</td>
</tr>
<tr>
<td>CJEI0</td>
<td>F</td>
<td>8.11e-15</td>
<td></td>
<td>Zero-bias value</td>
</tr>
<tr>
<td>ZEI</td>
<td>-</td>
<td>0.5</td>
<td></td>
<td>Exponent coefficient</td>
</tr>
<tr>
<td>ALJEI</td>
<td>-</td>
<td>1.8</td>
<td></td>
<td>Ratio of max. to zero-bias value</td>
</tr>
</tbody>
</table>

**BC Depletion Capacitance Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJCI0</td>
<td>F</td>
<td>1.16e-15</td>
<td>M^2</td>
<td>Zero-bias value</td>
</tr>
<tr>
<td>VDCI</td>
<td>V</td>
<td>0.8</td>
<td></td>
<td>Built-in voltage</td>
</tr>
<tr>
<td>ZCI</td>
<td>-</td>
<td>0.333</td>
<td></td>
<td>Exponent coefficient</td>
</tr>
<tr>
<td>VPTCI</td>
<td>V</td>
<td>416</td>
<td></td>
<td>Punch-through voltage (≈q Nci w^2ci/(2epsilon))</td>
</tr>
</tbody>
</table>
### Forward Transit Time Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T0</td>
<td>s</td>
<td>4.75e-12</td>
<td></td>
<td>Low current transit time at $V_{B'C'}=0$</td>
</tr>
<tr>
<td>DT0H</td>
<td>s</td>
<td>2.1e-12</td>
<td></td>
<td>Time constant for base and BC SCR width modulation</td>
</tr>
<tr>
<td>TBVL</td>
<td>s</td>
<td>40e-12</td>
<td></td>
<td>Voltage for modeling carrier jam at low $V_{C'E}$</td>
</tr>
<tr>
<td>TEF0</td>
<td>s</td>
<td>1.8e-12</td>
<td></td>
<td>Storage time in neutral emitter</td>
</tr>
<tr>
<td>GTFE</td>
<td>-</td>
<td>1.4</td>
<td></td>
<td>Exponent factor for current dep. emitter transit time</td>
</tr>
<tr>
<td>THCS</td>
<td>s</td>
<td>3.0e-11</td>
<td></td>
<td>Saturation time constant at high current densities</td>
</tr>
<tr>
<td>ALHC</td>
<td>-</td>
<td>0.75</td>
<td></td>
<td>Smoothing factor for current dep. C and B transit time</td>
</tr>
<tr>
<td>FTHC</td>
<td>-</td>
<td>0.6</td>
<td></td>
<td>Partitioning factor for base and collection portion</td>
</tr>
<tr>
<td>ALQF</td>
<td>-</td>
<td>0.225</td>
<td></td>
<td>Factor for additional delay time of $Q_f$</td>
</tr>
</tbody>
</table>

### Critical Current Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCl0</td>
<td>Ohm</td>
<td>127.8</td>
<td>1/M</td>
<td>Low-field resistance of internal collector region</td>
</tr>
<tr>
<td>VLIM</td>
<td>V</td>
<td>0.7</td>
<td></td>
<td>Voltage separating ohmic and SCR regime</td>
</tr>
<tr>
<td>VPT</td>
<td>V</td>
<td>5.0</td>
<td></td>
<td>Epi punch-through vtg. of BC SCR</td>
</tr>
<tr>
<td>VCES</td>
<td>V</td>
<td>0.1</td>
<td></td>
<td>Internal CE sat. vtg.</td>
</tr>
</tbody>
</table>
### Inverse Transit Time Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TR</td>
<td>s</td>
<td>1.0e-9</td>
<td></td>
<td>Time constant for inverse operation</td>
</tr>
</tbody>
</table>

### Base Current Components Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBEIS</td>
<td>A</td>
<td>1.16e-20</td>
<td>M</td>
<td>BE saturation current</td>
</tr>
<tr>
<td>MBEI</td>
<td>-</td>
<td>1.015</td>
<td></td>
<td>BE saturation current</td>
</tr>
<tr>
<td>IREIS</td>
<td>A</td>
<td>1.16e-6</td>
<td>M</td>
<td>BE recombination saturation current</td>
</tr>
<tr>
<td>MREI</td>
<td>-</td>
<td>2.0</td>
<td></td>
<td>BE recombination non-ideality factor</td>
</tr>
<tr>
<td>IBCIS</td>
<td>A</td>
<td>1.16e-20</td>
<td>M</td>
<td>BC saturation current</td>
</tr>
<tr>
<td>MBCI</td>
<td>-</td>
<td>1.015</td>
<td></td>
<td>BC non-ideality factor</td>
</tr>
</tbody>
</table>

### Weak BC Avalanche Breakdown Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAVL</td>
<td>1/V</td>
<td>1.186</td>
<td></td>
<td>Prefactor for CB avalanche effect</td>
</tr>
<tr>
<td>QAVL</td>
<td>As</td>
<td>1.11e-14</td>
<td>M</td>
<td>Exponent factor for CB avalanche effect</td>
</tr>
</tbody>
</table>

### Internal Base Resistance Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBI0</td>
<td>Ohm</td>
<td>0</td>
<td>1/M</td>
<td>Value at zero-bias</td>
</tr>
<tr>
<td>FDQR0</td>
<td>-</td>
<td>0.0</td>
<td></td>
<td>Correction factor for modulation by BE and BC SCR</td>
</tr>
<tr>
<td>FGEO</td>
<td>-</td>
<td>0.73</td>
<td></td>
<td>Geometry factor (value corresponding to long emitter stripe)</td>
</tr>
</tbody>
</table>
### Lateral Scaling

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LATB</td>
<td></td>
<td>3.765</td>
<td></td>
<td>Scaling factor for Qfc in 1_E</td>
</tr>
<tr>
<td>LATL</td>
<td></td>
<td>0.342</td>
<td></td>
<td>Scaling factor for Qfc in I_E direction</td>
</tr>
</tbody>
</table>

### Peripheral Elements

#### BE Depletion Capacitance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJEP0</td>
<td>F</td>
<td>2.07e-15</td>
<td>M</td>
<td>Zero-bias value</td>
</tr>
<tr>
<td>VDEP</td>
<td>V</td>
<td>1.05</td>
<td></td>
<td>Built-in voltage</td>
</tr>
<tr>
<td>ZEP</td>
<td>-</td>
<td>0.4</td>
<td></td>
<td>Depletion coeff</td>
</tr>
<tr>
<td>ALJEP</td>
<td>-</td>
<td>2.4</td>
<td></td>
<td>Ratio of max. to zero-bias value</td>
</tr>
</tbody>
</table>

### Base Current

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBEPS</td>
<td>A</td>
<td>3.72e-21</td>
<td>M</td>
<td>Saturation current</td>
</tr>
<tr>
<td>MBEP</td>
<td>-</td>
<td>1.015</td>
<td></td>
<td>Non-ideality factor</td>
</tr>
<tr>
<td>IREPS</td>
<td>A</td>
<td>1e-30</td>
<td>M</td>
<td>Recombination saturation factor</td>
</tr>
<tr>
<td>MREP</td>
<td>-</td>
<td>2.0</td>
<td></td>
<td>Recombination non-ideality factor</td>
</tr>
</tbody>
</table>
### BE Tunneling

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBETS</td>
<td>A</td>
<td>0</td>
<td>M</td>
<td>Saturation current</td>
</tr>
<tr>
<td>ABET</td>
<td>-</td>
<td>0.0</td>
<td></td>
<td>Exponent coefficient</td>
</tr>
</tbody>
</table>

### External Elements

#### BC Capacitance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJCX0</td>
<td>F</td>
<td>5.393e-15</td>
<td>M</td>
<td>Zero-bias depletion value</td>
</tr>
<tr>
<td>VDCX</td>
<td>V</td>
<td>0.7</td>
<td></td>
<td>Built-in voltage</td>
</tr>
<tr>
<td>ZCX</td>
<td>-</td>
<td>0.333</td>
<td></td>
<td>Exponent coefficient</td>
</tr>
<tr>
<td>VPTCX</td>
<td>V</td>
<td>100</td>
<td></td>
<td>Punch-through voltage</td>
</tr>
<tr>
<td>CCOX</td>
<td>F</td>
<td>2.97e-15</td>
<td>M</td>
<td>Collector oxide capacitance</td>
</tr>
<tr>
<td>FBC</td>
<td>-</td>
<td>0.1526</td>
<td></td>
<td>Partitioning factor for C_BCX (=C'<em>{BCx}+C''</em>{BCx})</td>
</tr>
</tbody>
</table>

#### BC Base Current Component

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBCXS</td>
<td>A</td>
<td>4.39e-20</td>
<td>M</td>
<td>Saturation current</td>
</tr>
<tr>
<td>MBCX</td>
<td>-</td>
<td>1.03</td>
<td></td>
<td>Non-ideality factor</td>
</tr>
</tbody>
</table>

### Other External Elements
### Substrate Transistor Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITSS</td>
<td>A</td>
<td>0.0</td>
<td>M</td>
<td>Transfer saturation current</td>
</tr>
<tr>
<td>MSF</td>
<td>-</td>
<td>0.0</td>
<td></td>
<td>Non-ideality factor (forward transfer current)</td>
</tr>
<tr>
<td>TSF</td>
<td>-</td>
<td>0.0</td>
<td></td>
<td>Minority charge storage transit time</td>
</tr>
<tr>
<td>ISCS</td>
<td>A</td>
<td>0.0</td>
<td>M</td>
<td>Saturation current of CS diode</td>
</tr>
<tr>
<td>MSC</td>
<td>-</td>
<td>0.0</td>
<td></td>
<td>Non-ideality factor of CS diode</td>
</tr>
</tbody>
</table>

### Collector-Substrate Depletion Capacitance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJS0</td>
<td>F</td>
<td>3.64e-14</td>
<td>M</td>
<td>Zero-bias value of CS depletion cap</td>
</tr>
<tr>
<td>VDS</td>
<td>V</td>
<td>0.6</td>
<td></td>
<td>Built-in voltage</td>
</tr>
<tr>
<td>ZS</td>
<td>-</td>
<td>0.447</td>
<td></td>
<td>Exponent coefficient</td>
</tr>
<tr>
<td>VPTS</td>
<td>V</td>
<td>1000</td>
<td></td>
<td>Punch-through voltage</td>
</tr>
</tbody>
</table>

### Substrate Coupling Network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSU</td>
<td>Ohm</td>
<td>0</td>
<td>1/M</td>
<td>Substrate series resistance</td>
</tr>
<tr>
<td>CSU</td>
<td>F</td>
<td>0</td>
<td></td>
<td>Substrate capacitance from permittivity of bulk material</td>
</tr>
</tbody>
</table>
### Noise Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KF</td>
<td>-</td>
<td>1.43e-8</td>
<td></td>
<td>Flicker noise factor (no unit only for AF=2!)</td>
</tr>
<tr>
<td>AF</td>
<td>-</td>
<td>2.0</td>
<td></td>
<td>Flicker noise exponent factor</td>
</tr>
<tr>
<td>KRBI</td>
<td>-</td>
<td>1.17</td>
<td></td>
<td>Factor for internal base resistance</td>
</tr>
</tbody>
</table>

### Temperature Dependence Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGB</td>
<td>V</td>
<td>1.17</td>
<td></td>
<td>Bandgap-voltage</td>
</tr>
<tr>
<td>ALB</td>
<td>1/K</td>
<td>6.3e-3</td>
<td></td>
<td>Relative temperature coefficient of forward current gain</td>
</tr>
<tr>
<td>ALT0</td>
<td>1/K</td>
<td>0</td>
<td></td>
<td>First-order relative temperature coefficient of TEF0</td>
</tr>
<tr>
<td>KT0</td>
<td>1/K</td>
<td>0</td>
<td></td>
<td>Second-order relative temperature coefficient of TEF0</td>
</tr>
<tr>
<td>ZETACI</td>
<td>-</td>
<td>1.6</td>
<td></td>
<td>Temperature exponent factor RCI0</td>
</tr>
<tr>
<td>ALVS</td>
<td>1/K</td>
<td>1e-3</td>
<td></td>
<td>Relative temperature coefficient of saturation drift velocity</td>
</tr>
<tr>
<td>ALCES</td>
<td>1/K</td>
<td>0.4e-3</td>
<td></td>
<td>Relative temperature coefficient of VCES</td>
</tr>
<tr>
<td>ZETARBI</td>
<td>-</td>
<td>0.588</td>
<td></td>
<td>Temperature exponent factor of RBI0</td>
</tr>
<tr>
<td>ZETARBX</td>
<td>-</td>
<td>0.2060</td>
<td></td>
<td>Temperature exponent factor of RBX</td>
</tr>
<tr>
<td>ZETARCX</td>
<td>-</td>
<td>0.2230</td>
<td></td>
<td>Temperature exponent factor of RCX</td>
</tr>
<tr>
<td>ZETARE</td>
<td>-</td>
<td>0</td>
<td></td>
<td>Temperature exponent factor of RE</td>
</tr>
</tbody>
</table>
### Self-Heating Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTH</td>
<td>K/W</td>
<td>0</td>
<td>1/M</td>
<td>Thermal resistance (not supported)</td>
</tr>
<tr>
<td>CTH</td>
<td>Ws/K</td>
<td>0</td>
<td>M</td>
<td>Thermal resistance (not supported)</td>
</tr>
</tbody>
</table>

### Other Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBCS</td>
<td>-</td>
<td>-1.0</td>
<td></td>
<td>Determine external BC capacitance partitioning</td>
</tr>
<tr>
<td>IS</td>
<td>-1.0</td>
<td>A</td>
<td></td>
<td>Ideal saturation current</td>
</tr>
<tr>
<td>KRBI</td>
<td>-</td>
<td>1.0</td>
<td></td>
<td>Noise analysis of internal resistance</td>
</tr>
<tr>
<td>MCF</td>
<td>-</td>
<td>1.0</td>
<td></td>
<td>Non-ideality factor of reverse current between base and collector. VT=VT*MCF</td>
</tr>
<tr>
<td>MSR</td>
<td>-</td>
<td>1.0</td>
<td></td>
<td>Non-ideality factor of reverse current in substrate transistor. VT=VT*MSR</td>
</tr>
<tr>
<td>ZETACX</td>
<td>-</td>
<td>1.0</td>
<td></td>
<td>Temperature exponent factor (epi-layer)</td>
</tr>
</tbody>
</table>
Netlist Input and Output Formats

This section provides the syntax for LEVEL 8 and an example of an input netlist and output format.

Syntax

```
Qxxx nc nb ne <ns> mname <area> <M=val> <DTEMP=val>
```

- **Qxxx**: BJT element name
- **nc**: Collector terminal node
- **nb**: Base terminal node
- **ne**: Emitter terminal node
- **ns**: Substrate terminal node
- **mname**: BJT model name reference
- **area**: Emitter area multiplying factor which affects currents, resistances and capacitances (default=1)
- **M**: Multiplier to simulate multiple BJTs in parallel
- **DTEMP**: Difference between the element temperature and the circuit temperature in Celsius. (Default=0.0)

Example

This is an example of a BJT Q1 with collector, base and emitter and substrate connected to nodes 1, 2 and 3 and 4, where the BJT model is given by QM:

```
Q1 1 2 0 4 QM area=1*0.5*5 dtemp=0.002
```
**Circuit Diagram**

**Figure 16-15: Large-signal HiCUM/LEVEL2 equivalent circuit**

(a) The external BC capacitance consists of a depletion and a bias independent (e.g., oxide) capacitance with the ratio $C'_{BCX} / C''_{BCX}$ being adjusted with respect to proper modelling of the h.f. behavior.

(b) Thermal network used for self-heating calculation.

**Input Netlist**

```plaintext
.DATA test_data vbe vce vsub

0.0 0.0 0.0
0.1 0.0 0.0
0.2 0.0 0.0
0.3 0.0 0.0
0.4 0.0 0.0
0.5 0.0 0.0
0.6 0.0 0.0
```

LEVEL 8 HiCUM Model

0.7 0.0 0.0
0.8 0.0 0.0
0.9 0.0 0.0
1.0 0.0 0.0

.ENDDATA

.OPTIONS
.TEMP 26.85

VIN 2 0 vbe
VC 1 0 vce
VS 4 0 vsub
VE 3 0 0

Q1 1 2 3 4 hicum

.DC data= test_data

.PRINT DC I(VIN) i2(q1) I(VC) i1(q1) I(VCS) i4(q1)

.MODEL hicum NPN LEVEL=8
+ tref = 26.85
+ c10=.3760000E-31 qp0=.2780000E-13 ich=.2090000E-01
+ hfc=.1000000E+01
+ hfe=1.0000000E+00 hjei=.0000000E+00
+ hjci=.1000000E+01 tr=1.0000000E-9
+ cjei0=.81100E-14 vdei=.9500000E+00 zei=.5000000E+00
+ aljei=.180000E+01
+ cjci0=.116000E-14 vdci=.8000000E+00 zci=.3330000E+00
+ vptci=.416000E+03
+ rci0=.1278000E+03 vlim=.7000000E+00 vpt=.5000000E+01
+ vces=.1000000E+00
+ t0=.47500000E-11 dt0h=.210000E-11 tbvl=.4000000E-11
+ tef0=.1800000E-11 gtfe=.1400000E+01 thcs=.3000000E-10
+ alhc=.7500000E+00
+ fthc=.6000000E+00
+ latb=.3765000E+01 latl=.3420000E+00 fqi=.9055000E+00
+ alit=.4500000E+00 alqf=.2250000E+00
+ favl=.1186000E+01 gavl=.1110000E-13 alfav=.8250000E-04

Using BJT Models

LEVEL 8 HiCUM Model

+ alqav=.19600E-03
+ ibeis=.11600E-19 mbei=.101500E+01 ibeps=.10000E-29
+ mbep=.20000E+01
+ ireis=.11600E-15 mrei=.200000E+01 ireps=.10000E-29
+ mrep=.20000E+01
+ rbi0=.00000E+00 fdqr0=.00000E+00 fgeo=.730000E+00
+ fcrbi=.00000E+00
+ cjep0=.00000E+00 vdep=.105000E+01 zep=.4000000E+00
+ aljep=.240000E+01
+ ccex=.00000E+00
+ cjcx0=.00000E+00 vdcx=.700000E+00 zcx=.3330000E+00
+ vptcx=.10000E+03
+ ccox=.00000E+00 fbc=.1526000E+00
+ ibcxs=.10000E-29 mbcx=.200000E+01 ibcis=.11600E-19
+ mbci=.101500E+01
+ cjs0=.00000E+00 vds=.6000000E+00 zs=.44700000E+00
+ vpts=.10000E+04
+ rcx=.0000000E+00 rbx=.0000000E+00 re=.0000000E+00
+ kf=.0000000E+00 af=.0000000E+00
+ vgb=.1170000E+01 alb=.6300000E-02 alt0=.0000000E+00
+ kn=.0000000E+00
+ zetaci=.1600E+01 alvs=.100000E-02 alces=.40000E-03
+ zetarbi=0.5880E+00
+ zetarbx=0.2230E+00
+ zetare=.00000E+00
+ rth=0.0 cth=0.0
+ ibets=.00000E+00 abet=.0000000E+00
+ itss=.00000E+00 msf=.0000000E+00 tsf=0.0000000E+00
+ iscs=.00000E+00
+ msc=.0000000E+00
+ rsu=.0000000E+00 csu=.0000000E+00

.END
Chapter 17

Using JFET and MESFET Models

Star-Hspice contains three JFET/MESFET DC model levels. The same basic equations are used for both gallium arsenide MESFETs and silicon-based JFETs. This is possible because special materials definition parameters are included in these models. These models are also useful in modeling indium phosphide MESFETs.

This chapter covers the following topics:

- Understanding JFETs
- Specifying a Model
- Understanding the Capacitor Model
- Using JFET and MESFET Equivalent Circuits
- Using JFET and MESFET Model Statements
- JFET and MESFET Noise Models
- JFET and MESFET Temperature Equations
- Using TriQuint Model (TOM) Extensions to LEVEL=3
Understanding JFETs

JFETs are formed by diffusing a gate diode between the source and drain, while MESFETs are formed by applying a metal layer over the gate region, and creating a Schottky diode. Both technologies control the flow of carriers by modulating the gate diode depletion region. These field effect devices are called bulk semiconductor devices and are in the same category as bipolar transistors. Compared to surface effect devices such as MOSFETs, bulk semiconductor devices tend to have higher gain because bulk semiconductor mobility is always higher than surface mobility.

Enhanced characteristics of JFETs and MESFETs, relative to surface effect devices, include lower noise generation rates and higher immunity to radiation. These advantages have created the need for newer and more advanced models.

Features for JFET and MESFET modeling include:

- Charge-conserving gate capacitors
- Backgating substrate node
- Mobility degradation due to gate field
- Computationally efficient DC model (Curtice and Statz)
- Subthreshold equation
- Physically correct width and length (ACM)

The Star-Hspice GaAs model LEVEL=3 (See A MESFET Model for Use in the Design of GaAs Integrated Circuits, IEEE Transactions on Microwave Theory, Vol. MTT-28 No. 5) assumes that GaAs device velocity saturates at very low drain voltages. The Star-Hspice model has been further enhanced to include drain voltage induced threshold modulation and user-selectable materials constants. These features let you use the model for other materials such as silicon, indium phosphide, and gallium aluminum arsenide.

The Curtice model (See GaAs FET Device and Circuit Simulation in SPICE, IEEE Transactions on Electron Devices Volume ED-34) in Star-Hspice has been revised and the TriQuint model (TOM) is implemented as an extension of the earlier Statz model.
Specifying a Model

To specify a JFET or MESFET model in Star-Hspice, use a JFET element statement and a JFET model statement. The model parameter Level selects either the JFET or MESFET model. LEVEL=1 and LEVEL=2 select the JFET, and LEVEL=3 selects the MESFET. Different submodels for the MESFET LEVEL=3 equations are selected using the parameter SAT.

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>Model Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SPICE model</td>
</tr>
<tr>
<td>2</td>
<td>Modified SPICE model, gate modulation of LAMBDA</td>
</tr>
<tr>
<td>3</td>
<td>Hyperbolic tangent MESFET model (Curtice, Statz, Meta, TriQuint Models)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SAT</th>
<th>Model Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Curtice model (Default)</td>
</tr>
<tr>
<td>1</td>
<td>Curtice model with user defined VGST exponent</td>
</tr>
<tr>
<td>2</td>
<td>Cubic approximation of Curtice model with gate field degradation (Statz model)</td>
</tr>
<tr>
<td>3</td>
<td>Avant! variable saturation model</td>
</tr>
</tbody>
</table>

The model parameter CAPOP selects the type of capacitor model:

<table>
<thead>
<tr>
<th>CAPOP</th>
<th>Model Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SPICE depletion capacitor model</td>
</tr>
<tr>
<td>1</td>
<td>Charge conserving, symmetric capacitor model (Statz)</td>
</tr>
<tr>
<td>2</td>
<td>Avant! improvements to CAPOP=1</td>
</tr>
</tbody>
</table>

CAPOP=0, 1, 2 can be used for any model level. CAPOP=1 and 2 are most often used for the MESFET LEVEL 3 model.

The model parameter ACM selects the area calculation method:

<table>
<thead>
<tr>
<th>ACM</th>
<th>Model Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SPICE method (default)</td>
</tr>
<tr>
<td>1</td>
<td>Physically based method</td>
</tr>
</tbody>
</table>
Example

1. The following example selects the n channel MESFET model, LEVEL=3. It uses the SAT, ALPHA, and CAPOP=1 parameter:

   J1 7 2 3 GAASFET
   .MODEL GAASFET NJF LEVEL=3 CAPOP=1 SAT=1 VTO=-2.5
   + BETA=2.8E-3 LAMBDA=2.2M RS=70 RD=70 IS=1.7E-14
   + CGS=14P CGD=5P UCRIT=1.5 ALPHA=2

2. The following example selects an n-channel JFET:

   J2 7 1 4 JM1
   .MODEL JM1 NJF (VTO=-1.5, BETA=5E-3, CGS=5P, CGD=1P,
   + CAPOP=1 ALPHA=2)

3. The following example selects a p-channel JFET:

   J3 8 3 5 JX
   .MODEL JX PJF (VTO=-1.2, BETA=.179M, LAMBDA=2.2M
   + CGS=100P CGD=20P CAPOP=1 ALPHA=2)
Understanding the Capacitor Model

The SPICE depletion capacitor model (CAPOP=0) uses a diode-like capacitance between source and gate, where the depletion region thickness (and therefore the capacitance) is determined by the gate-to-source voltage. A similar diode model is often used to describe the normally much smaller gate-to-drain capacitance.

These approximations have serious shortcomings such as:

1. **Zero source-to-drain voltage:** The symmetry of the FET physics gives the conclusion that the gate-to-source and gate-to-drain capacitances should be equal, but in fact they can be very different.

2. **Inverse-biased transistor:** Where the drain acts like the source and the source acts like the drain. According to the model, the large capacitance should be between the original source and gate; but in this circumstance, the large capacitance is between the original drain and gate.

When low source-to-drain voltages inverse biased transistors are involved, large errors can be introduced into simulations. To overcome these limitations, use the Statz charge-conserving model by selecting model parameter CAPOP=1. The model selected by CAPOP=2 contains further improvements.

Model Applications

MESFETs are used to model GaAs transistors for high speed applications. Using MESFET models, transimpedance amplifiers for fiber optic transmitters up to 50 GHz can be designed and simulated.
Control Options

Control options that affect the simulation and design of both JFETs and MESFETs include:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCAP</td>
<td>Capacitance equation selector</td>
</tr>
<tr>
<td>GMIN, GRAMP, GMINDC</td>
<td>Conductance options</td>
</tr>
<tr>
<td>SCALM</td>
<td>Model scaling option</td>
</tr>
<tr>
<td>DCCAP</td>
<td>Invokes capacitance calculation in DC analysis</td>
</tr>
</tbody>
</table>

Override a global depletion capacitance equation selection that uses the .OPTION DCAP=<val> statement in a JFET or MESFET model by including DCAP=<val> in the device’s .MODEL statement.

Convergence

Enhance convergence for JFET and MESFET by using the GEAR method of computation (.OPTIONS METHOD=GEAR), when you include the transit time model parameter. Use the options GMIN, GMINDC, and GRAMP to increase the parasitic conductance value in parallel with pn junctions of the device.

Capacitor Equations

The option DCAP selects the equation used to calculate the gate-to-source and gate-to-drain capacitance for CAPOP=0. DCAP can be set to 1, 2 or 3. The default is 2.
Using JFET and MESFET Equivalent Circuits

Scaling

The AREA and M Element parameters, together with the SCALE and SCALM control options, control scaling. For all three model levels, the model parameters IS, CGD, CGS, RD, RS, BETA, LDEL, and WDEL, are scaled using the same equations.

Scaled parameters A, L, W, LDEL, and WDEL, are affected by option SCALM. SCALM defaults to 1.0. To enter the parameter W with units in microns, for example, set SCALM to 1e-6, then enter W=5; Star-Hspice sets W=5e-6 meters, or 5 microns.

Override global scaling that uses the .OPTION SCALM=<val> statement in a JFET or MESFET model by including SCALM=<val> in the .MODEL statement.

Understanding JFET Current Convention

The direction of current flow through the JFET is assumed in Figure 17-1. You can use either I(Jxxx) or I1(Jxxx) syntax when printing the drain current. I2 references the gate current and I3 references the source current. Jxxx is the device name. Figure 17-1 represents the Star-Hspice current convention for an n channel JFET.
For a p-channel device, the following must be reversed:
- Polarity of the terminal voltages vgd, vgs, and vds
- Direction of the two gate junctions
- Direction of the nonlinear current source id

**JFET Equivalent Circuits**

Star-Hspice uses three equivalent circuits in the analysis of JFETs: transient, AC, and noise circuits. The components of these circuits form the basis for all element and model equation discussion.

The fundamental component in the equivalent circuit is the drain to source current (ids). For noise and AC analyses, the actual ids current is not used. Instead, the partial derivatives of ids with respect to the terminal voltages, vgs, and vds are used.

The names for these partial derivatives are:

**Transconductance**

\[
gm = \left. \frac{\partial (ids)}{\partial (vgs)} \right|_{vds = const.}
\]
Output Conductance

\[ g_{ds} = \frac{\partial (i_{ds})}{\partial (v_{ds})} \bigg|_{v_{gs} = \text{const.}}. \]

The \( i_{ds} \) equation accounts for all DC currents of the JFET. The gate capacitances are assumed to account for transient currents of the JFET equations. The two diodes shown in Figure 17-2 are modeled by these ideal diode equations:

\[
\begin{align*}
  i_{gd} &= I_{S\text{eff}} \cdot \left( e^{\frac{v_{gd}}{N \cdot v_t}} - 1 \right) & v_{gd} > -10 \cdot N \cdot v_t \\
  i_{gd} &= -I_{S\text{eff}} & v_{gd} \leq -10 \cdot N \cdot v_t \\
  i_{gs} &= I_{S\text{eff}} \cdot \left( e^{\frac{v_{gs}}{N \cdot v_t}} - 1 \right) & v_{gs} > -10 \cdot N \cdot v_t \\
  i_{gs} &= -I_{S\text{eff}} & v_{gs} \leq -10 \cdot N \cdot v_t
\end{align*}
\]

**Figure 17-2: JFET/MESFET Transient Analysis**

---

**Note:** For DC analysis, the capacitances are not part of the model.
Figure 17-3: JFET/MESFET AC Analysis

Figure 17-4: JFET/MESFET AC Noise Analysis
### Table 17-2: Equation Variable Names and Constants

<table>
<thead>
<tr>
<th>Variable/Quantity</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgd</td>
<td>Gate to drain capacitance</td>
</tr>
<tr>
<td>cgs</td>
<td>Gate to source capacitance</td>
</tr>
<tr>
<td>ggd</td>
<td>Gate to drain AC conductance</td>
</tr>
<tr>
<td>ggs</td>
<td>Gate to source AC conductance</td>
</tr>
<tr>
<td>gds</td>
<td>Drain to source AC conductance controlled by vds</td>
</tr>
<tr>
<td>gm</td>
<td>Drain to source AC transconductance controlled by vgs</td>
</tr>
<tr>
<td>igd</td>
<td>Gate to drain current</td>
</tr>
<tr>
<td>igs</td>
<td>Gate to source current</td>
</tr>
<tr>
<td>ids</td>
<td>DC drain to source current</td>
</tr>
<tr>
<td>ind</td>
<td>Equivalent noise current drain to source</td>
</tr>
<tr>
<td>inrd</td>
<td>Equivalent noise current drain resistor</td>
</tr>
<tr>
<td>inrs</td>
<td>Equivalent noise current source resistor</td>
</tr>
<tr>
<td>rd</td>
<td>Drain resistance</td>
</tr>
<tr>
<td>rs</td>
<td>Source resistance</td>
</tr>
<tr>
<td>vgd</td>
<td>Internal gate-drain voltage</td>
</tr>
<tr>
<td>vgs</td>
<td>Internal gate-source voltage</td>
</tr>
<tr>
<td>f</td>
<td>Frequency</td>
</tr>
<tr>
<td>$\varepsilon_0$</td>
<td>Vacuum permittivity = 8.854e-12 F/m</td>
</tr>
<tr>
<td>k</td>
<td>1.38062e-23 (Boltzmann's constant)</td>
</tr>
<tr>
<td>q</td>
<td>1.60212e-19 (electron charge)</td>
</tr>
<tr>
<td>t</td>
<td>Temperature in °K</td>
</tr>
<tr>
<td>Dt</td>
<td>$t - t_{nom}$</td>
</tr>
</tbody>
</table>

vacuum permittivity = $8.854 \times 10^{-12}$ F/m
### Table 17-2: Equation Variable Names and Constants

<table>
<thead>
<tr>
<th>Variable/Quantity</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tnom</td>
<td>Nominal temperature of parameter measurements in °K (user-input in °C). Tnom = 273.15 + TNOM</td>
</tr>
<tr>
<td>vt(t)</td>
<td>k · t/q</td>
</tr>
<tr>
<td>vt(tnom)</td>
<td>k · tnom/q</td>
</tr>
</tbody>
</table>
Using JFET and MESFET Model Statements

Syntax

```
.MODEL mname NJF <LEVEL = val> <pname1 = val1> ...  
.MODEL mname PJF <LEVEL = val> <pname1 = val1> ...  
```

- **mname**: Model name. Elements refer to the model by this name.
- **NJF**: Identifies an N-channel JFET or MESFET model
- **LEVEL**: The LEVEL parameter selects different DC model equations.
- **pname1=val1**: Each JFET or MESFET model can include several model parameters.
- **PJF**: Identifies a P-channel JFET or MESFET model

JFET and MESFET Model Parameters

DC characteristics are defined by the model parameters VTO and BETA. These parameters determine the variation of drain current with gate voltage. LAMBDA determines the output conductance, and IS, the saturation current, of the two gate junctions. Two ohmic resistances, RD and RS, are included. The charge storage is modeled by nonlinear depletion-layer capacitances for both gate junctions that vary as the -M power of junction voltage, and are defined by the parameters CGS, CGD, and PB.

Use parameters KF and AF to model noise, which is also a function of the series source and drain resistances (RS and RD), in addition to temperature. Use the parameters ALPHA and A to model MESFETs.
The AREA model parameter is common to both the element and model parameters. The AREA element parameter always overrides the AREA model parameter.

### Table 17-3: JFET and MESFET Model Parameters

<table>
<thead>
<tr>
<th>Model Parameters Common to All Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometric</td>
</tr>
<tr>
<td>ACM, ALIGN, AREA, HDIF, L, LDEL, LDIF, RD, RG, RS, RSH, RSHG, RSHL, W, WDEL</td>
</tr>
<tr>
<td>Capacitance</td>
</tr>
<tr>
<td>CAPOP, CGD, CGS, FC, M, PB, TT</td>
</tr>
<tr>
<td>Subthreshold</td>
</tr>
<tr>
<td>ND, NG</td>
</tr>
<tr>
<td>Noise</td>
</tr>
<tr>
<td>AF, KF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LEVEL=1 Model Parameters (JFET)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
</tr>
<tr>
<td>BETA, IS, LAMBDA, N, VTO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LEVEL=2 Model Parameters (JFET)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
</tr>
<tr>
<td>BETA, IS, LAMBDA, LAM1, N, VTO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LEVEL=3 Model Parameters (MESFET)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
</tr>
<tr>
<td>ALPHA, BETA, D, GAMDS, IS, N, K1, LAMBDA, NCHAN, SAT, SATEXP, UCRIT, VBI, VGEXP, VP, VTO</td>
</tr>
</tbody>
</table>

The following subsections provide information about:
- Gate Diode DC Parameters
- DC Model LEVEL 1 Parameters
- DC Model LEVEL 2 Parameters
- DC Model LEVEL 3 Parameters
- ACM (Area Calculation Method) Parameter Equations
### Gate Diode DC Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td></td>
<td></td>
<td>Area calculation method. This parameter allows the selection between the old SPICE unitless gate area calculations and the new Star-Hspice area calculations (see the ACM section). If W and L are specified, AREA becomes: ACM=0 AREA=Weff/Leff ACM=1 AREA=Weff \cdot Leff</td>
</tr>
<tr>
<td>ALIGN</td>
<td>m</td>
<td>0</td>
<td>Misalignment of gate</td>
</tr>
<tr>
<td>AREA</td>
<td></td>
<td></td>
<td>Default area multiplier. This parameter affects the BETA, RD, RS, IS, CGS, and CGD model parameters. AREAeff=M \cdot AREA Override this parameter using the element effective area.</td>
</tr>
<tr>
<td>HDIF</td>
<td>m</td>
<td>0</td>
<td>Distance of the heavily diffused or low resistance region from source or drain contact to lightly doped region</td>
</tr>
<tr>
<td>IS</td>
<td>amp</td>
<td>1.0e-14</td>
<td>Gate junction saturation current ISeff = IS \cdot AREAeff</td>
</tr>
<tr>
<td>L</td>
<td>m</td>
<td>0.0</td>
<td>Default length of FET. Override this parameter using the element L. Leff = L \cdot SCALM + LDELeff</td>
</tr>
<tr>
<td>LDEL</td>
<td>m</td>
<td>0.0</td>
<td>Difference between drawn and actual or optical device length LDELeff = LDEL \cdot SCALM</td>
</tr>
<tr>
<td>LDIF</td>
<td>m</td>
<td>0</td>
<td>Distance of the lightly doped region from heavily doped region to transistor edge</td>
</tr>
</tbody>
</table>
### Using JFET and MESFET Model Statements

#### Name (Alias) | Units | Default | Description
--- | --- | --- | ---
N |  | 1.0 | Emission coefficient for gate-drain and gate-source diodes

#### RD | ohm | 0.0 | Drain ohmic resistance (see the ACM section) \( R_{D_{eff}} = \frac{R_D}{A_{R_{eff}}}, ACM=0 \)

#### RG | ohm | 0.0 | Gate resistance (see the ACM section) \( R_{G_{eff}} = R_G \cdot A_{R_{eff}}, ACM=0 \)

#### RS | ohm | 0.0 | Source ohmic resistance (see the ACM section) \( R_{S_{eff}} = R_S \cdot A_{R_{eff}}, ACM=0 \)

#### RSH | ohm/sq | 0 | Heavily doped region, sheet resistance

#### RSHG | ohm/sq | 0 | Gate sheet resistance

#### RSHL | ohm/sq | 0 | Lightly doped region, sheet resistance

#### W | m | 0.0 | Default width of FET. Override this parameter using the element W. \( W_{eff} = W \cdot SCALM + WDELeff \)

#### WDEL | m | 0.0 | Difference between drawn and actual or optical device width \( WDELeff = WDEL \cdot SCALM \)
## Gate Capacitance LEVEL 1, 2, and 3 Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPOP</td>
<td></td>
<td>0.0</td>
<td>Capacitor model selector:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CAPOP=0 – default capacitance equation based on diode depletion layer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CAPOP=1 – symmetric capacitance equations (Statz)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CAPOP=2 – Avant! improvement to CAPOP=1</td>
</tr>
<tr>
<td>CALPHA</td>
<td>ALPHA</td>
<td></td>
<td>Saturation factor for capacitance model (CAPOP=2 only)</td>
</tr>
<tr>
<td>CAPDS</td>
<td>F</td>
<td>0</td>
<td>Drain to source capacitance for TriQuint model</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CAPDS\text{eff} = CAPDS \cdot \frac{W_{eff}}{L_{eff}} M</td>
</tr>
<tr>
<td>CGAMDS</td>
<td>GAMDS</td>
<td></td>
<td>Threshold lowering factor for capacitance (CAPOP=2 only)</td>
</tr>
<tr>
<td>CGD</td>
<td>F</td>
<td>0.0</td>
<td>Zero-bias gate-drain junction capacitance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CGD\text{eff} = CGD \cdot \text{AREAeff}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Override this parameter by specifying GCAP.</td>
</tr>
<tr>
<td>CGS</td>
<td>F</td>
<td>0.0</td>
<td>Zero-bias gate-source junction capacitance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CGS\text{eff} = CGS \cdot \text{AREAeff}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Override this parameter by specifying GCAP.</td>
</tr>
<tr>
<td>CRAT</td>
<td></td>
<td>0.666</td>
<td>Source fraction of gate capacitance (used with GCAP)</td>
</tr>
<tr>
<td>GCAP</td>
<td>F</td>
<td></td>
<td>Zero-bias gate capacitance. If specified,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CGS\text{eff} = GCAP \cdot CRAT \cdot \text{AREAeff} and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CGD\text{eff} = GCAP \cdot (1-CRAT) \cdot \text{AREAeff}</td>
</tr>
</tbody>
</table>
### Name (Alias) | Units | Default | Description
--- | --- | --- | ---
FC |  | 0.5 | Coefficient for forward-bias depletion capacitance formulas (CAPOP=0 and 2 only)
CVTO | VTO |  | Threshold voltage for capacitance model (CAPOP=2 only)
M (MJ) |  | 0.5 | Grading coefficient for gate-drain and gate-source diodes (CAPOP=0 and 2 only)
|  |  | 0.50 - step junction 0.33 - linear graded junction
PB | V | 0.8 | Gate junction potential
TT | s | 0 | Transit time – option METHOD=GEAR is recommended when using transit time for JFET and MESFET

**Note:** Many DC parameters (such as VTO, GAMDS, ALPHA) can also affect capacitance.
## DC Model LEVEL 1 Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>LEVEL=1 invokes SPICE JFET model</td>
</tr>
</tbody>
</table>
| BETA         | amp/\sqrt{V^2} | 1.0e-4 | Transconductance parameter, gain  
  \[ BETA_{eff} = BETA \cdot \frac{W_{eff} \cdot M}{L_{eff}} \] |
| LAMBDA       | 1/V   | 0.0     | Channel length modulation parameter |
| ND           | 1/V   | 0.0     | Drain subthreshold factor (typical value=1) |
| NG           | 0.0   | Gate subthreshold factor (typical value=1) |
| VTO          | V     | -2.0    | Threshold voltage. If set, it overrides internal calculation. A negative VTO is a depletion transistor regardless of NJF or PJF. A positive VTO is always an enhancement transistor. |

## DC Model LEVEL 2 Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>Level of FET DC model. LEVEL=2 is based on modifications to the SPICE model for gate modulation of LAMBDA.</td>
</tr>
</tbody>
</table>
| BETA         | amp/\sqrt{V^2} | 1.0e-4 | Transconductance parameter, gain  
  \[ BETA_{eff} = BETA \cdot \frac{W_{eff} \cdot M}{L_{eff}} \] |
<p>| LAMBDA       | 1/V   | 0.0     | Channel length modulation parameter |
| LAM1         | 1/V   | 0.0     | Channel length modulation gate voltage parameter |
| ND           | 1/V   | 0.0     | Drain subthreshold factor (typical value=1) |
| NG           | 0.0   | Gate subthreshold factor (typical value=1) |</p>
<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTO</td>
<td>V</td>
<td>-2.0</td>
<td>Threshold voltage. When set, VTO overrides internal calculation. A negative VTO is a depletion transistor regardless of NJF or PJF. A positive VTO is always an enhancement transistor.</td>
</tr>
</tbody>
</table>
## DC Model LEVEL 3 Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>Level of FET DC model. LEVEL=3 is the Curtice MESFET model.</td>
</tr>
<tr>
<td>A</td>
<td>m</td>
<td>0.5µ</td>
<td>Active layer thickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$A_{eff} = A \cdot SCALM$</td>
</tr>
<tr>
<td>ALPHA</td>
<td>1/V</td>
<td>2.0</td>
<td>Saturation factor</td>
</tr>
<tr>
<td>BETA</td>
<td>amp/V(^2)</td>
<td>1.0e-4</td>
<td>Transconductance parameter, gain $BETA_{eff} = BETA \cdot \frac{W_{eff} \cdot M}{L_{eff}}$</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>11.7</td>
<td>Semiconductor dielectric constant: Si=11.7, GaAs=10.9</td>
</tr>
<tr>
<td>DELTA</td>
<td></td>
<td>0</td>
<td>I_{ds} feedback parameter of TriQuint model</td>
</tr>
<tr>
<td>GAMDS (GAMMA)</td>
<td></td>
<td>0</td>
<td>Drain voltage, induced threshold voltage lowering coefficient</td>
</tr>
<tr>
<td>LAMBDA</td>
<td>1/V</td>
<td>0.0</td>
<td>Channel length modulation parameter</td>
</tr>
<tr>
<td>K1</td>
<td>V(^{1/2})</td>
<td>0.0</td>
<td>Threshold voltage sensitivity to bulk node</td>
</tr>
<tr>
<td>NCHAN</td>
<td>atom/cm(^3)</td>
<td>1.552e16</td>
<td>Effective dopant concentration in the channel</td>
</tr>
<tr>
<td>ND</td>
<td>1/V</td>
<td>0.0</td>
<td>Drain subthreshold factor</td>
</tr>
<tr>
<td>NG</td>
<td></td>
<td>0.0</td>
<td>Gate subthreshold factor (typical value=1)</td>
</tr>
</tbody>
</table>
ACM (Area Calculation Method) Parameter Equations

The JFET model parameter ACM lets you select between the SPICE unitless gate area calculations and the Star-Hspice area calculations. The ACM=0 method (SPICE) uses the ratio of W/L to keep AREA unitless. The ACM=1 model (Star-Hspice) requires parameters such as IS, CGS, CGD, and BETA to have proper physics-based units.

In the following equations, lower case “m” indicates the element multiplier.

ACM=0

\[
AREA_{eff} = \frac{W_{eff}}{L_{eff}} \cdot m
\]
Using JFET and MESFET Models

\[ R_{D\text{eff}} = \frac{R_D}{A_{R\text{Eeff}}} \]

\[ R_{S\text{eff}} = \frac{R_S}{A_{R\text{Eeff}}} \]

\[ R_{G\text{eff}} = R_G \cdot \frac{A_{R\text{Eeff}}}{m^2} \]

**ACM=1**

\[ A_{R\text{Eeff}} = W_{\text{eff}} \cdot L_{\text{eff}} \cdot m \]

\[ R_{D\text{eff}} = \frac{R_D}{m} \]

Or if RD=0,

\[ R_{D\text{eff}} = R_{SH} \cdot \frac{H_{DIF}}{W_{\text{eff}} \cdot m} + R_{SHL} \cdot \frac{L_{DIF} + \text{ALIGN}}{W_{\text{eff}} \cdot m} \]

\[ R_{G\text{eff}} = \frac{R_G}{m} \]

or if RG=0,

\[ R_{G\text{eff}} = R_{SHG} \cdot \frac{W_{\text{eff}}}{L_{\text{eff}} \cdot m} \]

\[ R_{S\text{eff}} = \frac{R_S}{m} \]

or if RS=0,

\[ R_{S\text{eff}} = R_{SH} \cdot \frac{H_{DIF}}{W_{\text{eff}} \cdot m} + R_{SHL} \cdot \frac{L_{DIF} - \text{ALIGN}}{W_{\text{eff}} \cdot m} \]

**Resulting calculations**

\[ I_{S\text{eff}} = I_S \cdot A_{R\text{Eeff}} \]
Using JFET and MESFET Model Statements

\[ CGS_{\text{eff}} = CGS \cdot \text{AREA}_{\text{eff}} \]
\[ CGD_{\text{eff}} = CGD \cdot \text{AREA}_{\text{eff}} \]
\[ \text{BETA}_{\text{eff}} = \text{BETA} \cdot \frac{W_{\text{eff}}}{L_{\text{eff}}} \cdot m \]

**Note:** The model parameter units for IS, CGS, CGD, are unitless in ACM=0 and per square meter for ACM=1.

**Example**

```
j1 10 20 0 40 nj_acm0  w=10u l=1u
j2a 10 20 0 41 nj_acm1  w=10u l=1u

.model nj_acm0 njf LEVEL=3 capop=1 sat=3 acm=0
+ is=1e-14  cgs=1e-15 cgd=.3e-15
$$\text{note different units for is, cgs, cgd}$$
+ rs=100  rd=100  rg=5  beta=5e-4
+ vto=.3  n=1  ng=1.4  nd=1
+ k1=.2  vgexp=2  alpha=4  ucrit=1e-4  lambda=.1
+ satexp=2
+ eg=1.5  gap1=5e-4  gap2=200  d=13

.model nj_acm1 njf LEVEL=3 capop=1 sat=3 acm=1
+ is=1e-2  cgs=1e-3  cgd=.3e-3
$$\text{note different units for is, cgs, cgd}$$
+ rs=100  rd=100  rg=5  beta=5e-4
+ vto=.3  n=1  ng=1.4  nd=1
+ k1=.2  vgexp=2  alpha=4  ucrit=1e-4  lambda=.1
+ satexp=2
+ eg=1.5  gap1=5e-4  gap2=200  d=13
```
JFET and MESFET Capacitances

Gate Capacitance CAPOP=0

The DCAP option switch selects the diode forward bias capacitance equation:

DCAP=1

Reverse Bias:

\[ v_{gd} < FC \cdot PB \]

\[ c_{gd} = CGDeff \cdot \left( 1 - \frac{v_{gd}}{PB} \right)^{-M} \]

\[ \text{vgs} < FC \cdot PB \]

\[ c_{gs} = CGSeff \cdot \left( 1 - \frac{v_{gs}}{PB} \right)^{-M} \]

Forward Bias:

\[ v_{gd} \quad FC \cdot PB \]

\[ c_{gd} = TT \cdot \frac{\partial v_{gd}}{\partial v_{gd}} + CGDeff \cdot \frac{1 - FC \cdot (1 + M) + M \cdot v_{gd}}{(1 - FC)^{M+1} \cdot PB} \]

\[ v_{gs} \quad FC \cdot PB \]

\[ c_{gs} = TT \cdot \frac{\partial v_{gs}}{\partial v_{gs}} + CGSeff \cdot \frac{1 - FC \cdot (1 + M) + M \cdot v_{gs}}{(1 - FC)^{M+1} \cdot PB} \]
Using JFET and MESFET Model Statements

DCAP=2 (Star-Hspice Default)
Reverse Bias:

\[
\text{vgd} < 0
\]

\[
c_{gd} = CGDeff \cdot \left(1 - \frac{v_{gd}}{PB}\right)^{-M}
\]

\[
\text{vgs} < 0
\]

\[
c_{gs} = CGSeff \cdot \left(1 - \frac{v_{gs}}{PB}\right)^{-M}
\]

Forward Bias:

\[
\text{vgd} \quad 0
\]

\[
c_{gd} = TT \cdot \frac{\partial i_{gd}}{\partial v_{gd}} + CGDeff \cdot \left(1 + M \cdot \frac{v_{gd}}{PB}\right)
\]

\[
\text{vgs} \quad 0
\]

\[
c_{gs} = TT \cdot \frac{\partial i_{gs}}{\partial v_{gs}} + CGSeff \cdot \left(1 + M \cdot \frac{v_{gs}}{PB}\right)
\]

DCAP=3
Limits peak depletion capacitance to FC \cdot CGDeff or FC \cdot CGSeff, with proper fall-off when forward bias exceeds PB (FC \geq 1).
Gate Capacitance CAPOP=1

Gate capacitance CAPOP=1 is a charge conserving symmetric capacitor model most often used for MESFET model LEVEL 3.

\[
C_{gs} = \frac{CGS}{4 \sqrt{1 - \frac{v_{new}}{PB}}} \left[ 1 + \frac{veff - vte}{\sqrt{(veff - vte)^2 + (0.2)^2}} \right].
\]

\[
\left[ 1 + \frac{v_{ds}}{\sqrt{v_{ds}^2 + \left( \frac{1}{\text{ALPHA}} \right)^2}} \right] + \left[ \frac{CGD}{2} \cdot \left( 1 - \frac{v_{ds}}{\sqrt{v_{ds}^2 + \left( \frac{1}{\text{ALPHA}} \right)^2}} \right) \right]
\]

\[
C_{gd} = \left\{ \frac{CGS}{4 \sqrt{1 - \frac{v_{new}}{PB}}} \left[ 1 + \frac{veff - vte}{\sqrt{(veff - vte)^2 + (0.2)^2}} \right] \cdot \left[ 1 - \frac{v_{ds}}{\sqrt{v_{ds}^2 + \left( \frac{1}{\text{ALPHA}} \right)^2}} \right] \right\} + \left\{ \frac{CGD}{2} \cdot \left[ 1 + \frac{v_{ds}}{\sqrt{v_{ds}^2 + \left( \frac{1}{\text{ALPHA}} \right)^2}} \right] \right\}
\]

where:

\[
vte = VTO + GAMDS \cdot v_{ds} + K1(v_{bs}) = \text{effective threshold}
\]

\[
veff = \frac{1}{2} \left[ v_{gs} + v_{gd} + \sqrt{v_{ds}^2 + \left( \frac{1}{\text{ALPHA}} \right)^2} \right]
\]
and:

\[ v_{new} = \frac{1}{2} \left[ v_{eff} + v_{te} + \sqrt{(v_{eff} - v_{te})^2 + (0.2)^2} \right] \]

\[
CGD = \text{High } -v_{ds} \text{ Cgd at } v_{gs} = 0 \\
CGS = \text{High } -v_{ds} \text{ Cgs at } v_{gs} = 0 \\
CGD - CGDeff \\
CGS - CGSeff
\]

**Gate Capacitance CAPOP=2**

The Statz capacitance equations (See *H. Statz, P.Newman, I.W.Smith, R.A. Pucel, and H.A. Haus, GaAs FET Device and Circuit Simulation in Spice*) (CAPOP=1) contain mathematical behavior that has been found to be problematic when trying to fit data.

- For \( v_{gs} \) below the threshold voltage and \( V_{ds} > 0 \) (normal bias condition), \( C_{gd} \) is greater than \( C_{gs} \) and rises with \( V_{ds} \), while \( C_{gs} \) drops with \( V_{ds} \).
- Although \( C_{gd} \) properly goes to a small constant representing a sidewall capacitance, \( C_{gs} \) drops asymptotically to zero with decreasing \( V_{gs} \).
- (For the behavior for \( V_{ds} < 0 \), interchange \( C_{gs} \) and \( C_{gd} \) and replace \( V_{ds} \) with \( -V_{ds} \) in the above descriptions.)
- It can be difficult to simultaneously fit the DC characteristics and the gate capacitances (measured by S-parameters) with the parameters that are shared between the DC model and the capacitance model.
- The capacitance model in the CAPOP=1 implementation also lacks a junction grading coefficient and an adjustable width for the \( V_{gs} \) transition to the threshold voltage. The width is fixed at 0.2.
- Finally, an internal parameter for limiting forward gate voltage is set to 0.8 \( \cdot \) \( PB \) in the CAPOP=1 implementation. This is not always consistent with a good fit.

The CAPOP=2 capacitance equations help to solve the problems described above.
CAPOP=2 Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALPHA</td>
<td>ALPHA</td>
<td>Saturation factor for capacitance model</td>
</tr>
<tr>
<td>CGAMDS</td>
<td>GAMDS</td>
<td>Threshold lowering factor for capacitance</td>
</tr>
<tr>
<td>CVTO</td>
<td>VTO</td>
<td>Threshold voltage for capacitance model</td>
</tr>
<tr>
<td>FC</td>
<td>0.5</td>
<td>PB multiplier – typical value 0.9 gate diode limiting voltage=FC ⋅ PB.</td>
</tr>
<tr>
<td>M (MJ)</td>
<td>0.5</td>
<td>Junction grading coefficient</td>
</tr>
<tr>
<td>VDEL</td>
<td>0.2</td>
<td>Transition width for Vgs</td>
</tr>
</tbody>
</table>

Capacitance Comparison (CAPOP=1 and CAPOP=2)

Figure 17-5 and Figure 17-6 show comparisons of CAPOP=1 and CAPOP=2. Note in Figure 17-5 that below threshold (-0.6 v) Cgs for CAPOP=2 drops towards the same value as Cgd, while for CAPOP=1, CGS → 0.

Note in Figure 17-6 how the Cgs-Cgd characteristic curve “flips over” below threshold for CAPOP=1, whereas for CAPOP=2, it is well-behaved.
Figure 17-5: CAPOP=1 vs. CAPOP=2. Cgs, Cgd vs. Vgs for Vds=0, 1, 2, 3, 4
JFET and MESFET DC Equations

DC Model Level 1

JFET DC characteristics are represented by the nonlinear current source, ids. The value of ids is determined by the equations:

\[ v_{gst} = v_{gs} - VTO \]

\[ v_{gst} < 0 \quad \text{Channel pinched off} \]

\[ ids = 0 \]
0<\text{vgst}<\text{vds} \quad \text{Saturated region}
\[ \text{ids} = BETA_{eff} \cdot \text{vgst}^2 \cdot (1 + \text{LAMBDA} \cdot \text{vds}) \]

0<\text{vds}<\text{vgst} \quad \text{Linear region}
\[ \text{ids} = BETA_{eff} \cdot \text{vds} \cdot (2 \cdot \text{vgst} - \text{vds}) \cdot (1 + \text{LAMBDA} \cdot \text{vds}) \]

The drain current at zero vgs bias (\text{ids}) is related to VTO and BETA by the equation:
\[ \text{idss} = BETA_{eff} \cdot \text{VTO}^2 \]

At a given vgs, LAMBDA can be determined from a pair of drain current and drain voltage points measured in the saturation region where \text{vgst}<\text{vds}:
\[ \text{LAMBDA} = \left( \frac{\text{ids}_2 - \text{ids}_1}{\text{ids}_1 \cdot \text{vds}_2 - \text{ids}_2 \cdot \text{vds}_1} \right) \]

\textbf{DC Model LEVEL 2}

The DC characteristics of the JFET LEVEL 2 model are represented by the nonlinear current source (\text{ids}). The value of \text{ids} is determined by the equations:
\[ \text{vgst} = \text{vgs} - \text{VTO} \]

\text{vgst}<0 \quad \text{Channel pinched off}
\[ \text{ids} = 0 \]

0<\text{vgst} \leq \text{vds, vgs} \quad \text{Saturated region, forward bias}
\[ \text{ids} = BETA_{eff} \cdot \text{vgst}^2 \cdot [1 + \text{LAMBDA} \cdot (\text{vds} - \text{vgst}) \cdot (1 + LAM_{1} \cdot \text{vgs})] \]
Using JFET and MESFET Models

Using JFET and MESFET Model Statements

\[0 < v_{gst} < v_{ds}, \ v_{gs} < 0\] Saturated region, reverse bias

\[i_{ds} = BETA_{eff} \cdot v_{gst}^2 \cdot \left[ 1 - \frac{LAMBDA \cdot (v_{ds} - v_{gst})}{V_{TO}} \right] \cdot \frac{v_{gst}^2}{2}\]

\[0 < v_{ds} < v_{gst}\] Linear region

\[i_{ds} = BETA_{eff} \cdot v_{ds}(2 \cdot v_{gst} - v_{ds})\]

**DC Model LEVEL 3**

The DC characteristics of the MESFET LEVEL 3 model are represented by the nonlinear hyperbolic tangent current source \(i_{ds}\). The value of \(i_{ds}\) is determined by the equations:

\[v_{ds} > 0\] Forward region

If model parameters \(V_P\) and \(V_{TO}\) are not specified they are calculated as:

\[V_P = \frac{-q \cdot NCHAN \cdot A_{eff} f^2}{2 \cdot D \cdot \varepsilon_o}\]

\[V_{TO} = V_P + VBI\]

then,

\[v_{gst} = v_{gs} - [V_{TO} + GAMDS \cdot v_{ds} + K1(v_{bs})]\]

\[bet_{eff} = \frac{BETA_{eff}}{(1 + UCRT \cdot v_{gst})}\]

\[v_{gst} < 0\] Channel pinched off

\[i_{ds} = id_{subthreshold}(N0, ND, v_{ds}, v_{gs})\]
Using JFET and MESFET Model Statements

\textbf{vgst}\textgreater{}0, \textbf{SAT=}0 \quad \text{On region}

\[\text{ids} = \text{betteff} \cdot \text{vgst}^{\text{VGEXP}} \cdot (1 + \text{LAMBDA} \cdot \text{vds}) \cdot \tanh(\text{ALPHA} \cdot \text{vds}) + \text{idsubthreshold}(\text{N0, ND, vds, vgs})\]

\textbf{vgst}\textgreater{}0, \textbf{SAT=}1 \quad \text{On region}

\[\text{ids} = \text{betteff} \cdot \text{vgst}^{\text{VGEXP}} \cdot (1 + \text{LAMBDA} \cdot \text{vds}) \cdot \tanh\left(\text{ALPHA} \cdot \frac{\text{vds}}{\text{vgst}}\right) + \text{idsubthreshold}(\text{N0, ND, vds, vgs})\]

\textbf{vgst}\textgreater{}0, \textbf{SAT=}2, \textbf{vds}\textless{}3/\text{ALPHA} \quad \text{On region}

\[\text{ids} = \text{betteff} \cdot \text{vgst}^2 \cdot (1 + \text{LAMBDA} \cdot \text{vds}) \cdot \left[1 - \left(1 - \text{ALPHA} \cdot \frac{\text{vds}}{3}\right)^3\right] + \text{idsubthreshold}(\text{N0, ND, vds, vgs})\]

\textbf{vgst}\textgreater{}0, \textbf{SAT=}2, \textbf{vds}\textgreater{}3/\text{ALPHA} \quad \text{On region}

\[\text{ids} = \text{betteff} \cdot \text{vgst}^2 \cdot (1 + \text{LAMBDA} \cdot \text{vds}) + \text{idsubthreshold}(\text{N0, ND, vds, vgs})\]

If \text{vgst} >0, \text{SAT}=3 is the same as \text{SAT}=2, except exponent 3 and denominator 3 are parameterized as \text{SATEXP}, and exponent 2 of \text{vgst} is parameterized as \text{VGEXP}.

\textbf{Note:} \text{idsubthreshold is a special function that calculates the subthreshold currents given the model parameters N0 and ND.}
JFET and MESFET Noise Models

Noise Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>1.0</td>
<td>Flicker noise exponent</td>
</tr>
<tr>
<td>KF</td>
<td>0.0</td>
<td>Flicker noise coefficient. Reasonable values for KF are in the range 1e-19 to 1e-25 V^2 F.</td>
</tr>
<tr>
<td>NLEV</td>
<td>2.0</td>
<td>Noise equation selector</td>
</tr>
<tr>
<td>GDSNOI</td>
<td>1.0</td>
<td>Channel noise coefficient. Use with NLEV=3.</td>
</tr>
</tbody>
</table>

Noise Equations

The JFET noise model is shown in Figure 17-4. Thermal noise generation in the drain and source regions (RD and RS resistances) is modeled by the two current sources, \(inrd\) and \(inrs\). The units of \(inrd\) and \(inrs\) are:

\[
inrd = \left( \frac{4 \cdot k \cdot T}{r_d} \right)^{1/2}
\]

\[
inrs = \left( \frac{4 \cdot k \cdot T}{r_s} \right)^{1/2}
\]

Channel thermal and flicker noise are modeled by the current source \(ind\) and defined by the equation:

\[ ind = \text{channelthermalnoise} + \text{flickernoise} \]

If the model parameter NLEV is less than 3, then:

\[ \text{channelthermalnoise} = \left( \frac{8 \cdot k \cdot t \cdot g_m}{3} \right)^{1/2} \]
The previous formula is used in both saturation and linear regions, which can lead to wrong results in the linear region. For example, at VDS=0, channel thermal noise becomes zero, because gm=0. This is physically impossible. If the NLEV model parameter is set to 3, Star-Hspice uses a different equation, which is valid in both linear and saturation regions (See Tsividis, Yanis P., Operation and Modeling of the MOS Transistor, McGraw-Hill, 1987, p. 340).

For NLEV=3

\[
\text{channelthermalnoise} = \left( \frac{8kT}{3} \right) \frac{\text{BETAeff} \cdot (vgs - VTO)}{\alpha vgs - VTO} \cdot \frac{1 + \frac{a^2}{a}}{GDSNOI}
\]

where

\[
\alpha = 1 - \frac{vds}{vgs - VTO}, \quad \text{Linear region}
\]

\[
\alpha = 0, \quad \text{Saturation region}
\]

The flicker noise is calculated as:

\[
\text{flickernoise} = \left( \frac{KF \cdot \text{ids}^{AF}}{f} \right)^{1/2}
\]
Noise Summary Printout Definitions

- **RD, \(V^2/\text{HZ}\)**: Output thermal noise due to drain resistor
- **RS, \(V^2/\text{HZ}\)**: Output thermal noise due to source resistor
- **RG, \(V^2/\text{HZ}\)**: Output thermal noise due to gate resistor
- **ID, \(V^2/\text{HZ}\)**: Output thermal noise due to channel
- **FN, \(V^2/\text{HZ}\)**: Output flicker noise
- **TOT, \(V^2/\text{HZ}\)**: Total output noise (TOT = RD + RS + RG + ID + FN)
- **ONOISE**: Output noise
- **INOISE**: Input noise
JFET and MESFET Temperature Equations

Table 17-4 lists temperature effect parameters. The temperature effect parameters apply to LEVELs 1, 2, and 3. They include temperature parameters for the effect of temperature on resistance, capacitance, energy gap, and a number of other model parameters. The temperature equation selectors, TLEV and TLEVC, select different temperature equations for the calculation of energy gap, saturation current, and gate capacitance. TLEV is either 0, 1, or 2 while TLEVC is either 0, 1, 2, or 3.

Table 17-4: Temperature Parameters (LEVELs 1, 2, and 3)

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>capacitance</td>
<td>CTD, CTS</td>
</tr>
<tr>
<td>DC</td>
<td>M, TCV, XTI</td>
</tr>
<tr>
<td>energy gap</td>
<td>EG, GAP1, GAP2</td>
</tr>
<tr>
<td>equation selections</td>
<td>TLEV, TLEVC</td>
</tr>
<tr>
<td>grading</td>
<td>M</td>
</tr>
<tr>
<td>mobility</td>
<td>BEX</td>
</tr>
<tr>
<td>resistance</td>
<td>TRD, TRS</td>
</tr>
</tbody>
</table>
# Temperature Effect Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETATCE</td>
<td>1/°</td>
<td>0.0</td>
<td>Beta temperature coefficient for TriQuint model</td>
</tr>
<tr>
<td>BEX</td>
<td></td>
<td>0.0</td>
<td>Mobility temperature exponent, correction for low field mobility</td>
</tr>
<tr>
<td>CTD</td>
<td>1/°</td>
<td>0.0</td>
<td>Temperature coefficient for gate-drain junction capacitance. ( \text{TLEVC}=1 ) enables CTD to override the default temperature compensation.</td>
</tr>
<tr>
<td>CTS</td>
<td>1/°</td>
<td>0.0</td>
<td>Temperature coefficient for gate-source junction capacitance. ( \text{TLEVC}=1 ) enables CTS to override the default temperature compensation.</td>
</tr>
<tr>
<td>EG</td>
<td>eV</td>
<td>1.16</td>
<td>Energy gap for the gate to drain and gate to source diodes at 0 °K</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.17 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.69 - Schottky barrier diode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.67 - germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.52 - gallium arsenide</td>
</tr>
<tr>
<td>GAP1</td>
<td>eV/°</td>
<td>7.02e-4</td>
<td>First bandgap correction factor, from Sze, alpha term</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7.02e-4 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.73e-4 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.56e-4 - germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.41e-4 - gallium arsenide</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>GAP2</td>
<td>x</td>
<td>1108</td>
<td>Second bandgap correction factor, from Sze, beta term</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1108 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>636 - silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>210 - germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>204 - gallium arsenide</td>
</tr>
<tr>
<td>M (MJ)</td>
<td>0.50</td>
<td></td>
<td>Grading coefficient for gate-drain and gate-source diodes</td>
</tr>
<tr>
<td>N</td>
<td>1.0</td>
<td></td>
<td>Emission coefficient for gate-drain and gate-source diodes</td>
</tr>
<tr>
<td>TCV (VTOTC)</td>
<td>1/°</td>
<td>0.0</td>
<td>Temperature compensation coefficient for VTO (threshold voltage)</td>
</tr>
<tr>
<td>TLEV</td>
<td>0.0</td>
<td></td>
<td>Temperature equation selector for junction diodes. Interacts with the TLEVC parameter.</td>
</tr>
<tr>
<td>TLEVC</td>
<td>0.0</td>
<td></td>
<td>Temperature equation selector for junction capacitances and potential. Interacts with the TLEV parameter.</td>
</tr>
<tr>
<td>TPB</td>
<td>V/°</td>
<td>0.0</td>
<td>Temperature coefficient for PB. TLEVC=1 or 2 overrides the default temperature compensation.</td>
</tr>
<tr>
<td>TRD (TDR1)</td>
<td>1/°</td>
<td>0.0</td>
<td>Temperature coefficient for drain resistance</td>
</tr>
<tr>
<td>TRG (TRG1)</td>
<td>1/°</td>
<td>0</td>
<td>Temperature coefficient for gate resistance</td>
</tr>
<tr>
<td>TRS (TRS1)</td>
<td>1/°</td>
<td>0.0</td>
<td>Temperature coefficient for source resistance</td>
</tr>
</tbody>
</table>
Temperature Compensation Equations

Energy Gap Temperature Equations

To determine energy gap for temperature compensation, use the equation:

\[ T_{LEV} = 0 \text{ or } 1 \]

\[ e_{gnom} = 1.16 - 7.02e^{-4} \cdot \frac{t_{nom}^2}{t_{nom} + 1108.0} \]

\[ e_{g(t)} = 1.16 - 7.02e^{-4} \cdot \frac{t^2}{t + 1108.0} \]

\[ T_{LEV} = 2 \]

\[ e_{gnom} = EG - GAP1 \cdot \frac{t_{nom}^2}{t_{nom} + GAP2} \]

\[ e_{g(t)} = EG - GAP1 \cdot \frac{t^2}{t + GAP2} \]

Saturation Current Temperature Equations

The saturation current of the gate junctions of the JFET varies with temperature according to the equation:

\[ i_{s}(t) = IS \cdot e^{\frac{facin}{N}} \]
Gate Capacitance Temperature Equations

There are temperature equations for the calculation of gate capacitances. The parameters CTS and CTD are the linear coefficients. If the TLEVC is set to zero, the SPICE equations are used. To achieve a zero capacitance variation, set the coefficients to a very small value such as 1e-6 and TLEVC=1 or 2.

**TLEVC=0**

\[
CGS(t) = CGS \cdot \left[ 1 + M \cdot \left( 4.0 \cdot 10^{-4} \cdot \Delta t - \frac{PB(t)}{PB} + 1 \right) \right]
\]

\[
CGD(t) = CGD \cdot \left[ 1 + M \cdot \left( 4.0 \cdot 10^{-4} \cdot \Delta t - \frac{PB(t)}{PB} + 1 \right) \right]
\]

where:

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

**TLEVC=1**

\[
CGS(t) = CGS \cdot (1 + CTS \cdot \Delta t)
\]

\[
CGD(t) = CGD \cdot (1 + CTD \cdot \Delta t)
\]
where:

\[ PB(t) = PB - TPB \cdot \Delta t \]

**TLEVC=2**

\[ CGS(t) = CGS \cdot \left( \frac{PB}{PB(t)} \right)^M \]

\[ CGD(t) = CGD \cdot \left( \frac{PB}{PB(t)} \right)^M \]

where:

\[ PB(t) = PB - TPB \cdot \Delta t \]

**TLEVC=3**

\[ CGS(t) = CGS \cdot \left( 1 - 0.5 \cdot dpbdtdt \cdot \frac{\Delta t}{PB} \right) \]

\[ CGD(t) = CGD \cdot \left( 1 - 0.5 \cdot dpbdtdt \cdot \frac{\Delta t}{PB} \right) \]

where:

\[ PB(t) = PB + dpbdtdt \cdot \Delta t \]

**TLEV=0 or 1**

\[ dpbdtdt = \left[ \frac{egnom + 3 \cdot vt(tnom) + (1.16 - egnom) \cdot \left( 2 - \frac{tnom}{tnom + 1108} \right) - PB}{tnom} \right] \]
TLEV=2

\[
dpbdt = \left[ \frac{egnom + 3 \cdot vt(t_{nom}) + (EG - egnom) \cdot \left( \frac{tnom}{tnom + GAP^2} \right) - PB}{tnom} \right]
\]

**Threshold Voltage Temperature Equation**

The threshold voltage of the JFET varies with temperature according to the equation:

\[
VTO(t) = VTO - TCV \cdot \Delta t
\]

\[
CVTO(t) = CVTO - TCV \cdot \Delta t
\]

**Mobility Temperature Equation**

The mobility temperature compensation equation is updated as:

\[
BETA(t) = BETA \cdot \left( \frac{t}{tnom} \right)^{BEX} \quad \text{if } BETATCE=0
\]

Otherwise (TriQuint model):

\[
BETA(T) = BETA \cdot 1.01^{BETATCE(t - t_{nom})}
\]

**Parasitic Resistor Temperature Equations**

The RD and RS resistances in JFET vary with temperature according to the equations:

\[
RD(t) = RD \cdot (1 + TRD \cdot \Delta t)
\]

\[
RS(t) = RS \cdot (1 + TRS \cdot \Delta t)
\]

\[
RG(t) = RG \cdot (1 + TRG \cdot \Delta t)
\]
Using TriQuint Model (TOM) Extensions to LEVEL=3

TOM “TriQuint’s Own Model” (See A.J. McCamant, G.D. Mc Cormack, and D.H.Smith, An Improved GaAs MESFET Model for SPICE, IEEE) is implemented as part of the existing GaAs LEVEL 3 model. See W.Curtice, A MGFET Model For Use In the Design of GaAs Integrated Circuits, IEEE Tran, Microwave and H.Statiz, P.Newman, I.W.Smith, R.A. Pucel, and H.A. Haus, “GaAs FET Device And Circuit Simulation in SPICE”.

There are a few differences from the original implementation. The Star-Hspice version of the TOM model takes advantage of existing LEVEL 3 features to provide:

- Subthreshold model (NG, ND)
- Channel and source/drain resistances, geometrically derived from width and length (RD, RG, RS, RSH, RSHG, RSHL, HDIF, LDIF) (ACM=1)
- Photolithographic compensation (LDEL, WDEL, ALIGN)
- Substrate terminal
- Geometric model with width and length specified in the element (ACM=1)
- Automatic model selection as a function of width and length (WMIN, WMAX, LMIN, LMAX)
- User-defined band-gap coefficients (EG, GAP1, GAP2)

Several alias TOM parameters are defined for existing Star-Hspice LEVEL 3 parameters to make the conversion easier. An alias allows the original name or the alias name to be used in the .MODEL statement. However, the model parameter printout is in the original name. Please note that in two cases, a sign reversal is needed, even when using the TOM parameter name.

<table>
<thead>
<tr>
<th>Alias</th>
<th>HSPICE Printout Name</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>VGEXP</td>
<td></td>
</tr>
<tr>
<td>GAMMA</td>
<td>GAMDS</td>
<td>sign opposite of TriQuint’s original</td>
</tr>
<tr>
<td>VTOTC</td>
<td>TCV</td>
<td>sign opposite of TriQuint’s original</td>
</tr>
</tbody>
</table>
Using TriQuint Model (TOM) Extensions to LEVEL=3

TOM Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETATCE</td>
<td>Temperature coefficient for BETA If betatce is set to a nonzero value: $BETA_{\text{temp}} = BETA_{\text{tnom}} \cdot 1.01^{(BETATCE \cdot (\text{temp} - \text{tnom})}$ The more common HSPICE Beta temperature update is: $BETA_{\text{temp}} = BETA_{\text{tnom}} \cdot \left(\frac{\text{temp}}{\text{tnom}}\right)^{\text{BEX}}$</td>
</tr>
<tr>
<td>DELTA</td>
<td>Ids feedback parameter of the TOM model. This parameter is not used if its value is zero. DELTA can be negative or positive. $i_{ds} \Rightarrow \max\left[\left(-1 + v_{nto}\right)\cdot(Delta + v_{ds} \cdot i_{ds}), i_{ds}\right]$</td>
</tr>
<tr>
<td>CAPDS</td>
<td>Drain-to-source capacitance $CAPDSeff = CAPDS \cdot \frac{W_{eff}}{L_{eff}} \cdot M$</td>
</tr>
</tbody>
</table>

**Note:** In the original TOM implementation by TriQuint, parameters LAMBDA and UCRIT do not exist. Therefore, they must remain zero (their default value) in HSPICE LEVEL 3 in order to reproduce the TOM model. Use of nonzero values for these parameters with nonzero BETATCE, DELTA, or CAPDS results in a hybrid model.
Chapter 18

Using Transmission Lines

A transmission line delivers an output signal at a distance from the point of signal input. Any two conductors can make up a transmission line. The signal that is transmitted from one end of the pair to the other end is the voltage between the conductors. Power transmission lines, telephone lines, and waveguides are examples of transmission lines. Traces on printed circuit boards and multichip modules (MCMs) and within integrated circuits are other examples of transmission lines.

This chapter describes the basic transmission line simulation equations. It explains how to use these equations as an input to a transmission line model, the W Element. It also shows you an optional method of computing the parameters of the transmission line equations using the Star-Hspice field solver.

The W Element in Star-Hspice is a versatile transmission line model that can be applied to simulate from a simple lossless line to complex frequency-dependent lossy coupled lines with high efficiency and accuracy. Unlike the U Element discussed in “Ideal and Lumped Transmission Lines” on page C-1, the W Element does not require fine-tuning of optional parameters that are needed to get accurate results.

However, transmission line simulation is challenging and time-consuming since extracting the transmission line parameters from physical geometry takes a significant effort. A simple but efficient and accurate 2-D electromagnetic field solver is newly introduced in Star-Hspice to calculate the electrical parameters of a transmission line based on its cross-section and makeup.

This chapter covers the following topics:

- Using Transmission Line Equations and Parameters
- Using the W Element
- Tabular RLGC Model for W Element
- Tabular RLGC Model Syntax
- Extracting Transmission Line Parameters
- Field Solver Examples
Using Transmission Line Equations and Parameters

Maxwell’s equations for the transverse electromagnetic (TEM) waves on multiconductor transmission lines reduce to the telegrapher’s equations. The general form of the telegrapher’s equations in the frequency domain are given by:

\[-\frac{\partial}{\partial z}v(z, \omega) = [R(\omega) + j\omega L(\omega)]i(z, \omega)\]

\[-\frac{\partial}{\partial z}i(z, \omega) = [G(\omega) + j\omega C(\omega)]v(z, \omega)\]

where, boldface lower-case and upper-case symbols denote vectors and matrices, respectively. \(v\) is the voltage vector across the lines and \(i\) is the current vector along the lines.

For the TEM mode, the transverse distribution of electromagnetic fields at any instant of time is identical to that for the static solution. So, you can derive the four parameters for multiconductor TEM transmission lines, the resistance matrix \(R\), the inductance matrix \(L\), the conductance matrix \(G\), and the capacitance matrix \(C\), from a static analysis. The telegrapher’s equations and the four parameter matrices from a static analysis completely and accurately describe TEM lines.

Unfortunately, all lines do not support pure TEM waves; some multiconductor systems inherently produce longitudinal field components. In particular, waves propagating in the presence of conductor losses or dielectric inhomogeneity (but not dielectric losses) must have longitudinal components. However, if the transverse components of fields are significantly larger than the longitudinal components, the telegrapher’s equations and the four parameter matrices obtained from a static analysis still provide a good approximation. This is known as a quasi-static approximation. Multiconductor systems in which this approximation is valid are called quasi-TEM lines. For typical microstrip systems, the quasi-static approximation holds up to a few gigahertz.
Using Frequency-Dependent Resistance and Conductance Matrices

In contrast to the static (constant) $L$ and $C$ matrices, which provide good accuracy for a wide range of frequencies, the static (DC) $R$ is only good for a very limited frequency range mainly due to the skin effect. A good approximate expression of the resistance matrix $R$ with the skin effect is:

$$R(f) \equiv R_o + \sqrt{f}(1 + j)R_s$$

where $R_o$ is the DC resistance matrix and $R_s$ is the skin effect matrix. The imaginary term depicts the correct frequency response at high frequency; however, it may cause significant errors for low frequency applications. In the W Element, this imaginary term can optionally be excluded.

On the other hand, the conductance matrix $G$ is often approximated as:

$$G(f) \equiv G_o + \frac{f}{\sqrt{1 + (f/f_{gd})^2}}G_d$$

where $G_o$ models the shunt current due to free electrons in imperfect dielectrics and $G_d$ models the power loss due to the rotation of dipoles under the alternating field$^1$.

Determining Matrix Properties

All matrices in the previous section are symmetric. The diagonal terms of $L$ and $C$ are positive nonzero. The diagonal terms of $R_o$, $R_s$, $G_o$, and $G_d$ are nonnegative (can be zero). Off-diagonal terms of impedance matrices $L$, $R_o$, and $R_s$ are nonnegative. Off-diagonal terms of admittance matrices $C$, $G_o$, and $G_d$ are nonpositive. Off-diagonal terms of all matrices can be zero.

---

The elements of admittance matrices are related to the self/mutual admittances (as those inputted by U Element):

\[ Y_{ii} = \sum_{j=1}^{N} Y_{ij}^{(self)/(mutual)} \]

\[ Y_{ij} = -Y_{ij}^{mutual}, \text{ } i \neq j \]

where \( Y \) stands for \( C \), \( G_o \), or \( G_d \). The elements of the impedance matrices \( L \), \( R_o \), and \( R_s \) are the same as self/mutual impedances.

A diagonal term of an admittance matrix is the sum of all the self and mutual admittances in its row. It is larger in absolute value than the sum of all off-diagonal terms in its row or column. Admittance matrices are strictly diagonally dominant (except for a zero matrix).

**Understanding Wave Propagation on Transmission Lines**

To illustrate the physical processes of wave propagation and reflection in transmission lines, consider the line with simple terminations excited with the voltage step as shown in Figure 18-1.

At the time \( t=t_1 \), a voltage step from the source \( e_1 \) attenuated by the impedance \( Z_1 \) is propagating along the transmission line.

At \( t=t_2 \), the voltage wave arrives at the far end of the transmission line, gets reflected, and is propagating in the backward direction. The voltage at the load end is the sum of the incident and reflected waves.

At \( t=t_3 \), the reflected wave arrives back at the near end, gets reflected again, and is again propagating in the forward direction. The voltage at the source end is the sum of the attenuated voltage from the source \( e_1 \), the backward wave, and reflected forward wave.

---

Figure 18-1: Propagation of a Voltage Step in a Transmission Line

Propagating a Voltage Step in a Transmission Line

A summary of the process in Figure 18-1 is:

- Signals from the excitation sources spread out in the termination networks and propagate along the line.
- As the forward wave reaches the far-end termination, it reflects, propagates backward, reflects from the near-end termination, propagates forward again, and continues in a loop.
- The voltage at any point along the line, including the terminals, is a superposition of the forward and backward propagating waves.

Figure 18-2 shows the system diagram of this process.
The model reproduces the general relationship between the physical phenomena of wave propagation, transmission, reflection, and coupling in a distributed system. It can represent arbitrarily distributed systems such as transmission lines, waveguides, and plane-wave propagation. The model is very useful for system analysis of distributed systems, and lets you write the macrosolution for a distributed system without complicated mathematical derivations.

\( W_{vr} \) and \( W_{vb} \) are the forward and backward matrix propagation functions for voltage waves; \( T_1, T_2 \) and \( \Gamma_1, \Gamma_2 \) stand for the near- and far-end matrix transmission and reflection coefficients.
Transmission lines along with terminations form a feedback system (as shown in Figure 18-2). Since the feedback loop contains a delay, the phase shift and the sign of the feedback change periodically with frequency. This causes the oscillations in the frequency-domain responses of transmission lines, as those in Figure 18-7(b).

**Handling Line-to-Line Junctions**

An important special case occurs when the line terminates in another line. The system diagram of a line-to-line junction is shown in Figure 18-3. It can be used to solve multilayered plane-wave propagation problems, analyze common waveguide structures, and derive generalized transmission and reflection coefficient formulas and scattering parameter formulas.

![Figure 18-3: System Model for a Line-to-Line Junction](image)

The propagation functions, $W_{vr}$ and $W_{vb}$, describe how a wave is affected by its propagation from one termination to another, and are equal for the forward and backward directions: $W_{vr}$ and $W_{vb}$. Coupling between the conductors of a
multiconductor line is represented by the off-diagonal terms of the propagation functions. As a wave propagates along the line, it experiences delay, attenuation and distortion (see Figure 18-4). Lines with frequency-dependent parameters, and, therefore, all real lines, do not contain the frequency-independent attenuation component.

**Figure 18-4: Propagation Function Transient Characteristics (unit-step response)**

![Propagation Function Transient Characteristics](image-url)
Using the W Element

The W Element, multiconductor lossy frequency-dependent transmission line, provides advanced modeling capabilities for transmission lines.

The W Element provides:
- Exact analytical solution for AC and DC analyses
- No limit on the number of coupled conductors
- No restrictions on the structure of RLGC matrices; all matrices can be full.
- No spurious ringing as that produced by the old U Element (see Figure 18-5)
- Accurate modeling of frequency-dependent loss in the transient analysis

**Figure 18-5: Spurious Ringing in U Element**
The W Element supports the analyses:

- AC
- DC
- Transient
- Parameter sweeps in DC, AC, and transient analyses
- Optimization
- Monte-Carlo

**Using Time-Step Control**

The W Element provides accurate results with just 1-2 time steps per excitation transient (0.1 ns in the above example). It supports Star-Hspice’s iteration count (the option `LVLTIM=0`) and `DVDT` (`LVLTIM=1` or `3`) time step control algorithms. It does not support the LTE (`LVLTIM=2`) algorithm yet. Star-Hspice’s default time-step control algorithm is `DVDT`.

The W Element limits the maximum time step by the smallest transmission line delay in the circuit.

The W Element supports the `TLINLIMIT` option like the T Element. The default value of `TLINLIMIT=0` enables special breakpoint building that improves transient accuracy for short lines, but reduces efficiency. To disable this special breakpoint building, set `TLINLIMIT=1`.

For longer transmission lines, there could be prolonged time intervals when nothing happens at the terminals when the wave propagates along the line. Star-Hspice increases the time step, and when the wave finally reaches the terminal, it decreases the accuracy of simulation. To prevent this, for longer lines excited with short pulses, set the `.option DELMAX` to limit the time step to 0.5-1 of the excitation transient.
.OPTION RISETIME

The .OPTION RISETIME used by U Elements to compute the number of lumped segments, also affects the transient simulation of W Elements with frequency-dependent parameters. It has no effect on AC analysis and W Elements with constant parameters ($R_s = G_d = 0$).

The option overrides W Element’s internal frequency-range control, and should only be used for longer (over 10 m) cables. Setting RISETIME to a smaller value than the actual value of the excitation transient, decreases simulation accuracy.

W Element and Field Solver Updates

Accuracy Improvements for W Element

Accuracy has been improved for:
- Short transmission lines with frequency-dependent parameters
- Voltage level changes due to reflection (for the frequency-dependent parameters only)

Field Solver Bug Fix

A slight accuracy problem with the Star-Hspice internal field solver for the case with both top and bottom ground planes has been discovered and fixed.

Dielectric Loss Modeling

New modeling equations have been implemented to address two issues:

1. Inaccurate dielectric loss.
2. The linear dependency at the high frequency region, causing undesirable effects.

To remedy these issues a new cutoff frequency for the $G_d$ term is introduced as follows:

$$G(f) \equiv G_o + \frac{f}{\sqrt{1 + (f/f_{gd})^2}} G_d$$
The default value of $f_{gd}$ is 0. You can specify an alternate value in the W Element statement:

```
Wxxx i1 i2 ... iN iR o1 o2 ... oN oR N=val L=val fgd=val
```

If you prefer to use the previous linear dependency, set $f_{gd}$ to 0.

**.OPTION RISETIME Setting**

The RISETIME parameter is used to compute the maximum frequency range for the transient analysis of W Element. Depending on its value, the following scheme is employed to determine the maximum frequency of interest:

- Positive value: the inverse of this user-specified value is used as the maximum frequency.
- No setting (recommended): the risetime is automatically determined from source statements. This method works for most cases; in special cases where the netlist contains the dependent source (which scales or shifts the frequency information) the risetime should be explicitly set.
- Zero: the internal W Element-bound computing algorithm is used for each individual transmission line without using the frequency information contained in source statements.

**Imaginary Term Handling for the Skin-Effect**

As of the 1999.4 release of Star-Hspice, to depict the correct frequency response at high frequency, the imaginary term of the skin-effect has been added, and the resulting modeling equation for the frequency dependent resistance is given by:

$$ R(f) = R_o + \sqrt{f}(1 + j)R_s $$

However, for low frequency applications, this may cause significant errors and this imaginary term can optionally be excluded from the W Element statement:

```
Wxxx i1 i2 ... iN iR o1 o2 ... oN oR N=val L=val
INCLUDERSIMAG=NO
```
W Element Transmission Line Properties Inputs

The W Element supports four different formats to input transmission line properties:

- Model 1: RLGC-Model specification.
- Model 2: U-Model specification, including:
  - RLGC input for up to five coupled conductors
  - Geometric input (planar, coax, twinlead)
  - Measured-parameter input
  - Skin effect
- Model 3: Star-Hspice internal field-solver model
- Model 4: External file containing RLGC information. This format is supported only for the purpose of backward compatibility and Model 1 should be used instead. Note that this model does not allow parameterized RLGC values.

Syntax

The syntax of the W Element statement is:

```
Wxxx i1 i2 ... iN iR o1 o2 ... oN oR N=val L=val
+ <FSMODEL=name or RLGCMODEL=name or RLGCFILE=name or
+ UMODEL=name>
+ <INCLUDERSIMAG=YES|NO> <FGD=val>
```

- **N**  
  Number of signal conductors (excluding the reference conductor)

- **i1...iN**  
  Nodes for the near-end signal-conductor terminal  
  (see Figure 18-6)

- **iR**  
  Nodes for the near-end reference-conductor terminal (should be the same node as oR)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>o1...oN</td>
<td>Nodes for the far-end signal-conductor terminal</td>
</tr>
<tr>
<td>oR</td>
<td>Nodes for the far-end reference-conductor terminal</td>
</tr>
<tr>
<td>L</td>
<td>Length of a transmission line</td>
</tr>
<tr>
<td>RLGCMODEL</td>
<td>Name of the RLGC model</td>
</tr>
<tr>
<td>FSMODEL</td>
<td>Name of the field-solver model</td>
</tr>
<tr>
<td>RLGCFI LE</td>
<td>Name of the external file with RLGC parameters (see “Input Model 4: W Element RLGC File” on page 18-24)</td>
</tr>
<tr>
<td>UMODEL</td>
<td>Name of U .MODEL (see “Input Model 2: U Model” on page 18-20).</td>
</tr>
<tr>
<td>INCLUDEDSIMAG</td>
<td>Specifies the imaginary term of the skin effect to be considered. The default value is YES. (see “Using Frequency-Dependent Resistance and Conductance Matrices” on page 18-4).</td>
</tr>
<tr>
<td>FGD</td>
<td>Specifies the cut-off frequency of dielectric loss. (see “Dielectric Loss Modeling” on page 18-12).</td>
</tr>
</tbody>
</table>
You can specify parameters in the W Element card in any order. Specify the number of signal conductors, \(N\), after the list of nodes. You can intermix the nodes and parameters in the W Element card.

You can specify only one RLGCmodel, FSmodel, Umodel, or RLGCfile in a single W Element card.

**Input Model 1: W Element RLGC Model**

The section, “Using Transmission Line Equations and Parameters” on page 18-3 describes the W Element inputs \(R\), \(L\), \(G\), \(C\), \(R_s\) (skin-effect), and \(G_d\) (dielectric-loss) per-unit-length matrices. There are no limitations on the number of coupled conductors, shape of the matrices, line loss, length or amount of frequency dependence. The RLGC text file contains frequency-dependent RLGC matrices per unit length.

The W Element also handles frequency-independent (RLGC) and lossless (LC) lines. It does not support RC lines.

Since RLGC matrices are symmetric, only the lower-triangular parts of the matrices are specified in the RLGC model. The syntax of W Element RLGC model is:

```
.MOdel name W MODELTYPE=RLGC N=val Lo=matrix_entries
```
Using Transmission Lines

Using the W Element

\[ \text{N} \quad \text{Number of signal conductors (same as that in the element card)} \]

\[ \text{L} \quad \text{DC inductance matrix per unit length} \left[ \frac{H}{m} \right] \]

\[ \text{C} \quad \text{DC capacitance matrix per unit length} \left[ \frac{F}{m} \right] \]

**Optional Parameters (default values are zero)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ro</td>
<td>DC resistance matrix per unit length</td>
<td>( \left[ \frac{\Omega}{m} \right] )</td>
</tr>
<tr>
<td>Go</td>
<td>DC shunt conductance matrix per unit length</td>
<td>( \left[ \frac{S}{m} \right] )</td>
</tr>
<tr>
<td>Rs</td>
<td>Skin-effect resistance matrix per unit length</td>
<td>( \left[ \frac{\Omega}{m \cdot \sqrt{Hz}} \right] )</td>
</tr>
<tr>
<td>Gd</td>
<td>Dielectric-loss conductance matrix per unit length</td>
<td>( \left[ \frac{S}{m \cdot Hz} \right] )</td>
</tr>
<tr>
<td>Lgnd</td>
<td>DC inductance matrix per unit length for grounds (reference line)</td>
<td>( \left[ \frac{H}{m} \right] )</td>
</tr>
<tr>
<td>Rognd</td>
<td>DC resistance matrix per unit length for grounds</td>
<td>( \left[ \frac{\Omega}{m} \right] )</td>
</tr>
<tr>
<td>Rsgnd</td>
<td>Skin-effect resistance matrix per unit length for grounds</td>
<td>( \left[ \frac{\Omega}{m \cdot \sqrt{Hz}} \right] )</td>
</tr>
</tbody>
</table>

**Example**
The following is example.sp input netlist file showing the RLGC file input usage of the $W$ Element:

* W-Element example, four-conductor line
W1 N=3 1 3 5 0 2 4 6 0 RLGCMODEL=example_rlc l=0.97
V1 1 0 AC=1v DC=0v pulse(4.82v 0v 5ns 0.1ns 0.1ns 25ns).AC
lin 1000 0Hz 1GHz
.DC v1 0v 5v 0.1v
.tran 0.1ns 200ns
* RLGC matrices for a four-conductor lossy
.MODEL example_rlc W MODELTYPE=RLGC N=3
+ Lo=
  + 2.311e-6
  + 4.14e-7  2.988e-6
  + 8.42e-8  5.27e-7  2.813e-6
+ Co=
  + 2.392e-11
  + -5.41e-12  2.123e-11
  + -1.08e-12  -5.72e-12  2.447e-11
+ Ro=
  + 42.5
  + 0  41.0
  + 0  0  33.5
+ Go=
  + 0.000609
  + -0.0001419  0.000599
  + -0.00002323  -0.00009  0.000502
+ Rs=
  + 0.00135
  + 0  0.001303
  + 0  0  0.001064
+ Gd=
  + 5.242e-13
  + -1.221e-13  5.164e-13
  + -1.999e-14  -7.747e-14  4.321e-13
.end

Figure 18-7 shows a plot of Star-Hspice simulation results (a) DC Sweep, b) AC response, and c) transient waveforms). It shows that the transmission-line behavior of interconnects has significant and complicated effect on the signal
integrity, and accurate transmission line modeling is necessary for verification of high-speed designs.

**Figure 18-7: Star-Hspice Simulation Results**

(a) DC Sweep

(b) AC Response
Input Model 2: U Model

The W Element accepts the U model as an input thus providing backward compatibility with the U Element, and taking advantage of the U model’s geometric and measured-parameter interfaces.

To use the W Element with the U model, specify \texttt{Umodel=U-model\_name} on the W Element card.

The W Element supports all U-model modes, including:

- Geometric, Elev=1
  - planar geometry, Plev=1
  - coax, Plev=2
  - twinlead, Plev=3
- RLGC, Elev=2
- Measured parameters, Elev=3
- Skin-effect, Nlay=2.

The only exception is Llev=1, which adds the second ground plane to the U model, and is not supported by the W Element. To model the extra ground
Using Transmission Lines

plane, add an extra conductor to the W Element in Elev=2, or use an external lumped capacitor in Elev=1 and 3. See “Ideal and Lumped Transmission Lines” on page C-1, for information on the U model.

Using RLGC Matrices

RLGC matrices in the W Element’s RLGC file are in the Maxwellian format. In the U model, they are in self/mutual format (see “Determining Matrix Properties” on page 18-4 for conversion information). When using the U model, the W Element performs the conversion internally. Table 18-1 shows how the U-model’s RLGC matrices are related to the W Element’s RLGC matrices, and how they are used by the W Element.

Handling the Dielectric-loss Matrix

Since the U model does not input the dielectric loss matrix $G_d$, the W Element defaults $G_d$ to zero when it uses the U-model input. In future Star-Hspice releases, the W Element will have its own .MODEL with $G_d$ capability. For this release, use RLGC file to specify nonzero $G_d$.

Handling the Skin-effect Matrix

The skin-effect resistance $R_s$ is used differently by U and W Elements. In a W Element, the $R_s$ matrix specifies the square-root dependence of the frequency-dependent resistance,

$$ R(f) = R_o + \sqrt{f}(1 + j)R_s $$

In a U Element, $R$ is the value of skin resistance at the frequency:

$$ R = R_c + R_s $$
where the core resistance \( R_c \) is equivalent to the W Element’s DC resistance \( R_o \). The frequency at which U Element computes \( R_s \) is:

\[
f_{\text{skin}} = \frac{1}{15 \cdot \text{RISETIME}}
\]

**Table 18-1: RLGC Matrices in W Element and U Model**

<table>
<thead>
<tr>
<th>W Element Parameters</th>
<th>U Model Parameters</th>
</tr>
</thead>
</table>
| \( L, C \)            | \[
\begin{bmatrix}
L_{11} & \ & \\
L_{12} & L_{22} & \\
L_{13} & L_{23} & L_{33}
\end{bmatrix}
\] |
| \( C_{r1} + C_{12} + C_{13} \) & \(-C_{12} & C_{r2} + C_{12} + C_{23} \) & \(-C_{13} & -C_{23} & C_{r3} + C_{13} + C_{23} \) |
| \( G_{o}, G_{d} \) | \[
\begin{bmatrix}
G_{r1} + G_{12} + G_{13} & \ & \\
-G_{12} & G_{r2} + G_{12} + G_{23} & \\
-G_{13} & -G_{23} & G_{r3} + G_{13} + G_{23}
\end{bmatrix}
\] |
| \( 0 & 0 & 0 \) |
| \( N_{\text{lay}}=1 \) (no skin effect) | \( N_{\text{lay}}=2 \) (skin effect present) |

| \( R_o \) | \[
\begin{bmatrix}
R_{11} + R_{rr} & \ & \\
R_{rr} & R_{22} + R_{rr} & \\
R_{rr} & R_{rr} & R_{33} + R_{rr}
\end{bmatrix}
\] |
| \( R_{1c} + R_{rc} & \ & \\
R_{rc} & R_{2c} + R_{rc} & \\
R_{rc} & R_{rc} & R_{3c} + R_{rc}
\] |
| \( N_{\text{lay}}=1 \) (no skin effect) | \( N_{\text{lay}}=2 \) (skin effect present) |

| \( R_s \) | \[
\begin{bmatrix}
0 & \ & \\
0 & 0 & \\
0 & 0 & 0
\end{bmatrix}
\] |
| \[
\frac{1}{\sqrt{f_{\text{skin}}}}
\begin{bmatrix}
R_{1s} + R_{rs} & \ & \\
R_{rs} & R_{2s} + R_{rs} & \\
R_{rs} & R_{rs} & R_{3s} + R_{rs}
\end{bmatrix}
\] |

If you do not specify the \textbf{RISETIME} option, the U Element uses Tstep from the .tran card.
Using Transmission Lines

**For U models with**

<table>
<thead>
<tr>
<th><strong>W Element</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>RLGC input; Elev=2</td>
</tr>
<tr>
<td>Geometric input; Elev=1</td>
</tr>
</tbody>
</table>

| Measured-parameter input; Elev=3 | Does not support skin effect. |

**Example**

The following Star-Hspice netlist is for a 4-conductor line shown in Figure 18-8.

```star
* W Element example, four-conductor line, U model
W1 1 3 5 0 2 4 6 0 Umodel=example N=3 l=0.97
.Model example U LEVEL=3 NL=3 Elev=2 Llev=0 Plev=1 Nlay=2
  + L11=2.311uH
  + L12=0.414uH L22=2.988uH
  + L13=84.2nH L23=0.527uH L33=2.813uH
  + C11=17.43pF
  + C12=5.41pF C22=10.1pF
  + C13=1.08pF C23=5.72pF C33=17.67pF
  + R1c=42.5 R2c=41.0 R3c=33.5
  + G1=0.044387mS
  + G12=0.1419mS Gr2=0.3671mS
  + G13=23.23uS G23=90uS Gr3=0.38877mS
  + R1s=0.00135 R2s=0.001303 R3s=0.001064
V1 1 0 AC=1v DC=0v pulse(4.82v 0v 5ns 0.1ns 0.1ns 25ns)
```

Using the \textit{W} Element

Instead of RLGC matrices, you can directly use geometric data with the \textit{W} Element using a new built-in field solver in Star-Hspice. To use the \textit{W} Element with a field solver, specify \texttt{FSmodel=model\_name} on the \textit{W} Element card. The Star-Hspice field solver is described in “Extracting Transmission Line Parameters” on page 18-34.

**Input Model 4: \textit{W} Element RLGC File**

RLGC matrices can also be specified in an external file (RLGC file). This external file format is more restricted than the RLGC model; for example, it cannot be parameterized and does not support the ground inductance and resistance. This format does not provides any advantage over the RLGC model and should not be used. (It is supported for the backward compatibility purpose only.)

Similar to the RLGC model, only the lower-triangular parts of the matrices are specified in the RLGC file. However, unlike the RLGC model, the RLGC file is order-dependent. The parameters in the RLGC file are in the following order:
Using Transmission Lines

Using the W Element

\[ N \]  Number of signal conductors (same as that in the element card)

\[ L \]  DC inductance matrix per unit length \( \frac{H}{m} \)

\[ C \]  DC capacitance matrix per unit length \( \frac{F}{m} \)

Optional Parameters

\[ R_0 \]  DC resistance matrix per unit length \( \frac{\Omega}{m} \)

\[ G_0 \]  DC shunt conductance matrix per unit length \( \frac{S}{m} \)

\[ R_s \]  Skin-effect resistance matrix per unit length \( \frac{\Omega}{m \cdot Hz} \)

\[ G_d \]  Dielectric-loss conductance matrix per unit length \( \frac{S}{m \cdot Hz} \)

\[ \text{Note:} \] You can skip optional parameters (they default to zero). But if you specify one of the optional parameters, you must specify all preceding parameters even if they are zero.

Comments and Separators

An asterisk ‘*’ comments out everything until the end of its line. You can separate numbers using any of the characters: space, tab, newline, ‘,’ ‘;’, ‘(’, ‘)’, ‘[’, ‘]’, ‘{’ or ‘}’.
Example

The netlist example used for the RLGC model in the previous section is rewritten using RLGC file below:

```
* W- Element example, four-conductor line
W1 N=3 1 3 5 0 2 4 6 0 RLGCfile=example.rlc l=0.97
V1 1 0 AC=1v DC=0v pulse(4.82v 0v 5ns 0.1ns 0.1ns 25ns).AC
lin 1000 0Hz 1GHz
.DC v1 0v 5v 0.1v
.tran 0.1ns 200ns
.end
```

This calls the following example.rlc RLGC file:

```
* RLGC parameters for a four-conductor lossy
* frequency-dependent line
* N (number of signal conductors)
  3
* Lo
  2.311e-6
  4.14e-7 2.988e-6
  8.42e-8 5.27e-7 2.813e-6
* Co
  2.392e-11
-5.41e-12 2.123e-11
-1.08e-12 -5.72e-12 2.447e-11
* Ro
  42.5
  0 41.0
  0 0 33.5
* Go
  0.000609
-0.0001419 0.000599
-0.00002323 -0.00009 0.000502
* Rs
  0.00135
  0 0.001303
  0 0 0.001064
* Gd
  5.242e-13
-1.221e-13 5.164e-13
-1.999e-14 -7.747e-14 4.321e-13
```

The RLGC file does not support Star-Hspice scale suffices such as n \((10^{-9})\) or p \((10^{-12})\).
Tabular RLGC Model for W Element

A new tabular RLGC model is a useful extension of the analytical RLGC model as it allows users to model arbitrary frequency-dependent behavior of transmission lines (RC lines are still not supported.)

W Element Syntax for a Tabular RLGC Model

The W Element syntax is extended to allow the specification of a table model given via a .MODEL statement of type w. The .MODEL statement can refer to .MODEL statements (described in the S-parameter section above) which contain the actual table data for the RLGC matrices.

Notation Used

- Lower-cased variable: Scalar quantity
- Upper-cased variable: Matrix quantity
- All upper-cased works: Keyword
- Parenthesis and commas: Optional

Syntax

Wxxx i1 i2 ... iN iR o1 o2 ... oN oR N=val L=val
+ < TABLEMODEL=name or RLGCTABLEMODEL=name or .. >

i1 i2 ... iN iR        Input terminal nodes
o1 o2 ... oN oR        Output terminal nodes
N                    Number of signal conductors
L                    Line length
TABLEMODEL            Table model
Tabular RLGC Model Syntax

.Model name W MODELTYPE=TABLE N=val LMODEL=l_freq_model + CMODEL=c_freq_model [ RMODEL=r_freq_model + GMODEL=g_freq_model ]

N Number of signal conductors.

l_freq_model .MODEL model name for the inductance matrix.
c_freq_model .MODEL model name for the capacitance matrix.
r_freq_model .MODEL model name for the resistance matrix; by default, it is zero.
g_freq_model .MODEL model name for the conductance matrix; by default, it is zero.

Example

This is an example of a 2-line system:
.Model ex1 W MODELTYPE=RLGC N=2 + Lo=3.93346e-7 4.93701e-8 3.93346e-7 + Co=5.60236e-11 -7.04724e-12 5.60236e-11 + Ro=8.77953e-2 6.29921e-3 8.77953e-2 + Rs=2.65950e-4 0 2.65730e-4 + Gd=3.87331e-12 -4.86081e-13 3.87306e-12 .MODEL ex1 W MODELTYPE=TABLE N=2 LMODEL=lmod1 + CMODEL=cmod1 RMODEL=rmod1 GMODEL=gmod1 .MODEL lmod1 sp N=2 SPACING=NONUNIFORM VALTYPE=REAL + DATA=( 1, + (0.000000e+00 5.602360e-11 -7.047240e-12) + ) .MODEL lmod1 N=2 SPACING=NONUNIFORM VALTYPE=REAL + INFINITY=(3.93346e-7 4.93701e-8 3.93346e-7) + DATA=( 34, + (0.000000e+00 3.933460e-07 4.937010e-08 3.933460e-07) + (3.746488e+06 4.152139e-07 4.937010e-08 4.151959e-07)
Using Transmission Lines Tabular RLGC Model Syntax


+ (7.726980e+06 4.085730e-07 4.937010e-08 4.085604e-07)
+ (1.196411e+07 4.055831e-07 4.937010e-08 4.055730e-07)
+ (1.648352e+07 4.037715e-07 4.937010e-08 4.037628e-07)
+ (2.131439e+07 4.025142e-07 4.937010e-08 4.025066e-07)
+ (2.649007e+07 4.015699e-07 4.937010e-08 4.015631e-07)
+ (3.204884e+07 4.008228e-07 4.937010e-08 4.008166e-07)
+ (3.803487e+07 4.002092e-07 4.937010e-08 4.002036e-07)
+ (4.449938e+07 3.996912e-07 4.937010e-08 3.996859e-07)
+ (5.150215e+07 3.992440e-07 4.937010e-08 3.992392e-07)
+ (5.911330e+07 3.988513e-07 4.937010e-08 3.988467e-07)
+ (6.741573e+07 3.985011e-07 4.937010e-08 3.984969e-07)
+ (7.650809e+07 3.981851e-07 4.937010e-08 3.981811e-07)
+ (8.650875e+07 3.978968e-07 4.937010e-08 3.978931e-07)
+ (9.756098e+07 3.976313e-07 4.937010e-08 3.976278e-07)
+ (1.098398e+08 3.973847e-07 4.937010e-08 3.973813e-07)
+ (1.235615e+08 3.971538e-07 4.937010e-08 3.971507e-07)
+ (1.389961e+08 3.969362e-07 4.937010e-08 3.969332e-07)
+ (1.564859e+08 3.967296e-07 4.937010e-08 3.967268e-07)
+ (1.764706e+08 3.965323e-07 4.937010e-08 3.965296e-07)
+ (1.995249e+08 3.963426e-07 4.937010e-08 3.963401e-07)
+ (2.264151e+08 3.961590e-07 4.937010e-08 3.961567e-07)
+ (2.581852e+08 3.959802e-07 4.937010e-08 3.959781e-07)
+ (2.962963e+08 3.958050e-07 4.937010e-08 3.958030e-07)
+ (3.428571e+08 3.956319e-07 4.937010e-08 3.956300e-07)
+ (4.010283e+08 3.954596e-07 4.937010e-08 3.954579e-07)
+ (4.757709e+08 3.952865e-07 4.937010e-08 3.952849e-07)
+ (5.753425e+08 3.951106e-07 4.937010e-08 3.951092e-07)
+ (7.145791e+08 3.949294e-07 4.937010e-08 3.949281e-07)
+ (9.230769e+08 3.947392e-07 4.937010e-08 3.947380e-07)
+ (1.269625e+09 3.945339e-07 4.937010e-08 3.945329e-07)
+ (1.959184e+09 3.943023e-07 4.937010e-08 3.943015e-07)
+ (4.000000e+09 3.940153e-07 4.937010e-08 3.940147e-07)
+ )

.MODEL rmod1 N=2 SPACING=NONUNIFORM VALTYPE=REAL
+ DATA=( 34,
+ (0.000000e+00 8.779530e-02 6.299210e-03 8.779530e-02)
+ (3.746488e+06 6.025640e-01 6.299210e-03 6.021382e-01)
+ (7.726980e+06 8.270684e-01 6.299210e-03 8.264568e-01)
Tabular RLGC Model Syntax

```
+ (1.196411e+07 1.007694e+00 6.299210e-03 1.006933e+00)
+ (1.648352e+07 1.167550e+00 6.299210e-03 1.166656e+00)
+ (2.131439e+07 1.315620e+00 6.299210e-03 1.314604e+00)
+ (2.649007e+07 1.456600e+00 6.299210e-03 1.455468e+00)
+ (3.204884e+07 1.593383e+00 6.299210e-03 1.592138e+00)
+ (3.803487e+07 1.72973e+00 6.299210e-03 1.726616e+00)
+ (4.49938e+07 1.861891e+00 6.299210e-03 1.860423e+00)
+ (5.150215e+07 1.996385e+00 6.299210e-03 1.994807e+00)
+ (5.911330e+07 2.132557e+00 6.299210e-03 2.130865e+00)
+ (6.741573e+07 2.271433e+00 6.299210e-03 2.269627e+00)
+ (7.650809e+07 2.414031e+00 6.299210e-03 2.412106e+00)
+ (8.650875e+07 2.561398e+00 6.299210e-03 2.559352e+00)
+ (9.756098e+07 2.714662e+00 6.299210e-03 2.712489e+00)
+ (1.098398e+08 2.875071e+00 6.299210e-03 2.872765e+00)
+ (1.235615e+08 3.044048e+00 6.299210e-03 3.041602e+00)
+ (1.389961e+08 3.223256e+00 6.299210e-03 3.220662e+00)
+ (1.564859e+08 3.41679e+00 6.299210e-03 3.414279e+00)
+ (1.764706e+08 3.620734e+00 6.299210e-03 3.617812e+00)
+ (1.995249e+08 3.844427e+00 6.299210e-03 3.841319e+00)
+ (2.264151e+08 4.089570e+00 6.299210e-03 4.086260e+00)
+ (2.581852e+08 4.361118e+00 6.299210e-03 4.357833e+00)
+ (2.962963e+08 4.665662e+00 6.299210e-03 4.661875e+00)
+ (3.428571e+08 5.012232e+00 6.299210e-03 5.008159e+00)
+ (4.010283e+08 5.413628e+00 6.299210e-03 5.409222e+00)
+ (4.757709e+08 5.888743e+00 6.299210e-03 5.883944e+00)
+ (5.753425e+08 6.466951e+00 6.299210e-03 6.461674e+00)
+ (7.145791e+08 7.197067e+00 6.299210e-03 7.191816e+00)
+ (9.230769e+08 8.167936e+00 6.299210e-03 8.162522e+00)
+ (1.269625e+09 9.564070e+00 6.299210e-03 9.556231e+00)
+ (1.959184e+09 1.185945e+01 6.299210e-03 1.184971e+01)
+ (4.000000e+09 1.690795e+01 6.299210e-03 1.689404e+01)
+ )

.MODEL gmod1 N=2 SPACING=NONUNIFORM VALTYPE=REAL
+ DATA= ( 34,
+ (0.000000e+00 5.967166e-11 0.000000e+00 5.967166e-11)
+ (3.746488e+06 1.451137e-05 -1.821096e-06 1.451043e-05)
+ (7.726980e+06 2.992905e-05 -3.755938e-06 2.992712e-05)
```
Using Transmission Lines Tabular RLGC Model Syntax

See the “Frequency Table Model” on page 6-4.

Example

This is an example of a 4-conductor transmission line system.

### Table 18-2: Input File Listing

<table>
<thead>
<tr>
<th>Header, options and sources</th>
<th>W Element Tabular Model Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTIONS POST</td>
<td></td>
</tr>
<tr>
<td>V1 7 0 ac=1v dc=0.5v pulse(0.5v 1.5v 0ns 0.1ns)</td>
<td></td>
</tr>
<tr>
<td>V2 8 0 dc=1v</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Analysis</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>.DC v1 0.5v 5.5v 0.1v SWEEP length POI 2 1.2 2</td>
<td></td>
</tr>
<tr>
<td>.AC lin 200 0Hz 1GHz SWEEP Ro POI 3 400 41.6667 400</td>
<td></td>
</tr>
<tr>
<td>.TRAN 0.1ns 50ns</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Termination</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R1 7 1 50</td>
<td></td>
</tr>
<tr>
<td>R2 4 0 450</td>
<td></td>
</tr>
<tr>
<td>R3 5 0 450</td>
<td></td>
</tr>
<tr>
<td>R8 6 0 450</td>
<td></td>
</tr>
<tr>
<td>R5 4 5 10800</td>
<td></td>
</tr>
<tr>
<td>R6 5 6 10800</td>
<td></td>
</tr>
<tr>
<td>R7 4 6 1393.5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Analytical RLGC model (W Element)</th>
<th>.SUBCKT sub 1 2 3 4 5 6 7 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1 n=3 1 2 3 4 5 6 7 8 l=0.1 fgd=5e6 RLGCMODEL=analymod</td>
<td></td>
</tr>
<tr>
<td>.MODEL analymod W MODELTYPE=RLGC N=3</td>
<td></td>
</tr>
<tr>
<td>+ Lo=2.41667e-6</td>
<td>+ 0 12.3457e-12</td>
</tr>
<tr>
<td>+ 0.694444e-6</td>
<td>+ 0.638889e-6 0.694444e-6 2.41667e-6</td>
</tr>
<tr>
<td>+ Co=20.9877e-12</td>
<td>+ -12.3457e-12 29.3210e-12</td>
</tr>
<tr>
<td>+ Go=41.6667</td>
<td>+ -4.01235e-12 -12.3457e-12</td>
</tr>
<tr>
<td>+ Ro=41.6667</td>
<td>+ 0 41.6667</td>
</tr>
<tr>
<td>+ Rs=0.785e-5</td>
<td>+ 0 0.585937e-3</td>
</tr>
<tr>
<td>+ Gd=0.285e-6</td>
<td>+ 0 0.785e-5</td>
</tr>
<tr>
<td>+ 0</td>
<td>+ 0 0 0.785e-5</td>
</tr>
<tr>
<td>+ 0</td>
<td>+ 0 0 0.285e-6</td>
</tr>
<tr>
<td>+ 0</td>
<td>+ 0 0 0.285e-6</td>
</tr>
<tr>
<td>.ENDS sub</td>
<td>.ENDS sub</td>
</tr>
</tbody>
</table>
### Tabular RLGC Model Syntax

**Tabular RLGC model (W Element)**

```
.W Element
.ALTER Tabular Model
.SUBCKT sub 1 2 3 4 5 6 7 8
W1 n=3 1 2 3 4 5 6 7 8 l=0.1 fgd=5e6 tablem odel=trmod
.INCLUDE table.txt
.ENDS sub
```

### Table 18-3: Tabular RLGC Model

<table>
<thead>
<tr>
<th>Element</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLGC table model definition</td>
<td><code>.MODEL trmod W MODELTYPE=TABLE N=3 + LMODEL=lmod CMODEL=cmod RMODEL=rmod GMODEL=gmod</code></td>
</tr>
<tr>
<td>C model</td>
<td><code>.MODEL cmod sp N=3 VALTYPE=REAL INTERPOLATION=LINEAR + DATA=( 1 2.09877e-11 -1.23457e-11 2.9321e-11 + -4.01235e-12 -1.23457e-11 2.09877e-11 )</code></td>
</tr>
<tr>
<td>R model</td>
<td><code>.MODEL rmod sp N=3 VALTYPE=REAL INTERPOLATION=LINEAR + FSTOP=1e+10 + DATA=( 200 + 41.6667 0 41.6667 0 41.6667 + 41.7223 0 41.7223 0 41.7223 + 42.4997 0 42.4997 0 42.4997 + 42.4517 0 42.4517 0 42.4517 )</code></td>
</tr>
<tr>
<td>G model</td>
<td><code>.MODEL gmod sp N=3 VALTYPE=REAL INTERPOLATION=LINEAR + FSTOP=1e+08 + DATA=( 100 + 0.000585937 0 0.000585937 0 0 0.000585937 + 0.282764 0 0.282764 0 0 0.282764 + 1.42377 0 1.42377 0 1.42377 + 1.42381 0 1.42381 0 1.42381 )</code></td>
</tr>
</tbody>
</table>
Extracting Transmission Line Parameters

Star-Hspice includes a built-in 2-D electromagnetic field solver, which is highly optimized for interconnects in stratified media. Its implementation makes use of the W Element and supports optimization and statistical analysis within transient simulation. The solver is based on an improved version of the boundary-element method and filament method that is also implemented in Raphael.

Filament Method

This section describes the filament method for the skin effect resistance and inductance solver.

The 2D filament method is introduced in release 2001.2 to extract frequency dependent resistance and inductance by taking magnetic coupling into account. This solver is invoked when COMPUTERS=yes is set in .fs option.

The principle of the filament method is that the original conductor system is divided into thin filaments. The distributed magnetic coupling of the inside and outside of the conductor is derived from the coupling of these filaments.

Once conductors are divided into thin filaments, the impedance matrix of the filament system is given as $Z_f (= R_f + j\omega L_f)$. Then, the current matrix $i_f$ can be solved by:

$v_f = Z_f i_f$

Here, $v_f$ is an excitation vector for the filament system. As a result, the partial current matrix of conductor system $i_p$ can be calculated as a summation of the filament current as follows:

$$i_{p(j,k)} = \sum_{\text{filaments in conductor } j} i_f \quad (\text{@ } k\text{-th excitation vector})$$

The partial impedance matrix \( Z_p \) is solved by:

\[ v_p = Z_p i_p \]

Loop impedance matrix components \( Z_{l(j,k)} \) which are needed for \( W \) element analysis are calculated from partial impedance matrix components \( Z_{p(j,k)} \) based on the following relationship:

\[ z_{l(j,k)} = z_{p(j,k)} - z_{p(j,0)} - z_{p(k,0)} + z_{p(0,0)} \]

Here, \( n \) is the number of signal (non-reference) conductors in the system.

**Modeling Geometries**

In geometry modeling:
- There is an arbitrary number of dielectric layers.
- The conductor cross-section can be arbitrarily shaped including an infinitely thin strip.
- There is no limit on the number of conductors.
- The current dielectric region must be planar.
- All conductors must be nonoverlapping.
- Magnetic materials are not supported.

**Using Geometric Modeling Output**

Geometric modeling outputs the Maxwellian (short-circuit) transmission line matrices: \( C, L, R_o, R_s, G_o, \) and \( G_d \) (see “Using Transmission Line Equations and Parameters” on page 18-3).
Solver Limitations

The Star-Hspice field solver has the following limitation: the arithmetic average values of conductivities and loss tangents are used for inhomogeneous media when it computes the conductance matrices, $G_o$ and $G_d$.

Using the Field-Solver Statement Syntax

The Star-Hspice input syntax related to the field solver contains five new statements: .MATERIAL, .LAYERSTACK, .SHAPE and .FSOPTIONS, and a .MODEL with statement. Include these new statements in a netlist to define the material properties, stacking of materials, material shapes, field solver options and the transmission-line model as follows.

Defining Material Properties

Use the .MATERIAL statement to define the properties of a material.

Syntax

```
.MATERIAL mname METAL|DIELECTRIC <ER=val> <UR=val> + <CONDUCTIVITY=val> <LOSSTANGENT=val>
```

- **mname**: Material name
- **METAL|DIELECTRIC**: Material type: METAL or DIELECTRIC
- **ER**: Dielectric constant (relative permittivity)
- **UR**: Relative permeability
- **CONDUCTIVITY**: Static field conductivity of conductor or lossy dielectric (S/m)
- **LOSSTANGENT**: Alternating field loss tangent of dielectric (tan δ)
Handling Metals

The Star-Hspice field solver assigns the following default values for metal: \text{CONDUCTIVITY} = -1 \ (\text{perfect conductor}), \text{ER} = 1, \text{UR} = 1. \text{PEC} \ is \ a \ predefined \ metal \ name \ with \ the \ default \ values \ and \ cannot \ be \ redefined.

Handling Dielectrics

The Star-Hspice field solver assigns the following default values for dielectrics: \text{CONDUCTIVITY} = 0 \ (\text{lossless} \ \text{dielectric}), \text{LOSSTANGENT} = 0 \ (\text{lossless} \ \text{dielectric}), \text{ER} = 1, \text{UR} = 1. \text{AIR} \ is \ a \ predefined \ dielectric \ name \ with \ default values \ and \ cannot \ be \ redefined.

\textbf{Note:} Because the Star-Hspice field solver does not currently support magnetic materials, it ignores UR values.

Creating Layer Stacks

A layer stack defines a stack of dielectric or/and metal layers. Each transmission line system is associated with exactly one layer stack. But a single-layer stack can be associated with many transmission line systems.

Syntax

\texttt{.LAYERSTACK sname <BACKGROUND=mname> + <LAYER=(mname,thickness) ...>}

- \textit{sname} \ Layer stack name
- \textit{mname} \ Material name
- \textit{BACKGROUND} \ Background dielectric material name. By default, \texttt{AIR} is assumed for the background.
- \textit{thickness} \ Layer thickness
In the layer stack:
- Layers are listed from bottom to top.
- Metal layers (ground planes) are located only at the bottom, top, or both top and bottom.
- Layers are stacked in y-direction, and the bottom of a layer stack is at y=0.
- All conductors must be located above y=0.
- Background material must be dielectric.

Limiting Cases
Free space without ground: `.LAYERSTACK mystack`

Free space with a (bottom) ground plane:
`.LAYERSTACK halfSpace PEC 0.1mm`

Defining Shapes
Use the SHAPE statement to define a shape (used to describe the conductor cross-section).

Syntax
```
.SHAPE sname Shape_Descriptor
```
- `sname`: Shape name.
- `Shape_Descriptor`: See the following subsections.

Defining Rectangles
Use a shape descriptor that defines a rectangle.

Syntax
```
RECTANGLE WIDTH=val HEIGHT=val <NW=val> <NH=val>
```
Using Transmission Lines

Extracting Transmission Line Parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIDTH</td>
<td>Width of rectangle (length in x-direction).</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>Height of rectangle (length in y-direction).</td>
</tr>
<tr>
<td>NW</td>
<td>Number of segments for the width discretization.</td>
</tr>
<tr>
<td>NH</td>
<td>Number of segments for the height discretization.</td>
</tr>
</tbody>
</table>

**Note:** Normally, you do not need to specify the values of NW and NH since they are automatically set by the solver depending on the accuracy mode. But you can specify only one of these values and let the solver determine the other. The discretization parameters of other shape descriptors follow the same convention.

---

**Figure 18-9: Coordinates of a Rectangle**

![Figure 18-9: Coordinates of a Rectangle](image)

**Defining Circles**

Use a shape descriptor that defines a circle. A circle is discretized as an inscribed regular polygon with N edges. Do not use the CIRCLE descriptor to model actual polygons; instead use the POLYGON descriptor.

**Syntax**

CIRCLE RADIUS=val <N=val>
Defining Strips

Shape descriptor that defines an infinitesimally thin strip.

Syntax

```
STRIP WIDTH=val <N=val>
```

- **WIDTH**  
  Width of strip (length in x-direction).
- **N**  
  Number of segments for discretization.

---

**Figure 18-10: Coordinates of a Circle**

![Diagram of a circle with origin, radius, and starting vertex of the inscribed polygon labeled.](image)

**Figure 18-11: Coordinates of a Strip Polygon**

![Diagram of a strip with origin and width labeled.](image)
Use a shape descriptor that defines a polygon. The specified coordinates are in the local coordinate with respect to the origin of a conductor.

Syntax

\[
\text{POLYGON VERTEX}=(x_1 \ y_1 \ x_2 \ y_2 \ ...) \ <N=(n_1,n_2,...)>
\]

**VERTEX**  
(x, y) coordinates of vertices. Listed either in clockwise or counter-clockwise direction.

**N**  
Number of segments for each edges. If only one value is specified, then this value is used for all edges. The first value of N, n1, corresponds to the number of segments for the edge from \((x_1 \ y_1)\) to \((x_2 \ y_2)\).

**Figure 18-12: Coordinates of a Polygon**

![Polygon Coordinates Diagram](image)

**Example**

Rectangular polygon using the default discretization:

\[
\text{POLYGON VERTEX}=(1 \ 10 \ 1 \ 11 \ 5 \ 11 \ 5 \ 10)
\]

Rectangular polygon using five segments for each edge:

\[
\text{POLYGON VERTEX}=(1 \ 10 \ 1 \ 11 \ 5 \ 11 \ 5 \ 10) \ N=5
\]
Rectangular polygon using the different number of segments for each edge:

\[
\text{POLYGON VERTEX}=(1 \ 10 \ 1 \ 11 \ 5 \ 11 \ 5 \ 10) \ \text{N}=(5 \ 3 \ 5 \ 3)
\]

**Field-Solver Options**

Use the `FSOPTIONS` statement to set various options for the solver.

**Syntax**

```
.FSOPTIONS name <ACCURACY=LOW|MEDIUM|HIGH>
+ <GRIDFACTOR=val> <PRINTDATA=YES|NO> <COMPUTEG0=YES|NO>
+ <COMPUTEGD=YES|NO> <COMPUTERO=YES|NO> <COMPUTERS=YES|NO>
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Option name.</td>
<td></td>
</tr>
<tr>
<td>ACCURACY</td>
<td>Sets the solver accuracy to either LOW, MEDIUM, or HIGH.</td>
<td>HIGH</td>
</tr>
<tr>
<td>GRIDFACTOR</td>
<td>Multiplication factor (integer) to determine the final number of segments used in discretization.</td>
<td>1</td>
</tr>
<tr>
<td>PRINTDATA</td>
<td>Specifies that the solver prints output matrices.</td>
<td>NO</td>
</tr>
<tr>
<td>COMPUTEG0</td>
<td>Specifies that the solver computes the static conductance matrix.</td>
<td>YES</td>
</tr>
<tr>
<td>COMPUTEGD</td>
<td>Specifies that the solver computes the dielectric loss matrix.</td>
<td>NO</td>
</tr>
<tr>
<td>COMPUTERO</td>
<td>Specifies that the solver computes the DC resistance matrix.</td>
<td>YES</td>
</tr>
<tr>
<td>COMPUTERS</td>
<td>Specifies that the solver computes the skin-effect resistance matrix.</td>
<td>NO</td>
</tr>
</tbody>
</table>

The **L** and **C** matrices are always computed.
For each accuracy mode, the solver uses either the predefined number of segments or the user-specified value for discretization. It then multiplies this number by the \texttt{GRIDFACTOR} to obtain the final number of segments.

\textbf{Note:} Because there is a wide range of Star-Hspice applications, the predefined accuracy level might not be good enough for certain applications. If you need a higher accuracy than the value set by the \texttt{HIGH} option, increase the \texttt{GRIDFACTOR} value.

\section*{Using the Field Solver Model}

Use the field-solver model to specify a W Element transmission-line geometry model.

In the field-solver model:

- The list of conductors must appear last.
- Conductors cannot overlap each other.
- Floating conductors are assumed to be electrically disconnected, and nonzero fixed charge is not supported.
- Metal layers in the layer stack are treated as the reference node.
- Conductors defined as \texttt{REFERENCE} are all electrically connected and correspond to the reference node in W Element.
- Signal conductors must be ordered according to the terminal list in the W Element statement; for instance, the \textit{i}th signal conductor, counting without reference and floating conductors, is associated with the \textit{i}th input and output terminals specified in the corresponding W Element. Floating and reference conductors can appear in any order.

\section*{Syntax}

\begin{verbatim}
.MODEL mname W MODELTYPE=FieldSolver LAYERSTACK=name + <FSOPTIONS=name> <RLGCFILE=name> + <OUTPUTFORMAT=RLGC|RLGCFILE> + CONDUCTOR=( SHAPE=name <MATERIAL=name> <ORIGIN=(x,y)> + <TYPE=SIGNAL|REFERENCE|FLOATING> ) ...
\end{verbatim}
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mname</td>
<td>Model name</td>
</tr>
<tr>
<td>LAYERSTACK</td>
<td>Associated layer stack name</td>
</tr>
<tr>
<td>FSOPTIONS</td>
<td>Associated option name. If this entry is not specified, the default options are used.</td>
</tr>
<tr>
<td>RLGCFIELD</td>
<td>Specifies the output file for RLG matrices instead of the standard error output device. In case the specified file already exists, the output is simply appended. PRINTDATA in .FSOPTIONS must set to YES (which is default) to get any output.</td>
</tr>
<tr>
<td>OUTPUTFORMAT</td>
<td>Specifies the W Element model syntax format for RLG matrices in RLGCFIELD. The default format is a RLG model.</td>
</tr>
<tr>
<td>SHAPE</td>
<td>Shape name</td>
</tr>
<tr>
<td>x y</td>
<td>The coordinate of the local origin.</td>
</tr>
<tr>
<td>MATERIAL</td>
<td>Conductor material name. If this entry is not specified, PEC is assumed.</td>
</tr>
<tr>
<td>TYPE</td>
<td>One of the following conductor types:</td>
</tr>
<tr>
<td></td>
<td>SIGNAL: a signal node in W Element.</td>
</tr>
<tr>
<td></td>
<td>REFERENCE: the reference node in W Element.</td>
</tr>
<tr>
<td></td>
<td>FLOATING: floating conductor, no reference to W Element. The default value of TYPE is SIGNAL.</td>
</tr>
</tbody>
</table>
Field Solver Examples

The following example shows you how to use the Star-Hspice field solver. All the examples shown in this section are run with the HIGH accuracy mode and GRIDFACTOR = 1.

Example 1: Cylindrical Conductor Above a Ground Plane

In the first example, consider a copper cylindrical conductor above an ideal (lossless) ground plane. Figure 18-13 shows the geometry. Table 18-4 lists the corresponding Star-Hspice netlist.

In this case, you can derive the exact analytical formulas for all of the transmission line parameters:\(^1\)

\[
C = \frac{2\pi \varepsilon}{\text{acosh}\left(\frac{2H}{d}\right)}
\]

\[\sigma_c = 5.76e7\]
\[\varepsilon_r = 4.0\]
\[d = 1\, \text{mm}\]
\[H = 3\, \text{mm}\]
\[\tan \delta = 1.2e-3\]

Figure 18-13: Cylindrical Conductor Above a Ground Plane

---

Table 18-4: Input File Listing for Example 1

| Header, options and sources | *Example 1: cylindrical conductor
.OPTION PROBE POST
VIMPULSE in1 gnd PULSE 4.82v 0v 5n 0.5n 0.5n 25n |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>W Element</td>
<td>W1 in1 gnd out1 gnd FSmodel=cir_trans N=1 l=0.5</td>
</tr>
</tbody>
</table>
| Materials                   | .MATERIAL diel_1 DIELECTRIC ER=4, + LOSSTANGENT=1.2e-3
.MATERIAL copper METAL + CONDUCTIVITY=57.6meg |
| Shapes                      | .SHAPE circle_1 CIRCLE RADIUS=0.5mm               |
| Defines a half-space        | .LAYERSTACK halfSpace BACKGROUND=diel_1, + LAYER=(PEC,1mm) |
| Option settings             | .FSOPTIONS opt1 PRINTDATA=YES, + COMPUTERS=yes, COMPUTEGD=yes |
| Model definition            | .MODEL cir_trans W MODELTYPE=FieldSolver + LAYERSTACK=halfSpace, FSOPTIONS=opt1, + RLGCFILE=ex1.rlgc + CONDUCTOR=(SHAPE=circle_1, + ORIGIN=(0,4mm), MATERIAL=copper) |
| Analysis, outputs and end   | .TRAN 0.5n 100n
.PROBE v(out1)
.END |

\[
L = \frac{1}{\mu \varepsilon} C^{-1}
\]

\[
G = \frac{\sigma_d}{\varepsilon} C = \omega \cdot \tan(\delta) \cdot C
\]

\[
R = \frac{1}{\sigma_c \pi d} \left[ \frac{2H/d}{\sqrt{(2H/d)^2 - 1}} \right] = \frac{1}{\sigma_c \pi d} \left[ \frac{\pi \mu}{2H/d} - 1 \right]
\]

Compare the computed results with the analytical solutions in Table 18-5. The resistance and conductance are computed at the frequency of 200 MHz, and the
DC resistance (R₀) and conductance (G₀) are not included in the computed values.

**Table 18-5: Comparison Result for Example 1**

<table>
<thead>
<tr>
<th>Value</th>
<th>Exact</th>
<th>Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>C (pF/m)</td>
<td>89.81</td>
<td>89.66</td>
</tr>
<tr>
<td>L (nH/m)</td>
<td>494.9</td>
<td>495.7</td>
</tr>
<tr>
<td>G (mS/m)</td>
<td>0.1354</td>
<td>0.1352</td>
</tr>
<tr>
<td>R (Ω/\text{m})</td>
<td>1.194</td>
<td>1.178</td>
</tr>
</tbody>
</table>

**Example 2: Stratified Dielectric Media**

**Figure 18-14: Three Traces Immersed in Stratified Dielectric Media**

Figure 18-14 shows an example of three traces immersed in stratified dielectric media. The input file listing is shown in Table 18-6.

Table 18-7 compares the computed capacitance matrix with results from two other numerical methods.
Table 18-6: Input File Listing for Example 2

| Header, options and sources                          | *Example 2, three traces in dielectric
+ VIMPULSE in1 gnd PULSE 4.82v 0v 5n 0.5n 0.5n + 25n |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>W Element</td>
<td>W1 in1 in2 in3 gnd out1 out2 out3 gnd + FSmode=cond3_sys N=3 l=0.5</td>
</tr>
</tbody>
</table>
| Materials                                            | .MATERIAL die1_1 DIELECTRIC ER=4.3
+ .MATERIAL die1_2 DIELECTRIC ER=3.2 |
| Shapes                                               | .SHAPE rect_1 RECTANGLE WIDTH=0.35mm, + HEIGHT=0.07mm |
| Uses the default AIR background                       | .LAYERSTACK stack_1
+ LAYER=(PEC,1um),LAYER=(die1_1,0.2mm),
+ LAYER=(die1_2,0.1mm) |
| Option settings                                      | .FSOPTIONS opt1 PRINTDATA=YES |
| Three conductors share the same shape                | .MODEL cond3_sys W MODELTYPE=FieldSolver,
+ LAYERSTACK=stack1, FSOPTIONS=opt1,
+ RLGCFILE=ex2.rlgc
+ CONDUCTOR=(SHAPE=rect_1,ORIGIN=
+ (0,0.201mm)),
+ CONDUCTOR=(SHAPE=rect_1,
+ ORIGIN=(0.5mm,0.301mm)),
+ CONDUCTOR=(SHAPE=rect_1,ORIGIN=
+ (1mm,0.301mm)) |
| Analysis, outputs and end                            | .TRAN 0.5n 100n
+ PROBE v(out1)
+ END |
Table 18-7: Comparison Result for Example 2

<table>
<thead>
<tr>
<th></th>
<th>Computed</th>
<th>Raphael (Finite-Difference Solver)</th>
<th>Reference\textsuperscript{a}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\begin{bmatrix} 141.1 &amp; -21.36 &amp; -0.90 \ -21.36 &amp; 92.66 &amp; -17.72 \ -0.90 &amp; -17.72 &amp; 87.26 \end{bmatrix}$ (pF/m)</td>
<td>$\begin{bmatrix} 139.5 &amp; -23.46 &amp; -1.89 \ -23.69 &amp; 94.60 &amp; -19.89 \ -1.82 &amp; -19.52 &amp; 85.48 \end{bmatrix}$ (pF/m)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\begin{bmatrix} 142.1 &amp; -21.76 &amp; -0.89 \ -21.73 &amp; 93.53 &amp; -18.10 \ -0.89 &amp; -18.10 &amp; 87.96 \end{bmatrix}$ (pF/m)</td>
<td></td>
</tr>
</tbody>
</table>


Figure 18-15 shows the results of convergence analysis performed based on the total capacitance of the first conductor with respect to the \texttt{GRIDFACTOR} parameter.
Example 3: Two Traces Between Two Ground Planes

The following example uses the coupled strip line case shown in Figure 18-16. Table 18-8 lists the complete input netlist. Table 18-9 shows the comparison between the computed result and the Finite Element (FEM) solver result.
Table 18-8: Input File Listing for Example 3

| Header, options and sources                  | *Example 3: two traces between gnd planes  
 |                                          | .OPTION PROBE POST  
 |                                          | + IMPULSE in1 gnd PULSE 4.82v 0v 5n 0.5n 0.5n  
 |                                          | + 25n  
 | W Element                                  | W1 in1 in2 gnd out1 out2 gnd FSmodel=cond2_sys  
 |                                          | +N=2 l=0.5  
 | Materials                                  | .MATERIAL diel_1 DIELECTRIC ER=10.0  
 |                                          | .MATERIAL diel_2 DIELECTRIC ER=2.5  
 | Shapes                                     | .SHAPE rect RECTANGLE WIDTH=1mm,  
 |                                          | + HEIGHT=0.2mm,  
 | Top and bottom ground planes               | .LAYERSTACK stack_1,  
 |                                          | + LAYER=(PEC,1mm), LAYER=(diel_1,2mm),  
 |                                          | + LAYER=(diel_2,3mm), LAYER=(PEC,1mm)  
 | Option settings                            | .FSOPTIONS opt1 PRINTDATA=YES  
 | Two conductors share the same shape        | .MODEL cond2_sys W MODELTYPE=FieldSolver,  
 |                                          | + LAYERSTACK=stack1, FSOPTIONS=opt1  
 |                                          | + RLGCFILE=ex3.rlgc  
 |                                          | + CONDUCTOR=(SHAPE=rect, ORIGIN=  
 |                                          | + (0,3mm)),  
 |                                          | + CONDUCTOR=(SHAPE=rect,  
 |                                          | + ORIGIN=(1.2mm,3mm))  
 | Analysis, outputs and end                  | .TRAN 0.5n 100n  
 |                                          | .PROBE v(out1)  
 |                                          | .END  

Table 18-9: Comparison Result for Example 3

| Computed          | 214.1  -105.2  
 |                  | -105.2  214.1  (pF/m)  
 | FEM Solver       | 217.7  -108.2  
 |                  | -108.2  217.7  (pF/m)  

Example 4: Using Field Solver with Monte Carlo Analysis

The following example shows how to perform transient analysis using Monte Carlo analysis to model variations in the manufacturing of the microstrip. Table 18-10 shows the Star-Hspice input listing with the W Element. Figure 18-17 shows the transient output waveforms.

Figure 18-17: Monte Carlo Analysis with a Field Solver and W Element

![Monte Carlo Analysis with a Field Solver and W Element](image)

Table 18-10: Input File Listing for Example 4

| Header, options and sources | *PETL Example 4: example 2 with Monte-Carlo
.OPTION PROBE POST + VIMPULSE in1 gnd AC=1v PULSE 4.82v 0v 5ns + 0.5ns 0.5ns 25ns |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter definitions</td>
<td>.PARAM x1=Gauss(0,0.02,1) + x2=Gauss(0.5mm,0.02,1) x3=Gauss(1mm,0.02,1) .PARAM dRef=1u dY1=Gauss(2mm,0.02,1) + dY2=Gauss(1mm,0.02,1)</td>
</tr>
<tr>
<td>W Element</td>
<td>W1 in1 in2 in3 0 out1 out2 out3 0 + FSMODEL=cond3_sys N=3 l=0.5</td>
</tr>
</tbody>
</table>
### Materials

```
.MATERIAL die1_1 DIELECTRIC ER=4.3
.MATERIAL die1_2 DIELECTRIC ER=3.2
```

### Shapes

```
.SHAPE r1 RECTANGLE WIDTH=0.35mm,
+ HEIGHT=0.070mm
```

### Uses the default AIR background

```
.LAYERSTACK stack_1
+ LAYER= (PEC,dRef),LAYER=(die1_1,dY1),
+ LAYER= (die1_2,dY2)
```

### Three conductors share the same shape

```
.MODEL cond3_sys W MODELTYPE=FieldSolver,
+ LAYERSTACK=stack1,
+ CONDUCTOR=(SHAPE=r1,ORIGIN=
+ (x1,'dRef+dY1')),
+ CONDUCTOR=(SHAPE=r1,ORIGIN=
+ (x2,'dRef+dY1+dY2')),
+ CONDUCTOR=(SHAPE=r1,ORIGIN=
+ (x3,'dRef+dY1+dY2'))
```

### Analysis, outputs and end

```
.PROBE TRAN v(in1) v(out1) v(in3)
.PROBE AC v(out1) v(out3)
.PROBE DC v(in1) v(out1) v(out3)
.AC LIN 200 0Hz 0.3GHz
.DC v1 0v 5v 0.01v
.TRAN 0.5ns 100ns SWEEP MONTE=3
.END
```
Chapter 19

Using IBIS Models

The Input/Output Buffer Information Specification (IBIS) is being developed by the IBIS Open Forum, which is affiliated with the Electronic Industries Alliance (EIA). IBIS specifies a standard form for presentation of information in ASCII format in special files. This information describes behavior of various I/O buffers that send electrical signals outside the silicon chip or receive such signals. The type of information includes output I-V curves for output buffers in LOW and HIGH states, V(t) curves describing the exact form of transitions from LOW to HIGH states and from HIGH to LOW states for a given load, values for die capacitance, electrical parameters of the packages, and so on. The IBIS standard only specifies the “form” of information and does not specify how the information should be processed or used by the simulator.

However, the IBIS standard contains a section devoted to recommendations on how information should be derived through the simulation or from the silicon measurement. In addition, the IBIS Open Forum has sponsored development of a parser for IBIS files—called the golden parser. The golden parser is freely available as an executable and should be used for verification of IBIS files. The golden parser is incorporated into Hspice. When the golden parser processes an IBIS file, it produces warnings and/or error messages which by default appear in the Hspice output.

Star-Hspice implements I/O buffers as a new element type, which we refer to as a buffer. The name of this element starts with the letter b. Using buffers is similar to using other Star-Hspice elements, such as transistors: give a name to the buffer, specify a list of nodes that are used to connect the buffer to the rest of the circuit, and specify parameters. Only parameters that specify a model for the buffer (file name and model name) are required.
Two significant differences from the use of other elements are: (1) number of external nodes is variable depending on the buffer type and can be from 4 to 8; and (2) nodes that are suppose to connect to power/ground rails must not be connected in the netlist because Star-Hspice does this connection by default.

This chapter is not intended to introduce the IBIS standard because it is a large document; familiarity with the standard is assumed. A significant amount of information is available on the internet and appropriate links to other sites are given in “References” on page 19-49.

Three types of analysis are supported for input/output buffers:

- DC analysis
- Transient analysis
- AC analysis

This chapter covers the following topics:

- Understanding IBIS Conventions
- Buffers
- Specifying Common Keywords
- Differential Pins
- Scaling Buffer Strength
- Example
- Additional Notes
- Understanding Warning and Error Messages
- References
Understanding IBIS Conventions

The general syntax of an element card for I/O buffers is:

**General Form:**

```
bname node_1 node_2 ... node_N
+ keyword_1=value_1 ... [keyword_M=value_M]
```

where:

- **bname**: Buffer element name. Must begin with B, which can be followed by up to 1023 alphanumerical characters.
- **node_1 node_2 ... node_N**: List of I/O buffer external nodes. The number of nodes and corresponding rules are specific to different buffer types (see later sections in this chapter).
- **keyword_i=value_i**: Assigns value, \textit{value}_i, to the keyword, \textit{keyword}_i. Optional keywords are given in square brackets (see “Specifying Common Keywords” on page 19-26 for more information).

![Circuit Diagram for Package](image)

The \textit{gnd} node on the circuit diagram for buffers denotes the ideal SPICE ground node (the notation \textit{node 0} [zero] is also used). This node is always available to Star-Hspice. Do not include this node in the node list on the buffer card. If the \textit{gnd} node appears on a circuit diagram, Star-Hspice makes the node connection
to the ideal ground. Node gnd is used on circuit diagrams to explain the connection of individual parts inside buffers.

In some cases, buffer nodes have different rules than nodes for other Star-Hspice elements. Some nodes may already be connected to voltage sources (Star-Hspice makes such connections) so it is incorrect to connect a voltage source to such nodes. Conversely, some nodes should be connected to voltage sources and it is incorrect not to connect voltage sources to these nodes.

---

**Note:** See “Specifying Common Keywords” on page 19-26 and the following sections on individual buffer types for detailed explanations on the use of these nodes.

---

Buffers correspond to models in IBIS files and do not include packages. At this time, corresponding packages should be added manually. For example, if node_out and node_pin are nodes for output of the output buffer and corresponding pin, then add the following lines to the netlist:

\[
\begin{align*}
\mathrm{R}_{\mathrm{pkg}} & \quad \mathrm{node}_{\mathrm{out}} & \quad \mathrm{node}_{\mathrm{pkg}} & \quad \mathrm{R}_{\mathrm{pkg}} \_value \\
\mathrm{L}_{\mathrm{pkg}} & \quad \mathrm{node}_{\mathrm{pkg}} & \quad \mathrm{node}_{\mathrm{pin}} & \quad \mathrm{L}_{\mathrm{pkg}} \_value \\
\mathrm{C}_{\mathrm{pkg}} & \quad \mathrm{node}_{\mathrm{pin}} & \quad \mathrm{gnd} & \quad \mathrm{C}_{\mathrm{pkg}} \_value
\end{align*}
\]

where values for \( \mathrm{R}_{\mathrm{pkg}}, \ \mathrm{L}_{\mathrm{pkg}}, \) and \( \mathrm{C}_{\mathrm{pkg}} \) are taken from the IBIS file (see Figure 19-1 for the circuit diagram).

**Terminology**

The following are definitions of terms used frequently in this chapter:

- **card, buffer card**: Used to denote a line(s) from the netlist that specifies the buffer name (should begin with the letter b), a list of external nodes, required keyword, and optional keywords.
- **buffer, I/O buffer, input/output buffer**: One of 14 IBIS models as specified in the standard, version 3.2, and implemented in Star-Hspice.
- **RWF, FWF**: Rising waveform, falling waveform
Limitations and Restrictions

The series, series switch, and terminator buffers are not implemented in the 2000.4 version of Star-Hspice.

You can simulate the terminator by using other existing Star-Hspice elements: resistors, capacitors, and voltage dependent current sources.
Buffers

This section describes the buffers as implemented in Star-Hspice. Please refer to “Specifying Common Keywords” on page 19-26 for details on using keywords shown in the syntax examples in the following sections.

Input Buffer

The syntax of an input buffer element card is:

\[ \text{B\_INPUT nd\_pc nd\_gc nd\_in nd\_out\_of\_in} \]
\[ \quad + \text{file='filename' model='model\_name'} \]
\[ \quad + \left[ \text{typ={typ|min|max|fast|slow}} \right] \]
\[ \quad + \left[ \text{power={on|off}} \right] \]
\[ \quad + \left[ \text{buffer={1|input}} \right] \]
\[ \quad + \left[ \text{interpol={1|2}} \right] \]
\[ \quad + \left[ \text{nowarn} \right] \]
\[ \quad + \left[ \text{c\_com\_pc=c\_com\_pc\_value} \right] \]
\[ \quad + \left[ \text{c\_com\_gc=c\_com\_gc\_value} \right] \]
\[ \quad + \left[ \text{pc\_scal=pc\_scal\_value} \right] \]
\[ \quad + \left[ \text{gc\_scal=gc\_scal\_value} \right] \]

where the total number of external nodes is equal to 4.

If keyword \text{power=on} (default) is specified, nodes \text{nd\_pc} and \text{nd\_gc} are connected to voltage sources with values taken from the IBIS file. You should not connect these nodes to voltage sources. However, names for these nodes should be provided, so you can print out the voltage values if required. For example:

\[ \text{.PRINT V(nd\_pc) V(nd\_gc)} \]

If keyword \text{power=off} is specified, Star-Hspice does not connect these nodes to voltage sources. You should connect the nodes to voltage sources either directly or through an RLC network, or a transmission line.

There are no special rules for \text{node\_in} and \text{node\_in} can connect to I, E, F, G, and H elements. The buffer measures and processes the voltage on this node and sends a response to node \text{nd\_out\_of\_in}. The node \text{nd\_out\_of\_in} is connected to the voltage source as shown in Figure 19-2. It is an error to connect this node to
a voltage source. If power=off, nodes nd_pc, nd_gc can be connected to the ground is the intention is to specify voltage zero on these nodes.

**Figure 19-2: Input Buffer**

![Input Buffer Diagram]

$V_{out\_of\_in}$ is a digital signal that assumes values of either 0 or 1 depending on the voltages $V_{in}$, $V_{inh}$, $V_{inl}$, and Polarity. Star-Hspice processes $V_{out\_of\_in}$ according to the following rules.

If:

- **Polarity=Non-Inverting**
  - Initially $V_{out\_of\_in}$ is set to 0 if $V_{in} < (V_{inh}+V_{inl})/2$ and to 1 in the opposite case.
  - and if $V_{out\_of\_in}=1$ then it goes to 0 only if $V_{in} < V_{inl}$
  - and if $V_{out\_of\_in}=0$ then it goes to 1 only if $V_{in}>V_{inh}$

- **Polarity=Inverting**
  - Initially $V_{out\_of\_in}$ is set to 0 if $V_{in} > (V_{inh}+V_{inl})/2$ and to 1 in the opposite case.
  - and if $V_{out\_of\_in}=1$ then it goes to 0 only if $V_{in} > V_{inh}$
  - and if $V_{out\_of\_in}=0$ then it goes to 1 only if $V_{in}<V_{inl}$
Figure 19-2 shows a single circuit specified on a single element card. 
$V_{out\_of\_in}$ is a voltage source whose value is a function of $V_{in}$ (as well as of thresholds $V_{inl}$, $V_{inh}$, and parameter Polarity). It can be used to drive other circuits.

If pc_scal or gc_scal arguments exist and pc_scal_value or gc_scal_value do not equal 1.0, then PC or GC iv curve will be adjusted using the pc_scal_value or gc_scal_value.

**Output Buffer**

The syntax for an output buffer element card is:

```plaintext
B_OUTPUT nd_pu nd_pd nd_out nd_in [nd_pc nd_gc]
+ file='file_name' model='model_name'
+ [typ={typ|min|max|fast|slow}] [power={on|off}]
+ [buffer={2|output}]
+ [xv_pu=state_pu] [xv_pd=state_pd]
+ [interpol={1|2}]
+ [ramp_fwf={0|1|2}] [ramp_rwf={0|1|2}]
+ [fwf_tune=fwf_tune_value] [rfw_tune=rwf_tune_value]
+ [nowarn]
+ [c_com_pu=c_com_pu_value]
+ [c_com_pd=c_com_pd_value]
+ [c_com_pc=c_com_pc_value]
+ [c_com_gc=c_com_gc_value]
+ [pu_scal=pu_scal_value]
+ [pd_scal=pd_scal_value]
+ [pc_scal=pc_scal_value]
+ [gc_scal=gc_scal_value]
+ [rfw_scal=rfw_scal_value]
+ [fwf_scal=fwf_scal_value]
+ [spu_scal=spu_scal_value]
+ [spd_scal=spd_scal_value]
```

Nodes $nd\_pc$ and $nd\_gc$ are optional. However, either both or none can be specified. The total number of external nodes is either 4 or 6, any other number is an error. If nodes $nd\_pc$ and $nd\_gc$ are not given on the element card but
Power_Clamp and/or Ground_Clamp I-V curves are present in the model in question, then the simulator will simply add Power_Clamp and/or Ground_Clamp I-V curves data to corresponding Pull_Up and/or Pull_Down I-V curves data.

However, the optional nodes \textit{nd\_pc} and \textit{nd\_gc} are needed if:

- IBIS keywords \textit{POWER Clamp Reference} and \textit{GND Clamp Reference} are present in the IBIS model and have different values than the IBIS keywords \textit{Pullup Reference} and \textit{Pulldown Reference}, or
- IBIS keywords \textit{Pullup Reference} and \textit{Pulldown Reference} do not exist and \textit{POWER Clamp Reference} and \textit{GND Clamp Reference} have different values than those determined by the \textit{Voltage Range} IBIS keyword.

If optional nodes \textit{nd\_pc} and \textit{nd\_gc} are needed, but omitted from the element card, Star-Hspice issues a warning and connects \textit{nd\_pc} to \textit{nd\_pu} and \textit{nd\_gc} to \textit{nd\_pd}.

\textbf{Figure 19-3: Output Buffer}

If keyword \textit{power=on} (default) is specified, nodes \textit{nd\_pu}, \textit{nd\_pd}, and if specified, \textit{nd\_pc} and \textit{nd\_gc}, are connected to voltage sources with values taken from the IBIS file. You should not connect these nodes to voltage sources. However, names for these nodes should be provided, so you can print out the voltage values if required. For example:

\texttt{.PRINT V(nd\_pu) V(nd\_pd)}
If keyword `power=off` is specified, Star-Hspice does not connect these nodes to voltage sources. You should connect the nodes to voltage sources either directly or through an RLC network or a transmission line.

There are no special rules for node `nd_out`. The voltage on this node is controlled by the digital signal on the node `nd_in`. Now any voltage source, current source, voltage controlled voltage source, voltage controlled current source, current controlled voltage source, or current controlled current source can be connected to the `nd_in` as shown in the following example:

```
V_in nd_in gnd 0V pulse( 0V 1V 1n 0.1n 0.1n 7.5n 15n )
```

If `power=off`, nodes `nd_pu`, `nd_pd`, `nd_pc`, `nd_gc` can be connected to the ground if the intention is to have zero voltage on these nodes.

`V_in` is a controlling signal representing a digital signal with values 0 and 1. However, Star-Hspice will take any signal and process according to the following rules:

If:

- **Polarity=Non-Inverting**

  At \( t=0 \) for transient analysis (or for DC analysis), the buffer goes to HIGH state if \( V_{in} > 0.5 \) and to LOW in the opposite case.

  Next, if the buffer is in HIGH state, it will go to LOW state if \( V_{in} < 0.2 \). If the buffer is in LOW state, it will go to HIGH state if \( V_{in} > 0.8 \).

- **Polarity=Inverting**

  At \( t=0 \) for transient analysis (or for DC analysis), the buffer goes to HIGH state if \( V_{in} < 0.5 \) and to LOW in the opposite case.

  Next, if the buffer is in HIGH state, it will go to LOW state if \( V_{in} > 0.8 \). If the buffer is in LOW state, it will go to HIGH state if \( V_{in} < 0.2 \).

If `pc_scal` (or `gc_scal`, `pu_scal`, `pd_scal`) argument exists and `pc_scal_value` (or `gc_scal_value`, `pu_scal_value`, `pd_scal_value`) does not equal to 1.0, the PC (or
GC, PU, PD) iv curve will be adjusted using the pc_scal_value (or gc_scal_value, pu_scal_value, pd_scal_value).

If rwf_scal (or fwt_scal) argument exists and rwf_scal_value (or fwt_scal_value) does not equal to 1.0, rising and falling vt curves will be adjusted using rwf_scal_value (or fwt_scal_value).

If spu_scal (or spd_scal) argument exists and spu_scal_value (or spd_scal_value) does not equal to 0.0, but at the same time power is equal to off and (spu_scal_value - spd_scal_value) does not equal to the corresponding value in the .ibs file, then the iv curves of PU (or PD) will be adjusted using spu_scal_value (or spd_scal_value).

**Tristate Buffer**

The syntax for a tristate buffer element card is:

```plaintext
B_3STATE nd_pu nd_pd nd_out nd_in nd_en [nd_pc nd_gc]
+ file='file_name' model='model_name'
+ [typ={typ|min|max|fast|slow}] [power={on|off}]
+ [buffer={4|three_state}]
+ [xv_pu=state_pu] [xv_pd=state_pd]
+ [interpol={1|2}]
+ [ramp_fwf={0|1|2}] [ramp_rwf={0|1|2}]
+ [fwt_tune=fwt_tune_value] [rwf_tune=rwf_tune_value]
+ [nowarn]
+ [c_com_pu=c_com_pu_value]
+ [c_com_pd=c_com_pd_value]
+ [c_com_pc=c_com_pc_value]
+ [c_com_gc=c_com_gc_value]
+ [pu_scal=pu_scal_value]
+ [pd_scal=pd_scal_value]
+ [pc_scal=pc_scal_value]
+ [gc Scalia=gc_scal_value]
+ [rwf_scal=rwf_scal_value]
+ [fwt_scal=fwt_scal_value]
+ [spu_scal=spu_scal_value]
+ [spd skal=spd_scal_value]
```
Buffers

Using IBIS Models

Nodes \textit{nd}_{pc} and \textit{nd}_{gc} are optional. However, either both or none can be specified. The total number of external nodes is either 5 or 7; any other number is an error. If nodes \textit{nd}_{pc} and \textit{nd}_{gc} are not given on the element card but Power_Clamp and/or Ground_Clamp I-V curves are present in the model in question, then the simulator will simply add Power_Clamp and/or Ground_Clamp I-V curves data to corresponding Pull_Up and/or Pull_Down I-V curves data.

However, the optional nodes \textit{nd}_{pc} and \textit{nd}_{gc} are needed if:

- IBIS keywords \textit{POWER Clamp Reference} and \textit{GND Clamp Reference} are present in the IBIS model and have different values than the IBIS keywords \textit{Pullup Reference} and \textit{Pulldown Reference}, or
- IBIS keywords \textit{Pullup Reference} and \textit{Pulldown Reference} do not exist and \textit{POWER Clamp Reference} and \textit{GND Clamp Reference} have different values than those determined by the \textit{Voltage Range} IBIS keyword.

If optional nodes \textit{nd}_{pc} and \textit{nd}_{gc} are needed, but omitted from the element card, Star-Hspice issues a warning and connects \textit{nd}_{pc} to \textit{nd}_{pu} and \textit{nd}_{gc} to \textit{nd}_{pd}.

\textbf{Figure 19-4: Tristate Buffer}
If keyword \texttt{power=on} (default) is specified, nodes \texttt{nd\_pu}, \texttt{nd\_pd}, and if specified, \texttt{nd\_pc} and \texttt{nd\_gc} are connected to voltage sources with values taken from the IBIS file. You should not connect these nodes to voltage sources.

However, names for these nodes should be provided in the netlist, so you can print out the voltage values if required. For example:

\begin{verbatim}
  .PRINT V(nd\_pu) V(nd\_pd)
\end{verbatim}

If keyword \texttt{power=off} is specified, Star-Hspice does not connect these nodes to voltage sources. You should connect the nodes to voltage sources either directly or through an RLC network, or a transmission line.

There are no special rules for \texttt{nd\_out}. The voltage on this node is controlled by the digital signal on the nodes \texttt{nd\_in}, \texttt{nd\_en}. Voltage sources must be connected to the nodes \texttt{nd\_in}, \texttt{nd\_en} as shown in the following example:

\begin{verbatim}
  V\_in nd\_in gnd 0V pulse( 0V 1V 1n 0.1n 0.1n 7.5n 15n )
  V\_en nd\_en gnd 0V pulse( 0V 1V 3n 0.1n 0.1n 7.5n 15n )
\end{verbatim}

Nodes \texttt{nd\_pu}, \texttt{nd\_pd}, \texttt{nd\_pc}, and \texttt{nd\_gc} can be connected to the ground if the intention is to have zero voltage on these nodes. Nodes \texttt{nd\_in}, \texttt{nd\_en} can not be connected to the ground.

\texttt{V\_in} and \texttt{V\_en} are controlling signals representing digital signals with values 0 and 1. Star-Hspice will take any signal and process according to the following rules:

The enable signal, \texttt{V\_en}, supersedes the input signal, \texttt{V\_in}.

If:

\begin{itemize}
  \item \textit{ENABLE = Active-High} \quad At t=0 for transient analysis (or for DC analysis), the buffer goes to the ENABLE state if \texttt{V\_en} > 0.5 and to DISABLE in the opposite case.
  \item \textit{ENABLE = Active-Low} \quad At t=0 for transient analysis (or for DC analysis), the buffer goes to ENABLE state if \texttt{V\_en} < 0.5 and to DISABLE in the opposite case.
\end{itemize}
Buffers

The buffer is in ENABLE state
Begin transition to DISABLE state if V_en < 0.2 (where Enable = Active-High) and if V_en > 0.8 (where Enable = Active-Low).

The buffer is in DISABLE state or in the process of transition from ENABLE state to DISABLE state
Begin transition to ENABLE state if V_en > 0.8 (where Enable = Active-High) and if V_en < 0.2 (where Enable = Active-Low).

The buffer is in ENABLE state
Response to the input signal, V_in, is the same as the output buffer.

Polarity=Non-Inverting
At t=0 for transient analysis (or for DC analysis), the buffer goes to HIGH state if V_in > 0.5 and to LOW in the opposite case.

Next, if the buffer is in HIGH state, it will go to LOW state if V_in < 0.2. If the buffer is in LOW state, it will go to HIGH state if V_in > 0.8.

Polarity=Inverting
At t=0 for transient analysis (or for DC analysis), the buffer goes to HIGH state if V_in < 0.5 and to LOW in the opposite case.

Next, if the buffer is in HIGH state, it will go to LOW state if V_in > 0.8. If the buffer is in LOW state, it will go to HIGH state if V_in < 0.2.

Note: After the buffer begins a transition from ENABLE state to DISABLE state, all memory about previous HIGH/LOW states is lost. If the buffer later goes to the ENABLE state, it compares the controlling signal, V_in, against the threshold 0.5 to decide whether to go to HIGH state or LOW state similar to the time moment t=0, rather than against the thresholds 0.2 and 0.8.
Input/Output Buffer

The syntax of an input/output buffer element card is:

```plaintext
B_IO nd_pu nd_pd nd_out nd_in nd_en V_out_of_in [nd_pc nd_gc]
+ file='file_name' model='model_name'
+ [typ={typ|min|max|fast|slow}] [power={on|off}]
+ [buffer={3|input_output}]
+ [xv_pu=state_pu] [xv_pd=state_pd]
+ [interpol={1|2}]
+ [ramp_fwf={0|1|2}] [ramp_rwf={0|1|2}]
+ [fwf_tune=fwf_tune_value] [rwf_tune=rwf_tune_value]
+ [nowarn]
+ [c_com_pu=c_com_pu_value]
+ [c_com_pd=c_com_pd_value]
+ [c_com_pc=c_com_pc_value]
+ [c_com_gc=c_com_gc_value]
+ [pu_scal=pu_scal_value]
+ [pd_scal=pd_scal_value]
+ [pc_scal=pc_scal_value]
+ [gc_scal=gc_scal_value]
+ [rwf_scal=rwf_scal_value]
+ [fwf_scal=fwf_scal_value]
+ [spu_scal=spu_scal_value]
+ [spd_scal=spd_scal_value]
```

Nodes `nd_pc` and `nd_gc` are optional. However, either both or none can be specified. The total number of external nodes is either 6 or 8; any other number is an error. If nodes `nd_pc` and `nd_gc` are not given on the element card but Power_Clamp and/or Ground_Clamp I-V curves are present in the model in question, then the simulator will simply add Power_Clamp and/or Ground_Clamp I-V curves data to corresponding Pull_Up and/or Pull_Down I-V curves data.

However, the optional nodes `nd_pc` and `nd_gc` are needed if:

- IBIS keywords `POWER Clamp Reference` and `GND Clamp Reference` are present in the IBIS model and have different values than the IBIS keywords `Pullup Reference` and `Pulldown Reference`, or
IBIS keywords *Pullup Reference* and *Pulldown Reference* do not exist and *
*POWER Clamp Reference* and *GND Clamp Reference* have different values than those determined by the *Voltage Range IBIS keyword*.

If optional nodes `nd_pc` and `nd_gc` are needed, but omitted from the element card, Star-Hspice issues a warning and connects `nd_pc` to `nd_pu` and `nd_gc` to `nd_pd`.

**Figure 19-5: Input-Output Buffer**
If keyword power=on (default) is specified, nodes nd_pu and nd_pd, and if specified, nd_pc and nd_gc, are connected to voltage sources with values taken from the IBIS file. You should not connect these nodes to voltage sources. However, names for these nodes should be provided in the netlist, so you can print out the voltage values if required. For example:

```plaintext
  .PRINT V(nd_pu) V(nd_pd)
```

If keyword power=off is specified, Star-Hspice does not connect these nodes to voltage sources. You should connect the nodes to voltage sources either directly or through an RLC network or a transmission line.

There are no special rules for node nd_out. The voltage on this node is controlled by the digital signal on the nodes nd_in, nd_en. Voltage sources must be connected to the nodes nd_in, nd_en as shown in the following example:

```plaintext
  V_in nd_in gnd 0V pulse (0V 1V 1n 0.1n 0.1n 7.5n 15n)
  V_en nd_en gnd 0V pulse (0V 1V 3n 0.1n 0.1n 7.5n 15n).
```

Nodes nd_pu, nd_pd, nd_pc, and nd_gc can be connected to the ground if the intention is to have zero voltage on these nodes.

The node nd_out_of_in is connected to a voltage source (see Figure 19-5). It is an error to connect this node to a voltage source or the ground.

The input-output buffer is a combination of the tristate buffer and the input buffer. See “Input Buffer” on page 19-6 and “Tristate Buffer” on page 19-11 for more information.

The input-output buffer can function as an input buffer. In this case, the resultant digital signal V_out_of_in on the node nd_out_of_in is controlled by the voltage V_out on the node nd_out.

For the input buffer, this controlling voltage is called V_in and any corresponding node is nd_in.

The input-output buffer uses V_in, nd_in notations to denote the controlling voltage and controlling input node for the output part of the buffer.

If the input-output buffer is not in the DISABLE state (this includes ENABLE state and transitions to ENABLE->DISABLE and DISABLE->ENABLE), then it functions as a tristate buffer. If input-output buffer is in the DISABLE state, it functions as an input buffer.
However, there is a difference in the digital output of the input part of the buffer (voltage $V_{\text{out-of-in}}$). Because $V_{\text{out-of-in}}$ is not always defined (e.g. the buffer is in ENABLE state, or $V_{\text{inl}} < V_{\text{out}} < V_{\text{inh}}$ at the time moment, when the transition to DISABLE state is completed) and because we want to preserve logical LEVELs 0 and 1 for LOW and HIGH states, $V_{\text{out-of-in}}$ takes the value 0.5 when it is undefined.

Figure 19-5 shows a single circuit specified on a single element card. The $V_{\text{out-of-in}}$ is a voltage source whose value is a function of $V_{\text{out}}$ (as well as of thresholds $V_{\text{inl}}$, $V_{\text{inh}}$ and parameter Polarity). It can be used to drive other circuits.

Open Drain, Open Sink, Open Source Buffers

Open drain and open sink buffers do not have pullup circuitry. Open source buffers do not have pulldown circuitry. However, the element cards for these three buffers coincide with the element card for the output buffer. Accordingly, you should always specify names for pullup and pulldown nodes, $nd_{\text{pu}}$ and $nd_{\text{pd}}$, even if the buffer does not have pullup or pulldown circuitry.

All rules given in “Output Buffer” on page 19-8 apply to open drain, open sink, and open source buffers with the following exceptions:

- Because open drain and open sink buffers do not have pullup circuitry, the option $xv_{\text{pu}}=nd_{\text{state\_pu}}$ should not be specified.
- Similarly, because open source buffers do not have pulldown circuitry, the option $xv_{\text{pd}}=nd_{\text{state\_pd}}$ should not be specified.

I/O Open Drain, I/O Open Sink, I/O Open Source Buffers

I/O open drain and I/O open sink buffers do not have pullup circuitry. I/O open source buffers do not have pulldown circuitry. However, the element cards for these three buffers coincide with the element card for the input-output buffer. Accordingly, you should always specify names for pullup and pulldown nodes, $nd_{\text{pu}}$ and $nd_{\text{pd}}$, even if the buffer does not have pullup or pulldown circuitry.
All rules given in “Input/Output Buffer” on page 19-15 are applicable to I/O open drain, I/O open sink, and I/O open source buffers with the following exceptions:

- Because I/O open drain and I/O open sink buffers do not have pullup circuitry, the option $xv_{pu}=nd\_state\_pu$ should not be specified.
- Similarly, because I/O open source buffers do not have pulldown circuitry, the option $xv_{pd}=nd\_state\_pd$ should not be specified.

### Input ECL Buffer

The syntax of the input ECL buffer element card is:

```plaintext
B_INPUT_ECL nd_pc nd_gc nd_in nd_out_of_in
+ file='file_name' model='model_name'
+ [typ={typ|min|max|fast|slow}] [power={on|off}]
+ [buffer={11|input_ecl}]
+ [interp={1|2}]
+ [nowarn]
+ [c_com_pc=c_com_pc_value]
+ [c_com_gc=c_com_gc_value]
+ [pc_scal=pc_scal_value]
+ [gc_scal=gc_scal_value]
```

The input ECL buffer is similar to the input buffer. The only difference is in default values for $Vinl$ and $Vinh$. 
The syntax of the output ECL buffer element card is:

```plaintext
B_OUTPUT_ECL nd_pu nd_out nd_in [nd_pc nd_gc]
+ file='file_name' model='model_name'
+ [typ={typ|min|max|fast|slow}] [power={on|off}]
+ [buffer={12|output_ecl}]
+ [xv_pu=state_pu] [xv_pd=state_pd]
+ [interpol={1|2}]
+ [ramp_fwf={0|1|2}] [ramp_rwf={0|1|2}]
+ [fwf_tune=fwf_tune_value] [rwf_tune=rwf_tune_value]
+ [nowarn]
+ [c_com_pu=c_com_pu_value]
+ [c_com_pc=c_com_pc_value]
+ [c_com_gc=c_com_gc_value]
+ [pu_scal=pu_scal_value]
+ [pd_scal=pd_scal_value]
+ [pc_scal=pc_scal_value]
+ [gc_scal=gc_scal_value]
```
Nodes `nd_pc` and `nd_gc` are optional. However, either both or none can be specified. The total number of external nodes is either 3 or 5, any other number is an error. The output ECL buffer does not have a pulldown node. The pulldown table in the IBIS file is referenced in respect to pullup voltage.

If nodes `nd_pc` and `nd_gc` are not given on the element card but Power_Clamp and/or Ground_Clamp I-V curves are present in the model in question, then the simulator will issue an error message (this simulator behavior is different from that for the output buffer).

In other respects, the output ECL buffer is similar to the output buffer. Please see “Output Buffer” on page 19-8 for more information.

Figure 19-7: Output ECL Buffer

Tristate ECL Buffer

The syntax for the tristate ECL buffer element card is:
Buffers Using IBIS Models

B_3STATE_ECL nd_pu nd_out nd_in nd_en [nd_pc nd_gc]
+ file='file_name' model='model_name'
+ [typ={typ|min|max|fast|slow}] [power={on|off}]
+ [buffer={14|three_state_ecl}]
+ [xv_pu=state_pu] [xv_pd=state_pd]
+ [interpol={1|2}]
+ [ramp_fwf={0|1|2}] [ramp_rwf={0|1|2}]
+ [fwf_tune=fwf_tune_value] [rwf_tune=rwf_tune_value]
+ [nowarn]
+ [c_com_pu=c_com_pu_value]
+ [c_com_pc=c_com_pc_value]
+ [c_com_gc=c_com_gc_value]
+ [pu_scal=pu_scal_value]
+ [pd_scal=pd_scal_value]
+ [pc_scal=pc_scal_value]
+ [gc_scal=gc_scal_value]
+ [rwf_scal=rwf_scal_value]
+ [fwf_scal=fwf_scal_value]
+ [spu_scal=spu_scal_value]
+ [spd_scal=spd_scal_value]

Nodes nd_pc and nd_gc are optional. However, either both or none can be specified. The total number of external nodes is either 4 or 6, any other number is an error. The tristate ECL buffer does not have a pulldown node. The pulldown table in the IBIS file is referenced in respect to pullup voltage.

If nodes nd_pc and nd_gc are not given on the element card but Power_Clamp and/or Ground_Clamp I-V curves are present in the model in question, then the simulator will issue an error message (this simulator behavior is different from that for the tristate buffer).

In other respects, the tristate ECL buffer is similar to the tristate buffer. Please see “Tristate Buffer” on page 19-11 for more information.
The syntax for the input-output ECL buffer element card is:

```
B_IO_ECL nd_pu nd_out nd_in nd_en nd_out_of_in [nd_pc nd_gc]
+ file='file_name' model='model_name'
+ [typ={typ|min|max|fast|slow}] [power={on|off}]
+ [buffer={13|io_ecl}]
+ [xv_pu=state_pu] [xv_pd=state_pd]
+ [interpol={1|2}]
+ [ramp_fwf={0|1|2}] [ramp_rwf={0|1|2}]
+ [fwf_tune=fwf_tune_value] [rwf_tune=rwf_tune_value]
+ [nowarn]
+ [c_com_pu=c_com_pu_value]
+ [c_com_pc=c_com_pc_value]
+ [c_com_gc=c_com_gc_value]
+ [pu_scal=pu_scal_value]
+ [pd_scal=pd_scal_value]
+ [pc_scal=pc_scal_value]
+ [gc_scal=gc_scal_value]
+ [rwf_scal=rwf_scal_value]
+ [fwf_scal=fwf_scal_value]
+ [spu_scal=spu_scal_value]
```
+ [spd_scal=spd_scal_value]

Nodes \textit{nd}_{pc} and \textit{nd}_{gc} are optional. However, either both or none can be specified. The total number of external nodes is either 5 or 7, any other number is an error. The tristate ECL buffer does not have a pulldown node. The pulldown table in the IBIS file is referenced in respect to pullup voltage.

If nodes \textit{nd}_{pc} and \textit{nd}_{gc} are not given on the element card but Power_Clamp and/or Ground_Clamp I-V curves are present in the model in question, then the simulator will issue an error message (this simulator behavior is different from that for Input-Output buffer).

In other respects, the input-output ECL buffer is similar to the input-output buffer. Please see “Input/Output Buffer” on page 19-15 for more information.
Figure 19-9: Input-Output ECL Buffer
Specifying Common Keywords

**Required Keywords**

**file='file_name'**

Identifies the IBIS file. *file_name* must be lower case and specify either the absolute path for the file or the relative path in respect to the directory from which Star-Hspice is run.

**Example**

```plaintext
file = '.ibis/at16245.ibs'
file = '/home/oneuser/ibis/models/abc.ibs'
```

**model='model_name'**

Identifies the model for a buffer from the IBIS file, specified with keyword file='...'. The keyword *model_name* is case sensitive and must match one of the models from the IBIS file.

**Example**

```plaintext
model = 'ABC_1234_out'
model = 'abc_1234_IN'
```

**Optional Keywords**

All other keywords are optional; if not used, default values will be selected. Optional keywords are enclosed in square brackets [ ] in the buffer cards.

The notation keyword={val_1|val_2|...|val_n} is used to denote that the keyword takes a value from the set val_1, val_2, ..., val_n. The order of the keywords is not important.
buffer= \{\text{Buffer\_Number} \mid \text{Buffer\_Type}\}

Where \text{buffer\_number} is an integer from the range \(1 \leq N \leq 17\). Each buffer has an assigned number as follows:

<table>
<thead>
<tr>
<th>Buffer Type</th>
<th>Buffer Number</th>
<th>Number of nodes (nominal or min/max)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>OUTPUT</td>
<td>2</td>
<td>4/6</td>
<td></td>
</tr>
<tr>
<td>INPUT_OUTPUT</td>
<td>3</td>
<td>6/8</td>
<td></td>
</tr>
<tr>
<td>THREE_STATE</td>
<td>4</td>
<td>5/7</td>
<td></td>
</tr>
<tr>
<td>OPEN_DRAIN</td>
<td>5</td>
<td>4/6</td>
<td></td>
</tr>
<tr>
<td>IO_OPEN_DRAIN</td>
<td>6</td>
<td>6/8</td>
<td></td>
</tr>
<tr>
<td>OPEN_SINK</td>
<td>7</td>
<td>4/6</td>
<td></td>
</tr>
<tr>
<td>IO_OPEN_SINK</td>
<td>8</td>
<td>6/8</td>
<td></td>
</tr>
<tr>
<td>OPEN_SOURCE</td>
<td>9</td>
<td>4/6</td>
<td></td>
</tr>
<tr>
<td>IO_OPEN_SOURCE</td>
<td>10</td>
<td>6/8</td>
<td></td>
</tr>
<tr>
<td>INPUT_ECL</td>
<td>11</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>OUTPUT_ECL</td>
<td>12</td>
<td>3/5</td>
<td></td>
</tr>
<tr>
<td>IO_ECL</td>
<td>13</td>
<td>5/7</td>
<td></td>
</tr>
<tr>
<td>THREE_STATE_ECL</td>
<td>14</td>
<td>4/6</td>
<td>was 17</td>
</tr>
<tr>
<td>SERIES</td>
<td>15</td>
<td>not implemented</td>
<td></td>
</tr>
<tr>
<td>SERIES_SWITCH</td>
<td>16</td>
<td>not implemented</td>
<td></td>
</tr>
<tr>
<td>TERMINATOR</td>
<td>17</td>
<td>not implemented</td>
<td>was 14</td>
</tr>
</tbody>
</table>

The value of \text{buffer\_number} and \text{buffer\_type} should match the buffer type specified by keyword \text{model}=\ldots\text{.} The keyword \text{buffer=} \{\text{Buffer\_Number} \mid \text{Buffer\_Type}\} provides an extra check for the input netlist. If the keyword is omitted, this checking is not performed.
**typ={typ|min|max|fast|slow}**

If the value of the buffer parameter *typ* is either *typ*, or *min*, or *max*, then this value signifies the column in the IBIS file from which the data are used for the current simulation. The default is *typ=typ*. If *min* or *max* data are not available, *typ* data are used instead.

If the value of the buffer parameter *typ* is *fast* or *slow*, then certain combinations of *min* and *max* data are used. The following table specifies the exact type of data used for *fast* and *slow* values. Note that the table lists all the parameters and data types for all implemented buffers. Specific buffers use relevant data only. No buffer uses all the data given in the table (for example, parameters *Rgnd, Rpower, Rac, Cac* are given and used only for the terminator).

<table>
<thead>
<tr>
<th>Parameter/Data</th>
<th>Fast</th>
<th>Slow</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_comp</td>
<td>min</td>
<td>max</td>
</tr>
<tr>
<td>Temp_Range</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>Voltage_Range</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>Pullup_Ref</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>Pulldown_Ref</td>
<td>min</td>
<td>max</td>
</tr>
<tr>
<td>POWER_Clamp_Ref</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>GND_Clamp_Ref</td>
<td>min</td>
<td>max</td>
</tr>
<tr>
<td>Rgnd</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>Rpower</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>Rac</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>Cac</td>
<td>min</td>
<td>max</td>
</tr>
<tr>
<td>Pulldown</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>Pullup</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>GND_Clamp</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>POWER_Clamp</td>
<td>max</td>
<td>min</td>
</tr>
</tbody>
</table>
Using IBIS Models  Specifying Common Keywords

<table>
<thead>
<tr>
<th>Parameter/Data</th>
<th>Fast</th>
<th>Slow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ramp</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>Rising_waveform</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>Falling_waveform</td>
<td>max</td>
<td>min</td>
</tr>
<tr>
<td>V_fixture</td>
<td>max</td>
<td>min</td>
</tr>
</tbody>
</table>

**power={on|off}**

Default is `power=on`. Connect buffers to power sources that are specified in the IBIS file with keywords `Voltage Range`, `Pullup Reference`, `Pulldown Reference`, `POWER Clamp Reference`, and `GND Clamp Reference`.

By default, Star-Hspice connects required voltage sources for such external nodes as `Pullup`, `Pulldown`, `Power_Clamp`, and `Ground_Clamp` if applicable. You should not connect these nodes to voltage sources. However, names for these nodes should be provided, so you can print out the voltage values if required. For example:

```
.PRINT V(node_pullup)
```

If `power=off` is used, then the internal voltage sources are not included in the buffer and you are responsible for adding external voltage sources. Use this option if the voltage source is not connected directly to buffer nodes but through a circuit to account for parasitic RLC, to simulate power/ground bounce, and so on.

**interp={1|2}**

Default is `interp=1` (recommended). The I/V curves and V(t) curves need to be interpolated. Keyword `interp=1` uses linear interpolation and `interp=2` uses quadratic bi-spline interpolation.
xv_pu=nd_state_pu
xv_pd=nd_state_pd

The buffers with output function (output, input-output, tristate, and so on) are controlled by one (input) or two (input and enable) controlling signals. Describe the state of a buffer at any moment with two state variables, $St_{pu}$ and $St_{pd}$, which vary from 0 to 1. For example, if the output buffer is in LOW state, then $St_{pu}=0$, $St_{pd}=1$. If the output buffer transitions from a LOW state to HIGH state, then $St_{pu}$ continuously changes from 0 to 1, while $St_{pd}$ goes from 1 to 0. The actual time dependence for such a transition is derived from either ramp data or waveform(s).

You may want to know exactly how the transition takes place. The keywords xv_pu=nd_state_pu, xv_pd=nd_state_pd provide such information. Here nd_state_pu and nd_state_pd are names of additional nodes (which must be unique and are treated by Star-Hspice as any other node from the netlist, except for a 16-character limitation). If the keyword(s) are included, then Star-Hspice adds voltage source(s) (with 1MOhm parallel resistor).

The values of the voltages are equal to $St_{pu}$ and $St_{pd}$. They can be printed or displayed as follows:

```plaintext
.PRINT V(nd_state_pu) V(nd_state_pd)
```

**Figure 19-10: Equivalent Circuit for xv_pu=nd_state_pu Keyword**
ramp_fwf={0|1|2}
ramp_rwf={0|1|2}

Default is ramp_fwf=0, ramp_rwf=0. If ramp and/or waveform(s) data are available, then these options allow you to choose which data to use.

Parameter ramp_fwf controls falling waveform(s)/ramp. Parameter ramp_rwf controls rising waveform(s)/ramp.

- Value 0 denotes use ramp data.
- Value 1 denotes use one waveform:
  - For ramp_fwf=1, if more than one falling waveform is available, the first falling waveform found for the model in question will be used.
  - For ramp_rwf=1, if more than one rising waveform is available, the first rising waveform found for the model in question will be used.
- Value 2 denotes use two waveforms:
  - For ramp_fwf=2, if more than two falling waveforms are available, the first two falling waveforms found for the model in question will be used.
  - For ramp_rwf=2, if more than two rising waveforms are available, the first two rising waveforms found for the model in question will be used.

If Star-Hspice can not perform a specified type of processing (for example, if ramp_fwf=2 is specified, but only one falling waveform is found), it decrements values of ramp_fwf and/or ramp_rwf by one and attempts to process the new value(s) of ramp_fwf and/or ramp_rwf. In this case, a warning is printed (unless the nowarn option is set).

**Note:** Parameters ramp_fwf and ramp_rwf are independent and can take different values.
Keywords `fwf_tune` and `rwf_tune` are numbers between 0 and 1. Default is `fwf_tune=0.1, rwf_tune=0.1`.

The following two parameters control the algorithm for processing ramp and waveforms:
- `fwf_tune` is used only if `ramp_fwf` is 0 or 1.
- `rwf_tune` is used only if `ramp_rwf` is 0 or 1.

The effect of these parameters when switching the output buffer from LOW to HIGH is illustrated in Figure 19-11.

**Figure 19-11: Change in Values of St_pu(t) and St_pd(t) When a Buffer is Switched from LOW to HIGH**

Initially, we have

```
St_pu(0)=0, St_pd(0)=1
```

Both ramp data and a single rising waveform provide information about the switching process and, in particular, a time interval, $\Delta_T$, during which the transition from LOW -> HIGH occurs.

The difference between the two data types (ramp and a single rising waveform) is that the shape of the waveform for ramp is fixed as a linearly growing function from LOW to HIGH, while an actual waveform accounts for an arbitrary time dependence.

However, this is not enough information to determine $St_pu(t)$ and $St_pd(t)$ [recall that $St_pu(0)=0, St_pd(0)=1, St_pu(\Delta_T)=1, St_pd(\Delta_T)=0$]. Mathematically, we have one linear equation with two unknowns that have an infinite number of solutions. To resolve this problem, additional conditions on $St_pu, St_pd$ should be imposed (some use $St_pu+St_pd=1$).
The following approach is adopted in Star-Hspice.

Because the circuitry that goes from ON to OFF (for rising waveforms, pulldown circuitry) usually undergoes this transition much faster than the circuitry that goes from OFF to ON (for rising waveforms, pullup circuitry), we specify a fraction of time in units of $\delta_T$ during which the circuitry that goes from ON to OFF undergoes the transition.

Therefore, if $\text{rwf\_tune}=0.1$, then during $0.1*\delta_T$ the pulldown circuitry switches from ON to OFF. The transition is a linear function of time. After imposing this additional condition, we can uniquely find the rate of transition for the circuitry that goes from the OFF state to ON state.

This approach is also valid for the $\text{fwf\_tune}$ parameter.

Parameters $\text{fwf\_tune}$ and $\text{rwf\_tune}$ should be considered tuning parameters. The significance of these parameters strongly depends on I/V curves for pullup and pulldown circuitries. A change in $\text{fwf\_tune}$ and $\text{rwf\_tune}$ can be insignificant or very significant depending on the I/V curves. We recommend that you adjust these parameters slightly to evaluate the accuracy of the model.

Note, that in the case of two waveforms, the corresponding system of equations is completely defined mathematically and parameters $\text{fwf\_tune}$ and $\text{rwf\_tune}$ are not used (ignored if specified). However, if data given in two waveforms are inaccurate or inconsistent with other data, Star-Hspice can use single waveform or ramp data instead of two waveforms (giving a warning). If this occurs, $\text{fwf\_tune}$ and/or $\text{rwf\_tune}$ are used even if ramp$_{\text{fwf}}=2$, ramp$_{\text{rwf}}=2$.

If the two-waveform data are inconsistent or inaccurate, the results can be less accurate than ramp or one-waveform results. We recommend that two-waveform results be compared against ramp and one-waveform results.

The algorithm used to find the evolution of states in the case of ramp data and single waveform can be augmented by other algorithms if there are such requests from the users.

The keywords, $\text{xv\_pu=nd\_state\_pu}$ and $\text{xv\_pd=nd\_state\_pd}$, can be used to print and/or view the state evolution functions $St_{\text{pu}}(t)$ and $St_{\text{pd}}(t)$.
nowarn

The keyword nowarn suppresses warning messages from the IBIS parser. Note that there is no equal sign “=” and value after the nowarn keyword. Do not use nowarn as the first keyword after the nodes list. There should be at least one keyword followed by “=” and a value between the list of nodes and the keyword nowarn.

\[ c_{\text{com}} = c_{\text{com} \_\text{pu}} \]
\[ c_{\text{com}} = c_{\text{com} \_\text{pd}} \]
\[ c_{\text{com}} = c_{\text{com} \_\text{pc}} \]
\[ c_{\text{com}} = c_{\text{com} \_\text{gc}} \]

By default (default 1) the die capacitance \( C_{\text{comp}} \) is connected between \( \text{node}_\text{out} \) (\( \text{nd}_\text{in} \) for input buffer) and ideal ground. For simulating power bounce and ground bounce it may be desirable to split \( C_{\text{comp}} \) into several parts and connect between \( \text{node}_\text{out} \) (\( \text{nd}_\text{in} \) for input buffer) and some (or all) of nodes \( \text{node}_\text{pu} \), \( \text{node}_\text{pd} \), \( \text{node}_\text{pc} \), \( \text{node}_\text{gc} \).

If at least one of the optional parameters \( c_{\text{com} \_\text{pu}}, c_{\text{com} \_\text{pd}}, c_{\text{com} \_\text{pc}}, c_{\text{com} \_\text{gc}} \) is specified then the default 1 is not applicable and unspecified parameters takes value zero (default 2). Values \( c_{\text{com} \_\text{pu}}, c_{\text{com} \_\text{pd}}, c_{\text{com} \_\text{pc}}, c_{\text{com} \_\text{gc}} \) are dimensionless and denote fractions of \( C_{\text{comp}} \) connected between \( \text{node}_\text{out} \) (\( \text{nd}_\text{in} \) for input buffer) and respective nodes (either \( \text{node}_\text{pu}, \text{node}_\text{pd}, \text{node}_\text{pc}, \text{or} \text{node}_\text{gc} \)). In other words \( C_{\text{comp}} * c_{\text{com} \_\text{pu}} \) is capacitance connected between \( \text{node}_\text{out} \) and \( \text{node}_\text{pu} \), etc.

If given, values of \( c_{\text{com} \_\text{pu}}, c_{\text{com} \_\text{pd}}, c_{\text{com} \_\text{pc}}, \text{and} c_{\text{com} \_\text{gc}} \) should be nonnegative.

It is expected that \( c_{\text{com} \_\text{pu}} + c_{\text{com} \_\text{pd}} + c_{\text{com} \_\text{pc}} + c_{\text{com} \_\text{gc}} = 1 \), however Hspice does not enforce this requirement and only gives warning if the requirement is not satisfied. In such case the states are derived under THE assumption that the die has \( C_{\text{comp}} \) specified in the IBIS files while during simulation different value of \( C_{\text{comp}} \) used, namely \( C_{\text{comp}} *(c_{\text{com} \_\text{pu}}+c_{\text{com} \_\text{pd}}+c_{\text{com} \_\text{pc}}+c_{\text{com} \_\text{gc}}) \). Effectively it means
that some additional capacitance is connected in parallel to C_comp (possibly negative).

In the case of buffer types: output, input-output, 3-state, if nodes node_pc and node_gc are not specified in the netlist, c_com_pc is added to c_com_pu and c_com_gc is added to c_com_pd (after that c_com_pc and c_com_gc are not used anymore).

In the case of buffer types: open drain, open sink, input-output open drain, input-output open sink if nodes node_pc and node_gc are not specified in the netlist, c_com_pc if given is ignored, c_com_gc is added to c_com_pd (after that c_com_gc is not used anymore).

In the case of buffer types: open source, input-output open source, if nodes node_pc and node_gc are not specified in the netlist, c_com_gc if given is ignored, c_com_pc is added to c_com_pu (after that c_com_pc is not used anymore).

In the case of buffer types: output ECL, input-output ECL, 3-state ECL if nodes node_pc and node_gc are not specified in the netlist, c_com_pc and c_com_gc are ignored (assign zero values).

In the case of buffer types: output ECL, input-output ECL, 3-state ECL if c_com_pd is not zero, it is added to c_com_pu (after that c_com_pd is not used anymore).
Differential Pins

Differential pins refer to the relationship between buffers. Differential pins are specified in the “Component Description” section of the IBIS standard. The following examples explain how differential pins can be simulated with the Star-Hspice implementation of IBIS.

**Figure 19-12: Output buffers**

You must use two separate buffers, each of which is specified by a separate card in the netlist. They are related only through their input, which is differential.

The only way the inverter can implement in this situation is by specifying two independent voltage sources that have opposite polarity.
Nodes `out_of_in_1`, `out_of_in_2` must be specified but are not used. In the case of differential input buffers, the voltage will be probed between `nd_in_1` and `nd_in_2` and processed by a voltage dependent voltage source as described below.

$V_{\text{diff}}$ is a differential voltage parameter from the IBIS file (default is 200 mV). Add definition of parameter $V_{\text{diff}}$, voltage controlled voltage source $E_{\text{diff\_out\_of\_in}}$, and a resistor $R_{\text{diff\_out\_of\_in}}$.

**Example**

```plaintext
.PARAM V_diff = 0.2
E_diff_out_of_in  diff_out_of_in  0 PWL(1)  nd_in_1  nd_in_2
  + '- V_diff' 0 '+ V_diff' 1 R_diff_out_of_in
  + diff_out_of_in 0 1x
```

Use the voltage across $R_{\text{diff\_out\_of\_in}}$ as the output of the differential input buffer.

- If $V(nd_in_1) - V(nd_in_2) < V_{\text{diff}}$, $V(diff_out_of_in) = 0$
- If $V(nd_in_1) - V(nd_in_2) > V_{\text{diff}}$, $V(diff_out_of_in) = 1$
Scaling Buffer Strength

Sometimes there is a need to scale buffer strength (that is, increase or decrease current for output type buffers for a given value of the output voltage). This enables the same IBIS file to be used to simulate buffers of different strengths. Let us designate K as a factor for current multiplication. For the original buffer, the value of K=1. This section describes how to accomplish this scaling using the F-element for a single output buffer and differential output buffer.

The original circuit for a single output buffer is as follows:

```
Buffer  nd_pu  nd_pd  nd_out  nd_pc  nd_gc
+    file=<filename>  model=<modelname>
+  Rload  nd_out  gnd  Rload_val
```

The scaled circuit for a single output buffer is as follows:

```
Buffer  nd_pu  nd_pd  nd_out  nd_pc  nd_gc
+    file=<filename>  model=<modelname>
+  Vsenser  nd_out  nd_out_prime  V=0
+  Rload  nd_out_prime  gnd  Rload_val
+  F_element  gnd  nd_out_prime  Vsenser  K-1
```

The original circuit for a differential output buffer is as follows:

```
Buffer1  nd_pu1  nd_pd1  nd_out1  nd_pc1  nd_gc1
+    file=<filename1>  model=<modelname1>
Buffer2  nd_pu2  nd_pd2  nd_out2  nd_pc2  nd_gc2
+    file=<filename2>  model=<modelname2>
+  R_load  n_out1  n_out2  R_load_value
```

The scaled circuit for a differential output buffer is as follows:

```
Buffer1  nd_pu1  nd_pd1  nd_out1  nd_pc1  nd_gc1
+    file=<filename1>  model=<modelname1>
Buffer2  nd_pu2  nd_pd2  nd_out2  nd_pc2  nd_gc2
+    file=<filename2>  model=<modelname2>
+  V_sense  n_out1  n_out1_prime  0V
+  F_element  n_out2  n_out1_prime  v_sense  K-1
+  R_load  n_out1_prime  n_out2  R_load_value
```
Please note the polarity of the F-element. For the scaling factor K=1, the current controlled current source does not supply any current and effectively we have the original circuit.

Buffers in subcircuits

-----------------------------
* example 1 * buffers in subcircuit, power=on
-----------------------------

v_in1 nd_in1 0 pulse
+ ( 0V 1.0V CLK_Q_PRD DLT_TIME DLT_TIME CLK_H_PRD CLK_PRD )
v_en1 nd_en1 0 1V
v_in2 nd_in2 0 pulse
+ ( 1.1V 0V CLK_Q_PRD DLT_TIME DLT_TIME CLK_H_PRD CLK_PRD )
v_en2 nd_en2 0 1V

x1 nd_out1 nd_in1 nd_en1 nd_outofin1 buffer11
x2 nd_out2 nd_in2 nd_en2 nd_outofin2 buffer11

R_load nd_out1 nd_out2 50

.subckt buffer11 nd_out0 nd_in0 nd_en0 nd_outofin0
b_io_0 nd_pu0 nd_pd0 nd_out nd_in0 nd_en0 nd_outofin0 nd_pc0
nd_g0
+ file = '92lv090b.ibs'
+ model = 'DS92LV090A_DOUT'
+ typ=typ power=on
+ buffer=3
+ interpol=1
+ xpin nd_out nd_out0 pin22
.ends

.subckt pin22 nd_out nd_out0
R_pin nd_out_c nd_out0 50m
C_pin nd_out_c 0 0.3p
L_pin nd_out nd_out_c 2n
.ends
In this example buffers are connected to power sources implicitly by Hspice inside subcircuit. Subcircuit external terminals does not need to include nd_pu, nd_pd, nd_pc, nd_gc.

*****************************************************************************
* example 2* buffers in subcircuit, power=off ****************************
*****************************************************************************
  v_in1 nd_in1  0  pulse
  + ( 0V 1.0V CLK_Q_PRD DLT_TIME DLT_TIME CLK_H_PRD CLK_PRD )
  v_en1 nd_en1  0  1V
  v_in2 nd_in2  0  pulse
  + ( 1.1V 0V CLK_Q_PRD DLT_TIME DLT_TIME CLK_H_PRD CLK_PRD )
  v_en2 nd_en2  0  1V

  x1 nd_power 0 nd_out1 nd_in1 nd_en1 nd_outofin1 nd_power 0
  buffer11
  x2 nd_power 0 nd_out2 nd_in2 nd_en2 nd_outofin2 nd_power 0
  buffer11

  R_load nd_out1 nd_out2  50

  .subckt buffer11 nd_pu0 nd_pd0 nd_out0 nd_in0 nd_en0 nd_outofin0
  nd_pc0 nd_gc0
  r_0 nd_pu0 nd_pd0 1.23456789x
  b_io_0         nd_pu0 nd_pd0 nd_out nd_in0 nd_en0 nd_outofin0
  nd_pc0 nd_gc0
  + file = '92lv090b.ibs'
  + model = 'DS92LV090A_DOUT'
  + typ=typ power=off
  + buffer=3
  + interpol=1
  + xpin nd_out nd_out0 pin22
  .ends

  .subckt pin22 nd_out nd_out0
  R_pin nd_out_c nd_out0  50m
  C_pin nd_out_c 0        0.3p
  L_pin nd_out      nd_out_c 2n
  .ends

  V_power nd_power 0  3.3V
In this example, only one voltage source, V\_power, is used to power all buffers. All power nodes, \texttt{nd\_pu}, \texttt{nd\_pd}, \texttt{nd\_pc}, \texttt{nd\_gc}, should be explicitly provided.
Example

Below is a complete example of the netlist that contains an output buffer, transmission line, and input buffer.

A digital signal is supplied to the node \textit{nd\_in}. It is transmitted by the output buffer to a network, goes through a transmission line, is received by an input buffer, and transformed into digital form and available on node \textit{nd\_out\_of\_in\_1}. IBIS file at16245a.ibs resides in the directory .ibs, which is located in the directory from which Star-Hspice is run.

\begin{verbatim}
**********************************************************
* test for iob: output buffer -- transmission line --input buffer
**********************************************************
.option post probe
**********************************************************
* Analysis
**********************************************************
.tran 0.05n 20n
**********************************************************
* Stimuli
**********************************************************
V_in nd_in 0 1V pulse ( 0V 1V 1n 1n 1n 4n 10n )
**********************************************************
* Output
**********************************************************
.print tran v(nd_pu) v(nd_pd) v(nd_out) v(nd_in) 
+ v(nd_in_1) v(nd_out_of_in_1)
**********************************************************
* OUTPUT BUFFER
\end{verbatim}
b2 nd_pu nd_pd nd_out nd_in nd_pc nd_gc
+ file = '.ibis/at16245a.ibs'
+ model = 'AT16245_OUT'

**TRANSMISSION LINE**

.PARAM Z_0=50
.PARAM T_delay=10ns
.PARAM Length=1mm

W1 N=1 nd_out GND nd_in_1 GND Umodel=Uname L=Length
.model Uname u nl=1 LEVEL=3 elev=3 llev=0 plev=1 nlay=2
+ zk=Z_0 delay=T_delay

**INPUT BUFFER**

b1 nd_pc_1 nd_gc_1 nd_in_1 nd_out_of_in_1
+ file = '.ibis/at16245a.ibs'
+ model = 'AT16245_IN'

.end
**Additional Notes**

This section provides some additional notes to clarify technical issues. Most of this information was developed as a result of customer interaction.

**Keywords**

Parameters *fwf_tune, rwf_tune* specify transition time for circuitry (either pullup or pulldown) that goes from the ON to OFF state as a fraction of time, *delta_T*, for a transition for the opposite circuitry (either pulldown or pullup) from OFF to ON state. The *delta_T* value for ramp data transition time is different from the value for single waveform transition time (*delta_T* depends on parameters *ramp_fwf* and *ramp_rwf*). Consequently, the absolute values for transition time from the ON to OFF state are different for ramp data and single waveform data.

**Voltage Thresholds**

Voltages applied to the input node and enable node are digital signals. They should be either 0 or 1. It is acceptable to specify input voltage as:

\[ V_{in} \text{ nd}_{in} 0 \text{ pulse (0 3.3 0 0.5n 0.5n 4n 8n)} \]

However, Star-Hspice currently detects only two thresholds, 20% and 80% of [0,1] swing, i.e., 0.2V and 0.8V. If a buffer is non-inverting and in a LOW state, it will start transition to a HIGH state, if \( V_{in} > 0.8V \). If the buffer is in HIGH state, it will start the transition to LOW state, if \( V_{in} < 0.2V \). Specifying input voltage in the range \([0 , 3.3V]\) as in the above example does not make LOW -> HIGH transitions better in any way, but can add uncertainty over time interval 0.5ns, when the transition actually occurs.

**.OPTION D_IBIS**

The new option d_ibis is introduced to specify the directory where IBIS files are located. Example of usage:

```
.OPTION d_ibis='/home/user/ibis/models'
```
If several directories are specified, then Hspice will look for IBIS files in the local directory (the directory where Hspice is run), then in the directories specified through .option d_ibis in the order that .option cards appear in the netlist. At most, four directories can be specified through d_ibis option.

Examples of usage:

```
.OPTION d_ibis='/nfs/port/user/hspice/run/subckt/optd'
+ buffer nd_pu nd_pd nd_out nd_in
+ file = 'opt.ibs'
+ model = 'DS92LV090A_DOUT'
```

Option d_ibis is case-sensitive.
Understanding Warning and Error Messages

If certain conditions are met (or not met), Star-Hspice prints warnings or error messages. Some examples of these messages are described below.

Warnings are issued if Star-Hspice determines that the input data are inconsistent. In this case, Star-Hspice modifies data to make consistent and runs the simulation with modified data. Static I/V curves take precedence over V(t) curves and ramp data. If data are modified by Star-Hspice, it is unlikely the results of simulation with a test load will match V(t) curves given in the IBIS file. To achieve high accuracy, input data should be consistent. Pay close attention to warnings and understand the causes.

Errors are issued if Star-Hspice determines that the run can not be continued with the given data.

Example

An example of a warning:
```
** warning**  iob_eles2:205:
  text of the warning, line1
  text of the warning, line2
```

An example of an error:
```
** error**  iob_eles2:205:
  text of the error, line1
  text of the error, line2
```

Text that follows the comments '** warning**' and '** error**', such as 'iob_eles2:205;' identifies the location where the problem occurs. It is intended to help the developer solve the problem.

The following information is intended for users. A list of selected warnings follows.
1. I/V Curves
PC and GC I-V curves should be equal to zero at zero voltage, I(V=0)=0. If different values are found, a warning is issued. For example, for a PC I/V curve the following warning is issued.

```
** warning** ffffffff:NNN:
POWER_Clamp curve should be zero at origin
found value  1.85800E-01
I/V curve is not modified
```

This warning occurs for Power_Clamp and Ground_Clamp I/V curves. Star-HSpice takes these I/V curves as given but an error is likely to occur if I(V=0) is not zero.

2. I-V curves and rising/falling waveforms (RWF, FWF) should be consistent. Star-HSpice verifies consistency for the end points of RWF, FWF. If inconsistency is detected, the I/V curves take precedence over V(t) curves, and V(t) curves are modified to make them consistent with I-V curves.

```
** warning** ffffffff:NNN:
Falling WF min estimate and given value differ
estimate =  3.3540E-01 given =  6.6000E-01
```

In this example, I/V curves give a value of 3.3540E-01V for minimum voltage when pullup is OFF (if pullup is available) and pulldown is ON (if pulldown is available) for load specified for FWF. The minimum voltage value of FWF in the IBIS file is 6.6000E-01V. In this case, the FWF is modified by Star-HSpice to be consistent with I-V curves.

3. I/V curves and ramp data should be consistent (if ramp is used). Star-HSpice verifies consistency for end points of rising edge and falling edge. The IBIS standard requires that ramp data correspond to 20% to 80% transition of the total voltage change. If inconsistency is detected, the following warning is given.

```
** warning** ffffffff:NNN:
Inconsistency between Ramp and PD/PU data is detected.
dV_r=  1.59000E+00   V_rwf_max=  3.30000E+00   V_rwf_min=  3.35400E-01
transition from 20% to 80% is not satisfied
FILE  = gtl-plus.ibs
MODEL = io_buf
```
In this example, voltage changes from 0.3354 V to 3.3 V on the rising edge, as calculated from I/V curves. 60% of this range is 1.78 V. The IBIS file gives 1.59 V. Star-Hspice gives warnings, takes derivative for ramp as given in the IBIS file and the voltage range as calculated from I/V curves.
References

The official IBIS Open Forum web site is located at:
http://www.eia.org/EIG/IBIS/ibis.htm

This site contains articles introducing IBIS, text of the IBIS standard, examples of IBIS files, and tools such as the golden parser. The site also links to other web sites devoted to IBIS.

Other web and ftp sites that have information about IBIS are:
http://www.eda.org/pub/ibis/
http://www.vhdl.org/pub/ibis/
Chapter 20

Introducing MOSFETs

A MOSFET is defined by the MOSFET model and element parameters, and two submodels selected by the CAPOP and ACM model parameters. The CAPOP model parameter specifies the model for the MOSFET gate capacitances. The ACM (Area Calculation Method) parameter selects the type of diode model to be used for the MOSFET bulk diodes. Each of these submodels has associated parameters that define the characteristics of the gate capacitances and bulk diodes.

MOSFET models are either p-channel or n-channel models; they are classified according to level such as LEVEL 1 or LEVEL 50.

This chapter covers the design model and simulation aspects of MOSFET models, parameters of each model level, and associated equations. MOSFET diode and MOSFET capacitor model parameters and equations are also described. For information about individual models and their parameters, see Selecting MOSFET Models: LEVEL 1-40 and Selecting MOSFET Models: LEVEL 47-62.

This chapter describes:

- Understanding MOSFET Models
- Selecting Models
- Using the General MOSFET Model Statement
- Using Nonplanar and Planar Technologies
- Using MOSFET Equivalent Circuits
- Using a MOSFET Diode Model
- Using MOS Diode Equations
- Using Common Threshold Voltage Equations
Introducing MOSFETs

- Performing MOSFET Impact Ionization
- Using MOS Gate Capacitance Models
- Using Noise Models
- Using Temperature Parameters and Equations
Understanding MOSFET Models

The selection of the MOSFET model type for use in analysis usually depends on the electrical parameters critical to the application. LEVEL 1 models are most often used for simulation of large digital circuits where detailed analog models are not needed. LEVEL 1 models offer low simulation time and a relatively high level of accuracy with regard to timing calculations. When precision is required, as for analog data acquisition circuitry, use the more detailed models, such as the LEVEL 6 IDS model or one of the BSIM models (LEVEL 13, 39, or 49).

For precision modeling of integrated circuits, the BSIM models consider the variation of model parameters as a function of sensitivity of the geometric parameters. The BSIM models also reference a MOS charge conservation model for precision modeling of MOS capacitor effects.

Use the SOSFET model (LEVEL 27) to model silicon-on-sapphire MOS devices. You can include photocurrent effects at this level.

Use LEVELS 5 and LEVEL 38 for depletion MOS devices.

LEVEL 2 models consider bulk charge effects on current. LEVEL 3 models require less simulation time and provides as much accuracy as LEVEL 2 and have a greater tendency to converge. LEVEL 6 models are compatible with models originally developed with ASPEC. Use LEVEL 6 models to model ion-implanted devices.
Selecting Models

A MOS transistor is described by use of an element statement and a .MODEL statement.

The element statement defines the connectivity of the transistor and references the .MODEL statement. The .MODEL statement specifies either an n- or p-channel device, the level of the model, and a number of user-selectable model parameters.

Example

The following example specifies a PMOS MOSFET with a model reference name, PCH. The transistor is modeled using the LEVEL 13 BSIM model. The parameters are selected from the model parameter lists in this chapter.

M3 3 2 1 0 PCH <parameters>
.MODEL PCH PMOS LEVEL=13 <parameters>

Selecting MOSFET Model LEVELs

MOSFET models consist of client private and public models selected by the parameter .MODEL statement LEVEL parameter. New models are constantly being added to Star-Hspice.

Not all MOSFET models are available in the PC version of Star-Hspice. The following table shows what is available for PC users. Models listed are either on all platforms, including PC, as indicated in the third column, or they are available on all platforms except the PC, as indicated in the last column.

<table>
<thead>
<tr>
<th>Level</th>
<th>MOSFET Model Description</th>
<th>All Platforms including PC</th>
<th>All Platforms except PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Schichman-Hodges model</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MOS2 Grove-Frohman model (SPICE 2G)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MOS3 empirical model (SPICE 2G)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Level</td>
<td>MOSFET Model Description</td>
<td>All Platforms including PC</td>
<td>All Platforms except PC</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------</td>
<td>----------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>4</td>
<td>Grove-Frohman: LEVEL 2 model derived from SPICE 2E.3</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>AMI-ASPEC depletion and enhancement (Taylor-Huang)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Lattin-Jenkins-Grove (ASPEC style parasitics)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Lattin-Jenkins-Grove (SPICE style parasitics)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>advanced LEVEL 2 model</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>9 **</td>
<td>AMD</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>10 **</td>
<td>AMD</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Fluke-Mosaid model</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>12 **</td>
<td>CASMOS model (GTE style)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>BSIM model</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>14 **</td>
<td>Siemens LEVEL=4</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>user-defined model based on LEVEL 3</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>not used</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>17</td>
<td>Cypress model</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>18 **</td>
<td>Sierra 1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>19 ***</td>
<td>Dallas Semiconductor model</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>20 **</td>
<td>GE-CRD FRANZ</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>21 **</td>
<td>STC-ITT</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>22 **</td>
<td>CASMOS (GEC style)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Siliconix</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
## Selecting Models

### Introducing MOSFETs

<table>
<thead>
<tr>
<th>Level</th>
<th>MOSFET Model Description</th>
<th>All Platforms including PC</th>
<th>All Platforms except PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 **</td>
<td>GE-Intersil advanced</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>25 **</td>
<td>CASMOS (Rutherford)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>26 **</td>
<td>Sierra 2</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>27</td>
<td>SOSFET</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>28</td>
<td>BSIM derivative; Avant! proprietary model</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>29 ***</td>
<td>not used</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>30 ***</td>
<td>VTI</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>31 ***</td>
<td>Motorola</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>32 ***</td>
<td>AMD</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>33 ***</td>
<td>National Semiconductor</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>34*</td>
<td>(EPFL) not used</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>35 **</td>
<td>Siemens</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>36 ***</td>
<td>Sharp</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>37 ***</td>
<td>TI</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>38</td>
<td>IDS: Cypress depletion model</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>39</td>
<td>BSIM2</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>41</td>
<td>TI Analog</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>46 ***</td>
<td>SGS-Thomson MOS LEVEL 3</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>47</td>
<td>BSIM3 Version 2.0</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>49</td>
<td>BSIM3 Version 3 (Enhanced)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>50</td>
<td>Philips MOS9</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>53</td>
<td>BSIM3 Version 3 (Berkeley)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>54</td>
<td>UC Berkeley BSIM4 Model</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Selecting MOSFET Capacitors

The MOSFET capacitance model parameter, CAPOP, is associated with the MOS model. Depending on the value of CAPOP, different capacitor models are used to model the MOS gate capacitance, that is, the gate-to-drain capacitance, the gate-to-source capacitance, and the gate-to-bulk capacitance. CAPOP allows for the selection of several versions of the Meyer and charge conservation model.

Some of the capacitor models are tied to specific DC models; they are stated as such. Others are for general use by any DC model.

<table>
<thead>
<tr>
<th>Level</th>
<th>MOSFET Model Description</th>
<th>All Platforms including PC</th>
<th>All Platforms except PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>55</td>
<td>EPFL-EKV Model Ver 2.6, R 11</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>UC Berkeley BSIM3-SOI MOSFET Model Ver 2.0.1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>University of Florida SOI Model Ver 4.5 (Beta-98.4)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>UC Berkeley BSIM3-501 FD Model</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>RPI a-Si TFT Model</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>RPI Poli-Si TFT Model</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

* not officially released
** equations are proprietary – documentation not provided
*** requires a license and equations are proprietary – documentation not provided
Selecting Models

Introducing MOSFETs

CAPOP=4 selects the recommended charge-conserving model (from among CAPOP=11, 12, or 13) for the given DC model.

Table 20-1: CAPOP=4 Selections

<table>
<thead>
<tr>
<th>MOS Level</th>
<th>Default CAPOP</th>
<th>CAPOP=4 selects:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>13, 28, 39</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>others</td>
<td>2</td>
<td>11</td>
</tr>
</tbody>
</table>

LEVELs 49 and 53 use the Berkeley capacitance-model parameter CAPMOD. The proprietary models, as well as LEVELs 5, 17, 21, 22, 25, 27, 31, 33, 49, 53, 55, and 58 have their own built-in capacitance routines.
Selecting MOS Diodes

The model parameter ACM (Area Calculation Method), which controls the geometry of the source and drain diffusions, selects the modeling of the bulk-to-source and bulk-to-drain diodes of the MOSFET model. The diode model includes the diffusion resistance, capacitance, and DC currents to the substrate.

<table>
<thead>
<tr>
<th>ACM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM=0</td>
<td>SPICE model, parameters determined by element areas</td>
</tr>
<tr>
<td>ACM=1</td>
<td>ASPEC model, parameters function of element width</td>
</tr>
<tr>
<td>ACM=2</td>
<td>Avant! model, combination of ACM=0,1 and provisions for lightly doped drain technology</td>
</tr>
<tr>
<td>ACM=3</td>
<td>Extension of ACM=2 model that deals with stacked devices (shared source/drains) and source/drain periphery capacitance along gate edge.</td>
</tr>
</tbody>
</table>

Searching Models as Function of W, L

Model parameters are often the same for MOSFETs having width and length dimensions within specific ranges. To take advantage of this, create a MOSFET model for a specific range of width and length. Star-Hspice uses these MOSFET model parameters to select the appropriate model for the given width and length.

The Star-Hspice automatic model selection program searches a data file for a MOSFET model with the width and length range specified in the MOSFET element statement. This model statement is then used in the simulation.

To search a data file for MOSFET models within a given range of width and length, provide a root extension for the model reference name (in the .MODEL statement). Also, use the model geometric range parameters LMIN, LMAX, WMIN, and WMAX. These model parameters give the range of the physical length and width dimensions to which the MOSFET model applies. For example, if the model reference name in the element statement is NCH, the model selection program examines the models with the same root model reference name NCH such as NCH.1, NCH.2 or NCH.A. The model selection program selects the first MOSFET model statement whose geometric range parameters include the width and length specified in the associated MOSFET element statement.
The following example shows how to call the MOSFET model selection program from a data file. The model selector program examines the .MODEL statements that have the model reference names with root extensions NCHAN.2, NCHAN.3, NCHY.20, and NCHY.50.

**Example**

```
*FILE: SELECTOR.SP  TEST OF MOS MODEL SELECTOR
.OPTION LIST WL SCALE=1U SCALM=1U NOMOD
.OP
V1 1 0 5
V2 2 0 4
V3 3 0 1
V4 4 0 -1
M1 1 2 3 4 NCHAN 10 2
M2 1 2 3 4 NCHAN 10 3
M3 1 2 3 4 NCH 10 4
M4 1 2 3 4 NCHX 10 5
M5 1 2 3 4 NCHY 20 5
M6 1 2 3 4 NCHY 50 5
$$$$$$  FOR CHANNEL LENGTH SELECTION
.MODEL NCHAN.2 NMOS LEVEL=2 VTO=2.0 UO=800 TOX=500 NSUB=1E15
+    RD=10  RS=10  CAPOP=5
+    LMIN=1  LMAX=2.5  WMIN=2  WMAX=15
.MODEL NCHAN.3 NMOS LEVEL=2 VTO=2.2 UO=800 TOX=500 NSUB=1E15
+    RD=10  RS=10  CAPOP=5
+    LMIN=2.5  LMAX=3.5  WMIN=2  WMAX=15
$$$$$$  NO SELECTION FOR CHANNEL LENGTH AND WIDTH
.MODEL NCH  NMOS LEVEL=2 VTO=2.3 UO=800 TOX=500 NSUB=1E15
+    RD=10  RS=10  CAPOP=5
+$+    LMIN=3.5  LMAX=4.5  WMIN=2  WMAX=15
.MODEL NCHX NMOS LEVEL=2 VTO=2.4 UO=800 TOX=500 NSUB=1E15
+    RD=10  RS=10  CAPOP=5
+$+    LMIN=4.5  LMAX=100  WMIN=2  WMAX=15
$$$$$$  FOR CHANNEL WIDTH SELECTION
.MODEL NCHY.20 NMOS LEVEL=2 VTO=2.5 UO=800 TOX=500 NSUB=1E15
+    RD=10  RS=10  CAPOP=5
+    LMIN=4.5  LMAX=100  WMIN=15  WMAX=30
.MODEL NCHY.50 NMOS LEVEL=2 VTO=2.5 UO=800 TOX=500 NSUB=1E15
+    RD=10  RS=10  CAPOP=5
+    LMIN=4.5  LMAX=100  WMIN=30  WMAX=500
.END
```
Setting MOSFET Control Options

Specific control options (set in the .OPTIONS statement) used for MOSFET models include the following. For flag-type options, 0 is unset (off) and 1 is set (on).

- **ASPEC**: This option uses ASPEC MOSFET model defaults and set units. Default=0.
- **BYPASS**: This option avoids recomputation of nonlinear functions that do not change with iterations. Default=1.
- **MBYPASS**: BYPASS tolerance multiplier. Default=1.
- **DEFAD**: Default drain diode area. Default=0.
- **DEFAS**: Default source diode area. Default=0.
- **DEFL**: Default channel length. Default=1e⁻⁴ m.
- **DEFW**: Default channel width. Default=1e⁻⁴ m.
- **DEFNRD**: Default number of squares for drain resistor. Default=0.
- **DEFNRS**: Default number of squares for source resistor. Default=0.
- **DEFPD**: Default drain diode periphery. Default=0.
- **DEFPS**: Default source diode periphery. Default=0.
- **GMIN**: Pn junction parallel transient conductance. Default=1e⁻¹² mho.
- **GMINDC**: Pn junction parallel DC conductance. Default=1e⁻¹² mho.
- **SCALE**: Element scaling factor. Default=1.
- **SCALM**: Model scaling factor. Default=1.
- **WL**: This option changes the order of specifying MOS element VSIZE from the default order, length-width, to width-length. Default=0.
Override the defaults DEFAD, DEFAS, DEFL, DEFNRD, DEFNRS, DEFPD, DEFPS, and DEFW in the MOSFET element statement by specifying AD, AS, L, NRD, NRS, PD, PS, and W, respectively.

**Scaling Units**

Units are controlled by the options SCALE and SCALM. SCALE scales element statement parameters, and SCALM scales model statement parameters. SCALM also affects the MOSFET gate capacitance and diode model parameters. In this chapter, scaling applies only to those parameters specified as scaled. If SCALM is specified as a parameter in a .MODEL statement, it overrides the option SCALM. In this way, models using different values of SCALM can be used in the same simulation. MOSFET parameter scaling follows the same rules as for other model parameters, for example:

**Table 20-2: Model Parameter Scaling**

<table>
<thead>
<tr>
<th>Parameter Units</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>meter</td>
<td>multiplied by SCALM</td>
</tr>
<tr>
<td>meter²</td>
<td>multiplied by SCALM²</td>
</tr>
<tr>
<td>meter⁻¹</td>
<td>divided by SCALM</td>
</tr>
<tr>
<td>meter⁻²</td>
<td>divided by SCALM²</td>
</tr>
</tbody>
</table>

Override global model size scaling for individual MOSFET, diode, and BJT models that uses the .OPTION SCALM=<val> statement by including SCALM=<val> in the .MODEL statement. .OPTION SCALM=<val> applies globally for JFETs, resistors, transmission lines, and all models other than MOSFET, diode, and BJT models, and cannot be overridden in the model.

**Scaling for LEVEL 25 and 33**

When using the proprietary LEVEL 25 (Rutherford CASMOS) or LEVEL 33 (National) models, the SCALE and SCALM options are automatically set to 1e-6. If you use these models together with other scalable models, however, set the options, SCALE=1e-6 and SCALM=1e-6, explicitly.
Bypassing Latent Devices

Use the BYPASS (latency) option to decrease simulation time in large designs. It speeds simulation time by not recalculating currents, capacitances, and conductances if the voltages at the terminal device nodes have not changed. The BYPASS option applies to MOSFETs, MESFETs, JFETs, BJTs, and diodes. Use .OPTION BYPASS to set BYPASS.

BYPASS can result in a reduction in accuracy of the simulation for tightly coupled circuits such as op-amps, high gain ring oscillators, and so on. Use .OPTION MBYPAS to set MBYPAS to a smaller value to improve the accuracy of the results.
Using the General MOSFET Model Statement

Following is the syntax for all model specifications. All related parameter levels are covered in their respective sections.

Syntax

```
.MMODEL mname [PMOS | NMOS] (<LEVEL=val> <keyname1=val1>
+ <keyname2=val2>...) <VERSION=version_number>
```

or

```
.MMODEL mname NMOS(<LEVEL = val> <keyname1 = val1>
+ <keyname2=val2>...) <VERSION=version_number> ...)
```

*mname* Model name. Elements refer to the model by this name.

*PMOS* Identifies a p-channel MOSFET model

*NMOS* Identifies an n-channel MOSFET model

*LEVEL* The MOSFET model includes several device model types. Use the LEVEL parameter for selection. Default=1.0.

*VERSION* This parameter specifies the version number of the model, for LEVEL=13 BSIM and LEVEL=39 BSIM2 models only. See the .MODEL statement description for information about the effects of the VERSION parameter.

Example

```
.MMODEL MODP PMOS LEVEL=7 VTO=-3.25 GAMMA=1.0)
.MMODEL MODN NMOS LEVEL=2 VTO=1.85 TOX=735e-10)
.MMODEL MODN NMOS LEVEL=39 TOX=2.0e-02 TEMP=2.5e+01 VERSION=95.1
```
Using Nonplanar and Planar Technologies

Two MOSFET fabrication technologies have dominated integrated circuit design: nonplanar and planar technologies. Nonplanar technology uses metal gates. The simplicity of the process generally provides acceptable yields. The primary problem with metal gates is metal breakage across the field oxide steps. Field oxide is grown by oxidizing the silicon surface. When the surface is cut, it forms a sharp edge. Since metal must be affixed to these edges in order to contact the diffusion or make a gate, it is necessary to apply thicker metal to compensate for the sharp edges. This metal tends to gather in the cuts, making etching difficult. The inability to accurately control the metal width necessitates very conservative design rules and results in low transistor gains.

In planar technology, the oxide edges are smooth, with a minimal variance in metal thickness. Shifting to nitride was accomplished by using polysilicon gates. Adding a chemical reactor to the MOS fabrication process enables not only the deposition of silicon nitride, but also that of silicon oxide and polysilicon. The ion implanter is the key element in this processing, using implanters with beam currents greater than 10 milliamperes.

Since implanters define threshold voltages and “diffusions” as well as field thresholds, processes require a minimum number of high temperature oven steps. This enables low temperature processing and maskless pattern generation. The new wave processes are more similar to the older nonplanar metal gate technologies.
Using Field Effect Transistors

The metal gate MOSFET is nonisoplanar as shown in Figure 20-1 and Figure 20-2.

**Figure 20-1: Field Effect Transistor**

Looking at the actual geometry, from source-to-drain, Figure 20-2 shows a perspective of the nonisoplanar MOSFET.

**Figure 20-2: Field Effect Transistor Geometry**
To visualize the construction of the silicon gate MOSFET, observe how a source or drain to field cuts (Figure 20-3.) The cut A-B shows a drain contact (Figure 20-4).

**Figure 20-3: Isoplanar Silicon Gate Transistor**
The cut from the source to the drain is represented by C - D (Figure 20-5), which includes the contacts.
The planar process produces parasitic capacitances at the poly to field edges of the device. The cut along the width of the device demonstrates the importance of these parasitics (Figure 20-6).

The encroachment of the field implant into the channel not only narrows the channel width, but also increases the gate to bulk parasitic capacitance.

**Figure 20-6: Isoplanar MOSFET, Width Cut**

![Figure 20-6: Isoplanar MOSFET, Width Cut](image)

| 1 - 2 | Drawn width of the gate W |
| 3 - 4 | Depleted or accumulated channel (parameter WD) |
| 4 - 5 | Effective channel width W + XW - 2 WD |
| 3 - 6 | Physical channel width W + XW |
Using MOSFET Equivalent Circuits

Equation Variables

This section lists the equation variables and constants.

Table 20-3: Equation Variables and Constants

<table>
<thead>
<tr>
<th>Variable/Quantity</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>cbd</td>
<td>Bulk-to-drain capacitance</td>
</tr>
<tr>
<td>cbs</td>
<td>Bulk-to-source capacitance</td>
</tr>
<tr>
<td>cbg</td>
<td>Gate-to-bulk capacitance</td>
</tr>
<tr>
<td>cgd</td>
<td>Gate-to-drain capacitance</td>
</tr>
<tr>
<td>cgs</td>
<td>Gate-to-source capacitance</td>
</tr>
<tr>
<td>f</td>
<td>Frequency</td>
</tr>
<tr>
<td>gbd</td>
<td>Bulk-to-drain dynamic conductance</td>
</tr>
<tr>
<td>gbs</td>
<td>Bulk-to-source dynamic conductance</td>
</tr>
<tr>
<td>gds</td>
<td>Drain-to-source dynamic conductance controlled by vds</td>
</tr>
<tr>
<td>gdb</td>
<td>Drain-to bulk impact ionization conductance</td>
</tr>
<tr>
<td>gm</td>
<td>Drain-to-source dynamic transconductance controlled by vgs</td>
</tr>
<tr>
<td>gmbs</td>
<td>Drain-to-source dynamic bulk transconductance controlled by vsb</td>
</tr>
<tr>
<td>ibd</td>
<td>Bulk-to-drain DC current</td>
</tr>
<tr>
<td>ibs</td>
<td>Bulk-to-source DC current</td>
</tr>
<tr>
<td>ids</td>
<td>Drain-to-source DC current</td>
</tr>
<tr>
<td>idb</td>
<td>Drain-to-bulk impact ionization current</td>
</tr>
<tr>
<td>ind</td>
<td>Drain-to-source equivalent noise circuit</td>
</tr>
<tr>
<td>inrd</td>
<td>Drain resistor equivalent noise circuit</td>
</tr>
</tbody>
</table>
Using the MOSFET Current Convention

Figure 20-7 shows the assumed direction of current flow through a MOS transistor. When printing the drain current, use either I(M1) or I1(M1) syntax. I2 produces the gate current, I3 produces the source current, and I4 produces the substrate current. References to bulk are the same as references to the substrate.
Using MOSFET Equivalent Circuits

Star-Hspice uses three equivalent circuits in the analysis of MOSFETs: DC, transient, and AC and noise equivalent circuits. The components of these circuits form the basis for all element and model equation discussion. The equivalent circuit for DC sweep is the same as the one used for transient analysis, except capacitances are not included. Figures 20-8 through Figure 20-10 display the MOSFET equivalent circuits.

The fundamental component in the equivalent circuit is the DC drain-to-source current (ids). For the noise and AC analyses, the actual ids current is not used. Instead, the model uses the partial derivatives of ids with respect to the terminal voltages vgs, vds, and vbs. The names for these partial derivatives are:

**Transconductance**

\[ g_m = \frac{\partial (ids)}{\partial (vgs)} \]

**Conductance**

\[ g_{ds} = \frac{\partial (ids)}{\partial (vds)} \]
**Bulk Transconductance**

\[ g_{mb} = \frac{\partial ids}{\partial v_{bs}} \]

The \( ids \) equation describes the basic DC effects of the MOSFET. The effects of gate capacitance and of source and drain diodes are considered separately from the DC \( ids \) equations. In addition, the impact ionization equations are treated separately from the DC \( ids \) equation, even though its effects are added to \( ids \).

**Figure 20-8: Equivalent Circuit, MOSFET Transient Analysis**
Figure 20-9: Equivalent Circuit, MOSFET AC Analysis
Introducing MOSFETs Using MOSFET Equivalent Circuits

Figure 20-10: Equivalent Circuit, MOSFET AC Noise Analysis
Using a MOSFET Diode Model

The Area Calculation Method (ACM) parameter allows for the precise control of modeling bulk-to-source and bulk-to-drain diodes within MOSFET models. The ACM model parameter is used to select one of three different modeling schemes for the MOSFET bulk diodes. This section discusses the model parameters and model equations used for the different MOSFET diode models.

Selecting MOSFET Diode Models

To select a MOSFET diode model, set the ACM parameter within the MOSFET model statements. If ACM=0, the pn bulk junctions of the MOSFET are modeled in the SPICE-style. The ACM=1 diode model is the original ASPEC model. The ACM=2 model parameter specifies the Star-Hspice improved diode model, which is based on a model similar to the ASPEC MOSFET diode model. The ACM=3 diode model is a further Star-Hspice improvement that deals with capacitances of shared sources and drains and gate edge source/drain-to-bulk periphery capacitance. If the ACM model parameter is not set, the diode model defaults to the ACM=0 SPICE model. ACM=0 and ACM=1 models do not permit the specification of HDIF. ACM=0 does not permit specification of LDIF. Furthermore, the geometric element parameters AD, AS, PD, and PS are not used for the ACM=1 model.

Enhancing Convergence

The GMIN and GMINDC options parallel a conductance across the bulk diodes and drain-source for transient and DC analysis, respectively. Use these options to enhance the convergence properties of the diode model, especially when the model has a high off resistance. Use the parameters RSH, RS, and RD to keep the diode from being overdriven in either a DC or transient forward bias condition. These parameters also enhance the convergence properties of the diode model.
Using MOSFET Diode Model Parameters

This section describes the diode model parameters for MOSFET.

DC Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td>Units</td>
<td>0</td>
<td>Area calculation method</td>
</tr>
<tr>
<td>JS</td>
<td>amp/m²</td>
<td>0</td>
<td>Bulk junction saturation current $J_{Sscaled} = J_S / SCALM$ for ACM=1, unit is amp/m and $J_{Sscaled} = J_S / SCALM$.</td>
</tr>
<tr>
<td>JSW</td>
<td>amp/m</td>
<td>0</td>
<td>Sidewall bulk junction saturation current $J_{SWscaled} = J_{SW} / SCALM$.</td>
</tr>
<tr>
<td>IS</td>
<td>amp</td>
<td>1e-14</td>
<td>Bulk junction saturation current. For the option ASPEC=1, default=0.</td>
</tr>
<tr>
<td>N</td>
<td>1</td>
<td>1</td>
<td>Emission coefficient</td>
</tr>
<tr>
<td>NDS</td>
<td>1</td>
<td>Reverse bias slope coefficient</td>
<td></td>
</tr>
<tr>
<td>VNDS</td>
<td>V</td>
<td>-1</td>
<td>Reverse diode current transition point</td>
</tr>
</tbody>
</table>

Using Capacitance Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBD</td>
<td>F</td>
<td>0</td>
<td>Zero bias bulk-drain junction capacitance. Used only when CJ and CJSW are 0.</td>
</tr>
<tr>
<td>CBS</td>
<td>F</td>
<td>0</td>
<td>Zero bias bulk-source junction capacitance. Used only when CJ and CJSW are 0.</td>
</tr>
</tbody>
</table>
### Using a MOSFET Diode Model

**Introducing MOSFETs**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJ (CDB, CSB, CJA)</td>
<td>F/m²</td>
<td>579.11 µF/m²</td>
<td>Zero-bias bulk junction capacitance: ( C_{Jscaled} = CJ/SCALM^2 ) For ACM=1 the unit is F/m and ( C_{Jscaled} = CJ/SCALM ) Default for option ASPEC=0 is: ( CJ = \left( \frac{\varepsilon_i \cdot q \cdot NSUB}{2 \cdot PB} \right)^{1/2} )</td>
</tr>
<tr>
<td>CJSW (CJP)</td>
<td>F/m</td>
<td>0</td>
<td>Zero-bias sidewall bulk junction capacitance ( C_{JSWscaled} = CJSW/SCALM ) Default = 0</td>
</tr>
<tr>
<td>CJGATE</td>
<td>F/m</td>
<td>CSJW</td>
<td>Zero-bias gate-edge sidewall bulk junction capacitance (ACM=3 only) ( C_{JGATEScaled} = CJGATE/SCALM ) Default = CJSW for Star-Hspice releases later than H9007D. Default = 0 for HSPICE releases H9007D and earlier, or if CJSW is not specified.</td>
</tr>
<tr>
<td>FC</td>
<td></td>
<td>0.5</td>
<td>Forward-bias depletion capacitance coefficient (not used)</td>
</tr>
<tr>
<td>MJ (EXA, EXJ, EXS, EXD)</td>
<td></td>
<td>0.5</td>
<td>Bulk junction grading coefficient</td>
</tr>
<tr>
<td>MJSW (EXP)</td>
<td></td>
<td>0.33</td>
<td>Bulk sidewall junction grading coefficient</td>
</tr>
<tr>
<td>NSUB (DNB, NB)</td>
<td>1/cm³</td>
<td>1.0e15</td>
<td>Substrate doping</td>
</tr>
<tr>
<td>PB (PHA, PHS, PHD)</td>
<td>V</td>
<td>0.8</td>
<td>Bulk junction contact potential</td>
</tr>
<tr>
<td>PHP</td>
<td>V</td>
<td>PB</td>
<td>Bulk sidewall junction contact potential</td>
</tr>
<tr>
<td>TT</td>
<td>s</td>
<td>0</td>
<td>Transit time</td>
</tr>
</tbody>
</table>
## Using Drain and Source Resistance Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD</td>
<td>ohm/sq</td>
<td>0.0</td>
<td>Drain ohmic resistance. This parameter is usually lightly doped regions’ sheet resistance for ACM 1.</td>
</tr>
<tr>
<td>RDC</td>
<td>ohm</td>
<td>0.0</td>
<td>Additional drain resistance due to contact resistance</td>
</tr>
<tr>
<td>LRD</td>
<td>ohm/m</td>
<td>0</td>
<td>Drain resistance length sensitivity. Use this parameter with automatic model selection in conjunction with WRD and PRD to factor model for device size.</td>
</tr>
<tr>
<td>WRD</td>
<td>ohm/m</td>
<td>0</td>
<td>Drain resistance length sensitivity (used with LRD)</td>
</tr>
<tr>
<td>PRD</td>
<td>ohm/m²</td>
<td>0</td>
<td>Drain resistance product (area) sensitivity (used with LRD)</td>
</tr>
<tr>
<td>RS</td>
<td>ohm/sq</td>
<td>0.0</td>
<td>Source ohmic resistance. This parameter is usually lightly doped regions’ sheet resistance for ACM 1.</td>
</tr>
<tr>
<td>LRS</td>
<td>ohm/m</td>
<td>0</td>
<td>Source resistance length sensitivity. Use this parameter with automatic model selection in conjunction with WRS and PRS to factor model for device size.</td>
</tr>
<tr>
<td>WRS</td>
<td>ohm/m</td>
<td>0</td>
<td>Source resistance width sensitivity (used with LRS)</td>
</tr>
<tr>
<td>PRS</td>
<td>ohm/m²</td>
<td>0</td>
<td>Source resistance product (area) sensitivity (used with LRS)</td>
</tr>
<tr>
<td>RSC</td>
<td>ohm</td>
<td>0.0</td>
<td>Additional source resistance due to contact resistance</td>
</tr>
<tr>
<td>RSH (RL)</td>
<td>ohm/sq</td>
<td>0.0</td>
<td>Drain and source diffusion sheet resistance</td>
</tr>
</tbody>
</table>
### Using MOS Geometry Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDIF</td>
<td>m</td>
<td>0</td>
<td>Length of heavily doped diffusion, from contact to lightly doped region (ACM=2, 3 only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HDIFscaled = HDIF \cdot SCALM</td>
</tr>
<tr>
<td>LD (DLAT, LATD)</td>
<td>m</td>
<td></td>
<td>Lateral diffusion into channel from source and drain diffusion. If LD and XJ are unspecified, LD default=0.0. When LD is unspecified, but XJ is specified, LD is calculated from XJ. LD default=0.75 \cdot XJ. For LEVEL 4 only, lateral diffusion is derived from LD \cdot XJ. LDscaled = LD \cdot SCALM</td>
</tr>
<tr>
<td>LDIF</td>
<td>m</td>
<td>0</td>
<td>Length of lightly doped diffusion adjacent to gate (ACM=1, 2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LDIFscaled = LDIF \cdot SCALM</td>
</tr>
<tr>
<td>WMLT</td>
<td>1</td>
<td></td>
<td>Width diffusion layer shrink reduction factor</td>
</tr>
<tr>
<td>XJ</td>
<td>m</td>
<td>0</td>
<td>Metallurgical junction depth</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XJscaled = XJ \cdot SCALM</td>
</tr>
<tr>
<td>XW (WDEL, DW)</td>
<td>m</td>
<td>0</td>
<td>Accounts for masking and etching effects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XWscaled = XW \cdot SCALM</td>
</tr>
</tbody>
</table>
Using an ACM=0 MOS Diode

The following example shows the parameter value settings for a MOSFET diode designed with a MOSFET that has a channel length of 3 µm and a channel width of 10 µm.

**Figure 20-11: ACM=0 MOS Diode**

Example

Consider a transistor with:

- **LD** = .5 µm
- **W** = 10 µm
- **L** = 3 µm
- **AD** = area of drain (about 80 pm²)
- **AS** = area of source (about 80 pm²)
- **CJ** = 4e-4 F/m²
- **CJSW** = 1e-10 F/m
- **JS** = 1e-8 A/m²
- **JSW** = 1e-13 A/m
- **NRD** = number of squares for drain resistance
- **NRS** = number of squares for source resistance
- **PD** = sidewall of drain (about 36 µm)
- **PS** = sidewall of source (about 36 µm)
Calculating Effective Areas and Peripheries

For ACM=0, the effective areas and peripheries are calculated as:

\[ AD_{\text{eff}} = M \cdot AD \cdot WMLT^2 \cdot SCALE^2 \]
\[ AS_{\text{eff}} = M \cdot AS \cdot WMLT^2 \cdot SCALE^2 \]
\[ PD_{\text{eff}} = M \cdot PD \cdot WMLT \cdot SCALE \]
\[ PS_{\text{eff}} = M \cdot PS \cdot WMLT \cdot SCALE \]

Calculating Effective Saturation Current

For ACM=0, the MOS diode effective saturation currents are calculated as:

Source Diode Saturation Current

Define:
\[ val = JS_{\text{scaled}} \cdot AS_{\text{eff}} + JSW_{\text{scaled}} \cdot PS_{\text{eff}} \]

If \( val > 0 \) then,
\[ isbs = val \]

Otherwise,
\[ isbd = M \cdot IS \]

Drain Diode Saturation Current

Define:
\[ val = JS_{\text{scaled}} \cdot AD_{\text{eff}} + JSW_{\text{scaled}} \cdot PD_{\text{eff}} \]

If \( val > 0 \) then,
\[ isbd = val \]

Otherwise,
\[ isbd = M \cdot IS \]
Calculating Effective Drain and Source Resistances

For ACM=0, the effective drain and source resistances are calculated as:

**Source Resistance**
Define:
\[ val = NRS \cdot RSH \]
If \( val > 0 \) then,
\[ RSeff = \frac{val + RSC}{M} \]
Otherwise:
\[ RSeff = \frac{RS + RSC}{M} \]

**Drain Resistance**
Define:
\[ val = NRD \cdot RSH \]
If \( val > 0 \) then,
\[ RDeff = \frac{val + RDC}{M} \]
Otherwise:
\[ RDeff = \frac{RD + RDC}{M} \]
Using an ACM=1 MOS Diode

Star-Hspice uses ASPEC-style diodes when the model parameter ACM=1 is specified. Parameters AD, PD, AS, and PS are not used, and the units JS and CJ differ from the SPICE style diodes (ACM=0).

Figure 20-12: ACM=1 MOS Diode

Example

The listings below are typical parameter value settings for a transistor with: LD=0.5 \( \mu \)m W=10 \( \mu \)m L=3 \( \mu \)m LDIF=0.5 \( \mu \)m

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJ</td>
<td>1e-10 F/m of gate width</td>
<td></td>
</tr>
<tr>
<td>CJSW</td>
<td>2e-10 F/m of gate width</td>
<td></td>
</tr>
<tr>
<td>JS</td>
<td>1e-14 A/m of gate width</td>
<td></td>
</tr>
<tr>
<td>JSW</td>
<td>1e-13 A/m of gate width</td>
<td></td>
</tr>
<tr>
<td>NRD</td>
<td>number of squares for drain resistance</td>
<td></td>
</tr>
<tr>
<td>NRS</td>
<td>number of squares for source resistance</td>
<td></td>
</tr>
</tbody>
</table>
Calculating Effective Areas and Peripheries

For ACM=1, the effective areas and peripheries are calculated as follows:

\[ A_{D eff} = W_{eff} \cdot WMLT \]
\[ A_{S eff} = W_{eff} \cdot WMLT \]
\[ P_{D eff} = W_{eff} \]
\[ P_{S eff} = W_{eff} \]

where:
\[ W_{eff} = M \cdot (W_{scaled} \cdot WMLT + XW_{scaled}) \]

**Note:** The \( W_{eff} \) is not quite the same as the \( w_{eff} \) given in the models LEVEL 1, 2, 3, 6, and 13 sections. The term \( 2 \cdot WD_{scaled} \) is not subtracted.

Calculating Effective Saturation Current

For ACM=1, the MOS diode effective saturation currents are calculated as follows:

**Source Diode Saturation Current**

Define:
\[ val = J_{Sscaled} \cdot A_{S eff} + J_{SWscaled} \cdot P_{S eff} \]

If \( val > 0 \) then,
\[ isbs = val \]

Otherwise:
\[ isbs = M \cdot IS \]
Drain Diode Saturation Current
Define:
\[ val = J_{scaled} \cdot A_{Deff} + J_{SWscaled} \cdot P_{Deff} \]
If \( val > 0 \) then,
\[ isbd = val \]
Otherwise,
\[ isbd = M \cdot IS \]

Calculating Effective Drain and Source Resistances
For ACM=1, the effective drain and source resistances are calculated as follows:

**Source Resistance**
For UPDATE=0,
\[ R_{Seff} = \frac{L_{scaled} + L_{DIFscaled}}{W_{eff}} \cdot R_S + \frac{N_{RS} \cdot R_{SH} + R_{SC}}{M} \]
If UPDATE \( \geq 1 \) and LDIF=0 and the ASPEC option is also specified then:
\[ R_{Seff} = \frac{1}{M} \cdot (R_S + N_{RS} \cdot R_{SH} + R_{SC}) \]
Drain Resistance

For UPDATE=0:

\[
R_{Deff} = \frac{LD_{scaled} + LDIF_{scaled}}{W_{eff}} \cdot RD + \frac{NRD \cdot RSH + RDC}{M}
\]

If UPDATE ≥ 1 and LDIF=0 and the ASPEC option is also specified then:

\[
R_{Deff} = \frac{1}{M} \cdot (RD + NRD \cdot RSH + RDC)
\]

Note: See LEVELs 6 and 7 for more possibilities.

Using an ACM=2 MOS Diode

Star-Hspice uses HSPICE style MOS diodes when the model parameter ACM=2 is specified. This allows a fold-back calculation scheme similar to the ASPEC method, retaining full model-parameter compatibility with the SPICE procedure. This method also supports both lightly and heavily doped diffusions (by setting the LD, LDIF, and HDIF parameters). The units of JS, JSW, CJ, and CJSW used in SPICE are preserved, permitting full compatibility.

ACM=2 automatically generates more reasonable diode parameter values than those for ACM=1. The ACM=2 geometry can be generated one of two ways:

- Element parameters: AD, AS, PD, and PS can be used for parasitic generation when specified in the element statement. Default options values for these parameters are not applicable.
- If the diode is to be suppressed, set IS=0, AD=0, and AS=0.

The source diode is suppressed if AS=0 is set in the element and IS=0 is set in the model. This setting is useful for shared contacts.
Figure 20-13: ACM=2 MOS Diode

Example

Transistor with LD=0.07\(\mu\)m W=10 \(\mu\)m L=2 \(\mu\)m LDIF=1 \(\mu\)m HDIF= 4 \(\mu\)m, typical MOSFET diode parameter values are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>Area of drain. Default option value for AD is not applicable.</td>
</tr>
<tr>
<td>AS</td>
<td>Area of source. Default option value for AS is not applicable.</td>
</tr>
<tr>
<td>CJ</td>
<td>1(\times)10^{-4} F/m^2</td>
</tr>
<tr>
<td>CJSW</td>
<td>1(\times)10^{-10} F/m</td>
</tr>
<tr>
<td>JS</td>
<td>1(\times)10^{-4} A/m^2</td>
</tr>
<tr>
<td>JSW</td>
<td>1(\times)10^{-10} A/m</td>
</tr>
<tr>
<td>HDIF</td>
<td>Length of heavy doped diffusion contact to gate (about 2 (\mu)m)</td>
</tr>
<tr>
<td>HDIFeff</td>
<td>HDIFeff=HDIF \cdot WMLT \cdot SCALM</td>
</tr>
<tr>
<td>LDIF+LD</td>
<td>Length of lightly doped diffusion (about 0.4(\mu)m)</td>
</tr>
<tr>
<td>NRD</td>
<td>Number of squares drain resistance. Default option value for NRD is not applicable.</td>
</tr>
<tr>
<td>NRS</td>
<td>Number of squares source resistance. Default option value for NRS is not applicable.</td>
</tr>
<tr>
<td>PD</td>
<td>Periphery of drain, including the gate width for ACM=2. No default.</td>
</tr>
<tr>
<td>PS</td>
<td>Periphery of source, including the gate width for ACM=2. No default.</td>
</tr>
<tr>
<td>RD</td>
<td>Resistance (ohm/square) of lightly doped drain diffusion (about 2000)</td>
</tr>
<tr>
<td>RS</td>
<td>Resistance (ohm/square) of lightly doped source diffusion (about 2000)</td>
</tr>
<tr>
<td>RSH</td>
<td>Diffusion sheet resistance (about 35)</td>
</tr>
</tbody>
</table>
Calculating Effective Areas and Peripheries

For ACM=2, the effective areas and peripheries are calculated as:

If AD is not specified then,
\[ A_{Deff} = 2 \cdot HDIF_{eff} \cdot Weff \]

Otherwise,
\[ A_{Deff} = M \cdot AD \cdot WMLT^2 \cdot SCALE^2 \]

If AS is not specified then,
\[ A_{Seff} = 2 \cdot HDIF_{scaled} \cdot Weff \]

Otherwise,
\[ A_{Seff} = M \cdot AS \cdot WMLT^2 \cdot SCALE^2 \]

If PD is not specified then,
\[ P_{Deff} = 4 \cdot HDIF_{eff} + 2 \cdot Weff \]

Otherwise,
\[ P_{Deff} = M \cdot PD \cdot WMLT \cdot SCALE \]

If PS is not specified then,
\[ P_{Seff} = 4 \cdot HDIF_{eff} + 2 \cdot Weff \]

Otherwise:
\[ P_{Seff} = M \cdot PS \cdot WMLT \cdot SCALE \]

where:
\[ Weff = M \cdot (W_{scaled} \cdot WMLT + X_{Wscaled}) \]
\[ HDIF_{eff} = HDIF_{scaled} \]
\[ HDIF_{scaled} = HDIF \cdot SCALM \cdot WMLT \]
Note: The $W_{eff}$ is not quite the same as the $W_{eff}$ given in the model LEVEL 1, 2, 3, and 6 sections. The term $2 \cdot WD_{scaled}$ is not subtracted.

Calculating Effective Saturation Currents

For ACM=2, the MOS diode effective saturation currents are calculated as:

**Source Diode Saturation Current**

Define:

$$val = J_{Sscaled} \cdot AS_{eff} + J_{SWscaled} \cdot P_{Seff}$$

If $val > 0$ then,

$$isbs = val$$

Otherwise:

$$isbs = M \cdot IS$$

**Drain Diode Saturation Current**

Define:

$$val = J_{Sscaled} \cdot AD_{eff} + J_{SWscaled} \cdot P_{Deff}$$

If $val > 0$ then,

$$isbd = val$$

Otherwise:

$$isbd = M \cdot IS$$
Calculating Effective Drain and Source Resistances

For ACM=2, the effective drain and source resistances are calculated as:

**Source Resistance**

If NRS is specified then,

\[
R_{Seff} = \frac{L_{Dscaled} + L_{DlFscaled}}{W_{eff}} \cdot R_S + \left(\frac{NRS \cdot R_{SH} + R_{SC}}{M}\right)
\]

Otherwise:

\[
R_{Seff} = \frac{R_{SC}}{M} + \frac{H_{DlFeff} \cdot R_{SH} + (L_{Dscaled} + L_{DlFscaled}) \cdot R_S}{W_{eff}}
\]

**Drain Resistance**

If NRD is specified then,

\[
R_{Deff} = \frac{L_{Dscaled} + L_{DlFscaled}}{W_{eff}} \cdot R_D + \left(\frac{NRD \cdot R_{SH} + R_{DC}}{M}\right)
\]

Otherwise:

\[
R_{Deff} = \frac{R_{DC}}{M} + \frac{H_{DlFeff} \cdot R_{SH} + (L_{Dscaled} + L_{DlFscaled}) \cdot R_D}{W_{eff}}
\]

**Using an ACM=3 MOS Diode**

Use ACM=3 to model MOS diodes of the stacked devices properly. In addition, the CJGATE model parameter separately models the drain and source periphery capacitances along the gate edge. Therefore, the PD and PS calculations do not include the gate periphery length. CJGATE defaults to CJSW, which, in turn, defaults to 0.
The AD, AS, PD, PS calculations depend on the layout of the device, which is determined by the value of element parameter GEO. The GEO can be specified on the MOS element description. It can have the following values:

GEO=0: indicates the drain and source of the device are not shared by other devices (default).

GEO=1: indicates the drain is shared with another device.

GEO=2: indicates the source is shared with another device.

GEO=3: indicates the drain and source are shared with another device.

**Figure 20-14: Stacked Devices and Corresponding GEO Values**

![Diagram showing stacked devices and corresponding GEO values]

**Calculating Effective Areas and Peripheries**

For ACM=3, the effective areas and peripheries are calculated differently, depending on the value of GEO.

If AD is not specified, then,

For GEO=0 or 2,

\[ A_{Deff} = 2 \cdot HDIF_{eff} \cdot W_{eff} \]

For GEO=1 or 3,

\[ A_{Deff} = HDIF_{eff} \cdot W_{eff} \]
Otherwise:

\[ A_{Deff} = M \cdot AD \cdot WMLT^2 \cdot SCALE^2 \]

If AS is not specified, then,

For GEO=0 or 1,

\[ A_{Seff} = 2 \cdot HDIF_{eff} \cdot W_{eff} \]

For GEO=2 or 3,

\[ A_{Seff} = HDIF_{eff} \cdot W_{eff} \]

Otherwise:

\[ A_{Seff} = M \cdot AS \cdot WMLT^2 \cdot SCALE^2 \]

If PD is not specified, then,

For GEO=0 or 2,

\[ P_{Deff} = 4 \cdot HDIF_{eff} + W_{eff} \]

For GEO=1 or 3,

\[ P_{Deff} = 2 \cdot HDIF_{eff} \]

Otherwise:

\[ P_{Deff} = M \cdot PD \cdot WMLT \cdot SCALE \]

If PS is not specified, then,

For GEO=0 or 1,

\[ P_{Seff} = 4 \cdot HDIF_{eff} + W_{eff} \]

For GEO=2 or 3,

\[ P_{Seff} = 2 \cdot HDIF_{eff} \]
Otherwise:

\[ P_{Seff} = M \cdot PS \cdot WMLT \cdot SCALE \]

The \( W_{eff} \) and \( HDIF_{eff} \) is calculated as follows:

\[ W_{eff} = M \cdot (W_{scaled} \cdot WMLT + X_{Wscaled}) \]

\[ HDIF_{eff} = HDIF_{scaled} \cdot WMLT \]

**Note:** The \( W_{eff} \) is not quite the same as the \( W_{eff} \) given in the model LEVEL 1, 2, 3, and 6 sections. The term \( 2 \cdot WD_{scaled} \) is not subtracted.

### Effective Saturation Current Calculations

The ACM=3 model calculates the MOS diode effective saturation currents the same as ACM=2.

### Effective Drain and Source Resistances

The ACM=3 model calculates the effective drain and source resistances the same as ACM=2.
Using MOS Diode Equations

This section describes the MOS diode equations.

**DC Current**

The drain and source MOS diodes are paralleled with GMINDC conductance in the DC analysis and with GMIN in the transient analysis. The total DC current is the sum of diode current and the conductance current. The diode current is calculated as follows.

**Drain and Source Diodes Forward Biased**

\( v_{bs} > 0, \)

\[ i_{bs} = i_{sbs} \cdot (e^{v_{bs}/(N \cdot v_{t})} - 1) \]

\( v_{bd} > 0, \)

\[ i_{bd} = i_{sbd} \cdot (e^{v_{bd}/(N \cdot v_{t})} - 1) \]

**Drain and Source Diodes Reverse Biased**

For \( 0 > v_{bs} > V_{NDS}, \)

\[ i_{bs} = g_{sbs} \cdot v_{bs} \]

For \( v_{bs} < V_{NDS}, \)

\[ i_{bs} = g_{sbs} \cdot V_{NDS} + \left( g_{sbs} \cdot \frac{V_{NDS}}{N_{DS}} \right) \cdot (v_{bs} - V_{NDS}) \]

For \( 0 > v_{bd} > V_{NDS}, \)

\[ i_{bd} = g_{sbd} \cdot v_{bd} \]
For \( \text{vbd} < \text{VNDS} \),

\[
\text{i}_{bd} = \text{gsbd} \cdot \text{VNDS} + \frac{\text{gsbd}}{\text{NDS}} \cdot (\text{vbd} - \text{VNDS})
\]

where

\[
|\text{gsbs}| = |\text{isbs}|, \text{ and } |\text{gdbd}| = |\text{isbd}|
\]

**Using MOS Diode Capacitance Equations**

Each MOS diode capacitance is the sum of diffusion and depletion capacitance. The diffusion capacitance is evaluated in terms of the small signal conductance of the diode and a model parameter \( \text{TT} \), representing the transit time of the diode. The depletion capacitance depends on the choice of ACM, and is discussed below.

Calculate the bias-dependent depletion capacitance by defining the intermediate quantities: \( \text{C0BS} \), \( \text{C0BD} \), \( \text{C0BS_SW} \), and \( \text{C0BD_SW} \), which depend on geometric parameters, such as \( \text{ASeff} \) and \( \text{PSeff} \) calculated under various ACM specifications.

When ACM=3, the intermediate quantities \( \text{C0BS_SW} \), and \( \text{C0BD_SW} \) include an extra term to account for \( \text{CJGATE} \).

For ACM=2, the parameter \( \text{CJGATE} \) has been added in a backward compatible manner. Therefore, the default behavior of \( \text{CJGATE} \) makes the intermediate quantities \( \text{C0BS_SW} \) and \( \text{C0BD_SW} \) the same as for previous versions. The default patterns are:

- If neither \( \text{CJSW} \) nor \( \text{CJGATE} \) is specified, both default to zero.
- If \( \text{CJGATE} \) is not specified, it defaults to \( \text{CJSW} \), which in turn defaults to zero.
- If \( \text{CJGATE} \) is specified, and \( \text{CJSW} \) is not specified, then \( \text{CJSW} \) defaults to zero.

The intermediate quantities \( \text{C0BS} \), \( \text{C0BS_SW} \), \( \text{C0BD} \), and \( \text{C0BD_SW} \) are calculated as follows.

\[
\text{C0BS} = \text{CJscaled} \cdot \text{ASeff}
\]
C0BD = CJscaled*ADeff

If (ACM= 0 or 1), then:
C0BS_SW = CJSWscaled*PSeff
C0BD_SW = CJSWscaled*PDeff

If (ACM=2):
If (PSeff < Weff), then:
C0BS_SW = CJGATEscaled*PSeff
Otherwise:
C0BS_SW = CJSWscaled*(PSeff-Weff) + CJGATEscaled*Weff

If (PDeff < Weff), then:
C0BD_SW = CJGATEscaled*PDeff
Otherwise:
C0BD_SW = CJSWscaled*(PDeff-Weff) + CJGATEscaled*Weff

If (ACM=3), then:
C0BS_SW = CJSWscaled*PSeff + CJGATEscaled*Weff
C0BD_SW = CJSWscaled*PDeff + CJGATEscaled*Weff

Source Diode Capacitance
If (C0BS + C0BS_SW) > 0, then:
For vbs < 0,
\[ \text{capbs} = TT \frac{\partial i_{bs}}{\partial v_{bs}} + C0BS \cdot \left(1 - \frac{v_{bs}}{PB}\right)^{-MJ} \]
\[ + C0BS_SW \cdot \left(1 - \frac{v_{bs}}{PHP}\right)^{-MJSW} \]
For $v_{bs} > 0$, 
\[ cap_{bs} = TT \cdot \frac{\partial i_{bs}}{\partial v_{bs}} + C_{0BS} \cdot \left(1 + MJ \cdot \frac{v_{bs}}{PB}\right) \]
\[ + C_{0BS\_SW} \cdot \left(1 + MJ_{SW} \cdot \frac{v_{bs}}{PHP}\right) \]

Otherwise, if $(C_{0BS} + C_{0BS\_SW}) \leq 0$, then:

For $v_{bs} < 0$, 
\[ cap_{bs} = TT \cdot \frac{\partial i_{bs}}{\partial v_{bs}} + M \cdot CBS \cdot \left(1 - \frac{v_{bs}}{PB}\right)^{-MJ} \]

For $v_{bs} > 0$, 
\[ cap_{bs} = TT \cdot \frac{\partial i_{bs}}{\partial v_{bs}} + M \cdot CBS \cdot \left(1 + MJ \cdot \frac{v_{bs}}{PB}\right) \]

**Drain Diode Capacitance**

If $(C_{0BD} + C_{0BD\_SW}) > 0$, then:

For $v_{bd} < 0$, 
\[ cap_{bd} = TT \cdot \frac{\partial i_{bd}}{\partial v_{bd}} + C_{0BD} \cdot \left(1 - \frac{v_{bd}}{PB}\right)^{-MJ} \]
\[ + P_{Deff} \cdot C_{0BD\_SW} \cdot \left(1 - \frac{v_{bd}}{PHP}\right)^{-MJ_{SW}} \]

For $v_{bd} > 0$, 
\[ cap_{bd} = TT \cdot \frac{\partial i_{bd}}{\partial v_{bd}} + C_{0BD} \cdot \left(1 + MJ \cdot \frac{v_{bd}}{PB}\right) \]
\[ + C_{0BD\_SW} \cdot \left(1 + MJ_{SW} \cdot \frac{v_{bd}}{PHP}\right) \]
Otherwise, if \((A_{\text{Eff}} \cdot C_{\text{Jscaled}} + P_{\text{Eff}} \cdot C_{\text{JSWscaled}}) \leq 0\), then:

For \(v_{bd} < 0\),

\[
cap_{bd} = TT \cdot \frac{\partial i_{bd}}{\partial v_{bd}} + M \cdot CBD \cdot \left(1 - \frac{v_{bd}}{PB}\right)^{-MJ}
\]

For \(v_{bd} > 0\),

\[
cap_{bd} = TT \cdot \frac{\partial i_{bd}}{\partial v_{bd}} + M \cdot CBD \cdot \left(1 + MJ \cdot \frac{v_{bd}}{PB}\right)
\]
Using Common Threshold Voltage Equations

This section describes the common threshold voltage equations.

Common Threshold Voltage Parameters

The parameters described in this section are applicable to all MOSFET models except LEVELs 5 and 13.

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELVTO</td>
<td>V</td>
<td>0.0</td>
<td>Zero-bias threshold voltage shift</td>
</tr>
<tr>
<td>GAMMA</td>
<td>V(^{1/2})</td>
<td>0.527625</td>
<td>Body effect factor. If GAMMA is not set, it is calculated from NSUB.</td>
</tr>
<tr>
<td>NGATE</td>
<td>1/cm(^3)</td>
<td></td>
<td>Polysilicon gate doping, used for analytical model only. Undoped polysilicon is represented by a small value. If NGATE (\leq 0.0), it is set to (1e+18).</td>
</tr>
<tr>
<td>NSS</td>
<td>1/cm(^2)</td>
<td>1.0</td>
<td>Surface state density</td>
</tr>
<tr>
<td>NSUB (DNB, NB)</td>
<td>1/cm(^3)</td>
<td>1e15</td>
<td>Substrate doping</td>
</tr>
<tr>
<td>PHI</td>
<td>V</td>
<td>0.576036</td>
<td>Surface potential. NSUB default=1e15.</td>
</tr>
<tr>
<td>TPG (TPS)</td>
<td>1.0</td>
<td>Type of gate material, used for analytical model only \nLEVEL 4 TPG default=0 where \nTPG = 0 al-gate \nTPG = 1 gate type same as source-drain diffusion \nTPG = -1 gate type opposite to source-drain diffusion</td>
<td></td>
</tr>
<tr>
<td>VTO (VT)</td>
<td>V</td>
<td></td>
<td>Zero-bias threshold voltage</td>
</tr>
</tbody>
</table>
Calculating PHI, GAMMA, and VTO

The model parameters PHI, GAMMA, and VTO are used in threshold voltage calculations. If these parameters are not user-specified, they are calculated as follows, except for the LEVEL 5 model.

If PHI is not specified, then,

\[ PHI = 2 \cdot vt \cdot \ln\left(\frac{NSUB}{ni}\right) \]

If GAMMA is not specified, then,

\[ GAMMA = \frac{(2 \cdot q \cdot \varepsilon_s \cdot NSUB)^{1/2}}{COX} \]

The energy gap, eg, and intrinsic carrier concentration for the above equations are determined by:

\[ eg = 1.16 - 7.02e^{-4} \cdot \frac{tnom^2}{tnom + 1108} \]

\[ ni = 1.45e^{10} \cdot \left(\frac{tnom}{300}\right)^{3/2} \cdot e^{-\frac{2 \cdot eg}{k \cdot \frac{1}{300} - \frac{1}{tnom}}}(1/cm^3) \]

where,

\[ tnom = TNOM + 273.15 \]

If VTO is not specified, then for Al-Gate (TPG=0), the work function \( \Phi_{ms} \) is determined by:

\[ \Phi_{ms} = -\frac{eg}{2} - type \cdot \frac{PHI}{2} - 0.05 \]

where type is +1 for n-channel and -1 for p-channel.
For Poly-Gate (TPG=±1), the work function is determined by:

If the model parameter NGATE is not specified,

\[
\Phi_{ms} = type \cdot \left( -TPG \cdot \frac{eg}{2} - \frac{PHI}{2} \right)
\]

Otherwise,

\[
\Phi_{ms} = type \cdot \left[ -TPG \cdot vt \cdot \ln \left( \frac{NGATE}{ni} \right) - \frac{PHI}{2} \right]
\]

Then VTO voltage is determined by:

\[
VTO = vfb + type \cdot (\text{GAMMA} \cdot PHI^{1/2} + PHI)
\]

where,

\[
vfb = \Phi_{ms} - \frac{q \cdot NSS}{COX} + DELVTO
\]

If VTO is specified, then,

\[
VTO = VTO + DELVTO
\]
Performing MOSFET Impact Ionization

The impact ionization current for MOSFETs is available for all levels. The controlling parameters are ALPHA, VCR, and IIRAT. The parameter IIRAT sets the fraction of the impact ionization current that goes to the source.

\[
I_{ds} = I_{ds\_normal} + IIRAT \cdot I_{impact}
\]

\[
I_{db} = I_{db\_diode} + (1-IIRAT) \cdot I_{impact}
\]

IIRAT defaults to zero, which sends all impact ionization current to bulk. Leave IIRAT at its default value unless data is available for both drain and bulk current.

Using Impact Ionization Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>1/V</td>
<td>0.0</td>
<td>Impact ionization current coefficient</td>
</tr>
<tr>
<td>LALPHA</td>
<td>µm/V</td>
<td>0.0</td>
<td>ALPHA length sensitivity</td>
</tr>
<tr>
<td>WALPHA</td>
<td>µm/V</td>
<td>0.0</td>
<td>ALPHA width sensitivity</td>
</tr>
<tr>
<td>VCR</td>
<td>V</td>
<td>0.0</td>
<td>Critical voltage</td>
</tr>
<tr>
<td>LVCR</td>
<td>µm⋅V</td>
<td>0.0</td>
<td>VCR length sensitivity</td>
</tr>
<tr>
<td>WVCR</td>
<td>µm⋅V</td>
<td>0.0</td>
<td>VCR width sensitivity</td>
</tr>
<tr>
<td>IIRAT</td>
<td></td>
<td>0.0</td>
<td>Portion of impact ionization current that goes to source</td>
</tr>
</tbody>
</table>

Calculating the Impact Ionization Equations

The current \( I_{\text{impact}} \) due to impact ionization effect is calculated as follows:

\[
I_{\text{impact}} = I_{ds} \cdot \text{ALPHA}_{eff} \cdot (v_{ds} - v_{dsat}) \cdot e^{\frac{-VCR_{eff}}{v_{ds} - v_{dsat}}}
\]
where:

\[
ALPHA_{\text{eff}} = ALPHA + LALPHA \cdot 1e-6 \cdot \left( \frac{1}{L_{\text{eff}}} - \frac{1}{L_{\text{REFeff}}} \right) \\
+ WALPHA \cdot 1e-6 \cdot \left( \frac{1}{W_{\text{eff}}} - \frac{1}{W_{\text{REFeff}}} \right)
\]

\[
VCR_{\text{eff}} = VCR + LVCR \cdot 1e-6 \cdot \left( \frac{1}{L_{\text{eff}}} - \frac{1}{L_{\text{REFeff}}} \right) \\
+ WVCR \cdot 1e-6 \cdot \left( \frac{1}{W_{\text{eff}}} - \frac{1}{W_{\text{REFeff}}} \right)
\]

where:

\[
L_{\text{REFeff}} = L_{\text{REF}} + XL_{\text{REF}} - 2 \cdot LD \text{ and}
\]

\[
W_{\text{REFeff}} = W_{\text{REF}} + XW_{\text{REF}} - 2 \cdot WD
\]

**Calculating Effective Output Conductance**

The element template output allows \(g_{ds}\) to be output directly, for example:

```
.PRINT I(M1) gds=LX8(M1)
```

When using impact ionization current, it is important to note that \(g_{ds}\) is the derivative of \(I_{ds}\) only, rather than the total drain current, which is \(I_{ds} + I_{db}\). The complete drain output conductance is:

\[
g_{dd} = \frac{\partial I_d}{\partial V_d} = \frac{\partial I_{ds}}{\partial V_{ds}} + \frac{\partial I_{db}}{\partial V_{db}} = \frac{\partial I_{ds}}{\partial V_{ds}} + \frac{\partial I_{bd}}{\partial V_{bd}} = g_{ds} + g_{bd}
\]

\[
G_{dd} = LX8 + LX10
\]

For example, to print the drain output resistance of device M1:

```
.PRINT rout=PAR('1.0/(LX8(M1)+LX10(M1))')
```
Cascoding Example

Drain-to-bulk impact ionization current limits the use of cascoding to increase output impedance. The following cascode example shows the effect of changing IIRAT. When IIRAT is less than 1.0, the drain-to-bulk current lowers the output impedance of the cascode stage.

Figure 20-16: Low-frequency AC Analysis Measuring Output Impedance
Cascode Circuit

Example

\begin{tabular}{llll}
  iirat & gout\_ac & rout \\
  0.0 & 8.86E-6 & 113 K \\
  0.5 & 4.30E-6 & 233 K \\
  1.0 & 5.31E-8 & 18.8 Meg \\
\end{tabular}

Input File

\begin{verbatim}$ cascode test
.param pvds=5.0 pvref=1.4 pvin=3.0

vdd dd 0 pvds ac 1
$v current monitor vd
vd dd d 0
vin in 0 pvin
vref ref 0 pvref
x1 d in ref cascode
.macro cascode out in ref
m1 out in 1 0 n L=1u W=10u
mref 1 ref 0 0 n L=1u W=10u
.eom

.param xiirat=0
.ac dec 2 100k 1x sweep xiirat poi 3 0, 0.5, 1.0
.print ir(vd)
.measure gout\_ac avg ir(vd)

.model n nmos level=3
+ tox=200 vto=0.8 gamma=0.7 uo=600 kappa=0.05
+ alpha=1 vcr=15 iirat=xiirat
.end
\end{verbatim}
Using MOS Gate Capacitance Models

You can use capacitance model parameters with all MOSFET model statements. Model charge storage using fixed and nonlinear gate capacitances and junction capacitances. Gate-to-drain, gate-to-source, and gate-to-bulk overlap capacitances are represented by three fixed-capacitance parameters: CGDO, CGSO, and CGBO. The algorithm used for calculating nonlinear, voltage-dependent MOS gate capacitance depends on the value of model parameter CAPOP.

Model MOS gate capacitances, as a nonlinear function of terminal voltages, using Meyer’s piecewise linear model for all MOS levels. The charge conservation model is also available for MOSFET model LEVELs 2 through 7, 13, and 27. For LEVEL 1, the model parameter TOX must be specified to invoke the Meyer model. The Meyer, Modified Meyer, and Charge Conservation MOS Gate Capacitance models are described in detail in the following subsections.

Some of the charge conserving models (Ward-Dutton or BSIM) can cause "timestep too small" errors when no other nodal capacitances are present.

Selecting Capacitor Models

Gate capacitance model selection has been expanded to allow various combinations of capacitor models and DC models. Older DC models can now be incrementally updated with the new capacitance equations without having to move to a new DC model. You can select the gate capacitance with the CAPOP model parameter to validate the effects of different capacitance models.

The capacitance model selection parameter CAPOP is associated with the MOS models. Depending on the value of CAPOP, different capacitor models are used to model the MOS gate capacitance: the gate-to-drain capacitance, the gate-to-source capacitance, or the gate-to-bulk capacitance. CAPOP allows for the selection of several versions of the Meyer and charge conservation model.

Some of the capacitor models are tied to specific DC models (DC model level in parentheses below). Other models are designated as general and can be used by any DC model.

CAPOP=0 SPICE original Meyer model (general)
CAPOP=1  Modified Meyer model (general)
CAPOP=2  Parameterized Modified Meyer model (general default)
CAPOP=3  Parameterized Modified Meyer model with Simpson integration (general)
CAPOP=4  Charge conservation model (analytic), LEVELs 2, 3, 6, 7, 13, 28, and 39 only
CAPOP=5  No capacitor model
CAPOP=6  AMI capacitor model (LEVEL 5)
CAPOP=9  Charge conservation model (LEVEL 3)
CAPOP=13  Generic BSIM model (default for LEVELs 13, 28, 39)
CAPOP=11  Ward-Dutton model (specialized, LEVEL 2)
CAPOP=12  Ward-Dutton model (specialized, LEVEL 3)
CAPOP=39  BSIM2 Capacitance model (LEVEL 39)

CAPOP=4 selects the recommended charge-conserving model from among CAPOP=11, 12, or 13 for the given DC model.

### Table 20-4: CAPOP = 4 Selections

<table>
<thead>
<tr>
<th>MOS Level</th>
<th>Default CAPOP</th>
<th>CAPOP=4 selects</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>13, 28, 39</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Other levels</td>
<td>2</td>
<td>11</td>
</tr>
</tbody>
</table>

The proprietary models, LEVEL 5, 17, 21, 22, 25, 31, 33, and the SOS model LEVEL 27 have their own built-in capacitance routines.
Introducing Transcapacitance

If you have a capacitor with two terminals, 1 and 2 with charges Q1 and Q2 on the two terminals that sum to zero, for example, Q1=−Q2, the charge is a function of the voltage difference between the terminals, V12=V1−V2. The small-signal characteristics of the device are completely described by one quantity, C=dQ1/dV12.

If you have a four-terminal capacitor, the charges on the four terminals must sum to zero (Q1+Q2+Q3+Q4=0), and they can only depend on voltage differences, but they are otherwise arbitrary functions. So there are three independent charges, Q1, Q2, Q3, that are functions of three independent voltages V14, V24, V34. Hence there are nine derivatives needed to describe the small-signal characteristics.

It is convenient to consider the four charges separately as functions of the four terminal voltages, Q1(V1,V2,V3,V4), Q2(V1,V2,V3,V4), Q3(V1,V2,V3,V4), and Q4(V1,V2,V3,V4). The derivatives form a four by four matrix, dQi/dVj, i=1,4, j=1,4. This matrix has a direct interpretation in terms of AC measurements. If an AC voltage signal is applied to terminal j with the other terminals AC grounded, and AC current into terminal i is measured, the current is the imaginary constant times 2*pi*frequency times dQi/dVj.

The fact that the charges sum to zero requires each column of this matrix to sum to zero, while the fact that the charges can only depend on voltage differences requires each row to sum to zero.

In general, the matrix is not symmetrical:

\[ \frac{dQ_1}{dV_1} \neq \frac{dQ_1}{dV_2} \]

This is not an expected event because it does not occur for the two terminal case. For two terminals, the constraint that rows and columns sum to zero.

\[ \frac{dQ_1}{dV_1} + \frac{dQ_2}{dV_1} = 0 \]

forces dQ1/dV2 = dQ2/dV1. For three or more terminals, this relation does not hold in general.

The terminal input capacitances are the diagonal matrix entries.
C_{ii} = \frac{dQ_i}{dV_i} \quad i=1,4

and the transcapacitances are the negative of off-diagonal entries

C_{ij} = -\frac{dQ_i}{dV_j} \quad i \text{ not equal to } j

All of the Cs are normally positive.

**Figure 20-17: MOS Capacitances**

In Figure 20-17, C_{ij} determines the current transferred out of node i from a voltage change on node j. The arrows, representing direction of influence, point from node j to node i.

A MOS device with terminals D G S B provides:

\[ CGG = \frac{dQ_g}{dV_G} \]

\[ CGD = \frac{dQ_g}{dV_D} \]

\[ CDG = \frac{dQ_D}{dV_G} \]
CGG represents input capacitance: a change in gate voltage requires a current equal to CGG\(x\)dVG/dt into the gate terminal.

CGD represents Miller feedback: a change in drain voltage gives a current equal to CGG\(x\)dVG/dt out of the gate terminal.

CDG represents Miller feedthrough, capacitive current out of the drain due to a change in gate voltage.

To see how CGD might not be equal to CDG, the following example presents a simplified model with no bulk charge, with gate charge a function of VGS only, and 50/50 partition of channel charge into QS and QD:

\[
Q_G = Q(vgs) \\
Q_S = -0.5 \cdot Q(vgs) \\
Q_D = -0.5 \cdot Q(vgs) \\
Q_B = 0
\]

Consequently:

\[
CGD = \frac{dQ_G}{dV_D} = 0 \\
CGD = \frac{dQ_D}{dV_G} = 0.5 \cdot \frac{dQ}{dvgs}
\]

Therefore, in this model there is Miller feedthrough, but no feedback.

**Using the Operating Point Capacitance Printout**

Six capacitances are reported in the operating point printout:

- cdtot \(dQD/dVD\)
- cgtot \(dQG/dVG\)
- cstot \(dQS/dVS\)
These capacitances include gate-drain, gate-source, and gate-bulk overlap capacitance, and drain-bulk and source-bulk diode capacitance. Drain and source refer to node 1 and 3 of the MOS element, that is, physical instead of electrical.

For the Meyer models, where the charges like QD are not well defined, the printout quantities are:

\[
\begin{align*}
\text{cdtot} & \quad dQB/dVB \\
\text{cgs} & \quad -dQG/dVS \\
\text{cgd} & \quad -dQG/dVD \\
\end{align*}
\]

Using the Element Template Printout

The MOS element template printouts for gate capacitance are LX18 – LX23 and LX32 – LX34. From these nine capacitances the complete four-by-four matrix of transcapacitances can be constructed. The nine LX printouts are:

\[
\begin{align*}
\text{LX18 (m)} & \quad = dQG/dVGB = CGGBO \\
\text{LX19 (m)} & \quad = dQG/dVDB = CGDBO \\
\text{LX20 (m)} & \quad = dQG/dVSB = CGSBO \\
\text{LX21 (m)} & \quad = dQB/dVGB = CBGBO \\
\text{LX22 (m)} & \quad = dQB/dVDB = CBDBO \\
\text{LX23 (m)} & \quad = dQB/dVSB = CBSBO \\
\text{LX32 (m)} & \quad = dQD/dVG = CDGBO \\
\text{LX33 (m)} & \quad = dQD/dVD = CDDBO \\
\end{align*}
\]
LX34 (m) = \frac{dQ_D}{dV_S} = C_{DSO}

These capacitances include gate-drain, gate-source, and gate-bulk overlap capacitance, and drain-bulk and source-bulk diode capacitance. Drain and source refer to node 1 and 3 of the MOS element, that is, physical instead of electrical.

For an NMOS device with source and bulk grounded, LX18 represents the input capacitance, LX33 the output capacitance, -LX19 the Miller feedback capacitance (gate current induced by voltage signal on the drain), and -LX32 represents the Miller feedthrough capacitance (drain current induced by voltage signal on the gate).

A device operating with node 3 as electrical drain, for example an NMOS device with node 3 at higher voltage than node 1, is said to be in reverse mode. The LXs are physical, but you can translate them into electrical definitions by interchanging D and S:

- CGG(reverse) = CGG = LX18
- CDD(reverse) = CSS = \frac{dQ_S}{dV_S} = \frac{d(-Q_G-Q_B-Q_D)}{dV_S} = -LX20-LX23-LX34
- CGD(reverse) = CGS = -LX20
- CDG(reverse) = CSG = -\frac{dQ_S}{dV_G} = \frac{d(Q_G+Q_B+Q_D)}{dV_G} = LX18+LX21+LX32

For the Meyer models, the charges QD, and so forth, are not well defined. The formulas such as LX18= CGG, LX19= -CGD are still true, but the transcapacitances are symmetrical; for example, CGD=CDG. In terms of the six independent Meyer capacitances, cgd, cgs, cgb, cdb, csb, cds, the LX printouts are:

- LX18 (m) = CGS+CGD+CGB
- LX19 (m) = LX32 (m) = -CGD
- LX20 (m) = -CGS
- LX21 (m) = -CGB
- LX22 (m) = -CDB
- LX23 (m) = -CSB
- LX33 (m) = CGD+CDB+CDS
- LX34 (m) = -CDS
Calculating Gate Capacitance

The following example shows a gate capacitance calculation in detail for a BSIM model. TOX is chosen so that:

\[
\frac{e_{ox}}{tox} = 1 \cdot e - 3F/m^2
\]

Vfb0, phi, k1 are chosen so that vth=1v. The AC sweep is chosen so that
\[2 \cdot \pi \cdot freq = 1e6s^{-1}\] for the last point.

Star-Hspice Input File

```star-hspice
$m d g 0 b nch l=0.8u w=100u ad=200e-12 as=200e-12
vd d 0 5
vg g 0 5 ac 1
vb b 0 0
.ac dec 1 1.59155e4 1.59155e5
.print CGG=lx18(m) CDD=lx33(m) CGD=par('-lx19(m)')
+ CDG=par('-lx32(m)')
.print ig_imag=ii2(m) id_imag=ii1(m)
.model nch nmos level=13 update=2
+ xqc=0.6 toxm=345.315 vfb0=-1 phi0=1 k1=1.0 muz=600
+ mus=650 acm=2
+ xl=0 ld=0.1u meto=0.1u cj=0.5e-4 mj=0 cjsw=0
.alter
vd d 0 5 ac 1
vg g 0 5
.end
```

Calculations

\[Leff = 0.6u\]

\[
\frac{e_{ox}}{tox} = 1 \cdot e - 3F/m^2
\]
Introducing MOSFETs Using MOS Gate Capacitance Models

\[ \text{Cap} = \frac{\text{Leff} \cdot \text{Weff} \cdot eox}{\text{tox}} = 60e - 15F \]

BSIM equations for internal capacitance in saturation with xqc=0.4:

\[ \text{body} = 1 + 0.5 \cdot \left(1 - \frac{1}{(1.744 + 0.8364 \cdot (PHI0 + vsb))}\right) \cdot \frac{K1}{\sqrt{(PHI0 + vsb)}} \]

\[ 1 + 0.5 \cdot \left(1 - \frac{1}{(1.744 + 0.8364)}\right) = 1.3062 \]

\[ c_{gg} = \text{Cap} \cdot \left(1 - \frac{1}{(3 \cdot \text{body})}\right) = \text{Cap} \cdot 0.7448 = 44.69F \]

\[ c_{gd} = 0 \]

\[ c_{dg} = \left(\frac{4}{15}\right) \cdot \text{Cap} = 16F \]

\[ c_{dd} = 0 \]

Gate-drain overlap = \((ld + meto) \cdot \text{Weff} \cdot \frac{eox}{tox} = 20e - 15F\]

Adding the overlaps,

\[ c_{gg} = 44.69F + 2 \cdot 20F = 84.69F \]

\[ c_{gd} = 20F \]

\[ c_{dg} = 36F \]

\[ c_{dd} = 20F \]

\[ \text{Drain-bulk diode cap} c_{j \cdot ad} = (0.5e - 4) \cdot (200e - 12) = 1 \]

Adding the diodes,

\[ c_{gg} = 84.69F \]
\[ c_{gd} = 20F \]
\[ c_{dg} = 36F \]
\[ c_{dd} = 30F \]

**Star-Hspice Results**

```
subckt
 element 0:m
 model 0:nch
 cdtot  30.0000f
 cgtot  84.6886f
 cstot  74.4684f
 cbtot  51.8898f
 cgs    61.2673f
 cdg    20.0000f

 freq    cgg         cdd        cgd        cdg
  15.91550k  84.6886f    30.0000f  20.0000f   35.9999f
 159.15500k  84.6886f    30.0000f   20.0000f  35.9999f

 freq    ig_imag      id_imag
  15.91550k   8.4689n     -3.6000n
 159.15500k  84.6887n    -35.9999n

Alter results

 freq    ig_imag      id_imag
 15.91550k    -2.0000n    3.0000n
 159.15500k   -20.0000n  30.0000n
```

The calculation and the Star-Hspice results match.

**Plotting Gate Capacitances**

The following input file shows how to plot gate capacitances as a function of bias. Set the .OPTION DCCAP to turn on capacitance calculations for a DC sweep. The model used is the same as for the previous calculations.
**Example**

```verbatim
$ gate capacitance plots
.option dccap=1 post
m d g 0 b nch l=0.8u w=100u ad=200e-12 as=200e-12
vd d 0 0
vg g 0 5
vb b 0 0
.dc vd 0 5 .1
.print vds=v(d) CGG=lx18(m)
+ CGD=par('lx19(m)') CDG=par('-lx32(m)')
+ CGS=par('-lx20(m)') CSG=par('lx18(m)+lx21(m)+lx32(m)')
+ CGB=par('lx18(m)+lx19(m)+lx20(m)') CBG=par('-lx21(m)')

.model nch nmos
+ level=13 update=2 xqc=0.6 toxm=345.315
+ vfb0=-1 phi0=1 k1=1.0 muz=600 mus=650
+ acm=2 xl=0 ld=0.1u meto=0.1u
+ cj=0.5e-4 mj=0 cjsw=0
.end
```
Using Capacitance Control Options

The control options affecting the CAPOP models are SCALM, CVTOL, DCSTEP, and DCCAP. SCALM scales the model parameters, CVTOL controls the error tolerance for convergence for the CAPOP=3 model (see “Defining CAPOP=3 — Gate Capacitances (Simpson Integration)” on page 20-86). DCSTEP models capacitances with a conductance during DC analysis. DCCAP invokes calculation of capacitances in DC analysis.

Scaling

The parameters scaled by the option SCALM are CGBO, CGDO, CGSO, COX, LD, and WD. SCALM scales these parameters according to fixed rules that are a function of the parameter’s units. When the model parameter’s units are in meters, the parameter is multiplied by SCALM. For example, the parameter LD has units in meters, its scaled value is obtained by multiplying the value of LD by SCALM. When the units are in meters squared, the parameter is multiplied...
by SCALM^2. If the units are in reciprocal meters, the parameter’s value is divided by SCALM. For example, since CGBO is in farads/meter the value of CGBO is divided by SCALM. When the units are in reciprocal meters squared, then the parameter is divided by SCALM^2. The scaling equations specific to each CAPOP level are given in the individual CAPOP subsections.

### Using MOS Gate Capacitance Model Parameters

#### Using Basic Gate Capacitance Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPOP</td>
<td></td>
<td>2.0</td>
<td>Capacitance model selector</td>
</tr>
<tr>
<td>COX (CO)</td>
<td>F/m^2</td>
<td>3.453e-4</td>
<td>Oxide capacitance. If COX is not input, it is calculated from TOX. The default value corresponds to the TOX default of 1e-7: ( \text{COX}_{\text{scaled}} = \frac{\text{COX}}{\text{SCALM}^2} )</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>1e-7</td>
<td>Represents the oxide thickness, calculated from COX when COX is input. The program uses default if COX is not specified. For TOX&gt;1, unit is assumed to be Angstroms. But a level-dependent default can override it. See specific level in “Selecting MOSFET Models: LEVEL 1-40” on page 21-1</td>
</tr>
</tbody>
</table>
### Using Gate Overlap Capacitance Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGBO (CGB)</td>
<td>F/m</td>
<td>0.0</td>
<td>Gate-bulk overlap capacitance per meter channel length. If CGBO is not set but WD and TOX are set, then CGBO is calculated. CGBOscaled = CGBO/SCALM</td>
</tr>
<tr>
<td>CGDO (CGD, C2)</td>
<td>F/m</td>
<td>0.0</td>
<td>Gate-drain overlap capacitance per meter channel width. If CGDO is not set but LD or METO and TOX are set, then CGDO is calculated. CGDOscaled = CGDO/SCALM</td>
</tr>
<tr>
<td>CGSO (CGS, C1)</td>
<td>F/m</td>
<td>0.0</td>
<td>Gate-source overlap capacitance per meter channel width. If CGSO is not set but LD or METO and TOX are set, then CGSO is calculated. CGSOscaled = CGSO/SCALM</td>
</tr>
<tr>
<td>LD (LATD, DLAT)</td>
<td>m</td>
<td></td>
<td>Lateral diffusion into channel from source and drain diffusion. When both LD and XJ are unspecified: LD default=0.0. If LD is not set but XJ is specified, then LD is calculated from XJ. LD default=0.75 \cdot XJ for all levels except LEVEL 4, for which LD default=0.75. LIscaled = LD \cdot SCALM LEVEL 4: LDscaled = LD \cdot XJ \cdot SCALM</td>
</tr>
<tr>
<td>METO</td>
<td>m</td>
<td>0.0</td>
<td>Fringing field factor for gate-to-source and gate-to-drain overlap capacitance calculation METOscaled = METO \cdot SCALM</td>
</tr>
<tr>
<td>WD</td>
<td>m</td>
<td>0.0</td>
<td>Lateral diffusion into channel from bulk along width WDiscaled = WD \cdot SCALM</td>
</tr>
</tbody>
</table>
### Using Meyer Capacitance Parameters CAPOP=0, 1, 2

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF1</td>
<td>V</td>
<td>0.0</td>
<td>Modified MEYER control for transition of cgs from depletion to weak inversion for CGSO (only for CAPOP=2)</td>
</tr>
<tr>
<td>CF2</td>
<td>V</td>
<td>0.1</td>
<td>Modified MEYER control for transition of cgs from weak to strong inversion region (only for CAPOP=2)</td>
</tr>
<tr>
<td>CF3</td>
<td></td>
<td>1.0</td>
<td>Modified MEYER control for transition of cgs and cgd from saturation to linear region as a function of vds (only for CAPOP=2)</td>
</tr>
<tr>
<td>CF4</td>
<td></td>
<td>50.0</td>
<td>Modified MEYER control for contour of cgb and cgs smoothing factors</td>
</tr>
<tr>
<td>CF5</td>
<td></td>
<td>0.667</td>
<td>Modified MEYER control for capacitance multiplier for cgs in saturation region</td>
</tr>
<tr>
<td>CF6</td>
<td></td>
<td>500.0</td>
<td>Modified MEYER control for contour of cgd smoothing factor</td>
</tr>
<tr>
<td>CGBEX</td>
<td></td>
<td>0.5</td>
<td>cgb exponent (only for CAPOP=1)</td>
</tr>
</tbody>
</table>

### Using Charge Conservation Parameters (CAPOP=4)

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XQC</td>
<td></td>
<td>0.5</td>
<td>Coefficient of channel charge share attributed to drain; its range is 0.0 to 0.5. This parameter applies only to CAPOP=4 and some of its level-dependent aliases.</td>
</tr>
</tbody>
</table>
Specifying XQC and XPART for CAPOP=4, 9, 11, 12 and 13

Parameter rules for gate capacitance charge sharing coefficient, XQC & XPART, in the saturation region:

- If neither XPART or XQC is specified, the 0/100 model is used.
- If both XPART and XQC are specified, XPART overrides XQC.
- If XPART is specified:
  - XPART=0 → 40/60
  - XPART=0.4 → 40/60
  - XPART=0.5 → 50/50
  - XPART=1 → 0/100
  - XPART = any other value less than 1 → 40/60
  - XPART >1 → 0/100
- If XQC is specified:
  - XQC=0 → 0/100
  - XQC=0.4 → 40/60
  - XQC=0.5 → 50/50
  - XQC=1 → 0/100
  - XQC = any other value less than 1 → 40/60
  - XQC>1 → 0/100

The only difference is the treatment of the parameter value 0.

After XPART/XQC is specified, the gate capacitance is ramped from 50/50 at Vds=0 volt (linear region) to the value (with Vds sweep) in the saturation region specified by XPART/XQC. This charge sharing coefficient ramping ensures the smoothness of the gate capacitance characteristic.
Using Overlap Capacitance Equations

The overlap capacitors are common to all models. You can input them explicitly, or the program calculates them. These overlap capacitors are added into the respective voltage-variable capacitors before integration and the DC operating point reports the combined parallel capacitance.

**Gate-to-Bulk Overlap Capacitance**

If $CGBO$ is specified, then

\[
CGBO_{\text{eff}} = M \cdot Leff \cdot CGBO_{\text{scaled}}
\]

Otherwise,

\[
CGBO_{\text{eff}} = 2 \cdot WD_{\text{scaled}} \cdot Leff \cdot COX_{\text{scaled}} \cdot M
\]

**Gate-to-Source Overlap Capacitance**

If $CGSO$ is specified, then

\[
CGSO_{\text{eff}} = Weff \cdot CGSO_{\text{scaled}}
\]

Otherwise,

\[
CGSO_{\text{eff}} = Weff \cdot (LD_{\text{scaled}} + MET_{\text{scaled}}) \cdot COX_{\text{scaled}}
\]

**Gate-to-Drain Overlap Capacitance**

If $CGDO$ is specified, then

\[
CGDO_{\text{eff}} = Weff \cdot CGDO_{\text{scaled}}
\]

Otherwise,

\[
CGDO_{\text{eff}} = Weff \cdot (LD_{\text{scaled}} + MET_{\text{scaled}}) \cdot COX_{\text{scaled}}
\]
The $\text{Leff}$ is calculated for each model differently, and it is given in the corresponding model section. The $\text{Weff}$ calculation is not quite the same as $\text{weff}$ given in the model LEVEL 1, 2, 3, 6, 7 and 13 sections.

$$\text{Weff} = M \cdot (W_{scaled} \cdot WMLT + X_{scaled})$$

The $2 \cdot W_{scaled}$ factor is not subtracted.

**Defining CAPOP=0 — SPICE Meyer Gate Capacitances**

Definition:

$$\text{cap} = COX_{scaled} \cdot \text{Weff} \cdot \text{Leff}$$

**Gate-Bulk Capacitance (cgb)**

Accumulation, $vgs \leq vth - PH1$

$$cgb = cap$$

Depletion, $vgs < vth$

$$cgb = cap \cdot \frac{vth - vgs}{PH1}$$

Strong Inversion, $vgs \geq vth$

$$cgb = 0$$

**Gate-Source Capacitance (cgs)**

Accumulation, $vgs \leq vth - \frac{PH1}{2}$

$$cgs = 0$$

Depletion, $vgs \leq vth$

$$cgs = CF5 \cdot cap + \frac{cap \cdot (vgs - cth)}{0.75 \cdot PH1}$$
Introducing MOSFETs

Using MOS Gate Capacitance Models

Strong Inversion Saturation Region, \( v_{gs} > v_{th} \) and \( v_{ds} \geq v_{dsat} \)

\[
c_{gs} = CF5 \cdot cap
\]

Strong Inversion Linear Region, \( v_{gs} > v_{th} \) and \( v_{ds} < v_{dsat} \)

\[
c_{gs} = CF5 \cdot cap \cdot \left\{ 1 - \frac{v_{dsat} - v_{ds}}{2 \cdot (v_{dsat} + v_{sb}) - v_{ds} - v_{sb}} \right\}^2
\]

Gate-Drain Capacitance (c_{gd})

The gate-drain capacitance has value only in the linear region.

Strong Inversion Linear Region, \( v_{gs} > v_{th} \) and \( v_{ds} < v_{dsat} \).

\[
c_{gd} = CF5 \cdot cap \cdot \left\{ 1 - \frac{v_{dsat} + v_{sb}}{2 \cdot (v_{dsat} + v_{sb}) - v_{ds} - v_{sb}} \right\}^2
\]

Example

*file capop0.sp---capop=0 capacitances
*
*this file is used to create spice meyer gate c-v plots
**
*(capop=0) for low v_{ds} and high v_{ds}
*
.options acct=2 post=2 dccap=1 nomod
.dc vgl -1 4 .01
.print dc cgb_vdsp05=par(’-1x21(m1)’)
+ cgd_vdsp05=par(’-1x19(m1)’) cgs_vdsp05=par(’-1x20(m1)’)
.print dc cgb_vdsp8=par(’-1x21(m2)’) cgd_vdsp8=par(’-1x19(m2)’) cgs_vdsp8=par(’-1x20(m2)’)
************************************************************
m1 d1 q1 0 0 mn l=5e-6 w=20e-6 $ create capacitances for
+ vds=0.05
m2 d2 q1 0 0 mn l=5e-6 w=20e-6 $ create capacitances for
+ vds=0.80
************************************************************
vd1 d1 0 dc 0.05
vd2 d2 0 dc 0.80
vg1 g1 0 dc 0.0
*
*****************************************************************************
*
.model mn nmos ( level = 2
+ vto = 1.0 gamma = 1.40 nsub = 7.20e15
+ uo = 817 ucrit = 3.04e4 phi=.6
+ uexp = 0.102 neff = 1.74 vmax = 4.59e5
+ tox = 9.77e-8 cj = 0 cjsw = 0 js = 0
+ capop=0 )
.end

Figure 20-19: CAPOP=0 Capacitances
Defining CAPOP=1 — Modified Meyer Gate Capacitances

Define
\[ \text{cap} = \text{COXscaled} \cdot \text{Weff} \cdot \text{Leff} \]

In the following equations, \( G^- \), \( G^+ \), \( D^- \), and \( D^+ \) are smooth factors. They are not user-defined parameters.

**Gate-Bulk Capacitance (cgb)**

Accumulation, \( vgs \leq vfb - vsb \)
\[ cgb = \text{cap} \]

Depletion, \( vgs \leq vth \)
\[ cgb = \frac{\text{cap}}{1 + 4 \cdot \left( vgs + vsb - vfb \right)^{\text{CGBEX}}} \]

Strong Inversion, \( vgs > vth \)
\[ cgb = \frac{G^+ \cdot \text{cap}}{1 + 4 \cdot \text{GAMMA} \cdot \left( vsb + \text{PHI} \right)^2 + vsb + \text{PHI}}^{\text{CGBEX}} \]

**Note:** In the above equations, \( \text{GAMMA} \) is replaced by effective \( \gamma \) for model level higher than 4.

**Gate-Source Capacitance (cgs)**

Low \( vds \) (\( vds < 0.1 \))

Accumulation, \( vgs \leq vth \)
\[ cgs = CF5 \cdot \text{cap} \cdot G^- \cdot D^- \]
Weak Inversion, $v_{gs} < v_{th} + 0.1$

$$c_{gs} = CF_5 \cdot \text{cap} \cdot \left\{ \frac{v_{gs} - v_{th}}{0.1} \cdot \left[ 1 - \left( \frac{0.1 - v_{ds}}{0.2 - v_{ds}} \right)^2 \right] - D^- + D^- \right\}$$

Strong Inversion, $v_{gs} \geq v_{th} + 0.1$

$$c_{gs} = CF_5 \cdot \text{cap} \cdot \left\{ 1 - \left[ \frac{v_{gs} - v_{th} - v_{ds}}{2 \cdot (v_{gs} - v_{th}) - v_{ds}} \right]^2 \right\}$$

**High $v_{ds}$ ($v_{ds} \geq 0.1$)**

Accumulation, $v_{gs} \leq v_{th}$

$$c_{gs} = CF_5 \cdot \text{cap} \cdot G^-$$

Saturation Region, $v_{gs} < v_{th} + v_{ds}$

$$c_{gs} = CF_5 \cdot \text{cap}$$

Linear Region, $v_{gs} \geq v_{th} + v_{ds}$

$$c_{gs} = CF_5 \cdot \text{cap} \cdot \left\{ 1 - \left[ \frac{v_{gs} - v_{th} - v_{ds}}{2 \cdot (v_{gs} - v_{th}) - v_{ds}} \right]^2 \right\}$$

**Gate-Drain Capacitance ($c_{gd}$)**

**Low $v_{ds}$ ($v_{ds} < 0.1$)**

Accumulation, $v_{gs} \leq v_{th}$

$$c_{gd} = CF_5 \cdot \text{cap} \cdot G^- \cdot D^+$$
Introducing MOSFETs Using MOS Gate Capacitance Models

Weak Inversion, \( v_{gs} < v_{th} + 0.1 \)

\[
c_{gd} = CF5 \cdot cap \cdot \left\{ D^+ + \frac{v_{gs} - v_{gh}}{0.1} \cdot \max\left[ 0, 1 - \left( \frac{0.1}{0.2 - v_{ds}} \right)^2 - D^+ \right] \right\}
\]

Strong Inversion, \( v_{gs} \geq v_{th} + 0.1 \)

\[
c_{gd} = CF5 \cdot cap \cdot \max\left\{ D^*, 1 - \frac{v_{gs} - v_{th}}{2 \cdot (v_{gs} - v_{th}) - v_{ds}} \right\}^2
\]

**High v_{ds}(v_{ds} \geq 0.1)**

Accumulation, \( v_{gs} \leq v_{th} \)

\[
c_{gd} = CF5 \cdot cap \cdot G^- \cdot D^+
\]

Saturation Region, \( v_{gs} < v_{th} + v_{ds} \)

\[
c_{gd} = CF5 \cdot cap \cdot D^+
\]

Strong Inversion, \( v_{gs} \geq v_{th} + v_{ds} \)

\[
c_{gd} = CF5 \cdot cap \cdot \max\left\{ D^*, 1 - \frac{v_{gs} - v_{th}}{2 \cdot (v_{gs} - v_{th}) - v_{ds}} \right\}^2
\]

**Example**

*file capop1.sp---capop1 capacitances
*
*this file creates the modified meyer gate c-v plots
*(capop=1) for low v_{ds} and high v_{ds}.
*
.options acct=2 post=2 dccap=1 nomod
.dc vgl -1 4 .01
.print dc cgb_vdsp05=par(‘1x21(m1)’)
+ cgd_vdsp05=par(‘-lx19(m1)’)  
+ cgs_vdsp05=par(‘-lx20(m1)’)  
.print dc cgb_vdsp8=par(‘-lx21(m2)’)  
+ cgd_vdsp8=par(‘-lx19(m2)’)  
+ cgs_vdsp8=par(‘-lx20(m2)’)  
*******************************************  
m1 dl g1 0 0 mn l=5e-6 w=20e-6 $creates capacitances  
+ for vds=0.05  
m2 d2 g1 0 0 mn l=5e-6 w=20e-6 $creates capacitances  
+ for vds=0.80  
*******************************************  
vd1 d1 0 dc 0.05  
vd2 d2 0 dc 0.80  
vgl g1 0 dc 0.0  
*  
*******************************************  
*  
.model mn nmos ( level = 2  
+ vto = 1.0 gamma = 1.40 nsub = 7.20e15  
+ tox = 9.77e-8 uo = 817 ucrit = 3.04e4  
+ uexp = 0.102 neff = 1.74 vmax = 4.59e5  
+ phi = 0.6 cj = 0 cjsw = 0 js = 0  
+ capop=1 )  
.end
Defining CAPOP=2—Parameterized Modified Meyer Capacitances

The CAPOP=2 Meyer capacitance model is the more general form of Meyer capacitance. The CAPOP=1 Meyer capacitance model is the special case of CAPOP=2 when CF1=0, CF2=0.1, and CF3=1.

In the following equations, $G^-$, $G^+$, $D^-$, and $DD^+$ are smooth factors. They are not user-defined parameters.

Definition

\[ \text{cap} = \text{COXscaled} \cdot \text{Weff} \cdot \text{Leff} \]

Gate-Bulk Capacitance (cgb)

Accumulation, \( v_{gs} \leq v_{fb} - v_{sb} \)

\[ cgb = \text{cap} \]
Depletion, $v_{gs} \leq v_{th}$

$$c_{gs} = \frac{cap}{\left(1 + 4 \cdot \frac{v_{gs} + v_{sb} - v_{fb}}{GAMMA^2}\right)^{1/2}}$$

Inversion, $v_{gs} > v_{th}$

$$c_{gb} = \frac{G^+ \cdot cap}{\left[1 + 4 \cdot \frac{GAMMA \cdot (PHI + v_{sb})^{1/2} + PHI + v_{sb}}{GAMMA^2}\right]^{1/2}}$$

**Note:** In the above equations, GAMMA is replaced by effective $\gamma$ for model level higher than 4.

---

**Gate-Source Capacitance ($c_{gs}$)**

**Low $v_{ds}$ ($v_{ds} < 0.1$)**

Accumulation, $v_{gs} < v_{th} - CF1$

$$c_{gs} = CF5 \cdot cap \cdot G^- \cdot D^-$$

Depletion, $v_{gs} \leq v_{th} + CF2 - CF1$

$$c_{gs} = CF5 \cdot cap \cdot \left\{\frac{v_{gs} - v_{th} + CF1}{CF2} \left[1 - \left(\frac{v_{ds}}{2 \cdot CF2 - v_{ds}}\right)^2 - D^-\right] + D^-\right\}$$

Strong Inversion, $v_{gs} > v_{th} + \max (CF2 - CF1, CF3 \cdot v_{ds})$ UPDATE=0

Strong Inversion, $v_{gs} > v_{th} + CF2 - CF1, UPDATE=1$

$$c_{gs} = CF5 \cdot cap \cdot \left\{1 - \left[\frac{v_{gs} - v_{th} + CF1 - v_{ds}}{2 \cdot (v_{gs} - v_{th} + CF1) - v_{ds}}\right]^2\right\}$$
**High vds (vds ≥ 0.1)**

Accumulation, $vgs < vth - CF1$

$$cgs = CF5 \cdot cap \cdot G^- \cdot D^+, \quad CF1 \neq 0$$

$$cgs = CF5 \cdot cap \cdot G^-, \quad CF1 = 0$$

Weak Inversion, $vgs < vth + CF2 - CF1, \quad CF1 \neq 0$

$$cgs = CF5 \cdot cap \cdot \max \left( \frac{vgs - vth + CF1}{CF2}, D^+ \right)$$

Saturation Region, $vgs < vth + CF3 \cdot vds$

$$cgs = CF5 \cdot cap$$

Linear Region, $vgs > vth + CF3 \cdot vds$

$$cgs = CF5 \cdot cap \cdot \left\{1 - \left[2 \cdot \left(\frac{vgs - vth - vds}{vgs - vth - vds}\right)^2\right]\right\}, \quad \text{UPDATE}=0, \quad CF1=0$$

$$cgs = CG5 \cdot cap \cdot \left\{1 - \left[2 \cdot \left(\frac{vgs - vth - CF3 \cdot vds}{vgs - vth - CF3 \cdot vds}\right)^2\right]\right\}, \quad \text{UPDATE}=1$$

**Gate-Drain Capacitance (cgd)**

**Low vds, (vds < 0.1)**

Accumulation, $vgs \leq vth - CF1$

$$cgd = CF5 \cdot cap \cdot G^- \cdot D^-$$

Weak Inversion, $vgs < vth + CF2 - CF1$

$$cgd = CF5 \cdot cap \cdot \left[D^- + \frac{vgs - vth + CF1}{CF2} \cdot \max \left(0, 1 - \left(\frac{CF2}{vgs - vth - vds}\right)^2 - D^-\right)\right]$$
Strong Inversion, \( v_{gs} \geq v_{th} + CF2 - CF1 \)

\[
cgd = CF5 \cdot cap \cdot \max \left\{ D^- \cdot 1 - \left[ \frac{v_{gs} - v_{th} + CF1}{2 \cdot (v_{gs} - v_{th} + CF1) - v_{ds}} \right]^2 \right\}
\]

**High vds (vds > 0.1)**

Accumulation, \( v_{gs} \leq v_{th} - CF1 \)

\[
cgd = CF5 \cdot cap \cdot G^- \cdot DD^+
\]

Saturation Region, \( v_{gs} \leq v_{th} + CF3 \cdot v_{ds} \)

\[
cgd = CF5 \cdot cap \cdot DD^+
\]

**Note:** In the above equation, \( DD^+ \) is a function of \( CF3 \), if \( UPDATE=1 \).

Linear Region, \( v_{gs} > v_{th} + CF3 \cdot v_{ds} \)

\[
cgd = CF5 \cdot cap \cdot \max \left\{ DD^+, 1 - \left[ \frac{v_{gs} - v_{th}}{2 \cdot (v_{gs} - v_{th}) - CF3 \cdot v_{ds}} \right]^2 \right\}
\]

**Example**

```
*file capop2.sp capop=2 capacitances
*
*(this file creates parameterized modified gate capacitances for low and high vds).
*
.options acct=2 post=2 dccap=1 nomod
.dc vgl -1 4 .01
.print dc cgb_vdsp05=par(‘-lx21(m1)’) + cgd_vdsp05=par(‘-lx19(m1)’) cgs_vdsp05=par(‘-lx20(m1)’)
.print dc cgb_vdsp8=par(‘-lx21(m2)’) cgd_vdsp8=par(‘-lx19(m2)’) + cgs_vdsp8=par(‘-lx20(m2)’)
**************************************************
```
Introducing MOSFETs Using MOS Gate Capacitance Models

m1 d1 g1 0 0 mn l=5e-6 w=20e-6 $creates capacitances for + vds=0.05
m2 d2 g1 0 0 mn l=5e-6 w=20e-6 $creates capacitances for + vds=0.80
*******************************************
v10 d1 0 dc 0.05
vd2 d2 0 dc 0.80
vg1 g1 0 dc 0.0
*
*******************************************
*
.model mn nmos ( level = 2
+ vto = 1.0 gamma = 1.40 nsub = 7.20e15
+ tox = 9.77e-8 uo = 817 ucrir = 3.04e4
+ uexp = 0.102 neff = 1.74 phi = 0.6
+ vmax = 4.59e5 cj = 0 cjsw = 0 js = 0
+ capop=2 cf1=0.15 cf2=.2 cf3=.8 cf5=.666)
.end

Figure 20-21: CAPOP=2 Capacitances
Defining CAPOP=3 — Gate Capacitances (Simpson Integration)

The CAPOP 3 model is the same set of equations and parameters as the CAPOP 2 model. The charges are obtained by Simpson numeric integration instead of the box integration found in CAPOP models 1, 2, and 6.

Gate capacitances are not constant values with respect to voltages. The capacitance values can best be described by the incremental capacitance:

\[ C(v) = \frac{dq(v)}{dv} \]

where \( q(v) \) is the charge on the capacitor and \( v \) is the voltage across the capacitor.

The formula for calculating the differential is difficult to derive. Furthermore, the voltage is required as the accumulated capacitance over time. The timewise formula is:

\[ i(t) = \frac{dq(v)}{dt} = C(v) \cdot \frac{dv(t)}{dt} \]

The charge is:

\[ q(v) = \int_{0}^{v} C(v)dv \]

For the calculation of current:

\[ i(t) = \frac{dq(v)}{dt} = \left( \frac{d}{dt} \right) \int_{0}^{v} C(v)dv \]

For small intervals:

\[ I(n + 1) = \frac{dq(v)}{dt} = \frac{1}{t(n + 1) - t(n)} \int_{V(n)}^{V(n + 1)} C(v)dv \]
The integral has been approximated in SPICE by:

\[ I(n + 1) = \left( \frac{V(n + 1) - V(n)}{t(n + 1) - t(n)} \right) \cdot \left( \frac{C[V(n + 1)] + C[V(n)]}{2} \right) \]

This last formula is the trapezoidal rule for integration over two points. The charge is approximated as the average capacitance times the change in voltage. If the capacitance is nonlinear, this approximation can be in error. To estimate the charge accurately, use Simpson’s numerical integration rule. This method provides charge conservation control.

To use this model, set the model parameter CAPOP to 3 and use the existing CAPOP=2 model parameters. Modify the OPTIONS settings RELV (relative voltage tolerance), RELMOS (relative current tolerance for MOSFETs), and CVTOL (capacitor voltage tolerance). The default of 0.5 is a good nominal value for CVTOL. The option CVTOL sets the number of integration steps with the formula:

\[ n = \frac{|V(n + 1) - V(n)|}{CVTOL} \]

Using a large value for CVTOL decreases the number of integration steps for the time interval n to n+1; this yields slightly less accurate integration results. Using a small CVTOL value increases the computational load, and in some instances, severely.

**Defining CAPOP=4 — Charge Conservation Capacitance Model**

The charge conservation method (See Ward, Donald E. and Robert W. Dutton “A Charge-Oriented Model for MOS Transistor”) is not implemented correctly into the SPICE2G.6 program. There are errors in the derivative of charges, especially in LEVEL 3 models. Also, channel charge partition is not continuous going from linear to saturation regions.

In Star-Hspice, these problems are corrected. By specifying model parameter CAPOP=4, the level-dependent recommended charge conservation model is
selected. The ratio of channel charge partitioning between drain and source is selected by the model parameter XQC. For example, if XQC=.4 is set, then the saturation region 40% of the channel charge is associated to drain and the remaining 60% is associated to the source. In the linear region, the ratio is 50/50. In Star-Hspice, an empirical equation is used to make a smooth transition from 50/50 (linear region) to 40/60 (saturation region).

Also, the capacitance coefficients, which are the derivative of gate, bulk, drain, and source charges, are continuous. Model LEVELs 2, 3, 4, 6, 7, and 13 have a charge conservation capacitance model that is invoked by setting CAPOP=4.

In the following example, only the charge conservation capacitance CAPOP=4 and the improved charge conservation capacitance CAPOP=9 for the model LEVEL 3 are compared. The capacitances CGS and CGD for CAPOP=4 model (SPICE2G.6) show discontinuity at the saturation and linear region boundary while the CAPOP=9 model does not have discontinuity. For the purpose of comparison, the modified Meyer capacitances (CAPOP=2) also is provided. The shape of CGS and CGD capacitances resulting from CAPOP=9 are much closer to those of CAPOP=2.

**Example**

```
FILE MCAP3.SP CHARGE CONSERVATION MOSFET CAPS., CAPOP=4,9
LEVEL=3
*
  * CGGB = LX18(M) DERIVATIVE OF QG WITH RESPECT TO VGB.
  * CGDB = LX19(M) DERIVATIVE OF QG WITH RESPECT TO VDB.
  * CGSB = LX20(M) DERIVATIVE OF QG WITH RESPECT TO VSB.
  * CBGB = LX21(M) DERIVATIVE OF QB WITH RESPECT TO VGB.
  * CBDB = LX22(M) DERIVATIVE OF QB WITH RESPECT TO VDB.
  * CBSB = LX23(M) DERIVATIVE OF QB WITH RESPECT TO VSB.
  * CGDB = LX32(M) DERIVATIVE OF QD WITH RESPECT TO VGB.
  * CDBD = LX33(M) DERIVATIVE OF QD WITH RESPECT TO VDB.
  * CDSB = LX34(M) DERIVATIVE OF QD WITH RESPECT TO VSB.
  * THE SIX NONRECIPOCAL CAPACITANCES CGB, CBG, CGS, CSG, CGD, AND CDG
    * ARE DERIVED FROM THE ABOVE CAPACITANCE FACTORS.
  *
  .OPTIONS DCCAP=1 POST NOMOD
```
.PARAM XQC=0.4 CAPOP=4
.DC VGG -2 5 .02
.print CGB=PAR('LX18(M)+LX19(M)+LX20(M)')
  + CBG=PAR('LX21(M)')
  + CGS=PAR('LX20(M)')
  + CSG=PAR('LX18(M)+LX21(M)+LX32(M)')
  + CGD=PAR('LX19(M)')
  + CDG=PAR('LX32(M)')
.print
  + CG =par('LX14(M)')
VDD D 0 2.5
VGG G 0 0
VBB B 0 -1
M D G 0 B MOS W=10U L=5U
.MODEL MOS NMOS LEVEL=3 COX=1E-4 VTO=.3 CAPOP=CAPOP
  + UO=1000 GAMMA=.5 PHI=.5 XQC=XQC
  + THETA=0.06 VMAX=1.9E5 ETA=0.3 DELTA=0.05 KAPPA=0.5 XJ=.3U
  + CGS=0 CGD=0 CGBO=0 CJ=0 JS=0 IS=0
*
.ALTER
.PARAM CAPOP=9
.END
Figure 20-22: CAPOP=4, 9 Capacitances for LEVEL 3 Model

Figure 20-23: CAPOP=2 Capacitances for LEVEL 3 Model
The following example tests the charge conservation capacitance model (Yang, P., B.D. Epler, and P.K. Chaterjee 'An Investigation of the Charge Conservation Problem) and compares the Meyer model and charge conservation model. As the following graph illustrates, the charge conservation model gives more accurate results.

**Example**

```plaintext
*FILE:CHRGPUMP.SP CHARGE CONSERVATION TEST FOR CHARGE PUMP CIRCUIT
*TEST CIRCUIT OF A MOSFET CAPACITOR AND A LINEAR CAPACITOR
.OPTIONS ACCT LIST NOMOD POST
+ RELTOL=1E-3 ABSTOL=1E-6 CHGTOL=1E-14
.PARAM CAPOP=2
.OP
.TRAN 2NS 470NS SWEEP CAPOP POI 2 2,9
.IC V(S)=1
*
VIN G 0 PULSE 0 5 15NS 5NS 50NS 100NS
VBB 0 B PULSE 0 5 0NS 5NS 50NS 100NS
VDD D D- PULSE 0 5 25NS 5NS 50NS 100NS
*
RC D- S 10K
C2 S 0 10P
M1 D G S B MM W=3.5U L=5.5U
+AD=100P AS=100P PD=50U PS=50U NRD=1 NRS=1
*
.MODELM M NMOS LEVEL=3 VTO=0.7 KP=50E-6 GAMMA=0.96
+PHI=0.5763 TOX=50E-9 NSUB=1.0E16 LD=0.5E-6
+VMAX=268139 THETA=0.05 ETA=1 KAPPA=0.5 CJ=1E-4
+CJSW=0.05E-9 RSH=20 JS=1E-8 PB=0.7
+CGD=0 CGS=0 IS=0 JS=0
+CAPOP=CAPOP
*
.PRINT TRAN VOUT=V(S) VIN=V(D) VBB=V(B)
+ VDD=V(D,D-)
.END
```
Using MOS Gate Capacitance Models

Introducing MOSFETs

Figure 20-24: Charge Pump Circuit

Figure 20-25: Charge Conservation Test: CAPOP=2 or 9
The following example applies a pulse through a constant capacitance to the gate of MOS transistor. Ideally, if the model conserves charge, the voltage at node 20 should become zero when the input pulse goes to zero. Consequently, the model that provides voltage closer to zero for node 20 conserves the charge better. As results indicate, the CAPOP=4 model is better than the CAPOP=2 model.

This example also compares the charge conservation models in SPICE2G.6 and Star-Hspice. The results indicate that Star-Hspice is more accurate.

**Example**

FILE MCAP2_A.SP

```plaintext
.OPTIONS SPICE NOMOD DELMAX=.25N
.PARAM CAPOP=4
.TRAN 1NS 40NS SWEEP CAPOP POI 2 4 2
.PRINT TRAN V(1) V(20)
VIN 1 0 PULSE (0V, 5V, 0NS, 5NS, 5NS, 5NS, 20NS)
CIN 1 20 1PF
RLEAK 20 0 1E+12
VDD 10 0 1.3
VBB 30 0 -1
M 10 20 0 30 MOS W=10U L=5U
.MODEL MOS NMOS LEVEL=2 TOX=250E-10 VTO=.3 + UO=1000 LAMBDA=1E-3 GAMMA=.5 PHI=.5 XQC=.5 + THETA=0.067 VMAX=1.956E5 XJ=.3U + CGSO=0 CGDO=0 CGBO=0 + CJ=0 JS=0 IS=0 + CAPOP=CAPOP
.END
```
Defining CAPOP=5 — Gate Capacitance

Use CAPOP=5 for no capacitors, and Star-Hspice will not calculate gate capacitance.

Defining CAPOP=6 — AMI Gate Capacitance Model

Define:

\[
vgst = vgs - \frac{(vth + vfb)}{2}
\]

\[
cox = \frac{\varepsilon_{ox}}{TOX \cdot 1e-10} \cdot W_{eff} \cdot L_{eff}
\]

The gate capacitance \( cgs \) is calculated according to the equations below in the different regions.
\[ 0.5 \cdot (v_{th} + v_{fb}) > v_{gs} \]
\[ c_{gs} = 0 \]

\[ 0.5 \cdot (v_{th} + v_{fb}) < v_{gs} < v_{th} \]
For \( v_{gst} < v_{ds} \),
\[ c_{gs} = \frac{4}{3} \cdot \frac{\text{cox} \cdot v_{gst}}{v_{th} - v_{fb}} \]

For \( v_{gst} > v_{ds} \),
\[ c_{gs} = \text{arg} \cdot \frac{4}{3} \cdot \frac{\text{cox} \cdot v_{gst}}{v_{th} - v_{fb}} \]

\[ v_{gs} > v_{th} \]
For \( v_{gst} < v_{ds} \),
\[ c_{gs} = \frac{2}{3} \cdot \text{cox} \]

For \( v_{gst} > v_{ds} \),
\[ c_{gs} = \text{arg} \cdot \frac{2}{3} \cdot \text{cox} \]
\[ \text{arg} = v_{gst} \cdot \frac{(3 \cdot v_{gst} - 2 \cdot v_{ds})}{(2 \cdot v_{gst} - v_{ds})^2} \]

The gate capacitance \( c_{gd} \) is calculated according to the equations below in the different regions.

\[ v_{gs} < v_{th} \]
\[ c_{gd} = 0 \]

\[ v_{gs} > v_{th} \text{ and } v_{gst} < v_{ds} \]
\[ c_{gd} = 0 \]
Using MOS Gate Capacitance Models

Introducing MOSFETs

\( vgs > vth \) and \( vgst > vds \)

\[
c_{gd} = arg \cdot \frac{2}{3} \cdot C_{ox}
\]

\[
arg = (3 \cdot vgst - vds) \cdot \frac{(vgst - vds)}{(2 \cdot vgst - vds)^2}
\]

The gate capacitance \( c_{gb} \) is combined with the calculation of both oxide capacitance and depletion capacitance as shown below.

\[
c_{gb} = \frac{c_{gbx} \cdot c_{d}}{c_{gbx} + c_{d}}
\]

Oxide capacitance \( c_{gbx} \), is calculated as:

\[
c_{gbx} = C_{ox} - c_{gs} - c_{gd}
\]

Depletion capacitance \( c_{d} \) is voltage-dependent.

\[
c_{d} = \frac{\varepsilon_s}{w_d} \cdot W_{eff} \cdot L_{eff}
\]

\[
w_d = \left( \frac{2 \cdot \varepsilon_s \cdot v_c}{q \cdot N_{SUB}} \right)^{1/2}
\]

\[
v_c = \text{The effective voltage from channel to substrate (bulk)}
\]

The following shows the equations for \( v_c \) under various conditions:

**\( vgs + vsb < vfb \)**

\[
v_c = 0
\]

**\( vgs + vsb > vfb \)**

\[
v_c = vgs + vsb - vfb
\]

**\( vgst > 0, vgs < vth, vgst < vds \)**

\[
v_c = \frac{1}{2} \cdot (vth - vfb) + \frac{3}{2} \cdot vgst + vsb
\]
Introducing MOSFETs Using MOS Gate Capacitance Models


 vgst > 0, vgs < vth, vgst > vds

\[ v_{c} = \frac{1}{2} \cdot (v_{th} - v_{fb}) + v_{gst} + \frac{1}{2} \cdot v_{ds} + v_{sb} \]

 vgsg > vth, vgst < vds

\[ v_{c} = v_{th} - v_{fb} + \frac{1}{2} \cdot v_{gst} + v_{sb} \]

 vgsg > vth, vgst > vds

\[ v_{c} = v_{th} - v_{fb} + \frac{1}{2} \cdot v_{ds} + v_{sb} \]

**Defining CAPOP=13 — BSIM1-based Charge-Conserving Gate Capacitance Model**


**Defining CAPOP=39 — BSIM2 Charge-Conserving Gate Capacitance Model**

See “LEVEL 39 BSIM2 Model” on page 21-179.

**Calculating Effective Length and Width for AC Gate Capacitance**

For some MOS processes and parameter extraction methods, it is helpful to allow different Leff and Weff values for AC analysis than for DC analysis. For AC gate capacitance calculations, substitute model parameters LDAC and WDAC for LD and WD in Leff and Weff calculations. You can use LD and WD in Leff and Weff calculations for DC current.
To use LDAC and WDAC, enter XL, LD, LDAC, XW, WD, WDAC in the .MODEL statement. The model uses the following equations for DC current calculations.

\[
\begin{align*}
    L_{eff} &= L + XL - 2 \cdot LD \\
    W_{eff} &= W + XW - 2 \cdot WD
\end{align*}
\]

and uses the following equations for AC gate capacitance calculations

\[
\begin{align*}
    L_{eff} &= L + XL - 2 \cdot LD_{AC} \\
    W_{eff} &= W + XW - 2 \cdot WD_{AC}
\end{align*}
\]

The noise calculations use the DC \( W_{eff} \) and \( L_{eff} \) values.

Use LDAC and WDAC with the standard Star-Hspice parameters XL, LD, XW, and WD. Do not use LDAC and WDAC with other parameters such as DL0 and DW0.
Using Noise Models

This section describes how to use noise models.

Using Noise Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>1.0</td>
<td></td>
<td>Flicker noise exponent</td>
</tr>
<tr>
<td>KF</td>
<td>0.0</td>
<td></td>
<td>Flicker noise coefficient. Reasonable values for KF are in the range 1e-19 to 1e-25 V^2/F.</td>
</tr>
<tr>
<td>NLEV</td>
<td>2.0</td>
<td></td>
<td>Noise equation selector</td>
</tr>
<tr>
<td>GDSNOI</td>
<td>1.0</td>
<td></td>
<td>Channel thermal noise coefficient (use with NLEV=3)</td>
</tr>
</tbody>
</table>

Using Noise Equations

The Star-Hspice MOSFET model noise equations have a selector parameter NLEV that is used to select either the original SPICE flicker noise or an equation proposed by Gray and Meyer.

Thermal noise generation in the drain and source resistors is modeled by the two sources inrd and inrs (units amp/(Hz)^1/2), as shown in Figure 20-10. The values of these sources can be determined by:

\[
inrs = \left(\frac{4kT}{rs}\right)^{1/2}
\]

\[
inrd = \left(\frac{4kT}{rd}\right)^{1/2}
\]
Channel thermal noise and flicker noise are modeled by the current source ind and defined by the equation:

\[ \text{ind}^2 = (\text{channel thermal noise})^2 + (\text{flicker noise})^2 \]

If the model parameter NLEV is less than 3, then

\[ \text{channel thermal noise} = \left( \frac{8kT \cdot gm}{3} \right)^{1/2} \]

The above formula is used in both saturation and linear regions, which can lead to wrong results in the linear region. For example, at VDS=0, channel thermal noise becomes zero because gm=0. This calculation is physically impossible. If NLEV model parameter is set to 3, Star-Hspice uses a different equation which is valid in both linear and saturation regions. See Tsividis, Yanis P., *Operation and Modeling of the MOS Transistor*, McGraw-Hill, 1987, p. 340.

For NLEV=3,

\[ \text{channel thermal noise} = \left( \frac{8kt}{3} \cdot \beta \cdot (vgs - vth) \cdot \frac{1 + a + a^2}{1 + a} \cdot GDSNOI \right)^{1/2} \]

where

\[ a = 1 - \frac{vds}{vdsat} \quad \text{Linear region} \]

\[ a = 0 \quad \text{Saturation region} \]

The two parameters AF and KF are used in the small-signal AC noise analysis to determine the equivalent flicker noise current generator connected between drain and source.

**NLEV=0 (SPICE):**

\[ \text{flicker noise} = \left( \frac{KF \cdot Ids^{AF}}{COX \cdot Leff^2 \cdot f} \right)^{1/2} \]

For NLEV=1 the Leff^2 in the above equation is replaced by Weff \cdot Leff.
NLEV=2, 3:

\[
\text{flicker noise} = \left( \frac{KF \cdot gm^2}{\text{COX} \cdot \text{Weff} \cdot \text{Leff} \cdot f^2} \right)^{1/2}
\]

**Noise Summary Printout Definitions**

- **RD, V^2/Hz**: Output thermal noise due to drain resistor
- **RS, V^2/Hz**: Output thermal noise due to source resistor
- **RX**: Transfer function of channel thermal or flicker noise to the output. This is not a noise, it is a transfer coefficient, reflecting the contribution of channel thermal or flicker noise to the output.
- **ID, V^2/Hz**: Output channel thermal noise: \( ID = RX^2 \cdot (\text{channel thermal noise})^2 \)
- **FN, V^2/Hz**: Output flicker noise: \( FN = RX^2 \cdot (\text{flicker noise})^2 \)
- **TOT, V^2/Hz**: Total output noise: \( TOT = RD + RS + ID + FN \)
Using Temperature Parameters and Equations

Temperature Parameters

The following temperature parameters apply to all MOSFET model levels and the associated bulk-to-drain and bulk-to-source MOSFET diode within the MOSFET model. The temperature equations used for the calculation of temperature effects on the model parameters are selected by the TLEV and TLEVC parameters.

Temperature Effects Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEX</td>
<td></td>
<td>-1.5</td>
<td>Low field mobility, UO, temperature exponent</td>
</tr>
<tr>
<td>CTA</td>
<td>1/K</td>
<td>0.0</td>
<td>Junction capacitance CJ temperature coefficient. Set TLEVC to 1 to enable CTA to override default Star-Hspice temperature compensation.</td>
</tr>
<tr>
<td>CTP</td>
<td>1/K</td>
<td>0.0</td>
<td>Junction sidewall capacitance CJSW temperature coefficient. Set TLEVC to 1 to enable CTP to override default Star-Hspice temperature compensation.</td>
</tr>
<tr>
<td>EG</td>
<td>eV</td>
<td></td>
<td>Energy gap for pn junction diode. Set default=1.11, for TLEV=0 or 1 and default=1.16, for TLEV=2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.17 – silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.69 – Schottky barrier diode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.67 – germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.52 – gallium arsenide</td>
</tr>
<tr>
<td>F1EX</td>
<td></td>
<td>0</td>
<td>Bulk junction bottom grading coefficient</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>GAP1</td>
<td>eV/°K</td>
<td>7.02e-4</td>
<td>First bandgap correction factor (from Sze, alpha term)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7.02e-4 – silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.73e-4 – silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.56e-4 – germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.41e-4 – gallium arsenide</td>
</tr>
<tr>
<td>GAP2</td>
<td>°K</td>
<td>1108</td>
<td>Second bandgap correction factor (from Sze, beta term)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1108 – silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>636 – silicon</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>210 – germanium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>204 – gallium arsenide</td>
</tr>
<tr>
<td>LAMEX</td>
<td>1/°K</td>
<td>0</td>
<td>LAMBDA temperature coefficient</td>
</tr>
<tr>
<td>N</td>
<td></td>
<td>1.0</td>
<td>Emission coefficient</td>
</tr>
<tr>
<td>MJ</td>
<td></td>
<td>0.5</td>
<td>Bulk junction bottom grading coefficient</td>
</tr>
<tr>
<td>MJSW</td>
<td></td>
<td>0.33</td>
<td>Bulk junction sidewall grading coefficient</td>
</tr>
<tr>
<td>PTA</td>
<td>V/°K</td>
<td>0.0</td>
<td>Junction potential PB temperature coefficient. Set TLEVC to 1 or 2 to enable PTA to override default Star-Hspice temperature compensation.</td>
</tr>
<tr>
<td>PTC</td>
<td>V/°K</td>
<td>0.0</td>
<td>Fermi potential PHI temperature coefficient. Set TLEVC to 1 or 2 to enable PTC to override default Star-Hspice temperature compensation.</td>
</tr>
<tr>
<td>PTP</td>
<td>V/°K</td>
<td>0.0</td>
<td>Junction potential PHP temperature coefficient. Set TLEVC to 1 or 2 to enable PTP to override default Star-Hspice temperature compensation.</td>
</tr>
<tr>
<td>TCV</td>
<td>V/°K</td>
<td>0.0</td>
<td>Threshold voltage temperature coefficient. Typical values are +1mV for n-channel and -1mV for p-channel.</td>
</tr>
</tbody>
</table>
Using Temperature Parameters and Equations

Introducing MOSFETs

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLEV</td>
<td></td>
<td>0.0</td>
<td>Temperature equation level selector. Set TLEV=1 for ASPEC style – default is SPICE style. When option ASPEC is invoked, the program sets TLEV for ASPEC.</td>
</tr>
<tr>
<td>TLEVC</td>
<td></td>
<td>0.0</td>
<td>Temperature equation level selector for junction capacitances and potentials, interacts with TLEV. Set TLEVC=1 for ASPEC style. Default is SPICE style. When option ASPEC is invoked, the program sets TLEVC for ASPEC.</td>
</tr>
<tr>
<td>TRD</td>
<td>1/K°</td>
<td>0.0</td>
<td>Temperature coefficient for drain resistor</td>
</tr>
<tr>
<td>TRS</td>
<td>1/K°</td>
<td>0.0</td>
<td>Temperature coefficient for source resistor</td>
</tr>
<tr>
<td>XTI</td>
<td></td>
<td>0.0</td>
<td>Saturation current temperature exponent. Use XTI=3 for silicon diffused junction. Set XTI=2 for Schottky barrier diode.</td>
</tr>
</tbody>
</table>

Using MOS Temperature Coefficient Sensitivity Parameters

Model levels 13 (BSIM1), 39 (BSIM2), and 28 (METAMOS) have length and width sensitivity parameters associated with them as shown in the following table. These parameters are used in conjunction with the Automatic Model Selector capability and enable more accurate modeling for various device sizes.
The default value of each sensitivity parameter is zero to ensure backward compatibility.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Sensitivity Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Length</td>
</tr>
<tr>
<td>BEX</td>
<td>Low field mobility, UO, temperature exponent</td>
<td>LBEX</td>
</tr>
<tr>
<td>FEX</td>
<td>Velocity saturation temperature exponent</td>
<td>LFEX</td>
</tr>
<tr>
<td>TCV</td>
<td>Threshold voltage temperature coefficient</td>
<td>LTCV</td>
</tr>
<tr>
<td>TRS</td>
<td>Temperature coefficient for source resistor</td>
<td>LTRS</td>
</tr>
<tr>
<td>TRD</td>
<td>Temperature coefficient for drain resistor</td>
<td>LTRD</td>
</tr>
</tbody>
</table>

**Using Temperature Equations**

This section describes how to use temperature equations.

**Calculating Energy Gap Temperature Equations**

To determine energy gap for temperature compensation use the equations:

\[
T_{LEV} = 0 \text{ or } 1:
\]

\[
e_{gnom} = 1.16 - 7.02 \times 10^{-4} \cdot \frac{t_{nom}^2}{t_{nom} + 1108.0}
\]

\[
e_{g(t)} = 1.16 - 7.02 \times 10^{-4} \cdot \frac{t^2}{t + 1108.0}
\]
TLEV = 2:

\[ e_{\text{egnom}} = E-G - \text{GAP1} \cdot \frac{t_{\text{nom}}^2}{t_{\text{nom}} + \text{GAP2}} \]

\[ e_{g(t)} = E-G - \text{GAP1} \cdot \frac{t^2}{t + \text{GAP2}} \]

Calculating Saturation Current Temperature Equations

\[ isbd(t) = isbd(t_{\text{nom}}) \cdot e_{\text{facln}/N} \]

\[ isbs(t) = isbs(t_{\text{nom}}) \cdot e_{\text{facln}/N} \]

where

\[ \text{facln} = \frac{e_{\text{egnom}}}{\text{vt}(t_{\text{nom}})} - \frac{e_{g(t)}}{\text{vt}(t)} + XTI \cdot \ln\left(\frac{t}{t_{\text{nom}}}\right) \]

These \(isbd\) and \(isbs\) are defined in “Using a MOSFET Diode Model” on page 20-26.

Calculating MOS Diode Capacitance Temperature Equations

TLEVC selects the temperature equation level for MOS diode capacitance.

TLEVC=0:

\[ PB(t) = PB \cdot \left(\frac{t}{t_{\text{nom}}}\right) - \text{vt}(t) \cdot \left[3 \cdot \ln\left(\frac{t}{t_{\text{nom}}}\right) + \frac{e_{\text{egnom}}}{\text{vt}(t_{\text{nom}})} - \frac{e_{g(t)}}{\text{vt}(t)}\right] \]

\[ PHP(t) = PHP \cdot \left(\frac{t}{t_{\text{nom}}}\right) - \text{vt}(t) \cdot \left[3 \cdot \ln\left(\frac{t}{t_{\text{nom}}}\right) + \frac{e_{\text{egnom}}}{\text{vt}(t_{\text{nom}})} - \frac{e_{g(t)}}{\text{vt}(t)}\right] \]

\[ CBD(t) = CBD \cdot \left[1 + MJ \cdot \left(400u \cdot \Delta t - \frac{PB(t)}{PB} + 1\right)\right] \]
Introducing MOSFETs Using Temperature Parameters and Equations

\[ CBS(t) = CBS \cdot \left[ 1 + MJ \cdot \left( 400u \cdot \Delta t - \frac{PB(t)}{PB} + 1 \right) \right] \]

\[ CJ(t) = CJ \cdot \left[ 1 + MJ \cdot \left( 400u \cdot \Delta t - \frac{PB(t)}{PB} + 1 \right) \right] \]

\[ CJSW(t) = CJSW \cdot \left[ 1 + MJSW \cdot \left( 400u \cdot \Delta t - \frac{PHP(t)}{PHP} + 1 \right) \right] \]

**TLEVC=1:**

\[ PB(t) = PB - PTA \cdot \Delta t \]
\[ PHP(t) = PHP - PTP \cdot \Delta t \]
\[ CBD(t) = CBD \cdot (1 + CTA \cdot \Delta t) \]
\[ CBS(t) = CBS \cdot (1 + CTA \cdot \Delta t) \]
\[ CJ = CJ \cdot (1 + CTA \cdot \Delta t) \]
\[ CJSW = CJSW \cdot (1 + CTP \cdot \Delta t) \]

**TLEVC=2:**

\[ PB(t) = PB - PTA \cdot \Delta t \]
\[ PHP(t) = PHP - PTP \cdot \Delta t \]
\[ CBD(t) = CBD \cdot \left( \frac{PB}{PB(t)} \right)^{MJ} \]
\[ CBS(t) = CBS \cdot \left( \frac{PB}{PB(t)} \right)^{MJ} \]
Using Temperature Parameters and Equations

Introducing MOSFETs

\[ CJ(t) = CJ \cdot \left( \frac{PB}{PB(t)} \right)^{MJ} \]

\[ CJSW(t) = CJSW \cdot \left( \frac{PHP}{PHP(t)} \right)^{MJSW} \]

**TLEVC=3:**

\[ PB(t) = PB + dpbd dt \cdot \Delta t \]

\[ PHP(t) = PHP + dphp dt \cdot \Delta t \]

\[ CBD(t) = CBD \cdot \left( 1 - 0.5 \cdot dpbd dt \cdot \frac{\Delta t}{PB} \right) \]

\[ CBS(t) = CBS \cdot \left( 1 - 0.5 \cdot dpbd dt \cdot \frac{\Delta t}{PB} \right) \]

\[ CJ(t) = CJ \cdot \left( 1 - 0.5 \cdot dpbd dt \cdot \frac{\Delta t}{PB} \right) \]

\[ CJSW(t) = CJSW \cdot \left( 1 - 0.5 \cdot dphp dt \cdot \frac{\Delta t}{PHP} \right) \]

**where for TLEV=0 or 1:**

\[ dpbd dt = \left[ \frac{egnom + 3 \cdot vt(tnom) + \left( 1.16 - egnom \right) \cdot \left( 2 - \frac{tnom}{tnom + 1108} \right) - PB}{tnom} \right] \]

\[ dphp dt = \left[ \frac{egnom + 3 \cdot vt(tnom) + \left( 1.16 - egnom \right) \cdot \left( 2 - \frac{tnom}{tnom + 1108} \right) - PHP}{tnom} \right] \]
Introducing MOSFETs Using Temperature Parameters and Equations

**TLEV=2:**

\[
dp_{bdt} = \frac{\left[ e_{gnom} + 3 \cdot v(t) + (E_{G} - e_{gnom}) \cdot \left( 2 - \frac{t}{t_{nom} + G_{AP}^2} \right) - P_{B} \right]}{t_{nom}}
\]

\[
dp_{hdt} = -\frac{\left[ e_{gnom} + 3 \cdot v(t) + (E_{G} - e_{gnom}) \cdot \left( 2 - \frac{t}{t_{nom} + G_{AP}^2} \right) - P_{H} \right]}{t_{nom}}
\]

**Calculating Surface Potential Temperature Equations**

**TLEVC=0:**

\[
PHI(t) = PHI \cdot \left( \frac{t}{t_{nom}} \right) - v(t) \cdot \left[ 3 \cdot \ln \left( \frac{t}{t_{nom}} \right) + \frac{e_{gnom}}{v(t)} - \frac{e_{g}(t)}{v(t)} \right]
\]

**TLEVC=1:**

\[
PHI(t) = PHI - PTC \cdot \Delta t
\]

If the PHI parameter is not specified, it is calculated as:

\[
PHI(t) = 2 \cdot v(t) \cdot \ln \left( \frac{N_{SUB}}{n_{i}} \right)
\]

The intrinsic carrier concentration, \( n_{i} \), must be temperature updated, and it is calculated from the silicon bandgap at room temperature.

\[
n_{i} = 145e^{16} \left( \frac{t}{t_{nom}} \right)^{3/2} \cdot \exp \left[ E_{G} \cdot \left( \frac{t}{t_{nom}} - 1 \right) \cdot \left( \frac{1}{2 \cdot v(t)} \right) \right]
\]

**TLEVC=2:**

\[
PHI(t) = PHI - PTC \cdot \Delta t
\]
TLEV=3:

\[ \Phi(t) = \Phi + \frac{d\Phi}{dt} \cdot \Delta t \]

where TLEV=0 or 1:

\[ \frac{d\Phi}{dt} = -\frac{e_{g}n_{om} + 3 \cdot v_{t}(t_{nom}) + (1.16 - e_{g}n_{om}) \cdot \left( 2 - \frac{t_{nom}}{t_{nom} + 1108} \right) - \Phi}{t_{nom}} \]

TLEV=2:

\[ \frac{d\Phi}{dt} = -\frac{e_{g}n_{om} + 3 \cdot v_{t}(t_{nom}) + (E_{G} - e_{g}n_{om}) \cdot \left( 2 - \frac{t_{nom}}{t_{nom} + G_{2}} \right) - \Phi}{t_{nom}} \]

**Calculating Threshold Voltage Temperature Equations**

The threshold temperature equations are:

TLEV=0:

\[ v_{bi}(t) = v_{bi}(t_{nom}) + \frac{\Phi(t) - \Phi}{2} + \frac{e_{g}n_{om} - e_{g}(t)}{2} \]

\[ V_{TO}(t) = v_{bi}(t) + \Gamma \cdot (\Phi(t))^{1/2} \]

TLEV=1:

\[ V_{TO}(t) = V_{TO} - TCV \cdot \Delta t \]

\[ v_{bi}(t) = V_{TO}(t) - \Gamma \cdot (\Phi(t))^{1/2} \]

TLEV=2:

\[ V_{TO}(t) = V_{TO} + \left( 1 + \frac{\Gamma}{2 \cdot (\Phi(t))^{1/2}} \right) \cdot \frac{d\Phi}{dt} \cdot \Delta t \]

\[ v_{bi}(t) = V_{TO}(t) - \Gamma \cdot (\Phi(t))^{1/2} \]
Calculating Mobility Temperature Equations

The MOS mobility temperature equations are:

\[ UO(t) = UO \cdot \left( \frac{t}{tnom} \right)^{BEX} \]

\[ KP(t) = KP \cdot \left( \frac{t}{tnom} \right)^{BEX} \]

\[ F1(t) = F1 \cdot \left( \frac{t}{tnom} \right)^{F1EX} \]

Calculating Channel Length Modulation Temperature Equation

The LAMBDA is modified with temperature if model parameter LAMEX is specified.

\[ LAMBDA(t) = LAMBDA \cdot (1 + LAMEX \cdot \Delta t) \]

Calculating Diode Resistance Temperature Equations

The following equation is an example of effective drain and source resistance:

\[ RD(t) = RS \cdot (1 + TRD \cdot \Delta t) \]

\[ RS(t) = RS \cdot (1 + TRS \cdot \Delta t) \]
Chapter 21

Selecting MOSFET Models: LEVEL 1-40

Now that you know more about MOSFET models from “Introducing MOSFETs” on page 20-1, it will be easier for you to choose which type of models you require for your needs.

This chapter lists various MOSFET models, and provides the specifications for each model. The following topics are covered in this chapter:

- LEVEL 1 IDS: Schichman-Hodges Model
- LEVEL 2 IDS: Grove-Frohman Model
- LEVEL 3 IDS: Empirical Model
- LEVEL 4 IDS: MOS Model
- LEVEL 5 IDS Model
- LEVEL 6 and LEVEL 7 IDS: MOSFET Model
- LEVEL 7 IDS Model
- LEVEL 8 IDS Model
- LEVEL 13 BSIM Model
- LEVEL 27 SOSFET Model
- LEVEL 28 Modified BSIM Model
- LEVEL 38 IDS: Cypress Depletion Model
- LEVEL 39 BSIM2 Model
- LEVEL 40 HP a-Si TFT Model

The remaining Hspice models are described in Chapter 22, “Selecting MOSFET Models: LEVEL 47-62.”
LEVEL 1 IDS: Schichman-Hodges Model

This section describes the parameters and equations for the LEVEL 1 IDS: Schichman-Hodges model.

LEVEL 1 Model Parameters

The LEVEL 1 model parameters follow.

Basic Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>DC model selector. LEVEL 1 is the Schichman-Hodges model.</td>
</tr>
<tr>
<td>COX</td>
<td>F/m²</td>
<td>3.453e-4</td>
<td>Oxide capacitance per unit gate area. If COX is not specified, it is calculated from TOX.</td>
</tr>
<tr>
<td>KP (BET, BETA)</td>
<td>A/V²</td>
<td></td>
<td>Intrinsic transconductance parameter. If KP is not specified and UO and TOX are entered, the parameter is computed from: KP = UO · COX</td>
</tr>
<tr>
<td>LAMBDA (LAM, LA)</td>
<td>V⁻¹</td>
<td>0.0</td>
<td>Channel-length modulation</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>1e-7</td>
<td>Gate oxide thickness</td>
</tr>
<tr>
<td>UO</td>
<td>cm²/(V·s)</td>
<td></td>
<td>Carrier mobility</td>
</tr>
</tbody>
</table>
## Effective Width and Length Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEL</td>
<td>m</td>
<td>0.0</td>
<td>Channel length reduction on each side. &lt;br&gt;( \text{DEL}_{\text{scaled}} = \text{DEL} \cdot \text{SCALM} )</td>
</tr>
<tr>
<td>LD (DLAT, LATD)</td>
<td>m</td>
<td></td>
<td>Lateral diffusion into channel from source and drain diffusion. &lt;br&gt;If LD and XJ are unspecified, LD Default=0.0. &lt;br&gt;When LD is unspecified but XJ is specified, LD is calculated as: LD Default=0.75 \cdot XJ &lt;br&gt;( \text{LD}_{\text{scaled}} = \text{LD} \cdot \text{SCALM} )</td>
</tr>
<tr>
<td>LDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as LD, but if LDAC is in the .MODEL statement, it replaces LD in the ( L_{\text{eff}} ) calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>LMLT</td>
<td>1.0</td>
<td></td>
<td>Length shrink factor</td>
</tr>
<tr>
<td>WD</td>
<td>m</td>
<td>0.0</td>
<td>Lateral diffusion into channel from bulk along width &lt;br&gt;( \text{WD}_{\text{scaled}} = \text{WD} \cdot \text{SCALM} )</td>
</tr>
<tr>
<td>WDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as WD, but if WDAC is in the .MODEL statement, it replaces WD in the ( W_{\text{eff}} ) calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>WMLT</td>
<td>1.0</td>
<td></td>
<td>Diffusion layer and width shrink factor</td>
</tr>
<tr>
<td>XJ</td>
<td>m</td>
<td>0.0</td>
<td>Metallurgical junction depth: &lt;br&gt;( \text{XJ}_{\text{scaled}} = \text{XJ} \cdot \text{SCALM} )</td>
</tr>
<tr>
<td>XL (DL, LDEL)</td>
<td>m</td>
<td>0.0</td>
<td>Accounts for masking and etching effects: &lt;br&gt;( \text{XL}_{\text{scaled}} = \text{XL} \cdot \text{SCALM} )</td>
</tr>
</tbody>
</table>
The LEVEL 1 MOSFET model should be used when accuracy is less important than simulation turn-around time. For digital switching circuits, especially when only a “qualitative” simulation of timing and function is needed, LEVEL 1 runtime can be about half that of a simulation using the LEVEL 2 model. The agreement in timing is approximately 10%. The LEVEL 1 model, however, results in severe inaccuracies in DC transfer functions of TTL-compatible input buffers, if these buffers are present in the circuit.
The channel-length modulation parameter LAMBDA is equivalent to the inverse of the Early voltage for the bipolar transistor. LAMBDA is a measure of the output conductance in saturation. When this parameter is specified, the MOSFET has a finite but constant output conductance in saturation. If LAMBDA is not input, the LEVEL 1 model assumes zero output conductance.

LEVEL 1 Model Equations

The LEVEL 1 model equations follow.

IDS Equations

In the LEVEL 1 model the carrier mobility degradation and the carrier saturation effect and weak inversion model are not included. This model determines the DC current as follows:

**Cutoff Region, \( v_{gs} \leq v_{th} \)**

\[
I_{ds} = 0.0
\]

**Linear Region, \( v_{ds} < v_{gs} - v_{th} \)**

\[
I_{ds} = KP \cdot \frac{W_{eff}}{L_{eff}} \cdot (1 + LAMBDA \cdot v_{ds}) \cdot \left( v_{gs} - v_{th} - \frac{v_{ds}}{2} \right) \cdot v_{ds}
\]

**Saturation Region, \( v_{ds} \geq v_{gs} - v_{th} \)**

\[
I_{ds} = \frac{KP}{2} \cdot \frac{W_{eff}}{L_{eff}} \cdot (1 + LAMBDA \cdot v_{ds}) \cdot (v_{gs} - v_{th})^2
\]

Effective Channel Length and Width

The model calculates the effective channel length and width from the drawn length and width as follows:
Threshold Voltage, $v_{th}$

$v_{sb} \geq 0$

$$v_{th} = v_{bi} + \text{GAMMA} \cdot (PHI + v_{sb})^{1/2}$$

$v_{sb} < 0$

$$v_{th} = v_{bi} + \text{GAMMA} \cdot \left( PHI^{1/2} + 0.5 \frac{v_{sb}}{PHI^{1/2}} \right)$$

Where the built-in voltage $v_{bi}$ is defined as:

$$v_{bi} = v_{fb} + PHI$$

or

$$v_{bi} = VTO - \text{GAMMA} \cdot PHI^{1/2}$$

**Note:** See “Common Threshold Voltage Parameters” on page 20-50 for calculation of VTO, GAMMA, and PHI if they are not specified.

Saturation Voltage, $v_{sat}$

The saturation voltage for the LEVEL 1 model is due to channel pinch off at the drain side and is computed by:

$$v_{sat} = v_{gs} - v_{th}$$

In the LEVEL 1 model, the carrier velocity saturation effect is not included.
LEVEL 2 IDS: Grove-Frohman Model

This section describes the parameters and equations for the LEVEL 2 IDS: Grove-Frohman model.

**LEVEL 2 Model Parameters**

The LEVEL 2 model parameters follow.

### Basic Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>DC model selector. LEVEL 2 is the Grove-Frohman model.</td>
</tr>
<tr>
<td>COX</td>
<td>F/m²</td>
<td>3.453e-4</td>
<td>Oxide capacitance per unit gate area. This parameter is calculated from TOX if not specified.</td>
</tr>
<tr>
<td>ECRIT (ESAT)</td>
<td>V/cm</td>
<td>0.0</td>
<td>Critical electric field for carrier velocity saturation. From Grove: electrons 6e4 holes 2.4e4 Use zero to indicate an infinite value. ECRIT is preferred over VMAX because the equation is more stable. ECRIT is estimated as: ECRIT = 100 \cdot (VMAX / UO)</td>
</tr>
<tr>
<td>KP (BET, BETA)</td>
<td>A/V²</td>
<td>2.0e-5</td>
<td>Intrinsic transconductance. If KP is not specified and UO and TOX are entered, KP is calculated from KP = UO \cdot COX</td>
</tr>
<tr>
<td>LAMBDA (LAM, LA)</td>
<td>V⁻¹</td>
<td>0.0</td>
<td>Channel length modulation</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>NEFF</td>
<td></td>
<td>1.0</td>
<td>Total channel charge (fixed and mobile) coefficient</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>1e-7</td>
<td>Gate oxide thickness</td>
</tr>
<tr>
<td>VMAX (VMX, VSAT)</td>
<td>m/s</td>
<td>0.0</td>
<td>Maximum drift velocity of carriers. Use zero to indicate an infinite value.</td>
</tr>
</tbody>
</table>

**Effective Width and Length Parameters**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| DEL         | m     | 0.0     | Channel-length reduction on each side: 
DEL_{scaled} = DEL \cdot SCALM |
| LD (DLAT, LADT) | m  |         | Lateral diffusion into channel from source and drain diffusion. 
If LD and XJ are unspecified, LD default=0.0. 
When LD is unspecified but XJ is specified, LD is calculated from: XJ. The default=0.75 \cdot XJ. 
LD_{scaled} = LD \cdot SCALM |
| LDAC        | m     |         | This parameter is the same as LD, but if LDAC is included in the .MODEL statement, it replaces LD in the Leff calculation for AC gate capacitance. |
| LMLT        |       | 1.0     | Length shrink factor |
| LREF        | m     | 0.0     | Channel length reference 
LREF_{scaled} = LREF \cdot SCALM |
| WD          | m     | 0.0     | Lateral diffusion into channel from bulk along width WD_{scaled} = WD \cdot SCALM |
### Threshold Voltage Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTA</td>
<td></td>
<td>0.0</td>
<td>Narrow width factor for adjusting threshold</td>
</tr>
<tr>
<td>GAMMA</td>
<td>V(^{1/2})</td>
<td>0.5276</td>
<td>Body effect factor. This parameter is calculated from NSUB if not specified (see “Common Threshold Voltage Parameters” on page 20-50).</td>
</tr>
<tr>
<td>LND</td>
<td>(\mu)m/V</td>
<td>0.0</td>
<td>ND length sensitivity</td>
</tr>
<tr>
<td>LN0</td>
<td>(\mu)m</td>
<td>0.0</td>
<td>N0 length sensitivity</td>
</tr>
<tr>
<td>ND</td>
<td>V(^{-1})</td>
<td>0.0</td>
<td>Drain subthreshold factor</td>
</tr>
<tr>
<td>N0</td>
<td></td>
<td>0.0</td>
<td>Gate subthreshold factor. Typical value=1.</td>
</tr>
</tbody>
</table>
### Mobility Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOB</td>
<td></td>
<td>0.0</td>
<td>Mobility equation selector. This parameter can be set to MOB=0 or MOB=7. If MOB=7, the model is changed, which also affects the channel length calculation.</td>
</tr>
</tbody>
</table>

**Note:** MOB=7 operates as a flag. It invokes the channel length modulation and mobility equations of MOSFET LEVEL 3.
The mobility parameters are best determined by curve fitting. In most cases UTRA should be specified between 0.0 and 0.5. Nonzero values for UTRA can result in negative resistance regions at the onset of saturation.

**LEVEL 2 Model Equations**

The LEVEL 2 model equations follow.

**IDS Equations**

The following section describes the way the LEVEL 2 MOSFET model calculates the drain current of n-channel and p-channel MOSFETs.

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>THETA</td>
<td>V⁻¹</td>
<td>0.0</td>
<td>Mobility modulation. THETA is used only when MOB=7. A typical value in this application is THETA=5e-2.</td>
</tr>
<tr>
<td>UCRIT</td>
<td>V/cm</td>
<td>1.0e4</td>
<td>Critical field for mobility degradation, UCRIT. The parameter is the limit at which the surface mobility UO begins to decrease in accordance with the empirical relation given later.</td>
</tr>
<tr>
<td>UEXP (F2)</td>
<td></td>
<td>0.0</td>
<td>Critical field exponent in the empirical formula which characterizes surface mobility degradation</td>
</tr>
<tr>
<td>UO (UB, UBO)</td>
<td>cm²/ (V·s)</td>
<td>600 (N) 250 (P)</td>
<td>Low-field bulk mobility. This parameter is calculated from KP if KP is input.</td>
</tr>
<tr>
<td>UTRA</td>
<td></td>
<td>0.0</td>
<td>Transverse field coefficient</td>
</tr>
</tbody>
</table>

Note: SPICE does not use UTRA. Star-Hspice uses it if supplied, but issues a warning because UTRA can hinder convergence.
**Cutoff Region, \( v_{gs} \leq v_{th} \)**

\[ I_{ds} = 0 \]  
(see subthreshold current)

**On Region, \( v_{gs} > v_{th} \)**

\[ I_{ds} = \beta \cdot \left( v_{gs} - v_{bi} - \frac{\eta \cdot v_{de}}{2} \right) \cdot v_{de} - \frac{2}{3} \gamma \cdot \left[ (PHI + v_{de} + v_{sb})^{3/2} - (PHI + v_{sb})^{3/2} \right] \]

where:

\[ v_{de} = \min(v_{ds}, v_{dsat}) \]

\[ \eta = 1 + \text{DELTA} \cdot \frac{\pi \cdot \varepsilon_s i}{4 \cdot \text{COX} \cdot W_{\text{eff}}} \]

\[ \beta = KP \cdot \frac{W_{\text{eff}}}{L_{\text{eff}}} \]

**Effective Channel Length and Width**

The model calculates effective channel length and width from the drawn length and width as follows:

\[ L_{\text{eff}} = L_{\text{scaled}} \cdot L_{MLT} + XL_{\text{scaled}} - 2 \cdot (LD_{\text{scaled}} + DL_{\text{scaled}}) \]

\[ W_{\text{eff}} = M \cdot (W_{\text{scaled}} \cdot W_{MLT} + XW_{\text{scaled}} - 2 \cdot WD_{\text{scaled}}) \]

\[ L_{\text{REF eff}} = L_{REF scaled} \cdot L_{MLT} + XL_{\text{scaled}} - 2 \cdot (LD_{\text{scaled}} + DL_{\text{scaled}}) \]

\[ W_{\text{REF eff}} = M \cdot (W_{\text{REF scaled}} \cdot W_{MLT} + XW_{\text{scaled}} - 2 \cdot WD_{\text{scaled}}) \]
Threshold Voltage, $v_{th}$

The model parameter VTO is an extrapolated zero-bias threshold voltage of a large device. The effective threshold voltage, including the device size effects and the terminal voltages, is calculated by:

$$v_{th} = v_{bi} + \gamma \cdot (PHI + v_{sb})^{1/2}$$

where:

$$v_{bi} = VTO - GAMMA \cdot (PHI)^{1/2} + (\eta - 1) \cdot (PHI + v_{sb})$$

The narrow width effect is included through $v_{bi}$ and $\eta$. To include the narrow width effect, specify the model parameter DELTA. The short-channel effect is included through the effective $\gamma$. To include short-channel effects, the model parameter XJ must be greater than zero. Then:

$$\gamma = GAMMA \cdot \left\{ 1 - \frac{XJ_{scaled}}{2 \cdot L_{eff}} \cdot \left[ \left( 1 + \frac{2 \cdot W_s}{XJ_{scaled}} \right)^{1/2} + \left( 1 + \frac{2 \cdot W_d}{XJ_{scaled}} \right)^{1/2} - 2 \right] \right\}$$

The depletion widths, $W_s$ and $W_d$, are determined by:

$$W_s = \left[ \frac{2 \cdot E_{si}}{q \cdot NSUB} \cdot (PHI + v_{sb}) \right]^{1/2}$$

$$W_d = \left[ \frac{2 \cdot E_{si}}{q \cdot NSUB} \cdot (PHI + v_{ds} + v_{sb}) \right]^{1/2}$$

Star-Hspice calculates parameters such as VTO, GAMMA, and PHI unless you specify them. The model uses these parameters to calculate threshold voltage. (See “Common Threshold Voltage Parameters” on page 20-50).
Saturation Voltage, $v_{dsat}$

If you do not specify the model parameter VMAX, the program computes the saturation voltage due to channel pinch off at the drain side. By including the corrections for small-size effects, $v_{sat}$ is:

$$v_{sat} = \frac{v_{gs} - v_{bi}}{\eta} + \frac{1}{2} \left( \frac{\gamma}{\eta} \right)^2 \cdot \left\{ 1 - \left[ 1 + 4 \cdot \left( \frac{\eta}{\gamma} \right)^2 \cdot \left( \frac{v_{gs} - v_{bi}}{\eta} + PHI + v_{sb} \right) \right]^{1/2} \right\}$$

$$v_{dsat} = v_{sat}$$

If you specify ECRIT, the program modifies $v_{sat}$ to include carrier velocity saturation effect.

$$v_{dsat} = v_{sat} + v_c - (v_{sat}^2 + v_c^2)^{1/2}$$

where:

$$v_c = ECRIT \cdot L_{eff}$$

Note: If VMAX is specified, a different $v_{dsat}$ calculation is performed. Refer to the Vladimirescu document for details.

Mobility Reduction, $u_{eff}$

The mobility of carriers in the channel decreases as the carriers’ speeds approach their scattering limited velocity. In Star-Hspice the mobility degradation for the LEVEL 2 MOS model uses two different equations, depending on the mobility equation selector value of MOB.

If MOB=0, (default):

$$u_{eff} = UO \cdot \frac{UCRIT \cdot E_{si}}{COX \cdot (v_{gs} - v_{th} - UTRA \cdot v_{ds})}^{UEXP}$$
Since \( u_{\text{eff}} \) is less than \( U_O \), the program uses the above equation if the bracket term is less than one; otherwise the program uses \( u_{\text{eff}} = U_O \).

If \( \text{MOB}=7 \), \( \text{THETA} \neq 0 \)

\[
    u_{\text{eff}} = \frac{U_O}{1 + \text{THETA} \cdot (v_{gs} - v_{th})}
\]

If \( v_{gs} < v_{th} \), \( u_{\text{eff}} = U_O \)

If \( \text{MOB}=7 \), \( \text{THETA}=0 \)

\[
    u_{\text{eff}} = U_O \cdot \left[ \frac{\text{UCRIT} \cdot E_{si}}{\text{COX} \cdot (v_{gs} - v_{th})} \right]^{\text{UEXP}}
\]

If \( \text{MOB}=7 \), \( \text{VMAX}>0 \)

\[
    u_{\text{eff}} = \frac{u_{\text{eff}}}{1 + u_{\text{eff}} \cdot \frac{v_{de}}{\text{VMAX} \cdot L_{eff}}}
\]

**Channel Length Modulation**

The LEVEL 2 MOS model includes the channel length modulation effect by modifying the \( I_{ds} \) current as follows:

\[
    I_{ds} = \frac{I_{ds}}{1 - \lambda \cdot v_{ds}}
\]

The model calculates the value of \( \lambda \) if you do not specify the model parameter \( \text{LAMBDA} \).

**LAMBDA > 0**

\[
    \lambda = \text{LAMBDA}
\]
LEVEL 2 IDS: Grove-Frohman Model

Selecting MOSFET Models: LEVEL 1-40

VMAX>0, NSUB >0, and LAMBDA ≤0

\[ \lambda = \frac{X_d}{NEF^{1/2} \cdot L_{eff} \cdot v_{ds}} \left\{ \left[ \frac{VMAX \cdot X_d}{2 \cdot NEF^{1/2} \cdot u_{eff}} \right]^2 + v_{ds} - v_{dsat} \right\}^{1/2} - \frac{VMAX \cdot X_d}{2 \cdot NEF^{1/2} \cdot u_{eff}} \]

VMAX=0, NSUB>0, and LAMBDA ≤0

If MOB=0

\[ \lambda = \frac{X_d}{L_{eff} \cdot v_{ds}} \left\{ \frac{v_{ds} - v_{dsat}}{4} + \left[ 1 + \left( \frac{v_{ds} - v_{dsat}}{4} \right)^2 \right]^{1/2} \right\} \]

If MOB=7

\[ \lambda = \frac{X_d}{L_{eff} \cdot v_{ds}} \left\{ \frac{v_{ds} - v_{dsat}}{4} + \left[ 1 + \left( \frac{v_{ds} - v_{dsat}}{4} \right)^2 \right]^{1/2} - 1 \right\} \]

where \( X_d \) is defined by:

\[ X_d = \left( \frac{2 \cdot E_{si}}{q \cdot NSUB} \right)^{1/2} \]

The above equations do not include the effect of the field between gate and drain and gate and pinch-off point, respectively. They tend to overestimate the output conductance in the saturation region.

The modification of \( I_{ds} \) by factor \( (1 - \lambda \cdot v_{ds}) \) is equivalent to replacing \( L_{eff} \) with:

\[ Le = L_{eff} - \lambda \cdot v_{ds} \cdot L_{eff} \]

To prevent the channel length (Le) from becoming negative, Star-Hspice limits the value of Le.
If Le < xwb, then Le is replaced by:

\[
\frac{xwb}{1 + \frac{xwb - Le}{xwb}}
\]

where:

\[xwb = X_d \cdot PB^{1/2}\]

**Subthreshold Current, \( I_{ds} \)**

This region of operation is characterized by the fast surface states model parameter, NFS. For NFS > 0 the model determines the modified threshold voltage (von) as follows:

\[von = v_{th} + fast\]

where:

\[fast = vt \cdot \left[ \eta + (PHI + v_{sb})^{1/2} \cdot \frac{\partial \gamma}{\partial v_{sb}} + \frac{\gamma}{2 \cdot (PHI + v_{sb})^{1/2}} + q \cdot NFS \cdot \frac{COX}{\phi v_{sb}} \right] \]

and vt is the thermal voltage.

The \( I_{ds} \) current for \( v_{gs} < von \) is given by:

\[I_{ds} = I_{ds}(von, vde, v_{sb}) \cdot e^{\frac{v_{gs} - von}{fast}}\]

\( v_{gs} \geq von \)

\[I_{ds} = I_{ds}(v_{gs}, vde, v_{sb})\]

where:

\[vde = \min(v_{ds}, v_{dsat})\]
Note: The modified threshold voltage (von), due to NFS specification, is also used in strong inversion instead of $v_{th}$, mainly in the mobility equations.

If WIC=3, the model calculates the subthreshold current differently. In this case the $I_{ds}$ current is:

$$I_{ds} = I_{ds}(v_{gs}, v_{de}, v_{sb}) + i_{sub}(N0_{eff}, ND_{eff}, v_{gs}, v_{ds})$$

The $N0_{eff}$ and $ND_{eff}$ are functions of effective device width and length.
LEVEL 3 IDS: Empirical Model

This sections provides the LEVEL 3 IDS: Empirical model parameters and equations.

LEVEL 3 Model Parameters

The LEVEL 3 model parameters follow.

Basic Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>DC model selector. LEVEL=3 is an empirical model.</td>
</tr>
<tr>
<td>COX</td>
<td>F/m²</td>
<td>3.453e-4</td>
<td>Oxide capacitance per unit gate area. If this parameter is not specified, it is calculated from TOX.</td>
</tr>
<tr>
<td>DERIV</td>
<td></td>
<td>1</td>
<td>Derivative method selector</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DERIV=0: analytic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DERIV=1: finite difference</td>
</tr>
<tr>
<td>KAPPA</td>
<td>V⁻¹</td>
<td>0.2</td>
<td>Saturation field factor. This parameter is used in the channel length modulation equation.</td>
</tr>
<tr>
<td>KP (BET, BETA)</td>
<td>A/V²</td>
<td>2.0e-5</td>
<td>Intrinsc transconductance parameter. If this parameter is not specified and UO and TOX are entered, KP is calculated from</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KP = UO · COX</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>1e-7</td>
<td>Gate oxide thickness</td>
</tr>
<tr>
<td>VMAX (VMX)</td>
<td>m/s</td>
<td>0.0</td>
<td>Maximum drift velocity of carriers. Use zero to indicate an infinite value.</td>
</tr>
</tbody>
</table>
Effective Width and Length Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| DEL          | m     | 0.0     | Channel length reduction on each side  
|              |       |         | $\text{DEL}_{\text{scaled}} = \text{DEL} \cdot \text{SCALM}$ |
| LD (DLAT, LATD) | m |         | Lateral diffusion into channel from source and drain diffusion,  
|              |       |         | If LD and XJ are unspecified, LD Default= 0.0.  
|              |       |         | If LD is unspecified but XJ is specified, LD is calculated from XJ as LD = 0.75 \cdot XJ. |
| LDAC         | m     |         | This parameter is the same as LD, but if LDAC is included in the .MODEL statement, it replaces LD in the $L_{\text{eff}}$ calculation for AC gate capacitance. |
| LREF         | m     | 0.0     | Channel length reference  
|              |       |         | $\text{LREF}_{\text{scaled}} = \text{LREF} \cdot \text{SCALM}$ |
| LMLT         |       | 1.0     | Length shrink factor |
| WD           | m     | 0.0     | Lateral diffusion into channel width from bulk  
|              |       |         | $\text{WD}_{\text{scaled}} = \text{WD} \cdot \text{SCALM}$ |
| WDAC         | m     |         | This parameter is the same as WD, but if WDAC is included in the .MODEL statement, it replaces WD in the $W_{\text{eff}}$ calculation for AC gate capacitance. |
| WMLT         |       | 1.0     | Diffusion layer and width shrink factor |
| WREF         | m     | 0.0     | Channel width reference  
|              |       |         | $\text{WREF}_{\text{scaled}} = \text{WREF} \cdot \text{SCALM}$ |
### Threshold Voltage Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTA</td>
<td></td>
<td>0.0</td>
<td>Narrow width factor for adjusting threshold</td>
</tr>
<tr>
<td>ETA</td>
<td></td>
<td>0.0</td>
<td>Static feedback factor for adjusting threshold</td>
</tr>
<tr>
<td>GAMMA</td>
<td>V(^{1/2})</td>
<td>0.5276</td>
<td>Body effect factor. This parameter is calculated from NSUB if not specified (See “Common Threshold Voltage Parameters” on page 20-50).</td>
</tr>
<tr>
<td>LND</td>
<td>(\mu m/V)</td>
<td>0.0</td>
<td>ND length sensitivity</td>
</tr>
<tr>
<td>LN0</td>
<td>(\mu m)</td>
<td>0.0</td>
<td>N0 length sensitivity</td>
</tr>
<tr>
<td>ND</td>
<td>(V^{-1})</td>
<td>0.0</td>
<td>Drain subthreshold factor</td>
</tr>
<tr>
<td>N0</td>
<td></td>
<td>0.0</td>
<td>Gate subthreshold factor (typical value=1)</td>
</tr>
<tr>
<td>NFS (DFS,NF, DNF)</td>
<td>(cm^{-2} V^{-1})</td>
<td>0.0</td>
<td>Fast surface state density</td>
</tr>
</tbody>
</table>
Mobility Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>THETA</td>
<td>V^{-1}</td>
<td>0.0</td>
<td>Mobility degradation factor</td>
</tr>
<tr>
<td>UO (UB,UBO)</td>
<td>cm^{2}/(V·s)</td>
<td>600(N) 250(P)</td>
<td>Low field bulk mobility. This parameter is calculated from KP if KP is specified.</td>
</tr>
</tbody>
</table>

LEVEL 3 Model Equations

The LEVEL 3 model equations follow.

IDS Equations

The following describes the way the LEVEL 3 MOSFET model calculates the drain current, $I_{ds}$. 

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSUB (DNB, NB)</td>
<td>cm^{-3}</td>
<td>1e15</td>
<td>Bulk surface doping. This parameter is calculated from GAMMA if not specified.</td>
</tr>
<tr>
<td>PHI</td>
<td>V</td>
<td>0.576</td>
<td>Surface inversion potential. This parameter is calculated from NSUB if not specified (see “Common Threshold Voltage Parameters” on page 20-50).</td>
</tr>
<tr>
<td>VTO (VT)</td>
<td>V</td>
<td></td>
<td>Zero-bias threshold voltage. This parameter is calculated if not specified (see “Common Threshold Voltage Parameters” on page 20-50).</td>
</tr>
<tr>
<td>WIC</td>
<td></td>
<td>0.0</td>
<td>Sub-threshold model selector</td>
</tr>
<tr>
<td>WND</td>
<td>μm/V</td>
<td>0.0</td>
<td>ND width sensitivity</td>
</tr>
<tr>
<td>WN0</td>
<td>μm</td>
<td>0.0</td>
<td>N0 width sensitivity</td>
</tr>
</tbody>
</table>
Cutoff Region, $v_{gs} \leq v_{th}$

\[ I_{ds} = 0 \quad (\text{See subthreshold current}) \]

On Region, $v_{gs} > v_{th}$

\[ I_{ds} = \beta \left( v_{gs} - v_{th} - \frac{1 + f_b}{2} \cdot v_{de} \right) \cdot v_{de} \]

where:

\[ \beta = K_P \cdot \frac{W_{eff}}{L_{eff}} \]

\[ = u_{eff} \cdot COX \cdot \frac{W_{eff}}{L_{eff}} \]

\[ v_{de} = \min(v_{ds}, v_{dsat}) \]

and:

\[ f_b = f_n + \frac{\text{GAMMA} \cdot f_s}{4 \cdot (\text{PHI} + v_{sb})^{1/2}} \]

**Note:** In the above equation the factor 4 should be 2, but since SPICE uses a factor of 4, Star-Hspice uses factor of 4 as well.

The narrow width effect is included through the $f_n$ parameter:

\[ f_n = \frac{\text{DELTA}}{W_{eff}} \cdot 1 \cdot \frac{2\pi \cdot E_{si}}{4 \cdot COX} \]
The term $f_s$ expresses the effect of the short channel and is determined as:

$$
 f_s = 1 - \frac{XJ_{scaled}}{L_{eff}} \left\{ \frac{L_{D_{scaled}} + W_{c}}{XJ_{scaled}} \cdot \left[ 1 - \left( \frac{W_{p}}{XJ_{scaled} + W_{p}} \right)^2 \right]^{1/2} - \frac{L_{D_{scaled}}}{XJ_{scaled}} \right\}
$$

$$
 W_{p} = X_d \cdot (PHI + v_{sb})^{1/2}
$$

$$
 X_d = \left( \frac{2 \cdot E_{si}}{q \cdot NSUB} \right)^{1/2}
$$

$$
 W_{c} = XJ_{scaled} \cdot \left[ 0.0631353 + 0.8013292 \cdot \left( \frac{W_{p}}{XJ_{scaled}} \right)^2 - 0.01110777 \cdot \left( \frac{W_{p}}{XJ_{scaled}} \right)^2 \right]
$$

**Effective Channel Length and Width**

The model determines effective channel length and width in the LEVEL 3 model as follows:

$$
 L_{eff} = L_{scaled} \cdot LMLT + XL_{scaled} - 2 \cdot (L_{D_{scaled}} + DEL_{scaled})
$$

$$
 W_{eff} = M \cdot (W_{scaled} \cdot WMLT + XW_{scaled} - 2 \cdot WD_{scaled})
$$

$$
 L_{REFF} = L_{REF_{scaled}} \cdot LMLT + XL_{scaled} - 2 \cdot (L_{D_{scaled}} + DEL_{scaled})
$$

$$
 W_{REF_{eff}} = M \cdot (W_{REF_{scaled}} \cdot WMLT + XW_{scaled} - 2 \cdot WD_{scaled})
$$
Threshold Voltage, $v_{th}$

The effective threshold voltage, including the device size and terminal voltage effects, is calculated by:

$$v_{th} = v_{bi} - \frac{8.14 \cdot 22 \cdot ETA}{COX \cdot L_{eff}} \cdot v_{ds} + GAMMA \cdot f_s \cdot (PHI + v_{sb})^{1/2} + f_n \cdot (PHI + v_{th})$$

where:

$$v_{bi} = v_{fb} + PHI$$

or

$$v_{bi} = VTO - GAMMA \cdot PHI^{1/2}$$

The VTO is the extrapolated zero-bias threshold voltage of a large device. If VTO, GAMMA, and PHI are not specified, Star-Hspice computes them (see “Common Threshold Voltage Parameters” on page 20-50).

Saturation Voltage, $v_{dsat}$

For the LEVEL 3 model, Star-Hspice determines saturation voltage due to channel pinch-off at the drain side. The model uses the parameter VMAX to include the reduction of the saturation voltage due to carrier velocity saturation effect.

$$v_{sat} = \frac{v_{gs} - v_{th}}{1 + f_b}$$

$$v_{dsat} = v_{sat} + v_c - (v_{sat}^2 + v_c^2)^{1/2}$$

where:

$$v_c = \frac{VMAX \cdot L_{eff}}{us}$$
The surface mobility parameter “us” is defined in the next section. If the model parameter VMAX is not specified, then:

\[ v_{dsat} = v_{sat} \]

**Effective Mobility, \( u_{eff} \)**

The model defines the carrier mobility reduction due to the normal field as the effective surface mobility (us).

For \( v_{gs} > v_{th} \)

\[
us = \frac{UO}{1 + \text{THETA} \cdot (v_{gs} - v_{th})}
\]

The model determines the degradation of mobility due to the lateral field and the carrier velocity saturation if you specify the VMAX model parameter.

**VMAX>0**

\[
u_{eff} = \frac{us}{1 + \frac{vde}{v_c}}
\]

otherwise,

\[ u_{eff} = us \]

**Channel Length Modulation**

For \( v_{ds} > v_{dsat} \), the channel length modulation factor is computed. The model determines the channel length reduction (\( \Delta L \)) differently, depending on the VMAX model parameter value.

**VMAX = 0**

\[
\Delta L = X_d \cdot [\text{KAPPA} \cdot (v_{ds} - v_{dsat})]^{1/2}
\]
VMAX>0

\[ \Delta L = -\frac{E_p \cdot X_d^2}{2} + \left( \frac{E_p \cdot X_d^2}{2} \right)^2 + KAPPA \cdot X_d^2 \cdot (v_{ds} - v_{dsat}) \right]^{1/2} \]

where \( E_p \) is the lateral electric field at the pinch off point. Its value is approximated by:

\[ E_p = \frac{v_c \cdot (v_c + v_{dsat})}{L_{eff} \cdot v_{dsat}} \]

The current \( I_{ds} \) in the saturation region is computed as:

\[ I_{ds} = \frac{I_{ds}}{1 - \frac{\Delta L}{L_{eff}}} \]

In order to prevent the denominator from going to zero, Star-Hspice limits the \( \Delta L \) value as follows:

If

\[ \Delta L > \frac{L_{eff}}{2} \]

then

\[ \Delta L = L_{eff} - \left( \frac{L_{eff}}{2} \right)^2 \]


**Subthreshold Current, I<sub>ds</sub>**

This region of operation is characterized by the model parameter for fast surface state (NFS). The modified threshold voltage (von) is determined as follows:

\[ \text{NFS} > 0 \]

\[ \text{von} = v_{th} + fast \]

where:

\[
fast = \frac{v_{tm} \cdot \left[ 1 + \frac{q \cdot NFS}{COX} + \frac{\text{GAMMA} \cdot f_s \cdot (PHI + v_{sb})^{1/2} + f_n \cdot (PHI + v_{sb})}{2 \cdot (PHI + v_{sb})} \right]}{\text{GAMMA}}
\]

The current I<sub>ds</sub> is given by:

\[ v_{gs} < \text{von} \]

\[ I_{ds} = I_{ds}(von, vde, v_{sb}) \cdot e^{\frac{v_{gs} - \text{von}}{fast}} \]

\[ v_{gs} = \text{von} \]

\[ I_{ds} = I_{ds}(v_{gs}, vde, v_{sb}) \]

---

**Note:** The model does not use the modified threshold voltage in strong inversion.

---

If WIC=3, the model calculates subthreshold current differently. In this case, the I<sub>ds</sub> current is:

\[ I_{ds} = I_{ds}(v_{gs}, vde, v_{sb}) + i_{sub}(N_{0_{eff}}, N_{D_{eff}}, v_{gs}, v_{ds}) \]

Subthreshold current isub for LEVEL=3 is the same as for LEVEL=13 (see page 21-115).

N<sub>0_{eff}</sub> and N<sub>D_{eff}</sub> are functions of effective device width and length.
Compatibility Notes

This section describes compatibility issues.

Star-Hspice versus SPICE3

Differences between Star-Hspice and Berkeley SPICE3 can arise in the following situations:

Small XJ

Star-Hspice and SPICE3 differ for small values of XJ, typically less than 0.05 microns. Such small values for XJ are physically unreasonable and should be avoided. XJ is used to calculate the short-channel reduction of the GAMMA effect,

\[ \text{GAMMA} \rightarrow f_s \cdot \text{GAMMA} \]

\( f_s \) is normally less than or equal to 1. For very small values of XJ, \( f_s \) can be greater than one. Star-Hspice imposes the limit \( f_s \leq 1.0 \), while SPICE3 allows \( f_s > 1.0 \).

ETA

Star-Hspice uses 8.14 as the constant in the ETA equation, which provides the variation in threshold with \( v_{ds} \). Berkeley SPICE3 uses 8.15.

Solution: To convert a SPICE3 model to Star-Hspice, multiply ETA by 815/814.

NSUB Missing

When NSUB is missing in SPICE3, the KAPPA equation becomes inactive. In Star-Hspice, a default NSUB is generated from GAMMA, and the KAPPA equation is active.

Solution: If NSUB is missing in the SPICE3 model, set KAPPA=0 in the Star-Hspice model.
LD Missing

If LD is missing, Star-Hspice uses the default $0.75 \cdot X_J$. SPICE3 defaults LD to zero.

Solution: If LD is missing in the SPICE3 model, set LD=0 in the HSPICE model.

Constants

- Boltzmann constant $k = 1.3806226 \times 10^{-23} \text{J} \cdot \text{K}^{-1}$
- Electron charge $e = 1.6021918 \times 10^{-19} \text{C}$
- Permittivity of silicon dioxide $\varepsilon_{\text{ox}} = 3.45314379969 \times 10^{-11} \text{F/m}$
- Permittivity of silicon $\varepsilon_{\text{si}} = 1.035943139907 \times 10^{-10} \text{F/m}$

Temperature Compensation

The example below verifies temperature dependence for LEVEL 3.

Input file

```plaintext
$ test of temp dependence for LEVEL=3 Tlevc=0 Tlev=1
.option ingold=2 numdgt=6
.temp 25 100
vd d 0 5
vg g 0 2
m1 d g 0 0 nch w=10u L=1u
.op
.print id=lx4(m1) vdsat=lv10(m1)
.model nch nmos LEVEL=3 tlev=1 tlevc=0 acm=3
+ uo=600 tox=172.6572
+ vto=0.8 gamma=0.8 phi=0.64
+ kappa=0 xj=0
+ nsub=1e16 rsh=0
+ tcv=1.5e-3 bex=-1.5
.end
```
This simple model, with XJ=0 and KAPPA=0, has a saturation current:

\[
I_{ds} = \frac{beta \cdot 0.5 \cdot (v_{gs} - v_{im})^2}{1 + fb}
\]

\[
beta = COX \cdot \left(\frac{W}{L}\right) \cdot UO(t) \quad fb = \frac{GAMMA}{(4 \cdot \sqrt{\phi(t)})}
\]

Using the model parameters in the input file and the equations from the previous page:

\[
beta = (1.2e - 3) \cdot \left(\frac{t}{t_{ref}}\right)^{BEX}
\]

\[
v_{im} = 0.8 - TCV \cdot (t - t_{ref})
\]

\[
\phi(t) = 0.64 \cdot \left(\frac{t}{t_{ref}}\right) - v_{therm} \cdot \left(egarg + 3 \cdot \log\left(\frac{t}{t_{ref}}\right)\right)
\]

At room temperature:

\[
beta = (1.2e - 3)
\]

\[
v_{im} = 0.8
\]

\[
\phi(t) = 0.64
\]

\[
I_{ds} = (1.2e - 3) \cdot 0.5 \cdot \left(\frac{2 - 0.8}{0.2}\right)^2 \cdot \frac{1 + \frac{0.2}{\sqrt{0.64}}}{6.912e - 4} = 6.912e - 4
\]
At $T=100$:

\[
\beta = 1.2e - 3 \cdot (1.251551)^{-1.5} = 0.570545e - 4
\]

\[
v_{tm} = 0.8 - (1.5e - 3) \cdot 75 = 0.6875
\]

\[
eg_{garg} = 9.399920 \quad v_{therm} = 3.215466e - 2
\]

\[
\phi(t) = 0.64 \cdot 1.251551 - 0.3238962 = 0.4770964
\]

\[
I_{ds} = \beta \cdot 0.5 \cdot \frac{(2 - vt)^2}{1 + \frac{0.2}{\sqrt{\phi(t)}}} = 5.724507e - 4
\]

Star-Hspice results:

- $T=25$, $id=6.91200e-04$
- $T=100$, $id=5.72451e-04$

These results are in agreement with the hand calculations.
LEVEL 4 IDS: MOS Model

The LEVEL 4 MOS model is the same as the LEVEL 2 model, with the following exceptions:

- No narrow width effects: $\eta = 1$
- No short-channel effects: $\gamma = \text{GAMMA}$
- For lateral diffusion, $LD_{\text{scaled}} = LD \cdot XJ \cdot \text{SCALM}$. The LD default = 0.75 if XJ is specified and 0 if XJ is not specified.
- TPG, the model parameter for type of gate materials, defaults to zero (AL gate). The default is 1 for other levels. This parameter computes VTO if that model parameter is not specified (see “Common Threshold Voltage Parameters” on page 20-50).
LEVEL 5 IDS Model

This section describes the LEVEL 5 IDS model parameters and equations.

**Note:** This model uses micrometer units rather than the typical meter units. Units and defaults are often unique in LEVEL 5. The option SCALM is ineffective for this level.

LEVEL 5 Model Parameters

The LEVEL 5 model parameters follow.

**Basic Model Parameters**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>Model level selector</td>
</tr>
<tr>
<td>DNB (NSUB)</td>
<td>cm⁻³</td>
<td>0.0</td>
<td>Surface doping</td>
</tr>
<tr>
<td>DP</td>
<td>µm</td>
<td>1.0</td>
<td>Implant depth (depletion model only)</td>
</tr>
<tr>
<td>ECV</td>
<td>V/µm</td>
<td>1000</td>
<td>Critical field</td>
</tr>
<tr>
<td>NI</td>
<td>cm⁻²</td>
<td>2e11</td>
<td>Implant doping (depletion model only)</td>
</tr>
<tr>
<td>PHI</td>
<td>V</td>
<td>0.8</td>
<td>Built-in potential</td>
</tr>
<tr>
<td>TOX</td>
<td>Å</td>
<td>0.0</td>
<td>Oxide thickness</td>
</tr>
<tr>
<td>TUH</td>
<td></td>
<td>1.5</td>
<td>Implant channel mobility temperature exponent (depletion model only)</td>
</tr>
<tr>
<td>ZENH</td>
<td></td>
<td>1.0</td>
<td>Mode flag (enhancement). Set ZENH=0.0 for depletion mode.</td>
</tr>
</tbody>
</table>
### Effective Width and Length Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEL (WDEL)</td>
<td>μm</td>
<td>0.0</td>
<td>Channel length reduction on each side</td>
</tr>
<tr>
<td>LATD (LD)</td>
<td>μm</td>
<td>1.7 \cdot XJ</td>
<td>Lateral diffusion on each side</td>
</tr>
<tr>
<td>LMLT</td>
<td>T</td>
<td>1.0</td>
<td>Length shrink factor</td>
</tr>
<tr>
<td>OXETCH</td>
<td>μm</td>
<td>0.0</td>
<td>Oxide etch</td>
</tr>
<tr>
<td>WMLT</td>
<td></td>
<td>1.0</td>
<td>Diffusion layer and width shrink factor</td>
</tr>
</tbody>
</table>

### Threshold Voltage Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSS (NFS)</td>
<td>cm(^{-2}) \cdot V(^{-1})</td>
<td>0.0</td>
<td>Number of fast surface states</td>
</tr>
<tr>
<td>NWM</td>
<td></td>
<td>0.0</td>
<td>Narrow width modifier</td>
</tr>
<tr>
<td>SCM</td>
<td></td>
<td>0.0</td>
<td>Short-channel drain source voltage multiplier</td>
</tr>
<tr>
<td>VT (VTO)</td>
<td>V</td>
<td>0.0</td>
<td>Extrapolated threshold voltage</td>
</tr>
<tr>
<td>XJ</td>
<td>μm</td>
<td>1.5</td>
<td>Junction depth</td>
</tr>
</tbody>
</table>

### Mobility Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRC</td>
<td>Å/s \cdot cm(^2)</td>
<td>0.0</td>
<td>Field reduction coefficient</td>
</tr>
<tr>
<td>FSB</td>
<td>V(^{1/2})/s/ cm(^2)</td>
<td>0.0</td>
<td>Lateral mobility coefficient</td>
</tr>
<tr>
<td>UB (UO)</td>
<td>cm(^2)/(V\cdot s)</td>
<td>0.0</td>
<td>Low field bulk mobility</td>
</tr>
</tbody>
</table>
The LEVEL 5 MOSFET model has been expanded to include two modes: enhancement and depletion. These two modes are accessed by the flag mode parameter, ZENH.

- **ZENH=1**: This enhancement model (default mode) is a portion of Star-Hspice MOS5 and is identical to AMI SPICE MOS LEVEL 4.
- **ZENH=0**: This depletion model is revised in Star-Hspice (from previous depletion mode) and is identical to AMI SPICE MOS LEVEL 5.

The Star-Hspice enhancement and depletion models are basically identical to the AMI models. However, certain aspects have been revised to enhance performance. Using the Star-Hspice enhancement and depletion models provides access to Star-Hspice features as described below.

The Star-Hspice version of the enhancement and depletion models allows the choice of either SPICE-style or ASPEC-style temperature compensation. For LEVEL 5, the default is TLEV=1, invoking ASPEC style temperature compensation. Setting TLEV=0 invokes SPICE-style temperature compensation.

### Capacitance Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UH</td>
<td>cm$^2$/s</td>
<td>900 (N) (300 \text{ (P)}^*)</td>
<td>Implant - channel mobility * (For depletion model only)</td>
</tr>
<tr>
<td>VST</td>
<td>cm/s</td>
<td>0.0</td>
<td>Saturation velocity</td>
</tr>
</tbody>
</table>

The LEVEL 5 MOSFET model has been expanded to include two modes: enhancement and depletion. These two modes are accessed by the flag mode parameter, ZENH.
CAPOP=6 represents AMI Gate Capacitance in Star-Hspice. CAPOP=6 is the default setting for LEVEL 5 only. The LEVEL 5 models can also use CAPOP =1, 2, 3.

The parameter ACM defaults to 0 in LEVEL 5, invoking SPICE-style parasitics. ACM also can be set to 1 (ASPEC) or to 2 (Star-Hspice). All MOSFET models follow this convention.

The Star-Hspice option SCALE can be used with the LEVEL 5 model; however, option SCALM cannot be used due to the difference in units.

You must specify the following parameters for MOS LEVEL 5: VTO (VT), TOX, UO (UB), FRC, and NSUB (DNB).

**IDS Equations**

The LEVEL 5 IDS equations follow.

**Cutoff Region, \( v_{gs} \leq v_{th} \)**

\[
I_{ds} = 0 \quad \text{(See “Subthreshold Current, \( I_{ds} \)” on page 21-41)}
\]

**On Region, \( v_{gs} \geq v_{th} \)**

\[
I_{ds} = \beta \cdot \left( v_{gs} - v_{th} - \frac{v_{de}}{2} \right) \cdot v_{de} \cdot \frac{2}{3} \cdot \gamma \cdot \left( \Phi_f + v_{de} + v_{sb} \right)^{3/2} - \left( \Phi_f + v_{sb} \right)^{3/2} \]

where:

\[
v_{de} = \min(v_{ds}, v_{dsat})
\]

\[
\beta = UB_{eff} \cdot Cox \cdot \frac{W_{eff}}{L_{eff}}
\]

\[
\Phi_f = 2 \cdot v_{tm} \cdot \ln\left( \frac{DNB}{ni} \right)
\]
and gate oxide capacitances per unit area are calculated by:
\[ \text{coax} = \frac{E_{ox}}{TOX \cdot 1E-10} \, \text{F/m} \]

**Effective Channel Length and Width**

The effective channel length and width in the LEVEL 5 model is determined as follows.
\[
\begin{align*}
W_{eff} &= W_{scaled} \cdot WMLT + OXETCH \\
L_{eff} &= L_{scaled} \cdot LMLT - 2 \cdot (LATD + DEL)
\end{align*}
\]

**Threshold Voltage, \( v_{th} \)**

The model parameter VTO is an extrapolated zero-bias threshold voltage of a large device. The effective threshold voltage, including the device size effects and the terminal voltages, is given by:
\[
v_{th} = v_{bi} + \gamma \cdot (\Phi_f + v_{sb})^{1/2}
\]

where:
\[
v_{bi} = v_{fb} + \Phi_f = VTO - \gamma_0 \cdot \Phi_f^{1/2}
\]
\[
\gamma_0 = \frac{(2 \cdot E_{si} \cdot q \cdot DNB)^{1/2}}{\text{coax}}
\]

*Note: For LEVEL 5 model, you must specify DNB and VTO parameters. The Star-Hspice program computes \( \gamma_0 \) using DNB and ignores the GAMMA model parameter.*
The effective body effect ($\gamma$), including the device size effects, is computed as follows.

$$
\gamma = \gamma_0 \cdot (1 - scf) \cdot (1 + ncf)
$$

If SCM $\leq 0$,

$$
scf = 0
$$

otherwise,

$$
scf = \frac{XJ}{L_{eff}} \cdot \left\{ \left[ 1 + \frac{2xd}{XJ} \cdot (SCM \cdot \nu_{ds} + \nu_{sb} + \Phi_f)^{1/2} \right]^{1/2} - 1 \right\}
$$

If NWM $\leq 0$,

$$
ncf = 0
$$

otherwise,

$$
n cf = \frac{NWM \cdot X_d \cdot (\Phi_f)^{1/2}}{W_{eff}}
$$

where:

$$
X_d = \left( \frac{2 \cdot E_{si}}{q \cdot DNB} \right)^{1/2}
$$

**Saturation Voltage, $\nu_{dsat}$**

The saturation voltage due to channel pinch-off at the drain side is computed by:

$$
\nu_{sat} = \nu_{gs} - \nu_{bi} + \frac{\gamma^2}{2} \cdot \left\{ 1 - \left[ 1 + \frac{4}{\gamma^2} \cdot \left( \nu_{gs} - \nu_{bi} + \Phi_f + \nu_{sb} \right) \right]^{1/2} \right\}
$$

$$
\nu_{dsat} = \nu_{sat}
$$
If ECV is not equal to 1000, then the program modifies $v_{sat}$ to include carrier velocity saturation effect:

$$v_{dsat} = v_{sat} + v_c - (v_{sat}^2 + v_c^2)^{1/2}$$

where:

$$v_c = ECV \cdot L_{eff}$$

**Mobility Reduction, $UB_{eff}$**

The mobility degradation effect in the LEVEL 5 model is computed by:

$$UB_{eff} = \frac{1}{UB + \frac{FRC \cdot (v_{gs} - v_{th})}{TOX} + \frac{vde}{VST \cdot L_e} + FSB \cdot v_{sb}^{1/2}}$$

where:

$$L_e = L_{eff} \quad \text{linear region}$$

$$L_e = L_{eff} - \Delta L \quad \text{saturation region}$$

The channel length modulation effect ($\Delta L$) is defined in the following section.

**Channel Length Modulation**

The LEVEL 5 model includes the channel length modulation effect by modifying the $I_{ds}$ current as follows:

$$I_{ds} = \frac{I_{ds}}{1 - \frac{\Delta L}{L_{eff}}}$$
where:

\[
\Delta L = 1e4 \cdot \left[ \frac{2.73e3 \cdot XJ}{DNB \cdot \ln \left( \frac{1e20}{DNB} \right)} \right]^{1/3} \cdot [(v_{ds} - v_{dsat} + PHI)^{1/3} - PHI^{1/3}]
\]

The \(\Delta L\) is in microns, assuming \(XJ\) is in microns and \(DNB\) is in \(\text{cm}^{-3}\).

**Subthreshold Current, \(I_{ds}\)**

This region of operation is characterized by the Fast Surface State (FSS) if it is greater than \(1e10\). Then the effective threshold voltage, separating the strong inversion region from the weak inversion region, is determined as follows:

\[v_{on} = v_{th} + fast\]

where:

\[fast = \frac{v_{tm} \cdot \left[ 1 + \frac{q \cdot FSS}{cox} + \frac{\gamma}{2 \cdot (\Phi_f + v_{sb})^{1/2}} \right]}{v_{ds} - v_{von}}\]

and \(vt\) is the thermal voltage.

The \(I_{ds}\) is given by:

**Weak Inversion Region, \(v_{gs} < v_{th}\)**

\[I_{ds} = (v_{on}, v_{de}, v_{sb}) \cdot e^{fast}\]

**Strong Inversion Region, \(v_{gs} \geq v_{th}\)**

\[I_{ds} = I_{ds}(v_{gs}, v_{de}, v_{sb})\]
**Note:** The modified threshold voltage \( (v_{on}) \) produced by FSS is also used in strong inversion; that is, in the mobility equations, \( v_{on} \) is used instead of \( V_{th} \).

### Depletion Mode DC Model ZENH=0

The LEVEL 5 MOS model uses depletion mode devices as the load element in contemporary standard n-channel technologies. This model was formulated assuming a silicon gate construction with an ion implant used to obtain the depletion characteristics. A special model is required for depletion devices because the implant used to create the negative threshold also results in a complicated impurity concentration profile in the substrate. The implant profile changes the basis for the traditional calculation of the bulk charge, \( Q_B \). The additional charge from the implant, \( Q_{BI} \), must be calculated.

This implanted layer also causes the formation of an additional channel, offering a conductive pathway through the bulk silicon, as well as through the surface channel. This second pathway can cause difficulties when trying to model a depletion device with existing MOS models. The bulk channel is partially shielded from the oxide interface by the surface channel, and the mobility of the bulk silicon can be substantially higher. Yet with all the differences, a depletion model still can share the same theoretical basis as the Ihantola and Moll gradual channel model.

The depletion model differs from the Ihantola and Moll model as follows:
- Implant charge accounted for
- Finite implant thickness (DP)
- Two channels are assumed: a surface channel and a bulk channel
- Bulk channel has a bulk mobility (\( U_H \))
- Bulk gain is assumed to be different from surface gain

In the depletion model, the gain is lower at low gate voltages and higher at high gate voltages. This variation in gain is the reason the enhancement models cannot generate an accurate representation for a depletion device. The physical
Selecting MOSFET Models: LEVEL 1-40

LEVEL 5 IDS Model

model for a depletion device is basically the same as an enhancement model, except that the depletion implant is approximated by a one-step profile with a depth DP.

Due to the implant profile, the drain current equation must be calculated by region. MOSFET device model LEVEL 5 has three regions: depletion, enhancement, and partial enhancement.

Depletion Region, \( v_{gs} - v_{fb} < 0 \)

The low gate voltage region is dominated by the bulk channel.

Enhancement Region, \( v_{gs} - v_{fb} > 0 \), \( v_{ds} < v_{gs} - v_{fb} \)

The region is defined by high gate voltage and low drain voltage. In the enhancement region, both channels are fully turned on.

Partial enhancement region, \( v_{gs} - v_{fb} > 0 \), \( v_{ds} > v_{gs} - v_{fb} \)

The region has high gate and drain voltages, resulting in the surface region being partially turned on and the bulk region being fully turned on.

IDS Equations, Depletion Model LEVEL 5

The IDS equations for a LEVEL 5 depletion model follow.

Depletion, \( v_{gs} - v_{fb} < 0 \)

\[
I_{ds} = \beta_1 \cdot \left\{ q \cdot NI \cdot v_{de} + c_{av} \cdot \left[ (v_{gs} - v_{fb}) \cdot v_{de} - \frac{v_{de}^2}{2} \right] \right. \\
\left. + \frac{2}{3} \cdot c_{av} \cdot \gamma \cdot \left[ (v_{de} - v_{sb} + \Phi_d)^{3/2} - (v_{sb} + \Phi_d)^{3/2} \right] \right\}
\]
Enhancement, $v_{gs} - v_{fb} > v_{de}$

$$I_{ds} = \beta_1 \cdot \left\{ q \cdot NI \cdot v_{de} - \frac{2}{3} \cdot cav \cdot \gamma \cdot [(v_{de} + v_{sb} + \Phi_d)^{3/2} - (v_{sb} + \Phi_d)^{3/2}] \right\}$$

$$\quad + \beta \cdot \left[ (v_{gs} - v_{fb}) \cdot v_{de} - \frac{v_{de}^2}{2} \right]$$

Partial Enhancement, $v_{gs} - v_{fb} < v_{de}$

$$I_{ds} = \beta_1 \cdot \left\{ q \cdot NI \cdot v_{de} + cav \cdot \left[ (v_{gs} - v_{fb}) \cdot v_{de} - \frac{v_{de}^2}{2} \right] \right\}$$

$$\quad - \frac{2}{3} \cdot cav \cdot \gamma \cdot [(v_{de} + v_{sb} + \Phi_d)^{3/2} - (v_{sb} + \Phi_d)^{3/2}] \right\}$$

$$\quad + \left( \frac{1}{2} \beta_1 \cdot cav \right) \cdot (v_{gs} - v_{fb})^2$$

where:

$$\beta_1 = UH \cdot \frac{W_{eff}}{L_{eff}}$$

$$\beta = UBeff \cdot cox \cdot \frac{W_{eff}}{L_{eff}}$$

$$cav = \frac{cox \cdot cs}{cox + cs}$$

$$cs = \frac{2.77E_{si}}{DP \cdot 1e-4}$$
Threshold Voltage, $v_{th}$

The model parameter $VTO$ is an extrapolated zero-bias threshold voltage for a large device. The effective threshold voltage, including the device size effects and the terminal voltages, is calculated as follows:

$$v_{th} = v_{fb} - \beta_d \cdot [v_{ch} - \gamma \cdot (\Phi_d + v_{sb})^{1/2}]$$

where:

$$v_{fb} = VTO + \beta_d \cdot (v_{ch} - \gamma_0 \cdot \Phi_d^{1/2})$$

$$\beta_d = \frac{UH \cdot cav}{UB \cdot cox}$$

$$v_{ch} = \frac{q \cdot NI}{cav}$$

$$\gamma_0 = \frac{(2 \cdot E_{st} \cdot q \cdot na)^{1/2}}{cav}$$

and:

$$\Phi_d = v_{im} \cdot \ln\left(\frac{DNB \cdot nd}{n_i^2}\right)$$

$$nd = \frac{NI \cdot 1e4}{DP}$$

and:

$$v_{de} = \min(v_{ds}, v_{dsat})$$

The saturation voltage, threshold voltage, and effective $\gamma$ are described in the following sections.
The effective \( \gamma \), including small device size effects, is computed as follows:

\[
\gamma = \gamma_0 \cdot (1 - scf) \cdot (1 + ncf)
\]

where:

If SCM \( \leq 0 \),

\( scf = 0 \)

otherwise,

\[
scf = \frac{X J}{L_{eff}} \cdot \left\{ \left[ 1 + \frac{2x d}{X J} \cdot (\text{SCM} \cdot v_{ds} + v_b + \Phi_d)^{1/2} \right]^{1/2} - 1 \right\}
\]

If NWM \( \leq 0 \),

\( ncf = 0 \)

otherwise,

\[
ncf = \frac{NWM \cdot X_d \cdot \Phi_d^{1/2}}{W_{eff}}
\]

where:

\[
X_d = \left( \frac{2 \cdot E_{si}}{q \cdot DNB} \right)^{1/2}
\]
Note: When $v_{gs} \leq v_{th}$, the surface is inverted and a residual DC current exists. When $v_{sb}$ is large enough to make $v_{th} > v_{in\text{th}}$, then $v_{th}$ is used as the inversion threshold voltage. In order to determine the residual current, $v_{in\text{th}}$ is inserted into the $I_{ds}$, $v_{sat}$, and mobility equation in place of $v_{gs}$ (except for $v_{gs}$ in the exponential term of the subthreshold current). The inversion threshold voltage at a given $v_{sb}$ is $v_{in\text{th}}$, which is computed as:

$$v_{in\text{th}} = v_{fb} - \frac{q \cdot N I_{COX}}{v_{sb}}$$

**Saturation Voltage, $v_{dsat}$**

The saturation voltage ($v_{sat}$) is determined as:

$$v_{sat} = v_{gs} - v_{fb} + v_{ch} + \frac{\gamma^2}{2} \cdot \left\{ 1 - \left[ 1 + \frac{4}{\gamma^2} \cdot (v_{gs} - v_{fb} + v_{ch} + v_{sb} + \Phi_d) \right]^{1/2} \right\}$$

$$v_{dsat} = v_{sat}$$

If ECV is not equal to 1000 (V/µm), Star-Hspice modifies $v_{sat}$ to include the carrier velocity saturation effect.

$$v_{dsat} = v_{sat} + v_{c} - (v_{sat} + v_{c})^{1/2}$$

where:

$$v_{c} = ECV \cdot L_{eff}$$

**Mobility Reduction, $UB_{eff}$**

The surface mobility ($UB$) is dependent upon terminal voltages as follows:

$$UB_{eff} = \frac{1}{UB} + \frac{FRC \cdot (v_{gs} - v_{th})}{TOX} + \frac{v_{de}}{VST \cdot Ie} + FSB \cdot v_{sb}^{1/2}$$
where:

\[ L_e = L_{\text{eff}} \]  
**Linear region**

\[ L_e = L_{\text{eff}} - \Delta L \]  
**Saturation region**

The channel length modulation effect (\( \Delta L \)) is defined next.

**Channel Length Modulation**

The channel length modulation effect is included by modifying the \( I_{ds} \) current as:

\[
I_{ds} = \frac{I_{ds}}{1 - \frac{\Delta L}{L_{\text{eff}}}}
\]

where:

\[
\Delta L = 1e4 \cdot \left[ \frac{2.73e3 \cdot XJ}{na1 \cdot \ln\left(\frac{1e20}{na1}\right)} \right]^{1/3} \cdot \left[ (v_{ds} - v_{dsat} + PHI)^{1/3} - PHI^{1/3} \right]^{-1}
\]

The \( \Delta L \) parameter is in microns, assuming \( XJ \) is in microns and \( na1 \) is in cm\(^{-3}\).

**Subthreshold Current, \( I_{ds} \)**

When device leakage currents become important for operation near or below the normal threshold voltage, the subthreshold characteristics are considered. The Star-Hspice LEVEL 5 model uses the subthreshold model only if the number of fast surface states (that is, the FSS) is greater than 1e10. An effective threshold voltage (von) is then determined:

\[
von = v_{th} + \text{fast}
\]
where:

\[ fast = v_{tm} \cdot \left[ 1 + q \cdot \frac{FSS}{Cox} + \frac{\gamma}{2 \cdot (\Phi_d + v_{sb})^{1/2}} \right] \]

If \( v_{on} < v_{in\_th} \), then \( v_{in\_th} \) is substituted for \( v_{on} \).

**Note:** The Star-Hspice LEVEL 5 model uses the following subthreshold model only if \( v_{gs} < v_{on} \) and the device is either in partial or full enhancement mode. Otherwise, it uses the model in enhancement mode (\( ZENH=1 \)). The subthreshold current calculated below includes the residual DC current.

If \( v_{gs} < v_{on} \) then,

**Partial Enhancement, \( v_{gs} - v_{fb} < v_{de} \)**

\[
I_{ds} = \beta_1 \cdot \left\{ q \cdot N_I \cdot v_{de} + cav \cdot \left[ (v_{on} - v_{fb}) \cdot v_{de} - \frac{v_{de}^2}{2} \right] \right. \\
\left. - \frac{2}{3} \cdot cav \cdot \gamma \cdot [(v_{de} + v_{sb} + \Phi_d)^{3/2} - (v_{sb} + \Phi_d)^{3/2}] \right\} \\
+ \frac{1}{2} \cdot \left[ \beta \cdot e \frac{v_{gs} - v_{on}}{fast} - \beta_1 \cdot cav \right] \cdot (v_{on} - v_{fb})^2
\]
Full Enhancement, $v_{gs} - v_{fb} > 0$

$$I_{ds} = \beta_1 \left\{ q \cdot N_I \cdot \frac{vde - 2}{3} \cdot cav \cdot \gamma \left[ (vde + v_{sb} + \Phi_d)^{3/2} - (v_{sb} + \Phi_d)^{3/2} \right] \right\}$$

$$+ \beta \cdot \left[ (v_{on} - v_{fb}) \cdot vde - \frac{vde^2}{2} \right] \cdot \frac{v_{gs} - v_{on}}{\text{fast}}$$

Example

FILE ML5IV.SP HSPICE LEVEL 5 MODEL EXAMPLES

*OUTPUT CHARACTERISTICS FOR ENHANCEMENT & DEPLETION MODE

.OPT ACCT LIST CO=132

.OP

VDS 3 0 .1
VGS 2 0
M1 1 2 0 0 MODEN L=20U W=20U

Enhancement Mode

.MODEL MODEN NMOS LEVEL=5
+ VT=.7 TOX=292 FRC=2.739E-2 DNB=2.423E16 UB=642.8
+ OXETCH=-.98 XJ=.29 LATD=.34 ECV=4 VST=5.595E7
+ FSB=7.095E-5 SCM=.4 FSS=2.2E11 NWM=.93 PHI=.61
+ TCV=1.45E-3 PTC=9E-5 BEX=1.8

* 

VIDS 3 1

.DC VGS 0 5 0.2

.PRINT DC I(VIDS) V(2)

.PLOT DC I(VIDS)

$$$$$$

.ALTER

$$$$$$

M1 1 2 0 0 MODDP L=20U W=20U
Selecting MOSFET Models: LEVEL 1-40

LEVEL 5 IDS Model

**Depletion Mode**

```
.DEVICE MODDP NMOS LEVEL=5 ZENH=0.
+ VT=-4.0 FRC=.03 TOX=800 DNB=6E14 XJ=0.8 LATD=0.7
+ DEL=0.4 CJ=0.1E-3 PHI=0.6 EXA=0.5 EXP=0.5 FSB=3E-5
+ ECV=5 VST=4E7 UB=850 SCM=0.5 NI=5.5E11 DP=0.7 UH=1200
*
.END
```
LEVEL 6 and LEVEL 7 IDS: MOSFET Model

These models represent ASPEC, MSINC, and ISPICE MOSFET model equations. The only difference between LEVEL 6 and LEVEL 7 equations is the handling of the parasitic elements and the method of temperature compensation. See “Mobility Parameters” on page 21-10 and “Channel Length Modulation” on page 21-15 for those model parameters.

LEVEL 6 and LEVEL 7 Model Parameters

The LEVEL 6 and LEVEL 7 model parameters are listed in this section.

Basic Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>IDS equation selector</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LEVEL=6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lattin-Jenkins-Grove model, using ASPEC-style parasitics</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> When option ASPEC is invoked, the program automatically selects LEVEL 6. However, specifying LEVEL 6 does not automatically invoke option ASPEC. (For complete information, see the end of the LEVEL 6 section.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LEVEL=7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lattin-Jenkins-Grove model, using SPICE-style parasitics</td>
</tr>
<tr>
<td>CLM (GDS)</td>
<td></td>
<td>0.0</td>
<td>Channel length modulation equation selector</td>
</tr>
<tr>
<td>DNB (NSUB)</td>
<td>1/cm³</td>
<td>1.0e15</td>
<td>Substrate doping</td>
</tr>
<tr>
<td>DNS (NI)</td>
<td>1/cm³</td>
<td>0.0</td>
<td>Surface substrate doping</td>
</tr>
<tr>
<td>ECRIT (ESAT)</td>
<td>V/cm</td>
<td>0.0</td>
<td>Drain-source critical field. Use zero to indicate an infinite value, typically 40,000 V/cm.</td>
</tr>
</tbody>
</table>
### GAMMA

**Units**: \( V^{1/2} \)

**Default**: Not specified

**Description**: Body effect factor. If this parameter is not input, GAMMA is calculated from DNB.

- GAMMA is the body effect when \( v_{sb} < V_{B0} \).
- If \( v_{sb} > V_{B0} \), LGAMMA is used.
- Using GAMMA, LGAMMA, and \( V_{B0} \) allows a two-step approximation of a non-homogeneous substrate.

### LGAMMA

**Units**: \( V^{1/2} \)

**Default**: 0.0

**Description**: This parameter is the body effect factor when \( v_{sb} > V_{B0} \).

- When the Poon-Yau GAMMA expression is used, LGAMMA is junction depth, in microns. In this case LGAMMA is multiplied by SCALM.

### MOB

**Units**: \( \text{cm}^2 / (V \cdot s) \)

**Default**: 0.0

**Description**: Mobility equation selector

- 600 (N)
- 250 (P)

This parameter is the low field bulk mobility. It is calculated from KP if KP is supplied.

### NWM

**Units**: 0.0

**Default**: 0.0

**Description**: Narrow width modulation of GAMMA

### SCM

**Units**: 0.0

**Default**: 0.0

**Description**: Short-channel modulation of GAMMA

### UO (UB, UBO)

**Units**: \( \text{cm}^2 / (V \cdot s) \)

**Default**: 600 (N), 250 (P)

**Description**: This parameter is the low field bulk mobility. It is calculated from KP if KP is supplied.

### UPDATE

**Units**: 0.0

**Default**: 0.0

**Description**: Selector for different version of LEVEL 6 model. For UPDATE=1 and 2 alternate saturation voltage, mobility equation (MOB=3) and series resistances RS and RD are modified to be compatible with ASPEC.

- UPDATE=1 provides continuous Multi-Level GAMMA model.
**Effective Length and Width Parameters**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEL</td>
<td>m</td>
<td>0.0</td>
<td>Channel length reduction on each side. DEL is applicable in most MOSFET models. An exception is the BSIM (LEVEL 13) model, where DEL is not present. DELscaled = DEL ⋅ SCALM</td>
</tr>
<tr>
<td>LD (DLAT, LATD)</td>
<td>m</td>
<td></td>
<td>Lateral diffusion into channel from source and drain diffusion. If LD and XJ are unspecified, LD Default=0.0. When LD is unspecified but XJ is specified, LD is calculated from XJ. LD Default=0.75 ⋅ XJ. LDscaled = LD ⋅ SCALM</td>
</tr>
</tbody>
</table>
### MOSFET Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as LD, but if LDAC is included in the .MODEL statement, it replaces LD in the Leff calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>LREF</td>
<td>m</td>
<td>0.0</td>
<td>Channel length reference</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$L_{REFscaled} = L_{REF} \cdot SCALM$</td>
</tr>
<tr>
<td>LMLT</td>
<td></td>
<td>1.0</td>
<td>Length shrink factor</td>
</tr>
<tr>
<td>WD</td>
<td>m</td>
<td>0.0</td>
<td>Lateral diffusion into channel from bulk along width</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$W_{scaled} = W \cdot SCALM$</td>
</tr>
<tr>
<td>WDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as WD, but if WDAC is included in the .MODEL statement, it replaces WD in the Weff calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>WMLT</td>
<td></td>
<td>1.0</td>
<td>Diffusion layer and width shrink factor</td>
</tr>
<tr>
<td>WREF</td>
<td>m</td>
<td>0.0</td>
<td>Channel width reference</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$W_{REFscaled} = W_{REF} \cdot SCALM$</td>
</tr>
<tr>
<td>XJ</td>
<td>m</td>
<td>0.0</td>
<td>Metallurgical junction depth</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$X_{Jscaled} = X_{J} \cdot SCALM$</td>
</tr>
<tr>
<td>XL (DL, LDEL)</td>
<td>m</td>
<td>0.0</td>
<td>Accounts for masking and etching effects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$X_{Lscaled} = X_{L} \cdot SCALM$</td>
</tr>
<tr>
<td>XW (DW, WDEL)</td>
<td>m</td>
<td>0.0</td>
<td>Accounts for masking and etching effects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$X_{Wscaled} = X_{W} \cdot SCALM$</td>
</tr>
</tbody>
</table>
Threshold Voltage Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDS</td>
<td>0.0</td>
<td>Field, drain to source, controls reduction of threshold due to source-drain electric field</td>
<td></td>
</tr>
<tr>
<td>LND</td>
<td>µm/V</td>
<td>0.0</td>
<td>ND length sensitivity</td>
</tr>
<tr>
<td>LN0</td>
<td>µm</td>
<td>0.0</td>
<td>N0 length sensitivity</td>
</tr>
<tr>
<td>ND</td>
<td>1/V</td>
<td>0.0</td>
<td>Drain subthreshold factor. Typical value=1.</td>
</tr>
<tr>
<td>N0</td>
<td>0.0</td>
<td>Gate subthreshold factor. Typical value=1.</td>
<td></td>
</tr>
<tr>
<td>NFS (DFS, NF)</td>
<td>cm⁻²⋅V⁻¹</td>
<td>0.0</td>
<td>Fast surface state density</td>
</tr>
<tr>
<td>NWE</td>
<td>m</td>
<td>0.0</td>
<td>Narrow width effect, direct compensation of VTO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NWEscaled = NWE ⋅ SCALM</td>
</tr>
<tr>
<td>UFDS</td>
<td>0.0</td>
<td>High field FDS</td>
<td></td>
</tr>
<tr>
<td>VFDS</td>
<td>V</td>
<td>0.0</td>
<td>Reference voltage for selection of FDS OR UFDS:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FDS used if vds ≤ VFDS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UFDS used if vds &gt; VFDS</td>
</tr>
<tr>
<td>VSH</td>
<td>V</td>
<td>0.0</td>
<td>Threshold voltage shifter for zero-bias threshold voltage (VTO) reduction as a function of the ratio of LD to Leff</td>
</tr>
<tr>
<td>VTO (VT)</td>
<td>V</td>
<td>0.0</td>
<td>Zero-bias threshold voltage. This parameter is calculated if not specified (see “Common Threshold Voltage Parameters” on page 20-50).</td>
</tr>
<tr>
<td>WEX</td>
<td></td>
<td></td>
<td>Weak inversion exponent</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 1-40  LEVEL 6 and LEVEL 7 IDS: MOSFET Model

Alternate Saturation Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIC</td>
<td></td>
<td>0.0</td>
<td>Subthreshold model selector</td>
</tr>
<tr>
<td>WND</td>
<td>μm/V</td>
<td>0.0</td>
<td>ND width sensitivity</td>
</tr>
<tr>
<td>WN0</td>
<td>μm</td>
<td>0.0</td>
<td>N0 width sensitivity</td>
</tr>
</tbody>
</table>

UPDATE Parameter for LEVEL 6 and LEVEL 7

The general form of the $I_{ds}$ equation for LEVEL 6 is the same as the LEVEL 2 MOS model, but the small size effects, mobility reduction, and channel length modulation are included differently. Also, you can use LEVEL 6 models to model the MOS transistors with ion-implanted channels through the multi-level GAMMA capability.

The LEVEL 6 model represents the ASPEC, MSINC, and ISPIECE programs MOSFET model. Use the enhanced model parameter UPDATE to invoke different versions of the LEVEL 6 model, as described next.
**UPDATE=0**

This is the original LEVEL 6 model in Star-Hspice which is not quite compatible with the ASPEC model. It has some discontinuities in weak inversion, mobility equations (MOB=3), and multi-level GAMMA equations.

**UPDATE=1**

This enhanced version of the LEVEL 6 model contains improved multi-level GAMMA equations. The saturation voltage, drain-source current, and conductances are continuous.

**UPDATE=2**

This version of the LEVEL 6 model is compatible with the ASPEC model. The multi-level GAMMA model is not continuous, which is the case in the ASPEC program. See “ASPEC Compatibility” on page 21-86.

Set UPDATE to 1.0 to implement changes to the device equations. Set UPDATE to 1.0 or 2 to implement the default handling of RS and RD are implemented. These values and changes provide a more accurate ASPEC model.

**UPDATE=1 or 2 then,**

\[
\begin{align*}
\text{TOX} &= 690 \\
\text{UO (UB)} &= 750 \text{ cm}^2/(V \cdot s) \quad \text{(N-ch)} \\
\text{UTRA (F3)} &= 0.0
\end{align*}
\]

**UPDATE=0 then,**

\[
\begin{align*}
\text{TOX} &= 1000 \\
\text{UO (UB)} &= 750 \text{ cm}^2/(V \cdot s) \quad \text{(N-ch)} \\
\text{UTRA (F3)} &= 0.0
\end{align*}
\]
Calculation of RD and RS in the MOSFET changes as follows when LDIF is not specified:

**UPDATE=1 or 2 and LDIF=0,**

\[
RD = \frac{(RD + NRD \cdot RL)}{M}
\]

\[
RS = \frac{(RS + NRS \cdot RL)}{M}
\]

**Note:** The ASPEC program does not use the multiplier M.

**LDIF ≠0,**

\[
RD = \frac{LATD_{scaled} + LDIF_{scaled}}{Weff} \cdot RD + NRD \cdot \frac{RL}{M}
\]

\[
RS = \frac{LATD_{scaled} + LDIF_{scaled}}{Weff} \cdot RS + NRS \cdot \frac{RL}{M}
\]

The \(vde\) in the mobility equations for alternate saturation model changes as follows:

\[
vde = \min\left(\frac{vds}{vfa}, vsat\right), UPDATE = 1 \text{ or } 2
\]

\[
vde = \min(vds, vfa \cdot vsat), UPDATE = 0
\]

The saturation voltage in the impact ionization equation is as follows:

\[
vdsat = vfa \cdot vsat, \quad UPDATE = 1 \text{ or } 2
\]

\[
vdsat = vsat, \quad UPDATE = 0
\]
Mobility equation MOB=3 changes as follows:

**UPDATE= 1 or 2 and** \((vgs - vth)^F2 > VF1\),

\[
ueff = \frac{UB}{F4 + (F1 - F3) \cdot VF1 + F3 \cdot (vgs - vth)^F2}
\]

**UPDATE= 0 and** \((vgs - vth)^F2 > VF1\),

\[
ueff = \frac{UB}{F4 + F3 \cdot (vgs - vth)^F2}
\]

**LEVEL 6 Model Equations, UPDATE=0,2**

**IDS Equations**

\[
ids = \beta \cdot \left( \left( vgs - vbi - \frac{\eta \cdot vde}{2} \right) \cdot vde - \frac{2}{3} \gamma \cdot \left( (PHI + vde + vsb)^{3/2} - (PHI + vsb)^{3/2} \right) \right)
\]

where:

\[
vde = \min(vds, vdsat)
\]

\[
\eta = 1 + \frac{NWEscaled}{weff}
\]

\[
\beta = ueff \cdot COX \cdot \frac{weff}{Leff}
\]

Include the narrow-width effect through \(\eta\), \(vbi\), and \(\gamma\) values. For the narrow-width effect, specify model parameters NWE and/or NWM. Include the short-channel effect through parameters \(vbi\) and \(\gamma\).
Effective Channel Length and Width

The model calculates effective channel length and width from the drawn length and width as follows:

\[
\begin{align*}
    \text{leff} &= \text{Lscaled} \cdot \text{LMLT} + \text{XLScaled} - 2 \cdot (\text{LDscaled} + \text{DELscaled}) \\
    \text{weff} &= M \cdot (\text{Wscaled} \cdot \text{WMLT} + \text{XWscaled} - 2 \cdot \text{WDscaled}) \\
    \text{LREFeff} &= \text{LREFscaled} \cdot \text{LMLT} + \text{XLScaled} - 2 \cdot (\text{LDscaled} + \text{DELscaled}) \\
    \text{WREFeff} &= M \cdot (\text{WREFscaled} \cdot \text{WMLT} + \text{XWscaled} - 2 \cdot \text{WDscaled})
\end{align*}
\]

Threshold Voltage, \( v\text{th} \)

The model determines effective threshold voltage as follows:

\[
v\text{th} = v\text{bi} + \gamma \cdot (\text{PHI} + v\text{sb})^{1/2}
\]

The built-in voltage \( v\text{bi} \) and \( \gamma \) is computed differently depending on the specified model parameters.

Single-Gamma, VBO=0

When model parameter VBO is zero, the single-gamma model is used. In this case the model treats the parameter LGAMMA as a junction depth. It then modifies the GAMMA parameter for short-channel effect by the scf factor, which is computed using the Poon and Yau formulation. In this case LGAMMA is multiplied by the SCALM option.

\[
\text{scf} = 1 - \frac{\text{LGAMMA}}{\text{leff}} \cdot \left\{ \left[ 1 + 2 \cdot \frac{\text{LAMBDA}}{\text{LGAMMA}} \cdot (\text{PHI} + v\text{sb})^{1/2} \right]^{1/2} - 1 \right\}
\]

Specify the model parameter XJ to modify the model parameter GAMMA by the short-channel factor (gl).

\[
\text{gl} = 1 - \frac{\text{XJscaled}}{\text{leff}} \cdot \left\{ \left[ 1 + \frac{2 \cdot \text{LAMBDA}}{\text{XJscaled}} \cdot (\text{PHI} + v\text{sb} + \text{SCM} \cdot \text{vds})^{1/2} \right]^{1/2} - 1 \right\}
\]
The $gl$ factor generally replaces the $scf$ factor for the multilevel GAMMA model.

The model also includes the narrow-width effect by modifying GAMMA with the $gw$ factor, which is computed as:

$$gw = \frac{1 + NWM \cdot xd}{weff}$$

where:

$$xd = \left( \frac{2 \cdot \varepsilon \text{s}i}{q \cdot DNB} \right)^{1/2}$$

Finally, the effective $\gamma$, including short-channel and narrow width effects, is:

$$\gamma = GAMMA \cdot gw \cdot gl \cdot scf$$

**Effective Built-in Voltage, $v_{bi}$**

The model includes the narrow-width effect, which is the increase in threshold voltage due to extra bulk charge at the edge of the channel, by modifying $v_{bi}$ if you specify the model parameter NWE.

The short-channel effect, which is the decrease in threshold voltage due to the induced potential barrier-lowering effect, is included through $v_{bi}$ modification. To include this effect, you must specify the model parameter FDS and/or UFDS and VFDS.

The expressions for $v_{bi}$, which sum up the above features, are:

\[ v_{bi} \leq VFDS, \text{ or } VFDS = 0 \]

\[ v_{bi} = \frac{V_{TO} - \gamma \cdot PHI^{1/2} + (\eta - 1) \cdot (PHI + vsb)}{L_{scaled}^{eff}} \cdot \frac{\varepsilon \text{s}i}{COX \cdot L_{eff}} \cdot FDS \cdot v_{ds} \]

\[ v_{ds} > VFDS \]

\[ v_{bi} = \frac{V_{TO} - \gamma \cdot PHI^{1/2} + (\eta - 1) \cdot (PHI + vsb)}{L_{scaled}^{eff}} \cdot \frac{\varepsilon \text{s}i}{COX \cdot L_{eff}} \cdot \left[ (FDS - UFDS) \cdot VFDS + UFDS \cdot v_{ds} \right] \]
The above equations describe piecewise linear variations of \( v_{bi} \) as a function of \( v_{ds} \). If you do not specify \( V_{FDS} \), the first equation for \( v_{bi} \) is used.

**Note:** *Star-Hspice calculates model parameters such as \( V_{TO} \), \( \Phi \), and \( \Gamma \), if they are not user-specified* (see “Common Threshold Voltage Parameters” on page 20-50).

### Multi-Level Gamma, \( V_{BO} > 0 \)

Use Multi-Level Gamma to model MOS transistors with Ion-Implanted channels. The doping concentration under the gate is approximated as step functions. \( \Gamma \) and \( \Phi \), respectively, represent the corresponding body effects coefficients for the implant layer and the substrate. Figure 21-1 shows the variation of \( v_{th} \) as a function of \( v_{sb} \) for Multi-Level Gamma.

**Figure 21-1: Threshold Voltage Variation**

![Threshold Voltage Variation Diagram](image)

The threshold voltage equations for different regions are as follows:
Channel Depletion Region is in the Implant Layer, \( v_{sb} \leq V_{BO} \)

\[
\gamma = \gamma_i
\]

\[
v_{th} = v_{bi} + \gamma_i \cdot (v_{sb} + \Phi) \frac{1}{2}
\]

\[
v_{bi} = V_{TO} - \gamma_i \cdot (\Phi) \frac{1}{2}
\]

Channel Depletion Region is Expanded into the Bulk, \( v_{sb} > V_{BO} \)

\[
\gamma = \gamma_b
\]

\[
v_{th} = v_{bi} + \gamma_b \cdot (v_{sb} + \Phi) \frac{1}{2}
\]

\[
v_{bi} = v_{tb} - \gamma_b \cdot (\Phi) \frac{1}{2}
\]

In order for the threshold voltage to be continuous at \( v_{sb} = V_{BO} \), \( v_{tb} \) must be:

\[
v_{tb} = V_{TO} + (\gamma_i - \gamma_b) \cdot [(V_{BO} + \Phi) \frac{1}{2} - (\Phi) \frac{1}{2}]
\]

The \( \gamma_i \) and \( \gamma_b \) are effective values of \( \text{GAMMA} \) and \( \text{LGAMMA} \), respectively. The model computes them as \( \gamma \) in single-gamma models, except the \( \text{scf factor} \) is 1.0.

\[
\gamma_i = \text{GAMMA} \cdot gw \cdot gl
\]

\[
\gamma_b = \text{LGAMMA} \cdot gw \cdot gl
\]

**Effective Built-in Voltage, \( v_{bi} \) for \( V_{BO} > 0 \)**

For \( v_{ds} \leq V_{FDS} \),

if \( v_{sb} \leq V_{BO} \),

\[
v_{bi} = V_{TO} - \gamma_i \cdot (\Phi) \frac{1}{2} + (\eta - 1) \cdot (\Phi + v_{sb}) - \frac{L_{Dscaled}}{Leff} \cdot V_{SH} - \frac{\varepsilon_{si}}{C_{OX \cdot Leff}} \cdot F_{DS} \cdot v_{ds}
\]
if $v_{sb} > V_{BO}$,

$$v_{bi} = V_{TO} - \gamma b \cdot (PHI)^{1/2} + (\gamma_i - \gamma b) \cdot [(V_{BO} + PHI)^{1/2} - (PHI)^{1/2}] + (\eta - 1)$$

$$+ (PHI + v_{sb}) - \frac{L_{scaled}}{L_{eff}} \cdot VSH - \frac{\varepsilon_{si}}{COX \cdot L_{eff}} \cdot FDS \cdot v_{ds}$$

For $v_{ds} > VF_{DS}$,

if $v_{sb} \leq V_{BO}$,

$$v_{bi} = V_{TO} - \gamma i \cdot (PHI)^{1/2} + (PHI + v_{sb}) - \frac{L_{scaled}}{L_{eff}} \cdot VSH - \frac{\varepsilon_{si}}{COX \cdot L_{eff}} \cdot [(FDS - UFDS) \cdot VF_{DS} + UFDS \cdot v_{ds}]$$

if $v_{sb} > V_{BO}$,

$$v_{bi} = V_{TO} - \gamma b \cdot (PHI)^{1/2} + (\gamma_i - \gamma b) \cdot [(V_{BO} + PHI)^{1/2} - (PHI)^{1/2}] + (\eta - 1)$$

$$\cdot PHI + v_{sb} - \frac{L_{scaled}}{L_{eff}} \cdot VSH - \frac{\varepsilon_{si}}{COX \cdot L_{eff}} \cdot [(FDS - UFDS) \cdot VF_{DS} + UFDS \cdot v_{ds}]$$

**Saturation Voltage, $v_{dsat}$ (UPDATE=0,2)**

The saturation voltage due to channel pinch-off at the drain side is determined by:

$$v_{sat} = \frac{v_{gs} - v_{bi}}{\eta} + \frac{1}{2(\eta)} \cdot \left[ 1 - \left( \frac{2}{\gamma} \right)^2 \cdot \left( \frac{v_{gs} - v_{bi}}{\eta} + PHI + v_{sb} \right) \right]^{1/2}$$

The reduction of saturation voltage due to the carrier velocity saturation effect is included as follows:

$$v_{dsat} = v_{sat} + v_c - (v_{sat}^2 + v_c^2)^{1/2}$$

where $v_c$ is determined if model parameter ECRIT >0, or VMAX >0, and KU≤1. If both ECRIT and VMAX are specified, then only the VMAX equation is used.
However, the VMAX equation is not used if MOB=4 or MOB=5, since these mobility equations already contain a velocity saturation term.

\[ v_c = ECRIT \cdot L_{eff} \]

or

\[ v_c = \frac{VMAX \cdot L_{eff}}{u_{eff}} \]

Because \( v_{sb} > V_{BO} \), \( \gamma \) is switched from \( \gamma_i \) to \( \gamma_b \), the ids, vsat, and conductances are not continuous. This problem is demonstrated in the following example. To correct the discontinuity problem, specify model parameter \( \text{UPDATE}=1 \). The next section discusses this improvement.

**Example**

This is an example of a multi-level gamma model, UPDATE=0.

```
$ TGAM2.SP---MULTI-LEVEL GAMMA MODEL, UPDATE=0
* THIS DATA IS FOR THE COMPARISON OF MULTI-LEVEL GAMMA
* UPDATE=0 OR 2 AND THE IMPROVED MULTI-LEVEL GAMMA UPDATE=1.
*
.OPTIONS ASPEC NOMOD POST VNTOL=.1U RELI=.001 RELV=.0001
*
.MODEL NCH NMOS BULK=99 UPDATE=0
 + FDS=0.9 KU=1.6 MAL=0.5 MOB=1 CLM=1
 + LATD=0.2 PHI=0.3 VT=0.9 GAMMA=0.72 LGAMMA=0.14
 + VB0=1.2 F1=0.08 ESAT=8.6E+4 KL=0.05
 + LAMBDA=3.2U UB=638 F3=0.22
 + KA=0.97 MBL=0.76 NFS=1.0E+12 WIC=0
 + LDEL=0.084 WDEL=0.037 TOX=365 VSH=0.7
*
VD 1 0 5
VB 0 99 0
VG 2 0 1
MA 1 2 0 99 NCH 26.0 1.4
.DC VB 1.0 1.3 .01
.PRINT IDS=PAR('I(MA)') VTH=PAR('LV9(MA)') VDSAT=PAR('LV10(MA)')
.PRINT GM=PAR('LX7(MA)') GDS=PAR('LX8(MA)') GMBS=PAR('LX9(MA)')
.END
```
Figure 21-2: Variation of IDS, VTH and VDSAT for UPDATE=0

Figure 21-3: Variation of GM, GDS and GMBS for UPDATE=0
Each plot compares $I_{DS}$, $V_{TH}$, $V_{DSAT}$, $GM$, $GDS$ and $GMBS$ as a function of $V_{SB}$ for $UPDATE=0$.

**Improved Multi-Level Gamma, UPDATE=1**

As demonstrated in previous sections, the regular Multi-Level Gamma displays some discontinuities in saturation voltage and drain current. This is because when $V_{SB}$ is less than $V_{BO}$, $\gamma_i$ is set to $\gamma_i$ and used in $i_{DS}$ and vsat calculation. This is not correct; if $(V_{DS} + V_{SB})$ exceeds $V_{BO}$, the depletion regions at drain side expands into the substrate region, which means $\gamma_b$ must be used instead of $\gamma_i$ in $v_{sat}$ computation. Since $v_{sat} = v_{gs} - v_{th}$ (drain), the threshold voltage at drain is computed using $\gamma_i$ for $V_{SB}<V_{BO}$. As a result, the existing model overestimates the threshold voltage, $(\gamma_i > \gamma_b)$, and, in turn, underestimates the saturation voltage and the drain current in the saturation region.

This causes a discontinuous increase in the saturation drain current crossing from the region $V_{SB}<V_{BO}$ to the region $V_{SB}>V_{BO}$.

There are two major differences between the improved Multi-Level model and the regular Multi-Level model: the saturation voltage equation and the drain current equations. To use the improved model, set the model parameter to $UPDATE=1$.

**Example**

This is an example of a multi-level gamma model, $UPDATE=2$.

```
$ TGAM2.SP---MULTI-LEVEL GAMMA MODEL, UPDATE=2
* THIS DATA IS FOR THE COMPARISON OF MULTI-LEVEL GAMMA
* UPDATE=0 OR 2 AND THE IMPROVED MULTI-LEVEL GAMMA UPDATE=1.
*
.OPTIONS ASPEC NOMOD POST VNTOL=.1U RELI=.001 RELV=.0001
*
.MODEL NCH NMOS BULK=99 UPDATE=1
+ FDS=0.9 KU=1.6 MAL=0.5 MOB=1 CLM=1
+ LATD=0.2 PHI=0.3 VT=0.9 GAMMA=0.72 LGAMMA=0.14
+ VBO=1.2 F1=0.08 ESAT=8.6E+4 KL=0.05
+ LAMBDA=3.2U UB=638 F3=0.22
+ KA=0.97 MBL=0.76 NFS=1.0E+12 WIC=0
+ LDEL=0.084 WDEL=0.037 TOX=365 VSH=0.7
```
Selecting MOSFET Models: LEVEL 1-40

LEVEL 6 and LEVEL 7 IDS: MOSFET Model

*  
VD 1 0 5  
VB 0 99 0  
VG 2 0 1  
MA 1 2 0 99  NCH 26.0 1.4  
.DC  VB 1.0 1.3 .01  
.PRINT  IDS=PAR('I(MA)')  VTH=PAR('LV9(MA)')  VDSAT=PAR('LV10(MA)')  
.PRINT  GM=PAR('LX7(MA)')  GDS=PAR('LX8(MA)')  GMBS=PAR('LX9(MA)')  
.END  

Figure 21-4: Variation of IDS, VTH and VDSAT for UPDATE=2

Each plot compares IDS, VTH, VDSAT, GM, GDS and GMBS as a function of vsb for UPDATE=1.

**Saturation Voltage, vsat**

To get the right value for vsat, two trial values of vsat corresponding to $\gamma_i$ and $\gamma_b$ are calculated:

$$vsat_1 = \frac{vgs - vbi_1}{\eta} + \frac{1}{2} \left( \frac{\gamma_i}{\eta} \right)^2 \left\{ 1 - \left[ 1 + \left( \frac{2}{\gamma_i} \right)^2 \left( \frac{vgs - vbi_1}{\eta} + PHI + vsb \right) \right]^{1/2} \right\}$$

$$vsat_2 = \frac{vgs - vbi_2}{\eta} + \frac{1}{2} \left( \frac{\gamma_b}{\eta} \right)^2 \left\{ 1 - \left[ 1 + \left( \frac{2}{\gamma_b} \right)^2 \left( \frac{vgs - vbi_2}{\eta} + PHI + vsb \right) \right]^{1/2} \right\}$$

$vbi_1$ and $vbi_2$ are built in potentials corresponding to $\gamma_i$ and $\gamma_b$, respectively.

If $(vdsat_1 + vsb) \leq VBO$, then $vdsat = vdsat_1$

If $(vdsat_2 + vsb) > VBO$, then $vdsat = vdsat_2$
LEVEL 6 IDS Equations, UPDATE=1

There are three equations for ids depending upon the region of operation. The model derives these equations by integrating the bulk charge \((vgs - vth(v) - v)\) from the source to the drain.

For \(v_{sb} < VBO-v_{de}\), the model forms an entire gate depletion region in the implant layer.

\[
ids = \beta \cdot \left( vgs - v_{bi1} - \frac{\eta \cdot v_{de}}{2} \right) \cdot v_{de}^{-\frac{2}{3}} \gamma_i \cdot \left[ (PHI + v_{de} + v_{sb})^{3/2} - (PHI + v_{sb})^{3/2} \right]
\]

where \(v_{bi1}\) is the same as \(v_{bi}\) for \(v_{sb} \leq VBO\).

For \(v_{sb} \geq VBO\), the entire gate depletion region expands into the bulk area.

\[
ids = \beta \cdot \left( vgs - v_{bi2} - \frac{\eta \cdot v_{de}}{2} \right) \cdot v_{de}^{-\frac{2}{3}} \gamma_b \cdot \left[ (PHI + v_{de} + v_{sb})^{3/2} - (PHI + v_{sb})^{3/2} \right]
\]

where \(v_{bi2}\) is the same as \(v_{bi}\) for \(v_{sb} > VBO\).

\[
ids = \beta \cdot \left( vgs - v_{bi2} - \frac{\eta \cdot v_{de}}{2} \right) \cdot v_{de}^{-\frac{2}{3}} \gamma_i \cdot \left[ (VBO + PHI)^{3/2} - (vsb + PHI)^{3/2} \right]
\]

\[
\quad + (\gamma_i - \gamma_b) \cdot (VBO + PHI)^{1/2} \cdot \left( VBO - vsb \right)
\]

For \(VBO-v_{de} < v_{sb} < VBO\), the source side gate depletion region is in the implant layer, but the drain side gate depletion region is expanded into the bulk area.
Alternate DC Model, (ISPICE model)

If model parameter KU>1, this model is invoked. Then, the model computes vfu and vfa scale factors to scale both the vds voltage and the ids current. These scale factors are functions of ECRIT and vgs voltage. The vfa and vfu factors are defined as follows:

\[
vfu = 1 - \frac{KU}{(\alpha^2 + KU^2)^{1/2} + \alpha(KU - 1)}
\]

\[
vfa = KA \cdot vfu^{(2 \cdot MAL)}
\]

where:

\[
\alpha = \frac{ECRIT \cdot Leff}{vgs - vth}
\]

---

**Note:** vfu factor is always less than one.

---

The current ids is modified as follows:

**NU=1**

\[
ids = vfu^{(2 \cdot MBL)} \cdot ids
\]

For NU=0, the factor \( vfu^{(2 \cdot MBL)} \) is set to one.

The current ids is a function of effective drain to source voltage, vde, which is determined as:

\[
vde = \min(vds/vfa, vsat)
\]

and:

\[
vdsat = vfa \cdot vsat
\]
This alternate model is generally coupled with the mobility normal field equations (MOB=3) and the channel length modulation drain field equation (CLM=3). The vde value used in the mobility equations is:

\[
\begin{align*}
de &= \min(vds, vfa \cdot vsat), \text{UPDATE}=0 \\
ds &= \min(vds/vfa, vsat), \text{UPDATE}=1,2
\end{align*}
\]

**Subthreshold Current, ids**

This region of operation is characterized by the choice of two different equations, selected through the model parameter WIC (Weak Inversion Choice). WIC can be designated as follows:

- **WIC=0** No weak inversion (default)
- **WIC=1** ASPEC-style weak inversion
- **WIC=2** Enhanced HSPICE-style weak inversion

In addition to WIC, set the parameter NFS. NFS represents the number of fast states per centimeter squared. Reasonable values for NFS range from 1e10 to 1e12.

**WIC=0**, no weak inversion.

**WIC=1**, the threshold voltage \(vth\) is increased by the term \(fast\).

\[
von = vth + fast
\]

where:

\[
fast = vt \cdot \left[ 1 + \frac{q \cdot NFS}{COX} + \frac{\gamma}{2 \cdot (vsb + PHI)^{1/2}} \right]
\]

and \(vt\) is the thermal voltage.
The current $ids$ for $vgs < von$ is given by:

$$ids = ids(von, vde, vsb) \cdot e^{\frac{vgs - von}{fast}}$$

if $vgs < von$, then

$$ids = ids(vge, vde, vsb)$$

**Note:** The modified threshold voltage ($von$) is not used for strong inversion conditions.

**WIC=2**

The subthreshold region is limited between cutoff and strong inversion regions. Although it appears that, if the gate voltage is less than $vth - PHI$, there can be no weak inversion conduction, there still can be diffusion conduction from the drain-to-bulk rather than drain-to-source.

$$von = vth + fast$$

where:

$$fast = vt \cdot \left[ 1 + \frac{q \cdot NFS}{COX} + \frac{\gamma}{2 \cdot (vsb + PHI)^{1/2}} \right]$$

**Cutoff Region, $vgs \leq vth - PHI$**

$$ids = 0$$

**Weak Inversion, $vth - PHI < vgs \leq von$**

$$ids = ids(von, vde, vsb) \cdot \left( 1 - \frac{von - vgs}{fast + PHI} \right)^{WEX}$$
**Strong Inversion, \(v_{gs} > v_{on}\)**

\[ ids = ids(v_{gs}, v_{de}, v_{sb}) \]

---

**Note:** The modified threshold voltage \((v_{on})\) is not used in strong inversion conditions.

If \(WIC=3\), the subthreshold current is calculated differently. In this case, the \(ids\) current is:

\[ ids = ids(v_{gs}, v_{de}, v_{sb}) + i_{sub}(N0_{eff}, N_{Deff}, v_{gs}, v_{ds}) \]

The \(N0_{eff}\) and \(N_{Deff}\) are functions of effective device width and length.

**Effective Mobility, \(u_{eff}\)**

All mobility equations have the general form:

\[ u_{eff} = UO \cdot \text{factor} \]

- \(u_{eff}\) Effective mobility at analysis temperature.
- \(\text{factor}\) Mobility degradation factor, see the following sections. Default=1.0

Use model parameter \(MOB\) to select the mobility modulation equation used by Star-Hspice as follows:

<table>
<thead>
<tr>
<th>MOB</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No mobility reduction (default)</td>
</tr>
<tr>
<td>1</td>
<td>(G_m) equation</td>
</tr>
<tr>
<td>2</td>
<td>Frohman-Bentchkowski equation</td>
</tr>
<tr>
<td>3</td>
<td>Normal field equation</td>
</tr>
<tr>
<td>4</td>
<td>Universal field mobility reduction</td>
</tr>
<tr>
<td>5</td>
<td>Universal field mobility reduction with independent drain field</td>
</tr>
</tbody>
</table>
These equations are described in the following sections.

**MOB=0 Default, No Mobility**

factor = 1.0
No mobility reduction

**MOB=1 Gm Equation**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>1/V</td>
<td>0.0</td>
<td>Gate field mobility reduction</td>
</tr>
<tr>
<td>UTRA (F3)</td>
<td>factor</td>
<td>0.0</td>
<td>Source-drain mobility reduction factor</td>
</tr>
</tbody>
</table>

The MOB=1 equation is useful for transistors with constant source-to-bulk voltage, since the factor does not contain a vsb term. Use of this equation can result in over-estimation of mobility for small gate voltages and large back-bias such as depletion pull-ups.

\[
  \text{factor} = \frac{1}{1 + F1 \cdot (vgs - vb1 - F3 \cdot vde)}
\]

\[
  vde = \min(vds, vdsat)
\]

**Note:** If the alternate saturation model is used, vde is different for UPDATE=0 and UPDATE=1. See “Alternate DC Model, (ISPIECE model)” on page 21-72. Also, if VMAX>0, then vde=min (vds, vsat), and if VMAX is not specified, then vde=min (vds, vdsat).
MOB=2 Frohman-Bentchkowski Equation

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>V/cm</td>
<td>0.0</td>
<td>Critical gate-bulk electric field at which mobility reduction becomes significant</td>
</tr>
<tr>
<td>UEXP (F2)</td>
<td></td>
<td>0.0</td>
<td>Mobility exponent. Use 0.36 factor for n-channel and 0.15 for p-channel.</td>
</tr>
<tr>
<td>UTRA (F3)</td>
<td>factor</td>
<td>0.0</td>
<td>Source-drain mobility reduction factor</td>
</tr>
<tr>
<td>VMAX (VMX)</td>
<td>cm/s</td>
<td>0.0</td>
<td>Maximum drift velocity of carriers. Whether or not VMAX is set determines which calculation scheme is used for vdsat. Use zero to indicate an infinite value.</td>
</tr>
</tbody>
</table>

Mobility reduction equation (MOB=2$^3$) produces good results for high gate voltages and drain fields with constant back-bias. This equation is typically used for p-channel pull-ups and n-channel pull-downs. Specify a value for VMAX to cause the proper calculation scheme to be used for vdsat. MOB=2 corresponds to MSINC UN=2 and is the SPICE default.

\[
factor = \left( \frac{F1 \cdot \varepsilon_s i}{C_OX \cdot (vgs - vbi - F3 \cdot vde)} \right)^{F2}
\]

where vde is defined the same as for MOB=1 equation.
MOB=3 Normal Field Equation

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>1/V</td>
<td>0.0</td>
<td>Low-field mobility multiplier</td>
</tr>
<tr>
<td>F4</td>
<td></td>
<td>1.0</td>
<td>Mobility summing constant</td>
</tr>
<tr>
<td>UEXP (F2)</td>
<td>1/V</td>
<td>0.0</td>
<td>Mobility exponent</td>
</tr>
<tr>
<td>UTRA (F3)</td>
<td>1/V</td>
<td>0.0</td>
<td>High-field mobility multiplier</td>
</tr>
<tr>
<td>VF1</td>
<td>V</td>
<td>0.0</td>
<td>Low to high field mobility (voltage switch)</td>
</tr>
</tbody>
</table>

This equation is the same as MSINC UN=1.

\[(vgs - vth)^F2 \leq VF1,\]

\[factor = \frac{1}{F4 + F1 \cdot (vgs - vth)^F2}\]

If UPDATE=0, and (vgs-vth)^F2 > VF1,

\[factor = \frac{1}{F4 + F3 \cdot (vgs - vth)^F2}\]

If UPDATE=1, 2 and (vgs-vth)^F2 > VF1,

\[factor = \frac{1}{F4 + (F1 - F3) \cdot VF1 + F3 \cdot (vgs - vth)^F2}\]
MOB=4 and MOB=5 Universal Field Mobility Reduction

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECRIT</td>
<td>V/cm</td>
<td>0.0</td>
<td>Critical electric drain field for mobility reduction. Use zero to indicate an infinite value.</td>
</tr>
<tr>
<td>F1</td>
<td>V/cm</td>
<td>0.0</td>
<td>Source-drain mobility reduction field (typical value 1e4 to 5e8)</td>
</tr>
<tr>
<td>MOB</td>
<td></td>
<td>0.0</td>
<td>Mobility equation selector. Set MOB=4 for critical field equation, or set MOB=5 for critical field equation with independent drain field.</td>
</tr>
<tr>
<td>UEXP (F2)</td>
<td>1/V(^{1/2})</td>
<td>0.0</td>
<td>Bulk mobility reduction factor (typical value 0 to 0.5)</td>
</tr>
<tr>
<td>UTRA (F3)</td>
<td>V/cm</td>
<td>0.0</td>
<td>Critical electric drain field for mobility reduction</td>
</tr>
</tbody>
</table>

The MOB=4 equation is the same as the MSINC UN=3 equation. The MOB=5 equation is the same as MOB=4 except that F3 substitutes for ECRIT in the expression for \(vc\).

The MOB=5 equation provides a better fit for CMOS devices in the saturation region. Do not specify a value for VMAX since velocity saturation is handled in the mobility equation.

\[
\text{factor} = \frac{1}{1 + \frac{COX}{F1 \cdot \varepsilon_{ox}} \cdot (vgs - cth) + \frac{vde}{vc} + F2 \cdot (v sb + PHI)^{1/2}}
\]

If MOB=4,

\[vc = ECRIT \cdot Leff\]

If MOB=5,

\[vc = F3 \cdot Leff\]
Note: If you use the alternate saturation model, vde is different for UPDATE=0 and UPDATE=1, 2.

MOB=6, 7 Modified MOB=3

This mobility equation is the same as MOB=3, except the equation uses VTO instead of vth. When MOB=6 is used, the current ids also is modified as follows:

$$\frac{ids}{1 + F1 \cdot \left( vgs - vth - \frac{vde}{2} \right) + \frac{UTRA}{Leff} \cdot vde}$$

Channel Length Modulation

The basic MOSFET current equation for ids describes a parabola, where the peak corresponds to the drain-to-source saturation voltage (vdsat). Long-channel MOSFETs generally demonstrate ideal behavior. For vds voltages greater than vdsat, there is no increase in the ids current. As the channel length decreases, the current in the saturation region continues to increase. This increase in current is modeled as a decrease in the effective channel length. Except for CLM=5 and 6, the channel length modulation equations are only calculated when the device is in the saturation region. Star-Hspice provides several channel length modulation equations; all (except for CLM=5) modify the ids equation as follows:

$$\frac{ids}{1 - \frac{\Delta L}{Leff}}$$

\(\Delta L\) is the change in channel length due to MOSFET electric fields.

Use model parameter CLM to designate the channel length modulation equation Star-Hspice uses as follows:

- CLM = 0 No channel length modulation (default)
- CLM = 1 one-sided step depletion layer drain field equation
These equations and the associated model parameters are discussed in the following sections.

**CLM=0 No Channel Modulation - Default**

\[
\Delta L = 0
\]

This is the default channel length equation, representing no channel length modulation; it corresponds to MSINC GDS=0.0

**CLM=1 Step Depletion Equation**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KL</td>
<td></td>
<td>0.0</td>
<td>Empirical constant (saturation voltage)</td>
</tr>
<tr>
<td>LAMBDA (LAM, LA)</td>
<td>cm/V^{1/2}</td>
<td>1.137e-4</td>
<td>Channel length modulation ((\Delta s) calculated from NSUB unless specified)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default LAMBDA corresponds to default NSUB value</td>
</tr>
</tbody>
</table>

\[
\Delta L = LAMBDA \cdot (vds - vsat)^{1/2} \cdot \left(\frac{vdsat}{vsat}\right)^{KL}
\]

If not user-specified, LAMBDA is calculated as:
This is a one-sided step depletion region formulation by Grove: $\Delta L$ varies with the depletion layer width, which is a function of the difference between the effective saturation voltage ($v_{dsat}$) and the drain-to-source channel voltage ($v_{ds}$). This equation is typically used for long channels and high dopant concentrations. This corresponds to GDS=1 in MSINC.

**CLM=2 Electrostatic Fringing Field**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>0.2</td>
<td></td>
<td>First fringing field factor gate-drain</td>
</tr>
<tr>
<td>A2</td>
<td>0.6</td>
<td></td>
<td>Second fringing field factor gate-vdsat</td>
</tr>
</tbody>
</table>

$$\Delta L = \frac{\varepsilon_{si}}{COX} \cdot \frac{v_{ds} - v_{dsat}}{A1 \cdot (v_{ds} - v_{gs} + v_{bi}) + A2 \cdot (v_{gs} - v_{bi} - v_{dsat})}$$

The fringing field equation, or electrostatic channel length reduction, developed by Frohman-Bentchkowski, is most often used for modeling short-channel enhancement transistors. In MSINC, the equivalent equation is GDS=2.
CLM=3 Carrier Velocity Saturation

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KA</td>
<td></td>
<td>1.0</td>
<td>vds scaling factor for velocity saturation</td>
</tr>
<tr>
<td>KCL</td>
<td></td>
<td>1.0</td>
<td>Exponent for vsb scaling factor</td>
</tr>
<tr>
<td>KU</td>
<td></td>
<td>0.0</td>
<td>Velocity saturation switch. If KU ≤ 1, the standard velocity saturation equation is used.</td>
</tr>
<tr>
<td>LAMBDA</td>
<td>cm/V^1/2</td>
<td>1.137e-4</td>
<td>Channel length modulation. This parameter is calculated from NSUB if not specified. The default LAMBDA corresponds to the default NSUB value.</td>
</tr>
<tr>
<td>MAL</td>
<td></td>
<td>0.5</td>
<td>vds exponent for velocity saturation</td>
</tr>
<tr>
<td>MCL</td>
<td></td>
<td>1.0</td>
<td>Short channel exponent</td>
</tr>
</tbody>
</table>

\[ \Delta L = vfu^{(2 \cdot MCL)} \cdot LAMBDA \cdot \left( (vds - vfa \cdot vsat + KCL \cdot vsb + PHI)^{1/2} - (KCL \cdot vsb + PHI)^{1/2} \right) \]

This equation is an extension of the first depletion layer equation, CLM=1, and includes the effects of carrier velocity saturation and the source-to-bulk voltage (vsb) depletion layer width. It represents the basic ISPICE equation. See “Alternate DC Model, (ISPICE model)” on page 21-72 for definitions of vfa and vfu.
CLM=4, Wang’s Equation

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>m</td>
<td>0.2</td>
<td>Junction depth</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A1scaled = A1 * SCALM</td>
</tr>
<tr>
<td>DND</td>
<td>cm⁻³</td>
<td>1e20</td>
<td>Drain diffusion concentration</td>
</tr>
</tbody>
</table>

Linearly Graded Depletion Layer

\[
\Delta L = \left[ \frac{2.73e5 \cdot A1scaled}{DNB \cdot \ln \left( \frac{DND}{DNB} \right)} \right]^{1/3} \cdot \left[ (vds - vdsat + PHI)^{1/3} - PHI^{1/3} \right]
\]

Wang’s equation allows the inclusion of junction characteristics in the calculation of channel length modulation. The equation assumes that the junction approximated a linearly-graded junction and provides a value of 0.33 for the exponent. This equation is similar to MSINC GDS=3.

CLM=5, Star-Hspice Channel Length Modulation

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAMBDA</td>
<td>amp/V²</td>
<td>0</td>
<td>Constant coefficient</td>
</tr>
<tr>
<td>VGLAM</td>
<td>1/V</td>
<td>0</td>
<td>Constant coefficient</td>
</tr>
</tbody>
</table>

When CLM=5, the current ids is increased by idssat, given as:

\[
idssat = \frac{weff}{Leff} \cdot LAMBDA \cdot vds \cdot (vgs - vth) \cdot \left[ 1 + VGLAM \cdot (vgs - vth) \right]
\]

\[
ids = ids + idssat
\]
Note: The equation adds the idssat term to ids in all regions of operation. Also, \( \text{LAMBDA} \) is a function of temperature.

**CLM=6, Star-Hspice \( \Delta L \) Equation**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAMBDA</td>
<td>1/( V^{KL} )</td>
<td>0</td>
<td>vds coefficient</td>
</tr>
<tr>
<td>LAM1</td>
<td>1/m</td>
<td>0</td>
<td>Channel length coefficient</td>
</tr>
<tr>
<td>KL</td>
<td></td>
<td>0</td>
<td>vds exponent</td>
</tr>
<tr>
<td>VGLAM</td>
<td>1/V</td>
<td>0</td>
<td>Gate drive coefficient</td>
</tr>
</tbody>
</table>

Unlike the other CLMs, this equation calculates the channel length modulation (\( \Delta L \)) in all regions of operations and uses it to modify current ids.

\[
\Delta L = \frac{L_{eff} \cdot LAMBDA \cdot vds^{KL} \cdot \left[1 + VGLAM \cdot (vgs - vth)\right]}{1 + LAM1 \cdot L_{eff}}
\]

and:

\[
ids = \frac{ids}{1 - \frac{\Delta L}{L_{eff}}}
\]

Note: \( \text{LAMBDA} \) is a function of temperature.
ASPEC Compatibility

Make MOSFET models compatible with ASPEC by specifying ASPEC=1 in the .OPTION statement and LEVEL=6 in the associated MOSFET model statement.

If you assign the element parameters without keynames, you must use the parameter sequence given in the general format. Star-Hspice assigns parameters in the order they are listed in the element statement. Errors occur if parameter names are also element keynames.

When Option ASPEC is in effect, a number of program variations occur. The MOSFET model parameter LEVEL is set to 6.

*Note:* Setting LEVEL=6 in the model does not invoke ASPEC.

ASPEC sets the following options:
- MOSFET Option: WL = 1
- General Options: SCALE = 1e-6, SCALM = 1e-6

Since the ASPEC option sets the SCALE and SCALM options, it effectively changes the default units of any parameters affected by these options; use parameter values consistent with these scaling factors.

ASPEC sets the following model parameter defaults:
- LEVEL = 6
- ACM = 1
- CJ = 0.0
- IS = 0.0
- NSUB = 1e15
**Note:** NSUB is not be calculated from GAMMA, if UPDATE=1 or 2.

<table>
<thead>
<tr>
<th>PHI</th>
<th>= 1 · $\Phi_f$ (the Fermi potential)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLEV</td>
<td>= 1</td>
</tr>
<tr>
<td>TLEVC</td>
<td>= 1</td>
</tr>
</tbody>
</table>

TLEV (TLEVC in turn, selects the ASPEC method of temperature update for the parameters CJ, CJSW, PB, PHP, VTO, and PHI.

**Note:** If PHI is entered explicitly, however, it is not updated for temperature. SCALM does not effect the scaling of parameters for the ASPEC mode. If SCALM is specified when using ASPEC, Star-Hspice generates an error stating that SCALM is ignored.
LEVEL 7 IDS Model

The LEVEL 7 model is the same as the LEVEL 6 model except for the value of PHI.

If PHI is specified, then

For LEVEL=6,

\[ \Phi_s = \frac{PHI}{2}, \]

where \( \Phi_s \) is the surface potential.

For LEVEL=7,

\[ \Phi_s = PHI \]

To transform a LEVEL 7 equation to LEVEL 6, make the following substitution:

\[ PHI \rightarrow 2 \cdot PHI \]

To transform a LEVEL 6 model into a LEVEL 7 model, make the following substitution:

\[ PHI(\text{Level 7}) = PHI(\text{Level 6})/2 \]
LEVEL 8 IDS Model

The LEVEL 8 model, derived from research at Intersil and General Electric, is an enhanced version of the LEVEL 2 ids equation. LEVEL 2 differs from LEVEL 8 in the following areas: the effective substrate doping, threshold voltage, effective mobility, channel length modulation, and subthreshold current.

LEVEL 8 Model Parameters

This section lists the LEVEL 8 model parameters.

Basic DC Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>IDS equation selector. Use LEVEL 8 for the advanced model using finite differences.</td>
</tr>
<tr>
<td>COX</td>
<td>F/m^2</td>
<td>3.45314e-4</td>
<td>Oxide capacitance per unit gate area. This parameter is calculated from TOX if not specified.</td>
</tr>
<tr>
<td>ECRIT (ESAT)</td>
<td>V/cm</td>
<td>0.0</td>
<td>Critical electric field for carrier velocity saturation, from Grove:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>electrons 6e4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>holes 2.4e4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Use zero to indicate an infinite value.</td>
</tr>
<tr>
<td>SNVB</td>
<td>1/ (V⋅cm^3)</td>
<td>0.0</td>
<td>Slope of doping concentration versus vsb (element parameter). (Multiplied by 1e6)</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>1e-7</td>
<td>Oxide thickness</td>
</tr>
<tr>
<td>VMAX (VMX, VSAT)</td>
<td>m/s</td>
<td>0.0</td>
<td>Maximum drift velocity of carriers. Use zero to indicate an infinite value.</td>
</tr>
</tbody>
</table>
# Effective Channel Width and Length Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEL</td>
<td>m</td>
<td>0.0</td>
<td>Channel length reduction on each side. DEL is applicable in most MOSFET models. An exception is the BSIM (LEVEL 13) model, where DEL is not present. &lt;br&gt;&lt;br&gt;DELScaled = DEL \cdot SCALM</td>
</tr>
<tr>
<td>LD (DLAT, LDTD)</td>
<td>m</td>
<td></td>
<td>Lateral diffusion into channel from source and drain diffusion. If LD and XJ are unspecified, LD default=0.0. &lt;br&gt;&lt;br&gt;When LD is unspecified, but XJ is specified, LD default=0.75 \cdot XJ. LDScaled = LD \cdot SCALM.</td>
</tr>
<tr>
<td>LDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as LD, but if LDAC is included in the .MODEL statement, it replaces LD in the Leff calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>WD</td>
<td>m</td>
<td>0.0</td>
<td>Lateral diffusion into channel from bulk along width &lt;br&gt;&lt;br&gt;WDScaled = WD \cdot SCALM</td>
</tr>
<tr>
<td>WDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as WD, but if WDAC is included in the .MODEL statement, it replaces WD in the Weff calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>LMLT</td>
<td></td>
<td>1.0</td>
<td>Length shrink factor</td>
</tr>
<tr>
<td>LREF</td>
<td>m</td>
<td>0.0</td>
<td>Channel length reference &lt;br&gt;&lt;br&gt;LREFscaled = LREF \cdot SCALM</td>
</tr>
<tr>
<td>WMLT</td>
<td></td>
<td>1.0</td>
<td>Diffusion layer and width shrink factor</td>
</tr>
</tbody>
</table>
### Threshold Voltage Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAV</td>
<td></td>
<td>0.0</td>
<td>Thermal voltage multiplier for the weak inversion equation</td>
</tr>
<tr>
<td>DELTA</td>
<td></td>
<td>0.0</td>
<td>Narrow width factor for adjusting threshold</td>
</tr>
<tr>
<td>ETA</td>
<td></td>
<td>0.0</td>
<td>Drain-induced barrier lowering (DIBL) effect coefficient for threshold voltage</td>
</tr>
<tr>
<td>GAMMA</td>
<td>$\sqrt[1/2]{}$</td>
<td></td>
<td>Body effect factor. This parameter is calculated from NSUB if not specified (see “Common Threshold Voltage Parameters” on page 20-50).</td>
</tr>
<tr>
<td>LND</td>
<td>(\mu m/V)</td>
<td>0.0</td>
<td>ND length sensitivity</td>
</tr>
<tr>
<td>LN0</td>
<td>(\mu m)</td>
<td>0.0</td>
<td>N0 length sensitivity</td>
</tr>
<tr>
<td>ND</td>
<td>(1/V)</td>
<td>0.0</td>
<td>Drain subthreshold factor (typical value=1)</td>
</tr>
<tr>
<td>N0</td>
<td></td>
<td>0.0</td>
<td>Gate subthreshold factor (typical value=1)</td>
</tr>
</tbody>
</table>
## Mobility Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOB</td>
<td></td>
<td>6.0</td>
<td>Mobility equation selector (can be set to 2, 3, 6, or 7 in LEVEL 8)</td>
</tr>
<tr>
<td>UCRIT</td>
<td>V/cm</td>
<td>1e4</td>
<td>MOB=6, UEXP&gt;0 Critical field for mobility degradation, UEXP operates as a switch.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MOB=6, UEXP≤0 Critical field for mobility degradation. Typical value is 0.01 V⁻¹.</td>
</tr>
<tr>
<td>UEXP (F2)</td>
<td></td>
<td>0.0</td>
<td>Critical field exponent in mobility degradation</td>
</tr>
<tr>
<td>UTRA</td>
<td>m/V</td>
<td>0.0</td>
<td>Transverse field coefficient (mobility)</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 1-40

LEVEL 8 IDS Model

Channel Length Modulation Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UO (UB, UBO)</td>
<td>cm²/V·s</td>
<td>600 (N)</td>
<td>Low field bulk mobility. This parameter is calculated from KP (BETA) if KP (BETA) is input.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250 (P)</td>
<td></td>
</tr>
</tbody>
</table>

IDS Equations

LEVEL 8 ids equations are the same as the LEVEL 2 model. These equations are repeated here for convenience.

Cutoff Region, vgs≤vth

\[ ids = 0 \quad (\text{See subthreshold current}) \]

On Region, vgs>vth

\[ ids = \beta \cdot \left( vgs - vb1 - \frac{n \cdot vde}{2} \right) \cdot vde \cdot \frac{2}{3} \cdot \gamma \cdot [\Phi + vde + vsb]^{3/2} - (\Phi + vsb)^{3/2} \]
where:
\[ vde = \min(vds, vdsat) \]
\[ \eta = 1 + DELTA \cdot \frac{\pi \cdot \varepsilon_{si}}{4 \cdot COX \cdot W_{eff}} \]
\[ \beta = KP \cdot \frac{W_{eff}}{L_{eff}} \]

**Effective Channel Length and Width**

The model calculates effective channel length and width from the drawn length and width as follows:

\[ L_{eff} = L_{scaled} \cdot LMLT + X_{scaled} - 2 \cdot (LD_{scaled} + DEL_{scaled}) \]
\[ W_{eff} = M \cdot (W_{scaled} \cdot WMLT + X_{Wscaled} - 2 \cdot WD_{scaled}) \]
\[ L_{REFeff} = L_{REFscaled} \cdot LMLT + X_{scaled} - 2 \cdot (LD_{scaled} + DEL_{scaled}) \]
\[ W_{REFeff} = M \cdot (W_{REFscaled} \cdot WMLT + X_{Wscaled} - 2 \cdot WD_{scaled}) \]

**Effective Substrate Doping, nsub**

Specify the model parameter SNVB to vary substrate doping concentration linearly as a function of vsb.

\[ n_{sub} = NSUB + SNVB \cdot vsb \]

The \( \gamma \), \( \Phi \), and \( xd \) parameters are computed using the above equation for nsub.

\[ \gamma = \frac{\sqrt{2} \cdot \varepsilon_{si} \cdot q \cdot n_{sub}}{COX} \]
If SNVB is zero, then $\gamma = \text{GAMMA}$. The $\gamma$ value is adjusted for short-channel effect the same way as the LEVEL 2 model. Also, $\Phi$ is calculated using NSUB.

**Threshold Voltage, $v_{th}$**

Specify ETA to include the threshold voltage reduction due to potential barrier lowering effect.

$$v_{bi} = V_{TO} - \gamma \cdot \sqrt{\Phi} - \frac{8.14 \cdot 22 \cdot \text{ETA}}{COX \cdot L_{eff}^3} \cdot v_{ds} + (\eta - 1) \cdot (v_{sb} + \Phi)$$

$$v_{th} = v_{bi} + \gamma \cdot \sqrt{v_{sb} + \Phi}$$

The $\gamma$ is modified for short-channel effect, the same as in the LEVEL 2 model, to get effective $\gamma$.

**Saturation Voltage $v_{dsat}$**

The saturation voltage $v_{dsat}$ is computed the same as in the LEVEL 2 model. The carrier velocity effect is included only when ECRIT is greater than zero.

$ECRIT > 0$,

$$v_{dsat} = v_{sat} + v_c - \sqrt{v_{sat}^2 + v_c^2}$$

where:

$$v_c = ECRIT \cdot L_{eff}$$
ECRIT \leq 0 \text{ or MOB=7},

\[ v_{dsat} = v_{sat} \]

\( v_{sat} \) is computed as in the LEVEL=2 model (See page 21-14).

**Effective Mobility, \( u_{eff} \)**

The mobility equation selector MOB controls the mobility reduction equations. In the LEVEL 8 model, set MOB to 2, 3, 6, or 7. Default=6.

**MOB=2 Mobility Reduction**

\[
 u_{eff} = UO \cdot \left[ \frac{\varepsilon_{se} \cdot UCRIT}{COX \cdot (v_{gs} - v_{th} - UTRA \cdot v_{de})} \right]^{UEXP}
\]

**MOB=3 Mobility Reduction**

\[
 u_{eff} = \frac{UO}{1 + \frac{2.1e^{-8} \cdot (v_{gs} + v_{th} + egfet - \Phi)}{6 \cdot TOX}}
\]

where \( egfet \) is the silicon energy gap at the analysis temperature.

\[
 egfet = 1.16 - \frac{7.02e^{-4} \cdot t^2}{t + 1108}
\]

where \( t \) is the temperature in degrees Kelvin.

**If VMAX>1,**

\[
 u_{eff} = \frac{u_{eff}}{1 + \frac{u_{eff}}{VMAX \cdot L_{eff}} \cdot v_{de}}
\]
MOB=6 Mobility Reduction

For $\text{UEXP}>0$,  

If 

$\left( vgs - vth \right) > \left( \frac{\varepsilon_{si} \cdot UCRIT}{COX} \right)^{UEXP}$

then 

$u_{eff} = \frac{UO \cdot \left[ \frac{\varepsilon_{si} \cdot UCRIT}{COX \cdot \left( vgs - vth \right)} \right]^{UEXP}}{1 + \frac{UTRA}{L_{eff}} \cdot vde}$

otherwise,  

$u_{eff} = \frac{UO}{1 + \frac{UTRA}{L_{eff}} \cdot vde}$

For $\text{UEXP}=0$

$u_{eff} = \frac{UO}{\left[ 1 + UCRIT \cdot \left( vgs - vth \right) \right] \cdot \left( 1 + \frac{UTRA}{L_{eff}} \cdot vde \right)}$

$\text{UCRIT for UEXP}=0 \text{ has a dimension of (1/V).}$

MOB=7 Mobility Reduction

$u_{eff} = \frac{UO}{1 + UTRA \cdot \left( vgs - vbi - \eta \cdot \frac{vde}{2} + \text{body} \right) \cdot vde}$

where:

$\text{body} = \frac{2}{3} \cdot \gamma \cdot \left[ \left( vde + vsb + \Phi \right)^{3/2} - \left( vsb + \Phi \right)^{3/2} \right]$
**Channel Length Modulation**

The equation selector CLM controls the channel length modulation equations. In the LEVEL 8 model, set CLM to 6, 7, and 8. Default=7.

**CLM=6 SPICE Channel Length Modulation**

If LAMBDA=0,

\[
\lambda = \frac{xd}{L_{eff} \cdot vds} \cdot \sqrt{\frac{vds - vdsat}{4}} + \sqrt{1 + \left(\frac{vds - vdsat}{4}\right)^2}
\]

otherwise,

\[
\lambda = LAMBDA
\]

then,

\[
\Delta L = \frac{\lambda \cdot L_{eff} \cdot vds}{1 + LAM1 \cdot L_{eff}}
\]

**Note:** The LEVEL 2 model has no LAM1 term.

The current is modified for channel length modulation effect in entire regions as:

\[
ids = \frac{ids}{1 - \frac{\Delta L}{L_{eff}}}
\]

**CLM=7 Intersil Channel Length Modulation**

The \( \Delta L \) is only computed for the saturation region.
vds > vdsat
\[ \Delta L = \frac{\text{LAMBDA} \cdot L_{\text{eff}}}{1 + \text{LAM1} \cdot L_{\text{eff}}} \cdot (vds - vdsat) \]

and:
\[ ids = \frac{ids}{L - \frac{\Delta L}{L_{\text{eff}}}} \]

**CLM=8**
The \( \Delta L \) is only computed for the saturation region.

\[ vds > vdsat \]
\[ \Delta L = \frac{L_{\text{eff}}}{1 + \frac{(1 + \text{LAM1} \cdot L_{\text{eff}}) \cdot (1 + vde)^{A1}}{\text{LAMBDA} \cdot (vds - vde)}} \]

and:
\[ ids = \frac{ids}{1 - \frac{\Delta L}{L_{\text{eff}}}} \]

**Subthreshold Current Ids**
The LEVEL 8 model has different subthreshold current equations, depending on the value of model parameter CAV.

Define:
\[ fast = v_{t} \cdot \left[ \eta + \frac{q \cdot NFS}{COX} + \frac{\gamma}{2 \cdot (v_{sb} + \Phi)^{1/2}} + \frac{\varepsilon_{si} \cdot q \cdot SNVB \cdot \sqrt{v_{sb} + \Phi}}{\gamma \cdot COX^2} \right] \]
CAV≠0

\[ von = vth + CAV \cdot fast \]

Subthreshold Region, \( vgs < von \)

If \( vgs > vth \)

\[ ids = ids(von, vde, vsb) \cdot e^{(-1 - \frac{CAV}{2}) \cdot e^{\left(\frac{1}{fast} \cdot \frac{(CAV - 2) \cdot (vgs - vth)}{2 \cdot CAV^2 \cdot fast^2}\right) (vgs - vth)}} \]

If \( vgs \leq vth \)

\[ ids = ids(von, vde, vsb) \cdot \left(1 - \frac{CAV}{2}\right) \cdot e^{\left(\frac{vgs - vth}{fast}\right)} \]

CAV=0

If CLM=8,

\[ von = vth + 3 \cdot fast \]

otherwise,

\[ von = vth + 2 \cdot fast \]

Subthreshold Region, \( vgs < von \)

\[ ids = ids(von, vde, vsb) \cdot e^{\left(\frac{vgs - von}{fast}\right)} \]

If WIC=3, the subthreshold current is calculated differently. In this case the \( ids \) current is:

\[ ids = ids(vgs, vde, vsb) + isub(N0eff, NDeff, vgs, vds) \]

\( N0eff \) and \( NDeff \) are functions of effective device width and length.
LEVEL 13 BSIM Model

The Star-Hspice LEVEL 13 MOSFET model is an adaptation of BSIM (Berkeley Short Channel IGFET) from SPICE 2G.6 (SPICE). The model is formulated on the device physics of small-geometry MOS transistors. To invoke the subthreshold region, set the model parameter N0 (low field weak inversion gate drive coefficient) to less than 200. The Star-Hspice wire model (from resistor element), which is compatible with SPICE BSIM interconnect model for polysilicon and metal layers, simulates resistors and capacitors generated with interconnect. The Star-Hspice capacitor model (from capacitor element) simulates capacitors generated with interconnect. The Star-Hspice MOSFET diffusion model is compatible with the SPICE BSIM diffusion model.

Two different types of formats are available for specifying the BSIM model parameters. Enter the model parameters as a sequence of numbers similar to SPICE, or set them using model parameter assignments. When converting from SPICE to Star-Hspice, the keyletter for the MOSFET device is S for SPICE BSIM and M for Star-Hspice. (Refer to the example of Star-Hspice BSIM model circuit file at the end of this section.) Some model parameter names have been modified due to the SPICE BSIM model installation in Star-Hspice.

BSIM Model Features

- Vertical field dependence of carrier mobility
- Carrier velocity saturation
- Drain-induced barrier lowering
- Depletion charge sharing by source and drain
- Non-uniform doping profile for ion-implanted devices
- Channel length modulation
- Subthreshold conduction
- Geometric dependence of electrical parameters
LEVEL 13 Model Parameters

*Note:* When reading parameter names, be aware of the difference in appearance between the upper case letter O, the lower case letter o, and the number zero (0).

For reference purposes only, the default values below are obtained from a medium size n-channel MOSFET device.

All LEVEL 13 parameters should be specified using NMOS conventions, even for PMOS (for example, ETA0=0.02, not ETA0=-0.02).

**Transistor Process Parameters**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1</td>
<td>MOSFET model level selector, set to 13 for the Star-Hspice BSIM model</td>
</tr>
<tr>
<td>CGBOM, (CGBO)</td>
<td>F/m</td>
<td>2.0e-10</td>
<td>Gate-to-bulk parasitic capacitance (F/m of length)</td>
</tr>
<tr>
<td>CGDOM, (CGDO)</td>
<td>F/m</td>
<td>1.5e-9</td>
<td>Gate-to-drain parasitic capacitance (F/m of width)</td>
</tr>
<tr>
<td>CGSOM, (CGSO)</td>
<td>F/m</td>
<td>1.5e-9</td>
<td>Gate-to-source parasitic capacitance (F/m of width)</td>
</tr>
<tr>
<td>DL0</td>
<td>µm</td>
<td>0.0</td>
<td>Difference between drawn poly and electrical</td>
</tr>
<tr>
<td>DW0</td>
<td>µm</td>
<td>0.0</td>
<td>Difference between drawn diffusion and electrical</td>
</tr>
<tr>
<td>DUM1</td>
<td></td>
<td>0.0</td>
<td>Dummy (not used)</td>
</tr>
<tr>
<td>DUM2</td>
<td></td>
<td>0.0</td>
<td>Dummy (not used)</td>
</tr>
<tr>
<td>ETA0</td>
<td></td>
<td>0.0</td>
<td>Linear vds threshold coefficient</td>
</tr>
<tr>
<td>LETA</td>
<td>mm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>WETA</td>
<td>µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>K1</td>
<td>V_{1/2}</td>
<td>0.5</td>
<td>Root-vsb threshold coefficient</td>
</tr>
<tr>
<td>LK1</td>
<td>V_{1/2}·µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WK1</td>
<td>V_{1/2}·µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>K2</td>
<td></td>
<td>0.0</td>
<td>Linear vsb threshold coefficient</td>
</tr>
<tr>
<td>LK2</td>
<td>µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WK2</td>
<td>µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>MUS</td>
<td>cm^2/(V·s)</td>
<td>600</td>
<td>High drain field mobility</td>
</tr>
<tr>
<td>LMS (LMUS)</td>
<td>µm·cm^2/(V·s)</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WMS (WMUS)</td>
<td>µm·cm^2/(V·s)</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>MUZ</td>
<td>cm^2/(V·s)</td>
<td>600</td>
<td>Low drain field first order mobility</td>
</tr>
<tr>
<td>LMUZ</td>
<td>µm·cm^2/(V·s)</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WMUZ</td>
<td>µm·cm^2/(V·s)</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>N0</td>
<td></td>
<td>0.5</td>
<td>Low field weak inversion gate drive coefficient (a value of 200 for N0 disables weak inversion calculation)</td>
</tr>
<tr>
<td>LN0</td>
<td></td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WN0</td>
<td></td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>NB0</td>
<td></td>
<td>0.0</td>
<td>Vsb reduction to low field weak inversion gate drive coefficient</td>
</tr>
<tr>
<td>LNB</td>
<td></td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>WNB</td>
<td></td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>ND0</td>
<td></td>
<td>0.0</td>
<td>Vds reduction to low field weak inversion gate drive coefficient</td>
</tr>
<tr>
<td>LND</td>
<td></td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WND</td>
<td></td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>PHI0</td>
<td>V</td>
<td>0.7</td>
<td>Two times the Fermi potential</td>
</tr>
<tr>
<td>LPHI</td>
<td>V.µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WPHI</td>
<td>V.µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>TREF</td>
<td>°C</td>
<td>25.0</td>
<td>Reference temperature of model (local override of TNOM)</td>
</tr>
<tr>
<td>TOXM, (TOX)</td>
<td>µm, (m)</td>
<td>0.02</td>
<td>Gate oxide thickness (TOXM or TOX &gt; 1 is interpreted as Angstroms)</td>
</tr>
<tr>
<td>U00</td>
<td>1/V</td>
<td>0.0</td>
<td>Gate field mobility reduction factor</td>
</tr>
<tr>
<td>LU0</td>
<td>µm/V</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WU0</td>
<td>µm/V</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>U1</td>
<td>µm/V</td>
<td>0.0</td>
<td>Drain field mobility reduction factor</td>
</tr>
<tr>
<td>LU1</td>
<td>µm²/V</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WU1</td>
<td>µm²/V</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>VDDDM</td>
<td>V</td>
<td>50</td>
<td>Critical voltage for high drain field mobility reduction</td>
</tr>
<tr>
<td>VFB0 (VFB)</td>
<td>V</td>
<td>-0.3</td>
<td>Flatband voltage</td>
</tr>
<tr>
<td>LVFB</td>
<td>V.µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WVFB</td>
<td>V.µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X2E</td>
<td>1/V</td>
<td>0.0</td>
<td>Vsb correction to linear vds threshold coefficient</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>LX2E</td>
<td>μm/V</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX2E</td>
<td>μm/V</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X2M (X2MZ)</td>
<td>cm²/(V²·s)</td>
<td>0.0</td>
<td>Vsb correction to low field first order mobility</td>
</tr>
<tr>
<td>LX2M (LX2MZ)</td>
<td>μm·cm²/(V²·s)</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX2M (WX2MZ)</td>
<td>μm·cm²/(V²·s)</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X2MS</td>
<td>cm²/(V²·s)</td>
<td>0.0</td>
<td>Vbs reduction to high drain field mobility</td>
</tr>
<tr>
<td>LX2MS</td>
<td>μm·cm²/(V²·s)</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX2MS</td>
<td>μm·cm²/(V²·s)</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X2U0</td>
<td>1/V²</td>
<td>0.0</td>
<td>Vsb reduction to GATE field mobility reduction factor</td>
</tr>
<tr>
<td>LX2U0</td>
<td>μm/V²</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX2U0</td>
<td>μm/V²</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X2U1</td>
<td>μm/V²</td>
<td>0.0</td>
<td>Vsb reduction to DRAIN field mobility reduction factor</td>
</tr>
<tr>
<td>LX2U1</td>
<td>μm²/V²</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX2U1</td>
<td>μm²/V²</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X3E</td>
<td>1/V</td>
<td>0.0</td>
<td>Vds correction to linear vds threshold coefficient</td>
</tr>
<tr>
<td>LX3E</td>
<td>μm/V</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX3E</td>
<td>μm/V</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
</tbody>
</table>
### Diffusion Layer Process Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJW, (CJSW)</td>
<td>F/m</td>
<td>0.0</td>
<td>Zero-bias bulk junction sidewall capacitance</td>
</tr>
<tr>
<td>CJM, (CJ)</td>
<td>F/m²</td>
<td>4.5e-5</td>
<td>Zero-bias bulk junction bottom capacitance</td>
</tr>
<tr>
<td>DS</td>
<td>m</td>
<td>0.0</td>
<td>Average variation of size due to side etching or mask compensation (not used)</td>
</tr>
<tr>
<td>IJS, (JS)</td>
<td>A/m²</td>
<td>0</td>
<td>Bulk junction saturation current</td>
</tr>
<tr>
<td>JSW</td>
<td>A/m</td>
<td>0.0</td>
<td>Sidewall bulk junction saturation current</td>
</tr>
<tr>
<td>MJ0, (MJ)</td>
<td></td>
<td>0.5</td>
<td>Bulk junction bottom grading coefficient</td>
</tr>
<tr>
<td>MJW, (MJSW)</td>
<td></td>
<td>0.33</td>
<td>Bulk junction sidewall grading coefficient</td>
</tr>
<tr>
<td>PJ, (PB)</td>
<td>V</td>
<td>0.8</td>
<td>Bulk junction bottom potential</td>
</tr>
</tbody>
</table>
### Basic Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD (DLAT, LATD)</td>
<td>m</td>
<td></td>
<td>Lateral diffusion into channel from source and drain diffusion.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If LD and XJ are unspecified, then LD default=0.0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>When LD is unspecified but XJ is specified, LD is calculated from XJ. LD Default=0.75⋅XJ.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LDscaled = LD ⋅ SCALM</td>
</tr>
<tr>
<td>LDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as LD, but if LDAC is included in the .MODEL statement, it replaces LD in the Leff calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>LMLT</td>
<td>1.0</td>
<td></td>
<td>Length shrink factor</td>
</tr>
<tr>
<td>LREF</td>
<td>m</td>
<td>0.0</td>
<td>Channel length reference</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LREFscaled = LREF ⋅ SCALM</td>
</tr>
</tbody>
</table>

**Note:** The wire model includes poly and metal layer process parameters.
### Name (Alias) | Units | Default | Description
---|---|---|---
WD | m | 0.0 | Lateral diffusion into channel from bulk along width
  \[
  W_{D\text{scaled}} = W_D \cdot SCALM
  \]
WDAC | m | | This parameter is the same as WD, but if WDAC is included in the .MODEL statement, it replaces WD in the \( W_{eff} \) calculation for AC gate capacitance.
WMLT | 1.0 | | Diffusion layer and width shrink factor
XL (DL, LDEL) | m | 0.0 | Accounts for masking and etching effects
  \[
  X_{L\text{scaled}} = X_L \cdot SCALM
  \]
XW (DW, WDEL) | m | 0.0 | Accounts for masking and etching effects
  \[
  X_{W\text{scaled}} = X_W \cdot SCALM
  \]
WREF | m | 0.0 * | Reference channel width
  \[
  W_{REF\text{scaled}} = WREF \cdot SCALM
  \]

**Note:** *If \( W_{REF} \) and \( WREF \) are not defined in the model, they take a value of infinity. The default of 0.0 is for Star-Hspice only.*
Temperature Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEX</td>
<td></td>
<td>-1.5</td>
<td>Temperature exponent for MUZ and MUS mobility parameters</td>
</tr>
<tr>
<td>FEX</td>
<td></td>
<td>0.0</td>
<td>Temperature exponent for mobility reduction factor U1</td>
</tr>
<tr>
<td>TCV</td>
<td>V/K</td>
<td>0.0</td>
<td>Flat-band voltage temperature coefficient</td>
</tr>
<tr>
<td>TREF</td>
<td>°C</td>
<td>25</td>
<td>Temperature at which parameters are extracted. This parameter defaults to the option TNOM, which defaults to 25 °C.</td>
</tr>
</tbody>
</table>

Sensitivity Factors of Model Parameters

For transistors, denote the L (channel length) and W (channel width) sensitivity factors of a basic electrical parameter are denoted by adding the characters ‘L’ and ‘W’ at the start of the name. For example, VFB0 sensitivity factors are LVFB and WVFB. If A0 is a basic parameter, then LA and WA are the corresponding L and W sensitivity factors of this parameter. LA and WA cannot be scaled using option SCALM in Star-Hspice. The model uses the general formula below to obtain this parameter value.

\[ A = A_0 + LA \cdot \left( \frac{1}{L_{eff}} - \frac{1}{L_{REFeff}} \right) + WA \cdot \left( \frac{1}{W_{eff}} - \frac{1}{W_{REFeff}} \right) \]

LA and WA are specified in units of microns times the units of A0.

The left side of the equation represents the effective model parameter value after device size adjustment. All the effective model parameters are in lower case and start with the character “z”, followed by the parameter name.
Example

\[ V_{FB0} = -0.350v \]

\[ L_{VFB} = -0.1v\mu \]

\[ W_{VFB} = 0.08v \cdot \mu \]

\[ L_{eff} = 1 \cdot 10^{-6}m = 1\mu \]

\[ W_{eff} = 2 \cdot 10^{-6}m = 2\mu \]

\[ L_{REFeff} = 2 \cdot 10^{-6}m = 2\mu \]

\[ W_{REFeff} = 1 \cdot 10^{-5}m = 10\mu \]

\[ z_{vfb} = V_{FB0} + L_{VFB} \cdot \left( \frac{1}{L_{eff}} - \frac{1}{L_{REFeff}} \right) + W_{VFB} \cdot \left( \frac{1}{W_{eff}} - \frac{1}{W_{REFeff}} \right) \]

\[ z_{vfb} = -0.35v - 0.1v \cdot \mu \cdot \left( \frac{1}{1\mu} - \frac{1}{2\mu} \right) + 0.08v \cdot \mu \cdot \left( \frac{1}{2\mu} - \frac{1}{10\mu} \right) \]

\[ z_{vfb} = -0.35v - 0.05v + 0.032v \]

\[ z_{vfb} = -0.368v \]
.MODEL VERSION Changes to BSIM Models

The VERSION parameter to the .MODEL statement allows portability of LEVEL 13 BSIM and LEVEL 39 BSIM2 models between Star-Hspice versions. Using the VERSION parameter in a LEVEL 13 .MODEL statement results in the following changes to the BSIM model:

<table>
<thead>
<tr>
<th>Model Version</th>
<th>Effect of VERSION on BSIM model</th>
</tr>
</thead>
<tbody>
<tr>
<td>9007B</td>
<td>LEVEL 13 BSIM model introduced: no changes</td>
</tr>
<tr>
<td>9007D</td>
<td>Removes the K2 limit</td>
</tr>
<tr>
<td>92A</td>
<td>Changes the TOX parameter default from 1000 A to 200 A</td>
</tr>
<tr>
<td>92B</td>
<td>Adds the K2LIM parameter, which specifies the K2 limit</td>
</tr>
<tr>
<td>93A</td>
<td>Introduces gds constraints</td>
</tr>
<tr>
<td>93A.02</td>
<td>VERSION parameter introduced</td>
</tr>
<tr>
<td>95.1</td>
<td>Fixes nonprinting TREF and incorrect GMBS problems</td>
</tr>
<tr>
<td>96.1</td>
<td>Flatband voltage temperature adjustment has been changed</td>
</tr>
</tbody>
</table>
LEVEL 13 Equations

This section lists the LEVEL 13 model equations.

Effective Channel Length and Width

The effective channel length and width for LEVEL 13 is determined differently, depending on the specified model parameters.

If DL0 is specified then,

\[ L_{eff} = L_{scaled} \cdot L_{MLT} - DL0 \cdot 1e-6 \]

\[ L_{REFeff} = L_{REFscaled} \cdot L_{MLT} - DL0 \cdot 1e-6 \]

Otherwise, if XL or LD is specified,

\[ L_{eff} = L_{scaled} \cdot L_{MLT} + XL_{scaled} - 2 \cdot LD_{scaled} \]

\[ L_{REFeff} = L_{REFscaled} \cdot L_{MLT} + XL_{scaled} - 2 \cdot LD_{scaled} \]

If DW0 is specified, then

\[ W_{eff} = W_{scaled} \cdot W_{MLT} - DW0 \cdot 1e-6 \]

\[ W_{REFeff} = W_{REFscaled} \cdot W_{MLT} - DW0 \cdot 1e-6 \]

Otherwise, if XW or WD is specified, then

\[ W_{eff} = W_{scaled} \cdot W_{MLT} + XW_{scaled} - 2 \cdot WD_{scaled} \]

\[ W_{REFeff} = W_{REFscaled} \cdot W_{MLT} + XW_{scaled} - 2 \cdot WD_{scaled} \]
**IDS Equations**

The device characteristics are modeled by process-oriented model parameters, which are mapped into model parameters at a specific bias voltage. The ids equations are as follows:

**Cutoff Region, \( v_{gs} \leq v_{th} \)**

\[
ids = 0 \quad \text{(see subthreshold current)}
\]

**On Region, \( v_{gs} > v_{th} \)**

For \( v_{ds} < v_{dsat} \), triode region:

\[
ids = \frac{\beta}{1 + xu1 \cdot v_{ds}} \cdot \left( \frac{(v_{gs} - v_{th}) \cdot v_{ds} - \frac{body}{2} \cdot v_{ds^2}}{v_{gs} - v_{th}} \right)
\]

For \( v_{ds} \geq v_{dsat} \), saturation region:

\[
ids = \frac{\beta}{2 \cdot body \cdot arg} \cdot (v_{gs} - v_{th})^2
\]

where:

\[
\beta = u_{eff} \cdot COX \cdot \frac{W_{eff}}{L_{eff}}
\]

\[
u_{eff} = \frac{uo}{1 + xu0 \cdot (v_{gs} - v_{th})}
\]

\[
xu0 = zu0 - zx2uo \cdot vsb
\]

The carrier mobility, \( uo \), is calculated by quadratic interpolation through three data points.

\[
uo|_{v_{ds} = 0} = MUZ - zx2mz \cdot vsb
\]

\[
uo|_{v_{ds} = V_{DDM}} = zmus - zx2ms \cdot vsb
\]
and the sensitivity of $u_0$ to $v_{ds}$ at $v_{ds}=V_{DDM}$, which is $zx3ms$.

The “body” factor is calculated by:

$$body = 1 + \frac{g \cdot zk1}{2 \cdot (zphi + vsb)^{1/2}}$$

where:

$$g = 1 - \frac{1}{1.744 + 0.8364 \cdot (zphi + vsb)}$$

The “arg” term in saturation region current is calculated by:

$$arg = \frac{1}{2} \cdot \left[ 1 + vc + (1 + 2 \cdot vc)\right]^{1/2}$$

where:

$$vc = \frac{xu1 \cdot (vgs - vth)}{body}$$

and:

$$xu1 = zu1 - zx2u1 \cdot vsb + zx3u1 \cdot (vds - VDDM), \quad \text{UPDATE}=2$$

$$xu1 = \frac{zu1 - zx2u1 \cdot vsb + zx3u1 \cdot (vds - VDDM)}{Leff}, \quad \text{UPDATE}=0, 1$$

**Threshold Voltage**

The threshold voltage can be expressed as:

$$vth = zvf + zphi + gamma \cdot (zphi + vsb)^{1/2} - xeta \cdot vds$$

where

$$gamma = zk1 - zk2 \cdot (zphi + vsb)^{1/2}$$
and:
\[ xeta = \zeta - zx2e \cdot vsb + zx3e \cdot (vds - VDDM), \text{ UPDATE}=0,2 \]
\[ xeta = \zeta + zx2e \cdot (zphi + vsb) + zx3e \cdot (vds - VDDM), \text{ UPDATE}=1 \]

**Saturation Voltage (vdsat)**

The saturation voltage in the BSIM model is calculated as follows:

\[ vdsat = \frac{vgs - vth}{body \cdot \arg^{1/2}} \]

**Subthreshold Current ids**

The subthreshold current isub is calculated when zn0 is less than 200 as follows:

\[ isub = \frac{Ilim \cdot Iexp}{Ilim + Iexp} \]

where:

\[ Iexp = \beta_o \cdot vt^2 \cdot e^{vgs-vth} \cdot \left( 1 - e^{\frac{vds}{vt}} \right) \]

\[ Ilim = 4.5 \cdot \beta_o \cdot vt^2 \]

\[ \beta_o = uo \cdot COX \cdot \frac{Weff}{Leff} \]

and:

\[ xn = zn0 - zn \cdot vsb + zn \cdot vds \]

---

**Note:** The current isub also is added to the ids current in the strong inversion.
Resistors and Capacitors Generated with Interconnects

See the Star-Hspice wire model table (resistor element) for the model parameters used.

Resistances:
\[ r = RSH \cdot \frac{L_{eff}}{W_{eff}} \]

Capacitances:
\[ c = COX \cdot L_{eff} \cdot W_{eff} + 2 \cdot CAPSW \cdot (L_{eff} + W_{eff}) \]

Temperature Effect
\[ MUZ(t) = MUZ \cdot \left(\frac{t}{t_{nom}}\right)^{BEX} \quad \text{UPDATE}=0, 1 \]
\[ zmus(t) = zmus \cdot \left(\frac{t}{t_{nom}}\right)^{BEX} \quad \text{UPDATE}=0, 1 \]
\[ uo(t) = uo \left(\frac{t}{t_{nom}}\right)^{BEX} \quad \text{UPDATE}=2 \]
\[ xu1(t) = xu1 \cdot \left(\frac{t}{t_{nom}}\right)^{FEX} \]
\[ zvfb(t) = zvfb - \Delta t \cdot TCV \]

where:
\[ \Delta t = t - t_{nom} \]
Charge-Based Capacitance Model

The Star-Hspice LEVEL 13 capacitance model conserves charge and has nonreciprocal attributes. Using charge as the state variable guarantees charge conservation. You can get total stored charge in each of the gate, bulk, and channel regions by integrating the distributed charge densities/area of the active region.

The channel charge is partitioned into drain and source components in two physically significant methods by using the model parameter XPART: 40/60, or 0/100 in the saturation region, which smoothly changes to 50/50 in the triode region. XPART=0 selects 40/60 drain/source charge-partitioning in the saturation region, while XPART=1 and XPART=0.5 select 0/100 and 50/50 for drain/source charge-partitioning in the saturation region, respectively.

Define:

\[ v_{tho} = z_{vfb} + z_{phi} + z_{k1} \cdot (z_{phi} + v_{sb})^{1/2} \]

\[ cap = COX \cdot Leff \cdot Weff \]

\[ v_{pof} = \frac{v_{gs} - v_{tho}}{body} \]

\[ argx = \frac{body \cdot v_{ds}}{12 \cdot (v_{gs} - v_{tho} - 0.5 \cdot body \cdot v_{ds})} \]

If \((v_{gs} - v_{tho} - 0.5 \cdot body \cdot v_{ds}) \leq 1e-8\) then,

\[ argx = \frac{1}{6} \]

\[ argy = \frac{(v_{gs} - v_{tho})^2 - 0.75 \cdot body \cdot (v_{gs} - v_{tho}) \cdot v_{ds} + 0.15 \cdot body^2 \cdot v_{ds}^2}{6 \cdot (v_{gs} - v_{tho} - 0.5 \cdot body \cdot v_{ds})^3} \]
If \((vgs - vtho - 0.5 \cdot body \cdot vds) \leq 1e-8\) then,

\[
\text{argy} = \frac{4}{15}
\]

**Regions Charge Expressions**

**Accumulation Region, \(vgs \leq vtho, vgs \leq zvfb - vsb\)**

\[
Qg = \text{cap} \cdot (vgs - zvfb + vsb)
\]

\[
Qb = -qg
\]

\[
Qs = 0
\]

\[
Qd = 0
\]

**Subthreshold Region, \(vgs \leq vtho, vgs > zvfb - vsb\)**

\[
Qg = \frac{\text{cap} \cdot zk1}{2} \cdot \left\{ \left( zk1 \right) \left[ \left( zk1 \right)^2 + 4(vgs - zvfb + vsb) \right]^{1/2} - zk1 \right\}
\]

\[
Qb = -qg
\]

\[
Qs = 0
\]

**50/50 Channel-Charge Partitioning for Drain and Source, XPART=.5**

**Triode Region, \(vgs > vtho, vds \leq vpof\)**

\[
Qg = \text{cap} \cdot (vgs - zvfb - zphi - 0.5 \cdot vds + vds \cdot \text{argx})
\]

\[
Qb = \text{cap} \cdot [-vtho + zvfb + zphi + (1 - \text{body}) \cdot (0.5 - \text{argx}) \cdot vds]
\]

\[
Qd = -0.5 \cdot (qg + qb)
\]

\[
Qs = Qd
\]
Saturation Region, $v_{gs} > v_{tho}, v_{ds} > v_{pof}$

$$Q_g = \frac{cap}{3} \cdot \left( v_{gs} - zvfb - zphi - \frac{v_{gs} - v_{tho}}{3 \cdot body} \right)$$

$$Q_b = \frac{cap}{3} \cdot \left[ zvfb + zphi - v_{tho} + (1 - body) \cdot \frac{(v_{gs} - v_{tho})}{3 \cdot body} \right]$$

$$Q_d = \frac{cap}{3} \cdot (v_{gs} - v_{tho})$$

$Q_s = Q_d$

40/60 Channel-Charge Partitioning for Drain and Source, XPART=0

Triode Region, $v_{gs} > v_{tho}, v_{ds} \leq v_{pof}$

$$Q_g = \frac{cap}{3} \cdot \left( v_{gs} - zvfb - zphi - 0.5 \cdot v_{ds} + argx \cdot v_{ds} \right)$$

$$Q_b = \frac{cap}{3} \cdot \left[ -v_{tho} + zvfb + zphi + (1 - body) \cdot (0.5 - argx) \cdot v_{ds} \right]$$

$$Q_d = -(cap \cdot [0.5 \cdot (v_{gs} - v_{tho} - body \cdot v_{ds}) + body \cdot argy \cdot v_{ds}])$$

$Q_s = -(Q_g + Q_b + Q_d)$

Saturation Region, $v_{gs} > v_{tho}, v_{ds} > v_{pof}$

$$Q_g = \frac{cap}{3} \cdot \left( v_{gs} - zvfb - zphi - \frac{v_{gs} - v_{tho}}{3 \cdot body} \right)$$

$$Q_b = \frac{cap}{3} \cdot \left[ zvfb + zphi - v_{tho} + (1 - body) \cdot \frac{(v_{gs} - v_{tho})}{3 \cdot body} \right]$$

$$Q_d = \frac{4 \cdot cap}{15} \cdot (v_{gs} - v_{tho})$$

$Q_s = \frac{3}{2} \cdot Q_d$
0/100 Channel-Charge Partitioning for Drain and Source, XPART=1

**Triode Region, \( vgs > vtho, vds \leq \text{vpof} \)**

\[
Q_g = \text{cap} \cdot (vgs - zvfb - zphi - 0.5 \cdot vds + vds \cdot argx) \\
Q_b = \text{cap} \cdot [-vtho + zvfb + zphi + (1 - body) \cdot (0.5 - argx) \cdot vds] \\
Q_d = -(\text{cap} \cdot [0.5 \cdot (vgs - vtho) - body \cdot vds \cdot (0.75 - 1.5 \cdot argx)]) \\
Q_s = -(Q_g + Q_b + Q_d)
\]

**Saturation Region, \( vgs > vtho, vds > \text{vpof} \)**

\[
Q_g = \text{cap} \cdot (vgs - zvfb - zphi - \frac{vgs - vtho}{3 \cdot body}) \\
Q_b = \text{cap} \cdot \left[zvfb + zphi - vtho + (1 - body) \cdot \frac{(vgs - vtho)}{3 \cdot body}\right] \\
Q_d = 0 \\
Q_s = -Q_g - Q_b
\]
Prevention of Negative Output Conductance

Star-Hspice internally protects against conditions in the LEVEL 13 model that would cause convergence problems due to negative output conductance. The constraints imposed are:

\[ ND \geq 0 \]

\[ MUS \geq MUZ + X3MS + VDD(M/2) \]

These constraints are imposed after length and width adjustment and \( V_{BS} \) dependence. This feature is gained at the expense of some accuracy in the saturation region, particularly at high \( V_{gs} \). Consequently, BSIM1 models might need to be requalified in the following situations:

1. Devices exhibit self-heating during characterization, which causes declining \( I_{ds} \) at high \( V_{ds} \). This would not occur if the device characterization measurement sweeps \( V_{ds} \).
2. The extraction technique produces parameters that result in negative conductance.
3. Voltage simulation is attempted outside the characterized range of the device.

Calculations Using LEVEL 13 Equations

To verify the equations, it is helpful to do very simple tests using Star-Hspice and check the results with a hand calculator. Check threshold, \( v_{dsat} \), and \( ids \) for a very simple model, with many parameters set to zero. There is no series resistance, \( R_{SH}=0 \). Diode current has been turned off, \( J_S=J_{SW}=I_S=0 \). The LEVEL 13 subthreshold current has been turned off by \( n_0=200 \). The geometry parameters are set to zero, so \( L_{eff}=L=1\mu, W_{eff}=W=1\mu \).

A value of \( TOX \) has been chosen to give:

\[ Cox = \frac{2.00000e-3F}{m^2} \]
The test is at \( v_{bs} = -0.35 \), so that \( \phi - v_{bs} = 1.0 \):

\[
\begin{align*}
\$ t1 & \\
.option & \text{ingold=2 numdgt=6} \\
vd & d \ 0 \ 5 \\
vg & g \ 0 \ 5 \\
vb & b \ 0 \ -0.35 \\
m1 & d \ g \ 0 \ b \ \text{nch} \ w=10u \ L=1u \\
.dc & vd \ 4 \ 5 \ 1 \\
.print & \text{id}=lx4(m1) \ \text{vth}=lv9(m1) \ \text{vdsat}=lv10(m1) \\
.model & \text{nch nmos LEVEL=13} \\
+ \text{vfb0}=-0.4 \ \text{lvfb}=0 \ \text{wvfb}=0 \\
+ \text{phi0}=0.65 \ \text{lphi}=0 \ \text{wphi}=0 \\
+ \text{k1}=0.5 \ \text{lk1}=0 \ \text{wk1}=0 \\
+ \text{k2}=0 \ \text{lk2}=0 \ \text{wk2}=0 \\
+ \text{eta0}=1e-3 \ \text{leta}=0 \ \text{weta}=0 \\
+ \text{muz}=600 \ \text{mus}=700 \ \text{x3ms}=10 \\
+ \text{x1}=0 \ \text{ld}=0 \ \text{xw}=0 \ \text{wd}=0 \\
+ \text{u00}=0 \ \text{lu0}=0 \ \text{wu0}=0 \\
+ \text{u1}=0 \ \text{lu1}=0 \ \text{wu1}=0 \\
+ \text{tox}=172.657 \\
+ \text{acm}=2 \ \text{rsh}=0 \ \text{js}=0 \ \text{jsw}=0 \ \text{is}=0 \ \text{n0}=200 \\
.end
\end{align*}
\]

**Results from Star-Hspice**

<table>
<thead>
<tr>
<th>\text{id}</th>
<th>\text{vth}</th>
<th>\text{vdsat}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.09907e-02</td>
<td>7.45000e-01</td>
<td>3.69000e+00</td>
</tr>
</tbody>
</table>

**Calculations at \( v_{gs} = v_{ds} = 5 \), \( v_{bs} = -0.35 \)**

\[
\begin{align*}
\phi - v_{bs} & = 1 \\
v_{th} & = -0.4 + 0.65 + (0.5 \cdot 1) - (\text{ETA} \cdot v_{ds}) = 0.75 - (0.001 \cdot v_{ds}) = 0.745 \\
g & = 1 - \frac{1}{(1.744 + 0.8364 \cdot 1)} = 0.612463
\end{align*}
\]
Selecting MOSFET Models: LEVEL 1-40

LEVEL 13 BSIM Model

At $v_{ds} = V_{DDM}$ (default $V_{DDM} = 5$), mobility = $\mu_s = 700$

$$body = 1 + \frac{g \cdot 0.5}{(2 \cdot 1)} = 1 + 0.25 \cdot g = 1.153116$$

$$vc = 0 \arg = 1$$

$$v_{dsat} = \frac{(v_{gs} - v_{th})}{body \cdot \sqrt{arg}} = \frac{(5 - 0.745)}{body} = 3.69000$$

These calculations agree with the Star-Hspice results given above.

Compatibility Notes

Model Parameter Naming

The following names are HSPICE-specific: U00, DL0, DW0, PH10, ETA0, NB0, ND0. A zero was added to the SPICE names to avoid conflicts with other standard Star-Hspice names. For example, U0 cannot be used because it is an alias for UB, the mobility parameter in many other levels. DL cannot be used because it is an alias for XL, a geometry parameter available in all levels.

Star-Hspice supports the use of DL0 and DW0, but the use of XL, LD, XW, WD is recommended instead (noting the difference in units).

Watch the units of TOX. It is safest to enter a number greater than one, which is always interpreted as Angstroms.
To avoid negative gds:
1. Set X3U1, LX3U1 and WX3U1 to zero.
2. Check that
   \[ zx3ms \geq 0, \text{ where } zx3ms = X3MS, \text{ with } L, W \text{ adjustment} \]
3. Check that
   \[ zmuz + VDDM \cdot zx3ms < zmus \]

**SPICE/Star-Hspice Parameter Differences**

A cross-reference table for UCB’s BSIM1 and Avant!’s LEVEL 13 model parameters is provided for comparison. Units are given in brackets. The Star-Hspice parameter name is given only if it differs from the SPICE name. The model specifies units for Star-Hspice parameters only if they differ from SPICE’s. Star-Hspice aliases are in parentheses. Note that some Star-Hspice aliases match the SPICE names.

An asterisk (*) in front of a UCB SPICE name denotes an incompatibility between the Star-Hspice name and the UCB SPICE name (that is, the Star-Hspice alias does not match, or units are different).

Even when there is a difference in parameter name between Star-Hspice and SPICE, the corresponding L and W sensitivity parameter names might not differ. L and W sensitivity parameters are only listed for the few cases for which there is a difference.

**Table 21-1: Comparison of Star-Hspice Parameters with UCB SPICE 2 and 3**

<table>
<thead>
<tr>
<th>UC Berkeley SPICE 2, 3</th>
<th>Avant!’s Star-Hspice</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFB [V]</td>
<td>VFB0 (VFB)</td>
</tr>
<tr>
<td>PHI [V]</td>
<td>PHI0</td>
</tr>
<tr>
<td>K1 [V^{1/2}]</td>
<td>same</td>
</tr>
<tr>
<td>K2</td>
<td>same</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 1-40

LEVEL 13 BSIM Model

Table 21-1: Comparison of Star-Hspice Parameters with UCB SPICE 2 and 3

<table>
<thead>
<tr>
<th>UC Berkeley SPICE 2, 3</th>
<th>Avant!’s Star-Hspice</th>
</tr>
</thead>
<tbody>
<tr>
<td>* ETA</td>
<td>ETA0</td>
</tr>
<tr>
<td>MUZ [cm²/V s]</td>
<td>same</td>
</tr>
<tr>
<td>* DL [µm]</td>
<td>DL0</td>
</tr>
<tr>
<td>* DW [µm]</td>
<td>DW0</td>
</tr>
<tr>
<td>* U0 [1/V]</td>
<td>U00</td>
</tr>
<tr>
<td>* U1 [µ/V]</td>
<td>same</td>
</tr>
<tr>
<td>X2MZ [cm²/V² s]</td>
<td>X2M (X2MZ)</td>
</tr>
<tr>
<td>LX2MZ [µm·cm²/V² s]</td>
<td>X2M (LX2MZ)</td>
</tr>
<tr>
<td>WX2MZ [µm·cm²/V² s]</td>
<td>WX2M (WX2MZ)</td>
</tr>
<tr>
<td>X2E [1/V]</td>
<td>same</td>
</tr>
<tr>
<td>X3E [1/V]</td>
<td>same</td>
</tr>
<tr>
<td>X2U0 [1/V²]</td>
<td>same</td>
</tr>
<tr>
<td>X2U1 [µm/V²]</td>
<td>same</td>
</tr>
<tr>
<td>MUS [cm²/V s]</td>
<td>same</td>
</tr>
<tr>
<td>LMUS [µm·cm²/V s]</td>
<td>LMS (LMUS)</td>
</tr>
<tr>
<td>WMUS [µm·cm²/V s]</td>
<td>WMS (WMUS)</td>
</tr>
<tr>
<td>X2MS [cm²/V² s]</td>
<td>same</td>
</tr>
<tr>
<td>X3MS [cm²/V² s]</td>
<td>same</td>
</tr>
<tr>
<td>X3U1 [µm/V²]</td>
<td>same</td>
</tr>
<tr>
<td>* TOX [µm]</td>
<td>TOX[µ] (TOX[m])</td>
</tr>
<tr>
<td>* TEMP [°C]</td>
<td>TREF</td>
</tr>
<tr>
<td>* VDD [V]</td>
<td>VDDM</td>
</tr>
</tbody>
</table>
Table 21-1: Comparison of Star-Hspice Parameters with UCB SPICE 2 and 3

<table>
<thead>
<tr>
<th>UC Berkeley SPICE 2, 3</th>
<th>Avant!’s Star-Hspice</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGDO [F/m]</td>
<td>CGDOM (CGDO)</td>
</tr>
<tr>
<td>CGSO [F/m]</td>
<td>CGSOM (CGSO)</td>
</tr>
<tr>
<td>CGBO [F/m]</td>
<td>CGBOM (CGBO)</td>
</tr>
<tr>
<td>XPART</td>
<td>same</td>
</tr>
<tr>
<td>N0</td>
<td>same</td>
</tr>
<tr>
<td>* NB</td>
<td>NB0</td>
</tr>
<tr>
<td>* ND</td>
<td>ND0</td>
</tr>
<tr>
<td>RSH [ohm/sq]</td>
<td>RSHM (RSH)</td>
</tr>
<tr>
<td>JS [A/m²]</td>
<td>IJS (JS)</td>
</tr>
<tr>
<td>PB [V]</td>
<td>PJ (PB)</td>
</tr>
<tr>
<td>MJ</td>
<td>MJ0 (MJ)</td>
</tr>
<tr>
<td>* PBSW [V]</td>
<td>PJW (PHP)</td>
</tr>
<tr>
<td>MJSW</td>
<td>MJW (MJSW)</td>
</tr>
<tr>
<td>CJ [F/m²]</td>
<td>CJM (CJ)</td>
</tr>
<tr>
<td>CJSW [F/m]</td>
<td>CCJW (CJSW)</td>
</tr>
<tr>
<td>* WDF [m]</td>
<td>−</td>
</tr>
<tr>
<td>* DELL [m]</td>
<td>−</td>
</tr>
</tbody>
</table>

In UCB SPICE, you must specify all BSIM model parameters. In Star-Hspice, there are defaults for the parameters.

**Parasitics**

ACM > 0 invokes Star-Hspice parasitic diodes. ACM=0 (default) is SPICE style.
Temperature Compensation

The model reference temperature TNOM’s default is 25°C in Star-Hspice unless .OPTION SPICE is set, causing TNOM to default to 27°C. This option also sets some other SPICE compatibility parameters. Star-Hspice TNOM is set in an .OPTION line in the netlist and can always be overridden locally (that is, for a model) with model parameter TREF. (The model “reference temperature” means that the model parameters were extracted at and are valid at that temperature).

In UCB SPICE, TNOM (default 27°C) is not effective for BSIM, and the model parameter TEMP is used instead (and must be specified) as both the model reference temperature and analysis temperature. The analysis at TEMP only applies to thermally activated exponentials in the model equations. There is no adjustment of model parameter values with TEMP. It is assumed that the model parameters were extracted at TEMP, TEMP being both the reference and the analysis temperature.

In contrast to UCB SPICE’s BSIM, Star-Hspice LEVEL 13 does provide for temperature analysis. The default analysis temperature is 25°C in Star-Hspice (and 27°C in UCB SPICE for all model levels except for BSIM, as explained in the previous paragraph). Use a .TEMP statement in the Star-Hspice netlist to change the Star-Hspice analysis temperature.

Star-Hspice provides two temperature coefficients for the LEVEL 13 model, TCV and BEX. Threshold voltage is adjusted by

\[ vth(t) = vth - TCV \cdot (t - tnom) \]

There are two implementations of the BEX factor, selected by the UPDATE parameter, which is described in the next section. The mobility in BSIM is a combination of five quantities: MUZ, zmus, z3ms, zx2mz, and zx2ms.

BEX Usage

\[ MUZ(t) = MUZ \cdot \left( \frac{t}{tnom} \right)^{BEX} \]
\[ zmus(t) = zmus \cdot \left( \frac{t}{tnom} \right)^{BEX} \]
\[ zx3ms(t) = zx3ms \cdot \left( \frac{t}{tnom} \right)^{BEX} \]
\[ zx2mz(t) = zx2mz \cdot \left( \frac{t}{tnom} \right)^{BEX} \]
\[ zx2ms(t) = zx2ms \cdot \left( \frac{t}{tnom} \right)^{BEX} \]

**Note:** This is equivalent to multiplying the final mobility by the \( \left( \frac{t}{tnom} \right)^{BEX} \) factor.

**UPDATE Parameter**

The **UPDATE** parameter selects between variations of the BSIM equations. **UPDATE=0** is the default, which is consistent with UCB SPICE3. **UPDATE=3** also is consistent with UCB SPICE3 and BEX usage.

Here is the sequence of **UPDATE** choices, which were responses to specific customer requests.

- **UPDATE=0** UCB compatible, previous BEX usage
- **UPDATE=1** Special X2E equation, previous BEX usage
- **UPDATE=2** Remove 1/Leff in U1 equation, present BEX usage
- **UPDATE=3** UCB compatible, present BEX usage

**Explanations**

The normal X2E equation is

\[ xeta = \zeta x2 e \cdot vsb + zx3 e \cdot (vds - VDDM) \]
The special X2E equation, for UPDATE=1 only, is
\[ xeta = zera + zx2e \cdot (zphi + vsb) + zx3e \cdot (vds - VDDM) \]

The special X2E equation was requested to match a parameter extraction program. Whenever you use a parameter extraction program, the equations should be checked carefully.

The original U1 equation divides by Leff in microns,
\[ xu1 = \frac{(zu1 - (zx2u1 \cdot vsb) + zx3u1 \cdot (vds - VDDM))}{Leff} \]

This is one of the few places where Leff enters explicitly into the BSIM equations; usually the Leff variation is handled by the L-adjustment model parameters, such as LU1. Physically xu1 should decrease as 1/Leff at long channels, but when dealing with short-channel devices, you can turn off this variation. Set UPDATE=2 to remove the 1/Leff factor in the xu1 equation.

UPDATE=2 introduces the present BEX usage as the 1/Leff removal ability. UPDATE=3 provides the present BEX usage with the previous xu1 equation.

**IDS and VGS Curves for PMOS and NMOS**

FILE:ML13IV.SP  IDS AND VGS CURVES FOR PMOS AND NMOS

**Two Different Types Of Model Parameter Formats Used**

OPTIONS ACCT LIST NOPAGE
.OP
.DC VDDN 0 5.0 .1 VBBN 0 -3 -3

*N-CHANNEL  I D S  CURVES (VD=0 to 5, VG=1,2,3,4,5, VB=0,-3)
.PRINT DC I(VN1) I(VN2) I(VN3) I(VN4) I(VN5) V(90)
.PLOT DC I(VN1) I(VN2) I(VN3) I(VN4) I(VN5)

*P-CHANNEL  I D S  CURVES (VD=0 to -5, VG=-1,-2,-3,-4,-5, VB=0,3)
.PRINT DC I(VP1) I(VP2) I(VP3) I(VP4) I(VP5) V(90)
.PLOT DC I(VP1) I(VP2) I(VP3) I(VP4) I(VP5)
VGS Curves

.PRINT DC I(VN6) I(VP6)
.PLOT DC I(VN6) I(VP6)
* N-CHANNEL LX7=GM (VD=5, VG=0 to ->5, VS=0, VB=0, -3)
* N-CHANNEL LX8=GD (VD=0 to 5, VG=5, VS=0, VB=0, -3)
* N-CHANNEL LX9=GB (VD=5, VG=5, VS=0, VB=0 to -5)
.PLOT DC LX7(M21) LX8(M5) LX9(M31)

* P-CHANNEL LX7=GM (VD=0, VG=0 to -5, VS=-5VB=0, 3)
* P-CHANNEL LX8=GD (VD=0 to -5, VG=-5, VS=-5VB=0, 3)
* P-CHANNEL LX9=GB (VD=0, VG=0, VS=-5VB=0, -5)
.PLOT DC LX7(M22) LX8(M15) LX9(M32)
*

VDDN 99 0 5.0
VBBN 90 0 0
EPD 98 0 99 0 -1
EPB 91 0 90 0 -1

V1 1 0 1
V2 2 0 2
V3 3 0 3
V4 4 0 4
V5 5 0 5
V11 11 0 -1
V12 12 0 -2
V13 13 0 -3
V14 14 0 -4
V15 15 0 -5
*

VN1 99 31 0
VN2 99 32 0
VN3 99 33 0
VN4 99 34 0
VN5 99 35 0

M1 31 1 0 90 PC_NM1 8U 8U
M2 32 2 0 90 PC_NM1 8U 8U
M3 33 3 0 90 PC_NM1 8U 8U
Selecting MOSFET Models: LEVEL 1-40

LEVEL 13 BSIM Model

M4  34  4  0  90 PC_Nm1  8U  8U
M5  35  5  0  90 PC_Nm1  8U  8U

*  VP1  98  41  0
  VP2  98  42  0
  VP3  98  43  0
  VP4  98  44  0
  VP5  98  45  0

M11  41  11  0  91 PC_Pm1  8U  8U
M12  42  12  0  91 PC_Pm1  8U  8U
M13  43  13  0  91 PC_Pm1  8U  8U
M14  44  14  0  91 PC_Pm1  8U  8U
M15  45  15  0  91 PC_Pm1  8U  8U

GM Test

M21  36  99  0  90 PC_Nm1  8U  8U
M22  46  98  15  91 PC_Pm1  8U  8U

GM  B CVN7  5  37  0

M31  37  5  0  98 PC_Nm1  8U  8U
M32  47  0  15  99 PC_Pm1  8U  8U

.PROCESS PC Filename=M57R

* Preliminary MOSIS BSIM parameters for SPICE3:
* The following parameters were extracted from a MOSIS
* experimental 1.2 um fabrication run.

For N-channel Devices

* NM1 PM1 PY1 ML1 ML2 DU1 DU2
*PROCESS=PC1
*RUN=m57r
*WAFER=11
*OPERATOR=david & ming
*DATE=6/12/87
LEVEL 13 BSIM Model

Selecting MOSFET Models: LEVEL 1-40

**First Model Parameter Format**

*S*mos model

```
.Model PC_NM1 NMOS LEVEL=13 VFB0=
+-8.27348E-01, 1.42207E-01, 3.48523E-02
+ 7.87811E-01, 0.00000E+00, 0.00000E+00
+ 9.01356E-01,-1.96192E-01, 1.89222E-02
+ 4.83095E-02,-4.10812E-02,-2.21153E-02
+ 2.11768E-03, 3.04656E-04,-1.14155E-03
+ 4.93528E+02, 5.39503E-02, 4.54432E-01
+ 5.81155E-02, 4.95498E-02,-1.96838E-02
+-5.88405E-02, 6.06713E-01, 4.88790E-03
+ 9.22649E+00,-8.66150E+00, 9.55036E+00
+-7.95688E-04, 2.67366E-03, 3.88974E-03
+ 2.14262E-03,-7.19261E-04, 3.56119E-03
+ 2.05529E-03,-3.66841E-03, 1.86866E-03
+-1.64733E-02,-3.63561E-03, 3.59209E-02
+ 4.84793E+02, 3.14763E+02,-3.91874E+01
+-4.21265E-00,-7.97847E+00, 3.50692E+01
+-5.83990E-00, 6.64867E+01,-1.99620E+00
+-1.44106E-02, 8.14508E-02, 7.56591E-04
+ 2.30000E-02, 2.30000E+01, 5.00000E+00
+ 5.04000E-10, 5.04000E-10, 1.91000E-09
+ 1.00000E+00, 0.00000E+00, 0.00000E+00
+ 2.00000E+02, 0.00000E+00, 0.00000E+00
+ 0.00000E+00, 0.00000E+00, 0.00000E+00
+ 0.00000E+00, 0.00000E+00, 0.00000E+00
+n diffusion layer
+80.0,7.000E-04,4.20E-010,1.00E-008,0.700E000
+0.8000e000,0.5,0.33,0,0
```

**PMOS Model**

```
.Model PC_PM1 PMOS LEVEL=13 VFB0=
+-5.63441E-01,-1.06809E-01, 1.32967E-01
+ 7.46390E-01, 0.00000E+00, 0.00000E+00
+ 6.57533E-01, 1.94464E-01,-1.60925E-01
+-2.55036E-03, 1.14752E-01,-8.78447E-02
+-5.59772E-03, 2.50199E-02,-5.66587E-04
+ 1.73854E+02, 2.72457E-01, 6.57818E-01
```
Selecting MOSFET Models: LEVEL 1-40

LEVEL 13 BSIM Model

+ 1.26943E-01, 4.25293E-02, -4.31672E-02
+-1.00718E-02, 1.50900E-01, -1.00228E-02
+ 1.03128E+01, -3.94500E+00, 1.87986E+00
+ 1.55874E-03, 4.80364E-03, -1.45355E-03
+ 4.20214E-04, -2.05447E-03, -7.44369E-04
+ 1.00044E-02, -4.43607E-03, 1.05796E-03
+-5.64102E-04, 1.97407E-03, 6.65336E-04
+ 1.77550E+02, 1.02937E+02, -2.94207E+01
+ 8.73183E+00, 1.51499E+00, 9.06178E-01
+ 1.11851E+00, 9.75265E+00, -1.88238E+00
+-4.70098E-05, 9.43069E-04, -9.19946E-05
+ 2.30000E-02, 2.30000E+01, 5.00000E+00
+ 1.00000E-09, 1.00000E-09, 1.00000E-09
+ 1.00000E+00, 0.00000E+00, 0.00000E+00
+ 2.00000E+02, 0.00000E+00, 0.00000E+00
+ 0.00000E+00, 0.00000E+00, 0.00000E+00
+ 0.00000E+00, 0.00000E+00, 0.00000E+00
*p+ diffusion layer
+140.0, 4.0E-004, 2.4E-10, 1.00E-008, 0.70000E00
+0.80000E00, 0.5, 0.33, 0, 0

Wire Model for Poly and Metal Layers
*NOT REFERENCED BY ANY ELEMENTS IN THIS CIRCUIT,
*JUST FOR MODEL EXAMPLES.
*
.MODEL PC_PY1 R
*poly layer
+ 65.0
.MODEL PC_ML1 R
*metal layer 1
+ 0.200
$$$$$$$
.ALTER
$$$$$$$

Second Model Parameter Format
*nmos model
.MODEL PC_NM1 NMOS LEVEL=13
+ VFB0=-8.27348E-01 LVFB=1.42207E-01 WVFB=3.48523E-02

21-133
LEVEL 13 BSIM Model

Selecting MOSFET Models: LEVEL 1-40

PHI0=7.87811E-01  LPHI=0.00000E+00  WPHI=0.00000E+00
K1=9.01356E-01  LK1=-1.96192E-01  WK1=1.89222E-02
K2=4.83095E-02  LK2=-4.10812E-02  WK2=1.89222E-02
ETAO=2.11768E-03  LETA=3.04656E-04  WETA=-1.14155E-03
MUZ=4.93528E+02  DL0=-5.39503E-02  DW0=4.54322E-01
U0=5.81155E-02  LU0=-5.84005E-02  WU0=4.54322E-01
U1=-5.84005E-02  LU1=6.06713E-01  WU1=4.88790E-03
X2M=9.22649E+00  LX2M=-8.66150E+00  WX2M=9.55036E+00
X2E=-7.95688E-04  LX2E=2.67366E-03  WX2E=3.88974E-03
X3E=2.14262E-03  LX3E=-7.19261E-04  WX3E=-3.56119E-03
X2U0=2.05529E-03  LX2U0=-3.66841E-03  WX2U0=1.86866E-03
X2U1=-1.64733E-02  LX2U1=-3.63561E-03  WX2U1=3.59209E-02
MUS=4.83095E+02  LMS=3.14763E+02  WMS=-3.91874E+01
X2MS=-4.21265E+00  LX2MS=-7.97847E+00  WX2MS=3.50692E+01
X3MS=-5.83990E+00  LX3MS=6.64867E+01  WX3MS=-1.96200E+00
X3U1=-1.4106E-02  LX3U1=-8.14508E-02  WX3U1=7.56519E-04
TOXM=2.30000E-02  TEMPM=2.30000E+01  VDDM=5.00000E+00
CGDOM=5.04000E-10  CGSOM=5.04000E-10  CGBOM=1.91000E-09
XPART=1.00000E+00  DUM1=0.00000E+00  DUM2=0.00000E+00
N0=2.00000E+02  LN0=0.00000E+00  WN0=0.00000E+00
NB0=0.00000E+00  LNB=0.00000E+00  WNB=0.00000E+00
ND0=0.00000E+00  LND=0.00000E+00  WND=0.00000E+00
N+ Diffusion Layer

RSHM=80.0  CJM=7.000E-004  CJW=4.200E-010
IJS=1.000E-008  PJ=0.700E000
PJW=0.80000E000  MJO=0.5  MJW=0.33
WDF=0  DS=0

PMOS Model

.MODEL PC_PM1 PMOS LEVEL=13
VFB0=-5.63414E-01  LVFB=-1.06809E-01  WVFB=1.32967E-01
PHI0=7.46330E-01  LP=0.00000E+00  WPHI=0.00000E+00
K1=6.57533E-01  LK1=-1.94464E-01  WK1=1.60925E-01
K2=2.55036E-03  LK2=1.14752E-01  WK2=-8.78474E-02
ETAO=5.59772E-03  LETA=2.50199E-02  WETA=-5.66587E-04
MUZ=1.73854E+02  DL0=2.72457E-01  DW0=6.57818E-01
U0=1.26934E-01  LU0=-4.25293E-02  WU0=-4.31672E-02
U1=-1.00718E-02  LU1=1.50900E-01  WU1=-1.00228E-02
X2M=1.03128E+01  LX2M=-3.94500E+00  WX2M=1.87986E+00

Selecting MOSFET Models: LEVEL 1-40

LEVEL 13 BSIM Model

+ X2E=1.55874E-03  LX2E=4.80364E-03  WX2E=-1.45355E-03
+ X3E=4.20214E-04  LX3E=-2.05447E-03  WX3E=-7.44369E-03
+ X2U0=1.00044E-02  LX2U0=-4.43607E-03  WX2U0=6.65336E-04
+ MUS=1.77550E+02  LMS=1.02937E+02  WMS=-2.94207E+01
+ X2MS=8.73183E+00  LX2MS=1.51499E+00  WX2MS=9.06178E+00
+ X3MS=1.11851E+00  LX3MS=9.75265E+00  WX3MS=-1.11851E+00
+ X3U1=-4.70098E-05  LX3U1=9.43069E-04  WX3U1=-9.19946E-05
+ TOXM=2.30000E-02  TEMPM=2.30000E+01  VDDM=5.00000E+00
+ CGDOM=1.00000E-09  CGSOM=1.00000E-09  CGBOM=1.91000E-09
+ XPART=1.00000E+00  DUM1=0.00000E+00  DUM2=0.00000E+00
+ NO=2.00000E+02  LNO=0.00000E+00  WN0=0.00000E+00
+ NB0=0.00000E+00  LNB=0.00000E+00  WNB=0.00000E+00
+ ND0=0.00000E+00  LND=0.00000E+00  WND=0.00000E+00
*p+ diffusion layer
+ RSHM=140.0  CJM=4.0E-004  CJW=2.4E-010
+ IJS=1.00E-008  PJ=0.700E000
+ PJW=0.8000E000  MJ0=0.5  MJW=0.33
+ WDF=0  DS=0

Wire Model for Poly and Metal Layers
*NOT REFERENCED BY ANY ELEMENTS IN THIS CIRCUIT,
*JUST FOR MODEL EXAMPLES.
*
.MODEL PC_PY1 R
*poly layer
+ RSH=65.0
.MODEL PC_ML1 R
*metal layer 1
+ RSH=0.200
*
.END
LEVEL 27 SOSFET Model

A three-terminal silicon-on-sapphire (SOS) FET transistor model is available in Star-Hspice\(^4\). This SOSFET model is based on a sapphire insulator that isolates the substrate and models the behavior of SOS devices more accurately than standard MOSFET models with physically unreal parameter values. The SOSFET model also includes a charge conservation model (Ward and Dutton model based).

Because the defaults of the SOSFET model parameters are channel-length dependent, you must specify the model parameter SOSLEV to select either the 5 \(\mu\)m or 3 \(\mu\)m processing model.

Setting SOSLEV=1 selects the 5 \(\mu\)m model; otherwise the 3 \(\mu\)m model is automatically set, including the second-order effects (default=3 \(\mu\)m).

*Note:* There is no bulk node specification for this model. If bulk nodes are specified, Star-Hspice ignores them.

This model does not use the model parameter ACM because the model includes no junction diodes. Also, the model parameter CAPOP only accepts a value of 7. Seven is its own charge conservation model, which cannot be used by the other level MOSFET models.

Temperature compensation equations for SOSFET model parameters VTO and UO are the same as those used for the MOSFET model.

*Note:* The model provides a special option for bulk nodes for silicon on sapphire. In the model definition, when you specify -1 for the bulk node, the model generates a special node for each element. This bulk node is named in the form, B#<element name>, where the element name is that of the defined element. Use this name in any statement, such as a .PRINT statement, to refer to the element’s bulk node.
**Syntax**

```
.MODEL mname PMOS <LEVEL=27> <SOSLEV=val> <pname1=val1>...
```

or

```
.MODEL mname NMOS <LEVEL=27> <SOSLEV=val> <pname=val1>...
```

- **mname**: The model name
- **PMOS**: Identifies a p-channel MOSFET model
- **NMOS**: Identifies an n-channel MOSFET model
- **LEVEL**: Model level selector
- **SOSLEV**: Selects the processing model. If you set SOSLEV=1, the default=5\(\mu\)m. The automatic default=3\(\mu\)m.
- **pname**: Parameter model
## LEVEL 27 Model Parameters

### 5-µm Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGDO</td>
<td>F/m</td>
<td></td>
<td>Gate-drain overlap capacitance per unit channel width. The default=3.1e-10 (n-type), 2.2e-10 (p-type).</td>
</tr>
<tr>
<td>CGSO</td>
<td>F/m</td>
<td></td>
<td>Gate-source overlap capacitance per unit channel width. The default=3.1e-10 (n-type), 2.2e-10 (p-type).</td>
</tr>
<tr>
<td>LD</td>
<td>m</td>
<td></td>
<td>Lateral diffusion. The default=0.6µ (n-type), 0.3µ (p-type).</td>
</tr>
<tr>
<td>RSH</td>
<td>ohm/sq</td>
<td></td>
<td>Drain and source diffusion sheet resistance. The default=25 (n-type), 100 (p-type).</td>
</tr>
<tr>
<td>SOSLEV</td>
<td></td>
<td>1</td>
<td>Model index</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>7.0e-8</td>
<td>Oxide thickness</td>
</tr>
<tr>
<td>UO</td>
<td>cm²/(V·s)</td>
<td></td>
<td>Surface mobility. The default=350 (n-type), 220 (p-type).</td>
</tr>
<tr>
<td>VTO</td>
<td>V</td>
<td></td>
<td>Threshold voltage. The default=1.25 (n-type), -1.25 (p-type).</td>
</tr>
</tbody>
</table>

### 3-µm Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>m/V</td>
<td>0.1µm</td>
<td>Channel length shortening coefficient (2nd effect)</td>
</tr>
<tr>
<td>ALPHA</td>
<td>V/m</td>
<td></td>
<td>Threshold voltage length dependence. The default=0.15µ (n-type), 0.18µ (p-type).</td>
</tr>
<tr>
<td>CAPOP</td>
<td></td>
<td>7</td>
<td>Capacitance model selector</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>CGDO</td>
<td>F/m</td>
<td></td>
<td>Gate-drain overlap capacitance per unit channel width. The default=4.6e-10 (n-type), 3.6e-10 (p-type).</td>
</tr>
<tr>
<td>CGSO</td>
<td>F/m</td>
<td></td>
<td>Gate-source overlap capacitance per unit channel width. The default=4.6e-10 (n-type), 3.6e-10 (p-type).</td>
</tr>
<tr>
<td>EC</td>
<td>V/m</td>
<td></td>
<td>Critical electric field for velocity saturation (2nd effect). The default=3.0e6 (n-type), 7.5e6 (p-type).</td>
</tr>
<tr>
<td>FB</td>
<td></td>
<td></td>
<td>Body effect coefficient (2nd effect). The default=0.15 (n-type), 0 (p-type).</td>
</tr>
<tr>
<td>LD</td>
<td>m</td>
<td></td>
<td>Lateral diffusion. The default=0.3µ (n-type), 0.2µ (p-type).</td>
</tr>
<tr>
<td>LEVEL</td>
<td>27</td>
<td></td>
<td>Model level selector</td>
</tr>
<tr>
<td>RSH</td>
<td>ohm/ sq</td>
<td></td>
<td>Drain and source diffusion sheet resistance. The default=25 (n-type), 80 (p-type).</td>
</tr>
<tr>
<td>SOSLEV</td>
<td>2</td>
<td></td>
<td>Model index</td>
</tr>
<tr>
<td>THETA</td>
<td>1/V</td>
<td></td>
<td>Mobility degradation coefficient (2nd effect). The default=0.055 (n-type), 0.075 (p-type).</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>3.4e-8</td>
<td>Oxide thickness</td>
</tr>
<tr>
<td>UO</td>
<td>cm²/ (V·s)</td>
<td></td>
<td>Surface mobility. The default=370 (n-type), 215 (p-type).</td>
</tr>
<tr>
<td>VTO</td>
<td>V</td>
<td></td>
<td>Threshold voltage. The default=0.83 (n-type), -0.74 (p-type).</td>
</tr>
</tbody>
</table>

**Example**

*FILE ML27IV.SP: IDS AND VGS CURVES FOR NMOS AND PMOS SOSFETS.*

.OPTIONS ACCT LIST NOPAGE NOMOD

.OP
* N-CHANNEL  IDS CURVES  (VD=0->5, VG=1,2,3,4,5)
.PRINT DC I(VN1) I(VN2) I(VN3) I(VN4) I(VN5)
.PLOT  DC I(VN1) I(VN2) I(VN3) I(VN4) I(VN5)

* P-CHANNEL  IDS CURVES  (VD=0->-5, VG=-1,-2,-3,-4,-5)
.PRINT DC I(VP1) I(VP2) I(VP3) I(VP4) I(VP5)
.PLOT  DC I(VP1) I(VP2) I(VP3) I(VP4) I(VP5)

* V G S  CURVES
.PRINT DC I(VN6) I(VP6)
.PLOT  DC I(VN6) I(VP6)

* N-CHANNEL  LX7=GM  (VD=5, VG=0->5, VS=0)
* N-CHANNEL  LX8=GD  (VD=0->5, VG=5, VS=0)
* N-CHANNEL  LX9=GB  (VD=5, VG=5, VS=0)
.PLOT  DC   LX7  (M21)   LX8(M5)  LX9(M31)

* P-CHANNEL  LX7=GM  (VD=0, VG=0->-5, VS=-5)
* P-CHANNEL  LX8=GD  (VD=0->-5, VG=-5, VS=-5)
* P-CHANNEL  LX9=GB  (VD=0, VG=0, VS=-5)
.PLOT  DC   LX7(M22) LX8(M15)  LX9(M32)

* VDDN 99 0 5.0
EPD 98 0 99 0 -1

V1 1 0 1
V2 2 0 2
V3 3 0 3
V4 4 0 4
V5 5 0 5
V11 11 0 -1
V12 12 0 -2
V13 13 0 -3
V14 14 0 -4
V15 15 0 -5

* 
VN1 99 31 0
VN2 99 32 0
Selecting MOSFET Models: LEVEL 1-40

LEVEL 27 SOSFET Model

VN3 99 33 0
VN4 99 34 0
VN5 99 35 0

M1 31 1 0 N1 8U 8U
M2 32 2 0 N1 8U 8U
M3 33 3 0 N1 8U 8U
M4 34 4 0 N1 8U 8U
M5 35 5 0 N1 8U 8U
*
VP1 98 41 0
VP2 98 42 0
VP3 98 43 0
VP4 98 44 0
VP5 98 45 0

M11 41 11 0 P1 8U 8U
M12 42 12 0 P1 8U 8U
M13 43 13 0 P1 8U 8U
M14 44 14 0 P1 8U 8U
M15 45 15 0 P1 8U 8U
*
G M Test
VN6 5 36 0
VP6 0 46 0
M21 36 99 0 N1 8U 8U
M22 46 98 15 P1 8U 8U
*
G M B Test
VN7 5 37 0
VP7 0 47 0
M31 37 5 0 98 N1 8U 8U
M32 47 0 15 99 P1 8U 8U
*

.MODEL N1 NMOS LEVEL=27 SOSLEV=2
+ VTO=0.814 TOX=0.34E-7 THETA=0.55E-1
+ FB=0.15 EC=0.3E7 A=0.1E-6
+ UO=370 CGSO=0.46E-9 CGDO=0.46E-9
+ RSH=25 LD=0.3E-6


21-141
Non-Fully Depleted SOI Model

When using Star-Hspice for SOS/SOI applications, several approaches are currently available. Star-Hspice has a 3-terminal SOS model (LEVEL=27) that is stable for circuit design usage, but has some limitations. The model does not have provisions for depleted bulk. Use it only with non-fully depleted applications and where kink effects are not considered.

The following circuit example is a 4-terminal SOI model for incompletely depleted bulk with kink effect. The example uses a subcircuit to allow a parasitic capacitance to the substrate. In this example, the bulk is considered to be the region under the channel. The substrate is assumed to be the conductive layer under the insulator.

For SOI, the insulator is usually silicon dioxide and the substrate is silicon. For SOS, the insulator is sapphire and the substrate is the metal that contacts the back of the integrated circuit die.

Model Components

The model consists of the following subcomponents:

- Core IDS model: any level works since the impact ionization and weak inversion models are common to all DC levels. The example uses a LEVEL=3 DC MOS model.
- Subthreshold model: the model parameter WIC=3 allows the older models to use the more advanced models found in the BSIM (LEVEL=13, LEVEL=28) models. Model parameter N0 should have a typical value around 1.0.
Impact ionization model: set the parameters ALPHA and VCR to enable the impact ionization model. Impact ionization is available to all MOS DC equations. Typical values are ALPHA=0.1 and VCR=18.

Charge conservation gate cap model (CAPOP=9 XQC=.4) keeps the floating bulk node from obtaining extreme values.

The automatic periphery diode area calculation method (ACM) is set to 3 to allow automatic calculation of the source and drain resistances and diode junction leakage and capacitance. (ACM=3 CJ=0 CJSW=0 CJSW=4e-10 JS=0 JSW=1e-9 LD=.1u HDIF=1.5u RS=40 RD=40 N=1).

**Note:** It is assumed that the source/drain diffusions extend to the buried oxide; thus, the area part of the diode has no capacitance to bulk. Linear capacitors to the substrate, however, are included in the subcircuit.

**Obtaining Model Parameters**

Use the Star-Hspice optimizing capabilities to obtain the core IDS model parameters.

Use the optimizer to get the core model, subthreshold, and impact ionization parameters. The subthreshold model selected is an improved BSIM type of model that was altered for the older models. The impact ionization model is similar to the Intel model.

The charge conservation model is more charge conserving than the original Ward-Dutton model in SPICE 2G6.

The automatic diode area and resistance calculation estimates the junction capacitance, saturation current, and resistance as a function of the transistor width. The parameters VNDS and NDS allow for a piecewise linear approximation to the reverse junction current characteristics.
**Example**

`ssoi.sp` LEVEL=3 floating bulk model

** non-fully depleted

* test 1st order soi model with floating substrate
  .option nomod post

  * substrate capacitance 3.45e-11 is for SiO2
  .param t_sub_ox=.5u  subcap='3.45e-11/t_sub_ox'  hdif=1.5u

  .global substrate
  .dc vd 0 5 0.1 sweep vg 1.5 3.5 0.5
  .print id=i(xm1.m)  vds=v(d)  vgs=v(g)
  .param vds=0  vgs=0  vbs=0
  vd d gnd vds
  vg g gnd vgs
  vs s gnd 0
  vsub substrate gnd vbs
  xm1 d g s  nch w=50u  L=5u

  .macro nch d g s  w=10u  l=2u
  * macro definition for fet+ parasitic cap to substrate
  * assumes existence of undepleted bulk
Selecting MOSFET Models: LEVEL 1-40

LEVEL 27 SOSFET Model

m d g s b nch w=w L=L
cx d substrate c='w*2*hdif*subcap’
cx s substrate c='w*2*hdif*subcap’
cx b substrate c='w*L*subcap’
eom

.model nch nmos LEVEL=3
+ lmin=.5u lmax=100u wmin=.5u wmax=500u $model selector
+ ld=0.1u wd=.15u xl=0 xw=0 $diffusion+photobias
+ acm=3 hdif=hdif rsh=30 rs=10k rd=10k $resistors
+ ldif=0.1u
$junction cap (ACM=3 (h9007 only) allows diode on gate edge
+ cj=0 cjsw=0 cgate=0.4e-9 mjsw=0.33 php=0.6
+ js=0 jsw=1e-9 n=1 vnds=.5 nds=1 $junction leakage
+ bex=-1.5 tcv=2m $temperature
+ tox=200 capop=9 xqc=.4 meto=0.08u $gate cap
+ alpha=0.1 vcr=18 $impact ionization
+ vto=0.7 phi=1 gamma=1 $threshold
+ eta=10 xj=0.1u $threshold
+ wic=3 n0=0.9 nd=0 $subthreshold
+ uo=400 theta=1m $dc mobility
+ vmax=100k kappa=0 $dc saturation
eom
Fully Depleted SOI Model Considerations

Fully-depleted transistors require additional modeling equations. The first-order effects are:

- Threshold sensitivity to the substrate
- No kink current
- Depletion capacitance hits a minimum determined by the silicon thickness

Lack of these effects is not a serious problem for an inverter circuit because the source-to-substrate voltage does not move. Digital circuits with good gate drive are not seriously affected because a large gate voltage renders the small Vth shift to a small change in IDS current.

Analog amplifiers with transistors at back-bias and low gate voltages and similar circuits can be affected by the substrate threshold sensitivity.
LEVEL 28 Modified BSIM Model

This section lists the LEVEL 28 parameters and equations for the modified BSIM model.

LEVEL 28 Features

The following are the significant features of the LEVEL 28 model.

- Vertical field dependence of carrier mobility
- Carrier velocity saturation
- Drain-induced barrier lowering
- Depletion charge sharing by source and drain
- Nonuniform doping profile for ion-implanted devices
- Channel length modulation
- Subthreshold conduction
- Geometric dependence of electrical parameters

LEVEL 28 Model Parameters

The LEVEL 28 model parameters follow.

Transistor Process Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1</td>
<td>MOSFET model level selector. Set this parameter to 28 for this Star-Hspice model.</td>
</tr>
<tr>
<td>B1</td>
<td></td>
<td>0.0</td>
<td>Lower vdsat transition point</td>
</tr>
<tr>
<td>LB1</td>
<td>μm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WB1</td>
<td>μm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>B2</td>
<td></td>
<td>1</td>
<td>Upper vdsat transition point</td>
</tr>
<tr>
<td>LB2</td>
<td>μm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>---------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>WB2</td>
<td>µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>CGBO</td>
<td>F/m</td>
<td>2.0e-10</td>
<td>Gate-to-bulk parasitic capacitance (F/m of length)</td>
</tr>
<tr>
<td>CGDO</td>
<td>F/m</td>
<td>1.5e-9</td>
<td>Gate-to-drain parasitic capacitance (F/m of width)</td>
</tr>
<tr>
<td>CGSO</td>
<td>F/m</td>
<td>1.5e-9</td>
<td>Gate-to-source parasitic capacitance (F/m of width)</td>
</tr>
<tr>
<td>ETA0</td>
<td></td>
<td>0.0</td>
<td>Linear vds threshold coefficient</td>
</tr>
<tr>
<td>LETA</td>
<td>µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WETA</td>
<td>µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>ETAMN</td>
<td></td>
<td>0.0</td>
<td>Minimum linear vds threshold coefficient</td>
</tr>
<tr>
<td>LETAMN</td>
<td>µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WETAMN</td>
<td>µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>GAMMN</td>
<td>V^{1/2}</td>
<td>0.0</td>
<td>Minimum root-vsb threshold coefficient</td>
</tr>
<tr>
<td>LGAMN</td>
<td>V^{1/2}µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WGAMN</td>
<td>V^{1/2}µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>K1</td>
<td>V^{1/2}</td>
<td>0.5</td>
<td>Root-vsb threshold coefficient</td>
</tr>
<tr>
<td>LK1</td>
<td>V^{1/2}µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WK1</td>
<td>V^{1/2}µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>K2</td>
<td></td>
<td>0.0</td>
<td>Linear vsb threshold coefficient</td>
</tr>
<tr>
<td>LK2</td>
<td>µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WK2</td>
<td>µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>MUZ</td>
<td>cm²/Vs</td>
<td>600</td>
<td>Low drain field first order mobility</td>
</tr>
<tr>
<td>LMUZ</td>
<td>µm·cm²/Vs</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WMUZ</td>
<td>$\mu m\cdot cm^2/V\cdot s$</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>N0</td>
<td></td>
<td>200</td>
<td>Low field weak inversion gate drive coefficient (value of 200 for N0 disables weak inversion calculation)</td>
</tr>
<tr>
<td>LN0</td>
<td>$\mu m$</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WN0</td>
<td>$\mu m$</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>NB0</td>
<td></td>
<td>0.0</td>
<td>Vsb reduction to low field weak inversion gate drive coefficient</td>
</tr>
<tr>
<td>LN0</td>
<td>$\mu m$</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WN0</td>
<td>$\mu m$</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>ND0</td>
<td></td>
<td>0.0</td>
<td>Vds reduction to low field weak inversion gate drive coefficient</td>
</tr>
<tr>
<td>LND</td>
<td>$\mu m$</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WND</td>
<td>$\mu m$</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>PHI0</td>
<td>$V$</td>
<td>0.7</td>
<td>Two times the Fermi potential</td>
</tr>
<tr>
<td>LPHI</td>
<td>$V\cdot \mu m$</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WPHI</td>
<td>$V\cdot \mu m$</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>TOXM (TOX)</td>
<td>$\mu m$ (m)</td>
<td>0.02</td>
<td>Gate oxide thickness (if TOXM or TOX &gt; 1, Angstroms is assumed)</td>
</tr>
<tr>
<td>U00</td>
<td>$1/V$</td>
<td>0.0</td>
<td>Gate field mobility reduction factor</td>
</tr>
<tr>
<td>LU0</td>
<td>$\mu m/V$</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WU0</td>
<td>$\mu m/V$</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>U1</td>
<td>$1/V$</td>
<td>0.0</td>
<td>Drain field mobility reduction factor</td>
</tr>
<tr>
<td>LU1</td>
<td>$\mu m/V$</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WU1</td>
<td>$\mu m/V$</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VDDM</td>
<td>V</td>
<td>5.0</td>
<td>Critical voltage for high drain field mobility reduction</td>
</tr>
<tr>
<td>VFB0 (VFB)</td>
<td>V</td>
<td>-0.3</td>
<td>Flatband voltage</td>
</tr>
<tr>
<td>LVFB</td>
<td>V.µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WVFB</td>
<td>V.µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>WFAC</td>
<td></td>
<td>4</td>
<td>Weak inversion factor</td>
</tr>
<tr>
<td>LWFAC</td>
<td>µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WWFAC</td>
<td>µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>WFACU</td>
<td></td>
<td>0.0</td>
<td>Second weak inversion factor</td>
</tr>
<tr>
<td>LWFACU</td>
<td>µm</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WWFACU</td>
<td>µm</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X2E</td>
<td>1/V</td>
<td>0.0</td>
<td>Vsb correction to linear vds threshold coefficient</td>
</tr>
<tr>
<td>LX2E</td>
<td>µm/V</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX2E</td>
<td>µm/V</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X2M (X2MZ)</td>
<td>cm²/V².s</td>
<td>0.0</td>
<td>Vsb correction to low field first order mobility</td>
</tr>
<tr>
<td>LX2M (LX2MZ)</td>
<td>µm.cm²/V².s</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX2M (WX2MZ)</td>
<td>µm.cm²/V².s</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X2U0</td>
<td>1/V²</td>
<td>0.0</td>
<td>Vsb reduction to GATE field mobility reduction factor</td>
</tr>
<tr>
<td>LX2U0</td>
<td>µm/V²</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX2U0</td>
<td>µm/V²</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 1-40

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X2U1</td>
<td>μm/V²</td>
<td>0.0</td>
<td>Vsb reduction to DRAIN field mobility reduction factor</td>
</tr>
<tr>
<td>LX2U1</td>
<td>μm²/V²</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX2U1</td>
<td>μm²/V²</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X33M</td>
<td>cm²/V²·s</td>
<td>0.0</td>
<td>Gate field reduction of X3MS</td>
</tr>
<tr>
<td>LX33M</td>
<td>μm·cm²/V²·s</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX33M</td>
<td>μm·cm²/V²·s</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X3E</td>
<td>1/V</td>
<td>0.0</td>
<td>Vds correction to linear vds threshold coefficient</td>
</tr>
<tr>
<td>LX3E</td>
<td>μm/V</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX3E</td>
<td>μm/V</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X3MS</td>
<td>cm²/V²·s</td>
<td>5.0</td>
<td>Vds correction for high drain field mobility</td>
</tr>
<tr>
<td>LX3MS</td>
<td>μm·cm²/V²·s</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX3MS</td>
<td>μm·cm²/V²·s</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>X3U1</td>
<td>1/V²</td>
<td>0.0</td>
<td>Vds reduction to drain field mobility reduction factor</td>
</tr>
<tr>
<td>LX3U1</td>
<td>μm/V²</td>
<td>0.0</td>
<td>Length sensitivity</td>
</tr>
<tr>
<td>WX3U1</td>
<td>μm/V²</td>
<td>0.0</td>
<td>Width sensitivity</td>
</tr>
<tr>
<td>XPART</td>
<td></td>
<td>1.0</td>
<td>Selector for gate capacitance charge sharing coefficient</td>
</tr>
</tbody>
</table>
Notes:

1. When reading parameter names, be aware of the difference in appearance between the capital letter O, and the number zero 0.

2. All LEVEL 28 parameters should be specified using NMOS conventions, even for PMOS—for example, ETA0 = 0.02, not ETA0 = -0.02.

3. The WL-product sensitivity parameter is available for any parameter with an L and W sensitivity. Replace the leading “L” of the L sensitivity parameter name with a “P”.

### Basic Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD (DLAT, LATD)</td>
<td>m</td>
<td></td>
<td>Lateral diffusion into channel from source and drain diffusion.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If LD and XJ are unspecified, the LD default=0.0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>When LD is unspecified but XJ is specified, LD is calculated from XJ. The LD default=0.75 XJ.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LDscaled = LD \cdot SCALM</td>
</tr>
<tr>
<td>LDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as LD, but if LDAC is included in the .MODEL statement, it replaces LD in the Leff calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>LMLT</td>
<td>1.0</td>
<td></td>
<td>Length shrink factor</td>
</tr>
<tr>
<td>LREF</td>
<td>m</td>
<td>0.0</td>
<td>Reference channel length</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LREFscaled = LREF \cdot SCALM</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>XLREF</td>
<td>m</td>
<td>0.0</td>
<td>Difference between physical (on wafer) and drawn reference channel length ( XLREF_{scaled} = XLREF \cdot SCALM )</td>
</tr>
<tr>
<td>WD</td>
<td>m</td>
<td>0.0</td>
<td>Lateral diffusion into channel from bulk along width ( WD_{scaled} = WD \cdot SCALM )</td>
</tr>
<tr>
<td>WDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as WD, but if WDAC is in the .MODEL statement, it replaces WD in the ( Weff ) calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>WMLT</td>
<td></td>
<td>1.0</td>
<td>Diffusion layer and width shrink factor</td>
</tr>
<tr>
<td>XL (DL, LDEL)</td>
<td>m</td>
<td>0.0</td>
<td>Accounts for masking and etching effects ( XL_{scaled} = XL \cdot SCALM )</td>
</tr>
<tr>
<td>XW (DW, WDEL)</td>
<td>m</td>
<td>0.0</td>
<td>Accounts for masking and etching effects ( XW_{scaled} = XW \cdot SCALM )</td>
</tr>
<tr>
<td>WREF</td>
<td>m</td>
<td>0.0</td>
<td>Reference channel width ( WREF_{scaled} = WREF \cdot SCALM )</td>
</tr>
<tr>
<td>XWREF</td>
<td>m</td>
<td>0.0</td>
<td>Difference between physical (on wafer) and drawn reference channel width ( XWREF_{scaled} = XWREF \cdot SCALM )</td>
</tr>
</tbody>
</table>
Temperature Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEX</td>
<td>-1.5</td>
<td></td>
<td>Temperature exponent for MUZ, X2M, X3MS, X33M mobility parameters</td>
</tr>
<tr>
<td>FEX</td>
<td>0.0</td>
<td></td>
<td>Temperature exponent for mobility reduction factor U1</td>
</tr>
<tr>
<td>TCV</td>
<td>V/°K</td>
<td>0.0</td>
<td>Flat-band voltage temperature coefficient</td>
</tr>
</tbody>
</table>

Sensitivity Factors of Model Parameters

For transistors, the L (channel length), W (channel width), and WL-product sensitivity factors of a basic electrical parameter are denoted by adding the characters L, W, and P, respectively, at the start of the name, and often dropping any ending “0”. For example, VFB0 sensitivity factors are LVFB, WVFB, and PVFB. If A0 is a basic parameter, LA, WA and PA are the corresponding sensitivity factors of this parameter (note that LA, WA and PA cannot be scaled using option SCALM in Star-Hspice). Then the model uses the following general formula to obtain the parameter value.

The left side of the equation represents the effective model parameter value after device size adjustment. All the effective model parameters are in lower case and start with the character z, followed by the parameter name.

\[
za = A0 + LA \cdot \left[ \frac{1}{Leff} - \frac{1}{LREFeff} \right] + WA \cdot \left[ \frac{1}{Weff} - \frac{1}{WREFeff} \right] + PA \cdot \left[ \frac{1}{Leff} - \frac{1}{LREFeff} \right] \cdot \left[ \frac{1}{Weff} - \frac{1}{WREFeff} \right]
\]

LA and WA are specified in units of microns times the units of A0. PA is specified in units of square microns times the units of A0.

If you set LREF or WREF=0, you effectively set the parameter to infinity. This is the default.
Example

\[ V_{FB0} = -0.350 v \]
\[ L_{FB} = -0.1 v \mu \]
\[ W_{FB} = 0.08 v \mu \]
\[ L_{eff} = 1 \cdot 10^{-6} m = 1 \mu \]
\[ W_{eff} = 2 \cdot 10^{-6} m = 2 \mu \]
\[ L_{REF_{eff}} = 2 \cdot 10^{-6} m = 2 \mu \]
\[ W_{REF_{eff}} = 1 \cdot 10^{-5} m = 10 \mu \]
\[ z_{vfb} = V_{FB0} + L_{FB} \cdot \left( \frac{1}{L_{eff}} - \frac{1}{L_{REF_{eff}}} \right) + W_{FB} \cdot \left( \frac{1}{W_{eff}} - \frac{1}{W_{REF_{eff}}} \right) \]
\[ z_{vfb} = -0.35 v - 0.1 v \mu \left( \frac{1}{1 \mu} - \frac{1}{2 \mu} \right) + 0.08 v \mu \left( \frac{1}{2 \mu} - \frac{1}{10 \mu} \right) \]
\[ z_{vfb} = -0.35 v - 0.05 v + 0.032 v \]
\[ z_{vfb} = -0.368 v \]

**LEVEL 28 Model Equations**

The LEVEL 28 model equations follow.
Effective Channel Length and Width

The effective channel length and width for LEVEL 28 is determined to be consistent with the LEVEL 3 model. L, W and the multiplier M are from the .MODEL statement in the netlist. SCALE and SCALM are options. When no scaling options or multipliers are used,

\[ L_{eff} = L + XL - 2 \cdot LD \]
\[ W_{eff} = W + XW - 2 \cdot WD \]

**Note:** If LDAC and WDAC are included in the .MODEL statement,

\[ L_{eff} = L + XL - 2 \cdot LD_{AC} \]
\[ W_{eff} = W + XW - 2 \cdot WD_{AC} \]

**Syntax**

\[ L_{scaled} = L \cdot SCALE \]
\[ W_{scaled} = W \cdot SCALE \]
\[ XL_{scaled} = XL \cdot SCALM \]
\[ LD_{scaled} = LD \cdot SCALM \]
\[ XW_{scaled} = XW \cdot SCALM \]
\[ WD_{scaled} = WD \cdot SCALM \]
\[ L_{eff} = L_{scaled} \cdot LMLT + XL_{scaled} - 2 \cdot LD_{scaled} \]
\[ L_{REF_{eff}} = L_{REF_{scaled}} \cdot LMLT + XL_{REF_{scaled}} - 2 \cdot LD_{scaled} \]
\[ W_{eff} = M \cdot (W_{scaled} \cdot WMLT + XW_{scaled} - 2 \cdot WD_{scaled}) \]
\[ W_{REF_{eff}} = M \cdot (W_{REF_{scaled}} \cdot WMLT + XW_{scaled} - 2 \cdot WD_{scaled}) \]

**Threshold Voltage**

Effective model parameter values for threshold voltage after device size adjustment are zphi, zvfb, zk1, zk2, zeta, zx2e, zx3e, zgammn, and zetamn. They are calculated from the model parameters PHI0, VFB0, K1, K2, ETA0, X2E, X3E, GAMMN, ETAMN, and their respective length and width sensitivity parameters.

\[ x_{bs} = (zphi - v_{bs})^{1/2} \]
\[ x_{eta} = zeta + zx2e \cdot v_{bs} + zx3e \cdot vds \]


\[ v_{th} = zvfb + zphi + zk1 \cdot xbs - zk2 \cdot xbs^2 - xeta \cdot vds \]

This equation is quadratic in \( xbs \) and \( vds \). It is joined to linear equations at \( \frac{d(vth)}{d(xbs)} = z\gammamn \) and at \( \frac{d(vth)}{d(vds)} = -z\etamn \), which prevents the quadratics from going in the wrong direction.

Both \( \gammamn \) and \( \etamn \) default to zero and typically do not affect behavior in the normal operating region.

### Effective Mobility

The effective model parameter values for mobility after device size adjustment are \( zmuz, zx2m, zx3m, zx33m, zu0, \) and \( zx2u0 \). They are calculated from the model parameters \( MUZ, X2M, X3m, X33M, U00, X2U0, \) and their respective length and width sensitivity parameters.

\[
\begin{align*}
  v_{gst} &= v_{gs} - v_{th} \\
  cx3ms &= \frac{zx3ms}{(muz + zx33m \cdot v_{gst})} \\
  m_{eff} &= (zmuz + zx2m \cdot v_{bs}) \\
  xu0 &= zu0 + zx2u0 \cdot v_{bs} \\
  u_{eff} &= \frac{m_{eff}}{1 + xu0 \cdot v_{gst}} \\
  beta &= u_{eff} \cdot COX \cdot \frac{W_{eff}}{L_{eff}}
\end{align*}
\]
Saturation Voltage (\(\text{vdsat}\))

The effective model parameter values for saturation voltage after device size adjustment are \(zu1\), \(zx2u1\), and \(zx3u1\). They are calculated from the model parameters \(U1\), \(X2U1\), \(X3U1\) and their respective length and width sensitivity parameters.

\[
x_{bs} = (zphi - v_{bs})^{1/2}
\]

\[
g = 1 - \frac{1}{(1.744 + 0.8364 \cdot x_{bs}^2)}
\]

\[
body = \frac{1 + g \cdot zk1}{(2 \cdot x_{bs})}
\]

\[
xu1 = zu1 + vbs \cdot zx2u1
\]

\[
rx = (body^2 + zu1 \cdot 2 \cdot body \cdot v_{gst} + zx3u1 \cdot 4 \cdot v_{gst}^2)^{1/2}
\]

\[
v_{dsat} = \frac{2 \cdot v_{gst}}{(body + rx)}
\]

This is the value of \(vds\) that makes the partial derivative of

\[
f(v_{ds}, v_{gst}, v_{bs}) = \left( v_{gst} - body/2 \cdot v_{ds} \right) \cdot \frac{v_{ds}}{\left( 1 + (xu1 + zx3u1 \cdot v_{ds}) \cdot v_{ds} \right)}
\]

with respect to \(vds\) equal to zero.
Transition Points
The effective model parameter values for transition points after device size adjustment are zb1 and zb2. They are calculated from the model parameters B1, B2, and their respective length and width sensitivity parameters.

\[ v_1 = v_{dsat} - zb1 \cdot \frac{v_{dsat}}{1 + v_{dsat}} \]

\[ v_2 = v_{dsat} + zb2 \cdot v_{gst} \]

Strong Inversion Current
For \( vds < v1 \),

\[ I_{ds} = beta \cdot (v_{gst} - body/2 \cdot v_{ds}) \cdot \frac{v_{ds}}{1 + (zu1 + zx3u1 \cdot v_{ds}) \cdot v_{ds}} \]

The \( vds \) derivative varies approximately linearly between \( v1 \) and \( v2 \).

For \( vds > v2 \), \( ids \) is a function of \( beta \) and \( vgst \) only. If \( zb1 \) and \( zb2 \) are both positive, their main effect is to increase the current in saturation.

Weak Inversion Current
The effective model parameter values for weak inversion current after device size adjustment are zn0, zn, znd, zwfac and zwfacu. They are calculated from the model parameters N0, ND0, NB0, WFAC, WFACU, and their respective length and width sensitivity parameters.

The weak inversion current is calculated when zn0 is less than 200. It is added to the strong inversion current,

\[ I_{total} = I_{strong} + I_{weak} \cdot \left( 1 - \exp\left( \frac{-v_{ds}}{v_{therm}} \right) \right) \]
In deep subthreshold,

\[ xn = zn0 + znb \cdot v_{bs} + znd \cdot v_{ds} \]


\[ v_{therm} = \frac{KT}{Q} \]

\[ x_{weak} = \frac{(v_{gs} - vt)}{(xn \cdot v_{therm})} \]

\[ I_{weak} = const \cdot \exp(x_{weak}) \]

The modification of this formula near threshold is controlled by zwfac and zwfacu. Just above threshold, the device is in saturation:

\[ I_{strong} = const \cdot x_{weak}^2 \]

so Iweak needs an \( x_{weak}^2 \) term to cancel the kink in gm at threshold. Then Iweak goes to zero for \( x_{weak} > A0 \), which is at a small voltage above threshold. Iweak has four regions:

(1) \( x_{weak} < -zwfac + A0 \)

\[ I_{weak} = const \cdot \exp(x_{weak}) \]

(2) \( -zwfac + A0 < x_{weak} < 0 \)

\[ I_{weak} = const \cdot \exp(x_{weak} - const \cdot wf) \]

where \( wf \) is the integral with respect to \( x_{weak} \) of

\[ dwf = \frac{(x_{weak} + zwfac - A0)^2}{[(1 + x_{weak} + zwfac - A0)(1 + zwfacu \cdot (x_{weak} + zwfac - A0))]} \]
(3) $0 < x_{\text{weak}} < A_0$

$$I_{\text{weak}} = (\text{same formula as in region 2}) - \text{const} \cdot x_{\text{weak}}^2$$

(4) $A_0 < x_{\text{weak}}$

$$I_{\text{weak}} = 0$$

A0 and the constants in the formulas above are not model parameters, but are uniquely determined by continuity conditions at the boundaries between regions.
LEVEL 38 IDS: Cypress Depletion Model

The LEVEL 38 Cypress Depletion MOSFET model (Cypress Semiconductor Corporation) is a further development of the Star-Hspice LEVEL 5 model and features:

- BSIM-style length and width sensitivities
- Degraded body effect at high substrate bias (second GAMMA)
- Empirical fitting parameters for Ids current calculations in the depletion mode of operations
- A comprehensive surface mobility equation
- Drain-induced barrier lowering

At the default parameter settings, the LEVEL 38 model is basically backwards-compatible with LEVEL 5 /ZENH=0.0, with the exception of the surface mobility degradation equation (see the discussion below). Refer to the documentation for LEVEL 5 for the underlying physics that forms the foundation for the Huang-Taylor construct.

In LEVEL 38, the temperature compensation for threshold is ASPEC-style, concurring with the default in LEVEL 5. This section introduces and documents model parameters unique to this depletion model and additional temperature compensation parameters.

LEVEL 38 allows the use of all Star-Hspice capacitance options (CAPOP). CAPOP=2 is the default setting for LEVEL 38. By setting CAPOP=6 (AMI capacitance model), LEVEL 38 capacitance calculations become identical to those of LEVEL 5.

The parameter ACM default (ACM=0 in LEVEL 38) invokes SPICE-style parasitics. ACM also can be set to 1 (ASPEC), or to 2 (Star-Hspice). All MOSFET models follow this convention.

Star-Hspice option SCALE can be used with the LEVEL 5 model. However, option SCALM cannot be used due to the difference in units. Option DERIV cannot be used.
The following parameters *must* be specified for MOS LEVEL 38: VTO (VT), TOX, UO (UB), FRC, ECV, and NSUB (DNB).

As with LEVEL 5, the Ids current is calculated according to three gate voltage regions:

**Depletion Region, vgs - vfb < 0**

The low gate voltage region dominated by the bulk channel.

**Enhancement Region, vgs - vfb > 0, vds < vgs - vfb**

The region defined by high gate voltage and low drain voltage. In the enhancement region, both channels are fully turned on.

**Partial enhancement region, vgs - vfb > 0, vds > vgs - vfb**

The region with high gate and drain voltages, resulting in the surface region being partially turned on and the bulk region being fully turned on.

To better model depletion region operations, empirical fitting constants have been added to the original Huang-Taylor mechanism to account for the effects caused by nonuniform channel implants and also to make up for an oversight in the average capacitance construct\(^5\). For the enhancement region, a significantly more elaborate surface mobility model is used.

Body effect in LEVEL 38 is calculated in two regions\(^6\):

**Bulk body effect, vsb-vsbc > 0.**

With sufficiently high (and negative) substrate bias (exceeding vsbc), the depletion region at the implanted channel-substrate junction reaches the Si-oxide interface. Under such circumstances, the free carriers can only accumulate at the interface (like in an enhancement device) and the body effect is determined by the bulk doping level.
Implant-dominated body effect, vsb-vsb < 0

Before reaching vsbc, and as long as the implant dose overwhelms the substrate doping level, the body effect of the depletion mode device is dominated by the deeply “buried” transistor due to the implant. The body effect coefficient $\gamma$ is proportional to both the substrate doping and, to first order, the implant depth. In this model level, the “amplification” of the body effect due to deep implant is accounted for by an empirical parameter, BetaGam.

Model parameters that start with L or W represent geometric sensitivities. In the model equations, a quantity denoted by $zX$ (X being the variable name) is determined by three model parameters: the large-and-wide channel case value $X$ and length and width sensitivities $LX$ and $WX$, according to $zX=X+LX/Leff+WX/Weff$. For example, the zero field surface mobility is given by

$$zUO = UO + \frac{LUO}{Leff} + \frac{WUO}{Weff}$$

Note: This model uses mostly micrometer units rather than the typical meter units. Units and defaults are often unique in LEVEL 38. The $I_{ds}$ derivatives that give small signal gains $gm$, $gds$, and $gmbs$ are calculated using the finite difference method. The options SCALM and DERIV are ineffective for this model.

LEVEL 38 Model Parameters

The LEVEL 38 model parameters follow.

Basic Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>1.0</td>
<td>Model level selector. This parameter is set to 38 for this model.</td>
</tr>
<tr>
<td>DNB (NSUB)</td>
<td>cm$^{-3}$</td>
<td>0.0</td>
<td>Surface doping density.</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 1-40

Effective Width and Length Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEL (WDEL)</td>
<td>m</td>
<td>0.0</td>
<td>Channel length reduction on each side</td>
</tr>
<tr>
<td>LATD (LD)</td>
<td>m</td>
<td>1.7 \cdot XJ</td>
<td>Lateral diffusion on each side</td>
</tr>
<tr>
<td>LDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as LD, but if LDAC is included in the .MODEL statement, it replaces LD in the Leff calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>LMLT</td>
<td></td>
<td>1.0</td>
<td>Length shrink factor</td>
</tr>
<tr>
<td>OXETCH</td>
<td>\mu m</td>
<td>0.0</td>
<td>Oxide etch</td>
</tr>
<tr>
<td>WMLT</td>
<td></td>
<td>1.0</td>
<td>Diffusion layer and width shrink factor</td>
</tr>
</tbody>
</table>

Threshold Voltage Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSS (NFS)</td>
<td>cm^{-2} \cdot V^{-1}</td>
<td>0.0</td>
<td>Number of fast surface states</td>
</tr>
<tr>
<td>NWM</td>
<td></td>
<td>0.0</td>
<td>Narrow width modifier</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SCM</td>
<td></td>
<td>0.0</td>
<td>Short-channel drain source voltage multiplier</td>
</tr>
<tr>
<td>BetaGam</td>
<td></td>
<td>1.0</td>
<td>Body effect transition ratio</td>
</tr>
<tr>
<td>LBetaGam</td>
<td>µm</td>
<td>0.0</td>
<td>BetaGam dependence on channel length</td>
</tr>
<tr>
<td>WBetaGam</td>
<td>µm</td>
<td>0.0</td>
<td>BetaGam dependence on channel width</td>
</tr>
<tr>
<td>DVSBC</td>
<td>V</td>
<td>0.0</td>
<td>Empirical body effect transition voltage adjustment</td>
</tr>
<tr>
<td>LDVSBC</td>
<td>V.µm</td>
<td>0.0</td>
<td>L-dependent body effect transition voltage adjustment</td>
</tr>
<tr>
<td>WDVSBC</td>
<td>V.µm</td>
<td>0.0</td>
<td>W-dependent body effect transition voltage adjustment</td>
</tr>
<tr>
<td>TDVSBC</td>
<td>V/K</td>
<td>0.0</td>
<td>Body effect transition voltage shift due to temperature</td>
</tr>
<tr>
<td>VT (VTO)</td>
<td>V</td>
<td>0.0</td>
<td>Extrapolated threshold voltage</td>
</tr>
<tr>
<td>LVT (LVTO)</td>
<td>V.µm</td>
<td>0.0</td>
<td>VT dependence on channel length</td>
</tr>
<tr>
<td>WVT (WVTO)</td>
<td>V.µm</td>
<td>0.0</td>
<td>VT dependence on channel width</td>
</tr>
<tr>
<td>ETA</td>
<td></td>
<td>0.0</td>
<td>Channel-length independent drain-induced barrier lowering</td>
</tr>
<tr>
<td>LETA(DIBL)</td>
<td>µm</td>
<td>0.0</td>
<td>Channel-length dependent drain-induced barrier lowering</td>
</tr>
<tr>
<td>WETA</td>
<td>µm</td>
<td>0.0</td>
<td>Channel-width dependent drain-induced barrier lowering</td>
</tr>
<tr>
<td>DVIN</td>
<td>V</td>
<td>0.0</td>
<td>Empirical surface inversion voltage adjustment</td>
</tr>
<tr>
<td>XJ</td>
<td>µm</td>
<td>1.5</td>
<td>Junction depth</td>
</tr>
</tbody>
</table>
# Mobility Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRC</td>
<td>Å·s/cm²</td>
<td>0.0</td>
<td>Field reduction coefficient</td>
</tr>
<tr>
<td>LFRC</td>
<td>$10^{-4}$Å·s/cm</td>
<td>0.0</td>
<td>FRC sensitivity to effective channel length</td>
</tr>
<tr>
<td>WFRC</td>
<td>$10^{-4}$Å·s/cm</td>
<td>0.0</td>
<td>FRC sensitivity to effective channel width</td>
</tr>
<tr>
<td>VFRC</td>
<td>Å·s/(cm²·V)</td>
<td>0.0</td>
<td>Field reduction coefficient variation due to drain bias</td>
</tr>
<tr>
<td>LVFRC</td>
<td>$10^{-4}$Å·s/(cm·V)</td>
<td>0.0</td>
<td>VFRC sensitivity to effective channel length</td>
</tr>
<tr>
<td>WVFRC</td>
<td>$10^{-4}$Å·s/(cm·V)</td>
<td>0.0</td>
<td>VFRC sensitivity to effective channel width</td>
</tr>
<tr>
<td>BFRC</td>
<td>Å·s/(cm²·V)</td>
<td>0.0</td>
<td>Field reduction coefficient variation due to substrate bias.</td>
</tr>
<tr>
<td>LBFRc</td>
<td>$10^{-4}$Å·s/(cm·V)</td>
<td>0.0</td>
<td>BFRC sensitivity to effective channel length</td>
</tr>
<tr>
<td>WBFRc</td>
<td>$10^{-4}$Å·s/(cm·V)</td>
<td>0.0</td>
<td>BFRC sensitivity to effective channel width</td>
</tr>
<tr>
<td>FSB</td>
<td>V⁰·²·s/cm²</td>
<td>0.0</td>
<td>Substrate bias-induced mobility degradation coefficient</td>
</tr>
<tr>
<td>LFSB</td>
<td>$10^{-4}$V⁰·²·s/cm</td>
<td>0.0</td>
<td>FSB sensitivity to effective channel length</td>
</tr>
<tr>
<td>WFSB</td>
<td>$10^{-4}$V⁰·²·s/cm</td>
<td>0.0</td>
<td>FSB sensitivity to effective channel width</td>
</tr>
<tr>
<td>UO (UB)</td>
<td>cm²/(V·s)</td>
<td>600</td>
<td>Low field bulk mobility</td>
</tr>
<tr>
<td>LUO(LUB)</td>
<td>cm².μm/(V·s)</td>
<td>0.0</td>
<td>UO sensitivity to effective channel length</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>WUO(WUB)</td>
<td>cm²·µm/(V·s)</td>
<td>0.0</td>
<td>UO sensitivity to effective channel width</td>
</tr>
<tr>
<td>FRCEX(F1EX)</td>
<td></td>
<td>0.0</td>
<td>Temperature coefficient for FRC</td>
</tr>
<tr>
<td>UH</td>
<td>cm²/(V·s)</td>
<td>900</td>
<td>Implant-channel mobility</td>
</tr>
<tr>
<td>KBeta1</td>
<td>(V·s)</td>
<td>1.0</td>
<td>Effective implant-channel mobility modifier</td>
</tr>
<tr>
<td>LKBeta1</td>
<td>µm</td>
<td>0.0</td>
<td>Length-dependent implant-channel mobility modifier</td>
</tr>
<tr>
<td>WKBeta1</td>
<td>µm</td>
<td>0.0</td>
<td>Width-dependent implant-channel mobility modifier</td>
</tr>
<tr>
<td>KI0(KIO)</td>
<td></td>
<td>1.0</td>
<td>Residue current coefficient</td>
</tr>
<tr>
<td>LKI0(LKIO)</td>
<td>µm</td>
<td>0.0</td>
<td>Length-dependent residue current coefficient</td>
</tr>
<tr>
<td>WKI0(WKIO)</td>
<td>µm</td>
<td>0.0</td>
<td>Width-dependent residue current coefficient</td>
</tr>
<tr>
<td>HEX(TUH)</td>
<td></td>
<td>-1.5</td>
<td>Implant channel mobility temperature exponent</td>
</tr>
<tr>
<td>BEX</td>
<td></td>
<td>-1.5</td>
<td>Surface channel mobility temperature exponent</td>
</tr>
<tr>
<td>VST</td>
<td>cm/s</td>
<td>0.0</td>
<td>Saturation velocity</td>
</tr>
<tr>
<td>UHSAT</td>
<td>µm/V</td>
<td>0.0</td>
<td>Implant-channel mobility saturation factor</td>
</tr>
</tbody>
</table>
Capacitance Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFC</td>
<td>1.0</td>
<td></td>
<td>Area factor for MOSFET capacitance</td>
</tr>
<tr>
<td>CAPOP</td>
<td>6</td>
<td>0.0</td>
<td>Gate capacitance selector</td>
</tr>
<tr>
<td>METO</td>
<td>µm</td>
<td>0.0</td>
<td>Metal overlap on gate</td>
</tr>
</tbody>
</table>

**LEVEL 38 Model Equations**

The LEVEL 38 model equations follow.

**IDS Equations**

**Depletion, vgs-vfb <0**

\[
ids = \beta_1 \left\{ q \cdot zK10 \cdot NI \cdot vde + \frac{cav}{2} \cdot (vgs - vfb) \cdot vde - \frac{vde^2}{2} \right\} \\
\]

\[
- \frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot \left[ (vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2} \right] + I_{crit}
\]

**Enhancement, vgs-vfb  vde >0**

\[
ids = \beta_1 \left\{ q \cdot zK10 \cdot NI \cdot vde - \frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot \left[ (vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2} \right] + I_{crit} \right\} \\
+ \beta \left[ (vgs - vfb) \cdot vde - \frac{vde^2}{2} \right]
\]
Partial Enhancement, $vgs-vfb<vde$

\[
ids = \beta_1 \left\{ q \cdot zK10 \cdot NI \cdot vde + cav \cdot \left[ (vgs-vfb) \cdot vde - \frac{vde^2}{2} \right] \right. \\
- \frac{2}{3} \cdot cav \cdot \gamma \cdot \left[ (vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2} \right] + I_{crit} \right. \\
+ \left( \frac{1}{2} - \frac{1}{2} \beta_1 \cdot cav \right) \cdot (vgs-vfb)^2
\]

where:

\[
\beta_1 = \frac{zKBeta1}{1 + UHSAT \cdot \frac{vde}{Leff}} \cdot UH \cdot \frac{Weff}{Leff}
\]

\[
\beta = UBeff \cdot cox \cdot \frac{Weff}{Leff}
\]

\[
cav = \frac{cox \cdot cs}{cox + cs}
\]

\[
cs = \frac{KCS \cdot \varepsilon_s}{DP \cdot 1e-4}
\]

\[
Phid = vt \cdot \ln \left( \frac{DNB \cdot nd}{ni^2} \right)
\]

\[
nd = \frac{NI \cdot 1e4}{DP}
\]
and:
\[ vde = \min(vds, vdsat) \]

The temperature dependence of the mobility terms assume the ordinary exponential form:
\[ UH(t) = UH(t_{nom}) \cdot \left( \frac{t}{t_{nom}} \right)^{TUH} \]
\[ zUO(t) = zUO(t_{nom}) \cdot \left( \frac{t}{t_{nom}} \right)^{TUH} \]

The continuity term at the body effect transition point is given by
\[ I_{crit} = \frac{2}{3} \cdot cav \cdot [(vde + vsbc + \Phi id)^{3/2} - (vsbc + \Phi id)^{3/2}] \cdot \gamma(1 - \frac{1}{zBetaGam} - 1) \]
for \( vsb > vsbc \); \( I_{crit} = 0 \) otherwise.

The saturation voltage, threshold voltage, body effect transition voltage, and body effect coefficient \( \gamma \) are described in the following sections.

**Threshold Voltage, \( vth \)**

The model parameter \( VTO \), often called the “pinch-off,” is a zero-bias threshold voltage extrapolated from a large device operating in the depletion mode. The effective pinch-off threshold voltage, including the device size effects and the terminal voltages, is given by:
\[ vth = vfb - \beta d \cdot [vch - \gamma \cdot (\Phi id + vsb)^{1/2}] + vcrit \]

where:
\[ vfb = zVTO - zETA \cdot vds + \beta d \cdot (vch - \gamma_0 \cdot \Phi id^{1/2}) \]
\[ vcrit = \left( \gamma - \frac{\gamma}{zBetaGam} \right) \cdot (\Phi id + vsbc)^{1/2} \] for \( vsb > vsbc \); 0 otherwise.
The effective $\beta_d$, including small device size effects, is computed as follows:

$$\beta_d = \frac{UH \cdot cav}{zUO \cdot cox}$$

$$v_{ch} = \frac{q \cdot NI}{cav}$$

$$\gamma_0 = \frac{(2 \cdot \varepsilon_i \cdot q \cdot n_a)}{cav}$$

$$n_{a1} = \frac{nd \cdot DNB}{nd + DNB}$$

$$nd = \frac{NI}{DP \cdot 1e-4}$$

The effective $\bar{\gamma}$, including small device size effects, is computed as follows:

$$\bar{\gamma} = \frac{\gamma}{zBetaGam}$$

for vsb>vsbc, and $=g$ otherwise.

$$\gamma = \gamma_0 \cdot (1 - scf) \cdot (1 + ncf)$$

where:

**If SCM $\leq$0,**

$$scf = 0$$

otherwise,

$$scf = \frac{XJ}{Leff} \cdot \left\{ 1 + \frac{2 xd}{XJ} \cdot (SCM \cdot vds + vsb + Phid)^{1/2} \right\}^{1/2} - 1$$

**If NWM $\leq$0,**

$$ncf = 0$$
otherwise,

\[ ncf = \frac{NWM \cdot xd \cdot (Phid)^{1/2}}{Weff} \]

where:

\[ xd = \left( \frac{2 \cdot \varepsilon si}{q \cdot DNB} \right)^{1/2} \]

The body effect transition point is calculated as follows:

\[ V_{sbc} = \frac{qDP^2}{2\varepsilon si} \left( \frac{NI}{DP \cdot 1 - 4 - DNB} \right) + zDVSCB + TDVSCB \cdot (t - t_{nom}) - Phid \]

When \( vgs \leq vth \), the surface is inverted and a residual DC current exists. When \( vsb \) is large enough to make \( vth > vinth \), then \( vth \) is used as the inversion threshold voltage.

In order to determine the residual current, \( vinth \) is inserted into the \( ids \), \( vsat \), and mobility equation in place of \( vgs \) (except for \( vgs \) in the exponential term of the subthreshold current). The inversion threshold voltage at a given \( vsb \) is \( vinth \), which is computed as:

\[ vinth = vfb - q \cdot NI \cdot cox - vsb + DVIN - ETA \cdot vds \]

**Saturation Voltage, vdsat**

The saturation voltage \( vsat \) is determined by:

\[ vsat = vgs - vfb + vch + \frac{1}{2} \cdot \left\{ 1 - \left[ 1 + \frac{4}{1^2} \cdot (vgs - vfb + vch + vsb + Phid) \right]^{1/2} \right\} \]

\[ vdsat = vsat \]
Star-Hspice modifies $v_{sat}$ to include carrier velocity saturation effect:

$$v_{dsat} = v_{sat} + v_c - (v_{sat}^2 + v_c^2)^{1/2}$$

where:

$$v_c = ECV \cdot L_{eff}$$

**Mobility Reduction, $UB_{eff}$**

The surface mobility $UB$ is dependent upon terminal voltages as follows:

$$UB_{eff} = \frac{1}{z^{UO}} + \frac{(z^{FRC} + z^{VFRC} \cdot v_{de} + z^{BFRC} \cdot v_{sb}) \cdot (v_{gs} - v_{fb}) \cdot \frac{V_{de}}{V_{ST} \cdot L_e + z^{FSB} \cdot v_{sb}^{1/2}}}{TOX}$$

where:

$$L_e = L_{eff} \quad \text{Linear region}$$

$$L_e = L_{eff} - \Delta L \quad \text{Saturation region}$$

and at elevated temperatures

$$z^{FRC}(t) = z^{FRC}(t_{nom}) \cdot \left( \frac{t}{t_{nom}} \right)^{FRCEX}$$

The $\Delta L$ is the channel length modulation effect, defined in the next section. Note that $v_{fb}$ assumes the role of $v_{th}$ in the LEVEL 5 mobility equation. The degradation parameters are semi-empirical and grouped together according to their (linearized) mathematical dependencies instead of physical origin to better provide parameter extraction.\(^7\).
Channel Length Modulation

The channel length modulation effect is included by modifying the ids current as follows:

\[
ids = \frac{ids}{1 - \frac{\Delta L}{Leff}}
\]

where:

\[
\Delta L = 1e4 \cdot \left[\frac{2.73 e3 \cdot XJ}{na1 \cdot \ln(1e20/na1)}\right]^{1/3} \cdot \left[(vds - vdsat + PHI)^{1/3} - PHI^{1/3}\right]
\]

The \(\Delta L\) is in microns, assuming \(XJ\) is in microns and \(na1\) is in \(\text{cm}^{-3}\).

Subthreshold Current, \(ids\)

When device leakage currents become important for operation near or below the normal threshold voltage, the model considers the subthreshold characteristics. In the presence of surface states, the effective threshold voltage \(von\) is determined by:

\[
von = \text{max}(vth,vinth) + fast
\]

where:

\[
fast = vt \cdot \left[1 + \frac{q \cdot FSS}{cox} + \frac{\gamma}{2 \cdot (Phid + vsb)^{1/2}}\right]
\]
If \( v_{gs} < v_{on} \), then

**Partial Enhancement, \( 0 < v_{gs-vfb} < v_{de} \)**

\[
ids = \beta_1 \cdot \left\{ q \cdot zK_{10} \cdot N_I \cdot v_{de} + cav \cdot \left[ (v_{on} - v_{fb}) \cdot v_{de} - \frac{v_{de}^2}{2} \right] \right\} \\
- \frac{2}{3} \cdot cav \cdot \tilde{q} \cdot \left[ (v_{de} + v_{sb} + Phid)^{3/2} - (v_{sb} + Phid)^{3/2} \right] + I_{crit}
\]

\[
+ \frac{1}{2} \left( \beta \cdot e^{\frac{v_{gs-von}}{v_{fast}}} - \beta_1 \cdot cav \right) \cdot (v_{on} - v_{fb})^2
\]

**Full Enhancement, \( v_{gs-vfb} - v_{de} > 0 \)**

\[
ids = \beta_1 \cdot \left\{ q \cdot zK_{10} \cdot N_I \cdot v_{de} - \frac{2}{3} \cdot cav \cdot \tilde{q} \cdot [(v_{de} + v_{sb} + Phid)^{3/2} - (v_{sb} + Phid)^{3/2}] + I_{crit} \right\}
\]

\[
+ \beta \cdot \left[ (v_{on} - v_{fb}) \cdot v_{de} - \frac{v_{de}^2}{2} \right] \cdot e^{\frac{v_{gs-von}}{v_{fast}}}
\]

**Depletion, \( v_{gs-vfb} < 0 \)**

\[
ids = \beta_1 \cdot \left\{ q \cdot zK_{10} \cdot N_I \cdot v_{de} + cav \cdot \left[ (v_{gs} - v_{fb}) \cdot v_{de} - \frac{v_{de}^2}{2} \right] \right\}
\]

\[
- \frac{2}{3} \cdot cav \cdot \tilde{q} \cdot \left[ (v_{de} + v_{sb} + Phid)^{3/2} - (v_{sb} + Phid)^{3/2} \right] + I_{crit} \cdot e^{\frac{v_{gs-von}}{v_{fast}}}
\]
Example Model File
$ file Depstor.mod
.MODEL DEPSTOR NMOS LEVEL=38
   * PARASITIC ELEMENTS
   + ACM=1
   + LD=0.15u WD=0.2u $ for LEFF AND WEFF
   + CJ=0.3E-16 MJ=0.4 PB=0.8 JS=2.0E-17 $ INTRINSIC DIODE
   + CJSW=0 MJSW=0.3
   + BULK=98 $ DEFAULT NODE FOR SUBSTRATE
   * THRESHOLD
   + VTO=-2.5 LVT=-0.25 WVT=0
   + leta=0.02 eta=0.0 weta=0.0
   + TCV=0.003$ TEMPERATURE COEFFICIENT
   * MISC
   + DVIN=0.5 PHI=0.75
   + NFS=2e10 DNB=3.0E16
Mobility Model
   + UH= 1300
   + UO=495 FRC= 0.020 FSB=5e-5 VFRC=-1e-4 BFRC=-0
   + LUO=-100 LFRC=.03 LFSB=-1e-5 LVFRC=-.002 LBFRC=-1e-3
   + WUO=-.30 WFRC=-.01 WFSB=5e-5 WVFRC=-.00
   + WBFRC=-.4e-3
   + KI0=.9KBETA1=.5
   + LK10=0.16LKBETA1=-0.15
   + WK10=0.0WKBETA1=-0.0
   + BEX=-1.3 TUH=-1.0 FrceX=1.0
Body Effect
   + NWM=0.5SCM=.1
   + DVSBC=0.1LDVSBC=0 WDVSBC=0
   + TDVSBC=.002
   + BetaGam=0.9LBetaGam=-.2 WBetaGam=.1
Saturation
+ ECV= 2.9 VST=8000 UHSAT=0
* CHANNEL LENGTH MODULATION
+ XJ= 0.1
* OXIDE THICKNESS AND CAPACITANCE
+ TOX=165 CGSO=0 CAPOP=2
* CHANNEL IMPLANT
+ NI=1.5e12 KCS=3 DP=0.25
*.END
LEVEL 39 BSIM2 Model

The BSIM2 (Berkeley Short-Channel IGFET Model 2)\textsuperscript{8,9} is available in Star-Hspice as LEVEL 39. Avant!’s implementation of this model is based on Berkeley SPICE 3E2.

Provide input to the model by assigning model parameters, as for other Star-Hspice models. Tabular model entry without model parameter names (as used for BSIM1) is not allowed for BSIM2.

LEVEL 39 Model Parameters

The following is a list of the BSIM2 parameters, their units, their Star-Hspice defaults (if any), and their descriptions. There are 47 BSIM2-specific parameters listed in the following table. Considering that three of the parameters (TEMP, DELL, DFW) are not used in Star-Hspice and, considering the width and length sensitivity parameters associated with all the remaining parameters except the first six (TOX, VDD, VGG, VBB, DL, DW), the total parameter count is 120. (Unlike Berkeley SPICE, Star-Hspice has L and W sensitivity for MU0). This count does not include the “generic” MOS parameters listed in a later table or the WL-product sensitivity parameters, which are Avant! enhancements.

BSIM2 Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOX</td>
<td>m</td>
<td>0.02</td>
<td>Gate oxide thickness. (TOX &gt; 1 is assumed to be in Angstroms)</td>
</tr>
<tr>
<td>TEMP</td>
<td>C</td>
<td>-</td>
<td>NOT USED IN Star-Hspice (see the following compatibility notes)</td>
</tr>
<tr>
<td>VDD</td>
<td>V</td>
<td>5</td>
<td>Drain supply voltage (NMOS convention)</td>
</tr>
<tr>
<td>VGG</td>
<td>V</td>
<td>5</td>
<td>Gate supply voltage (NMOS convention)</td>
</tr>
<tr>
<td>VBB</td>
<td>V</td>
<td>-5</td>
<td>Body supply voltage (NMOS convention)</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>---------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>DL</td>
<td>µ</td>
<td>0</td>
<td>Channel length reduction</td>
</tr>
<tr>
<td>DW</td>
<td>µ</td>
<td>0</td>
<td>Channel width reduction</td>
</tr>
<tr>
<td>VGHIGH</td>
<td>V</td>
<td>0</td>
<td>Upper bound of the weak-strong inversion transition region</td>
</tr>
<tr>
<td>VGLOW</td>
<td>V</td>
<td>0</td>
<td>Lower bound of same</td>
</tr>
<tr>
<td>VFB</td>
<td>V</td>
<td>-0.3</td>
<td>Flat band voltage</td>
</tr>
<tr>
<td>PHI</td>
<td>V</td>
<td>0.8</td>
<td>Surface potential</td>
</tr>
<tr>
<td>K1</td>
<td>V⁻¹</td>
<td>0.5</td>
<td>Body effect coefficient</td>
</tr>
<tr>
<td>K2</td>
<td>-</td>
<td>0</td>
<td>Second order body effect coefficient (for nonuniform channel doping)</td>
</tr>
<tr>
<td>ETA0</td>
<td>-</td>
<td>0</td>
<td>Drain-induced barrier lowering coefficient.</td>
</tr>
<tr>
<td>ETAB</td>
<td>V⁻¹</td>
<td>0</td>
<td>Sensitivity of drain-induced barrier lowering coefficient to $V_{bs}$</td>
</tr>
<tr>
<td>MU0</td>
<td>cm²/V·s</td>
<td>400</td>
<td>Low-field mobility</td>
</tr>
<tr>
<td>MU0B</td>
<td>cm²/V²·s</td>
<td>0</td>
<td>Sensitivity of low-field mobility to $V_{bs}$</td>
</tr>
<tr>
<td>MUS0</td>
<td>cm²/V·s</td>
<td>600</td>
<td>High drain field mobility</td>
</tr>
<tr>
<td>MUSB</td>
<td>cm²/V²·s</td>
<td>0</td>
<td>Sensitivity of high drain field mobility to $V_{bs}$</td>
</tr>
<tr>
<td>MU20</td>
<td>-</td>
<td>0</td>
<td>Empirical parameter for output resistance</td>
</tr>
<tr>
<td>MU2B</td>
<td>V⁻¹</td>
<td>0</td>
<td>Sensitivity of empirical parameter to $V_{bs}$</td>
</tr>
<tr>
<td>MU2G</td>
<td>V⁻¹</td>
<td>0</td>
<td>Sensitivity of empirical parameter to $V_{gs}$</td>
</tr>
<tr>
<td>MU30</td>
<td>cm²/V²·s</td>
<td>0</td>
<td>Empirical parameter for output resistance</td>
</tr>
<tr>
<td>MU3B</td>
<td>cm²/V³·s</td>
<td>0</td>
<td>Sensitivity of empirical parameter to $V_{bs}$</td>
</tr>
<tr>
<td>Name (Alias)</td>
<td>Units</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>---------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>MU3G</td>
<td>cm²/ V³.s</td>
<td>0</td>
<td>Sensitivity of empirical parameter to $V_{gs}$</td>
</tr>
<tr>
<td>MU40</td>
<td>cm²/ V³.s</td>
<td>0</td>
<td>Empirical parameter for output resistance</td>
</tr>
<tr>
<td>MU4B</td>
<td>cm²/ V⁴.s</td>
<td>0</td>
<td>Sensitivity of empirical parameter to $V_{bs}$</td>
</tr>
<tr>
<td>MU4G</td>
<td>cm²/ V⁴.s</td>
<td>0</td>
<td>Sensitivity of empirical parameter to $V_{gs}$</td>
</tr>
<tr>
<td>UA0</td>
<td>V⁻¹</td>
<td>0</td>
<td>First-order vertical-field mobility reduction factor</td>
</tr>
<tr>
<td>UAB</td>
<td>V²</td>
<td>0</td>
<td>Sensitivity of first-order factor to $V_{bs}$</td>
</tr>
<tr>
<td>UB0</td>
<td>V²</td>
<td>0</td>
<td>Second-order vertical-field mobility reduction factor</td>
</tr>
<tr>
<td>UBB</td>
<td>V⁻³</td>
<td>0</td>
<td>Sensitivity of second-order factor to $V_{bs}$</td>
</tr>
<tr>
<td>U10</td>
<td>V⁻¹</td>
<td>0</td>
<td>High drain field (velocity saturation) mobility reduction factor</td>
</tr>
<tr>
<td>U1B</td>
<td>V²</td>
<td>0</td>
<td>Sensitivity of mobility reduction factor to $V_{bs}$</td>
</tr>
<tr>
<td>U1D</td>
<td>V²</td>
<td>0</td>
<td>Sensitivity of mobility reduction factor to $V_{ds}$</td>
</tr>
<tr>
<td>N0</td>
<td>-</td>
<td>0.5</td>
<td>Subthreshold swing coefficient</td>
</tr>
<tr>
<td>NB</td>
<td>V¹/²</td>
<td>0</td>
<td>Sensitivity of subthreshold swing to $V_{bs}$</td>
</tr>
<tr>
<td>ND</td>
<td>V⁻¹</td>
<td>0</td>
<td>Sensitivity of subthreshold swing to $V_{ds}$</td>
</tr>
<tr>
<td>VOF0</td>
<td>-</td>
<td>0</td>
<td>Threshold offset (normalized to NKT/q) for subthreshold.</td>
</tr>
<tr>
<td>VOFB</td>
<td>V⁻¹</td>
<td>0</td>
<td>Sensitivity of offset to $V_{bs}$</td>
</tr>
<tr>
<td>VOFD</td>
<td>V⁻¹</td>
<td>0</td>
<td>Sensitivity of offset to $V_{ds}$</td>
</tr>
<tr>
<td>Al0</td>
<td>-</td>
<td>0</td>
<td>Impact ionization coefficient</td>
</tr>
</tbody>
</table>
All BSIM2 parameters should be specified according to NMOS convention, even for a PMOS model. Examples: VDD=5, not -5, and VBB=-5, not 5, and ETA0=0.02, not -0.02.

Also see the notes following the last table in this section.

### Other SPICE Parameters

The following generic SPICE MOS parameters are used with BSIM2 in Berkeley SPICE 3. All are also Star-Hspice parameters that can be used with Star-Hspice’s BSIM2. See “Gate Capacitance Modeling” on page 21-225 and “Selecting MOSFET Model LEVELs” on page 20-4 for more information.

### Generic SPICE MOS Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIB V^-1</td>
<td></td>
<td>0</td>
<td>Sensitivity of impact ionization coefficient to $V_{bs}$.</td>
</tr>
<tr>
<td>BI0 V</td>
<td></td>
<td>0</td>
<td>Impact ionization exponent.</td>
</tr>
<tr>
<td>BIB -</td>
<td></td>
<td>0</td>
<td>Sensitivity of impact ionization exponent to $V_{bs}$.</td>
</tr>
<tr>
<td>DELL m</td>
<td></td>
<td>-</td>
<td>Length reduction of source drain diffusion. NOT USED IN Star-Hspice!</td>
</tr>
<tr>
<td>WDF m</td>
<td></td>
<td>-</td>
<td>Default width. NOT USED IN Star-Hspice. Use &quot;.OPTION DEFW=#&quot; in the netlist instead.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGDO F/m</td>
<td></td>
<td>-</td>
<td>Gate-drain overlap capacitance. Calculated if not specified and if LD or METO, and TOX are.</td>
</tr>
<tr>
<td>CGSO F/m</td>
<td></td>
<td>-</td>
<td>Gate-source overlap capacitance. This parameter is calculated if not specified and if LD or METO, and TOX are.</td>
</tr>
</tbody>
</table>
Additionally, source/drain bulk diode sidewall reverse saturation current density, JSW[A/m], is available in Star-Hspice.

**Other Star-Hspice Model Parameters Affecting BSIM2**

The following Star-Hspice MOS model parameters are needed to use some Star-Hspice enhancements, such as LDD-compatible parasitics, model parameter geometry adjustment relative to a reference device, impact ionization modeling with bulk-source current partitioning, and element temperature adjustment of key model parameters.

This is a partial list. For complete information, see “Calculating Effective Length and Width for AC Gate Capacitance” on page 20-97, “Using Drain and Source Resistance Model Parameters” on page 20-29, “Using Impact Ionization Model Parameters” on page 20-53, and “Temperature Parameters” on page 20-102. See “.MODEL VERSION Changes to BSIM2 Models” on
page 21-193 for information about how the .MODEL statement VERSION parameter changes the BSIM2 model depending on the model version number.

Star-Hspice Model Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td></td>
<td>0</td>
<td>MOS S/D parasitics selector. ACM=0 is SPICE style. ACM=2 or 3 is recommended for LDD.</td>
</tr>
<tr>
<td>SPICE3</td>
<td></td>
<td>0</td>
<td>SPICE3 model compatibility selector. For accurate SPICE3 BSIM2, set SPICE3=1.</td>
</tr>
<tr>
<td>DERIV</td>
<td></td>
<td>0</td>
<td>Derivative selector: DERIV=0 ⇒ analytic. DERIV=1 ⇒ finite difference</td>
</tr>
</tbody>
</table>
| CAPOP        |       | *       | MOS gate cap model selector: CAPOP=39 for BSIM2, CAPOP=13 for BSIM1, CAPOP=4 is a synonym for CAPOP=13.  
* If SPICE3=0, default CAPOP=13. If SPICE3=1, default CAPOP=39. |
| LMLT         |       | 1.0     | Gate length shrink factor |
| XL           | m     | 0       | Difference between physical (on wafer) and drawn channel length. This parameter is used for L_{eff} calculation only if DL=0.  
XL_{scaled} = XL \cdot SCALM |
| LD           | m     | 0       | Lateral diffusion under gate (per side) of S/D junction. This parameter is used for L_{eff} calculation only if DL=0.  
LD_{scaled} = LD \cdot SCALM |
| LDAC         | m     |         | This parameter is the same as LD, but if LDAC is included in the .MODEL statement, it replaces LD in the L_{eff} calculation for AC gate capacitance. |
**Selecting MOSFET Models: LEVEL 1-40**  
LEV E 39 BSIM2 Model

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XW</td>
<td>m</td>
<td>0</td>
<td>Difference between physical (on wafer) and drawn S/D active width. This parameter is used for $W_{\text{eff}}$ calculation only if $D_W=0$. $XW_{\text{scaled}} = XW \cdot SCALM$</td>
</tr>
<tr>
<td>LREF</td>
<td>m</td>
<td>0 (∞)</td>
<td>Reference channel length for length adjustment of BSIM model parameters. For Berkeley compatibility (LREF-&gt;∞), use LREF=0. $LREF_{\text{scaled}} = LREF \cdot SCALM$</td>
</tr>
<tr>
<td>XLREF</td>
<td>m</td>
<td>0.0</td>
<td>Difference between physical and drawn reference channel length</td>
</tr>
<tr>
<td>DELVTO</td>
<td>V</td>
<td>0</td>
<td>Threshold voltage shift. This parameter is “type” sensitive. For example, $DELVTO &gt; 0$ increases the magnitude of n-channel threshold and decreases the magnitude of p-channel threshold. It adds to the element-line $DELVTO$ parameter.</td>
</tr>
<tr>
<td>WREF</td>
<td>m</td>
<td>0 (∞)</td>
<td>Reference device width for width adjustment of BSIM model parameters. For Berkeley compatibility (WREF-&gt;∞), use WREF=0. $WREF_{\text{scaled}} = WREF \cdot SCALM$</td>
</tr>
<tr>
<td>WMLT</td>
<td>-</td>
<td>1.0</td>
<td>Diffusion and gate width shrink factor</td>
</tr>
<tr>
<td>WD</td>
<td>m</td>
<td>0</td>
<td>Channel stop lateral diffusion under gate (per side). This parameter is used for $W_{\text{eff}}$ calculation only if $D_W=0$. $W_{\text{scaled}} = WD \cdot SCALM$</td>
</tr>
<tr>
<td>WDAC</td>
<td>m</td>
<td></td>
<td>This parameter is the same as WD, but if WDAC is included in the .MODEL statement, it replaces WD in the $W_{\text{eff}}$ calculation for AC gate capacitance.</td>
</tr>
<tr>
<td>WREF</td>
<td>m</td>
<td>0.0</td>
<td>Difference between physical and drawn reference channel width</td>
</tr>
</tbody>
</table>

**Star-Hspice Manual, Release 2001.2**

21-185
LEVEL 39 Model Equations

In the following expressions, model parameters are in all upper case Roman. It is assumed that all model parameters have already been adjusted for geometry, and that those without a trailing “0” have already been adjusted for bias, as appropriate. The exceptions are U1 and N, whose bias dependences are given explicitly below.

Threshold voltage, $V_{th}$:

$$V_{th} = V_{bi} + K1 \sqrt{PHI - V_{bs}} - K2 (PHI - V_{bs}) - ETA \cdot V_{ds}$$
where:

\[ V_{bi} = VFB + PHI \]

Strong inversion \( (V_{gs} > V_{th} + VGHIGH) \):

Linear region \( (V_{ds} < V_{dsat}) \) drain-source current \( I_{DS} \):

\[
I_{DS} = \frac{\beta \left( V_{gs} - V_{th} - \frac{a}{2} V_{ds} \right)V_{ds}}{1 + UA(V_{gs} - V_{th}) + UB(V_{gs} - V_{th})^2 + U1 \cdot V_{ds}}
\]

where:

\[ V_{dsat} = \frac{V_{gs} - V_{th}}{a\sqrt{K}}, \]

\[ K = \frac{1 + V_c + \sqrt{1 + 2V_c}}{2}, \]

\[ V_c = \frac{U1S(V_{gs} - V_{th})}{a[1 + UA(V_{gs} - V_{th}) + UB(V_{gs} - V_{th})^2]}, \]

\[ U1S = U10 + U1B \cdot V_{bs}, \]

\[ U1 = U1S \left[ 1 - \Theta(V_{dsat} - V_{ds}) \frac{U1D(V_{ds} - V_{dsat})^2}{V_{dsat}^2} \right] \]

where \( \Theta(x) \) is the usual unit step function,

\[ \beta' = \beta_0 + \beta_1 \tanh \left( MU2 \frac{V_{ds}}{V_{dsat}} \right) + \beta_3 V_{ds} - \beta_4 V_{ds}^2 \]
\[ \beta_0 = \frac{W_{\text{eff}}}{L_{\text{eff}}} \mu \cdot C_{\text{ox}}, \]

\[ \beta_1 = \beta_S - (\beta_0 + \beta_3VDD \cdot \beta_4VDD^2), \]

\[ \beta_i = \frac{W_{\text{eff}}}{L_{\text{eff}}} \mu_i \cdot C_{\text{ox}}, \quad i = S, 3, 4, \]

\[ a = 1 + \frac{gK}{2n[\Phi - V_{bs}} \]

and:

\[ g = 1 - \frac{1}{1.744 + 0.8364(\Phi - V_{bs})} \]

Saturation \((V_{ds} > V_{dsat})\) drain-source current, \(I_{DS}\):

\[ I_{DS} = \frac{\beta'(V_{gs} - V_{th})^2}{2aK[1 + UA(V_{gs} - V_{th}) + UB(V_{gs} - V_{th})^2]} \cdot (1 + f) \]

where the impact ionization term, \(f\) is

\[ f = AI \cdot e^{-\frac{B}{I}} \]

Weak Inversion \((V_{gs} < V_{th} + V_{GLOW}; [V_{GLOW} < 0])\):

Subthreshold drain-source current, \(I_{ds}\):

\[ I_{DS} = \beta' \cdot V_{tm}^2 \cdot \exp\left(\frac{V_{gs} - V_{th}}{N \cdot V_{tm}} + VOFF\right) \cdot \left[1 - \exp\left(\frac{-V_{ds}}{V_{tm}}\right)\right] \cdot (1 + f) \]

where \(V_{tm} = \frac{kT}{q}\) and \(N = N0 + \frac{NB}{\sqrt{\Phi - V_{bs}}} + ND \cdot V_{ds}\)
Selecting MOSFET Models: LEVEL 1-40
LEVEL 39 BSIM2 Model

Strong inversion-to-weak inversion transition region ($V_{th} + V_{GLOW} \leq V_{gs} \leq \xi_{th} + V_{GHIGH}$):

$$V_{geff}(V_{gst}) = \sum_{j=0}^{3} C_j V_{gst}^j$$

replaces $V_{gst} = V_{gs} - V_{th}$ in the linear or saturation drain currents, based on $V_{dsat}$ ($V_{geff}$). At the lower boundary $V_{gs} - V_{th} = V_{GLOW}$, the saturation equation is assumed to be valid for all $V_{ds}$ (that is, $V_{dsat}(V_{geff}(V_{GLOW})) = 0$), to allow a match to the subthreshold equation given above. The coefficients $C_j$ of the cubic spline $V_{geff}$ are internally determined by the conditions that $I_{DS}$ and $dI_{ds}/dV_{gs}$ both be continuous at the boundaries $V_{gs} = V_{th} + V_{GLOW}$ and $V_{gs} = V_{th} + V_{GHIGH}$.

**Effective Length and Width**

If DL is nonzero:

$$L_{eff} = L_{scaled} \cdot L_{MLT} - DL$$

$$L_{REF\, eff} = L_{REF\, scaled} \cdot L_{MLT} - DL$$

Otherwise,

$$L_{eff} = L_{scaled} \cdot L_{MLT} + XL_{scaled} - 2 \cdot LD_{scaled}$$

$$L_{REF\, eff} = L_{REF\, scaled} \cdot L_{MLT} + XL_{REF\, scaled} - 2 \cdot LD_{scaled}$$

If DW is nonzero:

$$W_{eff} = (W_{scaled} \cdot W_{MLT} - DW) \cdot M$$

$$W_{REF\, eff} = (W_{REF\, scaled} \cdot W_{MLT} - DW) \cdot M$$
Otherwise,

\[
W_{\text{eff}} = (W_{\text{scaled}} \cdot WMLT + XW - 2 \cdot WD_{\text{scaled}}) \cdot M
\]

\[
WREF_{\text{eff}} = (WREF_{\text{scaled}} \cdot WMLT + XWREF_{\text{scaled}} - 2 \cdot WD_{\text{scaled}}) \cdot M
\]

**Geometry and Bias Adjustment of Model Parameters**

Most of the BSIM2 parameters have associated width and length sensitivity parameters. Avant!-proprietary WL-product sensitivity parameters can also be specified. If \( P \) is a parameter, then its associated width, length, and WL-product sensitivity parameters are \( W_P \), \( L_P \), and \( WMLT_P \), respectively. The value of the parameter \( P' \) adjusted for width, length, and WL-product is:

\[
P' = P + WP \cdot \left( \frac{1}{W_{\text{eff}}} - \frac{1}{WREF_{\text{eff}}} \right) + LP \cdot \left( \frac{1}{L_{\text{eff}}} - \frac{1}{LREF_{\text{eff}}} \right)
\]

\[
+ PP \cdot \left( \frac{1}{W_{\text{eff}}} - \frac{1}{WREF_{\text{eff}}} \right) \cdot \left( \frac{1}{L_{\text{eff}}} - \frac{1}{LREF_{\text{eff}}} \right)
\]

The \( WREF \) and \( LREF \) terms do not appear in Berkeley SPICE. They are effectively infinite, which is the Star-Hspice default.

The following BSIM2 parameters have no associated geometry sensitivity parameters:

- TOX, TEMP (not used), VDD, VGG, VBB, DL, and DW.

The BSIM2 parameters ending in “0” are assumed to be valid at zero bias, and they have associated bias sensitivities, as given in the BSIM2 parameter table.

If \( PB \), \( PD \), and \( PG \) are the geometry-adjusted \( v_{bs} \), \( v_{ds} \), and \( v_{gs} \) sensitivity parameters, respectively, associated with the geometry-adjusted zero-bias parameter \( P0 \), then in general the bias-dependent parameter \( P \) is given by

\[
P = P0 + PB \cdot V_{bs} + PD \cdot V_{ds} + PG \cdot V_{gs}
\]

The exceptions are the velocity saturation factor \( U1 \) and the subthreshold swing coefficient \( N \). Expressions for their bias dependences is given later.
Compatibility Notes

SPICE3 Flag

If model parameter SPICE3=0 (default), certain Avant! corrections to the BSIM2 equations are effective. If SPICE3 is set to 1, the equations used are as faithful as possible to the BSIM2 equations for SPICE3E2. Even in this mode, certain numerical problems have been addressed and should not be noticeable under normal circumstances.

Temperature

The model reference temperature TNOM’s default is 25°C in Star-Hspice unless .OPTION SPICE is set. In this case TNOM defaults to 27°C. This option also sets some other SPICE compatibility parameters. Star-Hspice’s TNOM is set in an .OPTION line in the netlist and can be overridden locally (that is, for a model) with model parameter TREF. (“Reference temperature” means that the model parameters were extracted at, and are therefore valid at, that temperature.)

In UCB SPICE 3, TNOM (default 27°C) is not effective for the BSIM models, and model parameter TEMP is used (and must be specified) as both the model reference temperature and analysis temperature. The analysis at TEMP only applies to thermally activated exponentials in the model equations. There is no adjustment of model parameter values with TEMP. It is assumed that the model parameters were extracted at TEMP, TEMP being both the reference and analysis temperature.

For model levels other than 4 (BSIM1) and 5 (BSIM2) in UCB SPICE3, key model parameters are adjusted for the difference between TEMP (default 27°C) and TNOM, and TEMP is specified in the netlist with .TEMP #, just as in Star-Hspice.

In contrast to UCB SPICE’s BSIM models, Star-Hspice LEVEL 39 does provide for temperature analysis. The default analysis temperature is 25°C in Star-Hspice. Set .TEMP # in the Star-Hspice netlist to change the Star-Hspice analysis temperature (TEMP as a model parameter is NOT USED). Star-Hspice provides temperature adjustment of key model parameters, as explained later.
Parasitics

ACM > 0 invokes Star-Hspice MOS source-drain parasitics. ACM=0 (default) is SPICE style. See “Star-Hspice Enhancements” on page 21-196.

Gate Capacitance Selection

CAPOP=39 selects the BSIM2 charge-conserving capacitance model as shipped with Berkeley SPICE 3E2. This is the default selection if SPICE3=1 is set. Please note that XPART (charge-sharing flag) is currently not a BSIM2 model parameter, despite its specification in the sample BSIM2 input decks shipped with Berkeley SPICE 3E. It appears that its use in SPICE 3E was as a printback debug aid. Saturation charge sharing appears to be fixed at 60/40 (S/D) in the BSIM2 capacitance model. Charge equations are given later under “Charge-based Gate Capacitance Model (CAPOP=39)” on page 21-194. See also “Modeling Guidelines and Removal of Mathematical Anomalies” on page 21-199.

Other CAPOPs can be chosen. CAPOP=13 (recommended) selects Avant!’s BSIM1-based charge-conserving capacitance model that is in common usage with Star-Hspice MOS LEVELs 13 (BSIM1) and LEVEL 28 (modified BSIM1). This option is the default selection if SPICE3=0. With this capacitance model, charge sharing can be adjusted using model parameters XPART or XQC. See “LEVEL 13 BSIM Model” on page 21-101 for more information.

Unused Parameters

SPICE model parameters DELL (S/D diode length reduction) and WDF (default device width) are not used in Star-Hspice. The function of DELL in SPICE 3E cannot be determined. A default width can be specified in Star-Hspice on the .OPTION line as DEFW (which defaults to 100µ).
.MODEL VERSION Changes to BSIM2 Models

Star-Hspice provides a VERSION parameter to the .MODEL statement, which allows portability of LEVEL 13 BSIM and LEVEL 39 BSIM2 models between Star-Hspice versions. Using the VERSION parameter in a LEVEL 13 .MODEL statement results in the following changes to the BSIM model:

<table>
<thead>
<tr>
<th>Model Version</th>
<th>Effect of VERSION on BSIM2 Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>92A</td>
<td>LEVEL 39 BSIM2 model introduced: no changes</td>
</tr>
<tr>
<td>92B</td>
<td>No changes</td>
</tr>
<tr>
<td>93A</td>
<td>Introduces gds constraints, fixes WMU3B parameter defect, and introduces MU4 parameter defect</td>
</tr>
<tr>
<td>93A.02</td>
<td>VERSION parameter introduced, fixes MU4 parameter defect</td>
</tr>
<tr>
<td>95.1</td>
<td>Fixes defects that cause PMUSB, LDAC, WDAC parameter problems, fixes GMBS defect when gds constraints are used</td>
</tr>
<tr>
<td>96.1</td>
<td>Limited ETA + ETAB \cdot vb5 ≥ 0</td>
</tr>
</tbody>
</table>

Prevention of Negative Output Conductance

Star-Hspice internally protects against conditions in the LEVEL 13 model that cause convergence problems due to negative output conductance. The constraints imposed are:

\[
MU2 \geq 0 \quad ND \geq 0 \quad AI \geq 0
\]
These constraints are imposed after length and width adjustment and VBS dependence. This feature is gained at the expense of some accuracy in the saturation region, particularly at high Vgs. Consequently, BSIM2 models might need to be requalified in the following situations:

1. Devices exhibit self-heating during characterization, which causes declining Ids at high Vds. This would not occur if the device characterization measurement sweeps Vds.

2. The extraction technique produces parameters that result in negative conductance.

3. Voltage simulation is attempted outside the characterized range of the device.

**Charge-based Gate Capacitance Model (CAPOP=39)**

The BSIM2 gate capacitance model conserves charge and has non-reciprocal attributes. The use of charges as state variables guarantees charge conservation. Charge partitioning is fixed at 60/40 (S/D) in saturation and is 50/50 in the linear region. Qs = -(Qg+Qd+Qb) in all regions.

**Accumulation region** (Vgs < Vbs + VFB):

\[ Q_g = C_{ox} \cdot W_{eff} \cdot L_{eff} (V_{gs} - V_{bs} - VFB) \]

\[ Q_b = -Q_g \]

\[ Q_d = 0 \]

**Subthreshold region** (Vbs + VFB < Vgs < Vth + VGLOW):

\[ Q_g = C_{ox} \cdot W_{eff} \cdot L_{eff} (V_{gs} - V_{bs} - VFB) \]

\[ \cdot \left[ 1 - \frac{V_{gs} - V_{bs} - VFB}{V_{gs} - V_{bs} - VFB - V_{gst}} + \frac{1}{3} \left\{ \frac{V_{gs} - V_{bs} - VFB}{V_{gs} - V_{bs} - VFB - V_{gst}} \right\}^2 \right] \]
Selecting MOSFET Models: LEVEL 1-40

LEVEL 39 BSIM2 Model

\[
\begin{align*}
Q_b &= -Q_g \\
Q_d &= 0
\end{align*}
\]

Saturation region \((V_{ds} > V_{dsat})\):

\[
Q_g = \frac{2}{3} C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst} + Q_{bulk}
\]

where:

\[
Q_{bulk} = \frac{1}{3} C_{ox} W_{eff} \cdot L_{eff} [V_{th} - V_{bs} - V_{FB}]
\]

\[
Q_b = -Q_{bulk}
\]

\[
Q_d = -\frac{4}{10} \cdot \frac{2}{3} C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst} = \left(\frac{4}{15}\right) C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst}
\]

Linear region \((V_{ds} < V_{dsat})\):

\[
Q_g = \frac{2}{3} C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst} \cdot \left[3 \left(1 - \frac{V_{ds}}{V_{dsat}}\right) + \left(\frac{V_{ds}}{V_{dsat}}\right)^2\right] + Q_{bulk}
\]

\[
Q_b = -Q_{bulk}
\]

\[
Q_d = -\frac{1}{3} C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst}
\]
Star-Hspice Enhancements

In the following expressions, model parameters are in all upper case Roman. It is assumed that all model parameters without a trailing “0” have already been adjusted for both geometry and bias, as appropriate.

Temperature Effects

TLEV=1 is enforced for LEVEL=39. No other TLEV value is currently allowed.

Threshold voltage for LEVEL 39 TLEV=1 is adjusted according to:

\[
V_{th}(T) = V_{bi}(T) + K_1 \cdot \sqrt{\phi(T)} - V_{bs} - K_2 \cdot (\phi(T) - V_{bs}) - ETA \cdot V_{ds}
\]

where:

\[
V_{bi}(T) = V_{to}(T) - K_1 \cdot \sqrt{\phi(T)} + K_2 \cdot \phi(T),
\]

\[
V_{to}(T) = V_{to} - TCV \cdot (T - T_{nom}),
\]

and the nominal-temperature, zero-bias threshold voltage is given by

\[
V_{to} = V_{bi} + K_1 \cdot \sqrt{PHI} - K_2 \cdot PHI = V_{FB} + PHI + K_1 \cdot \sqrt{PHI} - K_2 \cdot PHI,
\]

and \( \phi(T) \) is calculated according to the value of TLEV as specified.
Mobility is adjusted according to

\[ \mu(T) = \mu(T_{nom}) \cdot \left( \frac{T}{T_{nom}} \right)^{BEX} \]

where \( \mu = \frac{\beta'}{C_{ox}(W_{eff}/L_{eff})} \).

Velocity saturation is adjusted through UIS according to

\[ U1S(T) = U1S \cdot \left( \frac{T}{T_{nom}} \right)^{FEX} \]

In addition, all of the usual Star-Hspice adjustments to capacitances and parasitic diodes and resistors are effective.

**Alternate Gate Capacitance Model**

Select CAPOP=13 for Avant!’s Star-Hspice’s charge-conserving capacitance model, widely used with LEVEL=13 (BSIM1) and LEVEL=28 (improved BSIM1). See “LEVEL 13 BSIM Model” on page 21-101 for more details.

**Impact Ionization**

You can select Star-Hspice impact ionization modeling (instead of BSIM2’s) by leaving AI0=0 and specifying model parameters ALPHA [ALPHA \cdot (V_{ds} - V_{dsat}) replaces AI in equation for \( f \) in the BSIM2 equations section above], VCR (replaces BI), and IIRAT (multiplies \( f \)).

Star-Hspice impact ionization modeling differs from BSIM2’s in two ways:

1. There is a bias term, \( V_{ds} - V_{dsat} \), multiplying the exponential, as well as ALPHA.

2. The impact ionization component of the drain current can be partitioned between the source and the bulk with model parameter IIRAT. IIRAT multiplies \( f \) in the saturation \( I_{ds} \) equation. Thus, the fraction IIRAT of the impact ionization current goes to the source, and the fraction 1-IIRAT goes to the bulk, adding to \( I_{DB} \). IIRAT defaults to zero (that is, 100% of impact ionization current goes to the bulk).
BSIM2’s impact ionization assumes that all of the impact ionization current is part of $I_{ds}$. In other words, it flows to the source. This assumption can lead to inaccuracies in, for example, cascode circuits. See “Calculating the Impact Ionization Equations” on page 20-53 for more details.

**Parasitic Diode for Proper LDD Modeling**

Star-Hspice has alternative MOS parasitic diodes to replace SPICE-style MOS parasitic diodes. These alternatives allow for geometric scaling of the parasitics with MOS device dimension, proper modeling of LDD parasitic resistances, allowance for shared sources and drains, and allowance for different diode sidewall capacitances along the gate edge and field edge.

The MOS parasitic diode is selected with model parameter ACM. ACM=0 (default) chooses SPICE style. The alternatives likely to be of most interest to the BSIM2 user are ACM=2 and 3.

ACM=2 allows for diode area calculation based on W, XW, and HDIF (contact to gate spacing). The calculation can be overridden from the element line. It further allows specification of LDIF (spacer dimension) and RS, RD (source and drain sheet resistance under the spacer) for LDD devices, as well as RSH (sheet resistance of heavily doped diffusion). Thus, total parasitic resistance of LDD devices is properly calculated.

ACM=3 uses all the features of ACM=2 and, in addition, its calculations of diode parasitics takes into account the sharing of source/drains, and different junction sidewall capacitances along the gate and field edges. Specify source/drain sharing from the element line with parameter GEO.

See “Selecting MOSFET Diode Models” on page 20-26 for more details.

**Skewing of Model Parameters**

The BSIM2 model file, like any other Star-Hspice model, can be set up for skewing to reflect process variation. Worst-case or Monte-Carlo analysis can be performed, based on fab statistics. For more information, see “Performing Worst Case Analysis” on page 13-8 and “Performing Monte Carlo Analysis” on page 13-14.
Star-Hspice Optimizer

The BSIM2 model, like any other Star-Hspice model, can be tied into the Star-Hspice optimizer for fitting to actual device data.

For more information, see “Optimization” on page 13-35. An example fit appears at the end of this section.

Modeling Guidelines and Removal of Mathematical Anomalies

Because of the somewhat arbitrary geometric and bias adjustments given to BSIM2 parameters, they can take on non-physical or mathematically unallowed values in Berkeley SPICE 3. This can lead to illegal function arguments, program crashes, and unexpected model behavior (for example, negative conductance). The following guidelines and corrections must be satisfied at all geometries of interest and at biases, up to double the supply voltages (that is, to $V_{ds} = 2 \cdot V_{DD}$, $V_{gs} = 2 \cdot V_{GG}$, and $V_{bs} = 2 \cdot V_{BB}$).

To avoid drain current discontinuity at $V_{ds} = V_{dsat}$, be sure that $BI \neq AI0$.

To prevent negative $g_{ds}$, be sure that $ETA > 0$ and that $MU3 > 0$ and $MU4 < MU3 / (4 \cdot V_{DD})$. This should ensure positive $g_{ds}$ at biases up to double the supply voltages. To simplify matters, set all $MU4$ parameters to zero. You can obtain reasonably good fits to submicron devices without using $MU4^{10}$.

In Star-Hspice, $U1S$ is prevented from becoming negative. A negative $U1S$ is physically meaningless and causes negative arguments in a square root function in one of the BSIM2 equations. It is also recommended that $U1D$ be kept less than unity (between 0 and 1).

For reasonable $V_{th}$ behavior, make sure that $K1 - 2K2 \cdot \sqrt{PHI - V_{bs}} \geq 0$.

For the equations to make sense, the following must hold: $N > 0$, $V_{GLOW} \leq 0$, and $V_{GHIGH} \geq 0$.

The BSIM2 gate capacitance model of SPICE 3E tends to display negative $C_{gs}$ in subthreshold. This appears to be due to $C_{gg} \to 0$ as $V_{gs} \to V_{th}$ by construction of the gate charge equation, so that $C_{gs} \approx C_{gg} \cdot C_{gd} \cdot C_{gb} \to - C_{gd} \cdot C_{gb} \approx - C_{gb}$. Therefore the use of $CAPOP=13$ (default) is recommend until an improved BSIM2 gate capacitance model is released by Berkeley.
Modeling Example

The following is the result of fitting data from a submicron channel-length NMOS device to BSIM2. The fitting was performed with Avant!’s ATEM characterization software and the Star-Hspice optimizer.

Figure 21-8: $I_{DS}$ vs. $V_{ds}$ for $V_{gs} = 1, 2, 3, 4, 5V$; BSIM2 Model vs. Data
Figure 21-9: $g_{ds}$ vs. $V_{ds}$ for $V_{gs} = 2, 3, 4, 5V$; BSIM2 Model vs. Data, LOG scale
Figure 21-10: $I_{DS}$ vs. $V_{gs}$ for $V_{ds} = 0.1V$, $V_{bs} = 0, -1, -2, -3, -4V$, Showing Subthreshold Region; Model vs. Data
Typical BSIM2 Model Listing

In this example, geometry sensitivities are set to zero because a fit at only one geometry has been performed. Note the extra HSPICE parameters for LDD, temperature, and geometry.

```
.MODEL NCH NMOS LEVEL = 39
+ TOX = 2.000000E-02  TEMP = 2.500000E+01
+ VDD = 5.000000E+00  VGG = 5.000000E+00  VBB = -5.000000E+00
+ DL = 0.000000E+00  DW = 0.000000E+00
+ VGHIGH = 1.270000E-01  LVGHIGH= 0.000000E+00
+ WGHIGH= 0.000000E+00
+ VGLOW = -7.820000E-02  LVGLOW = 0.000000E+00
+ WVLOW = 0.000000E+00
+ VFB = -5.760000E-01  LVFB = 0.000000E+00
+ WVFB = 0.000000E+00
+ PHI = 6.500000E-01  LPHI = 0.000000E+00
```
LEVEL 39 BSIM2 Model

Selecting MOSFET Models: LEVEL 1-40

+ WPHI = 0.000000E+00
+ K1 = 9.900000E-01 LK1 = 0.000000E+00 WK1 = 0.000000E+00
+ K2 = 1.290000E-01 LK2 = 0.000000E+00 WK2 = 0.000000E+00
+ ETA0 = 4.840000E-03 LETA0 = 0.000000E+00
+ WETA0 = 0.000000E+00
+ ETAB = -5.560000E-03 LETAB = 0.000000E+00
+ WETAB = 0.000000E+00
+ MU0 = 3.000000E+02
+ MU0B = 0.000000E+00 LMU0B = 0.000000E+00
+ WMU0B = 0.000000E+00
+ MU20 = 1.170000E+00 LMU20 = 0.000000E+00
+ WMU20 = 0.000000E+00
+ MU2B = 0.000000E+00 LMU2B = 0.000000E+00
+ WMU2B = 0.000000E+00
+ MU2G = 0.000000E+00 LMU2G = 0.000000E+00
+ WMU2G = 0.000000E+00
+ MU30 = 3.000000E+01 LMU30 = 0.000000E+00
+ WMU30 = 0.000000E+00
+ MU3B = 0.000000E+00 LMU3B = 0.000000E+00
+ WMU3B = 0.000000E+00
+ MU3G = -2.970000E+00 LMU3G = 0.000000E+00
+ WMU3G = 0.000000E+00
+ MU40 = 0.000000E+00 LMU40 = 0.000000E+00
+ WMU40 = 0.000000E+00
+ MU4B = 0.000000E+00 LMU4B = 0.000000E+00
+ WMU4B = 0.000000E+00
+ MU4G = 0.000000E+00 LMU4G = 0.000000E+00
+ WMU4G = 0.000000E+00
+ UA0 = 0.000000E+00 LUA0 = 0.000000E+00
+ WUA0 = 0.000000E+00
+ UAB = 0.000000E+00 LUAB = 0.000000E+00
+ WUAB = 0.000000E+00
+ UB0 = 7.450000E-03 LUB0 = 0.000000E+00
+ WUB0 = 0.000000E+00
+ UBB = 0.000000E+00 LUBB = 0.000000E+00
+ WUBB = 0.000000E+00
Selecting MOSFET Models: LEVEL 1-40

LEVEL 39 BSIM2 Model

+ U10 = 0.0000000E+00  LU10 = 7.9000000E-01
+ WU10 = 0.0000000E+00
+ U1B = 0.0000000E+00  LU1B = 0.0000000E+00
+ WU1B = 0.0000000E+00
+ U1D = 0.0000000E+00  LU1D = 0.0000000E+00
+ WU1D = 0.0000000E+00
+ N0 = 8.3700000E-01  LN0 = 0.0000000E+00  WN0 = 0.0000000E+00
+ NB = 6.6600000E-01  LNB = 0.0000000E+00  WNB = 0.0000000E+00
+ ND = 0.0000000E+00  LND = 0.0000000E+00  WND = 0.0000000E+00
+ VOF0 = 4.7700000E-01  LVOF0 = 0.0000000E+00
+ WVOF0 = 0.0000000E+00
+ VOFB =-3.4000000E-02  LVOFB = 0.0000000E+00
+ WVOFB = 0.0000000E+00
+ VOFD =-6.9000000E-02  LVOFD = 0.0000000E+00
+ WVOFD = 0.0000000E+00
+ AI0 = 1.8400000E+00  LAI0 = 0.0000000E+00
+ WA10 = 0.0000000E+00
+ AIB = 0.0000000E+00  LAIB = 0.0000000E+00
+ WAIB = 0.0000000E+00
+ B10 = 2.0000000E+01  LBI0 = 0.0000000E+00
+ WB10 = 0.0000000E+00
+ BIB = 0.0000000E+00  LBB = 0.0000000E+00
+ WBIB = 0.0000000E+00
+ DELL = 0.0000000E+00  WDF = 0.0000000E+00

Common SPICE Parameters
+ CGDO = 1.0000000E-09  CGSO = 1.0000000E-09
+ CBO = 2.5000000E-11
+ RSH = 3.6400000E+01  JS = 1.3800000E-06
+ PB = 8.0000000E-01  PBSW = 8.0000000E-01
+ CJ = 4.3100000E-04  CJSW = 3.9600000E-10
+ MJ = 4.5600000E-01  MJSW = 3.0200000E-01
Avant! Parameters
+ ACM = 3   LMLT = 8.500000E-01
+ WMLT = 8.500000E-01
+ XL =-5.000000E-08  LD = 5.000000E-08
+ XW = 3.000000E-07  WD = 5.000000E-07
+ C_JGATE = 2.000000E-10  HDIF = 2.000000E-06
+ LDIF = 2.000000E-07
+ RS = 2.000000E+03  TRS = 2.420000E-03
+ RD = 2.000000E+03  TRD = 2.420000E-03
+ TCV = 1.420000E-03  BEX =-1.720000E+00  FEX =-2.820000E+00
+ LMU0 = 0.000000E+00  WMU0 = 0.000000E+00  JSW=2.400000E-12
LEVEL 40 HP a-Si TFT Model

Star-Hspice LEVEL 40 is a Hewlett-Packard amorphous silicon thin-film transistor model.

Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UO</td>
<td>cm²/V·s</td>
<td>1.0</td>
<td>Mobility</td>
</tr>
<tr>
<td>VTO</td>
<td>V</td>
<td>0.0</td>
<td>Zero voltage threshold voltage</td>
</tr>
<tr>
<td>PHI</td>
<td>V</td>
<td>0.0</td>
<td>Surface potential</td>
</tr>
<tr>
<td>NFS</td>
<td>cm²</td>
<td>0.0</td>
<td>Fast surface state density</td>
</tr>
<tr>
<td>NSS</td>
<td>cm²</td>
<td>0.0</td>
<td>Surface state density</td>
</tr>
<tr>
<td>T1</td>
<td>m</td>
<td>280n</td>
<td>First thin film thickness</td>
</tr>
<tr>
<td>T2</td>
<td>m</td>
<td>0.0</td>
<td>Second thin film thickness</td>
</tr>
<tr>
<td>E1</td>
<td></td>
<td>3.9</td>
<td>Dielectric constant of 1st film</td>
</tr>
<tr>
<td>E2</td>
<td></td>
<td>0.0</td>
<td>Dielectric constant of 2nd film</td>
</tr>
<tr>
<td>THETA</td>
<td>V⁻¹</td>
<td>0.0</td>
<td>Mobility modulation</td>
</tr>
<tr>
<td>ETA</td>
<td>V⁻¹</td>
<td>0.0</td>
<td>Static feedback on threshold voltage (difficulty of band bending)</td>
</tr>
<tr>
<td>VMAX</td>
<td>m/s</td>
<td>1e6</td>
<td>Maximum drift velocity of carriers</td>
</tr>
<tr>
<td>GO</td>
<td>ohm⁻¹</td>
<td>10e-15</td>
<td>Conductance of TFT leakage current</td>
</tr>
<tr>
<td>DEFF</td>
<td></td>
<td>2.0</td>
<td>Drain voltage effect for TFT leakage current</td>
</tr>
<tr>
<td>NU</td>
<td></td>
<td>0.0</td>
<td>First order temperature gradient</td>
</tr>
<tr>
<td>CHI</td>
<td></td>
<td>0.5</td>
<td>Temperature exponential part</td>
</tr>
<tr>
<td>PSI</td>
<td></td>
<td>1e-20</td>
<td>Temperature exponential part</td>
</tr>
</tbody>
</table>
Using the HP a-Si TFT Model in Star-Hspice

1. Set LEVEL=40 to identify the model as the HP a-Si TFT model.
2. The default value for L is 10µm, and the default value for W is 40 µm.
3. Use the “M” designation for MOSFET rather than the “A” designation for a-Si TFT in the netlist.
4. Use the “NMOS” or “PMOS” designation for device type rather than the “NAT” or “PAT” designation.
   **Note:** Because of the unavailability of p-channel TFTs, PMOS model testing has been very limited.
5. The LEVEL 40 model is a three-terminal model. No bulk node exists; therefore, no parasitic drain-bulk or source-bulk diodes are appended to the model. A fourth node can be specified, but does not affect simulation results (except for GMIN terms).
6. Parasitic resistances and overlap capacitances are constant. They are not scaled with width, length, and temperature.

7. The capacitance expressions in this model do not conserve charge.

8. The HP a-Si TFT model has a TREF parameter that is an exponent in an expression for mobility temperature dependence.

Other models use the BEX parameter for similar mobility temperature dependence expressions. The HP a-Si TFT TREF model parameter is not the same as the reference temperature TREF used in other models. The reference temperature for the HP a-Si TFT model is 312 K (or 38.85 °C), and cannot be modified. Experimental results from TFT manufacturers indicate that amorphous silicon materials are most stable at this temperature.

9. The default room temperature is 25° C in Star-Hspice, but is 27° C in some other simulators. It is a matter of choice whether or not to set the nominal simulation temperature to 27° C, by adding .OPTION TNOM=27 to the netlist. Although the reference temperature of the HP a-Si TFT model is fixed at 312° K (or 38.85 °C), the behavior of the model adjusts to other simulation temperatures that are user specified or provided by Star-Hspice as defaults.

10. HP’s SPICE3E2 implementation of this model, on which this implementation is based, is not temperature-dependent. The LEVEL 40 has temperature dependency enabled.

11. The default value of CAPOP is 40, which is the HP a-Si TFT non-charge-conserving capacitance model. CAPOP values of 0, 1, 2, 3, 4, 5, 9, 12, or 13 are allowed, but have not been thoroughly tested.

12. The default of DERIV is zero, the analytical method. DERIV can be set to 1 to select the finite difference method.
Effect of SCALE and SCALM

The SCALE option has the same effect for LEVEL 40 as for other Star-Hspice models, such as LEVEL 3 or LEVEL 28. If the values of L and W are in microns rather than meters (for example, $L=1$ rather than $L=1\mu$ or $1e-6$), set .OPTION SCALE=1e-6.

The SCALM option is disabled in the LEVEL 40 model. For standard Star-Hspice models such as LEVEL 3, SCALM affects the scale of model parameters such as XL, XW, LD, WD, CJ, and CJSW.

Because the SCALM option is ignored by the LEVEL 40 model, LEVEL 40 models can be mixed in a simulation with other models in which the SCALM is set.

In general, netlists for Star-Hspice should be made as standard as possible. Also, it is best to convert L and W to meters scale instead of microns scale, so that the netlist can be used without the OPTION SCALE=1E-6. If these recommendations are followed, then a system-level Star-Hspice user can use I/O subcircuits from different vendors in one simulation.

Noise Model

The LEVEL 40 model uses the standard NLEV=0 noise model inherited from Star-Hspice.

Element DELVTO

DELVTO and DTEMP on the element line can be used with LEVEL 40.

Star-Hspice Model and Element Statement Example

```
.MODEL nch nmos LEVEL=40 UO=0.4229 VTO=1.645 PHI=1.25 NSS=0
+ NFS=2.248E+21 VMAX=1231
+ THETA=-0.01771 ETA=0.0002703 T1=2.6E-07 T2=0 E1=3.9 E2=0
+ GO=9.206E-15 NU=0 K2=2 CHI=0.5
+ PSI=1E-20 VTIME=0.01 TREF=1.5 CGSO=5.203E-14 CGDO=4.43E-14
+ CSC=0.0001447 RD=5097
+ RS=5097 FREQ=1E+06 DEFF=2.15 TAU=1.64E-07 FEFF=0.5
MCKT 1 2 3 nch L=1e-05 W=4e-05
```
LEVEL 40 Model Equations

In the following equations, model parameters are shown in all capital letters; working variables are in lower case. Model parameters and bias voltages \( v_{gs} \) and \( v_{ds} \) are inputs. \( I_d, g_m, \) and \( g_{ds} \) are the DC outputs, and the gate-to-source capacitance \( C_{gs} \) and the gate-to-drain capacitance \( C_{gd} \) are the AC outputs. Electron charge is \( q \), Boltzmann’s constant is \( k \), and the permittivity of a vacuum is \( \varepsilon_0 \).

Scaling by \text{SCALE} has been done prior to evaluation of the equations. Scaling by \text{M} is done after evaluation.

The variables \( g_{m_{fi}} \) and \( g_{ds_{fi}} \) are intermediate, not final, quantities.

A complete description of TFT technology and the device physics underlying these equations can be found in the Hewlett-Packard HP IC-CAP manual.

Initially, \( C_{gdi} = 0, C_{gsi} = 0, \phi_{i} = \phi I, v_{to} = V{T}O, u_{o} = UO \)

If \( u_{o} = 0 \) then \( u_{o} = 1 \)

\( c_{fm} \), the dielectric capacitance per unit area, is computed as follows:

\[
\text{If } T1 \neq 0 \text{ and } T2 \neq 0, \text{ then } C_{fm} = \frac{(\varepsilon_0 \cdot E_{1} \cdot E_{2})}{((T2 \cdot E1) + (T1 \cdot E2))}
\]

\[
\text{If } T1 = 0 \text{ and } T2 \neq 0, \text{ then } C_{fm} = \frac{(\varepsilon_0 \cdot E_{2})}{T2}
\]

\[
\text{If } T2 = 0 \text{ and } T1 \neq 0, \text{ then } C_{fm} = \frac{(\varepsilon_0 \cdot E_{1})}{T1}
\]

\[
kp = u_o \cdot C_{fm} \cdot 10^{-4}
\]

\( TEMP \) is the Star-Hspice device simulation temperature, specified in \( ^{\circ}C \), but converted to \( ^{\circ}K \) internally for the evaluation of these equations.

\[
v_{t} = \frac{(k \cdot TEMP)}{q}
\]

\[
eg_{g} = (2 \cdot 10^{4} \cdot (TEMP - 312)) + 1.4
\]
LEVEL 40 HP a-Si TFT Model

Selecting MOSFET Models: LEVEL 1-40

\[
vto = vto + (DELVTO_{\text{model} \cdot \text{type}}) + (DELVTO_{\text{element} \cdot \text{type}})
\]
\[
vbi = vto - DELVTO\cdot() + (DELVTO_{\text{element} \cdot \text{type}})
\]
\[
ratio = \frac{\text{TEMP}}{312}
\]

If \( VTIME \leq 1 \), then \( uo = uo \cdot (ratio^{TREF}) \) and \( kp = kp \cdot (ratio^{TREF}) \).

Note: \( TREF \) is the LEVEL 40 model parameter \( TREF \), which is an exponent in temperature adjustment equations. It is not the reference temperature of this device model.

\[
vfb = vto - (0.5 \cdot PHI) + (0.5 \cdot (1.4 - eg))
\]
\[
vbi = vfb + (0.5 + PHI \cdot ratio)
\]
\[
vto = vbi \text{ (printback definition)}
\]
\[
phi = phi \cdot ratio \text{ (printback definition)}
\]
\[
vfb = vbi - phi \text{ (printback definition)}
\]
\[
\text{vdsat} = 0
\]
\[
\beta = kp \cdot W \cdot L
\]
\[
\text{vth} = vbi + (ETA \cdot \text{vds})
\]

If \( NU \neq 0 \) and \( K2 \neq 0 \) and \( PSI \neq 0 \) and \( VTIME > 1 \), then:
\[
\text{vth} = \text{vth} + f(vgs, vds, NU, K2, PSI, CHI, VTIME, TEMP)
\]
\[
\text{von} = \text{vth}
\]

If \( NFS \neq 0 \), then:
\[
xn = 1 + \left( \frac{(q \cdot NFS \cdot 10^4 \cdot W \cdot L)}{Cfm} \right)
\]
\[
\text{von} = f(\text{vth}, (\text{vt} \cdot xn))
\]
Cutoff Region (NFS = 0, \(vgs \leq von\))

If \(NFS = 0\) and \(vgs \leq von\), then:

\[C_{gdi} = 0\]

\[C_{gsi} = 0\]

\[I_{ds} = GO \cdot f(vgs, (DEFF \cdot vds))\]

\[g_m = GO\]

\[g_{ds} = GO \cdot DEFF\]

Noncutoff Region (NFS ≠ 0)

If \(vgs > von\), then:

\[vgs_x = vgs\]

If \(vgs \leq von\), then:

\[vgs_x = von\]

Mobility modulation by \(vgs\):

\[\eta_{eff} = f(ao, \eta, vgs, \Theta)\]

If \(V_{MAX} > 0\), then:

\[v_{ds} = \frac{L \cdot V_{MAX}}{\eta_{eff}}\]

\[v_{dsat} = (vgs_x - v_{th}) + v_{ds} - \sqrt{((vgs_x - v_{th})^2 + v_{ds}^2)}\]

\[C_{fmlw} = \frac{(C_{fm} \cdot CSC)}{(C_{fm} + CSC)} \cdot L \cdot W\]

\(C_{fmlw}\) is the series combination of the dielectric and space charge capacitance of the MIS structure.
If $v_{ds} < v_{dsat}$, then:

$$v_{dsx} = v_{ds}$$

$$\varepsilon_{psfm} = C_{fm} \cdot \frac{(T_2 + T_1)}{\varepsilon_0}$$

$\varepsilon_{psfm}$ is the effective equivalent dielectric constant of the insulator layers.

$$fval = 0.8 + \left( \frac{\varepsilon_{psfm} - 0.8}{1 + (2 \cdot \pi \cdot FREQ \cdot TAU)^2} \right)$$

$$C_{gdi} = f(C_{fmlw} \cdot f(efm, 0.8) \cdot (\exp(fval, FEFF, v_{gs} - v_{th} - v_{ds})))$$

$$C_{gsi} = f(C_{fmlw} \cdot f(efm, 0.8) \cdot (\exp(fval, FEFF, (v_{gs} - v_{th}), v_{ds})))$$

Otherwise, $v_{ds} \geq v_{dsat}$:

$$v_{dsx} = v_{dsat}$$

$$C_{gdi} = C_{fmlw}$$

$$C_{gsi} = \frac{C_{fmlw}}{2}$$

If $v_{dsx} \neq 0$, then:

$$c_{dnom} = v_{dsx} \cdot \left( v_{gsx} - v_{th} - \frac{v_{dsx}}{2} \right)$$

Normalized drain current:

$$g_{mft} = v_{dsx}$$

$$g_{dsft} = v_{gsx} - v_{th} - v_{dsx}$$

$$c_{d1} = \beta \cdot c_{dnom}$$
Drain current without velocity saturation effect:

\[ \beta = \beta \cdot f_{\text{gate}} \]

\[ i_{\text{drain}} = \beta \cdot c_{\text{norm}} \]

\[ g_{\text{m}_{\text{f}}t} = (\beta \cdot g_{\text{m}_{\text{f}}}) + (dfg_{\text{dvg}} \cdot cd_{1}) \]

Velocity saturation factor—If \( V_{\text{MAX}} \neq 0 \), then:

\[ f_{\text{drain}} = \frac{1}{1 + \left( \frac{v_{\text{dsx}}}{v_{\text{dsc}}} \right)} \]

\[ df_{\text{ddvg}} = -df_{\text{gvg}} \cdot \frac{(f_{\text{drain}}^2) \cdot v_{\text{dsx}}}{(v_{\text{dsc}} \cdot f_{\text{gate}})} \]

\[ df_{\text{ddvd}} = \frac{-f_{\text{drain}}^2}{v_{\text{dsc}}} \]

Strong inversion current:

\[ g_{\text{m}_{\text{f}}t} = (f_{\text{drain}} \cdot g_{\text{m}_{\text{f}}}) + (df_{\text{ddvg}} \cdot i_{\text{drain}}) \]

\[ g_{\text{ds}_{\text{f}}t} = (f_{\text{drain}} \cdot g_{\text{ds}_{\text{f}}}) + (df_{\text{ddvd}} \cdot i_{\text{drain}}) \]

\[ i_{\text{drain}} = f_{\text{drain}} \cdot i_{\text{drain}} \]

\[ \beta = \beta \cdot f_{\text{drain}} \]

\[ I_{\text{ds}} = i_{\text{drain}} \cdot f(GO, v_{\text{gs}}, DEFF, v_{\text{ds}}) \]

\[ g_{\text{m}} = f(g_{\text{m}_{\text{f}}}, GO) \]

\[ g_{\text{ds}} = f(g_{\text{ds}_{\text{f}}}, GO, DEFF) \]
Weak inversion current—if $v_{gs} < v_{on}$, then:

$$i_{drain} = i_{drain} \cdot \exp\left(\frac{(v_{gs} - v_{on})}{(v_{t} \cdot x_{n})}\right)$$

$$i_{ds} = i_{drain} + f(GO, v_{gs}, DEFF, v_{ds})$$

$$g_{m_{f}} = \frac{i_{drain}}{(v_{t} \cdot x_{n})}$$

$$g_{m} = f(g_{m_{f}}, GO)$$

$$g_{ds_{f}} = g_{ds_{f}} \cdot \exp\left(\frac{(v_{gs} - v_{on})}{(v_{t} \cdot x_{n})}\right)$$

$$g_{ds} = g_{ds_{f}} + f(GO, DEFF)$$

$v_{dsx} = 0$:

$$i_{ds} = f(GO \cdot v_{gs}, DEFF, v_{ds})$$

$$g_{m} = GO$$

$$g_{ds_{f}} = \beta \cdot (v_{gsx} - v_{th})$$

If $NFS \neq 0$ and $v_{gs} < v_{on}$, then

$$g_{ds_{f}} = g_{ds_{f}} \cdot \exp\left(\frac{(v_{gs} - v_{on})}{(v_{t} \cdot x_{n})}\right)$$

$$g_{ds} = f(g_{ds_{f}}, GO, DEFF)$$

$C_{gd}$, $C_{gs}$

$$C_{gd} = C_{gdi} + CGDO$$

$$C_{gs} = C_{gsi} + CGSO$$
LEVEL 40 Model Topology

Figure 21-12 shows the topology of the LEVEL 40 model.

Figure 21-12: LEVEL 40 HP a-Si TFT Topology
Comparing MOS Models

This section reviews the history, motivation, strengths and weaknesses of the most commonly used MOS models in Star-Hspice:

- LEVEL 2    SPICE LEVEL 2
- LEVEL 3    SPICE LEVEL 3
- LEVEL 13    BSIM1
- LEVEL 28    Avant! proprietary model, based on BSIM1
- LEVEL 39    SIM2

History and Motivation

This section describes the history of and motivation for using MOS models in Star-Hspice.

Star-Hspice Model Enhancements

Avant! modified the standard SPICE models to satisfy the needs of customers. The modifications are in the areas of:

- Drawn dimensions with corrections for photolithography and diffusion
- Corrections for optical shrink
- Model-independent process variation parameters
- Uniform subthreshold equations
- Charge-conserving capacitance equations
- Impact ionization with selectable source/bulk partitioning of the excess drain current
- Enhanced temperature relationships
LEVEL 2

The LEVEL 2 model is an enhanced Grove equation. It is the most common of MOS equations in all simulators.

The basic current equation with the 3/2-power terms was developed by Ihantola and Moll in 1964. Channel length modulation was added by Reddi and Sah in 1965. The vertical field reduction was added by Crawford in 1967. The ECRIT parameter was added by Klassen in 1978.

LEVEL 3

The LEVEL 3 model was developed by Liu in 1981. It is computationally more efficient, replacing the 3/2-power terms with a first-order Taylor expansion. The drain-induced barrier lowering effect (ETA parameter) was added.

The LEVEL 3 models is impressively physical, modeling two-dimensional effects based on junction depth and depletion depths.

LEVEL 13 - BSIM

The BSIM1 model was developed by Sheu, Scharfetter, Poon and Hu at Berkeley in 1984, for higher accuracy modeling of short-channel devices. The approach is empirical rather than physical. It uses polynomials frequently. This makes it easier to write a parameter extraction program, but the polynomials often behave badly. For example, a quadratic function of VDS is used for mobility. Parameters specify the values at VDS=0 and 5 and the slope at VDS=5; unfortunately, values that look reasonable can produce a quadratic that is non-monotonic, giving a GDS<0 problem.

The Star-Hspice implementation of BSIM1 as LEVEL 13 removed discontinuities in the current function, added temperature parameters, and added diode and capacitance models consistent with other models. The Berkeley version did not include temperature parameters.
LEVEL 28

LEVEL 28 is a proprietary Star-Hspice model for submicron devices, designed to fix the following problems in BSIM1:

- Negative GDS
- Bad behavior of some polynomial expressions
- A kink in GM at threshold

LEVEL 28 is based on BSIM1, but some of the parameters are quite different. A BSIM1 parameter set cannot be used as a LEVEL 28 model. The LEVEL 28 model is designed for optimization; there is no simple extraction program. It has proven stable for automated model parameter generation.

Optimization of LEVEL 28 models to IDS, GDS, GM data is accomplished routinely by Avant!

LEVEL 39

The BSIM2 model was developed by Duster, Jeng, Ko, and Hu, and released in SPICE3 in 1991. It is designed for deep submicron devices. It uses a cubic spline to give smooth weak inversion transition and has many additional parameters for improved accuracy. The GDS transition at VDSAT is markedly smoother than in BSIM1.

Future for Model Developments

This sequence of models shows a trend towards empirical rather than physical models, and an ever-increasing number of parameters. It is unfortunate to lose contact with the physics, but it can be unavoidable, because the physics has become less universal. Short-channel devices are much more sensitive to the detail of the process. I-V curves from different manufacturers show qualitative differences in the shape of the curves. Therefore, the models need to be very flexible, requiring a large number of empirical parameters.
Model Equation Evaluation Criteria

This section describes the following aspects of the model equations:

- Potential for good fit to data
- Ease of fitting to data
- Robustness and convergence properties
- Behavior follows actual devices in all circuit conditions
- Ability to simulate process variation
- Gate capacitance modeling

Some of these aspects depend on general Star-Hspice features that are the same for all levels. Others result in simple objective measures for comparing the levels. These measures are summarized in the “Comparison of Star-Hspice Parameters with UCB SPICE 2 and 3” on page 21-124.

Potential for Good Fit to Data

Generally, the model with the largest number of parameters has the potential to give the best fit. For the purpose of comparing the models, the number of parameters are counted in two ways.

Measure: Number of Parameters

Only the drain current parameters are counted, not the diode or series resistance, nor gate capacitance and impact ionization parameters, since these are almost the same for all levels.

LEVEL 2: VTO, PHI, GAMMA, XJ, DELTA, UO, ECRIT, UCRIT, UTRA, UEXP, NSUB, LAMBDA, NFS (total=13).

LEVEL 3: VTO, PHI, GAMMA, XJ, DELTA, ETA, UO, THETA, VMAX, NSUB, KAPPA, NFS (total=12).

LEVEL 13: VFB0, PHI0, K1, K2, ETA0, X2E, X3E, MUZ, X2M, X3MS, MUS, X2MS, U00, X2U0, U1, X2U1, X3U1, N0, ND0, NB0, plus L- and W- variation parameters (total = 20*3 = 60).
LEVEL 28: similar to LEVEL 13, minus MUS, X2MS, plus X33M, WFAC, WFACU (total = 21*3 = 63).

LEVEL 39: VGHIGH, VGLOW, VFB, K1, K2, ETA0, ETAB, MU0, MU0B, MUS0, MUSB, MU20, MU2B, MU2G, MU30, MU3B, MU40, MU4B, MU4G, UA0, UAB, UB0, UBB, U10, U1B, U1D, N0, NB, ND, plus L- and W- parameters (total = 33*3=99).

**Measure: Minimal Number of Parameters**

The minimal number of parameters is defined as the subset of the above set of parameters that would normally be needed to fit a specific W/L device. For LEVEL 2, 3 DELTA is dropped, which is a W-effect parameter. For LEVEL 13 and 28, the L- and W- terms are dropped, as are X2E, X3E, ND0, which are second-order effects. For LEVEL 39, ETAB, MU40, MU4B, MU4G, ND are dropped. The resulting minimal parameter counts for the five models are LEVEL 2=12, LEVEL 3=11, LEVEL 13=17, LEVEL 28=18, and LEVEL 39=28.

**Ease of Fit to Data**

Generally, the larger the “minimal number of parameters”, the more time needs to be spent fitting the data. The systematic L and W effect parameters of LEVEL 13, 28, and 39 makes fitting easier because optimization can be done to individual W/L devices. Then the final model parameters, with L and W terms, can be calculated from the individual models. On the other hand, the more physical parameters of LEVEL 2 and 3 are helpful because it is easier to predict the value from a knowledge of the process, before fitting to I-V data. Examples of physical parameters are junction depths and doping concentrations.
Measure: Physical Percentage of Parameters

Starting with the minimal set of parameters, the percentage that are physical are calculated. For LEVEL 2—PHI, XJ, UO, ECRIT, NSUB, and NFS are physical, while VTO, GAMMA, UCRIT, UTRA, UEXP, LAMBDA are empirical, which gives 50% physical parameters. For LEVEL 3—PHI, XJ, UO, VMAX, NSUB, NFS are physical, which gives 55%. For LEVELs 13, 28, and 39—only PHI0 and MUZ are physical, giving 12%, 11%, and 7% physical parameters, respectively.

Robustness and Convergence Properties

A discontinuity in the derivatives GM, GDS, GMBS can cause convergence problems. Also, since real devices have continuous derivatives, a discontinuity leads to a large inaccuracy in the derivatives near that region. This can be annoying to an analog designer looking at a plot of gain versus bias, for example. The most common important discontinuities are GDS at vds=vdsat, and GM at vgs=vth. The LEVEL 2 and 3 models have these discontinuities, while the LEVEL 13, 28, and 39 models do not.

However, the LEVEL 13 model (BSIM1) often produces a negative GDS, which is obviously inaccurate, and causes oscillation, which can lead to convergence failure or a “timestep too small” error. It is possible for a LEVEL 13 model to avoid negative GDS, but it depends on complex relationships between the parameters MUZ, X2M, MUS, X2MS, X3MS, U1, X2U1, X3U1. Usually, a negative GDS can be removed by setting X3MS=0, but this lowers the accuracy of the model in the linear region. The LEVEL 39 (BSIM2) model also is capable of producing negative GDS unless you select parameters carefully. The LEVEL 28 model does not give negative GDS.

The BSIM1 model has a continuous GM at vgs=vth, but a plot of GM/IDS versus VGS shows a kink, while data from real devices is monotonic. This kink is annoying to analog designers working with devices in the weak and medium inversion region. LEVEL 28 and 39 have solved this problem, at the cost of additional parameters.
There are three more important measures, as follows:

**Measure: Continuous Derivatives**  
LEVELs 2 and 3 fail. LEVELs 13, 28, and 39 pass.

**Measure: Positive GDS**  
LEVELs 13 and 39 fail. LEVELs 2, 3, and 28 pass.

**Measure: Monotonic GM/IDS in weak inversion**  
LEVELs 2, 3, and 13 fail. LEVELs 28 and 39 pass.

**Behavior Follows Actual Devices In All Circuit Conditions**

A model can give a very good fit to IDS data in the normal operating region and still fail to be useful for simulating some circuits.

The first criterion of this type is that the model should have good temperature dependence. Star-Hspice provides temperature-dependence parameters for threshold voltage and mobility for all levels. The LEVEL 13, 28 and 39 models also have an FEX parameter that controls VDSAT variation with temperature.

The next most important criterion is that the model should have subthreshold current to provide accurate analog simulation. Even for digital circuits it aids in convergence. Fortunately, all of these models have subthreshold current.

Impact ionization causes a drain-to-bulk current that has a strong effect on cascode circuits. Star-Hspice provides parameters ALPHA and VCR for this current, which can be used for all levels.

The BSIM2 model has a more complex impact ionization model, with parameters AI0, AIB, BI0, BIB, but in the Berkeley SPICE3 release this current was all assigned to drain-to-source current, IDS. Using the Star-Hspice parameters ALPHA and VCR, the impact ionization current is assigned to IDB, which is essential for cascode simulation. The Star-Hspice parameter IIRAT allows the model to divide the current between IDS and IDB, if needed.
Ability to Simulate Process Variation

Usually, full model parameter extraction or optimization is only done on a small number of test wafers. Statistical data on process variation is gathered by in-fabrication measurements (for example, TOX) and simple electrical measurements (for example, VT), made on a large number of wafers. This statistical data gives variances that are used to simulate process variation, using a worst-case, Monte-Carlo, or Taguchi methodology.

In order to do this simulation, models must be modified to take into account variations in TOX, thresholds, line widths, and sheet resistance. In Star-Hspice, we have made the different levels similar in their use of these parameters. All of the models discussed here accept the following parameters: TOX, DELVTO, XL, XW, RSH. The DELVTO model parameter shifts the threshold.

For the LEVEL 2 and 3 models, setting DELVTO=0.1 is equivalent to adding 0.1 to VTO; for the LEVEL 13, 28, 39 models, it is equivalent to adding 0.1 to VFB0. The parameters XL and XW represent line width variation. The equation for effective channel length is:

\[ L_{eff} = L + XL - 2 \cdot LD \]

The Berkeley BSIM1 and BSIM2 models use \( L_{eff} = L - DL \). The DL and DW parameters (DL0, DW0 for BSIM1) are supported in Star-Hspice for compatibility, using XL, LD, XW, WD is recommended instead. In Star-Hspice, the geometry parameters (XL, LD, XW, WD) and the parasitic parameters (CJ, MJ, CJSW, MJSW, RSH) are kept simple and level-independent to use process variation information consistently.

Gate Capacitance Modeling

LEVEL 2 and 3 were released in Berkeley SPICE with the Meyer model for gate capacitance. This model is non-charge-conserving and sets \( dQG/dVD = dQD/dVG \), which is not valid in a real device, although provides an adequate response for most digital simulations. The BSIM1 and BSIM2 models were released from Berkeley with charge-conserving, non-symmetric capacitance models.
In Star-Hspice, several choices of capacitance models are available; the range of choices and the default varies with the model chosen. The default for LEVELs 2 and 3 is still the Meyer model, but you can also select a charge-conserving Ward-Dutton model.

**LEVEL Comparisons**

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>2</th>
<th>3</th>
<th>13</th>
<th>28</th>
<th>39</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of parameters</td>
<td>13</td>
<td>12</td>
<td>60</td>
<td>63</td>
<td>99</td>
</tr>
<tr>
<td>Minimal number of parameters</td>
<td>12</td>
<td>11</td>
<td>17</td>
<td>18</td>
<td>28</td>
</tr>
<tr>
<td>Physical parameters</td>
<td>50%</td>
<td>55%</td>
<td>12%</td>
<td>11%</td>
<td>7%</td>
</tr>
<tr>
<td>Continuous derivatives</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Positive GDS</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Monotonic GM/IDS</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Outline of Optimization Procedure**

1. Extract XL, LD, XW, WD, TOX, RSH, CGSO, CGDO, CGBO, CJ, MJ, CJSW, MJSW from resistor and capacitor data, and plots of Beta vs. W, L.
2. For each W/L device,
   a. Extract VT versus VBS from IDS vs. VGS data.
   b. Calculate ETA from log(IDS) vs. VGS plots at VDS=0.1, 5.0.
   c. Fit VT parameters to the VT vs. VBS data.
   d. Optimize the rest of the parameters, except L and W sensitivity parameters, to IDS, GDS, GM vs. VGS, VDS, VBS data.
3. For each W/L device, calculate L and W sensitivity parameters from the optimized parameters of nearby devices.
4. Fit the models together into one model using the Star-Hspice Lmin, Lmax, Wmin, Wmax feature.
Examples of Data Fitting

The following plots show fits of LEVELs 2, 3, 13, 28, 39 to data from a submicron device, fabricated by a modern CMOS process. All of the models were optimized to the same data. Similar optimization files were used, optimizing different parameters. The Star-Hspice impact ionization model, with parameters ALPHA, VCR, was used in all models except LEVEL 39, which has its own impact ionization parameters.

The problem of negative GDS in LEVEL 13 was avoided by improved optimization of parameter values, but the GDS discontinuity in LEVEL 3 and the GM discontinuity in LEVEL 2 could not be avoided.

Model versus data plots are presented for drain and gate sweeps. These are followed by close-up plots of the models with small step size to show GM and GDS problems with the individual levels.

LEVEL 28, 2, 3 - Ids Model vs. Data
- Ids vs. Vds at Vgs=1, 2, 3, 4, 5, Vbs=0
- Fits to IDS only (not GDS and GM) would have looked better for these plots, but would not have been acceptable for analog design.

Figure 21-13: LEVEL 2 Ids Model vs. Data Curves
Figure 21-14: LEVEL 28 Ids Model vs. Data Curves

Figure 21-15: LEVEL 3 Ids Model vs. Data Curves
LEVEL 13, 28, 39 - Ids Model vs. Data

Ids vs. Vds at Vgs= 1, 2, 3, 4, 5, Vbs=0

Figure 21-16: LEVEL 13 Ids vs. Vds Curves

Figure 21-17: LEVEL 28 Ids vs. Vds Curves
LEVEL 28, 2, 3 - Gds Model vs. Data

- gds -vs- Vds at Vgs=2, 3, 4, 5, Vbs=0
- This plot shows the inability of LEVEL 2 and 3 to model GDS accurately.

**Figure 21-19: LEVEL 2 gds vs. Vds Curves**
Selecting MOSFET Models: LEVEL 1-40 Comparing MOS Models

Figure 21-20: LEVEL 28 gds vs. Vds Curves

Figure 21-21: LEVEL 3 gds vs. Vds Curves
LEVEL 13, 28, 39 - Gds Model vs. Data

- $gds = v_s$ vs. $V_{ds}$ at $V_{gs}=2, 3, 4, 5, V_{bs}=0$
- These models still have a small change in slope of Gds at $V_{dsat}$, more visible for the LEVEL 13 model than for LEVEL 28 or 39.

Figure 21-22: LEVEL 13 gds vs. Vds Curves

Figure 21-23: LEVEL 28 gds vs. Vds Curves
Figure 21-24: LEVEL 39 gds vs. Vds Curves

LEVEL 2, 3, 28 - Ids Model vs. Data
Ids -vs- Vgs at Vds=0.1, Vgs =0, -1, -2, -3, -4

Figure 21-25: LEVEL 2 Ids vs. Vgs Curves
Figure 21-26: LEVEL 28 Ids vs. Vgs Curves

Figure 21-27: LEVEL 3 Ids vs. Vgs Curves
LEVEL 13, 28, 39 - Ids Model vs. Data

Ids vs. Vgs at Vds=0.1, Vbs =0, -1, -2, -3, -4

**Figure 21-28: LEVEL 13 Ids vs. Vgs Curves**

![LEVEL 13 Ids vs. Vgs Curves](image)

**Figure 21-29: LEVEL 28 Ids vs. Vgs Curves**

![LEVEL 28 Ids vs. Vgs Curves](image)
LEVEL 2, 3, 28 - Gm/Ids Model vs. Data

- $g_{m/Ids}$ vs. $V_{gs}$ at $V_{ds}=0.1$, $V_{bs}=0$, $-2$
- The LEVEL 2 and 3 models have spikes at $V_{gs}=V_{th}$. The data, and the LEVEL 28 model, is monotonic decreasing.

Figure 21-31: LEVEL 2 $g_{m/Ids}$ vs. $V_{gs}$ Curves
Figure 21-32: LEVEL 28 gm/Ids vs. Vgs Curves

Figure 21-33: LEVEL 3 gm/Ids vs. Vgs Curves
LEVEL 13, 28, 39 - Gm/Ids Model vs. Data

- **gm/Ids -vs- Vgs** at Vds=0.1, Vbs =0, -2
- LEVEL 13 has a kink at Vth, which is not visible at this resolution. LEVEL 28 and 39 are monotonic.

**Figure 21-34: LEVEL 13 gm/Ids vs. Vgs Curves**

**Figure 21-35: LEVEL 28 gm/Ids vs. Vgs Curves**
Selecting MOSFET Models: LEVEL 1-40

Comparing MOS Models

Figure 21-36: LEVEL 39 gm/Ids vs. Vgs Curves

Gds vs. Vds at Vgs=4, Vbs=0
This plot shows the behavior of gds at the linear to saturation transition. The LEVEL 3 model has a gds discontinuity.

Figure 21-37: LEVELs 2, 3, 28 gds vs. Vds Curves
Comparing MOS Models

Figure 21-38: LEVELs 13, 28, 39 gds vs. Vds Curves

This plot shows a gm discontinuity in the LEVEL 2 model, related to parameters UCRIT and UEXP.

Figure 21-39: LEVEL 2 gm/Ids vs. Vgs Curves
Selecting MOSFET Models: LEVEL 1-40

Comparing MOS Models

Figure 21-40: LEVEL 28 gm/Ids vs. Vgs Curves

Gm/Ids vs. Vgs at Vds=0.1, Vbs=0

This plot shows the ratio gm/Ids in the weak inversion transition region. The LEVEL 2, 3, and 13 models have kinks near threshold, while LEVELs 28 and 39 are monotonic.

Figure 21-41: LEVELs 2, 3, 28 gm/Ids vs. Vgs Curves
References


4. Fargher, H. E. and Mole, P. J. The Implementation Of A 3 Terminal SOSFET Model In SPICE For Circuit Simulation. GEC VLSI Research Laboratory, MOS1 Division.


The MOSFET models described in this chapter are the most currently developed and widely used models. Star-Hspice has introduced LEVELs that are compatible with models developed by Berkeley, The University of Florida, Rensselaer Polytechnic Institute, and others.

This chapter lists the various MOSFET models (LEVEL 47 to 62), and provides the specifications for each model. The following topics are covered in this chapter:

- LEVEL 47 BSIM3 Version 2 MOS Model
- LEVELs 49 and 53 BSIM3v3 MOS Models
- LEVEL 50 Philips MOS9 Model
- LEVEL 54 BSIM4.0 Model
- LEVEL 55 EPFL-EKV MOSFET Model
- LEVEL 57 UC Berkeley BSIM3-SOI Model
- LEVEL 58 University of Florida SOI Model
- LEVEL 59 UC Berkeley BSIM3-SOI FD Model
- LEVEL 60 UC Berkeley BSIM3-SOI DD Model
- LEVEL 61 RPI a-Si TFT Model
- LEVEL 62 RPI Poli-Si TFT Model

For information on MOSFET Models LEVELs 1 to 40, see Chapter 21, “Selecting MOSFET Models: LEVEL 1-40”.
LEVEL 47 BSIM3 Version 2 MOS Model

The BSIM3 version 2.0 MOS model from UC Berkeley is available as the LEVEL 47 Star-Hspice model.

LEVEL 47 Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTH0</td>
<td>V</td>
<td>0.7</td>
<td>Threshold voltage of long channel at $V_{bs} = 0$ and small $V_{ds}$ (0.7 for n-channel, - 0.7 for p-channel)</td>
</tr>
<tr>
<td>K1</td>
<td>V^{1/2}</td>
<td>0.53</td>
<td>First-order body effect coefficient</td>
</tr>
<tr>
<td>K2</td>
<td>-0.0186</td>
<td></td>
<td>Second-order body effect coefficient</td>
</tr>
<tr>
<td>K3</td>
<td>80.0</td>
<td></td>
<td>Narrow width effect coefficient</td>
</tr>
<tr>
<td>K3B</td>
<td>1/V</td>
<td>0</td>
<td>Body width coefficient of narrow width effect</td>
</tr>
<tr>
<td>KT1</td>
<td>V</td>
<td>-0.11</td>
<td>Temperature coefficient for threshold voltage</td>
</tr>
<tr>
<td>KT2</td>
<td></td>
<td>0.022</td>
<td>Body bias coefficient of threshold temperature effect</td>
</tr>
<tr>
<td>GAMMA1</td>
<td>V^{1/2}</td>
<td>See “LEVEL 47 Model Equations”</td>
<td>Body effect coefficient, near interface</td>
</tr>
<tr>
<td>GAMMA2</td>
<td>V^{1/2}</td>
<td>See “LEVEL 47 Model Equations”</td>
<td>Body effect coefficient in the bulk</td>
</tr>
<tr>
<td>W0</td>
<td>m</td>
<td>2.5e-6</td>
<td>Narrow width effect coefficient</td>
</tr>
<tr>
<td>NLX</td>
<td>m</td>
<td>1.74e-7</td>
<td>Lateral nonuniform doping along channel</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>150e-10</td>
<td>Gate oxide thickness</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>-----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>XJ</td>
<td>m</td>
<td>0.15e-6</td>
<td>Junction depth</td>
</tr>
<tr>
<td>DL</td>
<td>m</td>
<td>0.0</td>
<td>Channel length reduction on one side (multiplied by SCALM)</td>
</tr>
<tr>
<td>DW</td>
<td>m</td>
<td>0.0</td>
<td>Channel width reduction on one side (multiplied by SCALM)</td>
</tr>
<tr>
<td>NPEAK</td>
<td>cm⁻³</td>
<td>1.7e17</td>
<td>Peak doping concentration near interface</td>
</tr>
<tr>
<td>NSUB</td>
<td>cm⁻³</td>
<td>6.0e16</td>
<td>Substrate doping concentration</td>
</tr>
<tr>
<td>PHI</td>
<td>V</td>
<td>See &quot;LEVEL 47 Model Equations&quot;</td>
<td>Surface potential under strong inversion</td>
</tr>
<tr>
<td>XT</td>
<td>m</td>
<td>1.55e-7</td>
<td>Doping depth</td>
</tr>
<tr>
<td>VBM</td>
<td>V</td>
<td>-5.0</td>
<td>Maximum substrate bias</td>
</tr>
<tr>
<td>VBX</td>
<td>V</td>
<td>See &quot;LEVEL 47 Model Equations&quot;</td>
<td>$V_{bs}$ at which the depletion width equals $XT$</td>
</tr>
<tr>
<td>DVT0</td>
<td></td>
<td>2.2</td>
<td>Short-channel effect coefficient 0</td>
</tr>
<tr>
<td>DVT1</td>
<td></td>
<td>0.53</td>
<td>Short-channel effect coefficient 1</td>
</tr>
<tr>
<td>DVT2</td>
<td>1/V</td>
<td>-0.032</td>
<td>Short-channel effect coefficient 2</td>
</tr>
<tr>
<td>U0</td>
<td>m²/V sec</td>
<td>0.067</td>
<td>Low field mobility at $T = T_{REF}$</td>
</tr>
<tr>
<td></td>
<td>(see Note 8)</td>
<td></td>
<td>(0.067 for n-channel, 0.025 for p-channel)</td>
</tr>
<tr>
<td>UA</td>
<td>m/V</td>
<td>2.25e-9</td>
<td>First-order mobility degradation coefficient</td>
</tr>
<tr>
<td>UA1</td>
<td>m/V</td>
<td>4.31e-9</td>
<td>Temperature coefficient of $UA$</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UB</td>
<td>m²/V²</td>
<td>5.87e-19</td>
<td>Second-order mobility degradation coefficient</td>
</tr>
<tr>
<td>UB1</td>
<td>m²/V²</td>
<td>-7.61e-18</td>
<td>Temperature coefficient of UB</td>
</tr>
<tr>
<td>UC</td>
<td>1/V</td>
<td>0.0465</td>
<td>Body bias sensitivity coefficient of mobility</td>
</tr>
<tr>
<td>UC1</td>
<td>1/V</td>
<td>-0.056</td>
<td>Temperature coefficient of UC</td>
</tr>
<tr>
<td>VSAT</td>
<td>cm/sec</td>
<td>8e6</td>
<td>Saturation velocity of carrier at T = TREF</td>
</tr>
<tr>
<td>AT</td>
<td>m/sec</td>
<td>3.3e4</td>
<td>Temperature coefficient of VSAT</td>
</tr>
<tr>
<td>RDSW</td>
<td>ohm·µm/µm</td>
<td>0.0</td>
<td>Source drain resistance per unit width</td>
</tr>
<tr>
<td>RDS0</td>
<td>ohm</td>
<td>0.0</td>
<td>Source drain contact resistance</td>
</tr>
<tr>
<td>LDD</td>
<td>m</td>
<td>0.0</td>
<td>Total length of LDD region</td>
</tr>
<tr>
<td>ETA</td>
<td></td>
<td>0.3</td>
<td>Coefficient of drain voltage reduction</td>
</tr>
<tr>
<td>ETA0</td>
<td></td>
<td>0.08</td>
<td>Subthreshold region DIBL (Drain Induced Barrier Lowering) coefficient</td>
</tr>
<tr>
<td>ETAB</td>
<td>1/V</td>
<td>-0.07</td>
<td>Subthreshold region DIBL coefficient</td>
</tr>
<tr>
<td>EM</td>
<td>V/m</td>
<td>4.1e7</td>
<td>Electrical field in channel above which hot carrier effect dominates</td>
</tr>
<tr>
<td>NFACTOR</td>
<td></td>
<td>1.0</td>
<td>Subthreshold region swing</td>
</tr>
<tr>
<td>VOFF</td>
<td>V</td>
<td>-0.11</td>
<td>Offset voltage in subthreshold region</td>
</tr>
<tr>
<td>LITL</td>
<td>m</td>
<td></td>
<td>Characteristic length. The default is</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[ LITL = \left( \frac{\varepsilon_s T_{ox} X_j}{\varepsilon_{ox}} \right)^{1/2} ]</td>
</tr>
<tr>
<td>VGLOW</td>
<td>V</td>
<td>-0.12</td>
<td>Lower bound of the weak-strong inversion transition region</td>
</tr>
<tr>
<td>VGHIGH</td>
<td>V</td>
<td>0.12</td>
<td>Upper bound of the weak-strong inversion transition region</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>---------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>CDSC</td>
<td>F/m²</td>
<td>2.4e-4</td>
<td>Drain/source and channel coupling capacitance</td>
</tr>
<tr>
<td>CDSCB</td>
<td>F/Vm²</td>
<td>0</td>
<td>Body coefficient for CDSC</td>
</tr>
<tr>
<td>CIT</td>
<td>F/m²</td>
<td>0.0</td>
<td>Interface state capacitance</td>
</tr>
<tr>
<td>PCLM</td>
<td>F/m²</td>
<td>1.3</td>
<td>Coefficient of channel length modulation</td>
</tr>
<tr>
<td>PDIBL1</td>
<td></td>
<td>0.39</td>
<td>DIBL (Drain Induced Barrier Lowering) effect coefficient 1</td>
</tr>
<tr>
<td>PDIBL2</td>
<td></td>
<td>0.0086</td>
<td>DIBL effect coefficient 2</td>
</tr>
<tr>
<td>DROUT</td>
<td></td>
<td>0.56</td>
<td>DIBL effect coefficient 3</td>
</tr>
<tr>
<td>DSUB</td>
<td></td>
<td>DROUT</td>
<td>DIBL coefficient in subthreshold region</td>
</tr>
<tr>
<td>PSCBE1</td>
<td>V/m</td>
<td>4.24e8</td>
<td>Substrate current induced body effect exponent 1</td>
</tr>
<tr>
<td>PSCBE2</td>
<td>m/V</td>
<td>1.0e-5</td>
<td>Substrate current induced body effect coefficient 2</td>
</tr>
<tr>
<td>A0</td>
<td></td>
<td>1</td>
<td>Bulk charge effect. The default is 4.4 for PMOS.</td>
</tr>
<tr>
<td>TNOM (TREF)</td>
<td>°C</td>
<td>25</td>
<td>Temperature at which parameters are extracted. This parameter defaults to the option TNOM, which defaults to 25 °C. See 4 and 5 in “Reminders for this Installation,” below.</td>
</tr>
<tr>
<td>SUBTHMOD</td>
<td></td>
<td>2</td>
<td>Subthreshold model selector</td>
</tr>
<tr>
<td>SATMOD</td>
<td></td>
<td>2</td>
<td>Saturation model selector</td>
</tr>
<tr>
<td>KETA</td>
<td>1/V</td>
<td>-0.047</td>
<td>Body bias coefficient of the bulk charge effect</td>
</tr>
<tr>
<td>A1</td>
<td>1/V</td>
<td>0</td>
<td>First nonsaturation factor (0 for NMOS, 0.23 for PMOS)</td>
</tr>
</tbody>
</table>
Using the BSIM3 Version 2 MOS Model in Star-Hspice

The Star-Hspice LEVEL 47 model uses the same model parameters for source/drain diode current, capacitance, and resistance as do the other Star-Hspice MOS levels. The model parameter ACM controls the choice of source/drain equations.

The Star-Hspice LEVEL 47 model also uses the same noise equations as the other levels. The parameter NLEV controls the choice of noise equations.

This model, like all models in Star-Hspice, can be parametrized. This is useful for modeling process skew, either by worst-case corners or by Monte Carlo. For information on worst-case and Monte Carlo analysis, see “Performing Worst Case Analysis” on page 13-8 and “Performing Monte Carlo Analysis” on page 13-14.

Notes:

1. Set LEVEL=47 to identify the model as a BSIM3 model.

2. This model is based on BSIM version 2.0 from UC Berkeley. Code was received from UC Berkeley in July 1994, in the form of SPICE3e2. Changes
announced in a letter from UCB September 13, 1994, have been included. DC sweeps have been checked against SPICE3e2.

3. The default setting for CAPOP is CAPOP=13, which is the BSIM1 charge-conserving capacitance model. The BSIM3 capacitance model has not been installed.

4. The LEVEL 47 model supports the model parameter name TNOM as an alias for TREF. The conventional terminology in Star-Hspice is TREF, which is supported as a model parameter in all Star-Hspice MOS levels. The alternative name TNOM is supported for LEVEL 47, for compatibility with SPICE3.

5. The default room temperature is 25°C in Star-Hspice, but is 27°C in SPICE3. If the BSIM3 model parameters are specified at 27°C, TREF=27 should be added to the model, so that the model parameters is interpreted correctly. It is a matter of choice whether or not to set the nominal simulation temperature to 27, by adding .OPTION TNOM=27 to the netlist. This should be done when testing Star-Hspice versus SPICE3.

6. The default of DERIV is zero, the analytical method. DERIV can be set to 1 for the finite difference method. The analytic derivatives in the SPICE3e2 code are not exact in some regions. Setting DERIV=1 gives more accurate derivatives (GM, GDS, GMBS), but consumes more CPU time.

7. There are three ways for the BSIM3 model to calculate \( V_{th} \):
   - Using \( K1 \) and \( K2 \) values that are user specified
   - Using \( GAMMA1, GAMMA2, VBM, \) and \( VBX \) values entered in the .MODEL statement
   - Using \( NPEAK, NSUB, XT, \) and \( VBM \) values that are user specified

8. The model parameters \( NPEAK \) and \( U0 \) can be entered in meters or centimeters. \( NPEAK \) is converted to \( \text{cm}^{-3} \) as follows: if \( NPEAK \) is greater than \( 1e20 \), it is multiplied by \( 1e-6 \). \( U0 \) is converted to \( \text{m}^2/\text{Vsec} \) as follows: if \( U0 \) is greater than \( 1 \), it is multiplied by \( 1e-4 \). You must enter the parameter \( NSUB \) in \( \text{cm}^{-3} \) units.

9. The specified value of \( VTH0 \) for p-channel in the .MODEL statement should be negative.
10. The default value of $K_T$ is -0.11. The negative sign ensures that the absolute value of threshold decreases with increasing temperature for NMOS and PMOS.

11. Model parameter $L_{ITL}$ is not allowed to go below a minimum value of 1.0e-9 m, to avoid a possible divide by zero error.

12. $V_{SAT}$, after temperature adjustment, is not allowed to go below a minimum value of 1.0e4 m/sec, to assure that it is positive after temperature compensation.

13. There are seven model parameters for accommodating the temperature dependencies of six temperature dependent model variables. They are $K_T$, $K_T2$ for $V_{TH}$, $U_{TE}$ for $U_0$, $A_T$ for $V_{SAT}$, $U_{AI}$ for $U_A$, $U_{BI}$ for $U_B$, and $U_{CI}$ for $U_C$.

14. Set up the conversion of temperature between Star-Hspice and SPICE3 as follows:

   SPICE3: .OPTIONS TEMP=125
   .MODEL NCH NMOS LEVEL=8
      + TNOM =27  ...

   HSPICE: .TEMP 125
   .MODEL NCH NMOS LEVEL=47
      + TREF =27  ...

15. The option SCALM does not affect the parameters unique to this model, but it does affect the common MOS parameters, such as $X_L$, $L_D$, $X_W$, $W_D$, $C_J$, $C_JSW$, $J_S$, and $J_SW$.

16. LEVEL 47 uses the common Star-Hspice MOS parasitic models, specified by ACM.

17. LEVEL 47 uses the common Star-Hspice MOS noise models, specified by NLEV.

18. $DELVTO$ and $DTEMP$ on the element line can be used with LEVEL 47.

19. The impact ionization current determined by the model parameters PSCBE1 and PSCBE2 contributes to the drain-source current; it does not contribute to bulk current.
Leff and Weff Equations for BSIM3 Version 2.0

The standard Star-Hspice equations for Leff and Weff are:

\[
Leff = L + XL - (2 \cdot LD)
\]

\[
Weff = W + XW - (2 \cdot WD)
\]

The UCB SPICE3 equations used for BSIM3 are:

\[
Leff = L - (2 \cdot DL)
\]

\[
Weff = W - (2 \cdot DW)
\]

The units for these parameters are meters, with defaults of zero.

Star-Hspice uses the standard Star-Hspice equation for both cases, and accepting DL(DW) as the value for LD(WD). If both LD(WD) and DL(DW) are specified in an Star-Hspice .MODEL statement, Star-Hspice uses the LD(WD) value.

If LDAC and WDAC are included in the .MODEL statement,

\[
Leff = L+XL-2LDAC, \quad Weff = W+XW-2WDAC
\]

The model uses the values of LD(DL) and WD(DW) to generate defaults for CGSO, CGDO, and CGBO. The values are also used with parameters RS and RD for ACM>0.

Example

The following two models give the same Star-Hspice results:

* HSPICE style:

```plaintext
.MODEL n1 nmos LEVEL=47 XL=0.1e6 LD=0.15e-6
+ SatMod=2 SubthMod=2 BulkMod=1
+ CGSO=0.3e-9 CGDO=0.3e-9 CGBO=0
```

* SPICE3 style:

```plaintext
.MODEL n2 nmos LEVEL=47 LD=0.1e-6
+ SatMod=2 SubthMod=2 BulkMod=1
+ CGSO=0.3e-9 CGDO=0.3e-9 CGBO=0
```
LEVEL 47 Model Equations

The following model equations are based on the source code of BSIM3.

Threshold Voltage

Model Parameters

\[
V_{th0} \quad K1, K2, \phi_s, N_{lx}, K3, W_0, T_{ox}, V_{bi}, D_{vto}, D_{vt1}, D_{vt2}, N_{peak}, N_{sub}, V_1, Y_2, V_{bs}, V_{bm}, V_{bi}, X_r, T_{REF}
\]

\[
V_{th} = V_{th0} + K1(\phi_s - V_{bs} - \phi_s) - K2V_{bs} + K1\left(1 + \frac{N_{lx}}{L_{eff}}\phi_s - V_{bs} - 1\right)\phi_s
\]

\[
+ (K3 + K3B \cdot V_{bs}) \cdot \left(\frac{T_{ox}}{W_{eff} + W_0}\right)\phi_s - \Delta V_{th}
\]

\[
T_{ratio} = \frac{(TEMP + DTEMP + 273.15)}{(TREF + 273.15)}
\]

\[
\Delta V_{th} = \theta_{th}(L_{eff}) \cdot (V_{bi} - \phi_s)
\]

\[
\theta_{th}(L_{eff}) = D_{vto} \left[\exp\left(-\frac{D_{vt1} \cdot L_{eff}}{2l_t}\right) + 2\exp\left(-\frac{D_{vt1} \cdot L_{eff}}{l_t}\right)\right]
\]

\[
l_t = \sqrt{3 \cdot T_{ox} \cdot X_{dep} \cdot (1 + D_{vt2} \cdot V_{bs})}
\]

\[
X_{dep} = \sqrt{\frac{2 \cdot \varepsilon_{si} \cdot (\phi_s - V_{bs})}{q \cdot N_{peak}}}
\]

If \(\phi_s\) is not specified as a model parameter, then

\[
\phi_s = 2 \cdot V_{tm} \cdot ln\left(\frac{N_{peak}}{n_i}\right) \quad (N_{peak} \text{ and } n_i \text{ in } cm^{-3})
\]
If $K_1, K_2$ are not specified as model parameters, then they are calculated as follows:

$$K_1 = Y_2 - 2 \cdot K_2 \cdot \sqrt{\phi_s - V_{bm}}$$

$$K_2 = (Y_1 - Y_2) \cdot \frac{\sqrt{\phi_s - V_{bx} - \phi_s}}{2 \cdot \sqrt{\phi_s - V_{bm} - \phi_s} + V_{bm}}$$

$$Y_1 = \sqrt[2]{\frac{2 \cdot q \cdot \varepsilon_{si} \cdot N_{peak}}{C_{ox}}}$$

$$Y_2 = \sqrt[2]{\frac{2 \cdot q \cdot \varepsilon_{si} \cdot N_{sub}}{C_{ox}}}$$

$$V_{bx} = \phi_s - \left(\frac{q \cdot N_{peak} \cdot X_t^2}{2 \cdot \varepsilon_{si}}\right)$$

If $V_{bi}$ is not specified as a model parameter, then

$$V_{bi} = \frac{k \cdot T}{q} \cdot \ln \left(\frac{1.0e22 \cdot N_{peak}}{n_i^2}\right)$$
Mobility of Carrier

Model Parameters
\[ \mu_0, \ U_a, \ U_b, \ U_c \]
\[ \mu_{\text{eff}} = \frac{\mu_0}{1 + U_a \left( \frac{V_{gs} + V_{th}}{T_{ox}} \right) + U_b \left( \frac{V_{gs} + V_{th}}{T_{ox}} \right)^2 + U_c \cdot V_{bs}} \]

Drain Saturation Voltage

Model Parameters
\[ A_0, \ \nu_{\text{sat}}, \ \chi, \ A_1, \ A_2, \ R_{ds0}, \ R_{dsw} \]

Rds and Pfactor:
\[ R_{ds} = R_{ds0} + R_{dsw}/(1e6 \cdot W_{\text{eff}}) \]
\[ Pfactor = A_1 \cdot V_{gst} + A_2 \quad \text{(if } Pfactor > 1, \text{ it is set to } Pfactor = 1) \]
\[ V_{gst} = V_{gs} - V_{th} \]

Vdsat for the case Rds = 0 and Pfactor = 1:
\[ V_{dsat} = \frac{E_{\text{sat}} \cdot L_{\text{eff}} \cdot V_{gst}}{A_{\text{bulk}} \cdot E_{\text{sat}} \cdot L_{\text{eff}} + V_{gst}} \]

For BULKMOD = 1,
\[ A_{\text{bulk}} = \frac{1 + K1 \cdot A_0 \cdot L_{\text{eff}}}{(L_{\text{eff}} + T1) \cdot T1s \cdot 2} \cdot (1 + KETA \cdot V_{bs}) \]
For BULKMOD = 2,

\[
A_{\text{bulk}} = \left( \frac{K1 \cdot A_0 \cdot L_{\text{eff}}}{(L_{\text{eff}} + T1) \cdot \sqrt{\phi_s} \cdot 2} \right) / (1 + KETA \cdot V_{bs})
\]

\[
T1 = 2 \cdot \sqrt{X_j \cdot X_{\text{dep}}}
\]

For \( V_{bs} \leq 0 \),

\[
T1_s = \sqrt{\phi_s - V_{bs}}
\]

For \( V_{bs} \geq 0 \),

\[
T1_s = \frac{\phi_s \cdot \sqrt{\phi_s}}{\phi_s + \frac{V_{bs}}{2}}
\]

\[
E_{\text{sat}} = 2 \cdot \frac{V_{\text{sat}}}{\mu_{\text{eff}}}
\]

\( V_{dsat} \) for the general case:

\( V_{dsat} \) is the solution of \( Tmpa \cdot V_{dsat} - Tmpb \cdot V_{dsat} + Tmpc = 0 \)

\[
V_{dsat} = \left( Tmpb - \sqrt{\frac{Tmpb^2}{4} - 4 \cdot Tmpa \cdot Tmpc} \right) / (2 \cdot Tmpa)
\]

\[
Tmpa = A_{\text{bulk}} \cdot (A_{\text{bulk}} \cdot W_{\text{eff}} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot R_{ds} - 1 + 1/P\text{factor})
\]

\[
Tmpb = V_{gst} \cdot (2/P\text{factor} - 1) + (A_{\text{bulk}} \cdot E_{\text{sat}} \cdot L_{\text{eff}}) + (3 \cdot A_{\text{bulk}} \cdot V_{gst} \cdot W_{\text{eff}} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot R_{ds})
\]

\[
Tmpc = (V_{gst} \cdot E_{\text{sat}} \cdot L_{\text{eff}}) + (V_{gst}^2 \cdot 2 \cdot W_{\text{eff}} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot R_{ds})
\]
Linear Region

\[ I_{d_{\text{dlin0}}} = \mu_{\text{eff}} \cdot C_{\text{ox}} \cdot \frac{W_{\text{eff}}}{L_{\text{eff}}} \cdot \frac{1}{1 + V_{ds}/(E_{\text{sat}} \cdot L)} \cdot \left( V_{gs} - V_{th} - A_{\text{bulk}} \cdot \frac{V_{ds}}{2} \right) \cdot V_{ds} \]

\[ I_{ds} = \frac{I_{d_{\text{dlin0}}}}{1 + \frac{R_{ds} \cdot I_{d_{\text{dlin0}}}}{V_{ds}}} \]

Saturation Region

Model Parameters

\( \text{litl, } \eta ta, \ L_{dd}, \ E_{m}, \ D_{\text{rout}}, \ P_{\text{clm}}, \ P_{\text{dibl1}}, \ P_{\text{dibl2}}, \ P_{\text{scbe1}}, \ P_{\text{scbe2}} \)

Vasat and Fvag:

\[ V_{\text{asat}} = \frac{E_{\text{sat}} \cdot L_{\text{eff}} + V_{dsat} + 2R_{ds} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot W_{\text{eff}} \cdot \left( V_{g_{\text{sat}}} - \frac{A_{\text{bulk}} \cdot V_{dsat}}{2} \right)}{2/P_{factor} - 1 + R_{ds} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot W_{\text{eff}} \cdot A_{\text{bulk}}} \]

\[ F_{\text{vag}} = 1 + \frac{P_{vag} \cdot V_{g_{\text{sat}}}}{E_{\text{sat}} \cdot L_{\text{eff}}} \]

Early Voltage, satMod = 1:

\[ V_A = V_{\text{asat}} + F_{\text{vag}} \cdot \left( 1 + \frac{L_{dd}}{\text{litl}} \right) \cdot \frac{(A_{\text{bulk}} \cdot E_{\text{sat}} \cdot L_{\text{eff}} + V_{g_{\text{sat}}} - \lambda \cdot (V_{ds} - V_{dsat})) \cdot (V_{ds} - V_{dsat})}{E_{\text{sat}} \cdot \text{litl}} \]

\[ \lambda = \frac{A_{\text{bulk}} \cdot E_{\text{sat}} \cdot L_{\text{eff}} + (V_{g_{\text{sat}}})}{2 \cdot \text{litl} \cdot E_{m}} \]
Early Voltage, satMod = 2:

\[ V_A = V_{asat} + F_{vag} \cdot U_{vds} \cdot \left( \frac{1}{V_{aclm}} + \frac{1}{V_{adibl}} \right)^{-1} \]

\[ U_{vds} = 1 + \eta \cdot \frac{L_{dd}}{l_{itl}} \]

\[ V_{aclm} = \frac{1}{P_{clm}} \cdot \frac{A_{bulk} \cdot E_{sat} \cdot L_{eff} + V_{gsl}}{A_{bulk} \cdot E_{sat} \cdot l_{itl}} \cdot (V_{ds} - V_{dsat}) \]

\[ V_{adibl} = \frac{1}{\theta_{rout}} \cdot \left[ (V_{gs} - V_{th}) \cdot \left( \frac{1}{A_{bulk} \cdot V_{dsat}} + \frac{1}{V_{gsl}} \right)^{-1} \right] \]

\[ \theta_{rout} = P_{dibl1} \left[ \exp \left( \frac{-D_{rout} \cdot L_{eff}}{2 \cdot l_{t}} \right) + 2 \exp \left( \frac{-D_{rout} \cdot L_{eff}}{l_{t}} \right) \right] + P_{dibl2} \]

\[ V_{ahce} = \frac{P_{scbe2} \cdot \exp \left( \frac{-P_{scbel} \cdot l_{itl}}{V_{ds} - V_{dsat}} \right)}{L_{eff}} \]

Drain Current

\[ I_{dsat} = W_{eff} \cdot V_{sat} \cdot C_{ox} \cdot (V_{gs} - V_{th} - A_{bulk} \cdot V_{dsat}) \cdot P_{factor} \]

\[ P_{factor} = A_1 \cdot V_{gsl} + A_2 \]

\[ I_{ds} = I_{dsat} \cdot \left( 1 + \frac{V_{ds} - V_{dsat}}{V_A} \right) \cdot \left( 1 + \frac{V_{ds} - V_{dsat}}{V_{ahce}} \right) \]

Subthreshold Region

Model Parameters

\[ N_{factor}, C_{dsc}, C_{dsbc}, V_{off}, C_{it}, D_{sub}, \eta_a, \eta_b \]
n and DIBL:

\[
n = 1 + \frac{\text{Nfactor} \cdot 1.034 \cdot 10^{-10}}{X_{\text{deg}} \cdot C_{\text{ox}}} + \frac{(C_{\text{dsc}} + C_{\text{dscb}} \cdot V_{bs}) \left[ \exp \left( \frac{-L_{\text{eff}}}{2 \cdot l_t} \right) + 2 \exp \left( \frac{-L_{\text{eff}}}{l_t} \right) \right]}{C_{\text{ox}}} + C_{it}
\]

\[
\text{DIBL} = (\text{eta}_0 + \text{eta}_b \cdot V_{bs}) \cdot \left[ \exp \left( \frac{-D_{\text{sub}} \cdot L_{\text{eff}}}{2 \cdot l_t} \right) + 2 \exp \left( \frac{-D_{\text{sub}} \cdot L_{\text{eff}}}{l_t} \right) \right]
\]

\[
l_0 = \sqrt{\frac{3 \cdot T_{\text{ox}}}{X_{\text{dep0}}}}
\]

\[
X_{\text{dep0}} = \sqrt{\frac{2 \cdot \varepsilon_s \cdot \phi_s}{q \cdot N_{\text{peak}}}}
\]

If subthMod = 0,

\[
I_{ds} = g_m = g_{ds} = g_{mb} = 0
\]

If subthMod = 1,

\[
I_{ds} = \frac{I_{\text{limit}} \cdot I_{\text{exp}}}{I_{\text{limit}} + I_{\text{exp}}} \cdot \left[ 1 - \exp \left( \frac{-V_{ds}}{V_{tm}} \right) \right]
\]

\[
I_{\text{limit}} = \frac{9}{2} \cdot u_0 \cdot \sqrt{\frac{q \cdot \varepsilon_s \cdot N_{\text{peak}}}{2 \cdot \phi_s}} \cdot \frac{W_{\text{eff}}}{L_{\text{eff}}} \cdot V_{\text{tm}}^2
\]

\[
I_{\text{exp}} = u_0 \cdot \sqrt{\frac{q \cdot \varepsilon_s \cdot N_{\text{peak}}}{2 \cdot \phi_s}} \cdot \frac{W_{\text{eff}}}{L_{\text{eff}}} \cdot V_{\text{tm}}^2 \cdot \exp \left( \frac{V_{gs} - V_{th} - V_{off} + \text{DIBL} \cdot V_{ds}}{n \cdot V_{tm}} \right)
\]

If subthMod = 2,

\[
t_{ds} = u_0 \cdot \sqrt{\frac{q \cdot \varepsilon_s \cdot N_{\text{peak}}}{2 \cdot \phi_s}} \cdot \frac{W_{\text{eff}}}{L_{\text{eff}}} \cdot V_{\text{tm}}^2 \cdot \left[ 1 - \exp \left( \frac{-V_{ds}}{V_{tm}} \right) \right] \cdot \exp \left( \frac{V_{gs} - V_{th} - V_{off} + \text{DIBL} \cdot V_{ds}}{n \cdot V_{tm}} \right)
\]
Transition Region (for subthMod = 2 only)

Model Parameters

\[ V_{gshigh} - V_{gslow} \]

\[ I_{ds} = (1-t)^2 \cdot I_{dslow} + 2 \cdot (1-t) \cdot I_{p} + t^2 \cdot I_{dshigh} \]

\[ t = \left( \frac{V_p - V_{gslow}}{V_{gslow} - 2 \cdot V_p + V_{gshigh}} \right) \cdot \left( 1 + \frac{(V_{gslow} - 2 \cdot V_p + V_{gshigh})(V_g - V_{th} - V_{gslow})}{(V_p - V_{gslow})^2} - 1 \right) \]

\[ V_p = \frac{(g_{mhigh} \cdot V_{gshigh} - g_{mlow} \cdot V_{gslow}) - (I_{dshigh} - I_{dslow})}{g_{mhigh} - g_{mlow}} \]

\[ I_p = I_{dslow} + g_{mlow} \cdot (V_p - V_{gslow}) \]

Temperature Compensation

Model Parameters

\[ A_p, U_{a1}, U_{b1}, U_{c1}, KT1, KT2, UTE \]

\[ V_{th(temp)} = V_{th(tref)} + (KT1 + KT2 \cdot V_{bs}) \cdot (T_{ratio} - 1) \]

\[ u0(temp) = u0(tref) \cdot (T_{ratio})^{UTE} \]

\[ V_{sat(temp)} = V_{sat(tref)} - A_f \cdot (T_{ratio} - 1) \]

\[ U_a(temp) = U_a(tref) + U_{a1} \cdot (T_{ratio} - 1) \]

\[ U_b(temp) = U_b(tref) + U_{b1} \cdot (T_{ratio} - 1) \]

\[ U_c(temp) = U_c(tref) + U_{c1} \cdot (T_{ratio} - 1) \]
PMOS Model

The following is an example of a PMOS model. Note that VTH0 is negative.

```
.model pch PMOS LEVEL=47
+ Tnom=27.0
+ Npeak= 1.5E+23  Tox=7.0E-09  Xj=1.0E-07
+ dl= 0.2E-06  dw=-0.1E-06
+ SatMod= 2  SubthMod= 2  BulkMod= 1
+ Vth0= -.8  Phi= .7  K1= .5  K2=0.03  K3= 0
+ Dvt0= 48  Dvt1= .6  Dvt2=-5e-4
+ Nlx=0  W0= 0
+ Vsat= 9E6  Ua= 1E-09  Ub= 0  Uc= -3E-02
+ Rds0= 180  Rdsw= 0  U0= 7E-03
+ A0= .87
+ Voff=-.07  NFactor= 1.5  Cit=-3E-05
+ Cdsc= 6E-02  Vglow=-.12  Vghigh= .12
+ Pclm= 77  Pdibl1= 0  Pdibl2= 2E-011
+ Drout= 0  Pscbe1= 0  Pscbe2= 1E-28
+ Eta= 0  Litl= 4.5E-08
+ Em= 0  Ldd= 0
+ kt1=-.3  kt2=-.03
+ At= 33000
+ Ua1= 4E-09  Ub1= 7E-18  Uc1= 0
```
LEVELs 49 and 53 BSIM3v3 MOS Models

The BSIM3v3 MOS model from UC Berkeley is available in Star-Hspice as LEVEL 49 and LEVEL 53. LEVEL 49 is an Hspice-enhanced version of BSIM3v3 while LEVEL 53 (first released in Star-Hspice 98.2) maintains full compliance with the Berkeley release. This compliance includes numerically identical model equations, identical parameter default values, and identical parameter range limits. LEVEL 49 maintains compliance with the UC Berkeley release of BSIM3v3 with the following three exceptions:

1. **Default parameter values.** Eliminate differences in default parameter values by explicit assignment of the parameters CAPMOD, XPART and by setting ACM=10.

2. **Parameter range limits.** Provides parameter range limits that are identical to that of the Berkeley release. Differences occur only in the severity of warning for five parameters. LEVEL 49 issues a warning that the parameter range has been exceeded but continues with simulation, whereas, in the Berkeley release, a fatal error is issued and simulation is aborted. These five parameters include NGATE, DVT1W, DVT1, DSUB, DROUT. (See the Model Parameter Range Limits below for more details.)

3. **Improvements in numerical stability.** Provides improvements in numerical stability. In most practical situations, these improvements will not affect compliance with the Berkeley release, but will improve convergence and simulation time.

Both LEVELs 49 and 53 support a superset of model parameters that include Hspice-specific parameters. For LEVEL 53, in all cases, Hspice-specific parameters default to OFF. The single exception in LEVEL 49 is that ACM defaults to 0. LEVEL 49 compliance with Berkeley BSIM3v3 can be achieved by setting ACM=10.
Selecting Model Versions

Recommended BSIM3v3 Version

As of the Star-Hspice 99.2 release (June 1999), the recommended BSIM3v3 model specification is LEVEL=49, VERSION=3.22. This version provides the most stable and up-to-date representation of the UCB BSIM3v3.2.2 model. However, do not change the VERSION specification in existing model cards without consulting the foundry or model extraction group that created the original model cards.

There are, as of the 99.2 release, five official BSIM3v3 releases from Berkeley and several Star-Hspice LEVEL 49 releases. (See the BSIM3 home page at http://www-device.EECS.Berkeley.EDU/~bsim3/ for additional release information from the UCB group.) To minimize confusion and maintain back compatibility, you can select the model parameters VERSION and HSPVER. VERSION selects the Berkeley release version and HSPVER selects the Star-Hspice release version. For example, HSPVER=97.2 and VERSION=3.1 reproduce results from Hspice 97.2 using the BSIM3 Version 3.1 model.

HSPVER defaults to the current release being executed. The model parameter, VERSION, selects among the various Berkeley releases of BSIM3v3 as follows:

- **Version 3.0 Berkeley release (October 30, 1995) default for HSPICE96.1,96.2,96.3.** This version is invoked when VERSION=3.0 and HSPVER=98.0 are specified. To invoke the Star-Hspice version that most accurately represents the Berkeley release of October 1995, specify the parameters VERSION=3.0 and HSPVER=98.0.

- **Version 3.1 Berkeley (December 9, 1997) default for HSPICE97.1,97.2,97.4.** This version is invoked when VERSION=3.1 or 3.11 and HSPVER=98.0 are specified. To invoke the Star-Hspice version that most accurately represents the Berkeley release of December, 1996 specify the parameters VERSION=3.1 or 3.11 and HSPVER=98.0.

- **Berkeley Version 3.0, 3.1 bug fixes.** Berkeley corrected several Version 3.0 and 3.1 bugs in the June, 1998 release. These bug fixes are incorporated into Hspice98.2 and are represented when VERSION=3.0 and VERSION=3.1 are specified respectively with HSPVER=98.2. As a result of bug fixes,
some differences between Version 3.0/3.1 in Hspice98.2 and previous Version 3.0/3.1 releases are expected. Most notably, differences will occur when perimeter factors PD, PS less than Weff are specified (PD, PS < Weff are no longer clamped to Weff in Version 3.1) and when DLC and LINT are not identical (LeffCV calculation bug in Versions 3.0, 3.1). You can find a complete list of bug fixes at the BSIM3 web site:

http://www-device.eecs.berkeley.edu/~bsim3.

**Note:** Version 3.11 was introduced in Hspice97.4. This version represented Berkeley Version 3.1 (Dec., 1996) with Hspice bug fixes. Back compatibility will be maintained for this model. Starting with Hspice98.2, Version 3.1 and 3.11 will be identical and represent Version 3.1 with Berkeley June, 1998 bug fixes.

- **Version 3.2 Berkeley release (June 16, 1998).** This version is invoked when VERSION=3.2 and HSPVER=98.2 are specified.
- **Version 3.2.1 Berkeley release (April 20, 1999).** This version is invoked when VERSION=3.21 and HSPVER=99.2 are specified.
- **Version 3.2.2 Berkeley release (April 20, 1999).** This version is invoked when VERSION=3.22 and HSPVER=99.2 are specified.

**Note:** Versions 3.2.1 and 3.2.2 are identical except BSIM3v3.2.1 uses a bias-dependent Vfb and BSIM3v3.2.2 uses a bias-independent Vfb for the capacitance models capMod = 1 and 2.

The table below summarizes the Star-Hspice parameter settings required to match Berkeley releases:

<table>
<thead>
<tr>
<th>Berkeley Release</th>
<th>VERSION</th>
<th>HSPVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 3.0 (October 1995)</td>
<td>3.0</td>
<td>98.0</td>
</tr>
<tr>
<td>Version 3.0 with June 1998 bug fixes</td>
<td>3.0</td>
<td>98.2</td>
</tr>
<tr>
<td>Version 3.1 (December 1996)</td>
<td>3.1</td>
<td>98.0</td>
</tr>
</tbody>
</table>
In June, 1998 Berkeley released BSIM3 Version 3.2, which contains many new features. These features are summarized below.

- A new intrinsic capacitance model, CAPMOD=3, includes finite charge layer thickness effects; CAPMOD now defaults to 3 (new parameters: CAPMOD=3, ACDE, MOIN)
- Improved modeling of C-V characteristics at the weak-to-strong inversion transition (new parameters: NOFF, VOFFCV)
- Vth dependence on Tox (new parameter: TOXM)
- Flatband voltage parameter more accurately models different gate materials (new parameter: VFB)
- Improved substrate current scaleability with channel length, (new parameter: APLHA1)
- Restructured nonquasi-static (NQS) model includes pole-zero analysis and bug fixes. Note that NQSMOD is now a BSIM3 element parameter. Hspice supports only the model parameter not the element parameter.
- Junction diode model temperature dependence, (new parameters: TCJ, TCJSW, TCJSWG, TPB, TPBSW, TPBSWG)
- Adjustable current limiting in the junction diode current model (new parameter: IJTH)
- Option of using C-V inversion charge equations of CAPMOD=0,1,2,3 to calculate the thermal noise when NOIMOD=2 or 4
- Elimination of small negative capacitance values (Cgs, Cgd) in the accumulation-depletion regions
A separate set of length/width dependence parameters for the CV model (New parameters: LLC, LWC, LWLC, WLC, WWC, WWLC)
- Additional parameter checking
- Bug fixes

Note: If all new Version 3.2 parameters are defaulted, Version 3.2 and Version 3.1 (with June, 1998 bug fixes) will give identical DC results. However, transient and AC results will differ, in general. This discrepancy arises only from differences in flatband voltage calculations used in the intrinsic charge/capacitance models. These differences occur in all CAPMOD models 1-3.
HSPVER < 98.0 will be reset to 98.0 for LEVEL 53.
HSPVER < 98.2 will be reset to 98.2 when VERSION >=3.2 for LEVELs 49 and 53.
Version 3.0, 3.1, and 3.11 in Hspice do not support NQSMOD and CAPMOD=3. These are supported only by Version 3.2.

You can obtain additional information about the Berkeley releases from the BSIM3 web site:
http://www-device.eecs.berkeley.edu/~bsim3.

Nonquasi-Static (NQS) Model


To invoke the NQS model, specify the parameter NQSMOD=1 in the model card. NQSMOD can be used with any of the CAPMOD LEVELs (0-3) but is restricted to use with Version 3.2. NQS is not supported in Version 3.0 and 3.1. In future releases, the NQS will be supported in Versions 3.0, 3.1.
Star-Hspice Enhancements

Hspice Junction Diode Model and Area Calculation Method (ACM)

There are two junction diode models that can be used with both LEVELs 49 and 53: the Hspice junction model and the Berkeley junction model. The Hspice junction model is invoked by specifying the model parameter value ACM=0,1,2, or 3. The Berkeley junction model is invoked by specifying ACM=10,11,12, or 13. The default ACM value is 0 and 10 for LEVELs 49 and 53 respectively. The junction current, junction capacitance, and parasitic resistance equations corresponding to ACM=0,1,2,3 can be found in “Selecting MOSFET Diode Models” on page 20-26.

The effect of setting ACM=10,11,12, or 13 is to enable the Berkeley junction diodes and to add parasitic resistors to the MOSFET. The parasitic resistor equations for ACM=10-13 correspond to the ACM=0-3 parasitic resistor equations respectively. ACM=10-13 all use the Berkeley junction capacitance model equations:

\[
\begin{align*}
\text{(Bulk-source capacitance)} \\
\text{if } (Ps > Weff) \quad & Cbs = AS \times Cjbs + (PS - Weff) \times Cjbssw + Weff \times Cjbsswg \\
\text{else} \quad & Cbs = AS \times Cjbs + PS \times Cjbsswg
\end{align*}
\]

Area and perimeter factors AS, PS default to 0 if not specified on the element line.

\[
\begin{align*}
\text{if } (Vbs < 0) \quad & Cjbs = Cj \times (1 - (Vbs/Pb))^{-Mj} \\
& Cjbssw = Cjsw \times (1 - (Vbs/Pbsw))^{-Mjsw} \\
& Cjbsswg = Cjswg \times (1 - (Vbs/Pbswg))^{-Mjswg} \\
\text{else} \quad & Cjbs = Cj \times (1 + Mj \times (Vbs/Pb)) \\
& Cjbssw = Cjsw \times (1 + Mjsw \times (Vbs/Pbsw)) \\
& Cjbsswg = Cjswg \times (1 + Mjswg \times (Vbs/Pbswg))
\end{align*}
\]
Bulk-drain equations are analogous. Note that the Hspice equations for AS, PS, AD, PD are not used with ACM=10, 11, 12, 13 and, in accordance with the BSIM3v3 model, the default values for these area and perimeter factors are zero. However, starting with Star-Hspice version 98.2, it is possible to invoke the Hspice calculations for AS, PS, AD, PD by specifying the model parameter CALCACM=1.

Important: CALCACM is only invoked when used with ACM=12. The calculations used in ACM=10, 11, 13 are not consistent with the Berkeley diode calculations.

With CALCACM = 1 and ACM = 12 the following area and perimeter calculations are invoked:

if AD is not specified on the element line:
   \[ AD = 2 \times HDIFeff \times Weff \]
else:
   \[ AD = AD \times WMLT^2 \]

if AS is not specified on the element line:
   \[ AS = 2 \times HDIFeff \times Weff \]
else:
   \[ AS = AS \times WMLT^2 \]

if PS is not specified on the element line:
   \[ PS = 4 \times HDIFeff + 2 \times Weff \]
else:
   \[ PS = PS \times WMLT \]

if PD is not specified on the element line:
   \[ PD = 4 \times HDIFeff + 2 \times Weff \]
else:
   \[ PD = PD \times WMLT \]
Note: Weff is not the same Weff used in the BSIM3v3, and LEVELs 49 and 53 I-V, C-V model equations!

In the preceding equations the following simple form is used.

\[ \text{Weff} = \text{W} \times \text{WMLT} + \text{XW} \]

where:

- \( \text{HDIFeff} = \text{HDIF} \times \text{WMLT} \)
- \( \text{W} \) is the width specified on the element line
- \( \text{HDIF} \) is a heavy diffusion length specified in the model card
- \( \text{WMLT} \) is a shrink factor specified in the model card
- \( \text{XW} \) is an etch/mask effect factor specified in the model card

Note: SCALM, SCALE, and M factor effects have been ignored in these equations. Please see “Using a MOSFET Diode Model” on page 20-26 (ACM=2) for further details.

Parameter Differences

There are some differences in parameter names between the Star-Hspice and the Berkeley junction models. The Star-Hspice models (ACM=0-3) do not recognize the following BSIM3v3 parameters:

- \( \text{NJ} \) (ignored, instead use \( \text{N} \))
- \( \text{CJSWG} \) (ignored, instead use \( \text{CJGATE} \))
- \( \text{MJSWG} \) (ignored, there is no equivalent HSPICE parameter, the gate sidewall grading coefficient will be set = \( \text{MJSW} \))
- \( \text{PBSW} \) (ignored, instead use \( \text{PHP} \))
- \( \text{PBSWG} \) (ignored, there is no equivalent HSPICE parameter, the gate sidewall contact potential will be set = \( \text{PHP} \))
The Berkeley model (ACM=10,11,12,13) will not recognize the following Star-Hspice parameters:
- CJGATE (ignored, instead use CJSWG)
- PHP (ignored, instead use PBSW)

**Star-Hspice Noise Model**

The Hspice-specific parameter NLEV overrides the BSIM3v3 parameter NOIMOD. Specifying NLEV will invoke the Hspice noise model. See “Using Noise Models” on page 20-99 for further information. If NLEV is not specified, the Berkeley noise equations are invoked.

**Performance Improvements**

The performance of LEVELS 49 and 53 has been improved by reducing model equation complexity, replacing some calculations with spline functions, and compiler optimization. For LEVEL 49, the result is a reduction in simulation time of up to 40% compared to releases prior to 97.4 while maintaining accuracy to 5 digits or better. The use of spline functions can be enabled by setting the model parameter to SFVTFLAG=1 in the model card. SFVTFLAG=0, the default value, disables the spline functions. For LEVEL 53, all BSIM3v3 non-compliant features default to off. There is a significant reduction in simulation time compared to pre-97.4 releases remains.

**Reduced Parameter Set BSIM3v3 Model (BSIM3-lite)**

Setting the LEVEL 49 model parameter LITE=1 will invoke the BSIM3v3-lite model. This is a BSIM3v3 reduced parameter set model that is intended to be used with model binning. Without binning, the full BSIM3v3 model accounts for geometry effects through the specification of many model parameters. However, it is often difficult to extract a “global” BSIM3v3 model that is accurate over the entire geometry range. To improve accuracy over a range of geometries, Star-Hspice allows the user to bin model parameters. That is, the entire length-width geometry range is divided into rectangular regions or bins. A different set of parameters is extracted for each bin. The Hspice built-in bilinear parameter interpolation scheme maintains continuity (over length-width) at the boundaries between bins. Since many BSIM3 model parameters account for MOSFET
geometry effects, these geometry-effect parameters are redundant and can be eliminated when binning is used.

The BSIM3-lite model parameter set was created in response to the question: What BSIM3 parameters should be excluded when using a binned model? The BSIM3-lite model is invoked by specifying the model parameter LITE=1 in the model card. Star-Hspice will check the model card to determine if it conforms to the BSIM3-lite parameter set. BSIM3-lite takes advantage of the smaller number of calculations and will reduce simulation times by up to 10% compared to the full parameter set BSIM3 model. LITE=1 is supported only by LEVEL 49.

The following table lists model parameters (total 49) that are excluded from the BSIM3-lite model. All parameters in this list should either be excluded from the model card or explicitly set to the default value specified in the list. In some cases, as noted, the BSIM3-lite default value differs from the standard BSIM3v3 default value. Also, exclusion of WR, ALPHAO, CIT is only recommended but not required in the BSIM3-lite model card.

**Parameters Excluded from BSIM3-Lite Model**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>mobmod</td>
<td>Recommended default or set = 1</td>
</tr>
<tr>
<td>nqsmod</td>
<td>Recommended default or set = 0</td>
</tr>
<tr>
<td>toxm</td>
<td>default = tox</td>
</tr>
<tr>
<td>ll</td>
<td>default = 0</td>
</tr>
<tr>
<td>lln</td>
<td>default = 1</td>
</tr>
<tr>
<td>lw</td>
<td>default = 0</td>
</tr>
<tr>
<td>lwn</td>
<td>default = 1</td>
</tr>
<tr>
<td>lwl</td>
<td>default = 0</td>
</tr>
<tr>
<td>wl</td>
<td>default = 0</td>
</tr>
<tr>
<td>wln</td>
<td>default = 1</td>
</tr>
<tr>
<td>ww</td>
<td>default = 0</td>
</tr>
<tr>
<td>Parameter</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>wwn</td>
<td>default = 1</td>
</tr>
<tr>
<td>wwl</td>
<td>default = 0</td>
</tr>
<tr>
<td>dwg</td>
<td>default = 0</td>
</tr>
<tr>
<td>dwb</td>
<td>default = 0</td>
</tr>
<tr>
<td>llc</td>
<td>default = 0</td>
</tr>
<tr>
<td>lwc</td>
<td>default = 0</td>
</tr>
<tr>
<td>lwlc</td>
<td>default = 0</td>
</tr>
<tr>
<td>wlc</td>
<td>default = 0</td>
</tr>
<tr>
<td>wwc</td>
<td>default = 0</td>
</tr>
<tr>
<td>wwlc</td>
<td>default = 0</td>
</tr>
<tr>
<td>b0</td>
<td>default = 0</td>
</tr>
<tr>
<td>b1</td>
<td>default = 0</td>
</tr>
<tr>
<td>vbx</td>
<td>do not define</td>
</tr>
<tr>
<td>vbm</td>
<td>do not define</td>
</tr>
<tr>
<td>xt</td>
<td>do not define</td>
</tr>
<tr>
<td>nsub</td>
<td>do not define</td>
</tr>
<tr>
<td>nlx</td>
<td>default = 0, std default=1.74e-7</td>
</tr>
<tr>
<td>gamma1</td>
<td>do not define</td>
</tr>
<tr>
<td>gamma2</td>
<td>do not define</td>
</tr>
<tr>
<td>ngate</td>
<td>Recommended default or set = 0</td>
</tr>
<tr>
<td>k3</td>
<td>default = 0, std default=80</td>
</tr>
<tr>
<td>k3b</td>
<td>default = 0</td>
</tr>
<tr>
<td>w0</td>
<td>no effect</td>
</tr>
<tr>
<td>dvt0</td>
<td>default = 0, std default=2.2</td>
</tr>
</tbody>
</table>
Parameter Binning

Parameter binning is supported in the Berkeley BSIM3v3 release through the specification of LWP parameters. That is, a subset of model parameters can be bilinearly interpolated over $\frac{1}{L_{eff}}$ and $\frac{1}{W_{eff}}$ by specifying four terms: the parameter $X_o$, a length term $X_l$, a width term $X_w$, and a product term $X_p$. The parameter value at a given $L, W$ is then interpolated as:

$$X = X_o + \frac{X_l}{L_{eff}} + \frac{X_w}{W_{eff}} + \frac{X_p}{L_{eff}/W_{eff}}$$

See “Model Parameter Range Limit” on page 22-49 to determine whether a parameter can be binned. Star-Hspice adds parameters LMIN, LMAX, WMIN, WMAX and LREF, WREF to allow multiple cell binning. LMIN, LMAX, WMIN, WMAX define the cell boundary. LREF, WREF are offset values that provide a convenient interpolation scheme. LREF, WREF offsets are used when

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>dvt1</td>
<td>default = 0, std default=0.53</td>
</tr>
<tr>
<td>dvt2</td>
<td>default = 0, std default=-0.032</td>
</tr>
<tr>
<td>dvt0w</td>
<td>default = 0</td>
</tr>
<tr>
<td>dvt1w</td>
<td>default = 0, std default=5.3e6</td>
</tr>
<tr>
<td>dvt2w</td>
<td>default = 0, std default=-0.032</td>
</tr>
<tr>
<td>dsub</td>
<td>default = 0</td>
</tr>
<tr>
<td>prwg</td>
<td>default = 0</td>
</tr>
<tr>
<td>prwb</td>
<td>default = 0</td>
</tr>
<tr>
<td>wr</td>
<td>Recommended default or set = 1</td>
</tr>
<tr>
<td>drout</td>
<td>default = 0, std default=0.56</td>
</tr>
<tr>
<td>pdblc1</td>
<td>default = 0, std default=0.39</td>
</tr>
<tr>
<td>cit</td>
<td>Recommended default or set = 0</td>
</tr>
<tr>
<td>alpha0</td>
<td>Recommended default or set = 0 for Version 3.2</td>
</tr>
<tr>
<td>kt1l</td>
<td>default = 0</td>
</tr>
</tbody>
</table>
both values are defined and the model parameter BINFLAG $> 0.9$ is specified. The parameter value at a given $L,W$ is then interpolated as:

$$X = X_0 + X_L(1/L_{eff} - 1/L_{REF}) + X_W(1/W_{eff} - 1/W_{REF}) + X_p/(1/L_{eff} - 1/L_{REF})/(1/W_{eff} - 1/W_{REF})$$

The units for the lwp geometry parameters can be selected to be in microns by setting the model parameter BINUNIT = 1. For other choices of BINUNIT, the lengths are in units of meters. The Hspice parameters $XL, XLREF, XW, and XWREF$ are handled in a manner consistent with other Hspice models, and they produce shifts in parameter values without disrupting the continuity across bin boundaries.

**Charge Models**

In the December, 1996 release of BSIM3v3, Berkeley offers the BSIM1 capacitance model as CAPMOD=0. This is replaced with a modified BSIM1 capacitance model based on the Hspice CAPOP=13 model in LEVEL 49. LEVEL 53 uses the Berkeley BSIM1 capacitance model for CAPMOD=0. The following table lists CAPMOD defaults for the Berkeley BSIM3v3 model and for LEVELs 49 and 53.

<table>
<thead>
<tr>
<th>VERSION</th>
<th>BSIM3v3</th>
<th>LEVEL 49</th>
<th>LEVEL 53</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3.1</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3.2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

**Hspice VFBFLAG**

The capacitance model CAPMOD=0 normally calculates the threshold voltage as $V_{th} = v_{fbc} + \phi + k_1 \times \sqrt{\phi - v_{bs}}$, where $v_{fbc}$ is the model parameter VFBCV. This has the effect of eliminating any dependence on the parameter $VTH0$. To allow capacitance dependence on $VTH0$, set the model parameter VFBFLG=1. The capacitance model CAPMOD=0 will calculate the threshold voltage as $V_{th} = v_{th0} + k_1 \times \sqrt{\phi - v_{bs}} - k_1 \times \sqrt{\phi}$. The VFBFLG default value is 0.
Printback

Printback of all model parameters with units is now enabled. The printback also indicates whether Berkeley or Star-Hspice junction diodes and noise models are invoked and which parameters are not used (e.g. CJGATE is not used when ACM=0-3).

Using BSIM3v3 in Star-Hspice

The following are points to note when using BSIM3v3 in Star-Hspice:

1. Use either the LEVEL 49 or LEVEL 53 model. LEVEL 53 maintains full compliance with the Berkeley BSIM3v3 release. However, in most cases LEVEL 49, in comparison to LEVEL 53, will give identical results, run as fast or faster, show better convergence, and allow a wider range of parameter specifications.

2. Explicitly set all Berkeley-specific BSIM3 model parameters in the model card. This will minimize problems resulting from version changes and compatibility with other simulators. Explicitly setting all lwp binning parameters is not necessary.

3. To obtain matching results with simulations from previous Hspice versions use the model parameter HSPVER=YY.N, e.g., HSPVER=97.4. Do not use the full year specification (e.g., do not use 1997.4). Patch version numbers are implemented as HSPVER=YY.NN (e.g., HSPVER=98.21 for Hspice release 98.2.1).

4. LEVELs 49 and 53 support the model parameter name TNOM as an alias for TREF. The conventional terminology in Hspice is TREF, which is supported as a model parameter in all Star-Hspice MOS levels. The alternative name TNOM is supported in both LEVELs 49 and 53, for compatibility with SPICE3.

The default room temperature is 25°C in Star-Hspice, but is 27°C in SPICE3. If the BSIM3 model parameters are specified at 27°C, TNOM=27 should be added to the model, so that the model parameters are interpreted correctly. It is a matter of choice whether or not to set the nominal...
simulation temperature to 27, by adding .OPTION TNOM=27 to the netlist. Add this option when testing Star-Hspice versus SPICE3.

DELVTO and DTEMP on the element line can be used with LEVELs 49 and 53. The conversion of temperature setup between Star-Hspice and SPICE3 is as follows:

**SPICE3:**
```
.OPTIONS TEMP=125
.MODEL NCH NMOS LEVEL=8
  + TNOM =27 ...
```

**Star-Hspice:**
```
.TEMP 125
.MODEL NCH NMOS LEVEL=49
  + TNOM =27 ...
```

5. To invoke automatic calculation of drain and source area and perimeter factors with the Berkeley junction diode models use ACM=12 with CALCACM=1. Normally, ACM=10-13 will default area and perimeter factors to 0. This can only be overridden for ACM=12 by specifying CALCACM=1. Make sure that the Hspice-specific parameter HDIF is defined in the model card. If you do not want to have parasitic Rs and Rd in addition to the BSIM3v3 internal Rsd, then make sure that the Hspice-specific parameters RSH, RSC, RDC, RS, RD are either not specified (default will be 0) or explicitly set to 0.

6. Star-Hspice will either warn or abort with a fatal error when certain model parameter values are out of a normal range. To view all the warnings, the .OPTION WARNLIMIT value may have to be increased (default=1). To turn full parameter range checking, set the model parameter PARAMCHK=1 (default is 0). With PARAMCHK=0 a smaller set of parameters is checked. (See “Model Parameter Range Limit” on page 22-49 for more details regarding parameter limits.) Use the model parameter APWARN=1 (default=0) to turn off PS,PD < Weff warnings.

7. NQSMOD can only be used with Version 3.2 and can only be specified in the model card as of release Hspice 98.2.
LEVEL 49, 53 Model Parameters

The following tables describe all LEVEL 49 and LEVEL 53 model parameters including parameter name, units, default value, whether the parameter can be binned, and a description. Note that these tables are a superset of the BSIM3v3 model parameter set and include Hspice-specific parameters. These Hspice-specific parameters are noted in the description column and always default (for LEVEL 53) so that compliance with the BSIM3v3 standard is maintained. These parameters also apply to LEVEL 49 with the following exceptions: ACM default value = 0, XPART default value = 1, CAPMOD default value = 0.

Model Flags

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERSION</td>
<td>-</td>
<td>3.2</td>
<td>No</td>
<td>Selects from BSIM3 Versions 3.0, 3.1, 3.2. Warning is issued if not explicitly set.</td>
</tr>
<tr>
<td>HSPVER</td>
<td>-</td>
<td>98.2</td>
<td>No</td>
<td>Selects from Hspice Versions: 98.2, 97.4, 97.2, 96.4, 96.3, 96.1</td>
</tr>
<tr>
<td>PARAMCHK</td>
<td>-</td>
<td>0</td>
<td>No</td>
<td>PARAMCHK=1 will check model parameters for range compliance</td>
</tr>
<tr>
<td>APWARN</td>
<td>-</td>
<td>0</td>
<td>No</td>
<td>When &gt; 0 turns off warning message for PS,PD &lt; Weff (Hspice specific)</td>
</tr>
<tr>
<td>BINFLAG</td>
<td>-</td>
<td>0</td>
<td>No</td>
<td>Uses wref, lref when set &gt; 0.9 (Hspice specific)</td>
</tr>
<tr>
<td>MOBMOD</td>
<td>-</td>
<td>1</td>
<td>No</td>
<td>Mobility model selector</td>
</tr>
<tr>
<td>CAPMOD</td>
<td>-</td>
<td>3</td>
<td>No</td>
<td>Selects from charge models 0,1,2,3 LEVEL 49 CAPMOD defaults to 0.</td>
</tr>
<tr>
<td>CAPOP</td>
<td>-</td>
<td>-</td>
<td>No</td>
<td>Obsolete for LEVELs 49, 53. Ignored by Hspice (Hspice specific) in all versions.</td>
</tr>
<tr>
<td>NOIMOD</td>
<td>-</td>
<td>1</td>
<td>No</td>
<td>Berkeley noise model flag</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 47-62

LEVELs 49 and 53 BSIM3v3 MOS Models

### Basic Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NQSMOD</td>
<td>-</td>
<td>0 (off)</td>
<td>No</td>
<td>NQS Model flag</td>
</tr>
<tr>
<td>SFVTFLAG</td>
<td>-</td>
<td>0 (off)</td>
<td>No</td>
<td>Spline function for Vth (Hspice specific)</td>
</tr>
<tr>
<td>VFBFLAG</td>
<td>-</td>
<td>0 (off)</td>
<td>No</td>
<td>VFB selector for CAPMOD=0 (Hspice specific)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGSLIM</td>
<td>V</td>
<td>0</td>
<td>No</td>
<td>Asymptotic Vgs value, Min value is 5V. 0-value indicates an asymptote of infinity. (Hspice and LEVEL 49 specific)</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>150e-10</td>
<td>No</td>
<td>Gate oxide thickness</td>
</tr>
<tr>
<td>XJ</td>
<td>m</td>
<td>0.15e-6</td>
<td>Yes</td>
<td>Junction depth</td>
</tr>
<tr>
<td>NGATE</td>
<td>cm$^{-3}$</td>
<td>0</td>
<td>Yes</td>
<td>Poly gate doping concentration</td>
</tr>
<tr>
<td>VTH0</td>
<td>V</td>
<td>0.7 NMOS -0.7 PMOS</td>
<td>Yes</td>
<td>Threshold voltage of long channel device at $V_{ds} = 0$ and small $V_{ds}$</td>
</tr>
<tr>
<td>NSUB</td>
<td>cm$^{-3}$</td>
<td>6.0e16</td>
<td>Yes</td>
<td>Substrate doping concentration</td>
</tr>
<tr>
<td>NCH</td>
<td>cm$^{-3}$ See Note6</td>
<td>1.7e17</td>
<td>Yes</td>
<td>Peak doping concentration near interface</td>
</tr>
<tr>
<td>NLX</td>
<td>m</td>
<td>1.74e-7</td>
<td>Yes</td>
<td>Lateral nonuniform doping along channel</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default</td>
<td>Bin</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>---------</td>
<td>-----</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>K1</td>
<td>V^{1/2}</td>
<td>0.50</td>
<td>Yes</td>
<td>First-order body effect coefficient</td>
</tr>
<tr>
<td>K2</td>
<td>-</td>
<td>-0.0186</td>
<td>Yes</td>
<td>Second-order body effect coefficient</td>
</tr>
<tr>
<td>K3</td>
<td>-</td>
<td>80.0</td>
<td>Yes</td>
<td>Narrow width effect coefficient</td>
</tr>
<tr>
<td>K3B</td>
<td>1/V</td>
<td>0</td>
<td>Yes</td>
<td>Body width coefficient of narrow width effect</td>
</tr>
<tr>
<td>W0</td>
<td>m</td>
<td>2.5e-6</td>
<td>Yes</td>
<td>Narrow width effect coefficient</td>
</tr>
<tr>
<td>DVT0W</td>
<td>1/m</td>
<td>0</td>
<td>Yes</td>
<td>Narrow width coefficient 0, for Vth, at small L</td>
</tr>
<tr>
<td>DVT1W</td>
<td>1/m</td>
<td>5.3e6</td>
<td>Yes</td>
<td>Narrow width coefficient 1, for Vth, at small L</td>
</tr>
<tr>
<td>DVT2W</td>
<td>1/V</td>
<td>-0.032</td>
<td>Yes</td>
<td>Narrow width coefficient 2, for Vth, at small L</td>
</tr>
<tr>
<td>DVT0</td>
<td>-</td>
<td>2.2</td>
<td>Yes</td>
<td>Short channel effect coefficient 0, for Vth_</td>
</tr>
<tr>
<td>DVT1</td>
<td>-</td>
<td>0.53</td>
<td>Yes</td>
<td>Short channel effect coefficient 1, for Vth_</td>
</tr>
<tr>
<td>DVT2</td>
<td>1/V</td>
<td>-0.032</td>
<td>Yes</td>
<td>Short channel effect coefficient 2, for Vth_</td>
</tr>
<tr>
<td>ETA0</td>
<td>-</td>
<td>0.08</td>
<td>Yes</td>
<td>Subthreshold region DIBL (drain induced barrier lowering) coefficient</td>
</tr>
<tr>
<td>ETAB</td>
<td>1/V</td>
<td>-0.07</td>
<td>Yes</td>
<td>Subthreshold region DIBL coefficient</td>
</tr>
<tr>
<td>DSUB</td>
<td>-</td>
<td>DROUT</td>
<td>Yes</td>
<td>DIBL coefficient exponent in subthreshold region</td>
</tr>
<tr>
<td>VBM</td>
<td>V</td>
<td>-3.0</td>
<td>Yes</td>
<td>Maximum substrate bias, for Vth calculation</td>
</tr>
</tbody>
</table>
## Selecting MOSFET Models: LEVEL 47-62

**LEVELs 49 and 53 BSIM3v3 MOS Models**

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U0</td>
<td>cm²/V·sec</td>
<td>670 nmos</td>
<td>Yes</td>
<td>Low field mobility at ( T = T_{REF} = T_{NOM} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>250 pmos</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UA</td>
<td>m/V</td>
<td>2.25e-9</td>
<td>Yes</td>
<td>First-order mobility degradation coefficient</td>
</tr>
<tr>
<td>UB</td>
<td>m²/V²</td>
<td>5.87e-19</td>
<td>Yes</td>
<td>Second-order mobility degradation coefficient</td>
</tr>
<tr>
<td>UC</td>
<td>1/V</td>
<td>-4.65e-11 or</td>
<td>Yes</td>
<td>Body bias sensitivity coefficient of mobility</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.0465</td>
<td></td>
<td>-4.65e-11 for ( MOBMOD = 1, 2 ) or, -0.0465 for ( MOBMOD = 3 )</td>
</tr>
<tr>
<td>A0</td>
<td>-</td>
<td>1.0</td>
<td>Yes</td>
<td>Bulk charge effect coefficient for channel length</td>
</tr>
<tr>
<td>AGS</td>
<td>1/V</td>
<td>0.0</td>
<td>Yes</td>
<td>Gate bias coefficient of ( A_{bulk} )</td>
</tr>
<tr>
<td>B0</td>
<td>m</td>
<td>0.0</td>
<td>Yes</td>
<td>Bulk charge effect coefficient for channel width</td>
</tr>
<tr>
<td>B1</td>
<td>m</td>
<td>0.0</td>
<td>Yes</td>
<td>Bulk charge effect width offset</td>
</tr>
<tr>
<td>KETA</td>
<td>1/V</td>
<td>-0.047</td>
<td>Yes</td>
<td>Body-bias coefficient of bulk charge effect</td>
</tr>
<tr>
<td>VOFF</td>
<td>V</td>
<td>-0.08</td>
<td>Yes</td>
<td>Offset voltage in subthreshold region</td>
</tr>
<tr>
<td>VSAT</td>
<td>m/sec</td>
<td>8e4</td>
<td>Yes</td>
<td>Saturation velocity of carrier at ( T = T_{REF} = T_{NOM} )</td>
</tr>
<tr>
<td>A1</td>
<td>1/V</td>
<td>0</td>
<td>Yes</td>
<td>First nonsaturation factor</td>
</tr>
<tr>
<td>A2</td>
<td>-</td>
<td>1.0</td>
<td>Yes</td>
<td>Second nonsaturation factor</td>
</tr>
<tr>
<td>RDSW</td>
<td>ohm · μm</td>
<td>0.0</td>
<td>Yes</td>
<td>Parasitic source drain resistance per unit width</td>
</tr>
<tr>
<td>PRWG</td>
<td>1/V</td>
<td>0</td>
<td>Yes</td>
<td>Gate bias effect coefficient of RDSW</td>
</tr>
</tbody>
</table>

---

## Selecting MOSFET Models: LEVEL 47-62

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRWB</td>
<td>$1/V^{1/2}$</td>
<td>0</td>
<td>Yes</td>
<td>Body effect coefficient of RDSW</td>
</tr>
<tr>
<td>WR</td>
<td>-</td>
<td>1.0</td>
<td>Yes</td>
<td>Width offset from $W_{eff}$ for $R_d$ calculation</td>
</tr>
<tr>
<td>NFACTOR</td>
<td>-</td>
<td>1.0</td>
<td>Yes</td>
<td>Subthreshold region swing</td>
</tr>
<tr>
<td>CIT</td>
<td>F/m$^2$</td>
<td>0.0</td>
<td>Yes</td>
<td>Interface state capacitance</td>
</tr>
<tr>
<td>CDSC</td>
<td>F/m$^2$</td>
<td>2.4e-4</td>
<td>Yes</td>
<td>Drain/source and channel coupling capacitance</td>
</tr>
<tr>
<td>CDSCD</td>
<td>F/Vm$^2$</td>
<td>0</td>
<td>Yes</td>
<td>Drain bias sensitivity of CDSC</td>
</tr>
<tr>
<td>CDSCB</td>
<td>F/Vm$^2$</td>
<td>0</td>
<td>Yes</td>
<td>Body coefficient for CDSC</td>
</tr>
<tr>
<td>PCLM</td>
<td>-</td>
<td>1.3</td>
<td>Yes</td>
<td>Coefficient of channel length modulation values $\leq 0$ will result in an error message and program exit.</td>
</tr>
<tr>
<td>PDIBLC1</td>
<td>-</td>
<td>0.39</td>
<td>Yes</td>
<td>DIBL (drain induced barrier lowering) effect coefficient 1</td>
</tr>
<tr>
<td>PDIBLC2</td>
<td>-</td>
<td>0.0086</td>
<td>Yes</td>
<td>DIBL effect coefficient 2</td>
</tr>
<tr>
<td>PDIBLCB</td>
<td>1/V</td>
<td>0</td>
<td>Yes</td>
<td>Body effect coefficient of DIBL effect coefficients</td>
</tr>
<tr>
<td>DROUT</td>
<td>-</td>
<td>0.56</td>
<td>Yes</td>
<td>Length dependence coefficient of the DIBL correction parameter in $R_{out}$</td>
</tr>
<tr>
<td>PSCBE1</td>
<td>V/m</td>
<td>4.24e8</td>
<td>Yes</td>
<td>Substrate current induced body effect exponent 1</td>
</tr>
<tr>
<td>PSCBE2</td>
<td>V/m</td>
<td>1.0e-5</td>
<td>Yes</td>
<td>Substrate current induced body effect coefficient 2</td>
</tr>
<tr>
<td>PVAG</td>
<td>-</td>
<td>0</td>
<td>Yes</td>
<td>Gate dependence of Early voltage</td>
</tr>
<tr>
<td>DELTA</td>
<td>V</td>
<td>0.01</td>
<td>Yes</td>
<td>Effective Vds parameter</td>
</tr>
<tr>
<td>ALPHA0</td>
<td>m/V</td>
<td>0</td>
<td>Yes</td>
<td>First parameter of impact ionization current</td>
</tr>
</tbody>
</table>
### AC and Capacitance Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPART</td>
<td>-</td>
<td>0</td>
<td>No</td>
<td>Charge partitioning rate flag (default deviates from BSIM3V3=0) LEVEL 49 XPART defaults to 1</td>
</tr>
<tr>
<td>CGSO</td>
<td>F/m</td>
<td>p1</td>
<td>No</td>
<td>Non-LDD region source-gate overlap capacitance per unit channel length</td>
</tr>
<tr>
<td>CGDO</td>
<td>F/m</td>
<td>p2</td>
<td>No</td>
<td>Non-LDD region source-gate overlap capacitance per unit channel length</td>
</tr>
<tr>
<td>CGBO</td>
<td>F/m</td>
<td>0</td>
<td>No</td>
<td>Gate-bulk overlap capacitance per unit channel length</td>
</tr>
<tr>
<td>CGS1</td>
<td>F/m</td>
<td>0.0</td>
<td>Yes</td>
<td>Lightly doped source-gate overlap region capacitance</td>
</tr>
<tr>
<td>CGD1</td>
<td>F/m</td>
<td>0.0</td>
<td>Yes</td>
<td>Lightly doped drain-gate overlap region capacitance</td>
</tr>
<tr>
<td>CKAPPA</td>
<td>F/m</td>
<td>0.6</td>
<td>Yes</td>
<td>Coefficient for lightly doped region overlap capacitance fringing field capacitance</td>
</tr>
<tr>
<td>CF</td>
<td>F/m</td>
<td>(see Note3)</td>
<td>Yes</td>
<td>Fringing field capacitance</td>
</tr>
<tr>
<td>CLC</td>
<td>m</td>
<td>0.1e-6</td>
<td>Yes</td>
<td>Constant term for the short channel model</td>
</tr>
</tbody>
</table>
### Length and Width Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WINT</td>
<td>m</td>
<td>0.0</td>
<td>No</td>
<td>Width offset fitting parameter from I-V without bias</td>
</tr>
<tr>
<td>WLN</td>
<td>-</td>
<td>1.0</td>
<td>No</td>
<td>Power of length dependence of width offset</td>
</tr>
<tr>
<td>WW</td>
<td>m&lt;sup&gt;WWN&lt;/sup&gt;</td>
<td>0.0</td>
<td>No</td>
<td>Coefficient of width dependence for width offset</td>
</tr>
<tr>
<td>WWWN</td>
<td>-</td>
<td>1.0</td>
<td>No</td>
<td>Power of width dependence of width offset</td>
</tr>
<tr>
<td>WWL</td>
<td>m&lt;sup&gt;WWN&lt;/sup&gt; *m&lt;sup&gt;WLN&lt;/sup&gt;</td>
<td>0.0</td>
<td>No</td>
<td>Coefficient of length and width cross term for width offset</td>
</tr>
<tr>
<td>DWG</td>
<td>m/V</td>
<td>0.0</td>
<td>Yes</td>
<td>Coefficient of W&lt;sub&gt;eff&lt;/sub&gt;’s gate dependence</td>
</tr>
<tr>
<td>DWB</td>
<td>m/V&lt;sup&gt;1/2&lt;/sup&gt;</td>
<td>0.0</td>
<td>Yes</td>
<td>Coefficient of W&lt;sub&gt;eff&lt;/sub&gt;’s substrate body bias dependence</td>
</tr>
<tr>
<td>LINT</td>
<td>m</td>
<td>0.0</td>
<td>No</td>
<td>Length offset fitting parameter from I-V without bias</td>
</tr>
<tr>
<td>LL</td>
<td>m&lt;sup&gt;LLN&lt;/sup&gt;</td>
<td>0.0</td>
<td>No</td>
<td>Coefficient of length dependence for length offset</td>
</tr>
<tr>
<td>LLN</td>
<td>-</td>
<td>1.0</td>
<td>No</td>
<td>Power of length dependence of length offset</td>
</tr>
<tr>
<td>LW</td>
<td>m&lt;sup&gt;LWN&lt;/sup&gt;</td>
<td>0.0</td>
<td>No</td>
<td>Coefficient of width dependence for length offset</td>
</tr>
</tbody>
</table>

CLE - 0.6 Yes Exponential term for the short channel model
VFBCV -1.0 Yes Flat band voltage used only in CAPMOD=0 C-V calculations
**Temperature Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KT1</td>
<td>V</td>
<td>-0.11</td>
<td>Yes</td>
<td>Temperature coefficient for Vth</td>
</tr>
<tr>
<td>KT1L</td>
<td>m-V</td>
<td>0.0</td>
<td>Yes</td>
<td>Temperature coefficient for channel length dependence of Vth</td>
</tr>
<tr>
<td>KT2</td>
<td>T</td>
<td>0.022</td>
<td>Yes</td>
<td>Body bias coefficient of Vth temperature effect</td>
</tr>
<tr>
<td>UTE</td>
<td>T</td>
<td>-1.5</td>
<td>Yes</td>
<td>Mobility temperature exponent</td>
</tr>
<tr>
<td>UA1</td>
<td>m/V</td>
<td>4.31e-9</td>
<td>Yes</td>
<td>Temperature coefficient for UA</td>
</tr>
<tr>
<td>UB1</td>
<td>(m/V)^2</td>
<td>-7.61e-18</td>
<td>Yes</td>
<td>Temperature coefficient for UB</td>
</tr>
<tr>
<td>UC1</td>
<td>m/V^2</td>
<td>-5.69e-11</td>
<td>Yes</td>
<td>Temperature coefficient for UC</td>
</tr>
<tr>
<td>AT</td>
<td>m/sec</td>
<td>3.3e4</td>
<td>Yes</td>
<td>Temperature coefficient for saturation velocity</td>
</tr>
<tr>
<td>PRT</td>
<td>ohm·um</td>
<td>0</td>
<td>Yes</td>
<td>Temperature coefficient for RDSW</td>
</tr>
<tr>
<td>XTI</td>
<td>-</td>
<td>3.0</td>
<td>No</td>
<td>Junction current temperature exponent</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWN</td>
<td>-</td>
<td>1.0</td>
<td>No</td>
<td>Power of width dependence of length offset</td>
</tr>
<tr>
<td>LWL</td>
<td>m^(LWN * m^LWN)</td>
<td>0.0</td>
<td>No</td>
<td>Coefficient of length and width cross term for length offset</td>
</tr>
<tr>
<td>DLC</td>
<td>m</td>
<td>LINT</td>
<td>No</td>
<td>Length offset fitting parameter from CV</td>
</tr>
<tr>
<td>DWC</td>
<td>m</td>
<td>WINT</td>
<td>No</td>
<td>Width offset fitting parameter from CV</td>
</tr>
</tbody>
</table>
### Bin Description Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMIN</td>
<td>m</td>
<td>0.0</td>
<td>No</td>
<td>Maximum channel length</td>
</tr>
<tr>
<td>LMAX</td>
<td>m</td>
<td>1.0</td>
<td>No</td>
<td>Maximum channel length</td>
</tr>
<tr>
<td>WMIN</td>
<td>m</td>
<td>0.0</td>
<td>No</td>
<td>Minimum channel width</td>
</tr>
<tr>
<td>WMAX</td>
<td>m</td>
<td>1.0</td>
<td>No</td>
<td>Maximum channel width</td>
</tr>
<tr>
<td>BINUNIT</td>
<td></td>
<td></td>
<td></td>
<td>Assumes weff, leff, wref, lref units are in microns when BINUNIT=1 or meters otherwise</td>
</tr>
</tbody>
</table>

### Process Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAMMA1</td>
<td>V^{1/2}</td>
<td>see Note 8</td>
<td>Yes</td>
<td>Body effect coefficient near the surface</td>
</tr>
<tr>
<td>GAMMA2</td>
<td>V^{1/2}</td>
<td>see Note 9</td>
<td>Yes</td>
<td>Body effect coefficient in the bulk</td>
</tr>
<tr>
<td>VBX</td>
<td>V</td>
<td>see Note 10</td>
<td>Yes</td>
<td>VBX at which the depletion region width equals XT</td>
</tr>
<tr>
<td>XT</td>
<td>m</td>
<td>1.55e-7</td>
<td>Yes</td>
<td>Doping depth</td>
</tr>
</tbody>
</table>
## Noise Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIOA</td>
<td>-</td>
<td>1.0e20 nmos 9.9e18 pmos</td>
<td>No</td>
<td>Body effect coefficient near the surface</td>
</tr>
<tr>
<td>NOIB</td>
<td>-</td>
<td>5.0e4 nmos 2.4e3 pmos</td>
<td>No</td>
<td>Body effect coefficient in the bulk</td>
</tr>
<tr>
<td>NOIC</td>
<td>-</td>
<td>-1.4e-12 nmos 1.4e-12 pmos</td>
<td>No</td>
<td>VBX at which the depletion region width equals XT</td>
</tr>
<tr>
<td>EM</td>
<td>V/m</td>
<td>4.1e7</td>
<td>No</td>
<td>Flicker noise parameter</td>
</tr>
<tr>
<td>AF</td>
<td>-</td>
<td>1.0</td>
<td>No</td>
<td>Flicker noise exponent</td>
</tr>
<tr>
<td>KF</td>
<td>-</td>
<td>0.0</td>
<td>No</td>
<td>Flicker noise coefficient</td>
</tr>
<tr>
<td>EF</td>
<td>-</td>
<td>1.0</td>
<td>No</td>
<td>Flicker noise frequency exponent</td>
</tr>
</tbody>
</table>

**Note:** See also “Using Noise Models” on page 20-99, for Hspice noise model usage (Hspice parameter NLEV overrides Berkeley NOIMOD).
## Junction Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td>-</td>
<td>10</td>
<td>No</td>
<td>Area calculation method selector (Hspice specific) ACM=0-3 uses Hspice junction models ACM=10-13 uses Berkeley junction models LEVEL 49 ACM defaults to 0</td>
</tr>
<tr>
<td>JS</td>
<td>A/m²</td>
<td>0.0</td>
<td>No</td>
<td>Bulk junction saturation current (Default deviates from BSIM3v3 = 1.0e⁻⁴)</td>
</tr>
<tr>
<td>JSW</td>
<td>A/m²</td>
<td>0.0</td>
<td>No</td>
<td>Sidewall bulk junction saturation current</td>
</tr>
<tr>
<td>NJ</td>
<td>-</td>
<td>1</td>
<td>No</td>
<td>Emission coefficient (used only with Berkeley junction model, i.e., ACM=10-13)</td>
</tr>
<tr>
<td>N</td>
<td>-</td>
<td>1</td>
<td>No</td>
<td>Emission coefficient (Hspice-specific), (used only with Hspice junction model, i.e., ACM=0-3)</td>
</tr>
<tr>
<td>CJ</td>
<td>F/m²</td>
<td>5.79e⁻⁴</td>
<td>No</td>
<td>Zero-bias bulk junction capacitance (Default deviates from BSIM3v3 = 5.0e⁻⁴)</td>
</tr>
<tr>
<td>CJSW</td>
<td>F/m</td>
<td>0.0</td>
<td>No</td>
<td>Zero-bias sidewall bulk junction capacitance (Default deviates from BSIM3v3 = 5.0e⁻¹⁰)</td>
</tr>
<tr>
<td>CJSWG</td>
<td>F/m</td>
<td>CJSW</td>
<td>No</td>
<td>Zero-bias gate-edge sidewall bulk junction capacitance (only used with Berkeley junction model, i.e., ACM=10-13)</td>
</tr>
<tr>
<td>CJGATE</td>
<td>F/m</td>
<td>CJSW</td>
<td>No</td>
<td>Zero-bias gate-edge sidewall bulk junction capacitance (Hspice-specific) (only used with ACM=3!)</td>
</tr>
<tr>
<td>PB, PHIB</td>
<td>V</td>
<td>1.0</td>
<td>No</td>
<td>Bulk junction contact potential</td>
</tr>
<tr>
<td>PBSW</td>
<td>V</td>
<td>1.0</td>
<td>No</td>
<td>Sidewall bulk junction contact potential</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 47-62

LEVELs 49 and 53 BSIM3v3 MOS Models

Note: See “Using a MOSFET Diode Model” on page 20-26 for Hspice junction diode model usage.

NonQuasi-Static (NQS) Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELM</td>
<td>-</td>
<td>5.0</td>
<td>Yes</td>
<td>Elmore constant</td>
</tr>
</tbody>
</table>
## Version 3.2 Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOXM</td>
<td>m</td>
<td>TOX</td>
<td>No</td>
<td>Reference gate oxide thickness</td>
</tr>
<tr>
<td>VFB</td>
<td>V</td>
<td>See Note 11</td>
<td>Yes</td>
<td>DC flatband voltage</td>
</tr>
<tr>
<td>NOFF</td>
<td>-</td>
<td>1.0</td>
<td>Yes</td>
<td>I-V parameter for weak to strong inversion transition</td>
</tr>
<tr>
<td>VOFFCV</td>
<td>-</td>
<td>0.0</td>
<td>Yes</td>
<td>C-V parameter for weak to strong inversion transition</td>
</tr>
<tr>
<td>JTH</td>
<td>A</td>
<td>0.1</td>
<td>No</td>
<td>Diode limiting current</td>
</tr>
<tr>
<td>ALPHA1</td>
<td>V^{-1}</td>
<td>0.0</td>
<td>Yes</td>
<td>Substrate current parameter</td>
</tr>
<tr>
<td>ACDE</td>
<td>m/V</td>
<td>1.0</td>
<td>Yes</td>
<td>Exponential coefficient for charge thickness in the accumulation and depletion regions</td>
</tr>
<tr>
<td>MOIN</td>
<td>m/V</td>
<td>15.0</td>
<td>Yes</td>
<td>Coefficient for gate-bias dependent surface potential</td>
</tr>
<tr>
<td>TPB</td>
<td>V/K</td>
<td>0.0</td>
<td>No</td>
<td>Temperature coefficient of PB</td>
</tr>
<tr>
<td>TPBSW</td>
<td>V/K</td>
<td>0.0</td>
<td>No</td>
<td>Temperature coefficient of PBSW</td>
</tr>
<tr>
<td>TPBSWG</td>
<td>V/K</td>
<td>0.0</td>
<td>No</td>
<td>Temperature coefficient of PBSWG</td>
</tr>
<tr>
<td>TCJ</td>
<td>V/K</td>
<td>0.0</td>
<td>No</td>
<td>Temperature coefficient of CJ</td>
</tr>
<tr>
<td>TCJSW</td>
<td>V/K</td>
<td>0.0</td>
<td>No</td>
<td>Temperature coefficient of CJSW</td>
</tr>
<tr>
<td>TCJSWG</td>
<td>V/K</td>
<td>0.0</td>
<td>No</td>
<td>Temperature coefficient of CJSWG</td>
</tr>
<tr>
<td>LLC</td>
<td>m^lin</td>
<td>LL</td>
<td>No</td>
<td>Coefficient of length dependence for C-V channel length offset</td>
</tr>
<tr>
<td>LWC</td>
<td>m^lwn</td>
<td>LW</td>
<td>No</td>
<td>Coefficient of width dependence for C-V channel length offset</td>
</tr>
<tr>
<td>LWLC</td>
<td>m^{lin+lwn}</td>
<td>LWL</td>
<td>No</td>
<td>Coefficient of length and width for C-V channel length offset</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 47-62

LEVELs 49 and 53 BSIM3v3 MOS Models

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Bin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WLC</td>
<td>mWL</td>
<td>WL</td>
<td>No</td>
<td>Coefficient of length dependence for C-V channel width offset</td>
</tr>
<tr>
<td>WWC</td>
<td>mWW</td>
<td>WW</td>
<td>No</td>
<td>Coefficient of width dependence for C-V channel width offset</td>
</tr>
<tr>
<td>WWLC</td>
<td>mWL+WW</td>
<td>WWL</td>
<td>No</td>
<td>Coefficient of length and width cross terms for C-V channel width offset</td>
</tr>
</tbody>
</table>

Notes:

1. If \( C_{gs0} \) is not given, it is calculated as follows:
   If \( dlc \) is given and is greater than 0.0, then,
   \[ c_{gs0} = p1 = \max(0, dlc \times c_{ox} - c_{gs1}) \]
   Otherwise, \( c_{gs0} = 0.6 \times x_j \times c_{ox} \)

2. If \( C_{gdo} \) is not given, it is calculated as follows:
   if \( dlc \) is given and is greater than 0.0, then,
   \[ c_{gdo} = p2 = \max(0, dlc \times c_{ox} - c_{gd1}) \]
   Otherwise \( c_{gdo} = 0.6 \times x_j \times c_{ox} \)

3. If \( C_f \) is not given, it is calculated using:
   \[ C_f = \frac{2 \varepsilon_{ox}}{\pi} \log \left( 1 + \frac{4 \times 10^{-7}}{T_{ox}} \right) \]

4. If \( V_{th0} \) is not specified in the .MODEL statement, it is calculated with \( V_{fb} = -1 \), using:
   \[ V_{th0} = V_{fb} + \phi_s + K_1 \frac{\phi_s}{\phi_s} \]

5. If \( K_1 \) and \( K_2 \) are not given, they are calculated using:
   \[ K_1 = \gamma_2 + 2 \gamma_2 \frac{\phi_s - V_{bs}}{\phi_s} \]
   \[ K_2 = \frac{(\gamma_2 - \gamma_1)(\frac{\phi_s - V_{bs}}{\phi_s} - \frac{\phi_s}{\phi_s})}{2 \frac{\phi_s}{\phi_s} (\frac{\phi_s - V_{bm}}{\phi_s} - \frac{\phi_s}{\phi_s}) + V_{bm}} \]
6. If \( n \text{ ch} \) is not given, and \( \text{GAMMA}_1 \) is given, \( n \text{ ch} \) is calculated from:

\[
n \text{ ch} = \frac{\text{GAMMA}_1^2 \epsilon \text{Ox}^2}{2q\epsilon_s i}
\]

If both \( n \text{ ch} \) and \( \text{GAMMA}_1 \) are not given, \( n \text{ ch} \) defaults to \( 1.7e17 \) per cubic meter and \( \text{GAMMA}_1 \) is calculated from \( n \text{ ch} \).

7. If \( \Phi \) is not given, it is calculated using:

\[
\Phi_s = \frac{k_B T}{q} \log \left( \frac{n \text{ ch}}{n_i} \right)
\]

\[
n_i = 1.45 \times 10^{10} \left( \frac{T}{300.15} \right)^{1.5} \exp \left( 21.5565981 - \frac{qE_g(T)}{2k_B T} \right)
\]

\[
E_g(T) = 1.16 - \frac{7.02 \times 10^{-4} T^2}{T + 1108}
\]

8. If \( \text{GAMMA}_1 \) is not given, it is calculated using:

\[
\text{GAMMA}_1 = \frac{\sqrt{2q\epsilon_s \text{n ch}}}{C_{ox}}
\]

9. If \( \text{GAMMA}_2 \) is not given, it is calculated using:

\[
\text{GAMMA}_2 = \frac{\sqrt{2q\epsilon_s \text{n sub}}}{C_{ox}}
\]

10. If \( V_{bx} \) is not given, it is calculated using:

\[
V_{bx} = \Phi_s - \frac{qn \text{ ch}^2 \epsilon_i}{2\epsilon_s i}
\]

11. There are three ways for the BSIM3 model to calculate \( V_{th} \):
- Using \( K1 \) and \( K2 \) values that are user specified
- Using \( \text{GAMMA}_1, \text{GAMMA}_2, VBM, \) and \( VBX \) values entered in the .MODEL statement
- Using \( NPEAK, NSUB, XT, \) and \( VBM \) values that are user specified
The model parameter U0 can be entered in meters or centimeters. U0 is converted to m²/Vsec as follows: if U0 is greater than 1, it is multiplied by 1e-4. The parameter NSUB must be entered in cm⁻³ units.

Specify a negative value of VTH0 for p-channel in the .MODEL statement.

The impact ionization current determined by the model parameters PSCBE1 and PSCBE2 contributes to the bulk current.

**Parameter Range Limits**

Star-Hspice will report either warning or fatal error when BSIM3v3 parameters fall outside predefined ranges. These range limitations prevent or at least warn of potential numerical problems. LEVEL 53 follows exactly the BSIM3v3 range limit reporting scheme. LEVEL 49 deviates from the BSIM3v3 scheme as noted in the comments column of Model Parameter Range Limit.

To control the maximum number of Star-Hspice warning messages printing to the output file use:

```
.OPTION WARNLIMIT=#
```

where # is the maximum number of warning messages Star-Hspice will report. The default WARNLIMIT value is 1. In some cases (as noted in the following table) parameters are checked only when the model parameter PARMAMCHK=1 is set.

**Model Parameter Range Limit**

<table>
<thead>
<tr>
<th>Name</th>
<th>Limits</th>
<th>Comments</th>
</tr>
</thead>
</table>
| TOX  | <= 0 Fatal  
|      | < 10-9 Warn if parmchk=1 |          |
| TOXM | <= 0 Fatal  
|      | < 10-9 Warn if parmchk=1 |          |
| XJ   | <= 0 Fatal |          |
## LEVELs 49 and 53 BSIM3v3 MOS Models

**Selecting MOSFET Models: LEVEL 47-62**

<table>
<thead>
<tr>
<th>Name</th>
<th>Limits</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>NGATE</td>
<td>&lt; 0 Fatal</td>
<td>if ( &gt; 10^{23} ), NGATE is multiplied by 10^-6. This is done prior to the other limit checks. LEVEL 49 gives</td>
</tr>
<tr>
<td></td>
<td>( &gt; 10^{25} ) Fatal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( &lt;= 10^{18} ) Fatal if parmchk=1</td>
<td></td>
</tr>
<tr>
<td>NSUB</td>
<td>( &lt;= 0 ) Fatal</td>
<td>NSUB is ignored if k1,k2 are defined</td>
</tr>
<tr>
<td></td>
<td>( &lt;= 10^{14} ) Warn if parmchk=1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( &gt;= 10^{21} ) Warn if parmchk=1</td>
<td></td>
</tr>
<tr>
<td>NCH</td>
<td>( &lt;= 0 ) Fatal</td>
<td>if ( &gt; 10^{20} ), NCH is multiplied by 10^-6. This is done prior to the other limit checks.</td>
</tr>
<tr>
<td></td>
<td>( &lt;= 10^{15} ) Warn if parmchk=1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( &gt;= 10^{21} ) Warn if parmchk=1</td>
<td></td>
</tr>
<tr>
<td>NLX</td>
<td>( &lt; -L_{eff} ) Fatal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( &lt; 0 ) Warn if parmchk=1</td>
<td></td>
</tr>
<tr>
<td>W0</td>
<td>( = -W_{eff} ) Fatal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( w_0 + W_{eff} &lt; 10^{-7} ) Warn if parmchk=1</td>
<td></td>
</tr>
<tr>
<td>DVT1W</td>
<td>( &lt; 0 ) Fatal</td>
<td>( &lt; 0 ) LEVEL 49 gives Warn</td>
</tr>
<tr>
<td>DVT0</td>
<td>( &lt; 0 ) Warn if parmchk=1</td>
<td></td>
</tr>
<tr>
<td>DVT1</td>
<td>( &lt; 0 ) Fatal</td>
<td>( &lt; 0 ) LEVEL 49 gives Warn</td>
</tr>
<tr>
<td>ETA0</td>
<td>( &lt;= 0 ) Warn if parmchk=1</td>
<td></td>
</tr>
<tr>
<td>DSUB</td>
<td>( &lt; 0 ) Fatal</td>
<td>( &lt; 0 ) LEVEL 49 gives Warn</td>
</tr>
<tr>
<td>VBM</td>
<td>( &lt; 0 ) Fatal</td>
<td>Ignored if K1,K2 are defined</td>
</tr>
<tr>
<td>U0</td>
<td>( &lt;= 0 ) Fatal</td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>( = -W_{eff} ) Fatal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( B_1 + W_{eff} &lt; 10^{-7} ) Warn if parmchk=1</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Limits</td>
<td>Comments</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>VSAT</td>
<td>&lt;= 0 Fatal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt; 10^3 Warn if paramchk==1</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>-</td>
<td>See a2 conditions</td>
</tr>
<tr>
<td>A2</td>
<td>&lt; 0.01 Warn and reset a2=0.01 if</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paramchk=1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt; 1 Warn and reset a2=1,a1=0 if</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paramchk=1</td>
<td></td>
</tr>
<tr>
<td>DELTA</td>
<td>&lt; 0 Fatal</td>
<td></td>
</tr>
<tr>
<td>RDSW</td>
<td>&lt; 0.001 Warn if paramchk==1 and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>reset rdsw=0</td>
<td></td>
</tr>
<tr>
<td>NFACTOR</td>
<td>&lt; 0 Warn if paramchk==1</td>
<td></td>
</tr>
<tr>
<td>CDSC</td>
<td>&lt; 0 Warn if paramchk==1</td>
<td></td>
</tr>
<tr>
<td>CDSCD</td>
<td>&lt; 0 Warn if paramchk==1</td>
<td></td>
</tr>
<tr>
<td>PCLM</td>
<td>&lt;= 0 Fatal</td>
<td></td>
</tr>
<tr>
<td>PDIBLC1</td>
<td>&lt; 0 Warn if paramchk==1</td>
<td></td>
</tr>
<tr>
<td>PDIBLC2</td>
<td>&lt; 0 Warn if paramchk==1</td>
<td></td>
</tr>
<tr>
<td>PS</td>
<td>&lt; Weff Warn</td>
<td></td>
</tr>
<tr>
<td>DROUT</td>
<td>&lt; 0 Fatal if paramchk==1</td>
<td>LEVEL 49 gives Warn</td>
</tr>
<tr>
<td>PSCBE2</td>
<td>&lt;= 0 warn if paramchk==1</td>
<td></td>
</tr>
<tr>
<td>CGS0</td>
<td>&lt; 0 Warn and reset to 0 if</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paramchk==1</td>
<td></td>
</tr>
<tr>
<td>CGD0</td>
<td>&lt; 0 Warn and reset to 0 if</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paramchk==1</td>
<td></td>
</tr>
<tr>
<td>CGB0</td>
<td>&lt; 0 Warn and reset to 0 if</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paramchk==1</td>
<td></td>
</tr>
<tr>
<td>ACDE</td>
<td>&lt; 0.4, &gt; 1.6 Warn</td>
<td></td>
</tr>
</tbody>
</table>
**LEVEL 49, 53 Equations**

The effective channel length and width used in all model equations are:

\[ L_{\text{eff}} = L_{\text{drawn}} - 2dL \]

\[ W_{\text{eff}} = W_{\text{drawn}} - 2dW \]

\[ \dot{W}_{\text{eff}} = W_{\text{drawn}} - 2\dot{d}W \]

\[ W_{\text{drawn}} = W \times \text{WMULT} + \text{XW} \]

\[ L_{\text{drawn}} = L \times \text{LMULT} + \text{XL} \]
Selecting MOSFET Models: LEVEL 47-62 LEVELs 49 and 53 BSIM3v3 MOS Models

where the unprimed $W_{eff}$ is bias-dependent, and the primed quantity is bias-independent.

\[
dW = dW' + dW_g V_{gsteff} + dW_b(\sqrt{\phi_s - V_{bseff}} - \sqrt{\phi_s})
\]

\[
dW' = W_{int} + \frac{W_L}{L_{WLN}} + \frac{W_W}{W_{WNN}} + \frac{W_{WL}}{L_{WLW}}
\]

\[
dl = L_{int} + \frac{L_W}{L_{LLN}} + \frac{L_W}{L_{WWN}} + \frac{L_{WL}}{L_{WLN}}
\]

For C-V calculations $dW'$ is replaced with

\[
dW' = DWC + \frac{W_{LC}}{L_{WLN}} + \frac{W_{WC}}{W_{WNN}} + \frac{W_{WLC}}{L_{WLW}}
\]

and $dL'$ is replaced with

\[
dl = DLC + \frac{L_{LC}}{L_{LLN}} + \frac{L_{WC}}{L_{WWN}} + \frac{L_{WLC}}{L_{WLN}}
\]

Note: A detailed discussion of the BSIM3 Version 3 equations is available from the BSIM3 site: http://www-device.eecs.berkeley.edu/~bsim3/get.html

**MODEL CARDS NMOS Model**

```plaintext
.model nch nmos LEVEL=49
+ Tnom=27.0
+ nch=1.024685E+17 tox=1.00000E-08 xj=1.00000E-07
+ lint=3.75860E-08 wint=-2.02101528644562E-07
+ vth0=.6094574 k1=.4976366 a1=.0332883 a2=.0332883 a3=.0332883 a4=.0332883
+ vth0=6094574 k1=3.75860E-08 wint=-2.02101528644562E-07
+ vth0=.6094574 k1=.5341038 k2=1.703463E-03 k3=-17.24589
+ dvt0=.1767506 dvt1=.5109418 dvt2=-.05
+ nlx=9.979638E-08 w0=1e-6
+ k3b=4.139039
+ keta=-2.195445E-02 u0=307.2991 prwb=-2.24e-4
+ a0=.4976366
+ keta=-2.195445E-02 a1=.0332883 a2=.9
```

+ voff=-9.623903E-02 nFactor=.8408191 cit=3.994609E-04
+ cdsc=1.130797E-04
+ cdscb=2.4e-5
+ eta0=.0145072 etab=-3.870303E-03
+ dsub=.4116711
+ pclm=1.813153 pdiblc1=2.003703E-02 pdiblc2=.00129051
+ pdiblcb=-1.034e-3
+ drout=.4380235 pscbe1=5.752058E+08 pscbe2=7.510319E-05
+ pvag=.6370527 prt=68.7 ngate=1.e20 alpha0=1.e-7 beta0=28.4
+ prwg=-0.001 ags=1.2
+ dvt0w=0.58 dvtlw=5.3e6 dvt2w=-0.0032
+ kt1=-.3 kt2=-.03
+ at=33000
+ ute=-1.5
+ ual=4.31E-09 ub1=7.61E-18 ucl=-2.378e-10
+ kt1=1e-8
+ wr=1 b0=1e-7 bl=1e-7 dwg=5e-8 dwb=2e-8 delta=0.015
+ cgdl=1e-10 cgs1=1e-10 cgbo=1e-10 xpart=0.0
+ cgdo=0.4e-9 cgso=0.4e-9
+ clc=0.1e-6
+ cle=0.6
+ cktappa=0.6
PMOS Model

This is an example of a PMOS model. Note that VTH0 is negative.

```
.model pch PMOS LEVEL=49
 + Tnom=27.0
 + nch=5.73068E+16 tox=1.00000E-08 xj=1.00000E-07
 + lint=8.195860E-08 wint=-1.821562E-07
 + vth0=-.86094574 k1=.341038 k2=2.703463E-02 k3=12.24589
 + dvt0=.767506 dvt1=.65109418 dvt2=-0.145
 + n1x=1.979638E-07 w0=1.1e-6
 + k3b=-2.4139039
 + vsat=60362.05 ua=1.348481E-09 ub=3.178541E-19 uc=1.1623e-10
 + rdsw=498.873 u0=137.2991 prwb=-1.2e-5
 + a0=.3276366
 + keta=-1.8195445E-02 a1=.0232883 a2=.9
 + voff=-6.623903E-02 nFactor=1.0408191 cit=4.994609E-04
 + cdsc=1.030797E-3
 + cdscb=2.84e-4
 + eta0=.0245072 etab=-1.570303E-03
 + dsub=.24116711
 + pctlm=2.6813153 pdiblc1=4.003703E-02 pdiblc2=.00329051
 + pdiblcb=-2.e-4
 + droute=.1380235 pscbel=0 pscbe2=1.e-28
 + pvag=-.16370527
 + prwg=-0.001 ags=1.2
 + dvt0w=0.58 dvt1w=5.3e6 dvt2w=-0.0032
 + kt1=-.3 kt2=-.03 prt=76.4
 + at=33000
 + ute=-1.5
 + ual=4.31E-09 ubl=7.61E-18 ucl=2.378e-10
 + kt11=0
 + wr=1 b0=1e-7 b1=1e-7 dwg=5e-8 dwb=2e-8 delta=0.015
 + cgdl=1e-10 cgs1=1e-10 cgbo=1e-10 xpart=0.0
 + cgdo=0.4e-9 cgso=0.4e-9
 + clc=0.1e-6
 + cie=0.6
 + ckappa=0.6
```
LEVEL 50 Philips MOS9 Model


The model has been installed in its entirety, except for the gate noise current.

The Star-Hspice ACM Parasitic Diode Model, using parameters JS, JSW, N, CJ, CJSW, CJGATE, MJ, MIJSW, PB, PHP, ACM, and HDIF, has been added. The older parameter IS is not used. The Philips JUNCAP Parasitic Diode Model was added to the Star-Hspice, 99.2 Release. The model parameter JUNCAP=1 selects the JUNCAP Model, JUNCAP=0 (default) selects the Hspice ACM Model. For additional information regarding the MOS Model-9 please see http://www-us.semiconductors.com/Philips_Models.

LEVEL 50 Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default (N)</th>
<th>Default (P)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LER</td>
<td>m</td>
<td>1.1e-6</td>
<td>1.25e-6</td>
<td>Reference Leff</td>
</tr>
<tr>
<td>WER</td>
<td>m</td>
<td>20.0e-6</td>
<td>20.0e-6</td>
<td>Reference Weff</td>
</tr>
<tr>
<td>LVAR</td>
<td>m</td>
<td>-220.0e-9</td>
<td>-460.0e-9</td>
<td>Variation in gate length</td>
</tr>
<tr>
<td>LAP</td>
<td>m</td>
<td>100.0e-9</td>
<td>25.0e-9</td>
<td>Lateral diffusion per side</td>
</tr>
<tr>
<td>WVAR</td>
<td>m</td>
<td>-25.0e-9</td>
<td>-130.0e-9</td>
<td>Variation in active width</td>
</tr>
<tr>
<td>WOT</td>
<td>m</td>
<td>0.0</td>
<td>0.0</td>
<td>Channel-stop diffusion per side</td>
</tr>
<tr>
<td>TR</td>
<td>°C</td>
<td>21.0</td>
<td>21.0</td>
<td>Reference temperature for model</td>
</tr>
<tr>
<td>VTOR</td>
<td>V</td>
<td>730.0e-3</td>
<td>1.1</td>
<td>Threshold voltage at zero bias</td>
</tr>
<tr>
<td>STVTO</td>
<td>V/K</td>
<td>-1.2e-3</td>
<td>-1.7e-3</td>
<td>Temperature dependence of VTO</td>
</tr>
<tr>
<td>SLVTO</td>
<td>Vm</td>
<td>-135.0e-9</td>
<td>35.0e-9</td>
<td>Length dependence of VTO</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default (N)</td>
<td>Default (P)</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>SL2VTO</td>
<td>Vm²</td>
<td>0.0</td>
<td>0.0</td>
<td>Second length dependence of VTO</td>
</tr>
<tr>
<td>SWVTO</td>
<td>Vm</td>
<td>130.0e-9</td>
<td>50.0e-9</td>
<td>Width dependence of VTO</td>
</tr>
<tr>
<td>KOR</td>
<td>V⁻¹/²</td>
<td>650.0e-3</td>
<td>470.0e-3</td>
<td>Low-back-bias body factor</td>
</tr>
<tr>
<td>SLKO</td>
<td>V⁻¹/²m</td>
<td>-130.0e-9</td>
<td>-200.0e-9</td>
<td>Length dependence of KO</td>
</tr>
<tr>
<td>SWKO</td>
<td>V⁻¹/²m</td>
<td>2.0e-9</td>
<td>115.0e-9</td>
<td>Width dependence of KO</td>
</tr>
<tr>
<td>KR</td>
<td>V⁻¹/²</td>
<td>110.0e-3</td>
<td>470.0e-3</td>
<td>High-back-bias body factor</td>
</tr>
<tr>
<td>SLK</td>
<td>V⁻¹/²m</td>
<td>-280.0e-9</td>
<td>-200.0e-9</td>
<td>Length dependence of K</td>
</tr>
<tr>
<td>SWK</td>
<td>V⁻¹/²m</td>
<td>275.0e-9</td>
<td>115.0e-9</td>
<td>Width dependence of K</td>
</tr>
<tr>
<td>PHIBR</td>
<td>V</td>
<td>650.0e-3</td>
<td>650.0e-3</td>
<td>Strong inversion surface potential</td>
</tr>
<tr>
<td>VSBXR</td>
<td>V</td>
<td>660.0e-3</td>
<td>0.0</td>
<td>Transition voltage for dual-k-factor model</td>
</tr>
<tr>
<td>SLVSBX</td>
<td>Vm</td>
<td>0.0</td>
<td>0.0</td>
<td>Length dependence of VSBX</td>
</tr>
<tr>
<td>SWVSBX</td>
<td>Vm</td>
<td>-675.0e-9</td>
<td>0.0</td>
<td>Width dependence of VSBX</td>
</tr>
<tr>
<td>BETSQA</td>
<td>AV⁻²</td>
<td>83.0e-6</td>
<td>26.1e-6</td>
<td>Gain factor of infinite square transistor</td>
</tr>
<tr>
<td>ETABET</td>
<td>-</td>
<td>1.6</td>
<td>1.6</td>
<td>Exponent of temperature dependence of gain factor</td>
</tr>
<tr>
<td>THE1R</td>
<td>V⁻¹</td>
<td>190.0e-3</td>
<td>190.0e-3</td>
<td>Gate-induced mobility reduction coefficient</td>
</tr>
<tr>
<td>STTHE1R</td>
<td>V⁻¹/K</td>
<td>0.0</td>
<td>0.0</td>
<td>Temperature dependence coefficient of THE1R</td>
</tr>
<tr>
<td>SLTHE1R</td>
<td>V⁻¹m</td>
<td>140.0e-9</td>
<td>70.0e-9</td>
<td>Length dependence coefficient of THE1R</td>
</tr>
<tr>
<td>STLTHE1</td>
<td>V⁻¹m/K</td>
<td>0.0</td>
<td>0.0</td>
<td>Temperature dependence of, length dependence of THE1R</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default (N)</td>
<td>Default (P)</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>SWTHE1</td>
<td>V m</td>
<td>-58.0e-9</td>
<td>-80.0e-9</td>
<td>Width dependence coefficient of THE1R</td>
</tr>
<tr>
<td>THE2R</td>
<td>V 1/2</td>
<td>12.0e-3</td>
<td>165.0e-3</td>
<td>Back-bias induced mobility reduction coefficient</td>
</tr>
<tr>
<td>STTHE2R</td>
<td>V 1/2/K</td>
<td>0.0</td>
<td>0.0</td>
<td>Temperature dependence coefficient of THE2R</td>
</tr>
<tr>
<td>SLTHE2R</td>
<td>V 1/2/m</td>
<td>-33.0e-9</td>
<td>-75.0e-9</td>
<td>Length dependence coefficient of THE2R</td>
</tr>
<tr>
<td>STLTHE2</td>
<td>V 1/2 m/K</td>
<td>0.0</td>
<td>0.0</td>
<td>Temperature dependence of, length dependence of THE2R</td>
</tr>
<tr>
<td>SWTHE2</td>
<td>V 1/2/m</td>
<td>30.0e-9</td>
<td>20.0e-9</td>
<td>Width dependence coefficient of THE2R</td>
</tr>
<tr>
<td>THE3R</td>
<td>V 1</td>
<td>145.0e-3</td>
<td>27.0e-3</td>
<td>Lateral field induced mobility reduction coefficient</td>
</tr>
<tr>
<td>STTHE3R</td>
<td>V 1/K</td>
<td>-660.0e-6</td>
<td>0.0</td>
<td>Temperature dependence coefficient of THE3R</td>
</tr>
<tr>
<td>SLTHE3R</td>
<td>V 1/m</td>
<td>185.0e-9</td>
<td>27.0e-9</td>
<td>Length dependence coefficient of THE3R</td>
</tr>
<tr>
<td>STLTHE3</td>
<td>V 1/m/K</td>
<td>-620.0e-12</td>
<td>0.0</td>
<td>Temperature dependence of, length dependence of THE3R</td>
</tr>
<tr>
<td>SWTHE3</td>
<td>V 1/m</td>
<td>20.0e-9</td>
<td>11.0e-9</td>
<td>Width dependence coefficient of THE3R</td>
</tr>
<tr>
<td>GAM1R</td>
<td></td>
<td>145.0e-3</td>
<td>77.0e-3</td>
<td>Drain-induced threshold shift coefficient, for high gate drive</td>
</tr>
<tr>
<td>SLGAM1</td>
<td></td>
<td>160.0e-9</td>
<td>105.0e-9</td>
<td>Length dependence of GAM1R</td>
</tr>
<tr>
<td>SWGAM1</td>
<td></td>
<td>-10.0e-9</td>
<td>-11.0e-9</td>
<td>Width dependence of GAM1R</td>
</tr>
<tr>
<td>ETADSR</td>
<td></td>
<td>600.0e-3</td>
<td>600.0e-3</td>
<td>Exponent of drain dependence of GAM1R</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default (N)</td>
<td>Default (P)</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>-------------</td>
<td>-------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ALPR</td>
<td>-</td>
<td>3.0e-3</td>
<td>44.0e-3</td>
<td>Channel length modulation factor</td>
</tr>
<tr>
<td>ETAALP</td>
<td>-</td>
<td>150.0e-3</td>
<td>170.0e-3</td>
<td>Exponent of length dependence of ALPR</td>
</tr>
<tr>
<td>SLALP</td>
<td>-</td>
<td>-5.65e-3</td>
<td>9.0e-3</td>
<td>Coefficient of length dependence of ALPR</td>
</tr>
<tr>
<td>SWALP</td>
<td>m</td>
<td>1.67e-9</td>
<td>180.0e-12</td>
<td>Coefficient of width dependence of ALPR</td>
</tr>
<tr>
<td>VPR</td>
<td>V</td>
<td>340.0e-3</td>
<td>235.0e-3</td>
<td>Characteristic voltage for channel length modulation</td>
</tr>
<tr>
<td>GAMOOR</td>
<td>-</td>
<td>18.0e-3</td>
<td>7.0e-3</td>
<td>Drain-induced threshold shift coefficient, at zero gate drive, and zero back-bias</td>
</tr>
<tr>
<td>SLGAMOO</td>
<td>m²</td>
<td>20.0e-15</td>
<td>11.0e-15</td>
<td>Length dependence of GAMOOR</td>
</tr>
<tr>
<td>ETAGAMR</td>
<td>-</td>
<td>2.0</td>
<td>1.0</td>
<td>Exponent of back-bias dependence of zero gate-drive, drain-induced threshold shift</td>
</tr>
<tr>
<td>MOR</td>
<td>-</td>
<td>500.0e-3</td>
<td>375.0e-3</td>
<td>Subthreshold slope factor</td>
</tr>
<tr>
<td>STMO</td>
<td>K⁻¹</td>
<td>0.0</td>
<td>0.0</td>
<td>Temperature dependence coefficient of MOR</td>
</tr>
<tr>
<td>SLMO</td>
<td>m¹/²</td>
<td>280.0e-6</td>
<td>47.0e-6</td>
<td>Length dependence coefficient of MOR</td>
</tr>
<tr>
<td>ETAMR</td>
<td>-</td>
<td>2.0</td>
<td>1.0</td>
<td>Exponent of back-bias dependence of subthreshold slope</td>
</tr>
<tr>
<td>ZET1R</td>
<td>-</td>
<td>420.0e-3</td>
<td>1.3</td>
<td>Weak-inversion correction factor</td>
</tr>
<tr>
<td>ETAZET</td>
<td>-</td>
<td>170.0e-3</td>
<td>30.0e-3</td>
<td>Exponent of length dependence of ZET1R</td>
</tr>
<tr>
<td>SLZET1</td>
<td>-</td>
<td>-390.0e-3</td>
<td>-2.8</td>
<td>Length dependence coefficient of ZET1R</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default (N)</td>
<td>Default (P)</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>VSBTR</td>
<td>V</td>
<td>2.1</td>
<td>100.0</td>
<td>Limiting voltage for back-bias dependence</td>
</tr>
<tr>
<td>SLVSBT</td>
<td>Vm</td>
<td>-4.4e-6</td>
<td>0.0</td>
<td>Length dependence of VSBTR</td>
</tr>
<tr>
<td>A1R</td>
<td>-</td>
<td>6.0</td>
<td>10.0</td>
<td>Weak-avalanche current factor</td>
</tr>
<tr>
<td>STA1</td>
<td>K^{-1}</td>
<td>0.0</td>
<td>0.0</td>
<td>Temperature coefficient of A1R</td>
</tr>
<tr>
<td>SLA1</td>
<td>m</td>
<td>1.3e-6</td>
<td>-15.0e-6</td>
<td>Length dependence of A1R</td>
</tr>
<tr>
<td>SWA1</td>
<td>m</td>
<td>3.0e-6</td>
<td>30.0e-6</td>
<td>Width dependence of A1R</td>
</tr>
<tr>
<td>A2R</td>
<td>V</td>
<td>38.0</td>
<td>59.0</td>
<td>Exponent of weak-avalanche current</td>
</tr>
<tr>
<td>SLA2</td>
<td>Vm</td>
<td>1.0e-6</td>
<td>-8.0e-6</td>
<td>Length dependence of A2R</td>
</tr>
<tr>
<td>SWA2</td>
<td>Vm</td>
<td>2.0e-6</td>
<td>15.0e-6</td>
<td>Width dependence of A2R</td>
</tr>
<tr>
<td>A3R</td>
<td>-</td>
<td>650.0e-3</td>
<td>520.0e-3</td>
<td>Factor of minimum drain bias above which avalanche sets in</td>
</tr>
<tr>
<td>SLA3</td>
<td>m</td>
<td>-550.0e-9</td>
<td>-450.0e-9</td>
<td>Length dependence of A3R</td>
</tr>
<tr>
<td>SWA3</td>
<td>m</td>
<td>0.0</td>
<td>-140.0e-9</td>
<td>Width dependence of A3R</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>25.0e-9</td>
<td>25.0e-9</td>
<td>Oxide thickness</td>
</tr>
<tr>
<td>COL</td>
<td>F/m</td>
<td>320.0e-12</td>
<td>320.0e-12</td>
<td>Gate overlap capacitance per unit width</td>
</tr>
<tr>
<td>WDOG</td>
<td>m</td>
<td>0</td>
<td>0</td>
<td>Characteristic drawn gate width below which dogboning appears</td>
</tr>
<tr>
<td>FTHE1</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>Coefficient describing the width dependence of THE1 for W &lt; WDOG</td>
</tr>
<tr>
<td>NFMOD</td>
<td></td>
<td>0</td>
<td></td>
<td>Flicker noise selector 0 selects old flicker noise model added in release 98.4</td>
</tr>
<tr>
<td>NTR</td>
<td>J</td>
<td>24.4e-21</td>
<td>21.1e-21</td>
<td>Thermal noise coefficient</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default (N)</td>
<td>Default (P)</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>NFR</td>
<td>V²</td>
<td>70.0e-12</td>
<td>21.4e-12</td>
<td>Flicker noise coefficient</td>
</tr>
<tr>
<td>NFAR</td>
<td>V⁻¹m⁻²</td>
<td>7.15e+22</td>
<td>1.53xe+22</td>
<td>1st flicker noise coefficient added in release 98.4</td>
</tr>
<tr>
<td>NFBR</td>
<td>V⁻¹m⁻²</td>
<td>2.16e+06</td>
<td>4.06e+06</td>
<td>2nd flicker noise coefficient added in release 98.4</td>
</tr>
<tr>
<td>NFCR</td>
<td>V⁻¹</td>
<td>0.0</td>
<td>2.92e-10</td>
<td>3rd flicker noise coefficient added in release 98.4</td>
</tr>
<tr>
<td>SL3VTO</td>
<td>V</td>
<td>0</td>
<td>0</td>
<td>Third coefficient of the length dependence of V_TO</td>
</tr>
<tr>
<td>SL2KO</td>
<td>V¹/²m²</td>
<td>0</td>
<td>0</td>
<td>Second coefficient of the length dependence of K₀</td>
</tr>
<tr>
<td>SL2K</td>
<td>V¹/²m²</td>
<td>0</td>
<td>0</td>
<td>Second coefficient of the length dependence of K</td>
</tr>
<tr>
<td>LP1</td>
<td>M</td>
<td>1E-6</td>
<td>1E-6</td>
<td>Characteristic length of first profile</td>
</tr>
<tr>
<td>FBET1</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>Relative mobility decrease due to first profile</td>
</tr>
<tr>
<td>LP2</td>
<td>M</td>
<td>1E-8</td>
<td>1E-8</td>
<td>Characteristic length of second profile</td>
</tr>
<tr>
<td>FBET2</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>Relative mobility decrease due to second profile</td>
</tr>
<tr>
<td>GTHE1</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>Parameter that selects either the old (=0) or the new (=1) scaling rule of θ₁</td>
</tr>
<tr>
<td>SL2GAMOO</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>Second coefficient of the length dependence Υ₀₀</td>
</tr>
</tbody>
</table>
## JUNCAP Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JUNCAP</td>
<td>-</td>
<td>0</td>
<td>JUNCAP flag: 0-off, 1-on</td>
</tr>
<tr>
<td>DTA</td>
<td>°C</td>
<td>0.0</td>
<td>Temperature offset of the JUNCAP element with respect to $T_A$</td>
</tr>
<tr>
<td>VR</td>
<td>V</td>
<td>0.0</td>
<td>Voltage at which the parameters have been determined</td>
</tr>
<tr>
<td>JSGBR</td>
<td>$A\cdot m^{-2}$</td>
<td>1.00e-3</td>
<td>Bottom saturation-current density due to electron-hole generation at $V=V_R$</td>
</tr>
<tr>
<td>JSDBR</td>
<td>$A\cdot m^{-2}$</td>
<td>1.00e-3</td>
<td>Bottom saturation-current density due to diffusion from back contact</td>
</tr>
<tr>
<td>JSGSR</td>
<td>$A\cdot m^{-1}$</td>
<td>1.00e-3</td>
<td>Sidewall saturation-current density due to electron-hole generation at $V=V_R$</td>
</tr>
<tr>
<td>JSDSR</td>
<td>$A\cdot m^{-1}$</td>
<td>1.00e-3</td>
<td>Sidewall saturation-current density due to diffusion from back contact</td>
</tr>
<tr>
<td>JSGGR</td>
<td>$A\cdot m^{-1}$</td>
<td>1.00e-3</td>
<td>Gate edge saturation-current density due to electron-hole generation at $V=V_R$</td>
</tr>
<tr>
<td>JSDGGR</td>
<td>$A\cdot m^{-1}$</td>
<td>1.00e-3</td>
<td>Gate edge saturation-current density due to diffusion from back contact</td>
</tr>
<tr>
<td>NB</td>
<td>-</td>
<td>1.00</td>
<td>Emission coefficient of the bottom forward current</td>
</tr>
<tr>
<td>NS</td>
<td>-</td>
<td>1.00</td>
<td>Emission coefficient of the sidewall forward current</td>
</tr>
<tr>
<td>NG</td>
<td>-</td>
<td>1.00</td>
<td>Emission coefficient of the gate edge forward current</td>
</tr>
<tr>
<td>CJBR</td>
<td>$F\cdot m^{-2}$</td>
<td>1.00e-12</td>
<td>Bottom junction capacitance at $V=V_R$</td>
</tr>
<tr>
<td>CJSR</td>
<td>$F\cdot m^{-1}$</td>
<td>1.00e-12</td>
<td>Sidewall junction capacitance at $V=V_R$</td>
</tr>
</tbody>
</table>
Using the Philips MOS9 Model in Star-Hspice

1. Set LEVEL=50 to identify the model as the Philips MOS Model 9.

2. The default room temperature is 25 °C in Star-Hspice, but is 27 °C in most other simulators. When comparing to other simulators, set the simulation temperature to 27 with .TEMP 27 or with .OPTION TNOM=27.

3. The model parameter set should always include the model reference temperature, TR, which corresponds to TREF in other levels in Star-Hspice. The default for TR is 21.0 °C, to match the Philips simulator.

4. The model has its own charge-based capacitance model. The CAPOP parameter, which selects different capacitance models, is ignored for this model.

5. The model uses analytical derivatives for the conductances. The DERIV parameter, which selects the finite difference method, is ignored for this model.

### Philips MOS9 Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJGR</td>
<td>F*m(^{-1})</td>
<td>1.00e-12</td>
<td>Gate edge junction capacitance at (V=V_R)</td>
</tr>
<tr>
<td>VDBR</td>
<td>(v)</td>
<td>1.00</td>
<td>Diffusion voltage of the bottom junction at (T=T_R)</td>
</tr>
<tr>
<td>VDSR</td>
<td>(v)</td>
<td>1.00</td>
<td>Diffusion voltage of the sidewall junction at (T=T_R)</td>
</tr>
<tr>
<td>VDGR</td>
<td>(v)</td>
<td>1.00</td>
<td>Diffusion voltage of the gate edge junction at (T=T_R)</td>
</tr>
<tr>
<td>PB</td>
<td>-</td>
<td>0.40</td>
<td>Bottom-junction grading coefficient</td>
</tr>
<tr>
<td>PS</td>
<td>-</td>
<td>0.40</td>
<td>Sidewall-junction grading coefficient</td>
</tr>
<tr>
<td>PG</td>
<td>-</td>
<td>0.40</td>
<td>Gate edge-junction grading coefficient</td>
</tr>
</tbody>
</table>
6. DTEMP can be used with this model. It is set on the element line and increases the temperature of individual elements relative to the circuit temperature.

7. Since defaults are nonzero, it is strongly recommended that every model parameter listed in LEVEL 50 Model Parameters table be set in the .MODEL statement.

8. Use the model parameter JUNCAP to select one of two available parasitic junction diode models, ACM and JUNCAP. JUNCAP=1 selects the Philips JUNCAP model, JUNCAP=0 (default) selects the Star-Hspice ACM model. The JUNCAP model is available in the Star-Hspice release, version 1999.2 (June 1999).

**Star-Hspice Model Statement**

**Example**

This is an example of the Star-Hspice model statement.

```plaintext
.model nch nmos LEVEL=50
+ ler = 1e-6   wer = 10e-6
+ lvar = 0.0    lap = 0.05e-6
+ wvar = 0.0    wot = 0.0
+ tr = 27.00
+ vtor = 0.8    stvto = 0  slvto = 0  sl2vto= 0
+ swvto = 0
+ kor = 0.7     slko = 0  swko = 0
+ kr = 0.3      slk = 0  swk = 0
+ phibr = 0.65
+ vsbxr = 0.5    slvsbx = 0  swvsbx= 0
+ betsq = 120e-6
+ etabet = 1.5
+ the1r = 0.3
+ stthe1r = 0  slthe1r = 0  stlthe1= 0  swthe1 = 0
+ the2r = 0.06
+ stthe2r = 0  slthe2r = 0  stlthe2  = 0  swthe2  = 0
+ the3r = 0.1
```
```
+ stthe3r = 0 slthe3r = 0 stlthe3 = 0 swthe3 = 0
+ gam1r = 0.02 slgam1 = 0 swgam1 = 0
+ etadsr = 0.60
+ alpr = 0.01
+ etaalp = 0 slalp = 0 swalp = 0
+ vpr = 0.4
+ gamoor = 0.006
+ slgamoo = 0
+ etagamr = 2.0
+ mor = 0.5 stmo = 0 slmo = 0
+ etamr = 2.0
+ zet1r = 1.0
+ etazet = 0.5
+ slzet1 = 0
+ vsbtr = 2.5
+ slvsbt = 0
+ a1r = 10 stal = 0 sla1 = 0 swal = 0
+ a2r = 30 sla2 = 0 swa2 = 0
+ a3r = 0.8 sla3 = 0 swa3 = 0
+ tox = 15.00e-9
+ col = 0.3e-9
+ ntr = 2.0e-20
+ nfr = 5.0e-11
+ acm=2 hdif=1u js=1e-3
+ cj=1e-3 mj=0.5 pb=0.8
+ cjsw=1e-9 cjgate=1e-9 mjsw=0.3 php=0.8
```
LEVEL 54 BSIM4.0 Model

 UC Berkeley BSIM4.0 model is developed to explicitly address many issues in modeling sub-0.13 micron CMOS technology and RF high-speed CMOS circuit simulation. The BSIM4.0.0 MOS model for UC Berkeley is available as the LEVEL 54 Star-Hspice model.

BSIM4.0 has the following major improvements and additions over BSIM3v3:

- An accurate new model of the intrinsic input resistance (R_{ii}) for both RF, high-frequency analog, and high-speed digital applications
- A flexible substrate resistance network for RF modeling
- A new accurate channel thermal noise model and a noise partition model for the induced gate noise
- A non-quasi-static (NQS) model consistent with the R_{ii}-based RF model and a consistent AC model that accounts for the NQS effect in both transconductances and capacitances
- An accurate gate direct tunneling model
- A comprehensive and versatile geometry-dependent parasitics model for various source/drain connections and multi-finger devices
- An improved model for steep vertical retrograde doping profiles
- A better model for pocket-implanted devices in V_{th}, bulk charge effect model, and Rout
- An asymmetrical and bias-dependent source/drain resistance, either internal or external to the intrinsic MOSFET, at the user’s discretion
- An acceptance of either the electrical or physical gate oxide thickness as the model input (at the user’s choice) in a physically accurate manner
- The quantum mechanical charge-layer-thickness model for both IV and CV
- A more accurate mobility model for predictive modeling
- A gate-induced drain leakage (GIDL) current model, available in BSIM for the first time
- An improved unified flicker (1/f) noise model, which is smooth over all bias regions and considers the bulk charge effect
- Different diode IV and CV characteristics for source and drain junctions
- A junction diode breakdown with or without current limiting
- A dielectric constant of the gate dielectric as a model parameter

**LEVEL 54 Model Parameters**

**Model Selectors/Controllers**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERSION</td>
<td>4.0.0</td>
<td>NA</td>
<td>Model version number</td>
</tr>
<tr>
<td>BINUNIT</td>
<td>1</td>
<td>NA</td>
<td>Binning unit selector</td>
</tr>
<tr>
<td>PARAMCHK</td>
<td>1</td>
<td>NA</td>
<td>Switch for parameter value check</td>
</tr>
<tr>
<td>MOBMOD</td>
<td>1</td>
<td>NA</td>
<td>Mobility model selector</td>
</tr>
<tr>
<td>RDSMOD</td>
<td>0</td>
<td>NA</td>
<td>Bias-dependent source/drain resistance model selector</td>
</tr>
<tr>
<td>IGCMD</td>
<td>0</td>
<td>NA</td>
<td>Gate-to-channel tunneling current model selector</td>
</tr>
<tr>
<td>IGBMOD</td>
<td>0</td>
<td>NA</td>
<td>Gate-to-substrate tunneling current model selector</td>
</tr>
<tr>
<td>CAPMOD</td>
<td>2</td>
<td>NA</td>
<td>Capacitance model selector</td>
</tr>
<tr>
<td>RGA TEMOD</td>
<td>0 (no gate resistance)</td>
<td>NA</td>
<td>Gate resistance model selector</td>
</tr>
<tr>
<td>RBODYMOD</td>
<td>0 (network off)</td>
<td>NA</td>
<td>Substrate resistance network model selector</td>
</tr>
<tr>
<td>TRNQSMOD</td>
<td>0</td>
<td>NA</td>
<td>Transient NQS model selector</td>
</tr>
<tr>
<td>ACNQSMOD</td>
<td>0</td>
<td>NA</td>
<td>AC small-signal NQS model selector</td>
</tr>
<tr>
<td>FNOIMOD</td>
<td>1</td>
<td>NA</td>
<td>Flicker noise model selector</td>
</tr>
<tr>
<td>TNOIMOD</td>
<td>0</td>
<td>NA</td>
<td>Thermal noise model selector</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Binnable</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------</td>
<td>----------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>DIOMOD</td>
<td>1</td>
<td>NA</td>
<td>Source/drain junction diode IV model selector</td>
</tr>
<tr>
<td>PERMOD</td>
<td>1</td>
<td>NA</td>
<td>Whether PS/PD includes the gate-edge perimeter</td>
</tr>
<tr>
<td>GEOMOD</td>
<td>0 (isolated)</td>
<td>NA</td>
<td>Geometry-dependent parasitics model selector</td>
</tr>
<tr>
<td>RGEOMOD</td>
<td>0</td>
<td>NA</td>
<td>Source/drain diffusion resistance and contact model selector</td>
</tr>
</tbody>
</table>

**Process Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPSROX</td>
<td>3.9 (SiO&lt;sub&gt;2&lt;/sub&gt;)</td>
<td>No</td>
<td>Gate dielectric constant relative to vacuum</td>
</tr>
<tr>
<td>TOXE</td>
<td>3.0e-9m</td>
<td>No</td>
<td>Electrical gate equivalent oxide thickness</td>
</tr>
<tr>
<td>TOXP</td>
<td>TOXE</td>
<td>No</td>
<td>Physical gate equivalent oxide thickness</td>
</tr>
<tr>
<td>TOXM</td>
<td>TOXE</td>
<td>No</td>
<td>Tox at which parameters are extracted</td>
</tr>
<tr>
<td>DTOX</td>
<td>0.0m</td>
<td>No</td>
<td>Defined as (TOXE-TOXP)</td>
</tr>
<tr>
<td>XJ</td>
<td>1.5e-7m</td>
<td>Yes</td>
<td>S/D junction depth</td>
</tr>
<tr>
<td>GAMMA1</td>
<td>calculated (V&lt;sup&gt;1/2&lt;/sup&gt;)</td>
<td>Yes</td>
<td>Body-effect coefficient near the surface</td>
</tr>
<tr>
<td>GAMMA2</td>
<td>calculated (V&lt;sup&gt;1/2&lt;/sup&gt;)</td>
<td>Yes</td>
<td>Body-effect coefficient in the bulk</td>
</tr>
<tr>
<td>NDEP</td>
<td>1.7e17cm&lt;sup&gt;-3&lt;/sup&gt;</td>
<td>Yes</td>
<td>Channel doping concentration at depletion edge for zero body bias</td>
</tr>
<tr>
<td>NSUB</td>
<td>6.0e16cm&lt;sup&gt;-3&lt;/sup&gt;</td>
<td>Yes</td>
<td>Substrate doping concentration</td>
</tr>
<tr>
<td>NGATE</td>
<td>0.0cm&lt;sup&gt;-3&lt;/sup&gt;</td>
<td>Yes</td>
<td>Poly Si gate doping concentration</td>
</tr>
<tr>
<td>NSD</td>
<td>1.0e20cm&lt;sup&gt;-3&lt;/sup&gt;</td>
<td>Yes</td>
<td>Source/drain doping concentration</td>
</tr>
</tbody>
</table>
Basic Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTH0 or VTHO</td>
<td>0.7V (NMOS) -0.7V (PMOS)</td>
<td>Yes</td>
<td>Long-channel threshold voltage at $V_{bs}=0$</td>
</tr>
<tr>
<td>VFB</td>
<td>-1.0V</td>
<td>Yes</td>
<td>Flat-band voltage PHIN</td>
</tr>
<tr>
<td>PHIN</td>
<td>0.0V</td>
<td>Yes</td>
<td>Non-uniform vertical doping effect on surface potential</td>
</tr>
<tr>
<td>K1</td>
<td>$0.5V^{1/2}$</td>
<td>Yes</td>
<td>First-order body bias coefficient</td>
</tr>
<tr>
<td>K2</td>
<td>0.0</td>
<td>Yes</td>
<td>Second-order body bias coefficient</td>
</tr>
<tr>
<td>K3</td>
<td>80.0</td>
<td>Yes</td>
<td>Narrow width coefficient</td>
</tr>
<tr>
<td>K3B</td>
<td>$0.0V^{-1}$</td>
<td>Yes</td>
<td>Body effect coefficient of K3</td>
</tr>
<tr>
<td>W0</td>
<td>2.5e-6m</td>
<td>Yes</td>
<td>Narrow width parameter</td>
</tr>
<tr>
<td>LPE0</td>
<td>1.74e-7m</td>
<td>Yes</td>
<td>Lateral non-uniform doping parameter</td>
</tr>
<tr>
<td>LPEB</td>
<td>0.0m</td>
<td>Yes</td>
<td>Lateral non-uniform doping effect on K1</td>
</tr>
<tr>
<td>VBM</td>
<td>-3.0V</td>
<td>Yes</td>
<td>Maximum applied body bias in VTH0 calculation</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Binnable</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>DVT0</td>
<td>2.2</td>
<td>Yes</td>
<td>First coefficient of short-channel effect on ( V_{th} )</td>
</tr>
<tr>
<td>DVT1</td>
<td>0.53</td>
<td>Yes</td>
<td>Second coefficient of short-channel effect on ( V_{th} )</td>
</tr>
<tr>
<td>DVT2</td>
<td>-0.032V(^{-1})</td>
<td>Yes</td>
<td>Body-bias coefficient of short-channel effect on ( V_{th} )</td>
</tr>
<tr>
<td>DVTP0</td>
<td>0.0m</td>
<td>Yes</td>
<td>First coefficient of drain-induced ( V_{th} ) shift due to for long-channel pocket devices</td>
</tr>
<tr>
<td>DVTP1</td>
<td>0.0V(^{-1})</td>
<td>Yes</td>
<td>First coefficient of drain-induced ( V_{th} ) shift due to for long-channel pocket devices</td>
</tr>
<tr>
<td>DVT0W</td>
<td>0.0</td>
<td>Yes</td>
<td>First coefficient of narrow width effect on ( V_{th} ) for small channel length</td>
</tr>
<tr>
<td>DVT1W</td>
<td>5.3e6m(^{-1})</td>
<td>Yes</td>
<td>Second coefficient of narrow width effect on ( V_{th} ) for small channel length</td>
</tr>
<tr>
<td>DVT2W</td>
<td>-0.032V(^{-1})</td>
<td>Yes</td>
<td>Body-bias coefficient of narrow width effect for small channel length</td>
</tr>
<tr>
<td>U0</td>
<td>0.067m(^2)//(Vs) (NMOS); 0.025m(^2)//(Vs) (PMOS)</td>
<td>Yes</td>
<td>Low-field mobility</td>
</tr>
<tr>
<td>UA</td>
<td>1.0e-9m/V for MOBMOD=0 and 1; 1.0e-15m/V for MOBMOD=2</td>
<td>Yes</td>
<td>Coefficient of first-order mobility degradation due to vertical field</td>
</tr>
<tr>
<td>UB</td>
<td>1.0e-19m(^2)/V(^2)</td>
<td>Yes</td>
<td>Coefficient of second-order mobility degradation due to vertical field</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Binnable</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UC</td>
<td>-0.0465V(^{-1}) for MOB-MOD=1; -0.0465e-9 m/V(^2) for MOBMOD=0 and 2</td>
<td>Yes</td>
<td>Coefficient of mobility degradation due to body-bias effect</td>
</tr>
<tr>
<td>EU</td>
<td>1.67 (NMOS); 1.0 (PMOS)</td>
<td>No</td>
<td>Exponent for mobility degradation of MOBMOD=2</td>
</tr>
<tr>
<td>VSAT</td>
<td>8.0e4m/s</td>
<td>Yes</td>
<td>Saturation velocity</td>
</tr>
<tr>
<td>A0</td>
<td>1.0</td>
<td>Yes</td>
<td>Coefficient of channel-length dependence of bulk charge effect</td>
</tr>
<tr>
<td>AGS</td>
<td>0.0V(^{-1})</td>
<td>Yes</td>
<td>Coefficient of V(_{gs}) dependence of bulk charge effect</td>
</tr>
<tr>
<td>B0</td>
<td>0.0m</td>
<td>Yes</td>
<td>Bulk charge effect coefficient for channel width</td>
</tr>
<tr>
<td>B1</td>
<td>0.0m</td>
<td>Yes</td>
<td>Bulk charge effect width offset</td>
</tr>
<tr>
<td>KETA</td>
<td>-0.047V(^{-1})</td>
<td>Yes</td>
<td>Body-bias coefficient of bulk charge effect</td>
</tr>
<tr>
<td>A1</td>
<td>0.0V(^{-1})</td>
<td>Yes</td>
<td>First non-saturation effect parameter</td>
</tr>
<tr>
<td>A2</td>
<td>1.0</td>
<td>Yes</td>
<td>Second non-saturation factor</td>
</tr>
<tr>
<td>WINT</td>
<td>0.0m</td>
<td>No</td>
<td>Channel-width offset parameter</td>
</tr>
<tr>
<td>LINT</td>
<td>0.0m</td>
<td>No</td>
<td>Channel-length offset parameter</td>
</tr>
<tr>
<td>DWG</td>
<td>0.0m/V</td>
<td>Yes</td>
<td>Coefficient of gate bias dependence of W(_{eff})</td>
</tr>
<tr>
<td>DWB</td>
<td>0.0m/V(^{1/2})</td>
<td>Yes</td>
<td>Coefficient of body bias dependence of W(_{eff}) bias dependence</td>
</tr>
<tr>
<td>VOFF</td>
<td>-0.08V</td>
<td>Yes</td>
<td>Offset voltage in subthreshold region for large W and L</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Binnable</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>VOFFL</td>
<td>0.0mV</td>
<td>No</td>
<td>Channel-length dependence of VOFF</td>
</tr>
<tr>
<td>MINV</td>
<td>0.0</td>
<td>Yes</td>
<td>$V_{gsteff}$ fitting parameter for moderate inversion condition</td>
</tr>
<tr>
<td>NFACTOR</td>
<td>1.0</td>
<td>Yes</td>
<td>Subthreshold swing factor</td>
</tr>
<tr>
<td>ETA0</td>
<td>0.08</td>
<td>Yes</td>
<td>DIBL coefficient in subthreshold region</td>
</tr>
<tr>
<td>ETAB</td>
<td>-0.07V$^{-1}$</td>
<td>Yes</td>
<td>Body-bias coefficient for the subthreshold DIBL effect</td>
</tr>
<tr>
<td>DSUB</td>
<td>DROUT</td>
<td>Yes</td>
<td>DIBL coefficient exponent in subthreshold region</td>
</tr>
<tr>
<td>CIT</td>
<td>0.0F/m$^2$</td>
<td>Yes</td>
<td>Interface trap capacitance</td>
</tr>
<tr>
<td>CDSC</td>
<td>2.4e-4F/m$^2$</td>
<td>Yes</td>
<td>Coupling capacitance between source/drain and channel</td>
</tr>
<tr>
<td>CDSCB</td>
<td>0.0F/(V/m$^2$)</td>
<td>Yes</td>
<td>Body-bias sensitivity of CDSC</td>
</tr>
<tr>
<td>CDSCD</td>
<td>0.0(F/V/m$^2$)</td>
<td>Yes</td>
<td>Drain-bias sensitivity of CDSC</td>
</tr>
<tr>
<td>PCLM</td>
<td>1.3</td>
<td>Yes</td>
<td>Channel-length modulation parameter</td>
</tr>
<tr>
<td>PDIBLC1</td>
<td>0.39</td>
<td>Yes</td>
<td>Parameter for DIBL effect on Rout</td>
</tr>
<tr>
<td>PDIBLC2</td>
<td>0.0086</td>
<td>Yes</td>
<td>Parameter for DIBL effect on Rout</td>
</tr>
<tr>
<td>PDIBLCB</td>
<td>0.0V$^{-1}$</td>
<td>Yes</td>
<td>Body bias coefficient of DIBL effect on Rout</td>
</tr>
<tr>
<td>DROUT</td>
<td>0.56</td>
<td>Yes</td>
<td>Channel-length dependence of DIBL effect on Rout</td>
</tr>
<tr>
<td>PSCBE1</td>
<td>4.24e8V/m</td>
<td>Yes</td>
<td>First substrate current induced body-effect parameter</td>
</tr>
</tbody>
</table>
### Parameters for Asymmetric and Bias-Dependent $R_{ds}$ Model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDSW</td>
<td>$200.0 \text{ ohm(\mu m)}^\text{WR}$</td>
<td>Yes</td>
<td>Zero bias LLD resistance per unit width for RDSMOD=0</td>
</tr>
<tr>
<td>RDSWMIN</td>
<td>$0.0 \text{ ohm(\mu m)}^\text{WR}$</td>
<td>No</td>
<td>LDD resistance per unit width at high $V_{gs}$ and zero $V_{bs}$ for RDSMOD=0</td>
</tr>
<tr>
<td>RDW</td>
<td>$100.0 \text{ ohm(\mu m)}^\text{WR}$</td>
<td>Yes</td>
<td>Zero bias lightly-doped drain resistance $R_d(v)$ per unit width for RDSMOD=1</td>
</tr>
<tr>
<td>RDWMIN</td>
<td>$0.0 \text{ ohm(\mu m)}^\text{WR}$</td>
<td>No</td>
<td>Lightly-doped drain resistance per unit width at high $V_{gs}$ and zero $V_{bs}$ for RDSMOD=1</td>
</tr>
</tbody>
</table>
### Impact Ionization Current Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA0</td>
<td>0.0Am/V</td>
<td>Yes</td>
<td>First parameter of impact ionization current</td>
</tr>
<tr>
<td>ALPHA1</td>
<td>0.0A/V</td>
<td>Yes</td>
<td>Isub parameter for length scaling</td>
</tr>
<tr>
<td>BETA0</td>
<td>30.0V</td>
<td>Yes</td>
<td>The second parameter for impact ionization current</td>
</tr>
</tbody>
</table>

### Gate-Induced Drain Leakage Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGIDL</td>
<td>0.0ohm</td>
<td>Yes</td>
<td>Pre-exponential coefficient for GIDL</td>
</tr>
<tr>
<td>BGIDL</td>
<td>2.3e9V/m</td>
<td>Yes</td>
<td>Exponential coefficient for GIDL</td>
</tr>
<tr>
<td>CGIDL</td>
<td>0.5V^3</td>
<td>Yes</td>
<td>Parameter for body-bias effect on GIDL</td>
</tr>
</tbody>
</table>
### Gate Dielectric Tunneling Current Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGIDL</td>
<td>0.8V</td>
<td>Yes</td>
<td>Fitting parameter for band bending for GIDL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIGBACC</td>
<td>0.43 ((F_s^2/g)^0.5) m(^{-1}) (V^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gb}) in accumulation</td>
</tr>
<tr>
<td>BIGBACC</td>
<td>0.054 ((F_s^2/g)^0.5) m(^{-1}) V(^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gb}) in accumulation</td>
</tr>
<tr>
<td>CIGBACC</td>
<td>0.075 V(^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gb}) in accumulation</td>
</tr>
<tr>
<td>NIGBACC</td>
<td>1.0</td>
<td>Yes</td>
<td>Parameter for (I_{gb}) in accumulation</td>
</tr>
<tr>
<td>AIGBINV</td>
<td>0.35 ((F_s^2/g)^0.5) m(^{-1}) (V^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gb}) in inversion</td>
</tr>
<tr>
<td>BIGBINV</td>
<td>0.03 ((F_s^2/g)^0.5) m(^{-1}) V(^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gb}) in inversion</td>
</tr>
<tr>
<td>CIGINV</td>
<td>0.0006 V(^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gb}) in inversion</td>
</tr>
<tr>
<td>EIGINV</td>
<td>1.1V</td>
<td>Yes</td>
<td>Parameter for (I_{gb}) in inversion</td>
</tr>
<tr>
<td>NIGINV</td>
<td>3.0</td>
<td>Yes</td>
<td>Parameter for (I_{gb}) in inversion</td>
</tr>
<tr>
<td>AIGC</td>
<td>0.054 (NMOS) and 0.31 (PMOS) ((F_s^2/g)^0.5) m(^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gcs}) and (I_{gcd})</td>
</tr>
<tr>
<td>BIGC</td>
<td>0.054 (NMOS) and 0.024 (PMOS) ((F_s^2/g)^0.5) m(^{-1}) V(^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gcs}) and (I_{gcd})</td>
</tr>
<tr>
<td>CIGC</td>
<td>0.075 (NMOS) and 0.03 (PMOS) V(^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gcs}) and (I_{gcd})</td>
</tr>
<tr>
<td>AIGSD</td>
<td>0.43 (NMOS) and 0.31 (PMOS) ((F_s^2/g)^0.5) m(^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gs}) and (I_{gd})</td>
</tr>
<tr>
<td>BIGSD</td>
<td>0.054 (NMOS) 0.024 (PMOS) ((F_s^2/g)^0.5) m(^{-1}) V(^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gs}) and (I_{gd})</td>
</tr>
<tr>
<td>CIGSD</td>
<td>0.075 (NMOS) and 0.03 (PMOS) V(^{-1})</td>
<td>Yes</td>
<td>Parameter for (I_{gs}) and (I_{gd})</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Binnable</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
<td>----------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DLCIG</td>
<td>LINT</td>
<td>Yes</td>
<td>Source/drain overlap length for $I_{gs}$ and $I_{gd}$</td>
</tr>
<tr>
<td>NIGC</td>
<td>1.0</td>
<td>Yes</td>
<td>Parameter for $I_{gcs}$, $I_{gcd}$, $I_{gs}$ and $I_{gd}$</td>
</tr>
<tr>
<td>POXEDGE</td>
<td>1.0</td>
<td>Yes</td>
<td>Factor for the gate oxide thickness in source/drain overlap regions</td>
</tr>
<tr>
<td>PIGCD</td>
<td>1.0</td>
<td>Yes</td>
<td>$V_{ds}$ dependence of $I_{gcs}$ and $I_{gcd}$</td>
</tr>
<tr>
<td>NTOX</td>
<td>1.0</td>
<td>Yes</td>
<td>Exponent for the gate oxide ratio</td>
</tr>
<tr>
<td>TOXREF</td>
<td>3.0e-9m</td>
<td>No</td>
<td>Nominal gate oxide thickness for gate dielectric tunneling current model only</td>
</tr>
</tbody>
</table>

**Charge and Capacitance Model Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPART</td>
<td>0.0</td>
<td>No</td>
<td>Charge partition parameter</td>
</tr>
<tr>
<td>CGSO</td>
<td>calculated (F/m)</td>
<td>No</td>
<td>Non LDD region source-gate overlap capacitance per unit channel width</td>
</tr>
<tr>
<td>CGDO</td>
<td>calculated (F/m)</td>
<td>No</td>
<td>Non LDD region drain-gate overlap capacitance per unit channel width</td>
</tr>
<tr>
<td>CGBO</td>
<td>0.0 (F/m)</td>
<td>No</td>
<td>Gate-bulk overlap capacitance per unit channel length</td>
</tr>
<tr>
<td>CGSL</td>
<td>0.0F/m</td>
<td>Yes</td>
<td>Overlap capacitance between gate and lightly-doped source region</td>
</tr>
<tr>
<td>CGDL</td>
<td>0.0F/m</td>
<td>Yes</td>
<td>Overlap capacitance between gate and lightly-doped source region</td>
</tr>
<tr>
<td>CKAPPAS</td>
<td>0.6V</td>
<td>Yes</td>
<td>Coefficient of bias-dependent overlap capacitance for the source side</td>
</tr>
<tr>
<td>CKAPPAD</td>
<td>CKAPPAS</td>
<td>Yes</td>
<td>Coefficient of bias-dependent overlap capacitance for the drain side</td>
</tr>
</tbody>
</table>
### High-Speed/RF Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XRCRG1</td>
<td>12.0</td>
<td>Yes</td>
<td>Parameter for distributed channel-resistance effect for both intrinsic-input resistance and charge-deficit NQS models</td>
</tr>
<tr>
<td>XRCRG2</td>
<td>1.0</td>
<td>Yes</td>
<td>Parameter to account for the excess channel diffusion resistance for both intrinsic input resistance and charge-deficit NQS models</td>
</tr>
</tbody>
</table>
Flicker and Thermal Noise Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOIA</td>
<td>6.25e41 (eV)^-1s^1-\text{EF}m^-3 for NMOS; 6.188e40 (eV)^-1s^1-\text{EF}m^-3 for PMOS</td>
<td>No</td>
<td>Flicker noise parameter A</td>
</tr>
<tr>
<td>NOIB</td>
<td>3.125e26 (eV)^-1s^1-\text{EF}m^-1 for NMOS; 1.5e25 (eV)^-1s^1-\text{EF}m^-1 for PMOS</td>
<td>No</td>
<td>Flicker noise parameter B</td>
</tr>
<tr>
<td>NOIC</td>
<td>8.75 (eV)^-1S^1-\text{EF}m</td>
<td>No</td>
<td>Flicker noise parameter C</td>
</tr>
<tr>
<td>EM</td>
<td>4.1e7V/m</td>
<td>No</td>
<td>Saturation field</td>
</tr>
<tr>
<td>AF</td>
<td>1.0</td>
<td>No</td>
<td>Flicker noise exponent</td>
</tr>
<tr>
<td>EF</td>
<td>1.0</td>
<td>No</td>
<td>Flicker noise frequency exponent</td>
</tr>
</tbody>
</table>
### Layout-Dependent Parasitics Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMCG</td>
<td>0.0m</td>
<td>No</td>
<td>Distance from S/D contact center to the gate edge</td>
</tr>
<tr>
<td>DMCI</td>
<td>DMCG</td>
<td>No</td>
<td>Distance from S/D contact center to the isolation edge in the channel-length direction</td>
</tr>
<tr>
<td>DMDG</td>
<td>0.0m</td>
<td>No</td>
<td>Same as DMCG but for merged device only</td>
</tr>
<tr>
<td>DMCGT</td>
<td>0.0m</td>
<td>No</td>
<td>DMCG of test structures</td>
</tr>
<tr>
<td>NF</td>
<td>1</td>
<td>No</td>
<td>Number of device figures</td>
</tr>
<tr>
<td>DWJ</td>
<td>DWC (in CVmodel)</td>
<td>No</td>
<td>Offset of the S/D junction width</td>
</tr>
<tr>
<td>MIN</td>
<td>0</td>
<td>No</td>
<td>Whether to minimize the number of drain or source diffusions for even-number fingered device</td>
</tr>
<tr>
<td>XGW</td>
<td>0.0m</td>
<td>No</td>
<td>Distance from the gate contact to the channel edge</td>
</tr>
<tr>
<td>XGL</td>
<td>0.0m</td>
<td>No</td>
<td>Offset of the gate length due to variations in patterning</td>
</tr>
</tbody>
</table>
### Asymmetric Source/Drain Junction Diode Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NGCON</td>
<td>1</td>
<td>No</td>
<td>Number of gate contacts</td>
</tr>
<tr>
<td>IJTHSREV</td>
<td>IJTHSREV=0.1A</td>
<td>No</td>
<td>Limiting current in reverse bias region</td>
</tr>
<tr>
<td>IJTHDREV</td>
<td>IJTHDREV=IJTHSREV</td>
<td>No</td>
<td>Limiting current in reverse bias region</td>
</tr>
<tr>
<td>IJTHSFWD</td>
<td>IJTHSFWD=0.1A</td>
<td>No</td>
<td>Limiting current in forward bias region</td>
</tr>
<tr>
<td>IJTHDFWD</td>
<td>IJTHDFWD=IJTHSFWD</td>
<td>No</td>
<td>Limiting current in forward bias region</td>
</tr>
<tr>
<td>XJBVS</td>
<td>XJBVS=1.0</td>
<td>No</td>
<td>Fitting parameter for diode breakdown</td>
</tr>
<tr>
<td>XJBVD</td>
<td>XJBVD=XJBVS</td>
<td>No</td>
<td>Fitting parameter for diode breakdown</td>
</tr>
<tr>
<td>BVS</td>
<td>BVS=10.0V</td>
<td>No</td>
<td>Breakdown voltage</td>
</tr>
<tr>
<td>BVD</td>
<td>BVD=BVS</td>
<td>No</td>
<td>Breakdown voltage</td>
</tr>
<tr>
<td>JSS</td>
<td>JSS=1.0e-4A/m²</td>
<td>No</td>
<td>Bottom junction reverse saturation current density</td>
</tr>
<tr>
<td>JSD</td>
<td>JSD=JSS</td>
<td>No</td>
<td>Bottom junction reverse saturation current density</td>
</tr>
<tr>
<td>JSWS</td>
<td>JSWS=0.0A/m</td>
<td>No</td>
<td>Isolation-edge sidewall reverse saturation current density</td>
</tr>
<tr>
<td>JSWD</td>
<td>JSWD=JSWS</td>
<td>No</td>
<td>Isolation-edge sidewall reverse saturation current density</td>
</tr>
<tr>
<td>JSWG</td>
<td>JSWG=0.0A/m</td>
<td>No</td>
<td>Gate-edge sidewall reverse saturation current density</td>
</tr>
<tr>
<td>JSWG</td>
<td>JSWG=JSWG</td>
<td>No</td>
<td>Gate-edge sidewall reverse saturation current density</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Binnable</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>CJS</td>
<td>CJS=5.0e-4 F/m²</td>
<td>No</td>
<td>Bottom junction capacitance per unit area at zero bias</td>
</tr>
<tr>
<td>CJD</td>
<td>CJD=CJS</td>
<td>No</td>
<td>Bottom junction capacitance per unit area at zero bias</td>
</tr>
<tr>
<td>MJS</td>
<td>MJS=0.5</td>
<td>No</td>
<td>Bottom junction capacitance grading coefficient</td>
</tr>
<tr>
<td>MJD</td>
<td>MJD=MJS</td>
<td>No</td>
<td>Bottom junction capacitance grading coefficient</td>
</tr>
<tr>
<td>MJSWS</td>
<td>MJSWS=0.33</td>
<td>No</td>
<td>Isolation-edge sidewall junction capacitance grading coefficient</td>
</tr>
<tr>
<td>MJSWD</td>
<td>MJSWD=MJSWS</td>
<td>No</td>
<td>Isolation-edge sidewall junction capacitance grading coefficient</td>
</tr>
<tr>
<td>CJSWS</td>
<td>CJSWS=5.0e-10 F/m</td>
<td>No</td>
<td>Isolation-edge sidewall junction capacitance per unit area</td>
</tr>
<tr>
<td>CJSWD</td>
<td>CJSWD=CJSWS</td>
<td>No</td>
<td>Isolation-edge sidewall junction capacitance per unit area</td>
</tr>
<tr>
<td>CJSWGS</td>
<td>CJSWGS=CJSWS</td>
<td>No</td>
<td>Gate-edge sidewall junction capacitance per unit length</td>
</tr>
<tr>
<td>CJSWGD</td>
<td>CJSWGD=CJSWS</td>
<td>No</td>
<td>Gate-edge sidewall junction capacitance per unit length</td>
</tr>
<tr>
<td>MJSWGS</td>
<td>MJSWGS=MJSWS</td>
<td>No</td>
<td>Gate-edge sidewall junction capacitance grading coefficient</td>
</tr>
<tr>
<td>MJSWGD</td>
<td>MJSWGD=MJSWS</td>
<td>No</td>
<td>Gate-edge sidewall junction capacitance grading coefficient</td>
</tr>
<tr>
<td>PBS</td>
<td>PBS=1.0V</td>
<td>No</td>
<td>Bottom junction built-in potential</td>
</tr>
<tr>
<td>PBD</td>
<td>PBD=PBS</td>
<td>No</td>
<td>Bottom junction built-in potential</td>
</tr>
<tr>
<td>PBSWS</td>
<td>PBSWS=1.0V</td>
<td>No</td>
<td>Isolation-edge sidewall junction built-in potential</td>
</tr>
</tbody>
</table>
LEVEL 54 BSIM4.0 Model

Selecting MOSFET Models: LEVEL 47-62

Parameter | Default | Binnable | Description
--- | --- | --- | ---
PBSD | PBSWD=PBSWS | No | Isolation-edge sidewall junction built-in potential
PBSWGS | PBSWGS=PBSWS | No | Gate-edge sidewall junction built-in potential
PBSWGD | PBSWGD=PBSWS | No | Gate-edge sidewall junction built-in potential

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNOEM</td>
<td>27°C</td>
<td>No</td>
<td>Temperature at which parameters are extracted</td>
</tr>
<tr>
<td>UTE</td>
<td>-1.5</td>
<td>Yes</td>
<td>Mobility temperature exponent</td>
</tr>
<tr>
<td>KT1</td>
<td>-0.11V</td>
<td>Yes</td>
<td>Temperature coefficient for threshold voltage</td>
</tr>
<tr>
<td>KT1L</td>
<td>0.0Vm</td>
<td>Yes</td>
<td>Channel length dependence of the temperature coefficient for threshold voltage</td>
</tr>
<tr>
<td>KT2</td>
<td>0.022</td>
<td>Yes</td>
<td>Body-bias coefficient of Vth temperature effect</td>
</tr>
<tr>
<td>UA1</td>
<td>1.0e-9m/V</td>
<td>Yes</td>
<td>Temperature coefficient for UA</td>
</tr>
<tr>
<td>UB1</td>
<td>-1.0e-18 (m/V²)</td>
<td>Yes</td>
<td>Temperature coefficient for UB</td>
</tr>
<tr>
<td>UC1</td>
<td>0.067V¹ for MOBMOD=1; 0.025m/V² for MOBMOD=0 and 2</td>
<td>Yes</td>
<td>Temperature coefficient for UC</td>
</tr>
<tr>
<td>AT</td>
<td>3.3e4m/s</td>
<td>Yes</td>
<td>Temperature coefficient for saturation velocity</td>
</tr>
</tbody>
</table>
### Selecting MOSFET Models: LEVEL 47-62

**LEVEL 54 BSIM4.0 Model**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRT</td>
<td>0.0ohm-m</td>
<td>Yes</td>
<td>Temperature coefficient for Rdsw</td>
</tr>
<tr>
<td>NJS, NJD</td>
<td>NJS=1.0;</td>
<td>No</td>
<td>Emission coefficients of junction for source and drain junctions, respectively</td>
</tr>
<tr>
<td></td>
<td>NJD=NJS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XTIS, XTID</td>
<td>XTIS=3.0;</td>
<td>No</td>
<td>Junction current temperature exponents for source and drain junction, respectively</td>
</tr>
<tr>
<td></td>
<td>XTID=XTIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPB</td>
<td>0.0V/K</td>
<td>No</td>
<td>Temperature coefficient of PB</td>
</tr>
<tr>
<td>TPBSW</td>
<td>0.0V/K</td>
<td>No</td>
<td>Temperature coefficient of PBSW</td>
</tr>
<tr>
<td>TPBSWG</td>
<td>0.0V/K</td>
<td>No</td>
<td>Temperature coefficient of PBSWG</td>
</tr>
<tr>
<td>TCJ</td>
<td>0.0K⁻¹</td>
<td>No</td>
<td>Temperature coefficient of CJ</td>
</tr>
<tr>
<td>TCJSW</td>
<td>0.0K⁻¹</td>
<td>No</td>
<td>Temperature coefficient of CJSW</td>
</tr>
<tr>
<td>TCJSWG</td>
<td>0.0K⁻¹</td>
<td>No</td>
<td>Temperature coefficient of CJSWG</td>
</tr>
</tbody>
</table>

### Parameter Default Binnable Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WL</td>
<td>0.0mWLN</td>
<td>No</td>
<td>Coefficient of length dependence for width offset</td>
</tr>
<tr>
<td>WLN</td>
<td>1.0</td>
<td>No</td>
<td>Power of length dependence of width offset</td>
</tr>
<tr>
<td>WW</td>
<td>0.0mWWN</td>
<td>No</td>
<td>Coefficient of width dependence for width offset</td>
</tr>
<tr>
<td>WWN</td>
<td>1.0</td>
<td>No</td>
<td>Power of width dependence of width offset</td>
</tr>
<tr>
<td>WWL</td>
<td>0.0mWWN+WLN</td>
<td>No</td>
<td>Coefficient of length and width cross term dependence for width offset</td>
</tr>
<tr>
<td>LL</td>
<td>0.0mLLN</td>
<td>No</td>
<td>Coefficient of length dependence for length offset</td>
</tr>
<tr>
<td>LLN</td>
<td>1.0</td>
<td>No</td>
<td>Power of length dependence for length offset</td>
</tr>
</tbody>
</table>

### dW and dL Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WL</td>
<td>0.0mWLN</td>
<td>No</td>
<td>Coefficient of length dependence for width offset</td>
</tr>
<tr>
<td>WLN</td>
<td>1.0</td>
<td>No</td>
<td>Power of length dependence of width offset</td>
</tr>
<tr>
<td>WW</td>
<td>0.0mWWN</td>
<td>No</td>
<td>Coefficient of width dependence for width offset</td>
</tr>
<tr>
<td>WWN</td>
<td>1.0</td>
<td>No</td>
<td>Power of width dependence of width offset</td>
</tr>
<tr>
<td>WWL</td>
<td>0.0mWWN+WLN</td>
<td>No</td>
<td>Coefficient of length and width cross term dependence for width offset</td>
</tr>
<tr>
<td>LL</td>
<td>0.0mLLN</td>
<td>No</td>
<td>Coefficient of length dependence for length offset</td>
</tr>
<tr>
<td>LLN</td>
<td>1.0</td>
<td>No</td>
<td>Power of length dependence for length offset</td>
</tr>
</tbody>
</table>
Range Parameters for Model Application

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Binnable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMIN</td>
<td>0.0m</td>
<td>No</td>
<td>Minimum channel length</td>
</tr>
<tr>
<td>LMAX</td>
<td>1.0m</td>
<td>No</td>
<td>Maximum channel length</td>
</tr>
<tr>
<td>WMIN</td>
<td>0.0m</td>
<td>No</td>
<td>Minimum channel width</td>
</tr>
<tr>
<td>WMAX</td>
<td>1.0m</td>
<td>No</td>
<td>Maximum channel width</td>
</tr>
</tbody>
</table>
LEVEL 55 EPFL-EKV MOSFET Model

The EPFL-EKV MOSFET model is a scalable and compact simulation model built on fundamental physical properties of the MOS structure. This model is dedicated to the design and simulation of low-voltage, low-current analog, and mixed analog-digital circuits using submicron CMOS technologies.

This section provides a description of the equations and parameters used for the computer simulation version of the EPFL-EKV MOSFET model. The description concentrates on the intrinsic part of the MOSFET, and is intended to give the model user information on parameter handling and the actual equations used in the computer simulation.

The extrinsic part of the MOSFET is handled as it is often made for other MOSFET models. The extrinsic model includes the series resistances of the source and drain diffusions, which are handled as external elements, as well as junction currents and capacitances.

Single Equation Model

The EPFL-EKV MOSFET model is in principle 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. The analytical expressions of first-order derivatives as transconductances and transcapacitances are not presented in this section but are also available for computer simulation.

Effects Modeled

The EPFL-EKV MOSFET model version 2.6 includes modeling of the following physical effects:

- Basic geometrical and process related aspects as oxide thickness, junction depth, effective channel length and width
- Effects of doping profile, substrate effect
- Modeling of weak, moderate, and strong inversion behavior
- Modeling of mobility effects due to vertical and lateral fields, velocity saturation
- Short-channel effects as channel-length modulation (CLM), source and drain charge-sharing (including for narrow channel widths), reverse short channel effect (RSCE)
- Modeling of substrate current due to impact ionization
- Quasi-static charge-based dynamic model
- Thermal and flicker noise modeling
- First-order non-quasi-static model for the transadmittances
- Short-distance geometry- and bias-dependent device matching.

**Coherence of Static and Dynamic Models**

All aspects regarding the static, the quasi-static and non-quasi-static dynamic and noise models are all derived in a coherent way from a single characteristic, the normalized transconductance-to-current ratio. Symmetric normalized forward and reverse currents are used throughout these expressions. For quasi-static dynamic operation, both a charge-based model for the node charges and trans-capacitances, and a simpler capacitances model are available. The dynamic model, including the time constant for the nonquasi-static model, is described in symmetrical terms of the forward and reverse normalized currents. The charge formulation is further used to express effective mobility dependence of local field.

**Bulk Reference and Symmetry**

Voltages are all referred to the local substrate:

\[
V_G = V_{GB} \quad \text{Intrinsic gate-to-bulk voltage}
\]
\[
V_S = V_{SB} \quad \text{Intrinsic source-to-bulk voltage}
\]
\[
V_D = V_{DB} \quad \text{Intrinsic drain-to-bulk voltage}
\]

\(V_S\) and \(V_D\) are the intrinsic source and drain voltages, which means that the voltage drop over extrinsic resistive elements is supposed to have already been accounted for externally. \(V_D\) is the electrical drain voltage, and is chosen such
that \( V_D \geq V_S \). Bulk reference allows the model to be handled symmetrically with respect to source and drain, a symmetry that is inherent in common MOS technologies (excluding non-symmetric source-drain layouts).

**Note:** Intrinsic model equations are presented for an N-channel MOSFET. P-channel MOSFETs are dealt with as pseudo-N-channel, i.e. the polarity of the voltages \( V_G, V_S, V_D \), as well as \( V_{FB}, V_{TO}, \) and \( TC \) is reversed prior to computing the current for P-channel, which is given a negative sign. No other distinctions are made between N-channel and P-channel, with the exception of the \( \eta \) factor for effective mobility calculation.

**Equivalent Circuit**

This figure represents the intrinsic and extrinsic elements of the MOS transistor. For quasi-static dynamic operation, only the intrinsic capacitances from the simpler capacitances model are shown here. However, a charge-based transcapacitances model is also available for computer simulation.
### Device Input Variables

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>m</td>
<td>-</td>
<td>Channel length</td>
</tr>
<tr>
<td>W</td>
<td>m</td>
<td>-</td>
<td>Channel width</td>
</tr>
<tr>
<td>M or NP</td>
<td>-</td>
<td>1.0</td>
<td>Parallel multiple device number</td>
</tr>
<tr>
<td>N or NS</td>
<td>-</td>
<td>1.0</td>
<td>Series multiple device number</td>
</tr>
</tbody>
</table>

### EKV Intrinsic Model Parameters

**Process Related Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COX</td>
<td>$F/m^2$</td>
<td>0.7E-3</td>
<td>-</td>
<td>Gate oxide capacitance per unit area</td>
</tr>
<tr>
<td>XJ</td>
<td>m</td>
<td>0.1E-6</td>
<td>$\geq 1.0E-9$</td>
<td>Junction depth</td>
</tr>
<tr>
<td>DW</td>
<td>m</td>
<td>0</td>
<td>-</td>
<td>Channel width correction</td>
</tr>
<tr>
<td>DL</td>
<td>m</td>
<td>0</td>
<td>-</td>
<td>Channel length correction</td>
</tr>
</tbody>
</table>

a. The default value of COX can be calculated as function of TOX.
b. DL and DW parameters usually have a negative value; see effective length and width calculation.

### Basic Intrinsic Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTO</td>
<td>V</td>
<td>0.5</td>
<td>-</td>
<td>Long-channel threshold voltage</td>
</tr>
<tr>
<td>GAMMA</td>
<td>$\sqrt{V}$</td>
<td>1.0</td>
<td>$\geq 0$</td>
<td>Body effect parameter</td>
</tr>
<tr>
<td>PHI</td>
<td>V</td>
<td>0.7</td>
<td>$\geq 0.1$</td>
<td>Bulk Fermi potential (*2)</td>
</tr>
<tr>
<td>KP</td>
<td>$A/V^2$</td>
<td>50.0E-6</td>
<td>-</td>
<td>Transconductance parameter</td>
</tr>
<tr>
<td>E0 (EO)</td>
<td>$V/m$</td>
<td>1.0E12</td>
<td>$\geq 1E5$</td>
<td>Mobility reduction coefficient</td>
</tr>
</tbody>
</table>
Optional Parameters

The following parameters are introduced, to accommodate scaling behavior of the process and basic intrinsic model parameters, as well as statistical circuit simulation. Note that the parameters TOX, NSUB, VFB, UO, and VMAX are only used if COX, GAMMA and/or PHI, VTO, KP and UCRIT are not specified, respectively. Further, a simpler mobility reduction model due to vertical field is accessible. The mobility reduction coefficient THETA is only used if E0 is not specified.

<table>
<thead>
<tr>
<th>Name</th>
<th>Unita</th>
<th>Default</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCRIT</td>
<td>V/m</td>
<td>2.0E6</td>
<td>≥ 1E5</td>
<td>Longitudinal critical field</td>
</tr>
</tbody>
</table>

a. The default values of VTO, GAMMA, PHI, KP can be calculated as function of TOX, NSUB, UO, VFB for the purpose of statistical circuit simulation.
b. As $V_G$, VTO is also referred to the bulk.

c. Optional parameter used to calculate COX.
d. Optional parameter used to calculate VTO as a function of COX, GAMMA, PHI.
e. Optional parameter accounting for the dependence of KP on COX.
f. Optional parameter used to calculate UCRIT.
g. Optional parameter for mobility reduction due to vertical field.
## Channel Length Modulation and Charge Sharing Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAMBDA</td>
<td>-</td>
<td>0.5</td>
<td>≥ 0</td>
<td>Depletion length coefficient (channel length modulation)</td>
</tr>
<tr>
<td>WETA</td>
<td>-</td>
<td>0.25</td>
<td>-</td>
<td>Narrow-channel effect coefficient</td>
</tr>
<tr>
<td>LET A</td>
<td>-</td>
<td>0.1</td>
<td>-</td>
<td>Short-channel effect coefficient</td>
</tr>
</tbody>
</table>

## Reverse Short-channel Effect Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q0 (QO)</td>
<td>A/s/m²</td>
<td>0</td>
<td>-</td>
<td>Reverse short channel effect peak charge density</td>
</tr>
<tr>
<td>LK</td>
<td>m</td>
<td>0.29E-6</td>
<td>≥ 1.0E-8</td>
<td>Reverse short channel effect characteristic length</td>
</tr>
</tbody>
</table>

## Impact Ionization Related Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBA</td>
<td>1/m</td>
<td>0</td>
<td>-</td>
<td>First impact ionization coefficient</td>
</tr>
<tr>
<td>IBB</td>
<td>V/m</td>
<td>3.0E8</td>
<td>≥ 1.0E8</td>
<td>Second impact ionization coefficient</td>
</tr>
<tr>
<td>IBN</td>
<td>-</td>
<td>1.0</td>
<td>≥ 0.1</td>
<td>Saturation voltage factor for impact ionization</td>
</tr>
</tbody>
</table>

## Intrinsic Model Temperature Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCV</td>
<td>V/K</td>
<td>1.0E-3</td>
<td>Threshold voltage temperature coefficient</td>
</tr>
<tr>
<td>BEX</td>
<td>-</td>
<td>-1.5</td>
<td>Mobility temperature exponent</td>
</tr>
<tr>
<td>UCEX</td>
<td>-</td>
<td>0.8</td>
<td>Longitudinal critical field temperature exponent</td>
</tr>
<tr>
<td>IBBT</td>
<td>1/K</td>
<td>9.0E-4</td>
<td>Temperature coefficient for IBB</td>
</tr>
</tbody>
</table>
Matching Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVTO</td>
<td>$V_m$</td>
<td>$0^a$</td>
<td>Area related threshold voltage mismatch parameter</td>
</tr>
<tr>
<td>AKP</td>
<td>$m$</td>
<td>0</td>
<td>Area related gain mismatch parameter</td>
</tr>
<tr>
<td>AGAMMA</td>
<td>$\sqrt{V_m}$</td>
<td>0</td>
<td>Area related body effect mismatch parameter</td>
</tr>
</tbody>
</table>

a. Only DEV values are applicable to the statistical matching parameters AVTO, AGAMMA, AKP for Monte-Carlo type simulations. Default is $1E-6$ for all three parameters in some implementations, to allow sensitivity analysis to be performed on the matching parameters. LOT specifications should not be used for AVTO, AGAMMA, AKP.

Flicker Noise Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KF</td>
<td>$a$</td>
<td>0</td>
<td>Flicker noise coefficient</td>
</tr>
<tr>
<td>AF</td>
<td>-</td>
<td>1</td>
<td>Flicker noise exponent</td>
</tr>
</tbody>
</table>

a. Unit of KF may depend on flicker noise model chosen if options are available.

Setup Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NQS$^a$</td>
<td>-</td>
<td>0</td>
<td>Non-Quasi-Static (NQS) operation switch</td>
</tr>
<tr>
<td>SATLIM$^b$</td>
<td>-</td>
<td>exp(4)</td>
<td>Ratio defining the saturation limit $i_f/i_r$</td>
</tr>
<tr>
<td>XQC$^c$</td>
<td>-</td>
<td>0.4</td>
<td>Charge/capacitance model selector</td>
</tr>
</tbody>
</table>

a. NQS=1 switches Non-Quasi-Static operation on, default is off (NQS model option may not be available in all implementations).
b. Only used for operating point information. (SATLIM option may not be available in all implementations).
c. Selector for charges/transcapacitances (default) or capacitances only model. XQC=0.4: charges/transcapacitances model; XQC=1: capacitances only model. (XQC model option may not be available in all implementations).
Static Intrinsic Model Equations

Basic Definitions

\[ \varepsilon_{si} = SCALE \cdot 104.5 \times 10^{-12} [F/m] \quad \text{Permittivity of silicon} \]

\[ \varepsilon_{ox} = SCALE \cdot 34.5 \times 10^{-12} [F/m] \quad \text{Permittivity of silicon dioxide} \]

\[ q = 1.602 \times 10^{-19} [C] \quad \text{Magnitude of electron charge} \]

\[ k = 1.3807 \times 10^{-23} [JK^{-1}] \quad \text{Boltzmann constant} \]

\[ T_{ref} = 300.15 [K] \quad \text{Reference temperature} \]

\[ T_{nom} [K] \quad \text{Nominal temperature of model parameters} \]

\[ T [K] \quad \text{Model simulation temperature} \]

\[ V_{t}(T) = \frac{k \cdot T}{q} \quad \text{Thermal voltage} \]

\[ E_g(T) = \left( 1.16 - 0.000702 \cdot \frac{T^2}{T + 1108} \right) [eV] \quad \text{Energy gap} \]

\[ n_i(T) = 1.45 \times 10^{16} \cdot \left( \frac{T}{T_{ref}} \right) \exp \left( \frac{E_g(T_{ref})}{2 \cdot V_t(T_{ref})} - \frac{E_g(T)}{2 \cdot V_t(T)} \right) [m^{-3}] \quad \text{Intrinsic carrier concentration} \]

Parameter Preprocessing

Handling of Model Parameters for P-channel MOSFETs

For P-channel devices, the sign of \( V_{FB} \), \( V_{TO} \) and \( T_{CV} \) is inversed before processing. Therefore, \( V_{TO} \) and \( T_{CV} \) are usually positive and \( V_{FB} \) negative for N-channel, and vice versa for P-channel MOSFETs.
Intrinsic Parameters Initialization

The basic intrinsic model parameters \( COX \), \( GAMMA \), \( PHI \), \( VTO \), \( KP \) and \( UCRIT \) are related to the fundamental process parameters \( TOX \), \( NSUB \), \( VFB \), \( UO \), \( VMAX \), respectively, similarly as in early SPICE models. For the purpose of statistical circuit simulation, it is desirable to introduce parameter variations on the level of the latter parameters. These dependencies are also of interest if device scaling is to be analyzed, and are useful when parameter sets should be obtained from other MOSFET models. Therefore, the possibility is introduced to use the following relations:

If \( COX \) is not specified, then it is initialized as:

\[
COX = \begin{cases} \frac{\varepsilon_{ox}}{TOX} & \text{for: } TOX > 0 \\ \text{default} & \text{otherwise} \end{cases}
\]

If \( GAMMA \) is not specified, then it is initialized as:

\[
GAMMA = \begin{cases} \sqrt{\frac{2q\varepsilon_{si} \cdot (NSUB \cdot 10^6)}{COX}} & \text{for: } NSUB > 0 \\ \text{default} & \text{otherwise} \end{cases}
\]

If \( PHI \) is not specified, then it is initialized as:

\[
PHI = \begin{cases} 2V_T(T_{nom}) \cdot \ln\left(\frac{NSUB \cdot 10^6}{n_i(T_{nom})}\right) & \text{for: } NSUB > 0 \\ \text{default} & \text{otherwise} \end{cases}
\]

If \( VTO \) is not specified, then it is initialized as:

\[
VTO = \begin{cases} VFB + PHI + GAMMA \cdot \sqrt{PHI} & \text{if VFB specified} \\ \text{default} & \text{otherwise} \end{cases}
\]
If DP is not specified, then it is initialized as:

\[
KP = \begin{cases} 
(UO \cdot 10^{-4}) \cdot \text{COX} & \text{for: } UO > 0 \\
\text{default} & \text{otherwise}
\end{cases}
\]

If UCRIT is not specified, then it is initialized as:

\[
UCRIT = \begin{cases} 
\frac{VMAX}{(UO \cdot 10^{-4})} & \text{for: } VMAX > 0, UO > 0 \\
\text{default} & \text{otherwise}
\end{cases}
\]

If E0 is not specified, then a simplified mobility model is used with the parameter THETA:

\[
E0 = \begin{cases} 
0 & \text{if THETA specified} \\
\text{default} & \text{otherwise}
\end{cases}
\]

**Note:** The value zero is given to \(E0\) here, indicating that the simplified mobility model is used in conjunction with \(\text{THETA}\) instead of the standard mobility model.

Note that optional parameters may not be available in all implementations.

**Default Values and Parameter Ranges**

Model parameters that are not defined in the model parameter sets are either initialized according to the above relations, or set to their default values. For certain parameters, their numerical range has to be restricted to avoid numerical problems such as divisions by zero. If a parameter given in a parameter set falls outside the specified range (see range column in the parameter tables) then its value is set to the closest acceptable value.
Intrinsic Parameters Temperature Dependence

\[ V_{TO}(T) = V_{TO} - TCV \cdot (T - T_{nom}) \]

\[ K_P(T) = K_P \cdot \left( \frac{T}{T_{nom}} \right)^{BEX} \]

\[ U_{CRIT}(T) = U_{CRIT} \cdot \left( \frac{T}{T_{nom}} \right)^{UCEX} \]

\[ \Phi_H(T) = \Phi_H \cdot \frac{T}{T_{nom}} - 3 \cdot V_t \cdot \ln \left( \frac{T}{T_{nom}} \right) - E_g(T_{nom}) \cdot \frac{T}{T_{nom}} + E_g(T) \]

\[ I_{BB}(T) = I_{BB} \cdot [1.0 + I_{BBT} \cdot (T - T_{nom})] \]

Bulk Referenced Intrinsic Voltages

Volages are all referred to the local substrate ("Bulk Reference and Symmetry" on page 22-86):

\[ V_G = V_{GB} = V_{GS} - V_{BS} \quad \text{Intrinsic gate-to-bulk voltage} \]

\[ V_S = V_{SB} = -V_{BS} \quad \text{Intrinsic source-to-bulk voltage} \]

\[ V_D = V_{DB} = V_{DS} - V_{BS} \quad \text{Intrinsic drain-to-bulk voltage} \]

For P-channel devices, all signs of the above voltages are inverted prior to processing.

Effective Channel Length and Width

\[ W_{eff} = W + DW \]

\[ L_{eff} = L + DL \]

**Note:** Contrary to the convention adopted in other MOSFET models, DL and DW usually do have a negative value due to the above definition.
Short Distance Matching

Random mismatch between two transistors with identical layout and close to each other is in most cases suitably described by a law following the inverse of the square root of the transistors’ area. The following relationships have been adopted:

\[ V_{TO_a} = V_{TO} + \frac{AV_{TO}}{\sqrt{NP \cdot W_{eff} \cdot NS \cdot L_{eff}}} \]

\[ K_{P_a} = K_P \cdot \left( 1 + \frac{AK_P}{\sqrt{NP \cdot W_{eff} \cdot NS \cdot L_{eff}}} \right) \]

\[ \Gamma_{\lambda_a} = \Gamma_{\lambda} + \frac{AG_{\Gamma_{\lambda}}}{\sqrt{NP \cdot W_{eff} \cdot NS \cdot L_{eff}}} \]

These model equations are only applicable in Monte-Carlo and sensitivity simulations.

Note that since negative values for both \( K_{P_a} \) and \( \Gamma_{\lambda_a} \) are not physically meaningful, these are clipped at zero.

Reverse Short-channel Effect (RSCE)

\[ C_\epsilon = 4 \cdot (22 \times 10^{-3})^2 \quad C_A = 0.028 \]

\[ \xi = C_A \cdot \left( 10 \cdot \frac{L_{eff}}{L_K} - 1 \right) \]

\[ \Delta V_{RSCE} = \frac{2 \cdot Q_0}{C_{OX}} \cdot \frac{1}{\left[ 1 + \frac{1}{2} \cdot (\xi + \sqrt{\xi^2 + C_\epsilon}) \right]^2} \]

Effective Gate Voltage Including RSCE

\[ V_{G'} = V_G - V_{TO_a} - \Delta V_{RSCE} + \Phi + \Gamma_{\lambda_a} \sqrt{\Phi} \]
Effective substrate factor including charge-sharing for short and narrow channels

Pinch-off voltage for narrow-channel effect:

\[ V_{p0} = \begin{cases} \gamma_0 \cdot \phi - \gamma_0 \cdot \gamma \cdot \left( \frac{V_G + \left( \frac{\gamma_0^2}{2} \right)^2}{2} \right) & \text{for: } V_G > 0 \\ -\phi & \text{for: } V_G \leq 0 \end{cases} \]

Effective substrate factor accounting for charge-sharing:

\[ V'_{S(D)} = \frac{1}{2} \cdot \left[ V_{S(D)} + \phi + \sqrt{(V_{S(D)} + \phi)^2 + (4V_G)^2} \right] \]

**Note:** Equation prevents the argument of the square roots in the subsequent code from becoming negative.

\[ \gamma_0 = \text{GAMMA}_a - \frac{\varepsilon_{si}}{\text{COX}} \cdot \left[ \frac{\text{LETA}}{L_{\text{eff}}} \cdot \left( \sqrt{V'_S} + \sqrt{V'_D} \right) - \frac{3 \cdot \text{WETA}}{W_{\text{eff}}} \cdot \sqrt{V_{p0} + \phi} \right] \]

\[ \gamma' = \frac{1}{2} \cdot (\gamma_0 + \sqrt{\gamma_0^2 + 0.1 \cdot V_I}) \]

**Note:** The purpose of Equation is to prevent the effective substrate factor from becoming negative.

Pinch-off Voltage Including Short- and Narrow-channel Effects

\[ V_p = \begin{cases} \gamma' \cdot \phi - \gamma' \cdot \left( \frac{V_G + \left( \frac{\gamma'}{2} \right)^2}{2} \right) & \text{for: } V_G > 0 \\ -\phi & \text{for: } V_G \leq 0 \end{cases} \]

Note that the pinch-off voltage accounts for channel doping effects such as threshold voltage and substrate effect. For long-channel devices, \( V_p \) is a function of gate voltage; for short-channel devices, it becomes also a function of source and drain voltage due to the charge-sharing effect.
Slope Factor

\[ n = 1 + \frac{\text{GAMMA}_a}{2 \cdot \sqrt{V_p + \Phi + 4V_t}} \]

Note that the slope factor (or body effect factor), which is primarily a function of the gate voltage, is linked to the weak inversion slope.

Large Signal Interpolation Function

\( F(v) \) is the large-signal interpolation function relating the normalized currents to the normalized voltages. A simple and accurate expression for the transconductance-to-current ratio allows a consistent formulation of the static large-signal interpolation function, the dynamic model for the intrinsic charges (and capacitances) as well as the intrinsic time constant and the thermal noise model for the whole range of current from weak to strong inversion:

\[ \frac{g_{ms} \cdot V_t}{I_{DS}} = \frac{1}{\sqrt{0.25 + i + 0.5}} \]

Large-signal interpolation function:

\[
\begin{align*}
  y &= \sqrt{0.25 + i - 0.5} \\
  \nu &= 2y + \ln(y)
\end{align*}
\]

Unfortunately, Eqn. cannot be inverted analytically. However, it can be inverted using a Newton-Raphson iterative scheme. Currently, a simplification of this algorithm that avoids iteration is used, leading to a continuous expression for the large signal interpolation function. The (inverted) large signal interpolation function has the following asymptotes in strong and weak inversion respectively:

\[
F(v) = \begin{cases} 
  (v/2)^2 & \text{for: } v \gg 0 \\
  \exp(v) & \text{for: } v \ll 0 
\end{cases}
\]
Forward Normalized Current

\[ i_f = F \left[ \frac{V_p - V_S}{V_t} \right] \]

Velocity Saturation Voltage

\[ V_C = \text{UCRIT} \cdot \text{NS} \cdot L_{\text{eff}} \]

Note: Equation accounts for multiple series device number NS.

\[ V_{DSS} = V_C \cdot \left[ \frac{1}{\sqrt{4 + \frac{V_t}{V_C}} \cdot \sqrt{i_f - \frac{1}{2}}} \right] \]

Note: The variable \( V_{DSS} \) in this formulation for computer simulation is half the value of the actual saturation voltage.

Drain-to-source Saturation Voltage for Reverse Normalized Current

\[ V_{DSS}' = V_C \cdot \left[ \frac{1}{\sqrt{4 + \frac{V_t}{V_C}} \cdot \left( \sqrt{i_f - \frac{3}{4}} \cdot \ln(i_f) \right) - \frac{1}{2}} \right] + V_t \cdot \left[ \ln \left( \frac{V_C}{2V_t} \right) - 0.6 \right] \]

Channel-length Modulation

\[ \Delta V = 4 \cdot V_t \cdot \sqrt{\text{LAMBDA} \cdot \left( \sqrt{i_f - \frac{V_{DSS}}{V_t}} \right) + \frac{1}{64}} \]

\[ V_{ds} = \frac{V_D - V_S}{2} \]

\[ V_{ip} = \sqrt{V_{DSS}^2 + \Delta V^2 - \sqrt{(V_{ds} - V_{DSS})^2 + \Delta V^2}} \]
\[ L_C = \frac{e_{si}}{\sqrt{\epsilon_{si} \cdot XJ}} \]

\[ \Delta L = \text{LAMBDA} \cdot L_C \cdot \ln \left( 1 + \frac{V_{ds} - V_{ip}}{L_C \cdot UCRIT} \right) \]

**Equivalent Channel Length Including Channel-length Modulation and Velocity Saturation**

\[ L' = \text{NS} \cdot L_{eff} - \Delta L + \frac{V_{ds} + V_{ip}}{UCRIT} \]

\[ L_{min} = \text{NS} \cdot L_{eff}/10 \]

**Note:** Equation and Equation account also for multiple series device number NS.

\[ L_{eq} = \frac{1}{2} \cdot (L' + \sqrt{L'^2 + L_{min}^2}) \]

**Note:** Equation prevents the equivalent channel length to become zero or negative.

**Reverse Normalized Current**

Reverse normalized current:

\[ r' = F \left[ \frac{V_p - V_{ds} - V_S - \sqrt{V_{DSS}^2 + \Delta V^2} + \sqrt{(V_{ds} - V_{DSS}^2 + \Delta V^2)}}{V_t} \right] \]

Reverse normalized current for mobility model, intrinsic charges/capacitances, thermal noise model and NQS time-constant:
Transconductance Factor and Mobility Reduction Due to Vertical Field

\[ \beta_0 = \frac{K_P a}{L_{eq}} \cdot \frac{NP \cdot W_{eff}}{L_{eq}} \]

Note that the use of the device parameter NP (or M) gives accurate results for simulation of parallel devices, whereas the use of NS (or N) for series devices is only approximate. Note that \( L_{eq} \) accounts for multiple series device number NS.

\[ \eta = \begin{cases} 
\frac{1}{2} & \text{for NMOS} \\
\frac{1}{3} & \text{for PMOS} 
\end{cases} \]

\[ q_{B0} = \text{GAMMA}_a \cdot \sqrt{\Phi I} \]

\[ \beta_0' = \frac{\beta_0}{1 + \frac{\text{COX} \cdot E_0 \cdot \varepsilon_{si}}{\varepsilon_{B0}} \cdot q_{B0}} \]

\[ \beta = \frac{\beta_0'}{1 + \frac{\text{COX} \cdot E_0 \cdot \varepsilon_{si}}{V_t \cdot \varepsilon_{B0}} \cdot q_{B0} + \eta \cdot q_I} \]

For the definition of the normalized depletion and inversion charges \( q_B \) and \( q_I \) refer to the section on the node charges. The use of \( \beta_0' \) ensures that \( \beta = \beta_0 \) when \( q_I \ll q_B \). The formulation of \( \beta \) arises from the integration of the local effective field as a function of depletion and inversion charge densities along the channel. No substrate bias dependency is needed due to the inclusion of depletion charge. Note that the resulting mobility expression also depends on \( V_{DS} \).
Simple Mobility Reduction Model

For reasons of compatibility with the former EKV model versions (EKV model versions prior to v2.6), a possibility is introduced to choose the simpler mobility reduction model that uses the parameter THETA. In case the model parameter $E_0$ is *not* specified (see parameter preprocessing), the simpler mobility model is taken into account according to:

$$V_p' = \frac{1}{2} \cdot (V_p + \sqrt{V_p^2 + 2V_i^2})$$

$$\beta = \frac{\beta_0}{1 + THETA \cdot V_p'}$$

Specific Current

$$I_S = 2 \cdot n \cdot \beta \cdot V_i^2$$

Drain-to-source Current

$$I_{DS} = I_S \cdot (i_f - i_r)$$

For P-channel devices, $I_{DS}$ is given a negative sign.

*Note: This drain current expression is a single equation, valid in all operating regions: weak, moderate and strong inversion, non-saturation and saturation. It is therefore not only continuous among all these regions, but also continuously derivable.*

Transconductances

The transconductances are obtained through derivation of the drain current:

$$g_{mg} \equiv \frac{\partial I_{DS}}{\partial V_G} \quad g_{ms} \equiv -\frac{\partial I_{DS}}{\partial V_S} \quad g_{md} \equiv \frac{\partial I_{DS}}{\partial V_{DS}}$$

Note the following relationships with the derivatives where the source is taken as reference:
The analytic derivatives are available with the model code.

**Impact Ionization Current**

\[ V_{ib} = V_D - V_S - IBN \cdot 2 \cdot V_{DSS} \]

\[ I_{DB} = \begin{cases} 
I_{DS} \cdot \frac{IBA}{IBB} \cdot V_{ib} \cdot \exp\left(\frac{-IBB \cdot L}{V_{ib}}\right) & \text{for: } V_{ib} > 0 \\
0 & \text{for: } V_{ib} \leq 0 
\end{cases} \]

Note that the factor 2 in the expression for \( V_{ib} \) accounts for the fact that the numerical value of \( V_{DSS} \) is half the actual saturation voltage. Further note that the substrate current is intended to be treated as a component of the total extrinsic drain current, flowing from the drain to the bulk. The total drain current is therefore expressed as \( I_D = I_{DS} + I_{DB} \).

The substrate current therefore also affects the total extrinsic conductances, in particular the drain conductance.

**Quasi-static Model Equations**

Both a charge-based model for transcapacitances, allowing charge-conservation during transient analysis, and a simpler capacitances-based model are available. Note that the charges model is formulated in symmetric terms of the forward and reverse normalized currents, that is, symmetrical for both drain and source sides.

Further note that short-channel effects, as charge-sharing and reverse short-channel effects, are included in the dynamic model through the pinch-off voltage.
Dynamic Model for the Intrinsic Node Charges

\[ n_q = 1 + \frac{\text{GAMMA}_d}{2 \cdot \sqrt{V_p + \Phi I + 10^{-6}}} \]

Normalized Intrinsic Node Charges:

\[ x_f = \frac{1}{\sqrt{4}} + i_f \]
\[ x_r = \frac{1}{\sqrt{4}} + i_r \]

\[ q_D = -n_q \cdot \left( \frac{4}{15} \cdot \frac{3x_f^3 + 6x_f^2x_r + 4x_f^2x_r^2 + 2x_r^3}{(x_f + x_r)^2} - \frac{1}{2} \right) \]

\[ q_S = -n_q \cdot \left( \frac{4}{15} \cdot \frac{3x_r^3 + 6x_r^2x_f + 4x_r^2x_f^2 + 2x_f^3}{(x_f + x_r)^2} - \frac{1}{2} \right) \]

\[ q_I = q_S + q_D = -n_q \cdot \left( \frac{4}{3} \cdot \frac{x_f^2 + x_f^2x_r + x_r^2}{x_f + x_r} - 1 \right) \]

\[ q_B = \begin{cases} 
  (-\text{GAMMA}_d \cdot \sqrt{V_p + \Phi I + 10^{-6}}) \cdot \frac{1}{V_t} \cdot \left( \frac{n_q - 1}{n_q} \right) \cdot q_I & \text{for: } V'_G > 0 \\
  -V'_G \cdot \frac{1}{V_t} & \text{for: } V'_G \leq 0 
\end{cases} \]

\[ q_G = -q_I - q_{OX} - q_B \]

\( q_{OX} \) is a fixed oxide charge assumed to be zero. The above equation expresses the charge conservation among the four nodes of the transistor.
Total Charges:
\[ C_{ox} = \text{COX} \cdot \text{NP} \cdot W_{eff} \cdot N_S \cdot L_{eff} \]
\[ Q(I, B, D, S, G) = C_{ox} \cdot V_t \cdot q(I, B, D, S, G) \]

Intrinsic Capacitances

Transcapacitances

The intrinsic capacitances are obtained through derivation of the node charges with respect to the terminal voltages:
\[ C_{xy} = \pm \frac{\partial}{\partial V_y}(Q_x) \quad x, y = G, D, S, B \]

where the positive sign is chosen when \( x = y \) and the negative sign otherwise. This results in simple and continuous analytical expressions for all the transcapacitances in terms of \( x_f, x_r \), the pinch-off voltage and the slope factor, and derivatives thereof, from weak to strong inversion and non-saturation to saturation.

Normalized Intrinsic Capacitances

A simplified capacitive dynamic model, using the five intrinsic capacitances corresponding to the “Equivalent Circuit” on page 22-87, can be obtained when neglecting the slight bias dependence of the slope factor \( n \), resulting in the following simple functions:

\[ c_{gs} = \frac{2}{3} \cdot \left( 1 - \frac{x_r^2 + x_f + \frac{1}{2} x_f}{(x_f + x_r)^2} \right) \]
\[ c_{gd} = \frac{2}{3} \cdot \left( 1 - \frac{x_r^2 + x_f + \frac{1}{2} x_f}{(x_f + x_r)^2} \right) \]
Note that this simplified capacitances model can be chosen by setting $X_{QC} = 1$.

**Total Intrinsic Capacitances**

$$C_{(gs, gd, gb, sb, db)} = C_{ox} \cdot c_{(gs, gd, gb, sb, db)}$$

### Non-Quasi-Static (NQS) Model Equations

The EKV model includes a first order NQS model for small-signal (.AC) simulations. The expression of the NQS drain current is obtained from the quasi-static value of the drain current which is then 1st-order low-pass filtered. NQS is a flag (model parameter) allowing to disable the NQS model and $\tau$ is the bias dependent characteristic time constant.

$\tau_0$ is the intrinsic time constant defined as:

$$\tau_0 = \frac{(N_{S} \cdot L_{eff})^2}{2 \cdot \mu_{eff} \cdot V_t} = \frac{C_{ox}}{2 \cdot V_t \cdot \beta}$$

$$\tau = \tau_0 \cdot \frac{4}{15} \cdot \frac{(x_f^2 + 3x_f x_r + x_r^2)}{(x_f + x_r)^3}$$

$$I_{DS}(s) = \frac{I_{DS}}{1 + NQS \cdot s \cdot \tau}$$

The corresponding small-signal (.AC) transadmittances are then given by:

$$Y_{mg}(s) = \frac{g_{mg}}{1 + NQS \cdot s \cdot \tau}$$
Intrinsic Noise Model Equations

The noise is modeled by a current source $I_{NDS}$ between intrinsic source and drain. It is composed of a thermal noise component and a flicker noise component and has the following Power Spectral Density (PSD):

$$S_{INDS} = S_{thermal} + S_{flicker}$$

**Thermal Noise**

The thermal noise component PSD is given by:

$$S_{thermal} = 4kT \cdot \frac{\mu_{eff}}{(N_S \cdot L_{eff})^2} \cdot |Q_I| = 4kT \cdot \beta \cdot |q_I|$$

Note that the above thermal noise expression is valid in all regions of operation, including for small $V_{DS}$.

**Flicker Noise**

The flicker noise component PSD is given by:

$$S_{flicker} = \frac{K F \cdot g_{mg}^2}{NP \cdot \overline{W_{eff}} \cdot N_S \cdot L_{eff} \cdot COX \cdot f_{AF}}$$

Note that in some implementations, different expressions are accessible.
Operating Point Information

At operating points, the following information should be displayed as a help for circuit design:

**Numerical values of model internal variables:**

\[ V_G, V_S, V_D, I_{DS}, I_{DB}, g_{mg}, g_{ms}, g_{msr}, g_{md}, V_P, n, \beta, I_S, i_f, i_r, i_r', \tau, \tau_0 \]

intrinsic charges/capacitances

**Transconductance efficiency factor:**

\[ tef = g_{ms} \cdot V_t / I_{DS} \]

‘Early voltage’:

\[ VM = I_{DS} / g_{md} \]

‘Overdrive voltage’:

\[ n \cdot (V_p - V_s) = V_G - V_{TO} - n \cdot V_S \]

For P-channel devices, \( n \cdot (V_p - V_s) \) is given a negative sign.

‘SPICE-like’ threshold voltage:

\[ VTH = V_{TO} + \Delta V_{RSCE} + \gamma' \cdot \sqrt{V'_S - \text{GAMMA}_a \cdot \sqrt{PHI}} \]

**Note:** This expression is the ‘SPICE-like’ threshold voltage, referred to the source. It accounts also for charge-sharing and reverse short-channel effects on the threshold voltage.

For P-channel devices, \( VTH \) is given a negative sign.

**Saturation voltage:**

\[ V_{DSAT} = 2V_{DSS} + 4V_t \]

For P-channel devices, \( V_{DSAT} \) is given a negative sign.
Saturation / non-saturation flag:

\[
\text{'SAT' or '1' for } \frac{i_f}{i_r} > \text{SATLIM}
\]

\[
\text{'LIN' or '0' for } \frac{i_f}{i_r} \leq \text{SATLIM}
\]

**Note:** Implementation of operating point information may differ in some simulators (i.e. not all of the information may be available).

### Estimation and Limits of Static Intrinsic Model Parameters

The EKV intrinsic model parameters can roughly be estimated from SPICE level 2/3 parameters as indicated in the table below, if no parameter extraction facility is available. Attention has to be paid to units of the parameters. This estimation method can be helpful and generally gives reasonable results. Nevertheless, be aware that the underlying modeling in SPICE LEVEL 2/3 and in the EKV model is not the same, even if the names and the function of several parameters are similar. Therefore, it is preferred if parameter extraction is made directly from measurements.

Lower and upper limits indicated in the table should give an idea on the order of magnitude of the parameters but do not necessarily correspond to physically meaningful limits, nor to the range specified in the parameter tables. These limits may be helpful for obtaining physically meaningful parameter sets when using nonlinear optimization techniques to extract EKV model parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Example</th>
<th>‘Lower’</th>
<th>‘Upper’</th>
<th>Estimationa</th>
</tr>
</thead>
<tbody>
<tr>
<td>COX</td>
<td>F/m²</td>
<td>0.7E-3</td>
<td>3.45E-3</td>
<td>-</td>
<td>-</td>
<td>$\varepsilon_{ox}/\text{TOX}$</td>
</tr>
<tr>
<td>XJ</td>
<td>m</td>
<td>0.1E-6</td>
<td>0.15E-6</td>
<td>0.01E-6</td>
<td>1E-6</td>
<td>XJ</td>
</tr>
<tr>
<td>VTO</td>
<td>V</td>
<td>0.5</td>
<td>0.7</td>
<td>0</td>
<td>2</td>
<td>VTO</td>
</tr>
<tr>
<td>GAMMA</td>
<td>$\sqrt{N}$</td>
<td>1.0</td>
<td>0.7</td>
<td>0</td>
<td>2</td>
<td>$\sqrt{2q\varepsilon_{si}} \cdot \text{NSUB}/\text{COX}$</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default</td>
<td>Example</td>
<td>‘Lower’</td>
<td>‘Upper’</td>
<td>Estimation(^a)</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>------------</td>
<td>---------</td>
<td>----------</td>
<td>----------</td>
<td>------------------</td>
</tr>
<tr>
<td>PHI(^b)</td>
<td>V</td>
<td>0.7</td>
<td>0.5</td>
<td>0.3</td>
<td>2</td>
<td>(2V_t \cdot \ln(\text{NSUB}/n_i))</td>
</tr>
<tr>
<td>KP</td>
<td>A/V(^2)</td>
<td>50E-6</td>
<td>150E-6</td>
<td>1E-6</td>
<td>-</td>
<td>(\text{UO} \cdot \text{COX})</td>
</tr>
<tr>
<td>E0</td>
<td>V/m</td>
<td>1.0E12</td>
<td>200E6</td>
<td>0.1/(0.4 \cdot \text{TOX})</td>
<td>-</td>
<td>(0.2/(\text{THETA} \cdot \text{TOX}))</td>
</tr>
<tr>
<td>UCRIT</td>
<td>V/m</td>
<td>2.0E6</td>
<td>2.3E6</td>
<td>1.0E6</td>
<td>2.5E6</td>
<td>(\text{VMAX}/\text{UO})</td>
</tr>
<tr>
<td>DL</td>
<td>m</td>
<td>0</td>
<td>-0.15(L(_\text{min}))</td>
<td>-0.5(L(_\text{min}))</td>
<td>0.5(L(_\text{min}))</td>
<td>\text{XL} - 2 \cdot \text{LD}</td>
</tr>
<tr>
<td>DW</td>
<td>m</td>
<td>0</td>
<td>-0.1(W(_\text{min}))</td>
<td>-0.5(W(_\text{min}))</td>
<td>0.5(W(_\text{min}))</td>
<td>\text{XW} - 2 \cdot \text{WD}</td>
</tr>
<tr>
<td>LAMBD</td>
<td>-</td>
<td>0.5</td>
<td>0.8</td>
<td>0</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>LETA</td>
<td>-</td>
<td>0.1</td>
<td>0.3</td>
<td>0</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>WETA</td>
<td>-</td>
<td>0.25</td>
<td>0.2</td>
<td>0</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>Q0</td>
<td>As/m(^2)</td>
<td>0.0</td>
<td>230E-6</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LK</td>
<td>m</td>
<td>0.29E-6</td>
<td>0.4E-6</td>
<td>0.05E-6</td>
<td>2E-6</td>
<td>-</td>
</tr>
<tr>
<td>IBA</td>
<td>l/m</td>
<td>0.0</td>
<td>2.0E8</td>
<td>0.0</td>
<td>5.0E8</td>
<td>(\text{ALPHA} \cdot \text{VCR}/L_C)</td>
</tr>
<tr>
<td>IBB</td>
<td>V/m</td>
<td>3.0E8</td>
<td>2.0E8</td>
<td>1.8E8</td>
<td>4.0E8</td>
<td>(\text{VCR}/L_C)</td>
</tr>
<tr>
<td>IBN</td>
<td>-</td>
<td>1.0</td>
<td>0.6</td>
<td>0.4</td>
<td>1.0</td>
<td>-</td>
</tr>
</tbody>
</table>

\(^a\) Also compare with optional process parameters.

\(^b\) The minimum value of PHI also determines the minimum value of the pinch-off voltage. Due to the intrinsic temperature dependence of PHI, a lower value results for higher temperature, limiting the range of simulation for small currents.

\[
\begin{align*}
\varepsilon_{\text{ox}} &= 0.0345E-9 \text{ F/m} \\
q &= 1.609E-19 \text{ C} \\
k &= 1.381E-23 \text{ J/K} \\
L_c &= \sqrt{\varepsilon_{\text{si}} \cdot \text{XJ}/\text{COX}} \\
\varepsilon_{\text{si}} &= 0.104E-9 \text{ F/m} \\
n_i &= 1.45E16 \text{ m}^{-3} \\
V_t &= kT/q = 0.0259 \text{ V (at room temperature)}
\end{align*}
\]

**Note:** Parameters in this table suppose \( m \) (meter) has been chosen as length unit. \( L_{\text{min}} \) and \( W_{\text{min}} \) are the minimum drawn length and width of the transistors. Example values are indicated for enhancement N-channel devices.
Model Updates Description

Throughout the use of the EKV v2.6 MOSFET model by many designers, several enhancements have appeared to be necessary from the model formulation point of view, or desirable from the point of view of the application of the model. This paragraph provides a summary of the updates to the EKV v2.6 model formulation and documentation since its first release. Wherever possible, backward compatibility with former formulations is maintained.

Revision I, September 1997

*Description:* Narrow channel effect on the substrate factor is revised to improve the transcapacitances behavior. The narrow channel effect is not anymore a function of the source voltage $V_S$, but of the pinch-off voltage $V_p$.

*Consequence:* the narrow channel effect parameters $WETA, DW$ require different numerical values to achieve the same effect.

Revision II, July 1998

Intrinsic time constant

*Description:* Intrinsic time constant $\tau_0$ is calculated as a function of the effective $\beta$ factor (including vertical field dependent mobility and short-channel effects) instead of maximum mobility according to the $KP$ parameter.

*Consequence:* the NQS time constant has an additional gate voltage dependence, resulting in more conservative (lower) estimation of the NQS time constant at high $V_G$, and additional dependence on short-channel effects.

Thermal noise

*Description:* Thermal noise power spectral density $S_{thermal}$ is calculated as a function of the effective $\beta$ factor (including vertical field dependent mobility and short-channel effects) instead of maximum mobility according to the $KP$ parameter.

*Consequence:* $S_{thermal}$ has an additional gate voltage and short-channel effect dependence.
Optional process parameters for calculation of electrical intrinsic parameters

*Description:* The option is introduced to calculate the electrical parameters $\text{COX}$, $\text{GAMMA}$, and/or $\text{PHI}$, $\text{VTO}$, $\text{KP}$, and $\text{UCRIT}$ as a function of the optional parameters $\text{TOX}$, $\text{NSUB}$, $\text{VFB}$, $\text{UO}$, and $\text{VMAX}$, respectively. $\text{NSUB}$ and $\text{UO}$ have $\text{cm}$ as length units.

*Consequence:* This accommodates scaling behavior and allows more meaningful statistical circuit simulation due to decorrelation of physical effects. Compatible with former revisions except for default calculation of the parameters mentioned, if the optional parameters are specified.

Optional simplified mobility model

*Description:* The simple mobility model of former model versions, using the parameter $\text{THETA}$, is reinstated as an option.

*Consequence:* Simplifies adaptation from earlier model versions to the current version.

Parameter synonyms

*Description:* The parameters $\text{E0}$ and $\text{Q0}$ can be called by their synonyms $\text{EO}$ and $\text{QO}$, respectively.

*Consequence:* Accommodates certain simulators where only alphabetic characters are allowed.

Operating point information

*Description:* The analytical expression for the ‘SPICE’-like threshold voltage $V_{\text{TH}}$ in the operating point information is modified to include charge-sharing and reverse short-channel effects. The analytical expression for the saturation voltage $V_{\text{DSAT}}$ in the operating point information is modified such that its value is non-zero in weak inversion.

*Consequence:* Improved information for the designer.
**Corrections from EPFL R11, March, 1999**

Equation 45, Equation 53, Equation 54, and Equation 58 have been corrected for multiple series device behavior with the parameter $N_S$.

**Corrections from EPFL R12, July 30, 1999**

The following corrections were released by EPFL.

Correction 1- 99/07/30 mb (r12) corrected $d\Gamma_dV_G$ (narrow channel). An error has been detected in the analytical model derivatives. It is in the derivatives of $\Gamma$ variable, affecting the transconductances and transcapacitances.

Correction 2- 99/07/30 mb (r12) preventing $PHI$ from being smaller than 0.2 at init and after temperature update. For certain CMOS technologies, parameter values for the parameter $PHI$ turn out to be as low as 400mV required to account for particular process details. When increasing the temperature from room temperature, $PHI$ decreases due to its built-in temperature dependence, thus making $PHI$ attain very low values, or even be negative, when reaching 100degC. To allow the model to function at these temperatures, a lower limit for $PHI$ is introduced (200mV). Note: the usual range for this parameter is well above this value (600mV to 1V).

Correction 3- 99/06/28 mb (r12) corrected $COX$ and $KP$ initialization (rg).

Correction 4- 99/05/04 mb (r12) completed parameters init for $XQC$, $DL$, $DW$, removed $IBC$, $ibc$ (cd).
LEVEL 57 UC Berkeley BSIM3-SOI Model

The UC Berkeley SOI model (BSIM3SOI) supports Fully Depleted (FD), Partially Depleted (PD), and Dynamically Depleted (DD) SOI devices, of which BSIM3PD2.0.1 for PD SOI devices is now installed in Star-Hspice as LEVEL 57. This model is described in the “BSIM3PD2.0 MOSFET MODEL User’ Manual,” which can be found at “http://www-device.eecs.berkeley.edu/~bsim3soi”.

The general syntax for including a BSIM3/SOI MOSFET element in a Star-Hspice netlist is:

General Form

\[ M_{xxx} \text{ nd ng ns ne } <\text{np}> <\text{nb}> <\text{nT}> \text{ mname } <\text{L}=\text{val}> \]
\[ + <\text{W}=\text{val}> <\text{M}=\text{val}> <\text{AD}=\text{val}> <\text{AS}=\text{val}> <\text{PD}=\text{val}> <\text{PS}=\text{val}> \]
\[ + <\text{NRD}=\text{val}> <\text{NRS}=\text{val}> <\text{NRB}=\text{val}> <\text{RTH0}=\text{val}> <\text{CTH0}=\text{val}> \]
\[ + <\text{NBC}=\text{val}> <\text{NSEG}=\text{val}> <\text{PDBC}=\text{val}> <\text{PSBC}=\text{val}> \]
\[ + <\text{AGBC}=\text{val}> <\text{AEB}=\text{val}> <\text{VBSUSR}=\text{val}> <\text{TNODEOUT}=\text{val}> \]
\[ + <\text{off}> <\text{FRBODY}=\text{val}> <\text{BJT}=\text{val}> <\text{IC}=\text{Vds, Vgs, Vbs, Ves, Vps}> \]

where the angle brackets indicate optional parameters.

The arguments are as follows:

- **Mxxx**: SOI MOSFET element name. Must begin with M, which can be followed by up to 1023 alphanumeric characters.
- **nd**: Drain terminal node name or number
- **ng**: Front gate node name or number
- **ns**: Source terminal node name or number
- **ne**: Back gate (or substrate) node name or number
- **np**: External body contact node name or number
- **nb**: Internal body node name or number
- **nT**: Temperature node name or number
- **mname**: MOSFET model name reference
$L$ SOI MOSFET channel length in meters. This parameter overrides DEFL in an OPTIONS statement. Default=DEFL with a maximum of 0.1m.

$W$ MOSFET channel width in meters. This parameter overrides DEFW in an OPTIONS statement. Default=DEFW.

$M$ Multiplier to simulate multiple SOI MOSFETs in parallel. All channel widths, diode leakages, capacitances and resistances are affected by the setting of $M$. Default=1.

$AD$ Drain diffusion area. Overrides DEFAD in the OPTIONS statement. Default=DEFAD.

$AS$ Source diffusion area. Overrides DEFAS in the OPTIONS statement. Default=DEFAS.

$PD$ Perimeter of the drain junction, including the channel edge. Overrides DEFPD in the OPTIONS statement.

$PS$ Perimeter of the source junction, including the channel edge. Overrides DEFPS in the OPTIONS statement.

$NRD$ Number of squares of drain diffusion for drain series resistance. Overrides DEFNRD in the OPTIONS statement.

$NRS$ Number of squares of source diffusion for source series resistance. Overrides DEFNRS in the OPTIONS statement.

$NRB$ Number of squares for body series resistance.

$FRBODY$ Coefficient of distributed body resistance effects default = 1.0

$RTH0$ Thermal resistance per unit width

- If not specified, $RTH0$ is extracted from the model card.
- If specified, it will override the one in the model card.
CTH0  Thermal capacitance per unit width
  ■ If not specified, CTH0 is extracted from model card.
  ■ If specified, it will override the one in the model card.

NBC  Number of body contact isolation edge

NSEG  Number of segments for channel width partitioning

PDBCP  Parasitic perimeter length for the body contact at drain side

PSBCP  Parasitic perimeter length for the body contact at source side

AGBCP  Parasitic gate-to-body overlap area for body contact

AEBCP  Parasitic body-to-substrate overlap area for body contact

VBSUSR  Optional initial value of Vbs specified by user for transient analysis

TNODEOUT  Temperature node flag indicating the usage of T node

OFF  Sets initial condition to OFF for this element in DC analysis

BJTOFF  Turning off BJT if equal to 1

IC  Initial guess in the order (drain, front gate, internal body, back gate, external voltage). (Vps will be ignored in the case of 4-terminal device) These are used when UIC is present in the .TRAN statement and are overridden by the .IC statement.

Notes:
  ■ If TNODEOUT is not set, specifying 4 nodes for a device floats the body, specifying 5 nodes implies the fifth node is the external body contact node with a body resistance put between the internal and the external terminals. This configuration applies to a distributed body resistance simulation.
If TNODEOUT is set, the last node is interpreted as the temperature node. In this case, specifying 5 nodes floats the device, specifying 6 nodes implies a body-contacted case. If user specifies seven nodes, it is a body-contacted case with an accessible internal body node. The temperature node is useful for thermal coupling simulation.

**LEVEL 57 Model Parameters**

**Model Control Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td>-</td>
<td>-</td>
<td>LEVEL 57 for BSIM3SOI</td>
</tr>
<tr>
<td>SHMOD</td>
<td>-</td>
<td>0</td>
<td>Flag for self-heating: 0 - no self-heating, 1 - self-heating</td>
</tr>
<tr>
<td>MOBMOD</td>
<td>-</td>
<td>1</td>
<td>Mobility model selector</td>
</tr>
<tr>
<td>CAPMOD</td>
<td>-</td>
<td>2</td>
<td>Flag for the short channel capacitance model</td>
</tr>
<tr>
<td>NOIMOD</td>
<td>-</td>
<td>1</td>
<td>Flag for noise model</td>
</tr>
</tbody>
</table>
### Process Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSI</td>
<td>m</td>
<td>1.0e-7</td>
<td>Silicon film thickness</td>
</tr>
<tr>
<td>TBOX</td>
<td>m</td>
<td>3.0e-7</td>
<td>Buried oxide thickness</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>1.0e-8</td>
<td>Gate oxide thickness</td>
</tr>
<tr>
<td>XJ</td>
<td>m</td>
<td>-</td>
<td>S/D junction depth</td>
</tr>
<tr>
<td>NCH</td>
<td>1/cm³</td>
<td>1.7e17</td>
<td>Channel doping concentration</td>
</tr>
<tr>
<td>NSUB</td>
<td>1/cm³</td>
<td>6.0e16</td>
<td>Substrate doping concentration</td>
</tr>
<tr>
<td>NGATE</td>
<td>1/cm³</td>
<td>0</td>
<td>Poly gate doping concentration</td>
</tr>
</tbody>
</table>

### DC Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTH0</td>
<td>v</td>
<td>NMOS 0.7 PMOS -0.7</td>
<td>Threshold voltage @ Vbs=0 for long wide device</td>
</tr>
<tr>
<td>K1</td>
<td>V¹/²</td>
<td>0.6</td>
<td>First-order body effect coefficient</td>
</tr>
<tr>
<td>K1W1</td>
<td>m</td>
<td>0</td>
<td>First-order effect width dependent parameter</td>
</tr>
<tr>
<td>K1W2</td>
<td>m</td>
<td>0</td>
<td>Second-order effect width dependent parameter</td>
</tr>
<tr>
<td>K2</td>
<td>-</td>
<td>0</td>
<td>Second-order body effect coefficient</td>
</tr>
<tr>
<td>K3</td>
<td>-</td>
<td>0</td>
<td>Narrow coefficient</td>
</tr>
<tr>
<td>K3B</td>
<td>1/V</td>
<td>0</td>
<td>Body effect coefficient of k3</td>
</tr>
<tr>
<td>KB1</td>
<td>-</td>
<td>1</td>
<td>Backgate body charge coefficient</td>
</tr>
<tr>
<td>W0</td>
<td>m</td>
<td>0</td>
<td>Narrow width parameter</td>
</tr>
<tr>
<td>NLX</td>
<td>m</td>
<td>1.74e-7</td>
<td>Lateral non-uniform doping parameter</td>
</tr>
</tbody>
</table>
### Parameter Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVT0</td>
<td>-</td>
<td>2.2</td>
<td>First coefficient of short-channel effect on Vth</td>
</tr>
<tr>
<td>DVT1</td>
<td>-</td>
<td>0.53</td>
<td>Second coefficient of short-channel effect on Vth</td>
</tr>
<tr>
<td>DVT2</td>
<td>1/V</td>
<td>-0.032</td>
<td>Body-bias coefficient of short-channel effect on Vth</td>
</tr>
<tr>
<td>DVT0W</td>
<td>-</td>
<td>0</td>
<td>First coefficient of narrow width effect on Vth for small channel length</td>
</tr>
<tr>
<td>DVT1W</td>
<td>-</td>
<td>5.3e6</td>
<td>Second coefficient of narrow width effect on Vth for small channel length</td>
</tr>
<tr>
<td>DVT2W</td>
<td>1/V</td>
<td>-0.032</td>
<td>Body-bias coefficient of narrow width effect on Vth for small channel length</td>
</tr>
<tr>
<td>U0</td>
<td>cm²/(V·sec)</td>
<td>NMOS-670 PMOS-250</td>
<td>Mobility at Temp=Tnom</td>
</tr>
<tr>
<td>UA</td>
<td>m/V</td>
<td>2.25e-9</td>
<td>First-order mobility degradation coefficient</td>
</tr>
<tr>
<td>UB</td>
<td>(m/V)²</td>
<td>5.87e-19</td>
<td>Second-order mobility degradation coefficient</td>
</tr>
<tr>
<td>UC</td>
<td>1/V</td>
<td>-0.0465</td>
<td>Body-effect of mobility degradation coefficient</td>
</tr>
<tr>
<td>VSAT</td>
<td>m/sec</td>
<td>8e4</td>
<td>Saturation velocity at Temp=Tnom</td>
</tr>
<tr>
<td>A0</td>
<td>-</td>
<td>1.0</td>
<td>Bulk charge effect coefficient for channel length</td>
</tr>
<tr>
<td>AGS</td>
<td>1/V</td>
<td>0.0</td>
<td>Gate bias coefficient of A_{bulk}</td>
</tr>
<tr>
<td>B0</td>
<td>m</td>
<td>0.0</td>
<td>Bulk charge effect coefficient for channel width</td>
</tr>
<tr>
<td>B1</td>
<td>m</td>
<td>0.0</td>
<td>Bulk charge effect width offset</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>KETA</td>
<td>1/V</td>
<td>-0.6</td>
<td>Body-bias coefficient of bulk charge effect</td>
</tr>
<tr>
<td>KETAS</td>
<td>V</td>
<td>0.0</td>
<td>Surface potential adjustment for bulk charge effect</td>
</tr>
<tr>
<td>A1</td>
<td>1/V</td>
<td>0.0</td>
<td>First non-saturation effect parameter</td>
</tr>
<tr>
<td>A2</td>
<td>-</td>
<td>1.0</td>
<td>Second non-saturation effect parameter</td>
</tr>
<tr>
<td>RDSW</td>
<td>Ω*µm^w_r</td>
<td>100</td>
<td>Parasitic resistance per unit width</td>
</tr>
<tr>
<td>PRWB</td>
<td>1/V^1</td>
<td>0</td>
<td>Body effect coefficient of Rdsw</td>
</tr>
<tr>
<td>PRWG</td>
<td>1/V^1/2</td>
<td>0</td>
<td>Gate bias effect coefficient of Rdsw</td>
</tr>
<tr>
<td>WR</td>
<td>-</td>
<td>1</td>
<td>Width offset from Weff for Rds calculation</td>
</tr>
<tr>
<td>NFACTOR</td>
<td>-</td>
<td>1</td>
<td>Subthreshold swing factor</td>
</tr>
<tr>
<td>WINT</td>
<td>m</td>
<td>0.0</td>
<td>Width offset fitting parameter from I-V without bias</td>
</tr>
<tr>
<td>LINT</td>
<td>m</td>
<td>0.0</td>
<td>Length offset fitting parameter from I-V without bias</td>
</tr>
<tr>
<td>DWG</td>
<td>m/V</td>
<td>0.0</td>
<td>Coefficient of Weff’s gate dependence</td>
</tr>
<tr>
<td>DWB</td>
<td>m/V^1/2</td>
<td>0.0</td>
<td>Coefficient of Weff’s substrate body bias dependence</td>
</tr>
<tr>
<td>DWBC</td>
<td>m</td>
<td>0.0</td>
<td>Width offset for body contact isolation edge</td>
</tr>
<tr>
<td>VOFF</td>
<td>V</td>
<td>-0.08</td>
<td>Offset voltage in the subthreshold region for large W and L</td>
</tr>
<tr>
<td>ETA0</td>
<td>-</td>
<td>0.08</td>
<td>DIBL coefficient in the subthreshold region</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>---------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>ETAB</td>
<td>1/V</td>
<td>-0.07</td>
<td>Body-bias coefficient for the subthreshold DIBL effect</td>
</tr>
<tr>
<td>DSUB</td>
<td></td>
<td>0.56</td>
<td>DIBL coefficient exponent</td>
</tr>
<tr>
<td>CIT</td>
<td>F/m²</td>
<td>0.0</td>
<td>Interface trap capacitance</td>
</tr>
<tr>
<td>CDSC</td>
<td>F/m²</td>
<td>2.4e-4</td>
<td>Drain/source to channel coupling capacitance</td>
</tr>
<tr>
<td>CDSCB</td>
<td>F/m²</td>
<td>0</td>
<td>Body-bias sensitivity of cdsc</td>
</tr>
<tr>
<td>CDSCD</td>
<td>F/m²</td>
<td>0</td>
<td>Drain-bias sensitivity of cdsc</td>
</tr>
<tr>
<td>PCLM</td>
<td></td>
<td>1.3</td>
<td>Channel length modulation parameter</td>
</tr>
<tr>
<td>PDIBLC1</td>
<td></td>
<td>0.39</td>
<td>First output resistance DIBL effect correction parameter</td>
</tr>
<tr>
<td>PDIBLC2</td>
<td></td>
<td>0.0086</td>
<td>Second output resistance DIBL effect correction parameter</td>
</tr>
<tr>
<td>DROUT</td>
<td></td>
<td>0.56</td>
<td>L dependence coefficient of the DIBL correction parameter in Rout</td>
</tr>
<tr>
<td>PVAG</td>
<td></td>
<td>0.0</td>
<td>Gate dependence of Early voltage</td>
</tr>
<tr>
<td>DELTA</td>
<td></td>
<td>0.01</td>
<td>Effective $V_{ds}$ parameter</td>
</tr>
<tr>
<td>ALPHA0</td>
<td>m/V</td>
<td>0.0</td>
<td>The first parameter of impact ionization current</td>
</tr>
<tr>
<td>FBJTII</td>
<td></td>
<td>0.0</td>
<td>Fraction of bipolar current affecting the impact ionization</td>
</tr>
<tr>
<td>BETA0</td>
<td>1/V</td>
<td>0.0</td>
<td>First $V_{ds}$ dependence parameter of impact ionization current</td>
</tr>
<tr>
<td>BETA1</td>
<td></td>
<td>0.0</td>
<td>Second $V_{ds}$ dependence parameter of impact ionization current</td>
</tr>
<tr>
<td>BETA2</td>
<td>V</td>
<td>0.1</td>
<td>Third $V_{ds}$ dependence parameter of impact ionization current</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VDSATII0</td>
<td>V</td>
<td>0.9</td>
<td>Nominal drain saturation voltage at threshold for impact ionization current</td>
</tr>
<tr>
<td>TII</td>
<td>-</td>
<td>0</td>
<td>Temperature dependence parameter for impact ionization current</td>
</tr>
<tr>
<td>LII</td>
<td>-</td>
<td>0</td>
<td>Channel length dependence parameter for impact ionization current</td>
</tr>
<tr>
<td>ESATI</td>
<td>V/m</td>
<td>1.67</td>
<td>Saturation channel electric field for impact ionization current</td>
</tr>
<tr>
<td>SII0</td>
<td>1/V</td>
<td>0.5</td>
<td>First (V_{gs}) dependence parameter for impact ionization current</td>
</tr>
<tr>
<td>SII1</td>
<td>1/V</td>
<td>0.1</td>
<td>Second (V_{gs}) dependence parameter for impact ionization current</td>
</tr>
<tr>
<td>SII2</td>
<td>1/V</td>
<td>0</td>
<td>Third (V_{gs}) dependence parameter for impact ionization current</td>
</tr>
<tr>
<td>SIID</td>
<td>1/V</td>
<td>0</td>
<td>(V_{ds}) dependence parameter of drain saturation voltage for impact ionization current</td>
</tr>
<tr>
<td>AGIDL</td>
<td>1/W</td>
<td>0.0</td>
<td>GIDL constant</td>
</tr>
<tr>
<td>BGIDL</td>
<td>V/m</td>
<td>0.0</td>
<td>GIDL exponential coefficient</td>
</tr>
<tr>
<td>NGIDL</td>
<td>V</td>
<td>1.2</td>
<td>GIDL (V_{ds}) enhancement coefficient</td>
</tr>
<tr>
<td>NTUN</td>
<td>-</td>
<td>10.0</td>
<td>Reverse tunneling non-ideality factor</td>
</tr>
<tr>
<td>NDIODE</td>
<td>-</td>
<td>1.0</td>
<td>Diode non-ideality factor</td>
</tr>
<tr>
<td>NRECF0</td>
<td>-</td>
<td>2.0</td>
<td>Recombination non-ideality factor at forward bias</td>
</tr>
<tr>
<td>NRECR0</td>
<td>-</td>
<td>10</td>
<td>Recombination non-ideality factor at reversed bias</td>
</tr>
<tr>
<td>ISBJT</td>
<td>A/m²</td>
<td>1.0e-6</td>
<td>BJT injection saturation current</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>---------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>ISDIF</td>
<td>A/m²</td>
<td>0</td>
<td>Body to source/drain injection saturation current</td>
</tr>
<tr>
<td>ISREC</td>
<td>A/m²</td>
<td>1.0e-5</td>
<td>Recombination in depletion saturation current</td>
</tr>
<tr>
<td>ISTUN</td>
<td>A/m²</td>
<td>0.0</td>
<td>Reverse tunneling saturation current</td>
</tr>
<tr>
<td>LN</td>
<td>m</td>
<td>2.0e-6</td>
<td>Electron/hole diffusion length</td>
</tr>
<tr>
<td>VREC0</td>
<td>V</td>
<td>0.0</td>
<td>Voltage dependent parameter for recombination current</td>
</tr>
<tr>
<td>VTUN0</td>
<td>V</td>
<td>0.0</td>
<td>Voltage dependent parameter for tunneling current</td>
</tr>
<tr>
<td>NBJT</td>
<td>-</td>
<td>1</td>
<td>Power coefficient of channel length dependency for bipolar current</td>
</tr>
<tr>
<td>LBJT0</td>
<td>m</td>
<td>0.2e-6</td>
<td>Reference channel length for bipolar current</td>
</tr>
<tr>
<td>VABJT</td>
<td>V</td>
<td>10</td>
<td>Early voltage for bipolar current</td>
</tr>
<tr>
<td>AELY</td>
<td>V/m</td>
<td>0</td>
<td>Channel length dependency of early voltage for bipolar current</td>
</tr>
<tr>
<td>AHLI</td>
<td>-</td>
<td>0</td>
<td>High-level injection parameter for bipolar current</td>
</tr>
<tr>
<td>RBODY</td>
<td>ohm/m²</td>
<td>0.0</td>
<td>Intrinsic body contact sheet resistance</td>
</tr>
<tr>
<td>RBSH</td>
<td>ohm/m²</td>
<td>0.0</td>
<td>Extrinsic body contact sheet resistance</td>
</tr>
<tr>
<td>RSH</td>
<td>ohm/square</td>
<td>0.0</td>
<td>Source/drain sheet resistance in ohm per square</td>
</tr>
<tr>
<td>VEBV</td>
<td>V</td>
<td>0.075v</td>
<td>Electron tunneling from the valence band</td>
</tr>
</tbody>
</table>
### AC and Capacitance Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VECB</td>
<td>v</td>
<td>0.026v</td>
<td>Electron tunneling from conduction band</td>
</tr>
<tr>
<td>XPART</td>
<td>-</td>
<td>0</td>
<td>Charge partitioning rate flag</td>
</tr>
<tr>
<td>CGSO</td>
<td>F/m</td>
<td>calculated</td>
<td>Non LDD region source-gate overlap capacitance per channel length</td>
</tr>
<tr>
<td>CGDO</td>
<td>F/m</td>
<td>0</td>
<td>Non LDD region drain-gate overlap capacitance per channel length</td>
</tr>
<tr>
<td>CGEO</td>
<td>F/m</td>
<td>0</td>
<td>Gate substrate overlap capacitance per unit channel length</td>
</tr>
<tr>
<td>CJSWG</td>
<td>F/m²</td>
<td>1.e-10</td>
<td>Source/drain (gate side) sidewall junction capacitance per unit width (normalized to 100nm Tsi)</td>
</tr>
<tr>
<td>PBSWG</td>
<td>V</td>
<td>0.7</td>
<td>Source/drain (gate side) sidewall junction capacitance built in potential</td>
</tr>
<tr>
<td>MJSWG</td>
<td>V</td>
<td>0.5</td>
<td>Source/drain (gate side) sidewall junction capacitance grading coefficient</td>
</tr>
<tr>
<td>TT</td>
<td>second</td>
<td>1ps</td>
<td>Diffusion capacitance transit time coefficient</td>
</tr>
<tr>
<td>NDIF</td>
<td>-</td>
<td>-1</td>
<td>Power coefficient of channel length dependency for diffusion capacitance</td>
</tr>
<tr>
<td>LDIF0</td>
<td>-</td>
<td>1</td>
<td>Channel length dependency coefficient of diffusion cap.</td>
</tr>
<tr>
<td>VSDFB</td>
<td>V</td>
<td>cal.</td>
<td>Source/drain bottom diffusion capacitance flatband voltage</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>---------</td>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>VSDTH</td>
<td>V</td>
<td>cal.</td>
<td>Source/drain bottom diffusion capacitance threshold voltage</td>
</tr>
<tr>
<td>CSDMIN</td>
<td>V</td>
<td>cal.</td>
<td>Source/drain bottom diffusion minimum capacitance</td>
</tr>
<tr>
<td>ASD</td>
<td>V</td>
<td>0.3</td>
<td>Source/drain bottom diffusion smoothing parameter</td>
</tr>
<tr>
<td>CSDESW</td>
<td>F/m</td>
<td>0.0</td>
<td>Source/drain sidewall fringing capacitance per unit length</td>
</tr>
<tr>
<td>CGSL</td>
<td>F/m</td>
<td>0.0</td>
<td>Lightly doped source-gate region overlap capacitance</td>
</tr>
<tr>
<td>CGDL</td>
<td>F/m</td>
<td>0.0</td>
<td>Lightly doped drain-gate region overlap capacitance</td>
</tr>
<tr>
<td>CKAPPA</td>
<td>F/m</td>
<td>0.6</td>
<td>Coefficient for lightly doped region overlap capacitance fringing field capacitance</td>
</tr>
<tr>
<td>CF</td>
<td>F/m</td>
<td>cal.</td>
<td>Gate to source/drain fringing field capacitance</td>
</tr>
<tr>
<td>CLC</td>
<td>m</td>
<td>0.1e-7</td>
<td>Constant term for the short channel model</td>
</tr>
<tr>
<td>CLE</td>
<td>-</td>
<td>0.0</td>
<td>Exponential term for the short channel model</td>
</tr>
<tr>
<td>DLC</td>
<td>m</td>
<td>lint</td>
<td>Length offset fitting parameter for gate charge</td>
</tr>
<tr>
<td>DLCB</td>
<td>m</td>
<td>lint</td>
<td>Length offset fitting parameter for body charge</td>
</tr>
<tr>
<td>DLBG</td>
<td>m</td>
<td>0</td>
<td>Length offset fitting parameter for backgate charge</td>
</tr>
<tr>
<td>DWC</td>
<td>m</td>
<td>wint</td>
<td>Width offset fitting parameter from C-V</td>
</tr>
</tbody>
</table>
### Temperature Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNOM</td>
<td>°C</td>
<td>25</td>
<td>Temperature at which parameters are expected</td>
</tr>
<tr>
<td>UTE</td>
<td>-</td>
<td>-1.5</td>
<td>Mobility temperature exponent</td>
</tr>
<tr>
<td>KT1</td>
<td>V</td>
<td>-0.11</td>
<td>Temperature coefficient for the threshold voltage</td>
</tr>
<tr>
<td>KTIL</td>
<td>V*m</td>
<td>0</td>
<td>Channel length dependence of the temperature coefficient for the threshold voltage</td>
</tr>
<tr>
<td>KT2</td>
<td>-</td>
<td>0.022</td>
<td>Body-bias coefficient of the threshold voltage temperature effect</td>
</tr>
<tr>
<td>UA1</td>
<td>m/V</td>
<td>4.31e-9</td>
<td>Temperature coefficient for $U_a$</td>
</tr>
<tr>
<td>UB1</td>
<td>(m/V)$^2$</td>
<td>-7.61e-18</td>
<td>Temperature coefficient for $U_b$</td>
</tr>
<tr>
<td>UC1</td>
<td>1/V</td>
<td>-0.056</td>
<td>Temperature coefficient for $U_c$</td>
</tr>
<tr>
<td>AT</td>
<td>m/sec</td>
<td>3.3e4</td>
<td>Temperature coefficient for $U_a$</td>
</tr>
<tr>
<td>TCJSWG</td>
<td>1/K</td>
<td>0</td>
<td>Temperature coefficient of $C_{jswg}$</td>
</tr>
<tr>
<td>TPBSWG</td>
<td>V/K</td>
<td>0</td>
<td>Temperature coefficient of $P_{bswg}$</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 47-62

LEVEL 57 UC Berkeley BSIM3-SOI Model

Notes:

- BSIMP2D supports capmod=2 and 3 only. capmod=0 and 1 are not supported.
- In modern SOI technology, source/drain extension or LDD are commonly used. As a result, the source/drain junction depth (Xj) can be different from the silicon film thickness (Tsi). By default, if Xj is not given, it is set to Tsi. Xj is not allowed to be greater than Tsi.
- BSIMP2D refers substrate to the silicon below buried oxide, not the well region in BSIM3. It is used to calculate backgate flatband voltage (Vfbb) and parameters related to source/drain diffusion bottom capacitance (Vsdth, Vsdff, Csdmin). Positive nsub means the same type of doping as the body and negative nsub means opposite type of doping.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTH0</td>
<td>m°C/(W*s)</td>
<td>0</td>
<td>Normalized thermal capacity</td>
</tr>
<tr>
<td>PRT</td>
<td>Ω·um</td>
<td>0</td>
<td>Temperature coefficient for Rdsw</td>
</tr>
<tr>
<td>RTH0</td>
<td>m°C/W</td>
<td>0</td>
<td>Normalized thermal resistance</td>
</tr>
<tr>
<td>NTRECF</td>
<td>-</td>
<td>0</td>
<td>Temperature coefficient for N_{ref}</td>
</tr>
<tr>
<td>NTRECR</td>
<td>-</td>
<td>0</td>
<td>Temperature coefficient for N_{recr}</td>
</tr>
<tr>
<td>XBJT</td>
<td>-</td>
<td>1</td>
<td>Power dependence of j_{bjt} on temperature</td>
</tr>
<tr>
<td>XDIF</td>
<td>-</td>
<td>XBJT</td>
<td>Power dependence of j_{dif} on temperature</td>
</tr>
<tr>
<td>XREC</td>
<td>-</td>
<td>1</td>
<td>Power dependence of j_{rec} on temperature</td>
</tr>
<tr>
<td>XTUN</td>
<td>-</td>
<td>0</td>
<td>Power dependence of j_{tun} on temperature</td>
</tr>
</tbody>
</table>

Parameter Unit Default Description

Δ

LEVEL 57 Template Output

Additional element templates are added to this model for output of state variables, stored charges, capacitor currents and capacitances.

SOI MOSFET (LEVEL 57)

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>LV1</td>
<td>Channel length (L)</td>
</tr>
<tr>
<td>W</td>
<td>LV2</td>
<td>Channel width (W)</td>
</tr>
<tr>
<td>AD</td>
<td>LV3</td>
<td>Area of the drain diode (AD)</td>
</tr>
<tr>
<td>AS</td>
<td>LV4</td>
<td>Area of the source diode (AS)</td>
</tr>
<tr>
<td>ICVDS</td>
<td>LV5</td>
<td>Initial condition for drain-source voltage (VDS)</td>
</tr>
<tr>
<td>ICVGS</td>
<td>LV6</td>
<td>Initial condition for gate-source voltage (VGS)</td>
</tr>
<tr>
<td>ICVES</td>
<td>LV7</td>
<td>Initial condition for Substrate-source voltage (VES)</td>
</tr>
<tr>
<td>VTH</td>
<td>LV9</td>
<td>Threshold voltage (bias dependent)</td>
</tr>
<tr>
<td>VDSAT</td>
<td>LV10</td>
<td>Saturation voltage (VDSAT)</td>
</tr>
<tr>
<td>PD</td>
<td>LV11</td>
<td>Drain diode periphery (PD)</td>
</tr>
<tr>
<td>PS</td>
<td>LV12</td>
<td>Source diode periphery (PS)</td>
</tr>
<tr>
<td>RDS</td>
<td>LV13</td>
<td>Drain resistance (squares) (RDS)</td>
</tr>
<tr>
<td>RSS</td>
<td>LV14</td>
<td>Source resistance (squares) (RSS)</td>
</tr>
<tr>
<td>GDEFF</td>
<td>LV16</td>
<td>Effective drain conductance (1/RDeff)</td>
</tr>
<tr>
<td>GSEFF</td>
<td>LV17</td>
<td>Effective source conductance (1/RS_eff)</td>
</tr>
<tr>
<td>COVLGS</td>
<td>LV36</td>
<td>Gate-source overlap capacitance</td>
</tr>
<tr>
<td>COVLGD</td>
<td>LV37</td>
<td>Gate-drain overlap capacitance</td>
</tr>
<tr>
<td>COVLGE</td>
<td>LV38</td>
<td>Gate-substrate overlap capacitance</td>
</tr>
<tr>
<td>VES</td>
<td>LX1</td>
<td>Substrate-source voltage (VES)</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 47-62
LEVEL 57 UC Berkeley BSIM3-SOI Model

Meyer and Charge Conservation Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QB</td>
<td>LX12</td>
<td>Body charge (QB)</td>
</tr>
<tr>
<td>CQB</td>
<td>LX13</td>
<td>Body charge current (CQB)</td>
</tr>
<tr>
<td>QG</td>
<td>LX14</td>
<td>Gate charge (QG)</td>
</tr>
<tr>
<td>CQG</td>
<td>LX15</td>
<td>Gate charge current (CQG)</td>
</tr>
<tr>
<td>QD</td>
<td>LX16</td>
<td>Channel charge (QD)</td>
</tr>
<tr>
<td>CQD</td>
<td>LX17</td>
<td>Channel charge current (CQD)</td>
</tr>
<tr>
<td>CGGBO</td>
<td>LX18</td>
<td>( CGGBO = \frac{\partial Q_g}{\partial V_{gb}} = CGS + CGD + CGB )</td>
</tr>
<tr>
<td>CGDBO</td>
<td>LX19</td>
<td>( CGDBO = \frac{\partial Q_g}{\partial V_{db}} ),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(for Meyer CGD=CGDBO)</td>
</tr>
<tr>
<td>CGSBO</td>
<td>LX20</td>
<td>( CGSBO = \frac{\partial Q_g}{\partial V_{sb}} ),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(for Meyer CGS=CGSBO)</td>
</tr>
<tr>
<td>Name</td>
<td>Alias</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>CBGBO</td>
<td>LX21</td>
<td>( CBGBO = \frac{\partial Q_b}{\partial V_{gb}} ) (for Meyer CGB=CBGBO)</td>
</tr>
<tr>
<td>CBDBO</td>
<td>LX22</td>
<td>( CBDBO = \frac{\partial Q_b}{\partial V_{db}} )</td>
</tr>
<tr>
<td>CBSBO</td>
<td>LX23</td>
<td>( CBSBO = \frac{\partial Q_b}{\partial V_{sb}} )</td>
</tr>
<tr>
<td>CDGBO</td>
<td>LX32</td>
<td>( CDGBO = \frac{\partial Q_d}{\partial V_{gb}} )</td>
</tr>
<tr>
<td>CDDBO</td>
<td>LX33</td>
<td>( CDDBO = \frac{\partial Q_d}{\partial V_{db}} )</td>
</tr>
<tr>
<td>CDSBO</td>
<td>LX34</td>
<td>( CDSBO = \frac{\partial Q_d}{\partial V_{sb}} )</td>
</tr>
<tr>
<td>QE</td>
<td>LX35</td>
<td>Substrate charge (QE)</td>
</tr>
<tr>
<td>CQE</td>
<td>LX36</td>
<td>Substrate charge current (CQE)</td>
</tr>
<tr>
<td>CDEBO</td>
<td>LX37</td>
<td>( CDEBO = \frac{\partial Q_d}{\partial V_{eb}} )</td>
</tr>
<tr>
<td>CBEBO</td>
<td>LX38</td>
<td>( CBEBO = \frac{\partial Q_b}{\partial V_{eb}} )</td>
</tr>
<tr>
<td>CEEBO</td>
<td>LX39</td>
<td>( CEEBO = \frac{\partial Q_e}{\partial V_{eb}} )</td>
</tr>
<tr>
<td>CEGBO</td>
<td>LX40</td>
<td>( CEGBO = \frac{\partial Q_e}{\partial V_{gb}} )</td>
</tr>
<tr>
<td>CEDBO</td>
<td>LX41</td>
<td>( CEDBO = \frac{\partial Q_e}{\partial V_{db}} )</td>
</tr>
<tr>
<td>CESBO</td>
<td>LX42</td>
<td>( CESBO = \frac{\partial Q_e}{\partial V_{sb}} )</td>
</tr>
<tr>
<td>VBS</td>
<td>LX43</td>
<td>Body-source voltage (VBS)</td>
</tr>
<tr>
<td>ICH</td>
<td>LX44</td>
<td>Channel current</td>
</tr>
<tr>
<td>IBJT</td>
<td>LX45</td>
<td>Parasitic BJT collector current</td>
</tr>
<tr>
<td>III</td>
<td>LX46</td>
<td>Impact ionization current</td>
</tr>
<tr>
<td>IGIDL</td>
<td>LX47</td>
<td>GIDL current</td>
</tr>
<tr>
<td>ITUN</td>
<td>LX48</td>
<td>Tunneling current</td>
</tr>
</tbody>
</table>
LEVEL 57 Updates to BSIM3-SOI PD versions 2.2, 2.21, and 2.22

- BSIM PD version 2.2 is updated to enhance the model flexibility and accuracy from PD version 2.0, and the following are its major features.
  - The gate-body tunneling (substrate current) is added to enhance the model accuracy.
  - The body contact resistance is improved for modeling accuracy.
  - The binning is added in this release to enhance the model flexibility.
- BSIM PD version 2.21 is updated from the PD version 2.2 for bug fixes and S/D swapping for the gate current components.
- BSIM PD version 2.22 is updated from the 2.21 version for bug fixes and enhancements. The major features are:
  - Added a new instance parameter FRBODY
  - Improved the temperature dependence of the gate direct tunneling model
  - Added two new model parameters, VEVB and VECB
  - UC Berkeley code no longer supports the model parameters NECB and NEVB. These parameters are still accepted in 2.22 for backwards compatibility, but they have no effect.

Using BSIM3-SOI PD in Star-HSPICE

You can use BSIM3-SOI PD versions 2.0, 2.2, 2.21, and 2.22 in Star-HSPICE by applying the model parameter VERSION. For example:
- PD2.0 is invoked when VERSION=2.0
- PD2.2 and PD2.21 are invoked when VERSION=2.2
- PD2.22 is invoked when VERSION=2.22

For gate-body tunneling, the model parameter IGMOD should be set to 1.
Example

This is an example of the Star-HSPICE BSIM3-SOI PD model and element statement.

```plaintext
mckt drain gate source bulk nch L=10e-6 W=10e-6
.model nch nmos LEVEL=57 igmod=1 version=2.2
+ tnom=27 tox=4.5e-09 tsi=.0000001 tbox=8e-08
+ mobmod=0 capmod=2 shmod=0 paramchk=0
+ wint=0 lint=-2e-08 vth0=.42 kl=.49
+ k2=.1 k3=0 k3b=2.2 nlx=2e-7
+ dvt0=10 dvt1=.55 dvt2=-1.4 dvt0w=0
+ dvt1w=0 dvt2w=0 nch=4.7e+17 nsub=-1e+15
+ ngate=1e+20 agidl=1E-15 bgidl=1E9 ngidl=9.1
+ ndiode=1.13 ntun=14.0 nrecf0=2.5 nrecr0=4
+ vrec0=1.2 ntrecf=.1 ntrecr=.2 isbjt=1E-4
+ isdif=1E-5 istun=2E-5 isrec=4E-2 xbjt=.9
+ xdif=.9 xrec=.9 xtn=0.01 ahli=1e-9
+ lbj0=0.2e-6 ln=2e-6 nbj=.8 ndif=1
+ aely=1e8 vabjt=0 u0=352 u=1.3e-11
+ ub=1.7e-18 uc=-4e-10 w0=1.16e-06 a=1.4
+ A1=0 A2=1 b0=.01 b1=10
+ rdsw=0 prwg=0 prwb=.2 wr=1
+ rbbody=1E0 rbsh=0.0 a0=1.4 keta=0.1
+ ketas=0.2 vsat=13500 voff=-.14 nfactor=.7
+ vdsatii0=.8 esatii=1e7 voff=-.14 ci=0
+ cdsc=-.00002 cdiscb=0 cdscd=0
+ pclm=2.9 pvag=12 pdiblc1=.18 pdiblc2=.004
+ pdiblc=-.234 droul=.2 delt=.01 etal=0.05
+ etab=0 dsub=.2 rth0=.005 clc=.0000001
+ cle=.6 cf=1e-20 ckappa=.6 cdgl=1e-20
+ cgs=1e-20 k1=-.3 k1l=0 k2=0.022
+ uie=-1.5 ual=4.31e-09 ubl=7.61e-18 ucl=5.6e-11
+ prt=760 at=22400 cgso=1e-10 cgdo=1e-10
+ cjswg=1e-12 tt=3e-10 asd=0.3 csdesw=1e-12
+ tcjswg=1e-4 mjswg=.5 pbswg=1
```
LEVEL 58 University of Florida SOI Model

UFSOI has non-fully depleted (NFD) and fully depleted (FD) SOI models (no dynamic mode operating between NFD and FD allowed) that separately describe two main types of SOI devices. The UFSOI version 4.5F model has been installed in Star-Hspice as LEVEL 58. This model is described in the “UFSOI Model User’s Manual,” which can be found at “http://www.soi.tec.ufl.edu/”.

In some processes, there is an external contact to the body of the device. In the 1999.2 release, Star-Hspice only supports a 4-terminal device, which includes drain, front gate, source and back gate (or substrate). The additional body contact is currently not supported and is floated.

The effects of parasitic diodes in SOI are different from those in bulk MOSFET. The Star-Hspice junction model (ACM), developed for bulk MOSFETs, is not included in the SOI model.

The general syntax for including LEVEL 58 in a Star-Hspice netlist follows:

**General Form**

Mxxx nd ngf ns <ngb> mname <L=val> <W=val> <M=val>
+ <AD=val> <AS=val> <PD=val> <PS=val> <NRD=val>
+ <NRS=val> <NRB=val> <RTH=val> <CTH=val> <off>
+ <IC=Vds,Vgfs,VGbs>

where the angle brackets indicate optional parameters. The arguments are identical to those used for BSIM3-SOI model, except the thermal resistance and capacitance have different names:

- **RTH** Thermal resistance, unit in K·W⁻¹, defaulted to 0.0
- **CTH** Thermal capacitance, unit in W·s·K⁻¹, defaulted to 0.0

**Notes:**

- The defaulted value for channel length L and width W is 1.0e-6.
- The present version supports only 4 nodes, which means only floating-body devices are supported. AB is typically zero and should be specified accordingly.
When the self-heating option is activated (on the model line), RTH and CTH, typical values of which are 5e3 and 1e-12, respectively, but which can vary widely from one device to another, are used to define the thermal impedance of the device.

For \( M > 1 \), W, AD, AS, NRD, NRS, NRB, PDJ, PSJ, RTH, and CTH must be specified per gate finger.

The initial condition IC is in the order: drain voltage Vds, front gate voltage Vgfs, and back gate voltage Vbgs.

### LEVEL 58 FD/SOI MOSFET Model Parameters

The following tables describe the LEVEL 58 model parameters for fully depleted (FD) SOI including parameter name, descriptions, units, default and typical notes.

#### Flag Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Typical Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>LEVEL 57 for UFSOI</td>
</tr>
<tr>
<td>NFDMD</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>Model selector (0: FD)</td>
</tr>
<tr>
<td>BJT</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>Parasitic bipolar flag (0: off; 1: on)</td>
</tr>
<tr>
<td>SELFT</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>Self-heating flag (0: no self-heating; 1: approximate model; 2: full self-heating)</td>
</tr>
<tr>
<td>TPG</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>Type of gate polysilicon (+1: opposite to body; -1: same as body)</td>
</tr>
<tr>
<td>TPS</td>
<td>-</td>
<td>-1</td>
<td>-</td>
<td>Type of substrate (+1: opposite to body; -1: same as body)</td>
</tr>
</tbody>
</table>
### Structural Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Typical Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOXF</td>
<td>m</td>
<td>1.0e-8</td>
<td>(3-8)x10^-9</td>
<td>Front-gate oxide thickness</td>
</tr>
<tr>
<td>TOXB</td>
<td>m</td>
<td>0.5e-6</td>
<td>(80-400) x10^-9</td>
<td>Back-gate oxide thickness</td>
</tr>
<tr>
<td>NSUB</td>
<td>cm^-3</td>
<td>1.0e15</td>
<td>10^15-10^17</td>
<td>Substrate doping density</td>
</tr>
<tr>
<td>NGATE</td>
<td>cm^-3</td>
<td>0.0</td>
<td>10^19-10^20</td>
<td>Poly-gate doping density (0 for no poly-gate depletion)</td>
</tr>
<tr>
<td>NDS</td>
<td>cm^-3</td>
<td>5.0e19</td>
<td>10^19-10^20</td>
<td>Source/drain doping density</td>
</tr>
<tr>
<td>TB</td>
<td>m</td>
<td>0.1e-6</td>
<td>(30-100) x10^-9</td>
<td>Film (body) thickness</td>
</tr>
<tr>
<td>NBODY</td>
<td>cm^-3</td>
<td>5.0e16</td>
<td>10^17-10^18</td>
<td>Film (body) doping density</td>
</tr>
<tr>
<td>LLDD</td>
<td>m</td>
<td>0.0</td>
<td>(0.05-0.2) x10^-6</td>
<td>LDD/LDS region length (0 for no LDD)</td>
</tr>
<tr>
<td>NLDD</td>
<td>cm^-3</td>
<td>5.0e19</td>
<td>1 x10^19</td>
<td>LDD/LDS doping density (&gt;1e19: LDD/LDS treated as D/S extensions)</td>
</tr>
<tr>
<td>DL</td>
<td>m</td>
<td>0.0</td>
<td>(0.05-0.15) x10^-6</td>
<td>Channel-length reduction</td>
</tr>
<tr>
<td>DW</td>
<td>m</td>
<td>0.0</td>
<td>(0.1-0.5) x10^-6</td>
<td>Channel-width reduction</td>
</tr>
</tbody>
</table>
## Electrical Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Typical Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NQFF</td>
<td>cm⁻²</td>
<td>0.0</td>
<td>~ $10^{10}$</td>
<td>Front oxide fixed charge (normalized)</td>
</tr>
<tr>
<td>NQFB</td>
<td>cm⁻²</td>
<td>0.0</td>
<td>~ $10^{11}$</td>
<td>Back oxide fixed charge (normalized)</td>
</tr>
<tr>
<td>NQFSW</td>
<td>cm⁻²</td>
<td>0.0</td>
<td>~ ±$10^{12}$</td>
<td>Effective sidewall fixed charge (0 for no narrow-width effect)</td>
</tr>
<tr>
<td>NSF</td>
<td>cm⁻²⋅eV⁻¹</td>
<td>0.0</td>
<td>~$10^{10}$</td>
<td>Front surface state density</td>
</tr>
<tr>
<td>NSB</td>
<td>cm⁻²⋅eV⁻¹</td>
<td>0.0</td>
<td>~ $10^{11}$</td>
<td>Back surface state density</td>
</tr>
<tr>
<td>QM</td>
<td></td>
<td>0.0</td>
<td>-0.5</td>
<td>Energy quantization parameter (0 for no quantization)</td>
</tr>
<tr>
<td>UO</td>
<td>cm²⋅V⁻¹⋅s⁻¹</td>
<td>7.0e2</td>
<td>200-700 (nMOS) 70-400 (pMOS)</td>
<td>Low-field mobility</td>
</tr>
<tr>
<td>THETA</td>
<td>cm⋅V⁻¹</td>
<td>7.0e2</td>
<td>(0.1-3) x10⁻⁶</td>
<td>Mobility degradation coefficient</td>
</tr>
<tr>
<td>VSAT</td>
<td>cm⋅s⁻¹</td>
<td>1.0e-6</td>
<td>(0.5-1)x10</td>
<td>Carrier saturated drift velocity</td>
</tr>
<tr>
<td>ALPHA</td>
<td>cm⁻¹</td>
<td>0.0</td>
<td>2.45x10⁶</td>
<td>Impact-ionization coefficient (0 for no impact ionization)</td>
</tr>
<tr>
<td>BETA</td>
<td>V⋅cm⁻¹</td>
<td>0.0</td>
<td>1.92x10⁶</td>
<td>Impact-ionization exponential factor (0 for no impact ionization)</td>
</tr>
<tr>
<td>BGIDL</td>
<td>V⋅cm⁻¹</td>
<td>0.0</td>
<td>(4-8)x10⁹</td>
<td>Exponential factor for gate-induced drain leakage (0 for no GIDL)</td>
</tr>
<tr>
<td>GAMMA</td>
<td></td>
<td>0.3</td>
<td>0.3-1.0</td>
<td>BOX fringing field weighting factor</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Typical Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>KAPPA</td>
<td>-</td>
<td>0.5</td>
<td>0.5-1.0</td>
<td>BOX fringing field weighting factor</td>
</tr>
<tr>
<td>JRO</td>
<td>A·m⁻¹</td>
<td>1.0e-10</td>
<td>10⁻¹¹⁻¹⁻⁹</td>
<td>Body-source/drain junction recombination current coefficient</td>
</tr>
<tr>
<td>M</td>
<td>-</td>
<td>2.0</td>
<td>1.0-2.0</td>
<td>Body-source/drain junction recombination ideality factor</td>
</tr>
<tr>
<td>LDIFF</td>
<td>m</td>
<td>1.0e-7</td>
<td>(0.1-0.5) x10⁻⁶</td>
<td>Effective diffusion length in source/drain</td>
</tr>
<tr>
<td>SEFF</td>
<td>cm·s⁻¹</td>
<td>1.0e5</td>
<td>(0.5-5) x10⁵</td>
<td>Effective recombination velocity in source/drain</td>
</tr>
<tr>
<td>CGFDO</td>
<td>F·m⁻¹</td>
<td>0.0</td>
<td>1×10⁻¹⁰</td>
<td>Gate-drain overlap capacitance</td>
</tr>
<tr>
<td>CGFSO</td>
<td>F·m⁻¹</td>
<td>0.0</td>
<td>1×10⁻¹⁰</td>
<td>Gate-source overlap capacitance</td>
</tr>
<tr>
<td>CGFBO</td>
<td>F·m⁻¹</td>
<td>0.0</td>
<td>0.0</td>
<td>Gate-body overlap capacitance</td>
</tr>
<tr>
<td>RD</td>
<td>ohm·m</td>
<td>0.0</td>
<td>200-1000</td>
<td>Specific drain parasitic resistance</td>
</tr>
<tr>
<td>RS</td>
<td>ohm·m</td>
<td>0.0</td>
<td>200-1000</td>
<td>Specific source parasitic resistance</td>
</tr>
<tr>
<td>RHOB</td>
<td>ohm/sq.</td>
<td>0.0</td>
<td>30×10³</td>
<td>Body sheet resistance</td>
</tr>
<tr>
<td>FNK</td>
<td>F·A</td>
<td>0.0</td>
<td>0-10⁻²</td>
<td>Flicker noise coefficient</td>
</tr>
<tr>
<td>FNA</td>
<td>-</td>
<td>1.0</td>
<td>0.5-2</td>
<td>Flicker noise exponent</td>
</tr>
</tbody>
</table>
Optional Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Typical Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFBF</td>
<td>V</td>
<td>calc.</td>
<td>-1 (nMOS) 1 (pMOS)</td>
<td>Front-gate flatband voltage</td>
</tr>
<tr>
<td>VFBB</td>
<td>V</td>
<td>calc.</td>
<td>-</td>
<td>Back-gate flatband voltage</td>
</tr>
<tr>
<td>WKF</td>
<td>V</td>
<td>calc.</td>
<td>~ VFBF</td>
<td>Front-gate work function difference</td>
</tr>
<tr>
<td>WKB</td>
<td>V</td>
<td>calc.</td>
<td>-</td>
<td>Back-gate work function difference</td>
</tr>
<tr>
<td>TAUO</td>
<td>s</td>
<td>calc.</td>
<td>$10^{-7}$-$10^{-5}$</td>
<td>Carrier lifetime in lightly doped regions</td>
</tr>
<tr>
<td>BFACT</td>
<td>-</td>
<td>0.3</td>
<td>0.1-0.5</td>
<td>$V_{DS}$-averaging factor for mobility degradation</td>
</tr>
<tr>
<td>FVBJT</td>
<td>-</td>
<td>0.0</td>
<td>0-1</td>
<td>BJT current directional partitioning factor (0 for lateral 1D flow)</td>
</tr>
<tr>
<td>RHOSD</td>
<td>ohm/ sq.</td>
<td>0.0</td>
<td>50</td>
<td>Source/drain sheet resistance</td>
</tr>
</tbody>
</table>

LEVEL 58 NFD/SOI MOSFET Model Parameters

The following tables describe the LEVEL 58 model parameters for non-fully depleted (NFD) SOI including parameter name, descriptions, units, default and typical notes.
### Flag Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Typical Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>LEVEL 57 for UFSOI</td>
</tr>
<tr>
<td>NFDMOD</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>Model selector (1: NFD)</td>
</tr>
<tr>
<td>BJT</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>Parasitic bipolar flag (0: off; 1: on)</td>
</tr>
<tr>
<td>SELFET</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>Self-heating flag (0: no self-heating; 1: approximate model; 2: full self-heating)</td>
</tr>
<tr>
<td>TPG</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>Type of gate polysilicon (+1: opposite to body; -1: same as body)</td>
</tr>
<tr>
<td>TPS</td>
<td>-</td>
<td>-1</td>
<td>-1</td>
<td>Type of substrate (+1: opposite to body; -1: same as body)</td>
</tr>
</tbody>
</table>

### Structural Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Typical Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOXF</td>
<td>m</td>
<td>1.0e-8</td>
<td>(3-8)x10^-9</td>
<td>Front-gate oxide thickness</td>
</tr>
<tr>
<td>TOXB</td>
<td>m</td>
<td>0.5e-6</td>
<td>(80-400)x10^-9</td>
<td>Back-gate oxide thickness</td>
</tr>
<tr>
<td>NSUB</td>
<td>cm^-3</td>
<td>1.0e15</td>
<td>10^{15-17}</td>
<td>Substrate doping density</td>
</tr>
<tr>
<td>NGATE</td>
<td>cm^-3</td>
<td>0.0</td>
<td>10^{19-20}</td>
<td>Poly-gate doping density (0 for no poly-gate depletion)</td>
</tr>
<tr>
<td>NDS</td>
<td>cm^-3</td>
<td>5.0e19</td>
<td>10^{19-20}</td>
<td>Source/drain doping density</td>
</tr>
<tr>
<td>TF</td>
<td>m</td>
<td>0.2e-6</td>
<td>(3-8)x10^-9</td>
<td>Silicon film thickness</td>
</tr>
<tr>
<td>TB</td>
<td>m</td>
<td>0.1e-6</td>
<td>(30-100)x10^-9</td>
<td>Film (body) thickness</td>
</tr>
</tbody>
</table>
### Electrical Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Typical Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NQFF</td>
<td>cm(^{-2})</td>
<td>0.0</td>
<td>~(10^{10})</td>
<td>Front oxide fixed charge (normalized)</td>
</tr>
<tr>
<td>NQFB</td>
<td>cm(^{-2})</td>
<td>0.0</td>
<td>~(10^{11})</td>
<td>Back oxide fixed charge (normalized)</td>
</tr>
<tr>
<td>NQFSW</td>
<td>cm(^{-2})</td>
<td>0.0</td>
<td>~(±10^{12})</td>
<td>Effective sidewall fixed charge (0 for no narrow-width effect)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Typical Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>QM</td>
<td>-</td>
<td>0.0 - 0.5</td>
<td>Energy quantization parameter (0 for no quantization)</td>
<td></td>
</tr>
<tr>
<td>UO</td>
<td>cm²·V⁻¹·s⁻¹</td>
<td>7.0e2</td>
<td>200-700 (nMOS) 70-400 (pMOS)</td>
<td>Low-field mobility</td>
</tr>
<tr>
<td>THETA</td>
<td>cm·V⁻¹</td>
<td>7.0e2</td>
<td>(0.1-3) x10⁻⁶</td>
<td>Mobility degradation coefficient</td>
</tr>
<tr>
<td>VSAT</td>
<td>cm·s⁻¹</td>
<td>1.0e-6</td>
<td>(0.5-1) x10</td>
<td>Carrier saturated drift velocity</td>
</tr>
<tr>
<td>ALPHA</td>
<td>cm⁻¹</td>
<td>0.0</td>
<td>2.45x10⁶</td>
<td>Impact-ionization coefficient (0 for no impact ionization)</td>
</tr>
<tr>
<td>BETA</td>
<td>V·cm⁻¹</td>
<td>0.0</td>
<td>1.92x10⁶</td>
<td>Impact-ionization exponential factor (0 for no impact ionization)</td>
</tr>
<tr>
<td>BGIDL</td>
<td>V·cm⁻¹</td>
<td>0.0</td>
<td>(4-8)x10⁹</td>
<td>Exponential factor for gate-induced drain leakage (0 for no GIDL)</td>
</tr>
<tr>
<td>NTR</td>
<td>cm⁻³</td>
<td>0.0</td>
<td>10¹⁴-10¹⁵</td>
<td>Effective trap density for trap-assisted junction tunneling (0 for no tunneling)</td>
</tr>
<tr>
<td>JRO</td>
<td>A·m⁻¹</td>
<td>1.0e-10</td>
<td>10⁻¹¹-10⁻⁹</td>
<td>Body-source/drain junction recombination current coefficient</td>
</tr>
<tr>
<td>M</td>
<td>-</td>
<td>2.0</td>
<td>1.0-2.0</td>
<td>Body-source/drain junction recombination ideality factor</td>
</tr>
<tr>
<td>LDIFF</td>
<td>m</td>
<td>1.0e-7</td>
<td>(0.1-0.5) x10⁻⁶</td>
<td>Effective diffusion length in source/drain</td>
</tr>
<tr>
<td>SEFF</td>
<td>cm·s⁻¹</td>
<td>1.0e5</td>
<td>(0.5-5) x10⁵</td>
<td>Effective recombination velocity in source/drain</td>
</tr>
</tbody>
</table>
Optional Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Typical Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFBF</td>
<td>V calc.</td>
<td>-1 (nMOS)</td>
<td>1 (pMOS)</td>
<td>Front-gate flatband voltage</td>
</tr>
<tr>
<td>VFBB</td>
<td>V calc.</td>
<td>-</td>
<td></td>
<td>Back-gate flatband voltage</td>
</tr>
<tr>
<td>WKF</td>
<td>V calc.</td>
<td>~ VFBF</td>
<td></td>
<td>Front-gate work function difference</td>
</tr>
<tr>
<td>WKB</td>
<td>V calc.</td>
<td>-</td>
<td></td>
<td>Back-gate work function difference</td>
</tr>
<tr>
<td>TAUO</td>
<td>s calc.</td>
<td>10^{-7}-10^{-5}</td>
<td></td>
<td>Carrier lifetime in lightly doped regions</td>
</tr>
<tr>
<td>BFACT</td>
<td>-</td>
<td>0.3</td>
<td>0.1-0.5</td>
<td>V_{DS}-averaging factor for mobility degradation</td>
</tr>
<tr>
<td>FVBQT</td>
<td>-</td>
<td>0.0</td>
<td>0-1</td>
<td>BJT current directional partitioning factor (0 for lateral 1D flow)</td>
</tr>
</tbody>
</table>
Notes:

- The model line must include LEVEL=58 and NFDMOD=0 for FD or NFDMOD=1 for NFD devices.
- Specifying VFBF turns off the narrow-width effect defined by NQFSW (which can be positive or negative) and the reverse short-channel effect defined by LRSCE (and NBH, or NHALO if specified); the latter effect is also turned off when WKF is specified.
- For floating-body devices, CGFBO is small and should be specified to be 0.
- JRO and SEFF influence the gain of the BJT, but LDIFF affects only bipolar charge storage in the source/drain. The BJT gain is influenced by NBH and NHALO (if THALO is specified) as well.
- The value of TAUO should be loosely correlated with JRO in accord with basic pn-junction recombination/generation properties. Its default value is calculated based on JRO, which is appropriate for short L; for long L, body generation tends to predominate over that in the junctions, and hence TAUO should be specified.
- The (non-local) impact-ionization model is physical, and its parameters should not be varied arbitrarily.
- The LDD option intensifies the model, so it is advisable to set LLDD to 0 for large-scale circuit simulation, and add the unbiased LDD resistance to RD; this simplification is foisted when NLDD > 1e19 is specified.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Typical Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHOSD</td>
<td>ohm/sq.</td>
<td>0.0</td>
<td>50</td>
<td>Source/drain sheet resistance</td>
</tr>
</tbody>
</table>
LEVEL 58 Template Output

Some element templates are added to this model for output of state variables, stored charges and capacitor currents.

SOI MOSFET (LEVEL 58)

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>LV1</td>
<td>Channel length (L)</td>
</tr>
<tr>
<td>W</td>
<td>LV2</td>
<td>Channel width (W)</td>
</tr>
<tr>
<td>AD</td>
<td>LV3</td>
<td>Area of the drain diode (AD)</td>
</tr>
<tr>
<td>AS</td>
<td>LV4</td>
<td>Area of the source diode (AS)</td>
</tr>
<tr>
<td>ICVDS</td>
<td>LV5</td>
<td>Initial condition for drain-source voltage (VDS)</td>
</tr>
<tr>
<td>ICVGS</td>
<td>LV6</td>
<td>Initial condition for gate-source voltage (VGS)</td>
</tr>
<tr>
<td>ICVES</td>
<td>LV7</td>
<td>Initial condition for Substrate-source voltage (VES)</td>
</tr>
<tr>
<td>VTH</td>
<td>LV9</td>
<td>Threshold voltage (bias dependent)</td>
</tr>
<tr>
<td>VDSAT</td>
<td>LV10</td>
<td>Saturation voltage (VDSAT)</td>
</tr>
<tr>
<td>PD</td>
<td>LV11</td>
<td>Drain diode periphery (PD)</td>
</tr>
<tr>
<td>PS</td>
<td>LV12</td>
<td>Source diode periphery (PS)</td>
</tr>
<tr>
<td>RDS</td>
<td>LV13</td>
<td>Drain resistance (squares) (RDS)</td>
</tr>
<tr>
<td>RSS</td>
<td>LV14</td>
<td>Source resistance (squares) (RSS)</td>
</tr>
<tr>
<td>GDEFF</td>
<td>LV16</td>
<td>Effective drain conductance (1/RDeff)</td>
</tr>
<tr>
<td>GSEFF</td>
<td>LV17</td>
<td>Effective source conductance (1/RSeff)</td>
</tr>
<tr>
<td>VES</td>
<td>LX1</td>
<td>Substrate-source voltage (VES)</td>
</tr>
<tr>
<td>VGS</td>
<td>LX2</td>
<td>Gate-source voltage (VGS)</td>
</tr>
<tr>
<td>VDS</td>
<td>LX3</td>
<td>Drain-source voltage (VDS)</td>
</tr>
<tr>
<td>CDO</td>
<td>LX4</td>
<td>DC drain current (CDO)</td>
</tr>
</tbody>
</table>
### Star-Hspice Manual, Release 2001.2

#### Selecting MOSFET Models: LEVEL 47-62

**LEVEL 58 University of Florida SOI Model**

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMO</td>
<td>LX7</td>
<td>DC gate transconductance (GMO)</td>
</tr>
<tr>
<td>GDSO</td>
<td>LX8</td>
<td>DC drain-source conductance (GDSO)</td>
</tr>
<tr>
<td>GMESO</td>
<td>LX9</td>
<td>DC substrate transconductance (GMESO)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QB</td>
<td>LX12</td>
<td>Body charge (QB)</td>
</tr>
<tr>
<td>CQB</td>
<td>LX13</td>
<td>Body charge current (CQB)</td>
</tr>
<tr>
<td>QG</td>
<td>LX14</td>
<td>Gate charge (QG)</td>
</tr>
<tr>
<td>CQG</td>
<td>LX15</td>
<td>Gate charge current (CQG)</td>
</tr>
<tr>
<td>QD</td>
<td>LX16</td>
<td>Channel charge (QD)</td>
</tr>
<tr>
<td>CQD</td>
<td>LX17</td>
<td>Channel charge current (CQD)</td>
</tr>
<tr>
<td>CGGBO</td>
<td>LX18</td>
<td>(CGGBO = \frac{\partial Q_g}{\partial V_{gb}} = CGS + CGD + CGB)</td>
</tr>
<tr>
<td>CGDBO</td>
<td>LX19</td>
<td>(CGDBO = \frac{\partial Q_g}{\partial V_{db}}, \text{ (for Meyer CGD=CGDBO)})</td>
</tr>
<tr>
<td>CGSBO</td>
<td>LX20</td>
<td>(CGSBO = \frac{\partial Q_g}{\partial V_{sb}}, \text{ (for Meyer CGS=CGSBO)})</td>
</tr>
<tr>
<td>CBGBO</td>
<td>LX21</td>
<td>(CBGBO = \frac{\partial Q_b}{\partial V_{gb}}, \text{ (for Meyer CGB=CBGBO)})</td>
</tr>
<tr>
<td>CBDBO</td>
<td>LX22</td>
<td>(CBDBO = \frac{\partial Q_b}{\partial V_{db}})</td>
</tr>
<tr>
<td>CBSBO</td>
<td>LX23</td>
<td>(CBSBO = \frac{\partial Q_b}{\partial V_{sb}})</td>
</tr>
<tr>
<td>CDGBO</td>
<td>LX32</td>
<td>(CDGBO = \frac{\partial Q_d}{\partial V_{gb}})</td>
</tr>
<tr>
<td>CDDBO</td>
<td>LX33</td>
<td>(CDDBO = \frac{\partial Q_d}{\partial V_{db}})</td>
</tr>
<tr>
<td>CDSBO</td>
<td>LX34</td>
<td>(CDSBO = \frac{\partial Q_d}{\partial V_{sb}})</td>
</tr>
<tr>
<td>QE</td>
<td>LX35</td>
<td>Substrate charge (QE)</td>
</tr>
<tr>
<td>CQE</td>
<td>LX36</td>
<td>Substrate charge current (CQE)</td>
</tr>
<tr>
<td>Name</td>
<td>Alias</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>VBS</td>
<td>LX43</td>
<td>Body-source voltage (VBS)</td>
</tr>
<tr>
<td>ICH</td>
<td>LX44</td>
<td>Channel current</td>
</tr>
<tr>
<td>IBJT</td>
<td>LX45</td>
<td>Parasitic BJT collector current</td>
</tr>
<tr>
<td>III</td>
<td>LX46</td>
<td>Impact ionization current</td>
</tr>
<tr>
<td>IGIDL</td>
<td>LX47</td>
<td>GIDL current</td>
</tr>
<tr>
<td>ITUN</td>
<td>LX48</td>
<td>Tunneling current</td>
</tr>
</tbody>
</table>
LEVEL 59 UC Berkeley BSIM3-SOI FD Model

The UC Berkeley SOI (BSIM3-SOI) Fully Depleted (FD) model is now installed in Star-Hspice as LEVEL 59. This model is described in the “BSIM3SOI FD2.1 MOSFET MODEL User Manual,” which can be found at “http://www-device.eecs.berkeley.edu/~bsim3soi”.

The general syntax for including a BSIM3-SOI FD MOSFET element in a Star-Hspice netlist is:

**General Form**

```
Mxxx nd ng ns ne <np> mname <L=val>
+ <W=val> <M=val> <AD=val> <AS=val> <PD=val> <PS=val>
+ <NRD=val> <NRS=val> <NRB=val> <RTH0=val> <CTH0=val>
+ <off> <BJToff=val> <IC=Vds, Vgs, Vbs, Ves, Vps>
```

where the angle brackets indicate optional parameters.

The arguments are as follows:

- **Mxxx**: SOI MOSFET element name. Must begin with M, which can be followed by up to 1023 alphanumeric characters.
- **nd**: Drain terminal node name or number
- **ng**: Front gate node name or number
- **ns**: Source terminal node name or number
- **ne**: Back gate (or substrate) node name or number
- **np**: Optional external body contact node name or number
- **mname**: MOSFET model name reference
- **L**: SOI MOSFET channel length in meters. This parameter overrides DEFL in an OPTIONS statement. Default=DEFL with a maximum of 0.1m.
- **W**: MOSFET channel width in meters. This parameter overrides DEFW in an OPTIONS statement. Default=DEFW.
M Multiplier to simulate multiple SOI MOSFETs in parallel. All channel widths, diode leakages, capacitances and resistances are affected by the setting of M. Default=1.

AD Drain diffusion area. Overrides DEFAD in the OPTIONS statement. Default=DEFAD.

AS Source diffusion area. Overrides DEFAS in the OPTIONS statement. Default=DEFAS.

PD Perimeter of the drain junction, including the channel edge. Overrides DEFPD in the OPTIONS statement.

PS Perimeter of the source junction, including the channel edge. Overrides DEFPS in the OPTIONS statement.

NRD Number of squares of drain diffusion for drain series resistance. Overrides DEFNRD in the OPTIONS statement.

NRS Number of squares of source diffusion for source series resistance. Overrides DEFNRS in the OPTIONS statement.

NRB Number of squares for body series resistance.

RTH0 Thermal resistance per unit width
- If not specified, RTH0 is extracted from the model card.
- If specified, it overrides the one in the model card.

CTH0 Thermal capacitance per unit width
- If not specified, CTH0 is extracted from model card.
- If specified, it overrides the one in the model card.

OFF Sets initial condition to OFF for this element in DC analysis

BJTOFF Turning off BJT if equal to 1

IC Initial guess in the order (drain, front gate, internal body, back gate, external voltage). (Vps is ignored in the case of 4-terminal device) These are used when UIC is present in the .TRAN statement and are overridden by the .IC statement.
# LEVEL 59 Model Parameters

## Model Control Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPMOD</td>
<td>-</td>
<td>2</td>
<td>Flag for the short channel capacitance model</td>
</tr>
<tr>
<td>LEVEL</td>
<td>-</td>
<td>-</td>
<td>LEVEL 59 for BSIM3SOI</td>
</tr>
<tr>
<td>MOBMOD</td>
<td>-</td>
<td>1</td>
<td>Mobility model selector</td>
</tr>
<tr>
<td>NOIMOD</td>
<td>-</td>
<td>1</td>
<td>Flag for Noise model</td>
</tr>
<tr>
<td>SHMOD</td>
<td>-</td>
<td>0</td>
<td>Flag for self-heating:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 - no self-heating</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - self-heating</td>
</tr>
</tbody>
</table>

## Process Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCH</td>
<td>1/cm³</td>
<td>1.7e17</td>
<td>Channel doping concentration</td>
</tr>
<tr>
<td>NGATE</td>
<td>1/cm³</td>
<td>0</td>
<td>Poly gate doping concentration</td>
</tr>
<tr>
<td>NSUB</td>
<td>1/cm³</td>
<td>6.0e16</td>
<td>Substrate doping concentration</td>
</tr>
<tr>
<td>TBOX</td>
<td>m</td>
<td>3.0e-7</td>
<td>Buried oxide thickness</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>1.0e-8</td>
<td>Gate oxide thickness</td>
</tr>
<tr>
<td>TSI</td>
<td>m</td>
<td>1.0e-17</td>
<td>Silicon film thickness</td>
</tr>
</tbody>
</table>
### DC Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>-</td>
<td>1.0</td>
<td>Bulk charge effect coefficient for channel length</td>
</tr>
<tr>
<td>A1</td>
<td>1/V</td>
<td>0.0</td>
<td>First non-saturation effect parameter</td>
</tr>
<tr>
<td>A2</td>
<td>-</td>
<td>1.0</td>
<td>Second non-saturation effect parameter</td>
</tr>
<tr>
<td>ABP</td>
<td>-</td>
<td>1.0</td>
<td>Coefficient of $A_{beff}$ dependency on $V_{gst}$</td>
</tr>
<tr>
<td>ADICE0</td>
<td>-</td>
<td>1</td>
<td>DICE bulk charge factor</td>
</tr>
<tr>
<td>AGIDL</td>
<td>1/W</td>
<td>0.0</td>
<td>GIDL constant</td>
</tr>
<tr>
<td>AGS</td>
<td>1/V</td>
<td>0.0</td>
<td>Gate bias coefficient of $A_{bulk}$</td>
</tr>
<tr>
<td>AII</td>
<td>1/V</td>
<td>0.0</td>
<td>First $L_{eff}$ dependence $V_{dsat_{ii}}$ parameter</td>
</tr>
<tr>
<td>ALPHA0</td>
<td>m/V</td>
<td>0.0</td>
<td>The first parameter of impact ionization current</td>
</tr>
<tr>
<td>ALPHA1</td>
<td>1/V</td>
<td>1.0</td>
<td>The second parameter of impact ionization current</td>
</tr>
<tr>
<td>B0</td>
<td>m</td>
<td>0.0</td>
<td>Bulk charge effect coefficient for channel width</td>
</tr>
<tr>
<td>B1</td>
<td>m</td>
<td>0.0</td>
<td>Bulk charge effect width offset</td>
</tr>
<tr>
<td>BGIDL</td>
<td>V/m</td>
<td>0.0</td>
<td>GIDL exponential coefficient</td>
</tr>
<tr>
<td>BII</td>
<td>m/V</td>
<td>0.0</td>
<td>Second $L_{eff}$ dependence $V_{dsat_{ii}}$ parameter</td>
</tr>
<tr>
<td>CDSC</td>
<td>F/m²</td>
<td>2.4e-4</td>
<td>Drain/source to channel coupling capacitance</td>
</tr>
<tr>
<td>CDSCB</td>
<td>F/m²</td>
<td>0</td>
<td>Body-bias sensitivity of cdsc</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>---------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>CDSCD</td>
<td>F/m²</td>
<td>0</td>
<td>Drain-bias sensitivity of cdsc</td>
</tr>
<tr>
<td>CII</td>
<td>-</td>
<td>0.0</td>
<td>First $V_{ds}$ dependence $V_{dssatii}$ parameter</td>
</tr>
<tr>
<td>CIT</td>
<td>F/m²</td>
<td>0.0</td>
<td>Interface trap capacitance</td>
</tr>
<tr>
<td>DELP</td>
<td>V</td>
<td>0.02</td>
<td>Constant for limiting $V_{bseff}$ to surface potential</td>
</tr>
<tr>
<td>DELTA</td>
<td>-</td>
<td>0.01</td>
<td>Effective $V_{ds}$ parameter</td>
</tr>
<tr>
<td>DII</td>
<td>V</td>
<td>-1.0</td>
<td>Second $V_{ds}$ dependence $V_{dssatii}$ parameter</td>
</tr>
<tr>
<td>DROUT</td>
<td>-</td>
<td>0.56</td>
<td>$L$ dependence coefficient of the DIBL correction parameter in $R_{out}$</td>
</tr>
<tr>
<td>DSUB</td>
<td>-</td>
<td>0.56</td>
<td>DIBL coefficient exponent</td>
</tr>
<tr>
<td>DVBD0</td>
<td>V</td>
<td>0</td>
<td>First coefficient of $V_{bs0}$ dependency on $L_{eff}$</td>
</tr>
<tr>
<td>DVBD1</td>
<td>V</td>
<td>0</td>
<td>Second coefficient of $V_{bs0}$ dependency on $L_{eff}$</td>
</tr>
<tr>
<td>DVT0</td>
<td>-</td>
<td>2.2</td>
<td>First coefficient of short-channel effect on $V_{th}$</td>
</tr>
<tr>
<td>DVT0W</td>
<td>-</td>
<td>0</td>
<td>First coefficient of narrow width effect on $V_{th}$ for small channel length</td>
</tr>
<tr>
<td>DVT1</td>
<td>-</td>
<td>0.53</td>
<td>Second coefficient of short-channel effect on $V_{th}$</td>
</tr>
<tr>
<td>DVT1W</td>
<td>-</td>
<td>5.3e6</td>
<td>Second coefficient of narrow width effect on $V_{th}$ for small channel length</td>
</tr>
<tr>
<td>DVT2</td>
<td>1/V</td>
<td>-0.032</td>
<td>Body-bias coefficient of short-channel effect on $V_{th}$</td>
</tr>
<tr>
<td>DVT2W</td>
<td>1/V</td>
<td>-0.032</td>
<td>Body-bias coefficient of narrow width effect on $V_{th}$ for small channel length</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>DWB</td>
<td>m/V$^{1/2}$</td>
<td>0.0</td>
<td>Coefficient of Weff’s substrate body bias dependence</td>
</tr>
<tr>
<td>DWG</td>
<td>m/V</td>
<td>0.0</td>
<td>Coefficient of Weff’s gate dependence</td>
</tr>
<tr>
<td>EDL</td>
<td>m</td>
<td>2e-6</td>
<td>Electron diffusion length</td>
</tr>
<tr>
<td>ETA0</td>
<td>-</td>
<td>0.08</td>
<td>DIBL coefficient in the subthreshold region</td>
</tr>
<tr>
<td>ETAB</td>
<td>1/V</td>
<td>-0.07</td>
<td>Body-bias coefficient for the subthreshold DIBL effect</td>
</tr>
<tr>
<td>ISBJT</td>
<td>A/m$^2$</td>
<td>1.0e-6</td>
<td>BJT injection saturation current</td>
</tr>
<tr>
<td>ISDIF</td>
<td>A/m$^2$</td>
<td>0</td>
<td>Body to source/drain injection saturation current</td>
</tr>
<tr>
<td>ISREC</td>
<td>A/m$^2$</td>
<td>1.0e-5</td>
<td>Recombination in depletion saturation current</td>
</tr>
<tr>
<td>ISTUN</td>
<td>A/m$^2$</td>
<td>0.0</td>
<td>Reverse tunneling saturation current</td>
</tr>
<tr>
<td>K1</td>
<td>V$^{1/2}$</td>
<td>0.6</td>
<td>First-order body effect coefficient</td>
</tr>
<tr>
<td>K2</td>
<td>-</td>
<td>0</td>
<td>Second-order body effect coefficient</td>
</tr>
<tr>
<td>K3</td>
<td>-</td>
<td>0</td>
<td>Narrow coefficient</td>
</tr>
<tr>
<td>K3B</td>
<td>1/V</td>
<td>0</td>
<td>Body effect coefficient of k3</td>
</tr>
<tr>
<td>KB1</td>
<td>-</td>
<td>1</td>
<td>Coefficient of $V_{bs0}$ dependency on $V_{gbs}$</td>
</tr>
<tr>
<td>KB3</td>
<td>-</td>
<td>1</td>
<td>Coefficient of $V_{bs0}$ dependency on $V_{gs}$ at subthreshold region</td>
</tr>
<tr>
<td>KBJT1</td>
<td>m/V</td>
<td>0</td>
<td>Parasitic bipolar early effect coefficient</td>
</tr>
<tr>
<td>KETA</td>
<td>m</td>
<td>-0.6</td>
<td>Body-bias coefficient of bulk charge effect</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>---------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>LINT</td>
<td>m</td>
<td>0.0</td>
<td>Length offset fitting parameter from I-V without bias</td>
</tr>
<tr>
<td>MXC</td>
<td></td>
<td>-0.9</td>
<td>Fitting parameter for $A_{beff}$ calculation</td>
</tr>
<tr>
<td>NDIODE</td>
<td></td>
<td>1.0</td>
<td>Diode non-ideality factor</td>
</tr>
<tr>
<td>NFACTOR</td>
<td></td>
<td>1</td>
<td>Subthreshold swing factor</td>
</tr>
<tr>
<td>NGIDL</td>
<td>V</td>
<td>1.2</td>
<td>GIDL $V_{ds}$ enhancement coefficient</td>
</tr>
<tr>
<td>NLX</td>
<td>m</td>
<td>1.74e-7</td>
<td>Lateral non-uniform doping parameter</td>
</tr>
<tr>
<td>NTUN</td>
<td></td>
<td>10.0</td>
<td>Reverse tunneling non-ideality factor</td>
</tr>
<tr>
<td>PCLM</td>
<td></td>
<td>1.3</td>
<td>Channel length modulation parameter</td>
</tr>
<tr>
<td>PDIBL1</td>
<td></td>
<td>0.39</td>
<td>First output resistance DIBL effect correction parameter</td>
</tr>
<tr>
<td>PDIBL2</td>
<td></td>
<td>0.0086</td>
<td>Second output resistance DIBL effect correction parameter</td>
</tr>
<tr>
<td>PRWB</td>
<td>1/V^1</td>
<td>0</td>
<td>Body effect coefficient of $R_{dsw}$</td>
</tr>
<tr>
<td>PRWG</td>
<td>1/V^{1/2}</td>
<td>0</td>
<td>Gate bias effect coefficient of $R_{dsw}$</td>
</tr>
<tr>
<td>PVAG</td>
<td></td>
<td>0.0</td>
<td>Gate dependence of Early voltage</td>
</tr>
<tr>
<td>RBODY</td>
<td>ohm/m^2</td>
<td>0.0</td>
<td>Intrinsic body contact sheet resistance</td>
</tr>
<tr>
<td>RBSH</td>
<td>ohm/m^2</td>
<td>0.0</td>
<td>Extrinsic body contact sheet resistance</td>
</tr>
<tr>
<td>RDSW</td>
<td>Ω∗µm^2</td>
<td>100</td>
<td>Parasitic resistance per unit width</td>
</tr>
<tr>
<td>RSH</td>
<td>ohm/square</td>
<td>0.0</td>
<td>Source/drain sheet resistance in ohm per square</td>
</tr>
<tr>
<td>U0</td>
<td>cm^2/(V·sec)</td>
<td>NMOS-670 PMOS-250</td>
<td>Mobility at Temp=Tnom</td>
</tr>
</tbody>
</table>
### AC and Capacitance Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASD</td>
<td>V</td>
<td>0.3</td>
<td>Source/drain bottom diffusion smoothing parameter</td>
</tr>
<tr>
<td>CF</td>
<td>F/m</td>
<td>cal.</td>
<td>Gate to source/drain fringing field capacitance</td>
</tr>
<tr>
<td>CGDL</td>
<td>F/m</td>
<td>0.0</td>
<td>Lightly doped drain-gate region overlap capacitance</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>CGDO</td>
<td>F/m</td>
<td>calculated</td>
<td>Non LDD region drain-gate overlap capacitance per channel length</td>
</tr>
<tr>
<td>CGEO</td>
<td>F/m</td>
<td>0.0</td>
<td>Gate-substrate overlap capacitance per channel length</td>
</tr>
<tr>
<td>CGSL</td>
<td>F/m</td>
<td>0.0</td>
<td>Lightly doped source-gate region overlap capacitance</td>
</tr>
<tr>
<td>CGSO</td>
<td>F/m</td>
<td>calculated</td>
<td>Non LDD region source-gate overlap capacitance per channel length</td>
</tr>
<tr>
<td>CJSWG</td>
<td>F/m²</td>
<td>1.e-10</td>
<td>Source/drain (gate side) sidewall junction capacitance per unit width (normalized to 100nm Tsi)</td>
</tr>
<tr>
<td>CKAPPA</td>
<td>F/m</td>
<td>0.6</td>
<td>Coefficient for lightly doped region overlap capacitance fringing field capacitance</td>
</tr>
<tr>
<td>CLC</td>
<td>m</td>
<td>0.1e-7</td>
<td>Constant term for the short channel model</td>
</tr>
<tr>
<td>CLE</td>
<td>-</td>
<td>0.0</td>
<td>Exponential term for the short channel model</td>
</tr>
<tr>
<td>CSDESW</td>
<td>F/m</td>
<td>0.0</td>
<td>Source/drain sidewall fringing capacitance per unit length</td>
</tr>
<tr>
<td>CSDMIN</td>
<td>V</td>
<td>cal.</td>
<td>Source/drain bottom diffusion minimum capacitance</td>
</tr>
<tr>
<td>DLC</td>
<td>m</td>
<td>lint</td>
<td>Length offset fitting parameter for gate charge</td>
</tr>
<tr>
<td>DWC</td>
<td>m</td>
<td>wint</td>
<td>Width offset fitting parameter from C-V</td>
</tr>
<tr>
<td>MJSWG</td>
<td>V</td>
<td>0.5</td>
<td>Source/drain (gate side) sidewall junction capacitance grading coefficient</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PBSWG</td>
<td>V</td>
<td>0.7</td>
<td>Source/drain (gate side) sidewall junction capacitance built in potential</td>
</tr>
<tr>
<td>TT</td>
<td>second</td>
<td>1ps</td>
<td>Diffusion capacitance transit time coefficient</td>
</tr>
<tr>
<td>VSDFB</td>
<td>V</td>
<td>cal.</td>
<td>Source/drain bottom diffusion capacitance flatband voltage</td>
</tr>
<tr>
<td>VSDTH</td>
<td>V</td>
<td>cal.</td>
<td>Source/drain bottom diffusion capacitance threshold voltage</td>
</tr>
<tr>
<td>XPART</td>
<td>-</td>
<td>0</td>
<td>Charge partitioning rate flag</td>
</tr>
</tbody>
</table>

**Temperature Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>m/sec</td>
<td>3.3e4</td>
<td>Temperature coefficient for $U_a$</td>
</tr>
<tr>
<td>CTH0</td>
<td>$^\circ$C/(W*s)</td>
<td>0</td>
<td>Normalized thermal capacity</td>
</tr>
<tr>
<td>KT1</td>
<td>V</td>
<td>-0.11</td>
<td>Temperature coefficient for the threshold voltage</td>
</tr>
<tr>
<td>KT2</td>
<td></td>
<td>0.022</td>
<td>Body-bias coefficient of the threshold voltage temperature effect</td>
</tr>
<tr>
<td>KTIL</td>
<td>$V^\ast m$</td>
<td>0</td>
<td>Channel length dependence of the temperature coefficient for the threshold voltage</td>
</tr>
<tr>
<td>PRT</td>
<td>$\Omega$-um</td>
<td>0</td>
<td>Temperature coefficient for $R_{ds}$</td>
</tr>
<tr>
<td>RTH0</td>
<td>$^\circ$C/W</td>
<td>0</td>
<td>Normalized thermal resistance</td>
</tr>
<tr>
<td>TNOM</td>
<td>$^\circ$C</td>
<td>25</td>
<td>Temperature at which parameters are expected</td>
</tr>
<tr>
<td>UA1</td>
<td>m/V</td>
<td>4.31e-9</td>
<td>Temperature coefficient for $U_a$</td>
</tr>
<tr>
<td>UB1</td>
<td>(m/V)$^2$</td>
<td>-7.61e-18</td>
<td>Temperature coefficient for $U_b$</td>
</tr>
</tbody>
</table>
Selecting MOSFET Models: LEVEL 47-62

LEVEL 59 Template Output

Additional element templates are added to this model for output of state variables, stored charges, capacitor currents and capacitances.

BSIM3SOI MOSFET FD (LEVEL 59) Template Output

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>LV1</td>
<td>Channel length (L)</td>
</tr>
<tr>
<td>W</td>
<td>LV2</td>
<td>Channel width (W)</td>
</tr>
<tr>
<td>AD</td>
<td>LV3</td>
<td>Area of the drain diode (AD)</td>
</tr>
<tr>
<td>Name</td>
<td>Alias</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>AS</td>
<td>LV4</td>
<td>Area of the source diode (AS)</td>
</tr>
<tr>
<td>ICVDS</td>
<td>LV5</td>
<td>Initial condition for drain-source voltage (VDS)</td>
</tr>
<tr>
<td>ICVGS</td>
<td>LV6</td>
<td>Initial condition for gate-source voltage (VGS)</td>
</tr>
<tr>
<td>ICVES</td>
<td>LV7</td>
<td>Initial condition for Substrate-source voltage (VES)</td>
</tr>
<tr>
<td>VTH</td>
<td>LV9</td>
<td>Threshold voltage (bias dependent)</td>
</tr>
<tr>
<td>VDSAT</td>
<td>LV10</td>
<td>Saturation voltage (VDSAT)</td>
</tr>
<tr>
<td>PD</td>
<td>LV11</td>
<td>Drain diode periphery (PD)</td>
</tr>
<tr>
<td>PS</td>
<td>LV12</td>
<td>Source diode periphery (PS)</td>
</tr>
<tr>
<td>RDS</td>
<td>LV13</td>
<td>Drain resistance (squares) (RDS)</td>
</tr>
<tr>
<td>RSS</td>
<td>LV14</td>
<td>Source resistance (squares) (RSS)</td>
</tr>
<tr>
<td>GDEFF</td>
<td>LV16</td>
<td>Effective drain conductance (1/RDeff)</td>
</tr>
<tr>
<td>GSEFF</td>
<td>LV17</td>
<td>Effective source conductance (1/RSeff)</td>
</tr>
<tr>
<td>COVLGS</td>
<td>LV36</td>
<td>Gate-source overlap capacitance</td>
</tr>
<tr>
<td>COVLGD</td>
<td>LV37</td>
<td>Gate-drain overlap capacitance</td>
</tr>
<tr>
<td>COVLGE</td>
<td>LV38</td>
<td>Gate-substrate overlap capacitance</td>
</tr>
<tr>
<td>VES</td>
<td>LX1</td>
<td>Substrate-source voltage (VES)</td>
</tr>
<tr>
<td>VGS</td>
<td>LX2</td>
<td>Gate-source voltage (VGS)</td>
</tr>
<tr>
<td>VDS</td>
<td>LX3</td>
<td>Drain-source voltage (VDS)</td>
</tr>
<tr>
<td>CDO</td>
<td>LX4</td>
<td>DC drain current (CDO)</td>
</tr>
<tr>
<td>CBSO</td>
<td>LX5</td>
<td>DC source-body diode current (CBSO)</td>
</tr>
<tr>
<td>CBDO</td>
<td>LX6</td>
<td>DC drain-body diode current (CBDO)</td>
</tr>
<tr>
<td>GMO</td>
<td>LX7</td>
<td>DC gate transconductance (GMO)</td>
</tr>
<tr>
<td>GDSO</td>
<td>LX8</td>
<td>DC drain-source conductance (GDSO)</td>
</tr>
<tr>
<td>GMESO</td>
<td>LX9</td>
<td>DC substrate transconductance (GMBSO)</td>
</tr>
</tbody>
</table>
## Meyer and Charge Conservation Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QB</td>
<td>LX12</td>
<td>Body charge (QB)</td>
</tr>
<tr>
<td>CQB</td>
<td>LX13</td>
<td>Body charge current (CQB)</td>
</tr>
<tr>
<td>QG</td>
<td>LX14</td>
<td>Gate charge (QG)</td>
</tr>
<tr>
<td>CQG</td>
<td>LX15</td>
<td>Gate charge current (CQG)</td>
</tr>
<tr>
<td>QD</td>
<td>LX16</td>
<td>Channel charge (QD)</td>
</tr>
<tr>
<td>CQD</td>
<td>LX17</td>
<td>Channel charge current (CQD)</td>
</tr>
<tr>
<td>CGGBO</td>
<td>LX18</td>
<td>( \frac{\partial Q_g}{\partial V_{gb}} = CGS + CGD + CGB )</td>
</tr>
<tr>
<td>CGDBO</td>
<td>LX19</td>
<td>( \frac{\partial Q_g}{\partial V_{db}} ), (for Meyer ( CGD=-CGDBO ))</td>
</tr>
<tr>
<td>CGSBO</td>
<td>LX20</td>
<td>( \frac{\partial Q_g}{\partial V_{sb}} ), (for Meyer ( CGS=-CGSBO ))</td>
</tr>
<tr>
<td>CBGBO</td>
<td>LX21</td>
<td>( \frac{\partial Q_b}{\partial V_{gb}} ), (for Meyer ( CGB=-CBGBO ))</td>
</tr>
<tr>
<td>CBDBO</td>
<td>LX22</td>
<td>( \frac{\partial Q_b}{\partial V_{db}} )</td>
</tr>
<tr>
<td>CBSBO</td>
<td>LX23</td>
<td>( \frac{\partial Q_b}{\partial V_{sb}} )</td>
</tr>
<tr>
<td>CDGBO</td>
<td>LX32</td>
<td>( \frac{\partial Q_d}{\partial V_{gb}} )</td>
</tr>
<tr>
<td>CDDBO</td>
<td>LX33</td>
<td>( \frac{\partial Q_d}{\partial V_{db}} )</td>
</tr>
<tr>
<td>CDSBO</td>
<td>LX34</td>
<td>( \frac{\partial Q_d}{\partial V_{sb}} )</td>
</tr>
<tr>
<td>QE</td>
<td>LX35</td>
<td>Substrate charge (QE)</td>
</tr>
<tr>
<td>CQE</td>
<td>LX36</td>
<td>Substrate charge current (CQE)</td>
</tr>
<tr>
<td>CDEBO</td>
<td>LX37</td>
<td>( \frac{\partial Q_d}{\partial V_{eb}} )</td>
</tr>
<tr>
<td>Name</td>
<td>Alias</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>CBEBO</td>
<td>LX38</td>
<td>( CBEBO = \partial Q_b / \partial V_{eb} )</td>
</tr>
<tr>
<td>CEEBO</td>
<td>LX39</td>
<td>( CEEBO = \partial Q_e / \partial V_{eb} )</td>
</tr>
<tr>
<td>CEGBO</td>
<td>LX40</td>
<td>( CEGBO = \partial Q_e / \partial V_{gb} )</td>
</tr>
<tr>
<td>CEDBO</td>
<td>LX41</td>
<td>( CEDBO = \partial Q_e / \partial V_{db} )</td>
</tr>
<tr>
<td>CESBO</td>
<td>LX42</td>
<td>( CESBO = \partial Q_e / \partial V_{sb} )</td>
</tr>
<tr>
<td>VBS</td>
<td>LX43</td>
<td>Body-source voltage (VBS)</td>
</tr>
<tr>
<td>ICH</td>
<td>LX44</td>
<td>Channel current</td>
</tr>
<tr>
<td>IBJT</td>
<td>LX45</td>
<td>Parasitic BJT collector current</td>
</tr>
<tr>
<td>III</td>
<td>LX46</td>
<td>Impact Ionization current</td>
</tr>
<tr>
<td>IGIDL</td>
<td>LX47</td>
<td>GIDL current</td>
</tr>
<tr>
<td>ITUN</td>
<td>LX48</td>
<td>Tunneling current</td>
</tr>
</tbody>
</table>
LEVEL 60 UC Berkeley BSIM3-SOI DD Model

The UC Berkeley SOI model (BSIM3SOI) supports Fully Depleted (FD), Partially Depleted (PD), and Dynamically Depleted (DD) SOI devices.

BSIM3DD2.2 for DD SOI devices is now installed in Star-Hspice as LEVEL 60. This model is described in the “BSIM3DD2.1 MOSFET MODEL User’s Manual,” which can be found at http://www-device.eecs.berkeley.edu/~bsim3soi.

In BSIM3DD2.1, many advanced concepts are introduced to allow transition between PD and FD operation dynamically and continuously, namely the Dynamic Depletion approach.

Model Features

- Dynamic depletion approach is applied on both I-V and C-V. Charge and drain current are scaleable with Tbox and Tsi continuously
- Supports external body bias and backgate bias; a total of 6 nodes
- Real floating body simulation in both I-V and C-V. Body potential is properly bounded by diode and C-V formulation
- Self-heating implementation improved
- Improved impact ionization current model
- Various diode leakage components and parasitic bipolar current included
- New depletion charge model (EBCI) introduced for better accuracy in capacitive coupling prediction. An improved BSIM3v3 based model is added as well
- Dynamic depletion can suit different requirements for SOI technologies
- Single I-V expression as in BSIM3v3.1 to assure continuities of Ids, Gds, Gm and their derivatives for all bias conditions
Syntax

The general syntax for including a BSIM3SOI MOSFET element in a Star-Hspice netlist is:

**General Form**

```
Mxxx nd ng ns ne <np> mname <L=val> <W=val> <M=val> <AD=val> <AS=val>
+ <PD=val> <PS=val> <NRD=val> <NRS=val> <NRB=val> <RTHO=val> <CTHO=val>
+ <off> <BJToff=val> <IC=Vds, Vgs, Vbs, Ves, Vps>
```

where the angle brackets indicate optional parameters.

The arguments are as follows:

- **Mxxx**: SOI MOSFET element name. Must begin with M, which can be followed by up to 1023 alphanumeric characters
- **nd**: Drain terminal node name or number
- **ng**: Front gate node name or number
- **ns**: Source terminal node name or number
- **ne**: Back gate (or Substrate) node name or number
- **np**: External body contact node name or number
- **mname**: MOSFET model name reference
- **L**: SOI MOSFET channel length in meters. This parameter overrides DEFL in an OPTIONS statement Default=DEFL with a maximum of 0.1m
- **W**: SOI MOSFET channel width in meters. This parameter overrides DEFW in an OPTIONS statement Default=DEFW with a maximum of 0.1m
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Multiplier to simulate multiple SOI MOSFETs in parallel. All channel widths, diode leakages, capacitances and resistances are affected by the setting of M</td>
<td>Default=1</td>
</tr>
<tr>
<td>AD</td>
<td>Drain diffusion area. Overrides DEFAD in the OPTIONS statement</td>
<td>Default=DEFAD</td>
</tr>
<tr>
<td>AS</td>
<td>Source diffusion area. Overrides DEFAS in the OPTIONS statement</td>
<td>Default=DEFAS</td>
</tr>
<tr>
<td>PD</td>
<td>Perimeter of the drain junction, including the channel edge. Overrides DEFPD in the OPTIONS statement</td>
<td></td>
</tr>
<tr>
<td>PS</td>
<td>Perimeter of the source junction, including the channel edge. Overrides DEFPS in the OPTIONS statement</td>
<td></td>
</tr>
<tr>
<td>NRD</td>
<td>Number of squares of drain diffusion for drain series resistance. Overrides DEFNRD in the OPTIONS statement</td>
<td></td>
</tr>
<tr>
<td>NRS</td>
<td>Number of squares of source diffusion for source series resistance. Overrides DEFNRS in the OPTIONS statement</td>
<td></td>
</tr>
<tr>
<td>NRB</td>
<td>Number of squares for body series resistance</td>
<td></td>
</tr>
<tr>
<td>RDC</td>
<td>Additional drain resistance due to contact resistance with units of ohms. This value overrides the RDC setting in the model specification</td>
<td>Default =0.0</td>
</tr>
<tr>
<td>RSC</td>
<td>Additional source resistance due to contact resistance with units of ohms. This value overrides the RDC setting in the model specification</td>
<td>Default=0.0</td>
</tr>
</tbody>
</table>
RTHO Thermal resistance per unit width
   ■ If not specified, RTHO is extracted from the model card
   ■ If specified, it will override the one in the model card

CTHO Thermal capacitance per unit width
   ■ If not specified, CTHO is extracted from the model card
   ■ If specified, it will override the one in the model card

OFF Sets initial condition to OFF for this element in DC analysis

BJTOFF Turns off BJT if equal to 1

IC Initial guess in the order (drain, front gate, internal body, back gate, external voltage). $V_{ps}$ will be ignored in the case of 4-terminal device. These are used when UIC is present in the .TRAN statement and are overriden by the .IC statement

### Level 60 Model Parameters

*Note: All additional BSIM3v3 parameters are shown in bold.*

### BSIMSOI Model Control Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>level</td>
<td>Level 60 for BSIMSOI</td>
<td>-</td>
<td>9</td>
<td>-</td>
</tr>
<tr>
<td>shMod</td>
<td>Flag for self-heating</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0 - no self-heating</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - self-heating</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### BSIMSOI Model Control Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mobmod</td>
<td>Mobility model selector</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>capmod</td>
<td>Flag for the short channel capacitance model</td>
<td>-</td>
<td>2</td>
<td>nl-1</td>
</tr>
<tr>
<td>noimod</td>
<td>Flag for noise model</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

### Process Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tsi</td>
<td>Silicon film thickness</td>
<td>m</td>
<td>$10^{-7}$</td>
<td></td>
</tr>
<tr>
<td>Tbox</td>
<td>Buried oxide thickness</td>
<td>m</td>
<td>$3 \times 10^{-7}$</td>
<td></td>
</tr>
<tr>
<td>Tox</td>
<td>Gate oxide thickness</td>
<td>m</td>
<td>$1 \times 10^{-8}$</td>
<td></td>
</tr>
<tr>
<td>Nch</td>
<td>Channel doping concentration</td>
<td>1/cm$^3$</td>
<td>$1.7 \times 10^{17}$</td>
<td></td>
</tr>
<tr>
<td>Nsub</td>
<td>Substrate doping concentration</td>
<td>1/cm$^3$</td>
<td>$6 \times 10^{16}$</td>
<td>nl-2</td>
</tr>
<tr>
<td>ngate</td>
<td>Poly gate doping concentration</td>
<td>1/cm$^3$</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

### DC Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>vth0</td>
<td>Threshold voltage @ $V_{bs} = 0$ for long and wide device</td>
<td>-</td>
<td>0.7</td>
<td>nl-3</td>
</tr>
<tr>
<td>k1</td>
<td>First order body effect coefficient</td>
<td>$V^{1/2}$</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>k2</td>
<td>Second order body effect coefficient</td>
<td>-</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>k3</td>
<td>Narrow width coefficient</td>
<td>-</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
## DC Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k3b</td>
<td>Body efficient coefficient of k3</td>
<td>1/V</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Vbsa</td>
<td>Transition body voltage offset</td>
<td>V</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>delp</td>
<td>Constant for limiting $V_{b_{\text{seff}}} \to \phi_s$</td>
<td>V</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>Kb1</td>
<td>Coefficient of $V_{bs0}$ dependency on $V_{es}$</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Kb3</td>
<td>Coefficient of $V_{bs0}$ dependency on $V_{gs}$ at subthreshold region</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Dvbd0</td>
<td>First coefficient of $V_{bs0}$ dependency on $L_{eff}$</td>
<td>V</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Dvbd1</td>
<td>Second coefficient of $V_{bs0}$ dependency on $L_{eff}$</td>
<td>V</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>w0</td>
<td>Narrow width parameter</td>
<td>m</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>nlx</td>
<td>Lateral non-uniform doping parameter</td>
<td>m</td>
<td>1.74e-7</td>
<td></td>
</tr>
<tr>
<td>dvt0</td>
<td>First coefficient of short-channel effect on $V_{th}$</td>
<td>-</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td>dvt1</td>
<td>Second coefficient of short-channel effect on $V_{th}$</td>
<td>-</td>
<td>0.53</td>
<td></td>
</tr>
<tr>
<td>dvt2</td>
<td>Body-bias coefficient of short-channel effect on $V_{th}$</td>
<td>1/V</td>
<td>-0.032</td>
<td></td>
</tr>
<tr>
<td>dvt0w</td>
<td>First coefficient of narrow width effect on $V_{th}$ for small channel length</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>dvt1w</td>
<td>Second coefficient of narrow width effect on $V_{th}$ for small channel length</td>
<td>-</td>
<td>5.3e6</td>
<td>-</td>
</tr>
<tr>
<td>dvt2w</td>
<td>Body-bias coefficient of narrow width effect on $V_{th}$ for small channel length</td>
<td>1/V</td>
<td>-0.032</td>
<td>-</td>
</tr>
<tr>
<td>u0</td>
<td>Mobility at Temp = $T_{\text{nom}}$ for NMOSFET</td>
<td>$\text{cm}^2/(\text{V-sec})$</td>
<td>670/250</td>
<td></td>
</tr>
<tr>
<td>ua</td>
<td>First-order mobility degradation coefficient</td>
<td>m/V</td>
<td>2.25e-9</td>
<td></td>
</tr>
</tbody>
</table>
## DC Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ub</td>
<td>Second-order mobility degradation coefficient</td>
<td>(m/V)^2</td>
<td>5.9e-19</td>
<td></td>
</tr>
<tr>
<td>uc</td>
<td>Body-effect of mobility degradation coefficient</td>
<td>1/V</td>
<td>-0.0465</td>
<td></td>
</tr>
<tr>
<td>vsat</td>
<td>Saturation velocity at Temp = Tnom</td>
<td>m/sec</td>
<td>8e4</td>
<td></td>
</tr>
<tr>
<td>a0</td>
<td>Bulk charge effect coefficient for channel length</td>
<td>-</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>ags</td>
<td>Gate bias coefficient of $A_{bulk}$</td>
<td>1/V</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>b0</td>
<td>Bulk charge effect coefficient for channel width</td>
<td>m</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td>Bulk charge effect width offset</td>
<td>m</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>keta</td>
<td>Body-bias coefficient of bulk charge effect</td>
<td>m</td>
<td>-0.6</td>
<td></td>
</tr>
<tr>
<td>Abp</td>
<td>Coefficient of $A_{beff}$ dependency on $V_{gst}$</td>
<td>-</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>mxc</td>
<td>Fitting parameter for $A_{beff}$ calculation</td>
<td>-</td>
<td>-0.9</td>
<td></td>
</tr>
<tr>
<td>dice0</td>
<td>DICE bulk charge factor</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>First non-saturation effect parameter</td>
<td>1/V</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>Second non-saturation effect parameter</td>
<td>0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>rdswn</td>
<td>Parasitic resistance per unit width</td>
<td>$\Omega$-m$^{\mu}$-m$^{wr}$</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>prwb</td>
<td>Body effect coefficient Rdsw</td>
<td>1/V</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>prwg</td>
<td>Gate bias effect coefficient of Rdsw</td>
<td>1/V$^{1/2}$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>wr</td>
<td>Width offset from Weff for Rds calculation</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>wint</td>
<td>Width offset fitting parameter from I-V without bias</td>
<td>m</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>lint</td>
<td>Length offset fitting parameter from I-V without bias</td>
<td>m</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>
## DC Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dwg</td>
<td>Coefficient of ( W_{\text{eff}} )'s gate dependence</td>
<td>m/V</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>dwb</td>
<td>Coefficient of ( W_{\text{eff}} )'s substrate body bias dependence</td>
<td>m/V(^{1/2})</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>voff</td>
<td>Offset voltage in the subthreshold region for large ( W ) and ( L )</td>
<td>V</td>
<td>-0.08</td>
<td></td>
</tr>
<tr>
<td>nfactor</td>
<td>Subthreshold swing factor</td>
<td>-</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>etab</td>
<td>Body-bias coefficient for the subthreshold DIBL effect</td>
<td>1/V</td>
<td>-0.07</td>
<td></td>
</tr>
<tr>
<td>dsub</td>
<td>DIBL coefficient exponent</td>
<td>-</td>
<td>0.56</td>
<td></td>
</tr>
<tr>
<td>cit</td>
<td>Interface trap capacitance</td>
<td>F/m²</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>cdsc</td>
<td>Drain/Source to channel coupling capacitance</td>
<td>F/m²</td>
<td>2.4e-4</td>
<td></td>
</tr>
<tr>
<td>cdscb</td>
<td>Body-bias sensitivity of ( C_{\text{dsc}} )</td>
<td>F/m²</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>cdscd</td>
<td>Drain-bias sensitivity of ( C_{\text{dsc}} )</td>
<td>F/m²</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>pclm</td>
<td>Channel length modulation parameter</td>
<td>-</td>
<td>1.3</td>
<td></td>
</tr>
<tr>
<td>pdibl1</td>
<td>First output resistance DIBL effect correction parameter</td>
<td>-</td>
<td>0.39</td>
<td></td>
</tr>
<tr>
<td>pdibl2</td>
<td>Second output resistance DIBL effect correction parameter</td>
<td>-</td>
<td>0.086</td>
<td></td>
</tr>
<tr>
<td>drouth</td>
<td>L dependence coefficient of the DIBL correction parameter in ( R_{\text{out}} )</td>
<td>-</td>
<td>0.56</td>
<td></td>
</tr>
<tr>
<td>pvag</td>
<td>Gate dependence of early voltage</td>
<td>-</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>delta</td>
<td>Effective ( V_{\text{ds}} ) parameter</td>
<td>-</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>alli</td>
<td>1(^{st}) ( L_{\text{eff}} ) dependence ( V_{\text{dsatii}} ) parameter</td>
<td>1/V</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>bii</td>
<td>2(^{nd}) ( L_{\text{eff}} ) dependence ( V_{\text{dsatii}} ) parameter</td>
<td>m/V</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>cii</td>
<td>1(^{st}) ( V_{\text{ds}} ) dependence ( V_{\text{dsatii}} ) parameter</td>
<td>-</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>dlii</td>
<td>2(^{nd}) dependence ( V_{\text{dsatii}} ) parameter</td>
<td>V</td>
<td>-1.0</td>
<td></td>
</tr>
<tr>
<td>alpha0</td>
<td>First parameter of impact ionization current</td>
<td>m/V</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>
### DC Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha1</td>
<td>Second parameter of impact ionization current</td>
<td>1/V</td>
<td>1.0</td>
<td>-</td>
</tr>
<tr>
<td>beta0</td>
<td>Third parameter of impact ionization current</td>
<td>V</td>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>Agidl</td>
<td>GIDL constant</td>
<td>Ω⁻¹</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>Bgidl</td>
<td>GIDL exponential coefficient</td>
<td>V/m</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>Ngidl</td>
<td>GIDL $V_{ds}$ enhancement coefficient</td>
<td>V</td>
<td>1.2</td>
<td>-</td>
</tr>
<tr>
<td>ntun</td>
<td>Reverse tunneling non-ideality factor</td>
<td>-</td>
<td>10.0</td>
<td>-</td>
</tr>
<tr>
<td>Ndiode</td>
<td>Diode non-ideality factor</td>
<td>-</td>
<td>1.0</td>
<td>-</td>
</tr>
<tr>
<td>Isbjt</td>
<td>BJT injection saturation current</td>
<td>A/m²</td>
<td>1e-6</td>
<td>-</td>
</tr>
<tr>
<td>Isdif</td>
<td>Body to source/drain injection saturation current</td>
<td>A/m²</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>Isrec</td>
<td>Recombination in depletion saturation current</td>
<td>A/m²</td>
<td>1e-5</td>
<td>-</td>
</tr>
<tr>
<td>Istun</td>
<td>Reverse tunneling saturation current</td>
<td>A/m²</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>Edl</td>
<td>Electron diffusion length</td>
<td>m</td>
<td>2e-6</td>
<td>-</td>
</tr>
<tr>
<td>Kbjt1</td>
<td>Parasitic bipolar early effect coefficient</td>
<td>m/V</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Rbody</td>
<td>Intrinsic body contact sheet resistance</td>
<td>ohm/m²</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>Rbsh</td>
<td>Extrinsic body contact sheet resistance</td>
<td>ohm/m²</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>rsh</td>
<td>Source drain sheet resistance in ohm per square</td>
<td>Ω/square</td>
<td>0.0</td>
<td>-</td>
</tr>
</tbody>
</table>
### AC and Capacitance Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>xpart</td>
<td>Charge partitioning rate flag</td>
<td>-</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>cgso</td>
<td>Non LDD region source-gate overlap capacitance per channel length</td>
<td>F/m</td>
<td>calculated</td>
<td>nC-1</td>
</tr>
<tr>
<td>cgdo</td>
<td>Non LDD region drain-gate overlap capacitance per channel length</td>
<td>F/m</td>
<td>calculated</td>
<td>nC-2</td>
</tr>
<tr>
<td>cgeo</td>
<td>Gate substrate overlap capacitance per unit channel length</td>
<td>F/m</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>cjswg</td>
<td>Source/Drain (gate side) sidewall junction capacitance per unit width (normalized to 100nm T_{si})</td>
<td>F/m²</td>
<td>1e-10</td>
<td>-</td>
</tr>
<tr>
<td>pbswg</td>
<td>Source/Drain (gate side) sidewall junction capacitance built in potential</td>
<td>V</td>
<td>.7</td>
<td>-</td>
</tr>
<tr>
<td>mjswg</td>
<td>Source/Drain (gate side) sidewall junction capacitance grading coefficient</td>
<td>V</td>
<td>0.5</td>
<td>-</td>
</tr>
<tr>
<td>tt</td>
<td>Diffusion capacitance transit time coefficient</td>
<td>second</td>
<td>1ps</td>
<td>-</td>
</tr>
<tr>
<td>vsdfb</td>
<td>Source/drain bottom diffusion capacitance flatband voltage</td>
<td>V</td>
<td>calculated</td>
<td>nC-3</td>
</tr>
<tr>
<td>vsdth</td>
<td>Source/drain bottom diffusion capacitance threshold voltage</td>
<td>V</td>
<td>calculated</td>
<td>nC-4</td>
</tr>
<tr>
<td>csdmin</td>
<td>Source/drain bottom diffusion minimum capacitance</td>
<td>V</td>
<td>calculated</td>
<td>nC-5</td>
</tr>
<tr>
<td>asd</td>
<td>Source/drain bottom diffusion smoothing parameter</td>
<td>-</td>
<td>0.3</td>
<td>-</td>
</tr>
<tr>
<td>csdesw</td>
<td>Source/drain sidewall fringing capacitance per unit length</td>
<td>F/m</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>cgs1</td>
<td>Light doped source-gate region overlap capacitance</td>
<td>F/m</td>
<td>0.0</td>
<td>-</td>
</tr>
</tbody>
</table>
### AC and Capacitance Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgd1</td>
<td>Light doped drain-gate region overlap capacitance</td>
<td>F/m</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>ckappa</td>
<td>Coefficient for lightly doped region overlap capacitance fringing field capacitance</td>
<td>F/m</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>cf</td>
<td>Gate to source/drain fringing field capacitance</td>
<td>F/m</td>
<td>calculated</td>
<td>nC-6</td>
</tr>
<tr>
<td>clc</td>
<td>Constant term for the short channel mode</td>
<td>m</td>
<td>0.1\times10^{-1}</td>
<td></td>
</tr>
<tr>
<td>cle</td>
<td>Exponential term for the short channel mode</td>
<td>none</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>dlc</td>
<td>Length offset fitting parameter from C-V</td>
<td>m</td>
<td>lint</td>
<td></td>
</tr>
<tr>
<td>dwc</td>
<td>Width offset fitting parameter from C-V</td>
<td>m</td>
<td>wint</td>
<td></td>
</tr>
</tbody>
</table>

### Temperature Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tnom</td>
<td>Temperature at which parameters are expected</td>
<td>°C</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>ute</td>
<td>Mobility temperature exponent</td>
<td>none</td>
<td>-1.5</td>
<td></td>
</tr>
<tr>
<td>kt1</td>
<td>Temperature coefficient for threshold voltage</td>
<td>V</td>
<td>-0.11</td>
<td></td>
</tr>
<tr>
<td>kt11</td>
<td>Channel length dependence of the temperature coefficient for threshold voltage</td>
<td>V'm</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>kt2</td>
<td>Body-bias coefficient of the Vth temperature effect</td>
<td>none</td>
<td>0.022</td>
<td></td>
</tr>
</tbody>
</table>
## Temperature Parameters

<table>
<thead>
<tr>
<th>SPICE Symbol</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
<th>Notes (below)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ua1</td>
<td>Temperature coefficient for $U_a$</td>
<td>m/V</td>
<td>4.31e-9</td>
<td>-</td>
</tr>
<tr>
<td>ub1</td>
<td>Temperature coefficient for $U_b$</td>
<td>(m/V)^2</td>
<td>-7.61e-18</td>
<td>-</td>
</tr>
<tr>
<td>uc1</td>
<td>Temperature coefficient for $U_c$</td>
<td>1/V</td>
<td>-.056</td>
<td>nT-1</td>
</tr>
<tr>
<td>at</td>
<td>Temperature coefficient for saturation velocity</td>
<td>m/sec</td>
<td>3.3e4</td>
<td>-</td>
</tr>
<tr>
<td>cth0</td>
<td>Normalized thermal capacity</td>
<td>m°C/ (W*sec)</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>prt</td>
<td>Temperature coefficient for $R_{ds}$</td>
<td>$\Omega \cdot \mu m$</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>rth0</td>
<td>Normalized thermal resistance</td>
<td>m°C/ W</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>xbjt</td>
<td>Power dependence of $i_{bjt}$ on temperature</td>
<td>none</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>xdif</td>
<td>Power dependence of $i_{dif}$ on temperature</td>
<td>none</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>xrec</td>
<td>Power dependence of $i_{rec}$ on temperature</td>
<td>none</td>
<td>20</td>
<td>-</td>
</tr>
<tr>
<td>xtun</td>
<td>Power dependence of $i_{tun}$ on temperature</td>
<td>none</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

## Model Parameter Notes

| nI-1 | Capmod 0 and 1 do not have the dynamic depletion calculation. Therefore, ddMod does not work with capmod. |
**Model Parameter Notes**

### nI-2

BSIMSOI refers substrate to the silicon below buried oxide, not the well region in BSIM3. It is used to calculate backgate flatband voltage ($V_{fb}$) and parameters related to source/drain diffusion bottom capacitance ($V_{sdth}$, $V_{sdfb}$, $C_{sdmin}$). Positive $n_{sub}$ means the same type of doping as the body and negative $n_{sub}$ means opposite type of doping.

### nC-1

If $cgso$ is not given then it is calculated using:

- If ($dlc$ is given and is greater than 0) then,
  
  $cgso = pl = (dlc * cox) - cgs 1$

- If (the previously calculated $cgso < 0$), then
  
  $cgso = 0$

- Else $cgso = 0.6 * Tsi * cox$

### nC-2

$Cgdo$ is calculated similar to $Csdo$

### nC-3

If ($n_{sub}$ is positive)

$$V_{sdfb} = \frac{kT}{q} \log \left( \frac{10^{20} \cdot n_{sub}}{n_i \cdot n_i} \right) - 0.3$$

Else

$$V_{sdfb} = -\frac{kT}{q} \log \left( \frac{10^{20}}{n_{sub}} \right) + 0.3$$
## Model Parameter Notes

### nC-4

If \( n_{\text{sub}} \) is positive

\[
\phi_{sd} = \frac{2kT}{q} \log \left( \frac{n_{\text{sub}}}{n_i} \right), \quad \gamma_{sd} = \frac{5.753 \times 10^{-12} \sqrt{n_{\text{sub}}}}{C_{\text{box}}}
\]

\[
V_{sdth} = V_{sdfb} + \phi_{sd} + \gamma_{sd} \sqrt{\phi_{sd}}
\]

else

\[
\phi_{sd} = \frac{2kT}{q} \log \left( -\frac{n_{\text{sub}}}{n_i} \right), \quad \gamma_{sd} = \frac{5.753 \times 10^{-12} \sqrt{-n_{\text{sub}}}}{C_{\text{box}}}
\]

\[
V_{sdth} = V_{sdfb} - \phi_{sd} - \gamma_{sd} \sqrt{\phi_{sd}}
\]

### nC-5

\[
X_{sddep} = \frac{2\varepsilon_{si} \phi_{sd}}{q |n_{\text{sub}}| \cdot 10^6} C_{sddep} = \frac{\varepsilon_{si}}{X_{sddep}} C_{idmin} = \frac{C_{sddep} C_{\text{box}}}{C_{sddep} + C_{\text{box}}}
\]

### nC-6

If \( cf \) is not given then it is calculated using

\[
CF = \frac{2 \varepsilon_{ox}}{\pi} \ln \left( 1 + 4 \times 10^{-7} \frac{T_{ox}}{T_{ox}} \right)
\]

### nT-1

For \( \text{mobmod}=1 \text{ and } 2 \), the unit is m/V². Default is -5.6E-11. For \( \text{mobmod}=3 \), unit is 1/V and default is -0.056.
LEVEL 61 RPI a-Si TFT Model

Star-Hspice LEVEL 61 is an AIM-SPICE MOS15 amorphous silicon (a-Si) thin-film transistor (TFT) model.

Model Features

AIM-SPICE MOS15 a-Si TFT model features include:

- Modified charge control model; induced charge trapped in localized states
- Above threshold includes:
  - Field effect mobility becoming a function of gate bias
  - Band mobility dominated by lattice scattering
- Below threshold
  - Fermi level located in deep localized states
  - Relate position of Fermi level, including the deep DOS back to the gate bias
- Empirical expression for current at large negative gate biases for hole-induced leakage current
- Interpolation techniques are applied to the equations to unify the model

Using LEVEL 61 with Star-Hspice

When using the AIM-SPICE MOS15 a-Si TFT model:

1. Set LEVEL=61 to identify the model as the AIM-SPICE MOS15 a-Si TFT model.
2. The default value for L is 100 m, and the default value for W is 100 m.
3. The LEVEL 61 model is a 3-terminal model. No bulk node exists; therefore no parasitic drain-bulk or source-build diodes are appended to the model. A fourth node can be specified, but does not affect simulation results.
4. The default room temperature is 25°C in Star-Hspice, but is 27°C in some other simulators. The user may choose whether or not to set the nominal room temperature.
simulation temperature to 27°C, by adding .OPTION TNOM=27 to the netlist.

Example

This is an example of how the Star-Hspice model and element statement modified for use with LEVEL 61.

mckt drain gate source nch L=10e-6 W=10e-6

.MODEL nch nmos LEVEL=61
+ alphasat = 0.6 cgdo = 0.0 cgso = 0.0 def0 = 0.6
+ delta = 5.0 el = 0.35 emu = 0.06 eps = 11
+ epsi = 7.4 gamma = 0.4 gmin = 1e23 iol = 3e-14
+ kasat = 0.006 kvt = -0.036 lambda = 0.0008 m = 2.5
+ muband = 0.001 rd = 0.0 rs = 0.0 sima0 = 1e-14
+ tnom = 27 tox = 1.0e-7 v0 = 0.12 vaa = 7.5e3
+ vdsl = 7 vfb = -3 vgs1 = 7 vmin = 0.3 vto = 0.0

LEVEL 61 Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHASAT</td>
<td>-</td>
<td>0.6</td>
<td>Saturation modulation parameter</td>
</tr>
<tr>
<td>CGDO</td>
<td>F/m</td>
<td>0.0</td>
<td>Gate-drain overlap capacitance per meter channel width</td>
</tr>
<tr>
<td>CGSO</td>
<td>F/m</td>
<td>0.0</td>
<td>Gate-source overlap capacitance per meter channel width</td>
</tr>
<tr>
<td>DEF0</td>
<td>eV</td>
<td>0.6</td>
<td>Dark Fermi level position</td>
</tr>
<tr>
<td>DELTA</td>
<td>-</td>
<td>5</td>
<td>Transition width parameter</td>
</tr>
<tr>
<td>EL</td>
<td>eV</td>
<td>0.35</td>
<td>Activation energy of the hole leakage current</td>
</tr>
<tr>
<td>EMU</td>
<td>eV</td>
<td>0.06</td>
<td>Field effect mobility activation energy</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>---------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>EPS</td>
<td>-</td>
<td>11</td>
<td>Relative dielectric constant of substrate</td>
</tr>
<tr>
<td>EPSI</td>
<td>-</td>
<td>7.4</td>
<td>Relative dielectric constant of gate insulator</td>
</tr>
<tr>
<td>GAMMA</td>
<td>-</td>
<td>0.4</td>
<td>Power law mobility parameter</td>
</tr>
<tr>
<td>GMIN</td>
<td>m(^{-3})eV(^{-1})</td>
<td>1E23</td>
<td>Minimum density of deep states</td>
</tr>
<tr>
<td>IOL</td>
<td>A</td>
<td>3E-14</td>
<td>Zero bias leakage current parameter</td>
</tr>
<tr>
<td>KASAT</td>
<td>1/°C</td>
<td>0.006</td>
<td>Temperature coefficient of ALPHASAT</td>
</tr>
<tr>
<td>KVT</td>
<td>V/°C</td>
<td>-0.036</td>
<td>Threshold voltage temperature coefficient</td>
</tr>
<tr>
<td>LAMBDA</td>
<td>1/V</td>
<td>0.0008</td>
<td>Output conductance parameter</td>
</tr>
<tr>
<td>M</td>
<td>-</td>
<td>2.5</td>
<td>Knee shape parameter</td>
</tr>
<tr>
<td>MUBAND</td>
<td>m(^2)/Vs</td>
<td>0.001</td>
<td>Conduction band mobility</td>
</tr>
<tr>
<td>RD</td>
<td>μ</td>
<td>0.0</td>
<td>Drain resistance</td>
</tr>
<tr>
<td>RS</td>
<td>μ</td>
<td>0.0</td>
<td>Source resistance</td>
</tr>
<tr>
<td>SIGMA0</td>
<td>A</td>
<td>1E-14</td>
<td>Minimum leakage current parameter</td>
</tr>
<tr>
<td>TNOM</td>
<td>°C</td>
<td>25</td>
<td>Parameter measurement temperature</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>1E-7</td>
<td>Thin-oxide thickness</td>
</tr>
<tr>
<td>V0</td>
<td>V</td>
<td>0.12</td>
<td>Characteristic voltage for deep states</td>
</tr>
<tr>
<td>VAA</td>
<td>V</td>
<td>7.5E3</td>
<td>Characteristic voltage for field effect mobility</td>
</tr>
<tr>
<td>VDSL</td>
<td>V</td>
<td>7</td>
<td>Hole leakage current drain voltage parameter</td>
</tr>
<tr>
<td>VFB</td>
<td>V</td>
<td>-3</td>
<td>Flat band voltage</td>
</tr>
<tr>
<td>VGSL</td>
<td>V</td>
<td>7</td>
<td>Hole leakage current gate voltage parameter</td>
</tr>
<tr>
<td>VMIN</td>
<td>V</td>
<td>0.3</td>
<td>Convergence parameter</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>---------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>VTO</td>
<td>V</td>
<td>0.0</td>
<td>Zero-bias threshold voltage</td>
</tr>
</tbody>
</table>

**Equivalent Circuit**

![Equivalent Circuit Diagram]

**Model Equations**

**Drain Current**

\[
I_{ds} = I_{leakage} + I_{ab}
\]

\[
I_{ab} = g_{ch} V_{dse} (1 + \text{LAMBDA} \cdot V_{ds})
\]

\[
V_{dse} = \frac{V_{ds}}{[1 + (V_{ds}/V_{sate})^M]^{1/M}}
\]

\[
V_{sate} = \alpha_{sat} V_{gte}
\]

\[
g_{ch} = \frac{g_{chi}}{1 + g_{chi}(R_s + R_D)}
\]
\[ g_{chi} = qn_s W \cdot \text{MUBAND/L} \]

\[ n_s = \frac{n_{sa} n_{sb}}{n_{sa} + n_{sb}} \]

\[ n_{sa} = \frac{\text{EPSI} \cdot V_{gte} (V_{gte})^{\text{GAMMA}}}{q \cdot \text{TOX} \cdot \left( \frac{V_{aat}}{V_{aat}} \right)} \]

\[ n_{sb} = n_{so} \left( \frac{t_m V_{gfbe} \text{EPSI}}{\text{TOX} V_0 \text{EPS}} \right)^{2 \cdot V_0 / V_e} \]

\[ n_{so} = N_c t_m V_e \frac{V_0}{V_0 - V_{th}} \exp \left( \frac{-\text{DEF0}}{V_{th}} \right) \]

\[ N_c = 3.0 \cdot 10^{25} \text{ m}^{-3} \]

\[ V_e = \frac{2 \cdot V_0 \cdot V_{tho}}{2 \cdot V_0 - V_{th}} \]

\[ t_m = \sqrt{\frac{\text{EPS}}{2q \cdot \text{GMIN}}} \]

\[ V_{gte} = \frac{V_{\text{MIN}}}{2} \left[ 1 + \frac{V_{gt}}{V_{\text{MIN}}} + \sqrt{\text{DELTA}^2 + \left( \frac{V_{gt}}{V_{\text{MIN}} - 1} \right)^2} \right] \]

\[ V_{gt} = V_{gs} - V_T \]
\[ V_{gfbe} = \frac{V_{MIN}}{2} \left[ 1 + \frac{V_{gfb}}{V_{MIN}} + \sqrt{\text{DELT}^2 + \left( \frac{V_{gfb}}{V_{MIN}} - 1 \right)^2} \right] \]

\[ V_{gfb} = V_{gs} - V_{FB} \]

\[ I_{leakage} = I_{hl} + I_{min} \]

\[ I_{hl} = I_{OL} \left[ \exp \left( \frac{V_{ds}}{V_{DSL}} \right) - 1 \right] \exp \left( \frac{V_{gs}}{V_{GSL}} \right) \exp \left[ \frac{E}{q} \left( \frac{1}{V_{tho}} - \frac{1}{V_{th}} \right) \right] \]

\[ I_{min} = \text{SIGMA}0 \cdot V_{ds} \]

**Temperature Dependence**

\[ V_{tho} = k_B \cdot \text{TNOM} / q \]

\[ V_{th} = k_B \cdot (\text{TEMP}) / q \]

\[ V_{aat} = \text{VAA} \exp \left[ \frac{\text{EMU}}{q \cdot \text{GAMMA}} \left( \frac{1}{V_{th}} - \frac{1}{V_{tho}} \right) \right] \]

\[ V_T = VTO + \text{KVT} (\text{TEMP} - \text{TNOM}) \]

\[ \alpha_{sat} = \text{ALPHASAT} + \text{KASAT} (\text{TEMP} - \text{TNOM}) \]
Capacitance

\[ C_{gs} = C_f + \frac{2}{3} C_{gc} \left[ 1 - \left( \frac{V_{sate} - V_{dse}}{2V_{sate} - V_{dse}} \right)^2 \right] \]

\[ C_{gd} = C_f + \frac{2}{3} C_{gc} \left[ 1 - \left( \frac{V_{sate}}{2V_{sate} - V_{dse}} \right)^2 \right] \]

\[ C_f = 0.5 \cdot EPS \cdot W \]

\[ C_{gs} = q \frac{dn_{sc}}{dV_{gs}} \]

\[ n_{sc} = \frac{n_{sac} n_{sbc}}{n_{sac} + n_{sbc}} \]

\[ n_{sac} = \frac{EPSI \cdot V_{gte}}{q \cdot TOX} \]

\[ n_{sbc} = n_{sb} \]
LEVEL 62 RPI Poli-Si TFT Model

Star-Hspice LEVEL 62 is an AIM-SPICE MOS16 poly-silicon (Poli-Si) thin-film transistor (TFT) model.

Model Features

The AIM-SPICE MOS16 Poli-Si TFT model features include:

- A design based on the crystalline MOSFET model
- Field effect mobility that becomes a function of gate bias
- Effective mobility that accounts for trap states:
  - For low $V_{gs}$, it is power law
  - For high $V_{gs}$, it is constant
- Reverse bias drain current function of electric field near drain and temperature
- A design independent of channel length
- A unified DC model that includes all four regimes for channel lengths down to 4 m:
  - Leakage (thermionic emission)
  - Subthreshold (diffusion-like model)
  - Above threshold (c-Si-like, with mFet)
  - Kink (impact ionization with feedback)
- An AC model that accurately reproduces $C_{ge}$ frequency dispersion
- An automatic scaling of model parameters that accurately model a wide range of device geometries
Using LEVEL 62 with Star-Hspice

When using the AIM-SPICE MOS16 Poli-Si TFT model:

1. Set LEVEL=62 to identify the model as the AIM-SPICE MOS16 Poli-Si TFT model.
2. The default value for L is 100 m, and the default value for W is 100 m.
3. The LEVEL 62 model is a 3-terminal model. No bulk node exists; therefore no parasitic drain-bulk or source-bulk diodes are appended to the model. A fourth node can be specified, but does not affect simulation results.
4. The default room temperature is 25°C in Star-Hspice, but is 27°C in some other simulators. The user may choose whether or not to set the nominal simulation temperature to 27°C by adding .OPTION TNOM=27 to the netlist.

Example

This is an example of a Star-Hspice model and element statement modified for use with LEVEL 62:

mckt drain gate source nch L=10e-6 W=10e-6

.Model nch nmos LEVEL=62
+ asat = 1 at = 3e-8 blk = 0.001 bt = 0.0 cgdo = 0.0
+ cgso = 0.0 dasat = 0.0 dd = 1.4e-7 delta = 4.0
+ dg = 2.0e-7 dmu1 = 0.0 dvt = 0.0 dvto = 0.0 eb = 0.68
+ eta = 7 etac0 = 7 etac00 = 0 i0 = 6.0 i00 = 150
+ lasat = 0lkink = 19e-6 mc = 3.0 mk = 1.3 mmu = 3.0
+ mu0 = 100 mul = 0.0022 mus = 1.0 rd = 0.0 rdx = 0.0
+ rs = 0.0 rsx = 0.0 tnom = 27 tox = 1.0e=7 vfb = -0.1
+ vkink = 9.1 von = 0.0 vto = 0.0
## LEVEL 62 Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Unit</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASAT</td>
<td>-</td>
<td>1</td>
<td>Proportionality constant of Vsat</td>
</tr>
<tr>
<td>AT</td>
<td>m/V</td>
<td>3E-8</td>
<td>DIBL parameter 1</td>
</tr>
<tr>
<td>BLK</td>
<td>-</td>
<td>0.001</td>
<td>Leakage barrier lowering constant</td>
</tr>
<tr>
<td>BT</td>
<td>m/V</td>
<td>1.9E-6</td>
<td>DIBL parameter 2</td>
</tr>
<tr>
<td>CGDO</td>
<td>F/m</td>
<td>0</td>
<td>Gate-drain overlap capacitance per meter channel width</td>
</tr>
<tr>
<td>CGSO</td>
<td>F/m</td>
<td>0</td>
<td>Gate-source overlap capacitance per meter channel width</td>
</tr>
<tr>
<td>DASAT</td>
<td>1/°C</td>
<td>0</td>
<td>Temperature coefficient of ASAT</td>
</tr>
<tr>
<td>DD</td>
<td>m</td>
<td>1400 Å</td>
<td>Vds field constant</td>
</tr>
<tr>
<td>DELTA</td>
<td>-</td>
<td>4.0</td>
<td>Transition width parameter</td>
</tr>
<tr>
<td>DG</td>
<td>m</td>
<td>2000 Å</td>
<td>Vgs field constant</td>
</tr>
<tr>
<td>DMU1</td>
<td>cm²/Vs°C</td>
<td>0</td>
<td>Temperature coefficient of MU1</td>
</tr>
<tr>
<td>DVT</td>
<td>V</td>
<td>0</td>
<td>The difference between VON and the threshold voltage</td>
</tr>
<tr>
<td>DVTO</td>
<td>V/°C</td>
<td>0</td>
<td>Temperature coefficient of VTO</td>
</tr>
<tr>
<td>EB</td>
<td>EV</td>
<td>0.68</td>
<td>Barrier height of diode</td>
</tr>
<tr>
<td>ETA</td>
<td>-</td>
<td>7</td>
<td>Subthreshold ideality factor</td>
</tr>
<tr>
<td>ETAC0</td>
<td>-</td>
<td>ETA</td>
<td>Capacitance subthreshold ideality factor at zero drain bias</td>
</tr>
<tr>
<td>ETAC00</td>
<td>1/V</td>
<td>0</td>
<td>Capacitance subthreshold coefficient of drain bias</td>
</tr>
<tr>
<td>I0</td>
<td>A/m</td>
<td>6.0</td>
<td>Leakage scaling constant</td>
</tr>
<tr>
<td>I00</td>
<td>A/m</td>
<td>150</td>
<td>Reverse diode saturation current</td>
</tr>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>---------</td>
<td>-------------------------------------------------------------------</td>
</tr>
<tr>
<td>LASA T</td>
<td>M</td>
<td>0</td>
<td>Coefficient for length dependence of ASAT</td>
</tr>
<tr>
<td>LKINK</td>
<td>M</td>
<td>19E-6</td>
<td>Kink effect constant</td>
</tr>
<tr>
<td>MC</td>
<td>-</td>
<td>3.0</td>
<td>Capacitance knee shape parameter</td>
</tr>
<tr>
<td>MK</td>
<td>-</td>
<td>1.3</td>
<td>Kink effect exponent</td>
</tr>
<tr>
<td>MMU</td>
<td>-</td>
<td>3.0</td>
<td>Low field mobility exponent</td>
</tr>
<tr>
<td>MU0</td>
<td>cm²/Vs</td>
<td>100</td>
<td>High field mobility</td>
</tr>
<tr>
<td>MU1</td>
<td>cm²/Vs</td>
<td>0.0022</td>
<td>Low field mobility parameter</td>
</tr>
<tr>
<td>MUS</td>
<td>cm²/Vs</td>
<td>1.0</td>
<td>Subthreshold mobility</td>
</tr>
<tr>
<td>RD</td>
<td>μ</td>
<td>0</td>
<td>Drain resistance</td>
</tr>
<tr>
<td>RDX</td>
<td>Ω</td>
<td>0</td>
<td>Resistance in series with C&lt;sub&gt;gd&lt;/sub&gt;</td>
</tr>
<tr>
<td>RS</td>
<td>μ</td>
<td>0</td>
<td>Source resistance</td>
</tr>
<tr>
<td>RSX</td>
<td>Ω</td>
<td>0</td>
<td>Resistance in series with C&lt;sub&gt;gs&lt;/sub&gt;</td>
</tr>
<tr>
<td>TNOM</td>
<td>°C</td>
<td>25</td>
<td>Parameter measurement temperature</td>
</tr>
<tr>
<td>TOX</td>
<td>m</td>
<td>1e-7</td>
<td>Thin-oxide thickness</td>
</tr>
<tr>
<td>V0</td>
<td>V</td>
<td>0.12</td>
<td>Characteristic voltage for deep states</td>
</tr>
<tr>
<td>VFB</td>
<td>V</td>
<td>-0.1</td>
<td>Flat band voltage</td>
</tr>
<tr>
<td>VKINK</td>
<td>V</td>
<td>9.1</td>
<td>Kink effect voltage</td>
</tr>
<tr>
<td>VON</td>
<td>V</td>
<td>0</td>
<td>On-voltage</td>
</tr>
<tr>
<td>VTO</td>
<td>V</td>
<td>0</td>
<td>Zero-bias threshold voltage</td>
</tr>
</tbody>
</table>
Equivalent Circuit

Model Equations

Drain Current

The expression for the subthreshold current is given by:

\[ I_{sub} = MUS \cdot C_{ox} \frac{W}{L} V_{th}^2 V_{th} \exp \left( \frac{V_{GT}}{V_{th}} \right) \left[ 1 - \exp \left( -\frac{V_{DS}}{V_{th}} \right) \right] \]

\[ V_{sth} = ETA \cdot V_{th}, V_{th} = k_B \cdot TEMP / q \]

\[ C_{ox} = \varepsilon_i \cdot L \cdot W/TOX \]

\[ V_{GT} = V_{GS} - V_T \]

\[ V_T = V_{TX} - \frac{AT \cdot V_{DS}^2 + BT}{L} \]
where $\varepsilon_1$ is the dielectric constant of the oxide and $k_B$ is the Boltzmann constant.

Above threshold ($V_{GT} > 0$), the conduction current is given by:

$$I_a = \begin{cases} 
\mu_{FET} C_{ox} \frac{W}{L} \left( V_{GT} V_{DS} - \frac{V_{DS}^2}{2\alpha_{sat}} \right) & \text{for } V_{DS} < \alpha_{sat} V_{GT} \\
\mu_{FET} C_{ox} \frac{W}{L} \frac{V_{GT}^2 \alpha_{sat}}{2} & \text{for } V_{DS} \geq \alpha_{sat} V_{GT}
\end{cases}$$

$$\frac{1}{\mu_{FET}} = \frac{1}{\mu_{Muo}} + \frac{1}{\mu_1 (2V_{GT}/V_{sth})^{MMU}}$$

Subthreshold leakage current is the result of thermionic field emission of carriers through the grain boundary trap states and is described by:

$$I_{leak} = 10 \cdot W \left[ \exp \left( \frac{q \cdot BLK \cdot V_{DS}}{kT} \right) - 1 \right] \left[ X_{TFE} + X_{TE} \right] + I_{diode}$$

$$X_{TFE} = \frac{X_{TFE,lo} X_{TFE,hi}}{X_{TFE,lo} + X_{TFE,hi}}$$

$$X_{TE} = \exp(-W_C)$$

$$W_C = (E_c - E_t)/kT = 0.55 \text{eV}/kT$$
Selecting MOSFET Models: LEVEL 47-62

LEVEL 62 RPI Poli-Si TFT Model

\[
X_{TEF,lo} = \begin{cases} 
\frac{4\sqrt{\pi}}{3} f \exp\left(\frac{4}{27} f^2 - W_C\right) & \text{for } f \leq f_{lo} \\
X_{TFE,lo}(f_{lo}) \exp\left\{\left(\frac{1}{f_{lo}} + \frac{8}{27} f_{lo}\right)(f - f_{lo})\right\} & \text{for } f > f_{lo}
\end{cases}
\]

\[
f = \frac{F}{F_0}
\]

\[
F = \begin{bmatrix} V_{DS} \\ DD \\ -V_{GS} - V_{FB} \\ DG \end{bmatrix}
\]

\[
F_0 = (kT)^{3/2} \frac{4\pi \sqrt{2m^*}}{3} \frac{1}{qh}
\]

\[m^* = 0.27m_0\]

\[
f_{lo} = \frac{3}{2} \left(\sqrt{W_C + 1} - 1\right)
\]

\[
X_{TEF,hi} = \begin{cases} 
\frac{2W_C}{3} \exp\left(1 - \frac{2W_C}{3}\right) & \text{for } f < f_{hi} \\
\left(1 - \frac{3W_C}{2f}\right)^{-1} \exp\left(\frac{-W_C^{3/2}}{f}\right) & \text{for } f \geq f_{hi}
\end{cases}
\]

\[
f_{hi} = 3 \left(\frac{W_C^{3/2}}{2W_C - 3}\right)
\]
Finally, for very large drain biases, the kink effect is observed. It is modeled as impact ionization in a narrow region near the drain. The expression can be written as:

$$I_{kink} = A_{kink} I_a (V_{DS} - V_{DSAT}) \exp\left(-\frac{V_{KINK}}{V_{DS} - V_{DSAT}}\right)$$

$$A_{kink} = \frac{1}{V_{KINK}} \left(\frac{L_{KINK}}{L}\right)^{MK} V_{DSAT} = \alpha_{sat} V_{GT}$$

The impact ionization current, $I_{kink}$, is added to the drain current.

**Threshold Voltage**

If $VTO$ is not specified:

$$V_T = V_{ON} - DVT$$

else:

$$V_T = VTO$$

**Temperature Dependence**

$$V_{TX} = V_T - DVTO(TEMP - TNOM)$$

$$\mu_1 = MU1 + DMU1(TEMP - TNOM)$$

$$\alpha_{sat} = ASAT - \frac{L_{ASAT}}{L} - DASAT(TEMP - TNOM)$$
Capacitance

\[ C_{gs} = C_f + \frac{2}{3} C_{gcd} \left[ 1 - \left( \frac{V_{DSAT} - V_{DSE}}{2V_{DSAT} - V_{DSE}} \right)^2 \right] \]

\[ C_{gd} = C_f + \frac{2}{3} C_{gcs} \left[ 1 - \left( \frac{V_{DSAT}}{2V_{DSAT} - V_{DSE}} \right)^2 \right] \]

\[ C_f = 0.5 \cdot \text{EPS} \cdot W \]

\[ C_{gcd} = \frac{C_{ox}}{1 + \eta_{cd} \exp\left(-\frac{V_{GT}}{\eta_{cd} V_{th}}\right)} \]

\[ C_{gcs} = \frac{C_{ox}}{1 + \text{ETAC0} \cdot \exp\left(-\frac{V_{GT}}{\text{ETAC0} \cdot V_{th}}\right)} \]

\[ C_{ox} = W \cdot L \cdot \epsilon_i / \text{TOX}, \eta_{cd} = \text{ETAC0} + \text{ETAC00} \cdot V_{DSE} \]

\[ V_{DSE} = \frac{V_{DS}}{\left[1 + (V_{DS}/V_{DSAT})^{MC}\right]^{1/MC}} \]
Chapter 23

Using the Common Model Interface

Avant!’s Common Model Interface (CMI) is a program interface for adding proprietary models into the Star-Hspice simulator.

This chapter covers the following topics:

■ Understanding CMI
■ Examining the Directory Structure
■ Running Simulations with CMI Models
■ Supported Platforms
■ Adding Proprietary MOS Models
■ Testing CMI Models
■ Model Interface Routines
■ Interface Variables
■ Internal Routines
■ Supporting Extended Topology
■ Conventions
Understanding CMI

Star-Hspice uses a dynamically linked shared library to integrate models with CMI. Add the global option `cmiflag`. It loads the dynamically linked CMI library, `libCMImodel`. Star-Hspice first searches for the shared library `libCMImodel` in the path `$hspice_lib_models`. If it does not find the library, it then searches the path in `$installdir/$ARCH/lib/models`.

Dynamic loading results in better resource sharing than static binding. While several Star-Hspice jobs are running concurrently, only one copy of the dynamically linked CMI model is needed in memory, which will speed up the process. Theoretically, the static linking version will always be slightly faster when only one Star-Hspice job is running at a time. However, the performance difference between dynamic loading and static binding is small, usually less than 5 percent.

CMI is released with several source code examples for integration of MOS models in Star-Hspice. They are standard Berkeley SPICE MOSFET models (LEVEL 1, 2, 3, BSIM1, 2, 3). To minimize the effort required for adding models, Avant! provides installation scripts to automate the shared library generation process.

If your proprietary models are derived from SPICE models, the integration process is similar to the examples with minimal modifications.

**Note:** The actual Star-Hspice program contains device equations and programs for bias calculation, numerical integration, convergence checking, and matrix loading. These programs are not needed to complete a new model integration and are excluded from the source code examples.
Examining the Directory Structure

The CMI distribution is structured as shown in the following diagram. You must modify the shaded files or add new ones for new models.

- **HSPCMI** Subdirectory that contains utility for processing configuration file and makefile
- **get_arch** C shell script for identifying platforms
- **config** Configuration file
- **doc/** CMI documentation
- **link/** Main CMI routines
- **include/** CMI header files
- **makecmi** Master makefile
- **test/** Model testing example
- **lib** Shared library directory
- **obj** Object code
- **mos1/**, **mos2/**, **mos3/**, **b1/b2/b3/** Model directories
**Running Simulations with CMI Models**

CMI models are specified using model parameter `level`. Levels used by the example models are (same as those in Berkeley Spice-3):

- **LEVEL 1 (mos1/)** LEVEL 1 MOS model
- **LEVEL 2 (mos2/)** LEVEL 2 MOS model
- **LEVEL 3 (mos3/)** LEVEL 3 MOS model
- **LEVEL 4 (b1/)** BSIM model
- **LEVEL 5 (b2/)** BSIM2 model
- **LEVEL 8 (b3/)** BSIM3v3 model

To perform a simulation run on a CMI model, add the following line in the input netlist.

```
.option cmiflag
```

The LEVEL 8 example code located in the `b3` directory and Star-Hspice LEVEL 49 are both based on BSIM3, version 3. However, the speed of LEVEL 8 is sometimes 20 percent slower than LEVEL 49 in Star-Hspice. This occurs because LEVEL 49 in Star-Hspice is carefully implemented to ensure high accuracy and performance. In contrast, LEVEL 8 in the example code is created only as an example of CMI interface implementation. Therefore, the slower performance of the example code compared to Star-Hspice LEVEL 49 is expected.
Supported Platforms

CMI supports the platforms:

- Sun SPARC: SunOS 4.1.3, Solaris 5.5X
- HP-7X: HP-UX 10.20
Adding Proprietary MOS Models

The CMI interface enables you to enter proprietary models into Star-Hspice. This section describes how to use CMI to add a new MOS model. Use CMI to simplify the integration process.

Note: In the following examples, the percent sign (%) represents the UNIX shell prompt, and $(installdir) points to the Star-Hspice release directory. $ARCH is the OS type for the computer. Star-Hspice 98.4 CMI release supports the platforms Sun4, Solaris, and HP.

Creating a CMI Shared Library

To add a new model:

1. Create a directory environment and modify the configuration file.
2. Prepare and modify model routines.
3. Compile the shared library.
4. Set up the runtime shared library path.

Creating the Directory Environment

To create the CMI directory environment:

1. Copy the CMI directory from the Star-Hspice release directory to a new location, as shown in the following example:

   % cp -r $(installdir)/cmi /home/user1/userx/model

   The new CMI directory /home/user1/userx/model/cmi is your working directory. Make sure that you have read and write access to your working directory.
2. Copy an existing model subdirectory to a new model directory and create a subdirectory for the new model under the working CMI directory. For example, if you have a MOSFET model whose LEVEL is 222, you can copy the subdirectory from the existing MOS model LEVEL 3, as follows:

```bash
%cp -r mos3 mos222
```

3. Add the following line in the configuration file `config`:

```bash
mos222    222    "my own MOSFET model"
```

where `mos222` is the model name

`222` is the model LEVEL;

"my own MOSFET model" is the descriptive comment for the model. The model name and level must be unique in the configuration file. For more information, see the in-line comment in the configuration file.

### Preparing Model Routine Files

In the new model subdirectory `mos222`, rename `mos3` in all filenames to `mos222`, for example:

```bash
% mv CMImos3defs.h CMImos222defs.h
```

After renaming all the files, the new model subdirectory should contain the following group of files:

- `CMImos222defs.h`
- `CMImos222.c`
- `CMImos222GetIpar.c`
- `CMImos222SetIpar.c`
- `CMImos222GetMpar.c`
- `CMImos222SetMpar.c`
- `CMImos222eval.c`
- `CMImos222set.c`
- `CMImos222temp.c`

The purpose and detailed description of each routine can be found in “Model Interface Routines” on page 23-13. You must modify the functions as necessary. The majority of the work required to add a new model is for modification of these files.
Compiling the Shared Library

Follow the steps above to modify the model routine files and the configuration file, then compile the model routines and the shared library with a single `make` operation. Prior to the `make` operation, manually set the environment variable `HSPICE_CMI` to the working CMI directory (see “Creating the Directory Environment” on page 23-6):

```
% setenv HSPICE_CMI /home/user1/userx/model/cmi
```

With `HSPICE_CMI` set correctly, invoke the compilation process by entering the following:

```
% make -f makecmi
```

The new shared library called `libCMImodel` will be created in subdirectory `lib/` with all the object files generated in subdirectory `obj/`.

We recommend that you check the syntax of your C functions before launching the compilation process. Enter the following command:

```
% make -f makecmi lint
```

to list any syntax errors in your model routines.

---

**Note:** During compilation, CMI creates files (`makefile.SUN` on SUN, `makefile.HP` on HP), and subdirectories (`obj/`, `lib/`) in the CMI working directory. Do not manually modify these generated files.

---

Choosing a Compiler

You can use any functional compiler by properly setting the environment variable `CC` to the location of the compiler (`CC` is used in the autogenerated makefile, `makefile.SUN` and `makefile.HP`). Make sure that the appropriate compiler and link flags are properly set so the final CMI library build is in position independent code (PIC). PIC is needed for dynamic linking.
For SUNOS 5.4 platforms or later, the automatically generated compiler flag
-KPIC and link flag -G -z are for the Sun workshop compiler, cc or acc. These
compilers are typically installed in a directory such as /usr1/opt/SUNWspro/
SC4.2/bin. For HP9000/700 platforms, cc is installed in /opt/ansic/bin. For more
information, type man cc or man acc to display the on-line manual page for
these commands.

There are no restrictions on the optimization flags you use for the CMI library.
However, we recommend using the flag -fast for the Sun compiler cc or acc and
the flag -O for HP compiler cc.

Using the gcc Compiler

If you use the gcc compiler, modify the makefile (makefile.SUN or makefile.HP)
to set the compiler flags correctly.

For gcc, set the environment variable CC to gcc and modify the makefile to use
the -fPIC flag for compiling and the -r flag for linking. For example:

    gcc -c  -I../include -fPIC CMImain.c
    ...
    gcc -r  -o ./lib/libCMImodel obj/*.o

Using the /usr/ucb/cc Compiler

If you use the /usr/ucb/cc compiler, modify the makefile (makefile.SUN or
makefile.HP) to set the compiler flags correctly.

The /usr/ucb/cc compiler can not compile C source files until the “Language
Optional Source Package” is installed. Verify that this source package is
installed, then set the environment variable CC to /usr/ucb/cc.

Note: The “Language Optional Source Package” is not installed in Solaris by
default, but it is installed in SUNOS 4.1.x by default. You must either
install the optional language software or use a workable compiler such
as the Sun workshop cc or acc. The gcc compiler can also be used with
some minor modifications to the makefile.
Setting up the Runtime Shared Library Path

The shared library is now ready for use. You must update the shared model search path defined by the environment variable `hspice_lib_models` so that the system dynamic loader can find the new CMI shared library for Star-Hspice. Enter the following:

```
setenv hspice_lib_models $HSPICE_CMI/lib
```

Troubleshooting

Sometimes you can successfully build the CMI dynamic library, but at run time Star-Hspice returns the following error message:

```
**error**: Unable to load 
/home/ant/lib/models/libCMImodel and /home/ant/lib/models/libCMImodel.so
```

This problem is usually caused by undefined symbols at run time. The following is an example output of undefined symbols, using Sun workshop `cc` as a compiler on a SUNOS 5.5 machine (note that different compilers usually generate nm output in different formats)

**Example**

```
nm libCMImodel | grep UNDEF

[ 35]  | 0 | 0 | NOTY | LOCL | 0 | UNDEF |
[ 34]  | 0 | 0 | NOTY | LOCL | 0 | UNDEF |
[1779] | 0 | 0 | NOTY | GLOB | 0 | UNDEF |.mul
[1853] | 0 | 0 | NOTY | GLOB | 0 | UNDEF |__dtou
[1810] | 0 | 0 | NOTY | GLOB | 0 | UNDEF |__iob
[1822] | 0 | 0 | NOTY | WEAK | 0 | UNDEF |__ex_deregister
[1749] | 0 | 0 | NOTY | WEAK | 0 | UNDEF |__ex_register
[1857] | 0 | 0 | NOTY | GLOB | 0 | UNDEF | atan
[1878] | 0 | 0 | NOTY | GLOB | 0 | UNDEF | cos
[1820] | 0 | 0 | NOTY | GLOB | 0 | UNDEF | exp
[1777] | 0 | 0 | NOTY | GLOB | 0 | UNDEF | fabs
[1872] | 0 | 0 | NOTY | GLOB | 0 | UNDEF | fprintf
[1764] | 0 | 0 | NOTY | GLOB | 0 | UNDEF | log
[1767] | 0 | 0 | NOTY | GLOB | 0 | UNDEF | malloc
[1842] | 0 | 0 | NOTY | GLOB | 0 | UNDEF | memset
[1772] | 0 | 0 | NOTY | GLOB | 0 | UNDEF | pow
[1869] | 0 | 0 | NOTY | GLOB | 0 | UNDEF | sin
```
The above undefined symbols can be satisfied at run time by libraries such as libc or libm. However, any unsatisfied symbols other than those shown in the example may cause the “Unable to load” problem.
Testing CMI Models

After creating the shared library, you may test the new model by running Star-Hspice on a sample input file `mos3.sp` under the subdirectory `test`. This file contains a simple CMOS inverter using MOS LEVEL-3 models. Modify transistor sizes and model cards as necessary.

```
%hspice mos3.sp >mos3.lis
```

Avanwaves can now be used to inspect the I-V and C-V characteristics at different biasing conditions.

Use AvanWaves to carefully check the following aspects:

- Sign and value of channel current (`ids`)
- The monotonicity of channel current versus `vgs` and `vds`
- Sign and value of capacitance (`cgs, cgs, cgb, csb, cdb`)

Refer to the [AvanWaves User Guide](#) for more information.

To verify the CMI integration of your new model, run a DC sweep analysis and transient analysis on the test netlist.

**Note:** LEVELs from 100 to 200 are reserved for CMI customer models. Please choose levels from this range so there will be no conflicts with existing Star-Hspice model levels. Also, please add a special prefix or suffix for some of the auxiliary functions used in CMI, especially those from the public domain, such as the function called `modchk` or `dc3p1` from Berkeley Spice3. This will ensure that the function names are different from those used in the Star-Hspice core code.

After testing, if you are satisfied with your CMI library, put it in the default CMI library directory, `$installdir/$ARCH/lib/models`, where `$ARCH` is `sun4`, `sol4`, or `pa`, depending on the platform on which you compiled your CMI library.

The model interface routines accept input parameters from CMI. For each set of input conditions, the model routines are required to return transistor characteristics to CMI.
Model Interface Routines

Model interface routines accept input parameters from CMI. These input parameters include the following:

- Circuit and nominal model temperatures (*CKTtemp, CKTnomtemp*)
- Input biases (*vds, vgs, vbs*)
- Model parameters (*level, vto, tox, uo ...*)
- Instance parameters (*w, l, as, ad ...*)
- Mode of transistor (*mode*, 1 for normal, -1 for reverse)
- AC frequency (*freq*, this is passed from simulator to model code)
- Transient simulation integration order (*intorder*) (Star-Hspice returns the following codes: 0 - Trapezoidal, 1 - 1st order Gear, 2 - 2nd order Gear)
- Transient time step (*timestep*)
- Transient time point (*timepoint*)

For each set of input conditions, the model routines are required to return the following transistor characteristics to CMI:

- Charge and capacitance computation flag (1 for computation, 0 for no computation, *qflag*)
- Charge, capacitance model selector (0 for Meyer capacitance model*, 13 for charge-based model, *capop*)
- Channel current (*ids*)
- Channel conductance (*gds*)
- Transconductance (*gm*)
- Substrate transconductance (*gmbs*)
- Turn-on voltage (*von*)
- Saturation voltage (*vsat*)
- Gate overlap capacitances (*cgso, cgdo, cgbo*)
- Intrinsic MOSFET charges (*qg, qd, qs*)
- Intrinsic MOSFET capacitances referenced to bulk (*cggb, cgdb, cgsb, cbgb, cbdb, cbsb, cdgb, cddb, cdsb*)
- Parasitic source & drain conductances ($g_s, g_d$)
- Substrate diode current ($ib_d, ib_s$)
- Substrate diode conductance ($g_{bd}, g_{bs}$)
- Substrate diode charge ($q_{bd}, q_{bs}$)
- Substrate diode junction capacitance ($cap_{bd}, cap_{bs}$)
- Substrate impact ionization current ($isub$)
- Substrate impact ionization transconductances ($gb_{gs}=dI_{sub}/dV_{gs}$, $gb_{ds}=dI_{sub}/dV_{ds}$, $gb_{bs}=dI_{sub}/dV_{bs}$)
- Source resistance noise current squared ($nois_{irs}$)
- Drain resistance noise current squared ($nois_{ird}$)
- Thermal or Shot channel noise current squared ($nois_{idsth}$)
- Source resistance noise current squared ($nois_{idsfl}$)

**Note:** Currently, the Meyer capacitance model is not supported in Star-Hspice.

The transistor biases and output characteristics are transferred between CMI and model interface routines using variable type `CMI_VAR`, which can be found in the `include/CMIdef.h` file.

The entries $v_{ds}$, $v_{gs}$ and $v_{bs}$ are used to provide bias conditions, while the rest of the entries carry the results from evaluating the model equations.

```c
/* must be consistent with its counterpart in HSPICE */
typedef struct CMI_var {
  int    mode;       /* device mode */
  int    qflag;      /* flag for charge/cap computing */
  double vds;        /* vds bias */
  double vgs;        /* vgs bias */
  double vbs;        /* vbs bias */
```
/* device DC information */
double gd;   /* drain conductance */
double gs;   /* source conductance */
double cgso; /* gate-source overlap capacitance */
double cgdo; /* gate-drain overlap capacitance */
double cgbo; /* gate-bulk overlap capacitance */
double von;  /* turn-on voltage */
double vdsat; /* saturation voltage */
double ids;  /* drain dc current */
double gds;  /* output conductance (dIds/dVds) */
double gm;   /* trans-conductance (dIds/dVgs) */
double gmbs; /* substrate trans-conductance (dIds/dVbs) */

/* MOSFET capacitance model selection */
/* capop can have following values */
* 13 charge model
* 0 or else Meyer's model

Note: Currently, Meyer’s model is not supported.

*/
int capop;   /* capacitor selector */

/* Meyer's capacitances: intrinsic capacitance + overlap capacitance Note: currently, these 3 capacitances are ignored by Hspice. A charge-based model formulation is required. */
double capgs; /* Meyer's gate capacitance (dQg/dVgs + cgso) */
double capgd; /* Meyer's gate capacitance (dQg/dVds + cgdo) */
double capgb; /* Meyer's gate capacitance (dQg/dVbs + cgbo) */
/* substrate-junction information */
double ibs; /* substrate source-junction leakage current */
double ibd; /* substrate drain-junction leakage current */
double gbs; /* substrate source junction-conductance */
double gbd; /* substrate drain junction-conductance */
double capbs; /* substrate source-junction capacitance */
double capbd; /* substrate drain-junction capacitance */
double qbs; /* substrate source-junction charge */
double qbd; /* substrate drain-junction charge */

/* substrate impact ionization current */
double isub; /* substrate current */
double gbgs; /* substrate trans-conductance (dIsub/dVgs) */
double gbds; /* substrate trans-conductance (dIsub/dVds) */
double gbbs; /* substrate trans-conductance (dIsub/dVbs) */

/* charge-based model intrinsic terminal charges */
/* NOTE: these are intrinsic charges ONLY */
double qg;    /* gate charge */
double qd;    /* drain charge */
double qs;    /* source charge */

/* charge-based model intrinsic trans-capacitances*/
/* NOTE: these are intrinsic capacitances ONLY */
double cggb;
double cgdb;
double cgsb;
double cbgb;
double cbdb;
double cbsb;
double cdgb;
double cddb;
double cdsb;

/* noise parameters */
double nois_irs; /* Source noise current^2 */
double nois_ird; /* Drain noise current^2 */
double nois_idsth; /* channel thermal or shot noise current^2 */
double nois_idsfl; /* 1/f channel noise current^2 */
double freq; /* ac frequency */

/* extended model topology */
char *topovar; /* topology variables */
} CMI_VAR;

The nominal temperature and device temperature can be found in the global variable \texttt{CMIenv}. The \texttt{CMIenv} structure can be accessed via the global variable \texttt{pCMIenv} (pointer to the global \texttt{CMIenv} struct). The structure for \texttt{CMIenv} is defined in type \texttt{CMI_ENV}, which can be found in the \texttt{include/CMIdef.h} file:

/* environment variables */
typedef struct CMI_env {
double CKTtemp; /* simulation temperature */
double CKTnomTemp; /* nominal temperature */
double CKTgmin; /* GMIN for the circuit */
int CKTtempGiven; /* temp setting flag */
/* following are hspice-specific options */
double aspec;
double spice;
double scalm;
} CMI_ENV;
Interface Variables

To assign model/instance parameter values and evaluate I-V, C-V response, fifteen interface routines are required. For each new model, pointers to these routines and the model/instance variables are defined by an interface variable in type `CMI_MOSDEF`, which can be found in the `include/CMIdef.h` file.

```c
typedef struct CMI_MosDef {
    char  ModelName[100];
    char  InstanceName[100];
    char  *pModel;
    char  *pInstance;
    int  modelSize;
    int  instSize;
    int  (*CMI_ResetModel)(char*,int,int);
    int  (*CMI_ResetInstance)(char*);
    int  (*CMI_AssignModelParm)(char*,char*,double);
    int  (*CMI_AssignInstanceParm)(char*,char*,double);
    int  (*CMI_SetupModel)(char*);
    int  (*CMI_SetupInstance)(char*,char*);
    int  (*CMI_Evaluate)(CMI_VAR*,char*,char*);
    int  (*CMI_DiodeEval)(CMI_VAR*,char*,char*);
    int  (*CMI_Noise)(CMI_VAR *,char*,char*);
    int  (*CMI_PrintModel)(char*);
    int  (*CMI_FreeModel)(char*);
    int  (*CMI_FreeInstance)(char*,char*);
    int  (*CMI_WriteError)(int, char*);
    int  (*CMI_Start)(void);
    int  (*CMI_Conclude)(void);
} CMI_MOSDEF;
```

/* extended model topology, 0 is normal mos, 1 is berkeley SOI, etc. */
int  topoid;
```
All routines return 0 on success, or nonzero integer (a user-defined error code) in case of warning or error. The following sections describe each entry. Examples for the first seven functions are extracted from the MOS3 implementation. Examples for the remaining eight functions are not part of the actual MOS3 code, but are included for demonstration. The example MOS3 implementation contains one header file and eight C files. All routines are derived from SPICE-3 code.

**pModel, pInstance**

Structure entries of the interface variable are initialized at compile time.

**Example**

```c
/* function declaration */
int CMImos3ResetModel(char*,int,int);
int CMImos3ResetInstance(char*);
int CMImos3AssignMP(char*,char*,double);
int CMImos3AssignIP(char*,char*,double);
int CMImos3SetupModel(char*);
int CMImos3SetupInstance(char*,char*);
int CMImos3Evaluate(CMI_VAR*,char*,char*);
int CMImos3DiodeEval(CMI_VAR*,char*,char*);
int CMImos3Noise(CMI_VAR *,char*,char*);
int CMImos3PrintModel(char*);
int CMImos3FreeModel(char*);
int CMImos3FreeInstance(char*,char*);
int CMImos3WriteError(int, char*);
int CMImos3Start(void);
int CMImos3Conclude(void);

/* extended model topology, 0 is normal mos, 1 is berkeley SOI, etc. */

/* local */
static MOS3model    _Mos3Model;
static MOS3instance _Mos3Instance;
```
static CMI_MOSDEF CMI_mos3def = {
    (char*)&_Mos3Model,
    (char*)&_Mos3Instance,
    CMImos3ResetModel,
    CMImos3ResetInstance,
    CMImos3AssignMP,
    CMImos3AssignIP,
    CMImos3SetupModel,
    CMImos3SetupInstance,
    CMImos3Evaluate,
    CMImos3DiodeEval,
    CMImos3Noise,
    CMImos3PrintModel,
    CMImos3FreeModel,
    CMImos3FreeInstance,
    CMImos3,
    CMImos3Start,
    CMImos3Conclude
};
/* export */
CMI_MOSDEF *pCMI_mos3def = &CMI_mos3def;
*Note: the last 8 functions are optional. Any function not
defined should be replaced with NULL.

**CMI_ResetModel**

This routine initializes all the parameters of a model. All model parameters
should become undefined after initialization. Undefined means "not given in a
netlist model card". The flag *pmos* is used to set transistor type after
initialization.

```
int CMI_ResetModel(char* pmodel, int pmos, int level)
```
Example

```c
int
#ifdef __STDC__
CMImos3ResetModel(
    char *pmodel,
    int   pmos)
#else
CMImos3ResetModel(pmodel, pmos)
#endif
char *pmodel;
int   pmos;
#endif
{
/* reset all flags to undefined */
    (void)memset(pmodel, 0, sizeof(MOS3model));
/* Note: level contains model level value passed from parser */
    if(pmos)
        {
        ((MOS3model*)pmodel)->MOS3type = PMOS;
        ((MOS3model*)pmodel)->MOS3typeGiven = 1;
        }
    return 0;
} /* int CMImos3ResetModel() */
```
CMI_ResetInstance

This routine initializes all parameter settings of an instance. All instance parameters should be undefined after initialization. "Undefined" means "not given in a netlist MOS instance".

```c
int CMI_ResetInstance(char* pinst)
{
    pinst Pointer to the instance
```

Example

```c
int
#endif __STDC__
CMImos3ResetInstance(
    char *ptran)
#else
CMImos3ResetInstance(ptran)
    char *ptran;
#endif
{
    (void)memset(ptran, 0, sizeof(MOS3instance));

    ((MOS3instance*)ptran)->MOS3w = 1.0e-4;
    ((MOS3instance*)ptran)->MOS3l = 1.0e-4;

    return 0;
} /* int CMImos3ResetInstance() */
```

CMI_AssignModelParm

This routine sets the value of a model parameter.

```c
int CMI_AssignModelParm(char* pmodel, char* pname,double value)
    pmodel Pointer to the model instance
    pname String of parameter name
    value Parameter value
```
Example

```c
int
#ifdef __STDC__
CMImos3AssignMP(
    char   *pmodel,
    char   *pname,
    double  value)
#else
CMImos3AssignMP(pmodel,pname,value)
#endif
{  
    int param;

    CMImos3GetMpar(pname, &param);
    CMImos3SetMpar(param, value, (MOS3model*)pmodel);

    return 0;
}
/* int CMImos3AssignMP() */
```

**CMI_AssignInstanceParm**

This routine sets the value of an instance parameter.

```c
int CMI_AssignInstanceParm(char *pinst,char* pname,double value)
    pinst       Pointer to the instance
    pname       String of parameter name
    value       Parameter value
```
**Example**

```c
int
#ifdef __STDC__
CMImos3AssignIP(
    char *ptran,
    char *pname,
    double value)
#else
CMImos3AssignIP(ptran,pname,value)
#endif
char *ptran;
char *pname;
double value;
#endif
{
    int param;

    CMImos3GetIpar(pname, &param);
    CMImos3SetIpar(param, value, (MOS3instance*)ptran);

    return 0;
} /* int CMImos3AssignIP() */
```

**CMI_SetupModel**

This routine sets up a model after all the model parameters are specified.

```c
int CMI_SetupModel(char* pmodel)
    pmodel Pointer to the model
```

**Example**

```c
int
#ifdef __STDC__
CMImos3SetupModel(
```

Using the Common Model Interface

Interface Variables

```c
char *pmodel)
#else
CMImos3SetupModel(pmodel)
char *pmodel;
#endif
{
CMImos3setupModel((MOS3model*)pmodel);

return 0;
} /* int CMImos3SetupModel() */

CMI_SetupInstance

This routine sets up an instance after all the instance parameters are specified. Typically temperature and geometry processing are performed here.

```c
int CMI_SetupInstance(char* pinst)
pinst Pointer to the instance
```

Example

```c
int
#endif __STDC__
CMImos3SetupInstance(
char    *pmodel,
char    *ptran)
#else
CMImos3SetupInstance(pmodel,ptran)
char    *pmodel;
char    *ptran;
#endif
{
/* temperature modified parameters */
CMImos3temp((MOS3model*)pmodel,(MOS3instance*)ptran);
```
return 0;
} /* int CMImos3SetupInstance() */

**CMI_Evaluate**

Based on the bias conditions and model/instance parameter values, this routine evaluates the model equations and passes all transistor characteristics via the `CMI_VAR` variable.

```c
int CMI_Evaluate(CMI_VAR *pvar, char *pmodel, char *pinst)
```

- **pvar**  Pointer to `CMI_VAR` variable
- **pmodel** Pointer to the model
- **pinst**  Pointer to the instance

**Example**

```c
int
#ifdef __STDC__
CMImos3Evaluate(CMI_VAR *pslot, char *pmodel, char *ptr)
#else
CMImos3Evaluate(pslot, pmodel, ptr)
#endif
{
    CMI_ENV *penv;
    MOS3instance *ptran;
    penv = pCMIenv; /* pCMIenv is a global */
    ptran = (MOS3instance*)ptr;
```
/ * call model evaluation */
(void)CMImos3evaluate(penv,(MOS3model*)pmodel,ptran,
 pslot->vgs,pslot->vds,pslot->vbs);

pslot->gd = ptran->MOS3drainConductance;
pslot->gs = ptran->MOS3sourceConductance;
pslot->von = ptran->MOS3von;
pslot->ids = ptran->MOS3cd;
pslot->gds = ptran->MOS3gds;
pslot->gm = ptran->MOS3gm;
pslot->gmbs = ptran->MOS3gmbs;
pslot->gbd = ptran->MOS3gbd;
pslot->gbs = ptran->MOS3gbs;
pslot->cgs = ptran->MOS3capgs;
pslot->cgd = ptran->MOS3capgd;
pslot->cgb = ptran->MOS3capgb;
pslot->capdb = ptran->MOS3capbd;
pslot->capsb = ptran->MOS3capbs;
pslot->cbso = ptran->MOS3cbs;
pslot->cbdo = ptran->MOS3cbd;
...Additional CMI_VAR elements should be assigned here for
substrate model and overlap capacitances.

return 0;
} /* int CMImos3Evaluate() */

CMI_DiodeEval

Based on the bias conditions and model/instance parameter values, this routine
evaluates the MOS junction diode model equations and passes all transistor
characteristics via the CMI_VAR variable.

int CMI_DiodeEval(CMI_VAR *pvar ,char *pmodel. char *pinst)
Interface Variables

Using the Common Model Interface

Example

```c
int
#ifdef __STDC__
CMImos3DiodeEval(
    CMI_VAR *pslot,
    char    *pmodel,
    char    *ptr)
#else
CMImos3Diode(pslot,pmodel,ptr)
#endif
{
    CMI_ENV      *penv;
    MOS3instance *ptran;
    penv = pCMIenv; /* pCMIenv is global */
    ptran = (MOS3instance*)ptr;
    /* call model evaluation */
    (void)CMImos3diode(penv,(MOS3model*)pmodel,ptran,
        pslot->vgs,pslot->vds,pslot->vbs);

    pslot->ibs = ptran->MOS3ibs;
    pslot->ibd = ptran->MOS3ibd;
    pslot->gbs = ptran->MOS3gbs;
    pslot->gbd = ptran->MOS3gbd;
```

pvar  Pointer to CMI_VAR variable
pmodel  Pointer to the model
pinst  Pointer to the instance
Using the Common Model Interface

Interface Variables

```c
pslot->capbs = ptran->MOS3capbs;
pslot->capbd = ptran->MOS3capbd;
pslot->qbs = ptran->MOS3qbs;
pslot->qdb = ptran->MOS3qbd;
return 0;
} /* int CMImos3DiodeEval() */
```

**CMI_Noise**

Based on the bias conditions, temperature, and model/instance parameter values, this routine evaluates the noise model equations and returns noise characteristics via the *CMI_VAR* variable.

The value passed to `pslot->nois_irs` should be the thermal noise associated with the parasitic source resistance expressed as a mean square noise current in parallel with Rs.

The value passed to `pslot->nois_ird` should be the thermal noise associated with the parasitic drain resistance expressed as a mean square noise current in parallel with Rd.

The value passed to `pslot->nois_idsth` should be the thermal noise associated with the MOSFET expressed as a mean square noise current referenced across the MOSFET channel.

The value passed to `pslot->nois_idsfl` should be the flicker noise associated with the MOSFET expressed as a mean square noise current referenced across the MOSFET channel. Frequency is passed into CMI_Noise via `pslot->freq`.

```c
int CMI_Noise(CMI_VAR *pvar, char *pmodel, char *pinst)
```

- `pvar` Pointer to *CMI_VAR* variable
- `pmodel` Pointer to the model
- `pinst` Pointer to the instance
Example

```c
int
#ifdef __STDC__
CMImos3Noise(
CMI_VAR *pslot,
char    *pmodel,
char    *ptr)
#else
CMImos3Noise(pslot,pmodel,ptr)
#endif
CMI_VAR *pslot;
char    *pmodel;
char    *ptr;
#endif
{double freq,fourkt;
CMI_ENV      *penv
MOS3instance *ptran;
penv = pCMIenv; /* pCMIenv is a global */

ptran = (MOS3instance*)ptr;
fourkt = 4.0 * BOLTZMAN * ptran->temp; /* 4kT */
freq = pslot->freq;

/* Drain resistor thermal noise as current^2 source*/
pslot->nois_ird = fourkt * ptran->gdpr;

/* Source resistor thermal noise as current^2 source */
pslot->nois_irs = fourkt * ptran->gspr;

/* thermal noise assumed to be current^2 source referenced to channel. The source code for thermalnoise() is not shown here*/
pslot->nois_idsth = thermalnoise(model, here, fourkt);
```
/* flicker (1/f) noise assumed to be current^2 source referenced to channel. The source code for flickernoise() is not shown here */

pslot->nois_idsfl = flickernoise(model, here, freq);

return 0;
} /* int CMImos3Noise() */

CMI_PrintModel

This routine prints all model parameter names, values and units to standard output and is called by Star-Hspice for each model after CMI_SetupModel.

int CMI_PrintModel(char *pmodel)

Example

int

#ifdef __STDC__

CMImos3PrintModel(
char    *pmodel)
#else

CMImos3PrintModel(pmodel)
char    *pmodel;
#endif

{  
  CMI_ENV      *penv

  /* Note: source for CMImos3printmodel() not shown*/
  (void)CMImos3printmodel((MOS3model*)pmodel);

  return 0;
} /* int CMImos3PrintModel() */
CMI_FreeModel

This routine allows the user to free memory that previously was allocated for model related data. This routine is called during a loop over all models at post-simulation time.

```c
int CMI_FreeModel(char *pmodel)
```

- **pmodel** Pointer to the model

**Example**

```c
int
#ifdef __STDC__
CMImos3FreeModel(
char    *pmodel)
#else
CMImos3FreeModel(pmodel)
char    *pmodel;
#endif
{
/* free memory allocated for model data. Note CMImos3freemodel() source code not shown. */
(void)CMImos3freemodel((MOS3model*)pmodel);

return 0;
} /* int CMImos3FreeModel() */
```

CMI_FreeInstance

This routine allows the user to free memory that was previously allocated for storing instance-related data. This routine is called during an outer loop over all models and an inner loop over all instances associated with each model at post-simulation time.

```c
int CMI_FreeInstance(char *pmodel. char *pinst)
```

- **pmodel** Pointer to the model
- **pinst** Pointer to the instance
Example

```c
int
    #ifdef __STDC__
        CMImos3FreeInstance(
            char  *pmodel,
            char  *ptr)
    #else
        CMImos3FreeInstance(pmodel,ptr)
    #endif
{
    CMI_ENV      *penv
    MOS3instance *ptran;

    ptran = (MOS3instance*)ptr;
    /* free memory allocated for model data. Note
    CMImos3freeinstance() source code not shown. */
    (void)CMImos3freeinstance((MOS3model*)pmodel,ptran);

    return 0;
} /* int CMImos3FreeInstance() */
```

CMI_WriteError

This routine writes user-defined error messages to standard output when an error is detected during model evaluation. All CMI functions return a user-defined error code that is passed into CMI_WriteError(). In CMI_WriteError(), the user defines an error statement, copied to err_str, which is selected based on the error code value. CMI_WriteError() returns the error status: err_status>0 will cause Star-Hspice to write the error message and abort, err_status=0 will cause
Star-Hspice to write the warning message and continue. Star-Hspice calls CMI_WriteError() after every CMI function call.

```
int CMI_WriteError(int err_code, char *err_str)

err_code     Error code
err_str      Pointer to error message
```

returns error status (>0 Hspice aborts, ==0 Hspice continues)

**Example**

```
int
#ifdef __STDC__
CMImos3WriteError(
  void err_code,  
  char *err_str)
#else
CMImos3WriteError(err_code, err_str)
#endif
int err_code;  
char *err_str;
#endif
{
  /* */
  int err_status=0; 
  switch err_code 
  {
    case 1:
      strcpyn(err_str,"User Err: Eval()",& CMI_ERR_STR_LEN); 
      err_status=1;
    case 2:
      strcpyn(err_str,"User Warn: Eval()",& CMI_ERR_STR LEN); 
      err_status=1;
    default:
      strcpyn(err_str,"User Err:Generic",CMI_ERR_STR_LEN);
```
Using the Common Model Interface

Interface Variables

err_status=1;
}
return err_status;
} /* int CMImos3WriteError() */

CMI_Start
This routine allows user-defined startup functions to be run once prior to
simulation.
int CMI_Start(void)

Example
int
#ifdef __STDC__
CMImos3Start(void)
#else
CMImos3Start(void)
#endif
{
(void)CMImos3start();
return 0;
} /* int CMImos3Start() */

CMI_Conclude
This routine allows user-defined conclude functions to be run once at postsimulation time.
int CMI_Conclude(void)

Example
int
#ifdef __STDC__
CMImos3Conclude(void)
#else

23-35


CMI Function Calling Protocol

CMI calls the interface routines in the sequence as shown in Figure 23-1.
Figure 23-1: Interface Routines Calling Sequence

CMI_Start()

CMI_ResetModel()
CMI_AssignModelParm()
CMI_SetupModel()
CMI_WriteError()
CMI_ResetInstance()
CMI_AssignInstance Parm()
CMI_SetupInstance()
CMI_WriteError()
CMI_PrintModel()
CMI_DiodeEval()
CMI_WriteError()
CMI_Evaluate()
CMI_WriteError()
CMI_Noise()
CMI_WriteError()
CMI_FreeInstance()
CMI_FreeModel()
CMI_Conclude()
Internal Routines

In the example MOS3 implementation, the interface routines in *CMImos3.c* also call the following internal routines:

- **CMImos3GetIpar.c**: get instance parameter index
- **CMImos3SetIpar.c**: set instance parameter
- **CMImos3GetMpar.c**: get model parameter index
- **CMImos3SetMpar.c**: set model parameter
- **CMImos3eval.c**: evaluate model equations
- **CMImos3set.c**: setup a model
- **CMImos3temp.c**: setup an instance including temperature-effect

Figure 23-2 illustrates the hierarchical relationship between the interface routines and internal routines.

**Note:** For the automatic script to work, the name of the interface variable and all routine files must follow the naming convention, as follows:

- **pCMI_xxxdef**
- **CMIxxx.c**
- **CMIxxxSetIpar.c**
- **CMIxxxSetMpar.c**
- **CMIxxxGetIpar.c**
- **CMIxxxGetMpar.c**
- **CMIxxxeval.c**

where *xxx* is the model name.
Figure 23-2: Hierarchy of Interface and Internal Routines

<table>
<thead>
<tr>
<th>Interface Routines</th>
<th>Internal Routines</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMImos3.c</td>
<td>CMImos3setupModel()</td>
</tr>
<tr>
<td>CMI_ResetModel()</td>
<td>CMImos3temp()</td>
</tr>
<tr>
<td>CMI_ResetInstance()</td>
<td>CMImos3GetIpar()</td>
</tr>
<tr>
<td>CMI_SetupModel()</td>
<td>CMImos3GetMpar()</td>
</tr>
<tr>
<td>CMI_SetupInstance()</td>
<td>CMImos3SetIpar()</td>
</tr>
<tr>
<td>CMI_AssignModelParm()</td>
<td>CMImos3SetMpar()</td>
</tr>
<tr>
<td>CMI_AssignInstanceParm()</td>
<td>CMImos3SetMpar()</td>
</tr>
<tr>
<td>CMI_Evaluate()</td>
<td>CMImos3evaluate()</td>
</tr>
</tbody>
</table>

Optional Routines

- CMI_DiodeEval()
- CMI_Noise()
- CMI_PrintModel()
- CMI_FreeModel()
- CMI_FreeInstance()
- CMI_WriteError()
- CMI_Start()
- CMI_Conclude()
Supporting Extended Topology

In addition to conventional four terminal (topoid = 0) MOSFET topology, Star-Hspice can support other topologies. You must assign a unique topoid for different topologies.

BSIM SOI topology has been implemented in CMI and assigned a topoid of 1. If your own model topovar is the same as BSIM SOI, you can specify a topoid of 1 and use the Star-Hspice topology structure for stamping information. If your model topology is different from the conventional four-terminal model or the BSIM SOI, then you have to give Star-Hspice the topovar structure and Star-Hspice will assign a unique topoid for your topology.

For example, the following is the topovar structure for the BSIM SOI used in Star-Hspice CMI. The naming convention for the structure fields is the same as in BSIM SOI. For detailed information about fields in the structure, please refer to “BSIM3PD2.0 MOSFET MODEL User’ Manual,” which can be found at “http://www-device.eecs.berkeley.edu/~bsim3soi”.

```c
struct TOPO1 {
    double vps;
    double ves;
    double delTemp;    /* T node */
    double selfheat;

    double qsub;
    double qth;
    double cbodcon;

    double gbps;
    double gbpr;
    double gcde;
    double gcse;

    double gjdg;
    double gjdd;
};
```
double gjdb;
double gjdT;
double gjsg;
double gjsd;
double gjsb;
double gjsT;

double cdeb;
double cbeb;
double ceeb;
double cgeo;

/* add 4 for T */

double cgT;
double cdT;

double cbT;

double ceT;

double rth;

double cth;

double gmT;

double gbT;

double gbpT;

double gTtg;

double gTtd;

double gTtb;

double gTtt;

};
Conventions

Bias Polarity Conventions for N- and P-channel Devices

The input biases $vds$, $vgs$, and $vbs$ in CMI_VAR are defined as:

\[
\begin{align*}
    vds &= vd - vs \\
    vgs &= vg - vs \\
    vbs &= vb - vs
\end{align*}
\]

Negation of these biases for the P-channel device is required if your model code does not distinguish between n-channel and p-channel bias. In the example routines, the biases are multiplied by the model parameter "type", which is 1 for N-device and -1 for P-device. See for example the MOS3 model code:

```c
if (model->MOS3type < 0) { /* P-channel */
    vgs = -VgsExt;
    vds = -VdsExt;
    vbs = -VbsExt;
}
else { /* N-channel */
    vgs = VgsExt;
    vds = VdsExt;
    vbs = VbsExt;
}
```

This code should be used in both the CMI_Evaluate() and CMI_DiodeEval() functions.

The convention for outputting current components is shown in Figure 23-3. For channel current, drain-to-source is considered the positive direction. For substrate diodes, bulk-to-source/drain are considered the positive directions. The conventions are the same for both N-channel and P-channel devices.
Paragraph 1

Paragraph 2

Paragraph 3

Paragraph 4

Paragraph 5

Paragraph 6

Figure 23-3: MOSFET (node1, node2, node3, node4) - N-channel

The conventions for von are:
- N-channel, device is on if vgs > von
- P-channel, device is on if vgs < von

Derivatives (conductances and capacitances) should be provided based on the polarity conventions of the bias and current. The following code demonstrates the required polarity reversal for currents and Von, Vdsat for PMOS devices.

```c
if (model->type < 0)
{
    pslot->ids = -pslot->ids;
    pslot->ibs = -pslot->ibs;
    pslot->ibd = -pslot->ibd;
    pslot->von = -pslot->von;
    pslot->vdsat = -pslot->vdsat;
}
```

Source-Drain Reversal Conventions

Star-Hspice performs the appropriate computations when the MOSFET is operated in the reverse mode (i.e., when Vds < 0 for N-channel, Vds > 0 for P-channel). This includes a variable transformation (Vds -> -Vds, Vgs -> Vgd, Vbs->Vbd) and interchange of the source and drain terminals. This transformation is transparent to the model developer and simplifies the model coding task.
Thread-Safe Model Code

Star-Hspice uses shared-memory, multithreading algorithms during model evaluation. To ensure thread-safe model code, the following rules must be strictly adhered to:

- Do not use static variables in CMI_Evaluate(), CMIDiodeEval(), CMIWriteError(), and CMI_Noise() or in functions called by these routines.
- Never write to a global variable during execution of CMI_Evaluate(), CMIDiodeEval(), CMIWriteError(), and CMI_Noise().
Chapter 24
Performing Cell Characterization

Most ASIC vendors use Star-Hspice to characterize their standard cell libraries and prepare data sheets by using the basic capabilities of the .MEASURE statement. Input sweep parameters and the resulting measure output parameters are stored in the measure output data files design.mt0, design.sw0, and design.ac0. Multiple sweep data is stored in this file, and you can plot it by using AvanWaves. This lends itself to generating fanout plots of delay versus load. The slope and intercept of the loading curves can be used to calibrate VHDL, Verilog, Lsim, TimeMill, and Synopsys models.

This chapter covers:

- **Determining Typical Data Sheet Parameters**
  A series of typical data sheet examples show the flexibility of the MEASURE statement.

- **Cell Characterization Using Data Driven Analysis**
  Automates cell characterization, including timing simulator polynomial delay coefficient calculation. There is no limit on the number of parameters simultaneously varied or the number of analyses to be performed. Convenient ASCII file format for automated parameter input to Star-Hspice.
Determining Typical Data Sheet Parameters

This section describes how to determine typical data sheet parameters.

Rise, Fall, and Delay Calculations

The following example first calculates $v_{max}$, using the MAX function over the time region of interest. Then it calculates $v_{min}$ using the MIN function. Finally, the measured parameters can be used in subsequent calculations for accurate 10% and 90% points in the determination of the rise and fall time. Note that the RISE=1 is relative to the time window formed by the delay $TD_{val}$. Finally, the delay $T_{delay}$ is calculated using a fixed value for the measure threshold.

Example

```star-hsim
.MEAS TRAN v_{max} MAX V(out) FROM=TD_{val} TO=T_{stop}
.MEAS TRAN v_{min} MIN V(out) FROM=TD_{val} TO=T_{stop}
.MEAS TRAN Trise TRIG V(out) val='v_{min}+0.1*v_{max}' TD=TD_{val} + RISE=1 TARG V(out) val='0.9*v_{max}' RISE=1
.MEAS TRAN Tfall TRIG V(out) val='0.9*v_{max}' TD=TD_{val} + FALL=2 TARG V(out) val='v_{min}+0.1*v_{max}' FALL=2
.MEAS TRAN Tdelay TRIG V(in) val=2.5 TD=TD_{val} FALL=1 + TARG V(out) val=2.5 FALL=2
```

Figure 24-1: Rise, Fall, and Delay Time Demonstration
Ripple Calculation

This example performs the following:

- Delimits the wave at the 50% of VCC points
- Finds the midpoint $T_{mid}$
- Defines a bounded region by finding the pedestal voltage ($V_{mid}$) and then finding the first time that the signal crossed this value, $T_{from}$
- Measures the ripple in the defined region using the peak-to-peak (PP) measure function from $T_{from}$ to $T_{mid}$

Example

```
.MEAS TRAN Th1 WHEN V(out)='0.5*vcc' CROSS=1
.MEAS TRAN Th2 WHEN V(out)='0.5*vcc' CROSS=2
.MEAS TRAN Tmid PARAM='(Th1+Th2)/2'
.MEAS TRAN Vmid FIND V(out) AT='Tmid'
.MEAS TRAN Tfrom WHEN V(out)='Vmid' RISE=1
.MEAS TRAN Ripple PP V(out) FROM='Tfrom' TO='Tmid'
```

Figure 24-2: Waveform to Demonstrate Ripple Calculation
Sigma Sweep versus Delay

This file is set up to sweep sigma of the model parameter distribution while looking at the delay, giving the designer the delay derating curve for the model worst cases. This example is based on the demonstration file in $installdir/demo/hspice/cchar/sigma.sp. This technique of building a worst case sigma library is described in “Performing Worst Case Analysis” on page 13-8.

Example

```
.tran 20p 1.0n sweep sigma -3 3 .5
.meas m_delay trig v(2) val=vref fall=1 targ v(4) val=vref + fall=1
.param xlnew =’polycd-sigma*0.06u’ toxnew=’tox-sigma*10’
.model nch nmos LEVEL=28 xl = xlnew tox=toxnew
```

Figure 24-3: Inverter Pair Transfer Curves and Sigma Sweep vs. Delay
Delay versus Fanout

This example sweeps the subcircuit multiplier to quickly generate a family of five load curves. You can obtain more accurate results, by buffering the input source with one stage. The following example calculates the mean, variance, sigma, and average deviance for each of the second sweep variables (\textit{m\_delay} and \textit{rms\_power}). This example is based on the demonstration file $\texttt{installdir/demo/hspice/cchar/load1.sp}$.

\textbf{Input File Example}

\begin{verbatim}
tran 100p 2.0n sweep fanout 1 10 2
.param vref=2.5
.meas m_delay trig v(2) val=vref fall=1 + targ v(3) val=vref rise=1
.meas rms_power rms power

x1 in 2 inv
x2 2 3 inv
x3 3 4 inv m=fanout
\end{verbatim}

\textbf{Output Statistical Results}

\begin{verbatim}
meas\_variable = m\_delay
mean = 273.8560p varian = 1.968e-20
sigma = 140.2711p avgdev = 106.5685p

meas\_variable = rms\_power
mean = 5.2544m varian = 8.7044u
sigma = 2.9503m avgdev = 2.2945m
\end{verbatim}
Pin Capacitance Measurement

This example shows the effect of dynamic capacitance at the switch point. It sweeps the DC input voltage (\(pdcin\)) to the inverter and performs an AC analysis each 0.1 volt. The measure parameter \(incap\) is calculated from the imaginary current through the voltage source at the 10 kilohertz point in the AC curve (not shown). The peak capacitance at the switch point occurs when the voltage at the output side is changing in the opposite direction from the input side of the Miller capacitor, adding the Miller capacitance times the inverter gain to the total effective capacitance.

Example

```plaintext
mp out in 1 1 mp w=10u l=3u
mn out in 0 0 mn w=5u l=3u
vin in 0 DC= pdcin AC 1 0

.ac lin 2 10k 100k sweep pdcin 0 5 .1
.measure ac incap find par( '-1 * ii(vin)/ (hertz*twopi)' ) AT=10000hertz
```
Op-amp Characterization of ALM124

This example analyzes op-amps with .MEASURE statements to present a very complete data sheet. It references op-amp circuit output node out0 in the four .MEASURE statements using output variable operators for decibels vdb(out0), voltage magnitude vm(out0), and phase vp(out0). The example is taken from the demonstration file demo/apps/alm124.sp.

Input File Example

```
.measure ac 'unitfreq' trig at=1 targ vdb(out0) val=0 fall=1
.measure ac 'phasemargin' find vp(out0) when vdb(out0)=0
.measure ac 'gain(db)' max vdb(out0)
.measure ac 'gain(mag)' max vm(out0)
```
Measure Results

unitfreq = 9.0786E+05  targ= 9.0786E+05  trig= 1.0000E+00
phasemargin = 6.6403E+01
gain(db) = 9.9663E+01  at= 1.0000E+00  from= 1.0000E+00 + to= 1.0000E+07
gain(mag) = 9.6192E+04  at= 1.0000E+00  from= 1.0000E+00 + to= 1.0000E+07

Figure 24-6: Magnitude Plot of Op-Amp Gain
Cell Characterization Using Data Driven Analysis

This section provides example input files that perform cell characterization of an inverter based on 3-micron MOSFET technology. The program finds the propagation delay and rise and fall times for the inverter for best, worst, and typical cases for different fanouts. This data then can be used as library data for digital-based simulators such as those found in the simulation of gate arrays and standard cells.

The example, taken from the demonstration file $installdir/demo/hspice/apps/cellchar.sp, demonstrates the use of the .MEASURE statement, the .DATA statement, and the AUTOSTOP option in the characterization of a CMOS inverter. Figure 24-7 and Figure 24-8 are identical except that their input signals are complementary. The circuit in Figure 24-7 calculates the rise time and the low-to-high propagation delay time. The circuit in Figure 24-8 calculates the fall time and the high-to-low propagation delay time. When only one circuit is used, CPU time increases because the analysis time increases to calculate both rise and fall times.

**Figure 24-7: Cell Characterization Circuit 1**

![Cell Characterization Circuit 1](image)
The subcircuit XOUTL or XOUTH represents the fanout of the cell (inverter). Star-Hspice modifies fanout by specifying different multipliers (m) in the subcircuit calls.

Star-Hspice also provides local and global temperature specifications. This example characterizes the cell at global temperature 27, while devices M1 and M2 are at temperature (27+DTEMP). The .DATA statement specifies the DTEMP value.

The example uses a transient parameterized sweep with the .DATA and .MEASURE statements to determine the timing of the inverter for best, typical and worst cases. The parameters varied include power supply, input rise and fall time, fanout, MOSFET temperature, n-channel and p-channel threshold, and both the drawn width and length of the MOSFET. Use the AUTOSTOP option to speed simulation time and work with the .MEASURE statement. Once the .MEASURE statement determines the parameter to be measured, the AUTOSTOP option terminates the transient sweep, even though it has not completely swept the transient sweep range specified.

The .MEASURE statement uses quoted string parameter variables to measure the rise and fall times, as well as the propagation delays. Rise time starts when the voltage at node 3 (the output of the inverter) is equal to 0.1 \cdot VDD (that is,
V(3) = 0.1VDD) and ends when the voltage at node 3 is equal to 0.9 · VDD (that is, V(3) = 0.9VDD).

For more accurate results, start the .MEASURE calculation after a time delay, a simulation cycle specifying delay time in the .MEASURE statement, or in the input pulse statement.

The following example features:

- AUTOSTOP and .MEASURE statements
- Mean, variance, sigma, and avgdev calculations
- Circuit and element temperature
- Algebraic equation handling
- PAR( ) as output variable in the .MEASURE statement
- Subcircuit parameter passing and subcircuit multiplier
- .DATA statement

Example Input Files

FILE: CELLCHAR.SP

* .OPTIONS SPICE NOMOD AUTOSTOP
.PARAM TD=10N PW=50N TRR=5N TRF=5N VDD=5 LDEL=0 WDEL=0
+ NVT=0.8 PVT=-0.8 DTEMP=0 FANOUT=1
.GLOBAL VDD
* — global supply name
.TEMP 27

SUBCKT Definition

.SUBCKT INV IN OUT
M1 OUT IN VDD VDD P L=3U W=15U DTEMP=DTEMP
M2 OUT IN 0 0 N L=3U W=8U DTEMP=DTEMP
CL OUT 0 200E-15 .001
CI IN 0 50E-15 .001
.ENDS

SUBCKT Calls

XINVH 2 3 INV $--- INPUT START HIGH
XOUTL 3 4 INV M=FANOUT
XINVL 2030 INV $— INPUT START LOW
XOUTH 30 40 INV M=FANOUT
* — INPUT VOLTAGE SOURCES
VDD VDD 0 VDD
VINH 2 0 PULSE(VDD,0,TD,TRR,TRF,PW,200NS)
VINL 20 0 PULSE(0,VDD,TD,TRR,TRF,PW,200NS)
* — MEASURE STATEMENTS FOR RISE, FALL, AND PROPAGATION DELAYS
.MEAS RISETIME TRIG PAR('V(3) -0.1*VDD') VAL=0 RISE=1
+ TARG PAR('V(3) -0.9*VDD') VAL=0 RISE=1
.MEAS FALLTIME TRIG PAR('V(30) -0.9*VDD') VAL=0 FALL=1
+ TARG PAR('V(30) -0.1*VDD') VAL=0 FALL=1
.MEAS TPLH TRIG PAR('V(2) -0.5*VDD') VAL=0 FALL=1
+ TARG PAR('V(3) -0.5*VDD') VAL=0 RISE=1
.MEAS TPHL TRIG PAR('V(20) -0.5*VDD') VAL=0 RISE=1
+ TARG PAR('V(30) -0.5*VDD') VAL=0 FALL=1
* — ANALYSIS SPECIFICATION
.TRAN 1N 500N SWEEP DATA=DATNM
* — DATA STATEMENT SPECIFICATION
.DATA DATNM
VDD TRR TRF FANOUT DTEMP NVT PVT LDEL WDEL
5.0 2N 2N 2 0 0.8 -0.8 0 0 $ TYPICAL
5.5 1N 1N 1 -80 0.6 -0.6 -0.2U 0.2U $ BEST
4.5 3N 3N 10 100 1.0 -1.0 +0.2U -0.2U $ WORST
5.0 2N 2N 2 0 1.0 -0.6 0 0 $ STRONG P, WEAK N
5.0 2N 2N 2 0 0.6 -1.0 0 0 $ WEAK P, STRONG N
5.0 2N 2N 4 0 0.8 -0.8 0 0 $ FANOUT=4
5.0 2N 2N 8 0 0.8 -0.8 0 0 $ FANOUT=8
.ENDATA

Models
.MODEL N NMOS LEVEL=2 LDEL=LDEL WDEL=WDEL
+ VTO=NVT TOX=300 NSUB=1.34E16 UO=600
+ LD=0.4U WD=0.6U UCRIT=4.876E4 UEXP=.15
+ vmax=10E4 NEFF=15 PHI=.71 PB=.7
+ RS=10 RD=10 GAMMA=0.897 LAMBDA=0.004
Performing Cell Characterization

Cell Characterization Using Data Driven Analysis

+ DELTA=2.31 NFS =6.1E11 CAPOP=4
+ CJ=3.77E-4 CJSW=1.9E-10 MJ=.42 MJSW=.128
*

.MODEL P PMOS LEVEL=2 LDEL=LDEL WDEL=WDEL
+ VTO=PVT TOX=300 NSUB=0.965E15 UO=250
+ LD=0.5U WD=0.65U UCRIT=4.65E4 UEXP=.25
+ VMAX=1E5 NEFF=10 PHI=.574 PB=7
+ RS=15 RD=15 GAMMA=0.2 LAMBDA=.01
+ DELTA=2.486 NFS=5.2E11 CAPOP=4
+ CJ=1.75E-4 CJSW=2.3E-10 MJ=.42 MJSW=.128
.END

A sample of measure statements is printed:

*** MEASURE STATEMENT RESULTS FROM THE FIRST ITERATION ($ TYPICAL)
RISETIME  =  3.3551E-09 TARG=  1.5027E-08 TRIG=  1.1672E-08
FALLTIME  =  2.8802E-09 TARG=  1.4583E-08 TRIG=  1.1702E-08
TPLH      =  1.8537E-09 TARG=  1.2854E-08 TRIG=  1.1000E-08
TPHL      =  1.8137E-09 TARG=  1.2814E-08 TRIG=  1.1000E-08

*** MEASURE STATEMENT RESULTS FROM THE LAST ITERATION ($ FANOUT=8)
RISETIME  =  8.7909E-09 TARG=  2.0947E-08 TRIG=  1.2156E-08
FALLTIME  =  7.6526E-09 TARG=  1.9810E-08 TRIG=  1.2157E-08
TPLH      =  3.9922E-09 TARG=  1.4992E-08 TRIG=  1.1000E-08
TPHL      =  3.7995E-09 TARG=  1.4800E-08 TRIG=  1.1000E-08

MEAS_VARIABLE = RISETIME
MEAN =  6.5425E-09 VARIAN =  4.3017E-17
SIGMA =  6.5588E-09 AVGDEV =  4.6096E-09

MEAS_VARIABLE = FALLTIME
MEAN =  5.7100E-09 VARIAN =  3.4152E-17
SIGMA =  5.8440E-09 AVGDEV =  4.0983E-09

MEAS_VARIABLE = TPLH
MEAN =  3.1559E-09 VARIAN =  8.2933E-18
SIGMA =  2.8798E-09 AVGDEV =  1.9913E-09
MEAS_VARIABLE = TPHL
MEAN = 3.0382E-09  VARIAN = 7.3110E-18
SIGMA = 2.7039E-09  AVGDEV = 1.8651E-0

Figure 24-9: Plotting the Simulation Outputs
Figure 24-10: Verifying the Measure Statement Results by the Plots
Chapter 25

Signal Integrity

The performance of an IC design is no longer limited to how many million transistors a vendor fits on a single chip. With tighter packaging space and increasing clock frequencies, packaging and system-level performance issues such as crosstalk and transmission lines are becoming increasingly significant.

At the same time, with the popularity of multichip packages and increased I/O counts, package design itself is becoming more and more like chip design.

This chapter describes how to maintain signal integrity for your design, and covers the following topics:

- Preparing for Simulation
- Optimizing TDR Packaging
- Simulating Circuits with Signetics Drivers
- Simulating Circuits with Xilinx FPGAs
- PCI Modeling Using Star-Hspice
Preparing for Simulation

To simulate a PC board or backplane with Star-Hspice, you must consider models for:

- A driver cell, including the parasitic pin capacitances and package lead inductances
- Transmission lines
- A receiver cell with its parasitic pin capacitances and package lead inductances
- Terminations or other electrical elements on the line

It is important to model the transmission line as closely as possible—that is, to include all electrical elements exactly as they are laid out on the backplane or printed circuit board, to maintain the integrity of the simulation.

With readily available I/O drivers from ASIC vendors and Star-Hspice’s advanced lossy transmission lines, you can simulate the electrical behavior of the board interconnect, bus, or backplane to analyze the transmission line behavior under various conditions.

Simulation is possible because the critical models and simulation technology exist.

- Many manufacturers of high-speed components use Star-Hspice already.
- The complexity can be hidden from the system level.
- The necessary electrical characteristics are preserved with full transistor level library circuits.

Star-Hspice has been enhanced for systems simulation with:

- Systems level behavior, such as local component temperature and independent models, to allow accurate prediction of electrical behavior
- Automatic inclusion of library components via the SEARCH option
- Lossy transmission line models that:
  - Support common mode simulation
  - Include ground plane reactance
Include resistive loss of conductor and ground plane
Allow multiple signal conductors
Require minimum CPU computation time

The following vendor models are currently available in Star-Hspice:

- Signetics FAST Library
- Xilinx 3000/4000 Series FPGA
- Intel’s Peripheral Component Interconnect (PCI) high-speed local bus

**Signal Integrity Problems**

Some signal integrity problems that can cause failures in high-speed designs are listed in Table 25-1.

<table>
<thead>
<tr>
<th>Signal Integrity Problem</th>
<th>Causes</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise: delta I (current)</td>
<td>Multiple simultaneously switching drivers; high-speed devices create larger delta I</td>
<td>Adjust or evaluate location, size, and value of decoupling capacitors.</td>
</tr>
<tr>
<td>Noise: coupled (crosstalk)</td>
<td>Closely spaced parallel traces</td>
<td>Establish parallel line length design rules.</td>
</tr>
<tr>
<td>Noise: reflective</td>
<td>Impedance mismatch</td>
<td>Reduce the number of connectors and select proper impedance connectors.</td>
</tr>
<tr>
<td>Delay: path length</td>
<td>Poor placement and routing; too many or too few layers; chip pitch</td>
<td>Choose MCM or other high-density packaging technology.</td>
</tr>
<tr>
<td>Propagation speed</td>
<td>Dielectric medium</td>
<td>Choose dielectric with lowest dielectric constant.</td>
</tr>
<tr>
<td>Delay: rise time degradation</td>
<td>Resistive loss and impedance mismatch</td>
<td>Adjust width, thickness and length of line.</td>
</tr>
</tbody>
</table>
Analog Side of Digital Logic

Circuit simulation of a digital system only becomes necessary when the analog characteristics of the digital signals become electrically important. Is the digital circuit a new design, or simply a fast version of the old design? Many new digital products are really faster versions of existing designs. The transition from a 100 MHz to a 150 MHz Pentium PC may not require extensive logic simulations. However, the integrity of the digital quality of the signals may require very careful circuit analysis.

The source of a signal integrity problem is the digital output driver. A high-speed digital output driver can only drive a few inches before the noise and delay due to the wiring become a problem. To speed up circuit simulation and modeling, you can create analog behavioral models that mimic the full analog characteristics at a fraction of the traditional evaluation time. The simulation of the output buffer in Figure 25-1 demonstrates the analog behavior of a digital gate simulated in the Star-Hspice circuit simulator.
The roadblocks to successful high-speed digital designs are noise and signal delays. Digital noise can come from several sources. The fundamental digital noise sources are:

- Line termination noise
- Ground bounce noise
- Coupled line noise

Line termination noise is the additional voltage that is reflected from the load back to the driver because of impedance mismatch. Digital output buffers are not designed to have accurately controlled output impedance and most buffers have different rising and falling edge impedances.
Ground bounce noise is generated where leadframes or other circuit wires cannot be formed into transmission lines. The resulting inductance creates an induced voltage in the ground circuit, the supply circuit, and the output driver circuit. The ground bounce noise lowers the noise margins for the rest of the system. Coupled line noise is the noise induced from lines that are physically adjacent. This noise is generally most severe for data lines that are next to clock lines.

Circuit delays become critical as timing requirements become tighter. The key circuit delays are:
- Gate delays
- Line turnaround delays for tristate buffers
- Line length delays (clock skew)

Logic analysis only addresses gate delays. You can compute the variation in the gate delay from circuit simulation only if you understand the best case and worst case manufacturing conditions. The line turnaround delays add to the gate delays because extra margin must be added so that multiple tristate buffer drivers do not simultaneously turn on. The line length delay affects the clock skew most directly in most systems. As system cycle times approach the speed of electromagnetic signal propagation for the printed circuit board, consideration of the line length becomes critical. The system noises and line delays interact with the electrical characteristics of the gates and may require circuit level simulation.

Analog details find digital systems problems. Exceeding the noise quota may not cause a system to fail. Only when a digital input is being accepted does the maximum noise become a problem. If a digital systems engineer can decouple the system, much higher noise can be accepted.

Common decoupling methods are:
- Multiple ground and power planes on the PCB, MCM, PGA
- Separating signal traces with ground traces
- Decoupling capacitors
- Series resistors on output buffer drivers
- Twisted pair line driving
In present systems designs, you must select the best packaging methods at the printed circuit board level, the multi-chip module level, and the pin grid array level. Extra ground and power planes are often necessary to lower the supply inductance and provide decoupling. Decoupling capacitors must have very low internal inductance in order to be effective for high speed designs. Newer designs frequently use series resistance in the output drivers to lower circuit ringing. Finally, in critical high speed driver applications, twisted differential pair transmission lines are used.

The systems engineer must determine how to partition the logic. The propagation speed of signals on a printed circuit board is about 6 in/ns. As digital designs become faster, the wiring interconnect becomes a factor in deciding how to partition the logic. The critical wiring systems are:

- IC level wiring
- Package wiring for SIPs, DIPs, PGAs, MCMs
- Printed circuit board wiring
- Backplane and connector wiring
- Long lines – power, coax, twisted pair

Systems designers who use ASIC or custom integrated circuits as part of their system logic partitioning strategy find that they must make decisions about integrated circuit level wiring. The more familiar decisions involve the selection of packages and the arrangement of packages on a printed circuit board. Large systems generally have a central backplane that becomes the primary challenge at the system partition level.

Use the following equation to estimate wire length when transmission line effects become noticeable:

\[
\text{critical length} = \frac{(\text{rise time}) \times \text{velocity}}{8}
\]

For example, if rise time is 1 ns and board velocity is 6 in/ns, distortion becomes noticeable at a wire length of 3/4 in. The Star-Hspice circuit simulator automatically generates models for each type of wire to define full loss transmission line effects.
ECL logic design engineers typically partitioned the system by calculating the noise quota for each line. Now, most high-speed digital logic must be designed with respect to the noise quota so that the engineer knows how much noise and delay can be accepted before the timing and logic levels fail.

To solve the noise quota problem, you must calculate the noise associated with the wiring. Large integrated circuits can be separated into two parts: the internal logic and the external input and output amplifiers.

**Figure 25-2: Analog Drivers and Wires**

Using mixed digital and analog tools such as Avant!’s Star-Hspice and Viewlogic’s Viewsim A/D, you can merge a complete system together with full analog quality timing constraints and full digital representation. You can simultaneously evaluate noise quota calculation subject to system timing.
Optimizing TDR Packaging

Packaging plays an important role in determining the overall speed, cost, and reliability of a system. With today’s small feature sizes and high levels of integration, a significant portion of the total delay comes from the time required for a signal to travel between chips.

Multilayer ceramic technology has proven to be well suited for high-speed GaAs IC packages.

The multichip module minimizes the chip-to-chip spacing and reduces the inductive and capacitive discontinuity between the chips mounted on the substrate with a more direct path (die-bump-interconnect-bump-die), thus eliminating wirebonding. In addition, narrower and shorter wires on the ceramic substrate have much less capacitance and inductance than the PC board interconnections.

Time domain reflectometry (TDR) is the closest measurement to actual digital component function. It provides a transient display of the impedance versus time for pulse behavior.

With a digitized TDR file, you can automatically select design components using the Star-Hspice optimizer. You can extract critical points from digitized TDR files using the Star-Hspice .MEASURE statement, and use the results as electrical specifications for optimization. This process eliminates recurring design cycles to find component values to curve-fit the TDR files.

Figure 25-3: Optimization Process
The following is a method used for realistic high-speed testing of packaging.

Test fixtures were designed that closely emulate a high-speed system environment. A Star-Hspice model was constructed for measurements using ideal transmission lines and discrete components.

The circuit tested contained the following components:

- Signal generator
- Coax connecting the signal generator to ETF (engineering test fixture) board
- ETF board
- Package pins
- Package body

The package tests performed traditional time domain measurements using a digital sampling oscilloscope. Tests were designed to observe the reflected and
transmitted signals derived from the built-in high-speed pulse generator and translated output signals into digitized time domain reflectometer files (voltage vs. time).

A fully developed SPICE model was used to simulate the package-plus-test fixture. The simulated and measured reflected/transmitted signals were compared.

The input netlist file for this experiment is shown on the following pages. Output plots are shown in Figures 25-6 through 25-9.

You can further investigate this experiment using Star-Hspice’s advanced lossy transmission lines to include attenuation and dispersion.

**TDR Optimization Procedure**

**Measure Critical Points in the TDR Files**

```plaintext
Vin 1 0 PWL(TIME,VOLT)
    .DATA D_TDR
        TIME   VOLT
        0       0.5003mV
        0.1n    0.6900mV
        ...
        2.0n    6.4758mV
    .ENDDATA
    .TRAN DATA=D_TDR
    .MEAS ..... 
    .END
```

**Set Up an Input Optimization File**

```plaintext
$ SPICE MODEL FOR PACKAGE-PLUS-TEST Fixture
$ AUTHOR: DAVID H. SMITH & RAJ M. SAVARA
    .OPTION POST RELV=1E-4 RELVAR=1E-2

$ DEFINE PARAMETERS
    .PARAM LV=-0.05 HV=0.01 TD=1P TR=25P TF=50P TPW=10N TPER=15N

$ PARAMETERS TO BE OPTIMIZED
    .PARAM CSMA=OPT1(500f,90f,900f,5f) 
        +XTD=OPT1(150p,100p,200p)
```
Optimizing TDR Packaging

```
+      LPIN=OPT1(0.65n,0.10n,0.90n,0.2n)
+      LPK=OPT1(1.5n,0.75n,3.0n,0.2n)
+      LPKCL=0.33n
+      LPKV=0.25n

Signal Generator
VIN   S1 GND PULSE LV HV TD TR TF TPW TPER
RIN   S1 S2  50
CIN1  S2 GND 250f
TCOAX S2 GND SIG_OUT GND ZO=50 TD=50p

ETF Board
CSNAL SIG_OUT GND CSMA
TEFT2 SIG_OUT GND E3 GND ZO=50 TD=XTD
RLOSS1 E3 E4 10
CRPAD1 E4 GND 200f
TLIN2 E4 GND ETF_OUT GND ZO=50 TD=35p
CPAD2 ETF_OUT GND 300f
TLIN1 E5 GND E6 GND ZO=50 TD=35p
CPAD1 E5 GND 300f
CRPAD2 E6 GND 200f
RLOSS1 E6 E7 10
TEFT1 E7 GND E8 GND ZO=50 TD=XTD
CSMA2 E8 GND CSMA
TCOAX2 E8 GND VREF1 GND ZO=50 TD=50p
RIN1 VREF1 GND 50

Package Body
LPIN1 ETF_OUT P1 LPIN
LPK1 GND P5 LPK
LPKGCL P5 NVOUT2 LPKCL
CPKG1 P1 P5 250f
LPKV1 P1 P2 LPKV
TPKGL P2 NVOUT2 VOUT NVOUT2 ZO=65 TD=65p
CBPL VOUT NVOUT 1f
ROUT1 VOUT NVOUT 50meg
LPIN2 E5 P3 LPIN
CPKG2 P3 NVOUT2 250f
LPKV2 P3 P4 LPKV
TPKG2 P4 NVOUT2 VOUT2 NVOUT2 ZO=65 TD=65p
```
CBPD1 VOUT2 NVOUT2 1f
ROUT2 VOUT2 NVOUT2 50meg

$ BEFORE OPTIMIZATION
.TRAN .004NS 2NS

$ OPTIMIZATION SETUP
.MODEL OPTMOD OPT ITROPT=30
.TRAN .05NS 2NS SWEEP OPTIMIZE=OPT1
+ RESULTS=MAXV,MINV,MAX_2,COMP1,PT1,PT2,PT3
+ MODEL=OPTMOD

$ MEASURE CRITICAL POINTS IN THE REFLECTED SIGNAL
$ GOALS ARE SELECTED FROM MEASURED TDR FILES
.MEAS TRAN COMP1 MIN V(S2) FROM=100p TO=500p GOAL=-27.753
.MEAS TRAN PT1 FIND V(S2) AT=750p GOAL=-3.9345E-3 WEIGHT=5
.MEAS TRAN PT2 FIND V(S2) AT=775p GOAL=2.1743E-3 WEIGHT=5
.MEAS TRAN PT3 FIND V(S2) AT=800p GOAL=5.0630E-3 WEIGHT=5

$ MEASURE CRITICAL POINTS IN THE TRANSMITTED SIGNAL
$ GOALS ARE SELECTED FROM MEASURED TDR FILES
.MEAS TRAN MAXV FIND V(VREF1) AT=5.88E-10 GOAL=6.3171E-7 WEIGHT=7
.MEAS TRAN MINV FIND V(VREF1) AT=7.60E-10 GOAL=-9.9181E-3
.MEAS TRAN MAX_2 FIND V(VREF1) AT=9.68E-10 GOAL=4.9994E-3

$ COMPARE SIMULATED RESULTS WITH MEASURED TDR VALUES
.TRAN .004NS 2NS
.PRINT C_REF=V(S2) C_TRAN=V(VREF1)
.END
Figure 25-6: Reflected Signals Before Optimization
Figure 25-7: Reflected Signals After Optimization
Figure 25-8: Transmitted Signals Before Optimization
Figure 25-9: Transmitted Signals after Optimization
Simulating Circuits with Signetics Drivers

The Signetics I/O buffer library is distributed with Star-Hspice in the $installdir/parts/signet directory. These are high-performance parts used in backplane design. The transmission line model used describes two conductors.

**Figure 25-10: Planar Transmission Line DLEV=2: Microstrip Sea of Dielectric**

In the following application, a pair of drivers are driving about 2.5 inches of adjacent lines to a pair of receivers that drive about four inches of line.
Example

This is an example of connecting I/O chips with transmission lines.

```plaintext
 OPTIONS SEARCH='$/installdir/parts/signet'
 OPTIONS POST=2 TNOM=27 NOMOD LIST METHOD=GEAR
 TEMP 27

 DEFINE PARAMETER VALUES
 PARAM LV=0 HV=3 TD1=10n TR1=3n TF1=3n TPW=20n TPER=100n
 + TD2=20n TR2=2n TF2=2n LNGTH=101.6m

 POWER SUPPLY
 VCC VCC 0 DC 5.5

 INPUT SOURCES
 VIN1 STIM1 0 PULSE LV HV TD1 TR1 TF1 TPW TPER
 VIN2 STIM2 0 PULSE LV HV TD2 TR2 TF2 TPW TPER

 FIRST STAGE: DRIVER WITH TLINE
 X1ST_TOP STIM1 OUTPIN1 VCC GND IO_CHIP PIN_IN=2.6n PIN_OUT=4.6n
 X1ST_DN STIM2 OUTPIN2 VCC GND IO_CHIP PIN_IN=2.9n PIN_OUT=5.6n
 U_1ST OUTPIN1 OUTPIN2 GND TLOUT1 TLOUT2 GND USTRIP L=LNGTH

 SECOND STAGE: RECEIVER WITH TLINE

 X2ST_TOP TLOUT1 OUTPIN3 VCC GND IO_CHIP PIN_IN=4.0n PIN_OUT=2.5n
 X2ST_DN TLOUT2 OUTPIN4 VCC GND IO_CHIP PIN_IN=3.6n PIN_OUT=5.1n
 U_2ST OUTPIN3 OUTPIN4 GND TLOUT3 TLOUT4 GND USTRIP L=LNGTH
```
$ TERMINATING RESISTORS
R1 TLOUT3 GND 75
R2 TLOUT4 GND 75
$ IO CHIP MODEL - SIGNETICS
.SUBCKT IO_CHIP IN OUT VCC XGND PIN_VCC=7n PIN_GND=1.8n

X1 IN1 INVOUT VCC1 XGND1 ACTINPUT
X2 INVOUT OUT1 VCC1 XGND1 AC109EQ

Package Inductance
LIN_PIN IN IN1 PIN_IN
LOUT_PIN OUT1 OUT PIN_OUT
LVCC VCC VCC1 PIN_VCC
LGND XGND1 XGND PIN_GND
.ENDS

$ TLINE MODEL - 2 SIGNAL CONDUCTORS WITH GND
$ PLANE

.MODEL USTRIP U LEVEL=3 ELEV=1 PLEV=1
+ TH1=1.3mil HT1=10mil TS=32mil KD1=4.5 DLEV=0 WD1=8mil
+ XW=-2mil KD2=4.5 NL=2 SP12=5mil
$ ANALYSIS / PRINTS
.TRAN .1NS 100NS
.GRAPH IN1=V(STIM1) IN2=V(STIM2) VOUT1=V(TLOUT1) VOUT2=V(TOUT2)
.GRAPH VOUT3=V(TLOUT3) VOUT4=V(TLOUT4)
.END
Figure 25-12: Connecting I/O Chips with Transmission Lines
Simulating Circuits with Xilinx FPGAs

Avant!, in conjunction with Xilinx, maintains a library of Star-Hspice transistor level subcircuits for the 3000 and 4000 series Field Programmable Gate Arrays (FPGAs). These subcircuits model the input and output buffer.

The following simulations use the Xilinx input/output buffer (xil_iob.inc) to simulate the ground bounce effects for the 1.08µm process at room temperature and nominal model conditions. The IOB and IOB4 are parameterized Star-Hspice subcircuits that allow you to specify:

- Local temperature
- Fast/slow/typical speed selections
- Technology 1.2µ/1.08µ

These choices allow the system designer to perform a variety of simulations to measure:

- Ground bounce as a function of package, temperature, part speed and technology
- Coupled noise, both on-chip and chip-to-chip
- Full transmission line effects at the package and printed circuit board levels
- Peak current and instantaneous power consumption for power supply bussing considerations and chip capacitor placement

Syntax for IOB (xil_iob) and IOB4 (xil_iob4)

* EXAMPLE OF CALL FOR 1.2U PART:
* X1  I  O PAD TS FAST PPUB TTL VDD GND  XIL_IOB
*+ XIL_SIG=0 XIL_DTEMP=0 XIL_SHRINK=0

* EXAMPLE OF CALL FOR 1.08U PART:
* X1  I  O PAD TS FAST PPUB TTL VDD GND  XIL_IOB
*+ XIL_SIG=0 XIL_DTEMP=0 XIL_SHRINK=1
<table>
<thead>
<tr>
<th>Nodes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I (IOB only)</td>
<td>output of the TTL/CMOS receiver</td>
</tr>
<tr>
<td>O (IOB only)</td>
<td>input pad driver stage</td>
</tr>
<tr>
<td>I1 (IOB4 only)</td>
<td>input data 1</td>
</tr>
<tr>
<td>I2 (IOB4 only)</td>
<td>input data 2</td>
</tr>
<tr>
<td>DRIV_IN (IOB4 only)</td>
<td></td>
</tr>
<tr>
<td>PAD</td>
<td>bonding pad connection</td>
</tr>
<tr>
<td>TS</td>
<td>three-state control input (5 V disables)</td>
</tr>
<tr>
<td>FAST</td>
<td>slew rate control (5 V fast)</td>
</tr>
<tr>
<td>PPUB (IOB only)</td>
<td>pad pull-up enable (0 V enables)</td>
</tr>
<tr>
<td>PUP (IOB4 only)</td>
<td>pad pull-up enable (0 V enables)</td>
</tr>
<tr>
<td>PDOWN (IOB4 only)</td>
<td>pad pull-up enable (5 V enables)</td>
</tr>
<tr>
<td>TTL (IOB only)</td>
<td>CMOS/TTL input threshold select (5 V selects TTL)</td>
</tr>
<tr>
<td>VDD</td>
<td>5 volt supply</td>
</tr>
<tr>
<td>GND</td>
<td>ground</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
</table>
| XIL_SIG | model distribution: (default 0)  
-3⇒ slow  
0⇒ typical  
+3⇒ fast |
All grounds and supplies are common to the external nodes for ground and VDD. Star-Hspice allows you to redefine grounds for the addition of package models.

**Ground Bounce Simulation**

The ground bounce simulation presented duplicates Xilinx internal measurements methods; 8 to 32 outputs are simultaneously toggled. Each output is loaded with a 56-pF capacitance. The simulation uses an 84-pin package mode and an output buffer held at chip ground to measure the internal ground bounce.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XIL_DTEMP</td>
<td>Buffer temperature difference from ambient</td>
</tr>
<tr>
<td></td>
<td>The default = 0 degrees if ambient is 25 degrees and the buffer is 10 degrees hotter than XIL_DTEMP=10.</td>
</tr>
<tr>
<td>XIL_SHRINK</td>
<td>Old or new part; (default is new)</td>
</tr>
<tr>
<td></td>
<td>0==＞old</td>
</tr>
<tr>
<td></td>
<td>1==＞new</td>
</tr>
</tbody>
</table>

**Figure 25-13: Ground Bounce Simulation**
The simulation model is adjusted for the oscilloscope recordings for the two-bond wire ground.

**Star-Hspice Input File for Ground Bounce**

```
qabounce.sp test of xilinx  i/o buffers

* The following is the netlist for the above schematic(fig. 10-13)
.op
.option  post list
.tran 1ns 50ns sweep gates 8 32 4
.measure bounce  max v(out1x)
*.tran .1ns 7ns
.param gates=8
.print v(out1x) v(out8x) i(vdd) power
$.param xil_dtemp=-65  $ -40 degrees c  (65 degrees from +25 degrees)
.vdd vdd gnd 5.25
.vgnd return gnd 0
.upower1 vdd return ioblvdd ioblgnrd pcb_power L=600mil
* local power supply capacitors
.xcl_a ioblvdd ioblgnrd cap_mod cval=.1u
.xclb ioblvdd ioblgnrd cap_mod cval=.1u
.xclc ioblvdd ioblgnrd cap_mod cval=1u
.xgnd_b  ioblvdd ioblgnrd out8x out1x  xil_gnd_test
.xcout8x out8x iob1gnd cap_mod  m=gates
.xcout1x out1x iob1gnd cap_mod  m=1

.model pcb_power u LEVEL=3 elev=1 plev=1 nl=1 llev=1
+ th=1.3mil ht=10mil kd=4.5  dlev=1 wd=500mil xw=-2mil

.model cap_mod node1 node2  cval=56p
.Lr1 node1 node1x L=2nh R=0.05
cap node1x node2x c=cval
.Lr2 node2x node2 L=2nh R=0.05
.eom

.macro xil_gnd_test  vdd gnd outx outref
+ gates=8
* example of 8 iobuffers simultaneously switching
* through approx. 4nh lead inductance
* 1 iob is active low for ground bounce measurements

.vout drive chipgnd pwl 0ns 5v, 10ns 5v, 10.5ns 0v,
+. 20ns 0v, 20.5ns 5v, 40ns 5v R
.x8  I8 drive PAD8x TS FAST PPUB TTL chipvdd chipgnd xil_iob
+ xil_sig=0 xil_dtemp=0  xil_shrink=1  M=gates
```

Simulating Circuits with Xilinx FPGAs

Control Settings

rts  ts  chipgnd 1
rfast fast chipvdd 1
rppub ppub chipgnd 1
rttl ttl chipvdd 1
* pad model plcc84 rough estimates
lvdd vdd chipvdd L=3.0nh r=.02
lgnd gnd chipgnd L=3.0nh r=.02
lout8x outx pad8x L='5n/gates' r='0.05/gates'
lout1x outref pad1x L=5nh r=0.05
c_vdd_gnd chipvdd chipgnd 100n
.eom
.end

Figure 25-14: Results of Ground Bounce Simulation
Coupled Line Noise

This example uses coupled noise to separate IOB parts. The output of one part drives the input of the other part through 0.6 inch of PCB. The example also monitors an adjacent quiet line.

**Figure 25-15: Coupled Noise Simulation**

Coupled Noise

Star-Hspice Input File for qa8.sp test of xilinx 0.8u i/o buffers

* The following is the netlist for the above schematic (fig 10-15)
  .op
  .option nomod post=2
  *.tran .1ns 5ns sweep xil_sig -3 3 3
  .tran .1ns 15ns
  .print v(out1x) v(out3x) i(vdd) v(irec)

  vdd  vdd gnd 5
  vgnd return gnd 0
Simulating Circuits with Xilinx FPGAs

Signal Integrity

upower1 vdd return iob1vdd iob1gnd pcb_power L=600mil
upower2 vdd return iob2vdd iob2gnd pcb_power L=600mil
x4io iob1vdd iob1gnd out3x out1x outrec irec xil_iob4
cout3x out3x iob1gnd 9pf
u1x out1x outrec iob1gnd i_o_in i_o_out iob2gnd pcb_top L=2000mil
xrec iob2vdd iob2gnd i_o_in i_o_out xil_rec
.ic i_o_out 0v
.model pcb_top u LEVEL=3 elev=1 plev=1 nl=2 llev=1
+ th=1.3mil ht=10mil sp=5mil kd=4.5 dlev=1 wd=8mil xw=-2mil
.model pcb_power u LEVEL=3 elev=1 plev=1 nl=1 llev=1
+ th=1.3mil ht=10mil kd=4.5 dlev=1 wd=500mil xw=-2mil
.macro xil_rec vdd gnd tri1 tri2
* example of 2 ibuffers in tristate
xtri1 Irec 0 pad_tri1 TSrec FAST PPUB TTL chipvdd chipgnd xil_iob
+ xil_alg=0 xil_dtemp=0 xil_shrink=1 m=1
xtri2 Irec 0 pad_tri2 TSrec FAST PPUB TTL chipvdd chipgnd xil_iob
+ xil_alg=0 xil_dtemp=0 xil_shrink=1 m=1

Control Setting
rin_output 0 chipgnd 1
rtsrec tsrec chipvdd 1
rfast fast chipvdd 1
rppub ppub chipgnd 1
rttl ttl chipvdd 1
* pad model plcc84 rough estimates
lvdd vdd chipvdd L=1nh r=.01
lgnd gnd chipgnd L=1nh r=.01
ltri1 tril pad_tri1 L=3nh r=0.01
ltri2 tril2 pad_tri2 L=3nh r=.01
c_vdd_gnd chipvdd chipgnd 100n
.eom

.macro xil_iob4 vdd gnd out3x out1x outrec Irec
* example of 4 ibuffers simultaneously switching through approx.
* 3nh lead inductance
* 1 ib is a receiver (tristated)
.vout 0 chipgnd pwl 0ns 0v, 1ns 0v, 1.25ns 4v, 7ns 4v, 7.25ns 0v, 12ns 0v
x3 I3 0 PAD3x TS FAST PPUB TTL chipvdd chipgnd xil_iob
+ xil_alg=0 xil_dtemp=0 xil_shrink=1 m=3
xl  I1 O PAD1x TS FAST PPUB TTL chipvdd chipgnd xil_iob
+ xil_sig=0 xil_dtemp=0 xil_shrink=1 m=1
xrec  Irec O PADrec TSrec FAST PPUB TTL chipvdd chipgnd xil_iob
+ xil_sig=0 xil_dtemp=0 xil_shrink=1 m=1
* control settings
rts  ts  chipgnd 1
rtsrec tsrec chipvdd 1
rfast  fast  chipvdd 1
rppub  ppub  chipgnd 1
rttl  ttl  chipvdd 1
* pad model plcc84 rough estimates
lvdd  vdd  chipvdd L=1nh r=.01
lgnd  gnd  chipgnd L=1nh r=.01
lout3x  out3x  pad3x L=1nh r=.0033
lout1x  out1x  pad1x L=4nh r=0.01
loutrec  outrec  padrec L=4nh r=.01
c_vdd_gnd  chipvdd chipgnd 100n
.eom
.end
**Figure 25-16: Results of Coupled Noise Simulation**

![Graph showing coupled noise simulation results]

**IOB Buffer Module**

* XILINX IOB INPUT/OUTPUT CIRCUIT
* NAME: XIL_IOB.INC
* PURPOSE: XILINX INPUT/OUTPUT BLOCK MODEL
* EXAMPLE OF CALL FOR 1.2U PART:
  * X1  I O PAD TS FAST PPUB TTL VDD GND XIL_IOB
  * XIL_SIG=0 XIL_DTEMP=0 XIL_SHRINK=0
* EXAMPLE OF CALL FOR 1.08U PART:
  * X1  I O PAD TS FAST PPUB TTL VDD GND XIL_IOB
  * XIL_SIG=0 XIL_DTEMP=0 XIL_SHRINK=1
* NAME: XIL_IOB.INC
* PURPOSE: XILINX INPUT/OUTPUT BLOCK MODEL
* PINS:

---

Signal Integrity

Simulating Circuits with Xilinx FPGAs

* I     OUTPUT OF THE TTL/CMOS INPUT RECEIVER.
* O     INPUT TO THE PAD DRIVER STAGE.
* PAD   BONDING PAD CONNECTION.
* TS    THREE-STATE CONTROL INPUT. HIGH LEVEL
       DISABLES PAD DRIVER.
* FAST  SLEW RATE CONTROL. HIGH LEVEL SELECTS
       FAST SLEW RATE.
* PPUB  PAD PULL-UP ENABLE. ACTIVE LOW.
* TTL   CMOS/TTL INPUT THRESHOLD SELECT. HIGH
       SELECTS TTL.
* VDD   POSITIVE SUPPLY CONNECTION FOR INTERNAL
       CIRCUITRY.
* GND   CIRCUIT GROUND

ALL THE ABOVE SIGNALS ARE REFERENCED TO NODE 0.
THIS MODEL DOES CAUSE SOME DC CURRENT TO FLOW
INTO NODE 0 THAT IS AN ARTIFACT OF THE MODEL.

Description

THIS SUBCIRCUIT MODELS THE INTERFACE BETWEEN XILINX
3000 SERIES PARTS AND THE BONDING PAD. IT IS NOT
USEFUL FOR PREDICTING DELAY TIMES FROM THE OUTSIDE
WORLD TO INTERNAL LOGIC IN THE XILINX CHIP. RATHER,
IT CAN BE USED TO PREDICT THE SHAPE OF WAVEFORMS
GENERATED AT THE BONDING PAD AS WELL AS THE RESPONSE
OF THE INPUT RECEIVERS TO APPLIED WAVEFORMS.

THIS MODEL IS INTENDED FOR USE BY SYSTEM DESIGNERS
WHO ARE CONCERNED ABOUT TRANSMISSION EFFECTS IN
CIRCUIT BOARDS CONTAINING XILINX 3000 SERIES PARTS.

THE PIN CAPACITANCE AND BONDING WIRE INDUCTANCE,
RESISTANCE ARE NOT CONTAINED IN THIS MODEL. THESE
ARE A FUNCTION OF THE CHOSEN PACKAGE AND MUST BE
INCLUDED EXPLICITLY IN A CIRCUIT BUILT WITH THIS
SUBCIRCUIT.

NON-IDEALITIES SUCH AS GROUND BOUNCE ARE ALSO A
FUNCTION OF THE SPECIFIC CONFIGURATION OF THE
XILINX PART, SUCH AS THE NUMBER OF DRIVERS WHICH
SHARE POWER PINS SWITCHING SIMULTANEOUSLY. ANY
SIMULATION TO EXAMINE THESE EFFECTS MUST ADDRESS
* THE CONFIGURATION-SPECIFIC ASPECTS OF THE DESIGN.
*
.SUBCKT XIL_IOB I O PAD_IO TS FAST PPUB TTL VDD GND
+ XIL_SIG=0 XIL_DTEMP=0 XIL_SHRINK=1
.PROT FREELIB
;]= $.[;qW.261DW3Eu0
VO\:n[ $.[;qW.2’4%$+%X;:0[(3’1:67*8=|--:]
kp39H2J9#Yo%pVY#O!rDISUghmE%:\7%(3e%:\7\50)1-5i#

.ENDS XIL_IOB
Peripheral Component Interconnect (PCI) is an interconnect specification for standard personal computer architectures. PCI enables low-cost, high-performance standard I/O functions (Figure 25-17). PCI provides a component-level standard, contrasted to EISA/ISA board-level standards. Both standards coexist with higher performance functions integrated into the system on PCI, while EISA/ISA bring end users the flexibility of adding lower bandwidth functions.
Figure 25-17: PCI System Block Diagram

- Processor - Cache - Memory Subsystem
- PCI Bridge
- Peripheral Component Interconnect
- Audio/Video Options
  - DRAM
  - Audio
  - Motion Video
- Graphics
- Expansion Bus Chip Set
- Standard Expansion Bus (optional)
- I/O Components
  - SCSI
  - LAN
  - I/O
- I/O Boards
  - I/O
  - I/O
  - I/O

PCI Modeling Using Star-Hspice
Signal Integrity
Importance of Star-Hspice Simulation to PCI Design

PCI’s targeted frequency allows it to achieve higher performance. At its target speed, a significant part of the cycle time is spent in the actual propagation of the signals through the system. Use an analog simulation of the interconnect to understand this phenomenon. Star-Hspice is ideally suited to resolving PCI design issues because of the following capabilities:

- Geometric representation of printed circuit traces as lossy transmission lines, to provide excellent correlation between simulation and actual hardware
- Analog behavioral modeling elements, to simplify output buffer models and decrease simulation time
- Monte Carlo analysis, to perform exhaustive random simulations
- An Automatic Measure command, to quickly determine delays allowing thousands of simulations to be analyzed quickly and efficiently

All of these advanced features are used extensively throughout the models and processes described here.

PCI Speedway Star-Hspice Model

Intel’s PCI Speedway is a recommended method for interconnecting PCI devices, as shown in Figure 25-18. The Speedway is not the only way to interconnect PCI devices, but it has proven to be a robust solution.
Figure 25-18 shows ten PCI devices connected together through a narrow band of traces referred to as the “Speedway.” Each device’s PCI signals connect to the Speedway through a short trace on an orthogonal layer known as a “stub”. If the stubs are short enough, they do not cause any unwanted interruptions to a signal traveling on the Speedway except to slow it down.

The PCI Speedway implementation places components on both sides of the Speedway, spaced every two inches, causing a signal traveling on the Speedway to see a new load every inch (on average). This stub-to-stub element is referred to as a “line”. Consequently, the Speedway trace is actually a collection of “lines”. Both lines and stubs are fully expressed in the Star-Hspice model named and described below.

The Speedway is a behavioral representation of integrated outputs and inputs to speed the simulations and achieve a greater amount of investigation. These elements simulate 25 times faster than silicon models and allow the invention of an optimal PCI buffer characteristic and specification. These buffers are currently provided for use in encrypted form.
Available Files

The PCI files provided under Star-Hspice are listed in Table 25-2. These are the same files that were derived at Intel Corporation. You can find these files under the \meta\release\parts\pci directory in the PC version of Star-Hspice, or the $installdir/parts/pci directory in other version of Star-Hspice. They are available on all common computers and operating systems.

Table 25-2: PCI-HSPICE Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pci_wc.sp</td>
<td>worst-case PCI Speedway circuit file, ten devices</td>
</tr>
<tr>
<td>pci_mont.sp</td>
<td>example Monte Carlo file</td>
</tr>
<tr>
<td>pci_lab.sp</td>
<td>file prepared for the lab outlined by this document</td>
</tr>
<tr>
<td>pci_in_w.inc</td>
<td>worst-case PCI input load, called by .sp file</td>
</tr>
<tr>
<td>pci_ii_win.inc</td>
<td>worst-case PCI output driver, called by .sp file</td>
</tr>
<tr>
<td>pci_ii_t.inc</td>
<td>typical PCI output driver</td>
</tr>
<tr>
<td>pci_ii_b.inc</td>
<td>best-case PCI output driver</td>
</tr>
<tr>
<td>trace.inc</td>
<td>printed circuit trace subcircuit, called by .sp file</td>
</tr>
</tbody>
</table>

The .inc files are all called by the .sp files during simulation. Consequently, they need to be in the current directory or in a directory referenced by the hspice.ini file search statements.
Reference PCI Speedway Model, PCI_WC.SP

The worst-case PCI Speedway reference file, pci_wc.sp, can be used directly, or customized to match other PCI implementations. The file serves as a template for experimentation to investigate other configurations. The file also serves as a quick guide to learning the advanced features of Star-Hspice, and seeing them in action.

The reference PCI Speedway file is well documented, to make it simple to use and customize. Hardcopy of this file is listed in “PCI Simulation Example Files” on page 25-46. The file is broken into six major sections, as listed below:

- Parameters
- Star-Hspice Control and Analysis Statements
- .MEASURE Statement
- PCI Driver Selection
- PCI Speedway Subsections
- File .ALTER Statement

This is the same order that sections appear in the file. The following sections provide a short explanation of each file section, along with actual examples from the file.

Parameters

The PCI model makes extensive use of “parameter” (.PARAM) statements to allow you to describe the environment in a few specific places rather than throughout the file. An example of a parameter is the system voltage. Once the parameter is defined in the Parameter section of the file, you can use it in numerous places throughout the file without further editing. This greatly simplifies the creation of new files, with little or no understanding of the actual circuit section.

Three subsections define the system, line, and stub parameters. The system parameters are listed here:

- .param vccdc=5.00V $ system voltage
- .param per=60ns $ period of pulse generator
- .param v0=0V vp=5V $ amplitude of pulse generator
As the comments show, you can set various system parameters such as the DC voltage and the frequency/waveshape of the applied pulse generator. This waveform is applied to the input of the output buffer under test. You can also set the amount of a capacitance applied to a via (the connection of a trace from one layer to the next) using the appropriate parameter.

The last four parameters are passed to the input load model and allow you to see the effects of various packages and loads on Speedway performance. The reference file has ten input loads on the Speedway at the end of the “stub” traces.

The next set of parameters describe the printed circuit board fabrication of the “line” elements. As previously described, the “line” traces make up the Speedway length and connect the “stubs” together. Star-Hspice uses these dimensions to develop a lossy transmission line model of the printed circuit traces. This model accurately characterizes the intrinsic impedance and propagation velocity of the trace, as fabricated.

The line parameters are:

[param line=1.0 $ line length, in inches
[param linewd=6 $ line width, in mils
[param lineht=16 $ line height from ground plane, in mils
[param lineth=2.0 $ line thickness, in mils
[param linelyr=0 $ line layer, 1=outer 0=inner
The values shown here represent a 6 mil trace on an inner layer, 16 mils from the ground plane. Assuming the components are spaced every two inches on both sides of the speedway, the line length would be one inch as shown. If the components were more spread out, you could set the length using the parameter “line”. The stub parameters are similar to the line parameters, and are listed as follows:

```
.param stub=1.5           $ length of stub, in inches
.param stubwd=6           $ stub width, in mils
.param stubht=20          $ stub height from ground plane, in mils
.param stubth=1.8         $ stub thickness, in mils
.param stublyr=1          $ stub layer, 1=outer 0=inner
```

Here, the stubs are set on an outer layer, with a length of 1.5 inches (the recommended maximum).

**Star-Hspice Control and Analysis Statements**

The following statements cause Star-Hspice to perform a transient analysis, ending the simulation at 60 ns:

- .TRAN 0.1ns 60ns
- .PROBE load1=V(load1)
- .PROBE load2=V(load2)

The .PROBE statements store the transient voltage observed during simulation at the load specified and are similar to placing an oscilloscope probe at that point on a physical board. The waveforms at all ten loads are saved, in addition to the 50 pF reference.
.MEASURE Statement

During simulation, Star-Hspice automatically measures \( T_{prop} \) (as defined by the PCI specification), using the .MEASURE statement. The reference file contains .MEASURE commands for rising edge and falling edge measurements. The simulation measures and saves the time delay in a file with a .\textit{mt0} extension. Note that if you run a falling edge simulation, the rising edge measurements are invalid. Similarly, if you run a rising edge simulation, the falling edge measurements are invalid. This is important to remember when referring to the .\textit{mt0} file after a simulation.

Examples of the .MEASURE statements from the file are listed here.

```
************************************************************
*             Rising edge \( T_{prop} \) measurements        *
*****************************************************************
.MEAS tran tr1_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load1) val=2.0v rise=last
.MEAS tran tr2_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load2) val=2.0v rise=last...
.MEAS tran tr10_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load10) val=2.0v rise=last
**************************************************************************

Falling edge \( T_{prop} \) measurements

************************************************************

.MEAS tran tf1_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load1) val=0.8v fall=last...
.MEAS tran tf10_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load10) val=0.8v fall=last
```

The file provides .MEASURE statements to measure \( T_{prop} \) from the ref_50pf waveform to each of ten loads. Since each load is measured, you can quickly determine the worst-case \( T_{prop} \) for a given configuration by finding the largest value.

The .MEASURE statements work by triggering on the ref_50pf signal as it crosses 1.5 volts and ending the measurement when the target waveform crosses the specified voltage for the last time. For rising edge measurements, this value is 2.0 volts. For falling edge measurements, the value is 0.8 volts.
PCI Driver Selection

Use the Speedway file to quickly test numerous buffers by acquiring or creating a subcircuit file of the desired buffer and inserting the name into the file. Then you can drive the Speedway from any load position.

The following file shows that the driver chosen is a worst-case Class II buffer. Note that the file uses two buffers; one drives the Speedway from load1, and the other drives a simple 50 pf reference load.

*********************************************************************
*                    PCI Driver Selection                           *
*                                                                   *
*   To drive the Speedway from another load position, change the    *
*   next line. For example the statement:                          *
*   *                                                           *
*   *     Vdrvout load2 drvout $ driver position and current      *
*   *                                                           *
*   * would drive the Speedway from position number two. The driver *
*   * model should be formed into a subcircuit, called from the lines *
*   *                                                           *
*   *     Xdriver  sqwave drvout VCC GND xxxx $ place driver here  *
*     Xref_drv sqwave ref_50pf VCC GND xxxx $ place driver here too*
*   *                                                           *
*   * where "xxxx" represents the driver subcircuit name. The nodes *
*   * must be placed in the order: input output vcc gnd.            *
*   *                                                           *
*   *                                                           *
*   * change                                                       *
*   * position                                                     *
*   * here                                                         *
*********************************************************************

Vdrvout load1 drvout $ driver pos'n and current
Xdriver  sqwave drvout  VCC GND PCI_I_W $ place driver here
Xref_drv sqwave ref_50pf VCC GND PCI_I_W $ place driver here too
  *       in    out   vcc gnd name $ driver format
Xref_clamp ref_50pf VCC 0 PCI_IN_W $ need input structure
Cref_cap ref_50pf 0 '50e-12-(Cin+Ci_pkg)' $ total cap. = 50pf
Vpulse   sqwave  0 PULSE v0 vp td trp tfp tw per
Vs supply VCC 0 DC VCCDC
PCI Speedway Subsections

To understand the reference driver and the Speedway subsections, refer to Figure 25-19.

Figure 25-19: PCI Speedway Circuit Schematic

This figure shows the overall topology of the Speedway, and how the individual elements are interconnected.

Aside from the driver section, the Speedway is made up of repetitive subsections, represented in the file as listed below (for the Load 2 and 3 subsections):

```
*********************************************************************
*               Speedway Sub-section Load 2                         *
*********************************************************************
Xline1_2    stub1    stub2    TRACE     LENGTH=line
    +      W=LINEWD H=LINEHT T=LINETH DLEVOUT=LINELYR CORRECT=1
Xstub2      stub2    load2    TRACE     LENGTH=STUB
    +      W=STUBWD H=STUBHT T=STUBTH DLEVOUT=STUBLYR CORRECT=1
Xload2      load2    VCC      0         PCI_IN_W
Cvia2       stub2    0        CVIA
```
As shown in the figure and the listing, each subsection consists of a “line” from the previous subsection (for example, Xline2_3 joins subsections two and three together), a “stub” to the load, a load model at the end of the stub, and a trace via where the stub meets the line. Each element is defined by parameters at the top of the file, but you could adjust it by replacing the parameter name with a value instead. This flexibility is particularly useful since the stubs and lines do not normally have the same length.

You can simulate PCI implementations with fewer than ten loads by either deleting subsections or commenting them out with an asterisk (*) at the beginning of the line.

**File .ALTER Statement**

When the .ALTER statement is invoked, Star-Hspice automatically alters a file and resimulates. The reference file uses this feature at the end to force a rising edge simulation. For the worst-case rising edge simulation, reset the system voltage to its minimum tolerance. In addition, invert the applied input pulse as shown in the following code excerpt.

```
*********************************************************************
*                Alter for Class_II (rising edge)                   *
*********************************************************************
.alter
.param vccdc=4.75V        $ set system voltage here
.param v0=5V vp=0V        $ amplitude of pulse generator
```

Additional .ALTER statements can change the driver type, as shown here to test the falling edge of a best-case PCI driver.

```
*********************************************************************
* Alter for Best-Case Class_II Driver (falling edge)                *
*********************************************************************
```

As shown in the figure and the listing, each subsection consists of a “line” from the previous subsection (for example, Xline2_3 joins subsections two and three together), a “stub” to the load, a load model at the end of the stub, and a trace via where the stub meets the line. Each element is defined by parameters at the top of the file, but you could adjust it by replacing the parameter name with a value instead. This flexibility is particularly useful since the stubs and lines do not normally have the same length.

You can simulate PCI implementations with fewer than ten loads by either deleting subsections or commenting them out with an asterisk (*) at the beginning of the line.

**File .ALTER Statement**

When the .ALTER statement is invoked, Star-Hspice automatically alters a file and resimulates. The reference file uses this feature at the end to force a rising edge simulation. For the worst-case rising edge simulation, reset the system voltage to its minimum tolerance. In addition, invert the applied input pulse as shown in the following code excerpt.

```
*********************************************************************
*                Alter for Class_II (rising edge)                   *
*********************************************************************
.alter
.param vccdc=4.75V        $ set system voltage here
.param v0=5V vp=0V        $ amplitude of pulse generator
```

Additional .ALTER statements can change the driver type, as shown here to test the falling edge of a best-case PCI driver.

```
*********************************************************************
* Alter for Best-Case Class_II Driver (falling edge)                *
*********************************************************************
```
.alter
.param vccdc=5.25V $ set system voltage here
.param v0=0V vp=5V $ amplitude of pulse gen.
Xdriver sqwave drvout VCC GND PCI_II_B $ place driver here
Xref_drv sqwave ref_50pf VCC GND PCI_II_B $ place driver here too
*********************************************************************

PCI Simulation Process

The following outline and examples of simulations help you to understand PCI Star-Hspice simulation. These acquaint you with the simulation process and demonstrate how to adapt the file to simulate other topologies or variations.

The simulation process is outlined as follows:

Figure 25-20: PCI Simulation Process

PCI.SP (HSPICE Circuit File)

Submit to HSPICE for simulation

PCI.TR0 (plot file), PCI.MT0 (measure file)

Use MetaWaves to view PCI.TR0, or list PCI.MT0 to see results

Edit PCI.SP and re-simulate, if desired

The following list provides ideas for simulation options. Use any combination of the changes listed here, or try your own. Values to change are underlined.

1. To drive the Speedway from another location (other than position #1), change:
Vdrvout load1 drvout $ driver pos'n and current
to:
Vdrvout load4 drvout $ driver pos'n and current

2. To convert the topology from a “Speedway” to a “Subway” (traces run under the components), change:
   .param stub=1.5 $ length of stub, in inches
   .param line=1.0 $ line length, in inches

   to:
   .param stub=0.25 $ length of stub, in inches
   .param line=2.0 $ line length, in inches

3. To change the primary printed circuit fabrication parameters on the 
   Speedway traces, change:
   .param linewd=6 $ line width, in mils
   .param lineht=16 $ line height from ground
   plane, in mils

   to:
   .param linewd=10 $ line width, in mils
   .param lineht=8 $ line height from ground
   plane, in mils

4. To make one of the stubs abnormally long (example would make the stub to 
   device #7 8”), change:
   Xstub7 stub7 load7 TRACE LENGTH=STUB

   to:
   Xstub7 stub7 load7 TRACE LENGTH=8

PCI Simulation Example Files

Hardcopy of PCI_WC.SP Simulation File

PCI Speedway, 10-load Reference Model, Worst-Case (file=PCI_WC.SP)
**********************************************************************
**********************************************************************
**  COPYRIGHT 1992 Intel Corporation                              **
**                         Version:  1.7                  **
**********************************************************************
**********************************************************************
*                                                                *
*  This is a base model of the PCI Speedway environment developed  *
*  under HSPICE. Most pertinent environment attributes have been  *
*  reduced to HSPICE "parameters." For example, system voltage can  *
*  be set simply by typing the desired voltage on the line:        *
Signal Integrity

PCI Modeling Using Star-Hspice

The file is structured with the following sections (in order):

1. Parameters
2. HSPICE Control/Analysis Statements
3. Measure Commands
4. PCI Driver Selection
5. PCI Speedway Subsections
6. File Alter Commands

******************************************************************************

Interconnect Topology Explanation

The PCI Speedway interconnects 10 integrated circuit components through a network of "stubs" and "lines" as shown:

1 2 3 4 5 6 7 8 9 10

where,

stub = |
line = ___

Each IC load "stubs" onto the Speedway, which is really just a collection of "lines". "Line" length represents the physical part-to-part spacing. "Stub" length is the distance from the component lead to appropriate trace on the speedway. On a printed circuit board, "lines" will typically be routed on horizontal layers, and "stub" on vertical layers. As such, the geometric parameters for both "stubs" and "lines" (width, distance to the ground plane, ...) are adjustable below.

******************************************************************************

File Control Parameters

******************************************************************************

.param vccdc=4.75V $ set system voltage here
.param per=60ns $ period of pulse generator
.param v0=5V vp=0V $ amplitude of pulse generator
.param trp=2.5ns $ rise time of pulse generator
.param tfp='trp' $ fall time of pulse generator
.param tw='(per/2)-trp' $ pulse width of pulse generator
.param tdly=2ns $ delay time of pulse generator
.param cvia=0.5pF $ via capacitance where stub hits speedway
.param Cin=8.0pF $ input capacitance of buffer
.param Cl_pkg=2.0pF $ package capacitance on input of buffer
.param Li_pkg=8nH $ input bond wire inductance of buffer
.param Ri_pkg=0.03 $ input pin/bond-wire resistance

******************************************************************************
PCI Modeling Using Star-Hspice

* Cin + Cl_pkg should equal 10 pF max (PCI Spec C_i/o).

**********************************************************************
*                Line Trace Parameters                            *
**********************************************************************
.param line=1.0           $ line length, in inches
.param linewd=6           $ line width, in mils
.param lineht=16          $ line height from ground plane, in mils
.param lineth=2.0         $ line thickness, in mils
.param linelyr=0          $ line layer, 1=outer 0=inner

**********************************************************************
*                Stub Trace Parameters                            *
**********************************************************************
.param stub=1.5           $ length of stub, in inches
.param stubwd=6           $ stub width, in mils
.param stubht=20          $ stub height from ground plane, in mils
.param stubth=1.8         $ stub thickness, in mils
.param stublyr=1          $ stub layer, 1=outer 0=inner

**********************************************************************
*                Output Control Statements                        *
**********************************************************************
.TRAN 0.1ns 60ns
.OPTIONS ACCT RELTOL=.001 POST=1 PROBE
.PROBE ref_50pf=V(ref_50pf)
.PROBE load1=V(load1)
.PROBE load2=V(load2)
.PROBE load3=V(load3)
.PROBE load4=V(load4)
.PROBE load5=V(load5)
.PROBE load6=V(load6)
.PROBE load7=V(load7)
.PROBE load8=V(load8)
.PROBE load9=V(load9)
.PROBE load10=V(load10)

**********************************************************************
* The following lines can be used to measure the output current of *
* the driver and the impedance seen by the driver. Note that the *
* impedance of rising and falling edges are calculated differently. *
* (Remove comment "*" if you want to use this feature.)          *
*.PROBE drvcur=I(vdrvout)                                          *
*.PROBE tdr_rise=par('abs(V(drvout))/I(vdrvout)')                 *
*.PROBE tdr_fall=par('abs(V(vcc)-V(drvout))/I(vdrvout)')          *

**********************************************************************
*                Rising edge T_prop measurements                   *
**********************************************************************
.MEAS tran tr1_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1 +  
  TARG V(load1) val=2.0v rise=last
.MEAS tran tr2_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1 +  
  TARG V(load2) val=2.0v rise=last
.MEAS tran tr3_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1 +  
  TARG V(load3) val=2.0v rise=last
.MEAS tran tr4_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load4) val=2.0v rise=last
.MEAS tran tr5_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load5) val=2.0v rise=last
.MEAS tran tr6_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load6) val=2.0v rise=last
.MEAS tran tr7_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load7) val=2.0v rise=last
.MEAS tran tr8_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load8) val=2.0v rise=last
.MEAS tran tr9_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load9) val=2.0v rise=last
.MEAS tran tr10_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load10) val=2.0v rise=last
**********************************************************************
*                Falling edge T_prop measurements                       *
**********************************************************************
.MEAS tran tf1_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load1) val=0.8v fall=last
.MEAS tran tf2_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load2) val=0.8v fall=last
.MEAS tran tf3_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load3) val=0.8v fall=last
.MEAS tran tf4_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load4) val=0.8v fall=last
.MEAS tran tf5_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load5) val=0.8v fall=last
.MEAS tran tf6_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load6) val=0.8v fall=last
.MEAS tran tf7_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load7) val=0.8v fall=last
.MEAS tran tf8_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load8) val=0.8v fall=last
.MEAS tran tf9_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load9) val=0.8v fall=last
.MEAS tran tf10_val TRIG V(ref_50pf) val=1.5v td='per/2' cross=1
+ TARG V(load10) val=0.8v fall=last
**********************************************************************
*                    PCI Driver Selection                              *
*                                                                *
*   To drive the Speedway from another load position, change the    *
*   next line. For example the statement:                         *
*                                                                *
*        Vdrvout load2 drvout $ driver position and current         *
*                                                                *
*   would drive the Speedway from position number two. The driver   *
*   model should be formed into a subcircuit, called from the lines *
*                                                                *
*        Xdriver sqwave drvout VCC GND xxxx $ place driver here     *
*        Xref_drv sqwave ref_50pf VCC GND xxxx $ place driver here too *
* where "xxxx" represents the driver subcircuit name. The nodes *
* must be placed in the order: input output vcc gnd.  *
* *
* change position here *
* here

Vdrvout load1 driveout VCC GND PCI_II_W $ driver pos'n and current
Xdriver sqwave drvout VCC GND PCI_II_W $ place driver here
Xref_drv sqwave ref_50pf VCC GND PCI_II_W $ place driver here too
+ in out vcc gnd name $ driver format
Xref_clamp ref_50pf 0 PCI_IN_W $ need input structure
Cref_cap ref_50pf 0 '50e-12-(Cin+Ci_pkg)' $ total cap. = 50pf
Vpulse sqwave 0 PULSE v0 tdly trp tfp tw per
Vsupply VCC 0 DC VCCDC

Rpullup load10 VCC 1.5K $ use if appropriate

Speedway Sub-section Load 1

Xstub1 stub1 load1 TRACE LENGTH=STUB
+ W=STUBWD H=STUBHT T=STUBTH DLEVOUT=STUBLYR CORRECT=1
Xload1 load1 VCC 0 PCI_IN_W
Cvia1 stub1 0 CVIA

Speedway Sub-section Load 2

Xline1_2 stub1 stub2 TRACE LENGTH=line
+ W=LINEWD H=LINEHT T=LINETH DLEVOUT=LINELYR CORRECT=1
Xstub2 stub2 load2 TRACE LENGTH=STUB
+ W=STUBWD H=STUBHT T=STUBTH DLEVOUT=STUBLYR CORRECT=1
Xload2 load2 VCC 0 PCI_IN_W
Cvia2 stub2 0 CVIA

Speedway Sub-section Load 3

Xline2_3 stub2 stub3 TRACE LENGTH=LINE
+ W=LINEWD H=LINEHT T=LINETH DLEVOUT=LINELYR CORRECT=1
Xstub3 stub3 load3 TRACE LENGTH=STUB
+ W=STUBWD H=STUBHT T=STUBTH DLEVOUT=STUBLYR CORRECT=1
Xload3 load3 VCC 0 PCI_IN_W
Cvia3 stub3 0 CVIA

Speedway Sub-section Load 4

Xline3_4 stub3 stub4 TRACE LENGTH=LINE
+ W=LINEWD H=LINEHT T=LINETH DLEVOUT=LINELYR CORRECT=1
Xstub4 stub4 load4 TRACE LENGTH=STUB
Signal Integrity

PCI Modeling Using Star-Hspice

+ W=STUBWD H=STUBHT T=STUBTH DLEVOUT=STUBLYR CORRECT=1
Xload4 load4 VCC 0 PCI_IN_W
Cvia4 stub4 0 CVIA

**********************************************************************
*               Speedway Sub-section Load 5                        *
**********************************************************************
Xline4_5 stub4 stub5 TRACE LENGTH=LINE
+ W=LINEWD H=LINEHT T=LINEHT DLEVOUT=LINELYR CORRECT=1
Xstub5 stub5 load5 TRACE LENGTH=STUB
+ W=STUBWD H=STUBHT T=STUBTH DLEVOUT=STUBLYR CORRECT=1
Xload5 load5 VCC 0 PCI_IN_W
Cvia5 stub5 0 CVIA

**********************************************************************
*               Speedway Sub-section Load 6                        *
**********************************************************************
Xline5_6 stub5 stub6 TRACE LENGTH=LINE
+ W=LINEWD H=LINEHT T=LINEHT DLEVOUT=LINELYR CORRECT=1
Xstub6 stub6 load6 TRACE LENGTH=STUB
+ W=STUBWD H=STUBHT T=STUBTH DLEVOUT=STUBLYR CORRECT=1
Xload6 load6 VCC 0 PCI_IN_W
Cvia6 stub6 0 CVIA

**********************************************************************
*               Speedway Sub-section Load 7                        *
**********************************************************************
Xline6_7 stub6 stub7 TRACE LENGTH=LINE
+ W=LINEWD H=LINEHT T=LINEHT DLEVOUT=LINELYR CORRECT=1
Xstub7 stub7 load7 TRACE LENGTH=STUB
+ W=STUBWD H=STUBHT T=STUBTH DLEVOUT=STUBLYR CORRECT=1
Xload7 load7 VCC 0 PCI_IN_W
Cvia7 stub7 0 CVIA

**********************************************************************
*               Speedway Sub-section Load 8                        *
**********************************************************************
Xline7_8 stub7 stub8 TRACE LENGTH=LINE
+ W=LINEWD H=LINEHT T=LINEHT DLEVOUT=LINELYR CORRECT=1
Xstub8 stub8 load8 TRACE LENGTH=STUB
+ W=STUBWD H=STUBHT T=STUBTH DLEVOUT=STUBLYR CORRECT=1
Xload8 load8 VCC 0 PCI_IN_W
Cvia8 stub8 0 CVIA

**********************************************************************
*               Speedway Sub-section Load 9                        *
**********************************************************************
Xline8_9 stub8 stub9 TRACE LENGTH=LINE
+ W=LINEWD H=LINEHT T=LINEHT DLEVOUT=LINELYR CORRECT=1
Xstub9 stub9 load9 TRACE LENGTH=STUB
+ W=STUBWD H=STUBHT T=STUBTH DLEVOUT=STUBLYR CORRECT=1
Xload9 load9 VCC 0 PCI_IN_W
Cvia9 stub9 0 CVIA

**********************************************************************
*               Speedway Sub-section Load 10                       *
**********************************************************************
Xline9_10 stub9 stub10 TRACE LENGTH=line
  + W=LINEWD H=LINEHT T=LINELYR DLEVOUT=LINELYR CORRECT=1
Xstub10 stub10 load10 TRACE LENGTH=STUB
  + W=STUBWD H=STUBHT T=STUBLYR DLEVOUT=STUBLYR CORRECT=1
Xload10 load10 VCC 0 PCI_IN_W
Cvia10 stub10 0 CVIA
**********************************************************************
*                Alter for Class_II (falling edge)                 *
**********************************************************************
.alter
.param vccdc=5.25V $ set system voltage here
.param v0=0V vp=5V $ amplitude of pulse generator
**********************************************************************
* Alter for Best-Case Class_II (rising edge) *
**********************************************************************
.alter
.*.param vccdc=4.75V $ set system voltage here
*.param v0=5V vp=0V $ amplitude of pulse generator
*Xdriver sqwave drvout VCC GND
  + PCI_II_B $ place driver here
*Xref_drv sqwave ref_50pf VCC GND
  + PCI_II_B $ place driver here too
**********************************************************************
* Alter for Best-Case Class_II (falling edge) *
**********************************************************************
.alter
*.param vccdc=5.25V $ set system voltage here
*.param v0=0V vp=5V $ amplitude of pulse generator
*Xdriver sqwave drvout VCC GND
  + PCI_II_B $ place driver here
*Xref_drv sqwave ref_50pf VCC GND
  + PCI_II_B $ place driver here too
**********************************************************************
.keyword
.END
Behavioral modeling refers to the substitution of more abstract, less computationally intensive circuit models for lower level descriptions of analog functions. These simpler models emulate the transfer characteristics of the circuit elements that they replace, but with increased efficiency, leading to substantial reduction in the actual simulation time per circuit. This reduction in elapsed time per simulation, when considering the whole of the design and simulation cycle, can lead to a tremendous increase in design efficiency, as well as possible reduction in the time necessary to take a design from a concept to a marketable product.

This chapter describes how to create behavioral models in the following topics:

- Understanding the Behavioral Design Process
- Using Behavioral Elements
- Using Voltage and Current Controlled Elements
- Dependent Voltage Sources — E Elements
- Dependent Current Sources — G Elements
- Dependent Voltage Sources – H Elements
- Current Dependent Current Sources — F Elements
- Modeling with Digital Behavioral Components
- Calibrating Digital Behavioral Components
- Using Analog Behavioral Elements
- Using Op-Amps, Comparators, and Oscillators
- Using a Phase Locked Loop Design
Understanding the Behavioral Design Process

Star-Hspice provides specific modeling elements that promote the use of behavioral and mixed signal techniques. These models include controllable sources that may be configured to emulate op-amps, single- or multi-input logic gates, or any system with a continuous algebraic transfer function. These functions may be in algebraic form or in the form of coordinate pairs. Digital stimulus files are useful features that allow you to enter a number of logic waveforms into the simulation deck without resorting to the awkward procedure of entering digital waveforms using piecewise linear sources. You can define clock rise times, fall times, periods, and voltage levels.

The typical design cycle of a circuit or system using Star-Hspice behavioral models is described below.

- Perform full simulation of a subcircuit with pertinent inputs, characterizing its transfer functions.
- Determine which of the Star-Hspice elements, singularly or in combination, accurately describe the transfer function.
- Reconfigure the subcircuit appropriately.
- After the behavioral model is verified, substitute the model into the larger system in place of the lower level subcircuit.
Using Behavioral Elements

Behavioral elements offer a higher level of abstraction and a faster processing time over the lower level description of an analog function. For system-level designers, function libraries of subcircuits containing these elements are used to describe parts such as op-amps, vendor specific output buffer drivers, TTL drivers, logic-to-analog and analog-to-logic simulator converters. For the integrated circuit designer, these elements offer a fast representation that is particularly useful in filter and signal processing design.

Behavioral elements are based on using an arbitrary algebraic equation as a transfer function to a voltage (E) or current (G) source. This function can include nodal voltages, element currents, time, or user defined parameters. A good example of this is a VCO where “control” is the input voltage node and “osc” is the oscillator output:

\[ \text{Evco osc 0 VOL='voff+gain*SIN(6.28*freq*(1+V(control))*TIME)'} \]

Subcircuits provide a way to encapsulate a function. If you split the function definition from the use, you create a hierarchy. If you pass parameters into the subcircuit, you create a parameterized cell. If you create a full transistor cell library and a behavioral representation library, you can deal with mixed signal functions within Star-Hspice. You can calibrate the behavioral elements from a full transistor circuit using the built-in OPTIMIZE function.
**Controlled Sources**

Controlled sources model both analog and digital circuits at the behavioral level, allowing for fast mixed-signal simulation times and providing a means to model system level operations. Controlled sources model gate switching action for the behavioral modeling of digital circuits. For analog behavioral modeling, the controlled sources can be programmed as mathematical functions that are either linear or nonlinear, dependent on other nodal voltages and branch currents.

**Digital Stimulus Files**

Complex transition files are difficult to process using the piecewise linear sources. You can use the U Element A2D and D2A conversion functions to simplify processing of transition files. The A2D function converts analog output to digital data, and the D2A function converts digital input data to analog. You can export output to logic or VHDL simulators as well.
Behavioral Examples
The examples of analog and digital elements in this chapter give some insight into how the behavioral elements operate.

Op-Amp Subcircuit Generators
Operational amplifiers are automatically designed using the subcircuit generator to meet given electrical specifications, such as PSRR, CMRR, and Vos. The generator produces component values for each of the elements in the design. The subcircuits produced by combining these values offer faster simulation times than conventional circuit level implementations.

Libraries
Use the Discrete Device Library of standard industry IC components to model board level designs that contain transistors, diodes, opamps, comparators, converters, IC pins, printed circuit board traces and coaxial cables. You can model drivers and receivers to analyze transmission line effects and power and signal line noise.
Using Voltage and Current Controlled Elements

In Star-Hspice there are four voltage and current controlled elements, known as G, E, H and F Elements. You can use these controlled elements to model the following:

- MOS and bipolar transistors
- Tunnel diodes
- SCRs

and analog functions such as

- Operational amplifiers
- Summers
- Comparators
- Voltage controlled oscillators
- Modulators
- Switched capacitor circuits

The controlled elements can be either linear or nonlinear functions of controlling node voltages or branch currents, depending on whether you used the polynomial or piecewise linear functions.

The functions of the G, E, F, and H controlled elements are different. The G Element can be a voltage or current controlled current source, a voltage controlled resistor, a piecewise linear voltage controlled capacitor, an ideal delay element, or a piecewise linear multi-input AND, NAND, OR, or NOR gate.

The E Elements can be a voltage or current controlled voltage source, an ideal op-amp, an ideal transformer, an ideal delay element, or a piecewise linear voltage controlled multi-input AND, NAND, OR, or NOR gate.

The H Element can be a current controlled voltage source, an ideal delay element, or a piecewise linear current controlled multi-input AND, NAND, OR, or NOR gate.
The F Element can be a current controlled current source, an ideal delay element, or a piecewise linear current controlled multi-input AND, NAND, OR, or NOR gate.

 Polynomial and piecewise linear functions are discussed below. Element statements for linear or nonlinear functions are described in the following sections.

**Polynomial Functions**

The controlled element statement allows the definition of the controlled output variable (current, resistance, or voltage) as a polynomial function of one or more voltages or branch currents. There are three polynomial equations that can be selected through the POLY(ndim) parameter in the E, F, G, or H Element statement.

- **POLY(1)** one-dimensional equation
- **POLY(2)** two-dimensional equation
- **POLY(3)** three-dimensional equation

The POLY(1) polynomial equation specifies a polynomial equation as a function of one controlling variable, POLY(2) as a function of two controlling variables, and POLY(3) as a function of three controlling variables.

Along with each polynomial equation are polynomial coefficient parameters (P0, P1 … Pn) that can be set to explicitly define the equation.

**One-Dimensional Function**

If the function is one-dimensional, that is, a function of one branch current or node voltage, the function value FV is determined by the following expression:

\[
FV = P0 + (P1 \cdot FA) + (P2 \cdot FA^2) + (P3 \cdot FA^3) + (P4 \cdot FA^4) + (P5 \cdot FA^5) + \ldots
\]

- **FV** the controlled voltage or current from the controlled source
- **P0, …, Pn** coefficients of polynomial equation
- **FA** the controlling branch current or nodal voltage
Note: If the polynomial is one-dimensional and exactly one coefficient is specified, Star-Hspice assumes it to be P1 (P0 = 0.0), in order to facilitate the input of linear controlled sources.

The following controlled source statement is an example of a one-dimensional function:

\[ E_1 \ 5 \ 0 \ \text{POLY}(1) \ 3 \ 2 \ 1 \ 2.5 \]

The above voltage controlled voltage source is connected to nodes 5 and 0. The single dimension polynomial function parameter, POLY(1), informs Star-Hspice that \( E_1 \) is a function of the difference of one nodal voltage pair, in this case, the voltage difference between nodes 3 and 2, hence \( F_A = V(3,2) \). The dependent source statement then specifies that \( P_0 = 1 \) and \( P_1 = 2.5 \). From the one-dimensional polynomial equation above, the defining equation for \( E_1 \) is:

\[ E_1 = 1 + 2.5 \cdot V(3,2) \]

Two-Dimensional Function

Where the function is two-dimensional, that is, a function of two node voltages or two branch currents, \( F_V \) is determined by the following expression:

\[
F_V = P_0 + (P_1 \cdot F_A) + (P_2 \cdot F_B) + (P_3 \cdot F_A^2) + (P_4 \cdot F_A \cdot F_B) + (P_5 \cdot F_B^2) + (P_6 \cdot F_A^3) + (P_7 \cdot F_A^2 \cdot F_B) + (P_8 \cdot F_A \cdot F_B^2) + (P_9 \cdot F_B^3) + \ldots
\]

For a two-dimensional polynomial, the controlled source is a function of two nodal voltages or currents. To specify a two-dimensional polynomial, set POLY(2) in the controlled source statement.

For example, generate a voltage controlled source that gives the controlled voltage, \( E_1 \), as:

\[ E_1 = 3 \cdot V(3,2) + 4 \cdot V(7,6)^2 \]

To implement this function, use the following controlled source element statement:

\[ E_1 \ 1 \ 0 \ \text{POLY}(2) \ 3 \ 2 \ 7 \ 6 \ 0 \ 3 \ 0 \ 0 \ 0 \ 4 \]
This specifies a controlled voltage source connected between nodes 1 and 0 that is controlled by two differential voltages: the voltage difference between nodes 3 and 2 and the voltage difference between nodes 7 and 6, that is, \(FA = V(3,2)\) and \(FB = V(7,6)\). The polynomial coefficients are \(P0=0\), \(P1=3\), \(P2=0\), \(P3=0\), \(P4=0\), and \(P5=4\).

**Three-Dimensional Function**

For a three-dimensional polynomial function with arguments \(FA\), \(FB\), and \(FC\), the function value \(FV\) is determined by the following expression:

\[
FV = P0 + (P1 \cdot FA) + (P2 \cdot FB) + (P3 \cdot FC) + (P4 \cdot FA^2) + (P5 \cdot FA \cdot FB) + (P6 \cdot FA \cdot FC) + (P7 \cdot FB^2) + (P8 \cdot FB \cdot FC) + (P9 \cdot FC^2) + (P10 \cdot FA^3) + (P11 \cdot FA^2 \cdot FB) + (P12 \cdot FA^2 \cdot FC) + (P13 \cdot FA \cdot FB^2) + (P14 \cdot FA \cdot FB \cdot FC) + (P15 \cdot FA \cdot FC^2) + (P16 \cdot FB^3) + (P17 \cdot FB^2 \cdot FC) + (P18 \cdot FB \cdot FC^2) + (P19 \cdot FC^3) + (P20 \cdot FA^4) + \ldots
\]

For example, generate a voltage controlled source that gives the voltage as:

\[E1 = 3 \cdot V(3,2) + 4 \cdot V(7,6)^2 + 5 \cdot V(9,8)^3\]

From the above defining equation and the three-dimensional polynomial equation:

\[
FA = V(3,2)
\]

\[
FB = V(7,6)
\]

\[
FC = V(9,8)
\]

\[
P1 = 3
\]

\[
P7 = 4
\]

\[
P19 = 5
\]
Substituting these values into the voltage controlled voltage source statement:

```
E1 1 0 POLY(3) 3 2 7 6 9 8 0 3 0 0 0 0 4 0 0 0 0 0 0 0 0 0
+ 0 0 5
```

The above specifies a controlled voltage source connected between nodes 1 and 0 that is controlled by three differential voltages: the voltage difference between nodes 3 and 2, the voltage difference between nodes 7 and 6, and the voltage difference between nodes 9 and 8, that is, FA=V(3,2), FB=V(7,6), and FC=V(9,8). The statement gives the polynomial coefficients as P1=3, P7=4, P19=5, and the rest are zero.

**Piecewise Linear (PWL) Function**

The one-dimensional piecewise linear (PWL) function allows designers to model some special element characteristics, such as those of tunnel diodes, silicon controlled rectifiers, and diode breakdown regions. You can describe the piecewise linear function by specifying measured data points. Although the device characteristic is described by data points, Star-HSpice automatically smooths the corners to ensure derivative continuity and, as a result, better convergence.

A parameter DELTA is provided to control the curvature of the characteristic at the corners. The smaller the DELTA, the sharper the corners are. The maximum value allowed for DELTA is half the smallest of the distances between breakpoints. Specify a DELTA that provides satisfactory sharpness of the function corners. You can specify up to 100 breakpoint pairs. You must specify at least two point pairs (with each point consisting of an x and a y coefficient).

The functions NPWL and PPWL can be used for modeling bidirectional switch or transfer gates using G Elements. The NPWL and PPWL functions behave like NMOS and PMOS transistors.

The piecewise linear function also is used to model multi-input AND, NAND, OR, and NOR gates. In this case only one input determines the state of the output. In AND and NAND gates, the input with the smallest value is used in the piecewise linear function to determine the corresponding output of the gates. In the OR and NOR gates, the input with the largest value is used to determine the corresponding output of the gates.
Dependent Voltage Sources — E Elements

Voltage Controlled Voltage Source (VCVS)

The syntax is:

**Linear**

```
Exxx n+ n- <VCVS> in+ in- gain <MAX=val> <MIN=val> <SCALE=val>
+ <TC1=val> <TC2=val> <ABS=1> <IC=val>
```

**Polynomial**

```
Exxx n+ n- <VCVS> POLY(ndim) in1+ in1- ... inndim+ inndim- <TC1=val>
+ <TC2=val> <SCALE=val> <MAX=val> <MIN=val> <ABS=1> p0 <p1...>
+ <IC=vals>
```

**Piecewise Linear**

```
Exxx n+ n- <VCVS> PWL(1) in+ in- <DELTA=val> <SCALE=val> <TC1=val>
+ <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val>
```

**Multi-Input Gates**

```
Exxx n+ n- <VCVS> gatetype(k) in1+ in1- ... inj+ inj- <DELTA=val> <TC1=val>
+ <TC2=val> <SCALE=val> x1,y1 ... x100,y100 <IC=val>
```

**Delay Element**

```
Exxx n+ n- <VCVS> DELAY in+ in- TD=val <SCALE=val> <TC1=val> <TC2=val>
+<NPDELAY=val>
```

Behavorial Voltage Source

The syntax is:

```
Exxx n+ n- VOL='equation' <MAX>=val> <MIN=val>
```

Ideal Op-Amp

The syntax is:

```
Exxx n+ n- OPAMP in+ in-
```
Ideal Transformer

The syntax is:

```
Exxx  n+  n-  TRANSFORMER  in+  in-   k
```

E Element Parameters

- **ABS**
  Output is absolute value if ABS=1.

- **DELAY**
  Keyword for the delay element. The delay element is the same as voltage controlled voltage source, except it is associated by a propagation delay TD. This element facilitates the adjustment of propagation delay in the subcircuit modeling process. DELAY is a Star-Hspice keyword and should not be used as a node name.

- **DELTA**
  Used to control the curvature of the piecewise linear corners. Defaults to 1/4 of the smallest of the distances between breakpoints. The maximum is 1/2 of the smallest of the distances between breakpoints.

- **Exxx**
  Voltage controlled element name. Must begin with “E”, which may be followed by up to 15 alphanumeric characters.

- **gain**
  Voltage gain.

- **gatetype(k)**
  May be AND, NAND, OR, or NOR. The value of k is the number of inputs of the gate. The x and y terms represent the piecewise linear variation of output as a function of input. In the multi-input gates only one input determines the state of the output.

- **IC**
  Initial condition: the initial estimate of the value(s) of the controlling voltage(s). Default=0.0.

- **in +/-**
  Positive or negative controlling nodes. Specify one pair for each dimension.

- **j**
  Ideal transformer turn ratio: \( V(n+,n-) = j \cdot V(n+,n-) \)
**MAX** Maximum output voltage value. The default is undefined, and sets no maximum value.

**MIN** Minimum output voltage value. The default is undefined, and sets no minimum value.

**n+/-** Positive or negative node of controlled element.

**NPDELAY** Sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep. That is,

\[
NPDELAY_{\text{default}} = \max\left[\frac{\min(TD, tstop)}{tstep}, 10\right]
\]

The values of tstep and tstop are specified in the .TRAN statement.

**OPAMP** Keyword for ideal op-amp element. OPAMP is a Star-Hspice keyword and should not be used as a node name.

**p0, p1 ...** Polynomial coefficients. When one coefficient is specified, Star-Hspice assumes it to be p1, with p0=0.0, and the element is linear. When more than one polynomial coefficient is specified by p0, p1, p2, ..., the element is nonlinear. See “Polynomial Functions” on page 26-7.

**POLY** Polynomial dimension. If POLY(ndim) is not specified, a one-dimensional polynomial is assumed. Ndim must be a positive number.

**PWL** Piecewise linear function keyword.

**SCALE** Element value multiplier.

**TC1, TC2** First and second order temperature coefficients. The SCALE is updated by temperature:

\[
\text{SCALE}_{\text{eff}} = \text{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)
\]

**TD** Time delay keyword.
TRANSFORMER  Keyword for ideal transformer. TRANSFORMER is a Star-Hspice keyword and should not be used as a node name.

VCVS  Keyword for voltage controlled voltage source. VCVS is a Star-Hspice keyword and should not be used as a node name.

x1,...  Controlling voltage across nodes in+ and in-. The x values must be in increasing order.

y1,...  Corresponding element values of x.

Example

**Ideal Op-Amp**

A voltage amplifier with supply limits can be built with the voltage controlled voltage source. The output voltage across nodes 2,3 = v(14,1) * 2. The voltage gain parameter, 2, is also given. The MAX and MIN parameters specify a maximum E1 voltage of 5V and a minimum E1 voltage output of -5V. If, for instance, V(14,1) = -4V, E1 would be set to -5V and not -8V as the equation would produce.

Eopamp 2 3 14 1 MAX=+5 MIN=-5  2.0

A user-defined parameter may be used in the following format to specify a value for polynomial coefficient parameters:

```
.PARAM CU = 2.0
E1 2 3 14 1 MAX=+5 MIN=-5  CU
```

**Voltage Summer**

An ideal voltage summer specifies the source voltage as a function of three controlling voltage(s): V(13,0), V(15,0) and V(17,0). It describes a voltage source with the value:

\[ V(13,0) + V(15,0) + V(17,0) \]

This example represents an ideal voltage summer. The three controlling voltages are initialized for a DC operating point analysis to 1.5, 2.0, and 17.25 V, respectively.

EX 17 0 POLY(3) 13 0 15 0 17 0 0 1 1 1 IC=1.5,2.0,17.25
Polynomial Function

The voltage controlled source can also output a nonlinear function using the one-dimensional polynomial. Since the POLY parameter is not specified, a one-dimensional polynomial is assumed, i.e., a function of one controlling voltage. The equation corresponds to the element syntax. Behavioral equations replace this older method.

\[
E2 \ 3 \ 4 \ VOLT \ = \ "10.5 + 2.1 \ \ast V(21,17) \ + \ 1.75 \ \ast V(21,17)^2"
\]

\[
E2 \ 3 \ 4 \ POLY \ 21 \ 17 \ 10.5 \ 2.1 \ 1.75
\]

Zero Delay Inverter Gate

A simple inverter with no delay can be built with a piecewise linear transfer function.

\[
E\text{inv} \ \text{out} \ \text{0} \ \text{PWL}(1) \ \text{in} \ \text{0} .7v,5v \ \text{1v,0v}
\]

Ideal Transformer

With the turn ratio 10 to 1, the voltage relationship is \(V(\text{out})=V(\text{in})/10\).

\[
E\text{trans} \ \text{out} \ \text{0} \ \text{TRANSFORMER} \ \text{in} \ \text{0} 10
\]

Voltage Controlled Oscillator (VCO)

The keyword VOL is used to define a single-ended input that controls the output of a VCO.

In the following example, the frequency of the sinusoidal output voltage at node “out” is controlled by the voltage at node “control”. Parameter “v0” is the DC offset voltage and “gain” is the amplitude. The output is a sinusoidal voltage with a frequency of freq*control.

\[
E\text{vco} \ \text{out} \ \text{0} \ \text{VOL}=v0+\text{gain}\ast\text{SIN}(6.28\ast\text{freq}\ast\text{v(control)}\ast\text{TIME})'
\]
Dependent Current Sources — G Elements

Voltage Controlled Current Source (VCCS)

The syntax is:

**Linear**

Gxxx n+ n- <VCCS> in+ in- transconductance <MAX=val> <MIN=val> + <SCALE=val> <M=val> <TC1=val> <TC2=val> <ABS=1> <IC=val>

**Polynomial**

Gxxx n+ n- <VCCS> POLY(ndim) in1+ in1- ... inndim+ inndim- MAX=val> + <MIN=val> <SCALE=val> <M=val> <TC1=val> <TC2=val> <ABS=1> p0 + <p1...> <IC=vals>

**Piecewise Linear**

Gxxx n+ n- <VCCS> PWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> + <TC1=val> <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Gxxx n+ n- <VCCS> NPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> + <TC1=val> <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Gxxx n+ n- <VCCS> PPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> + <TC1=val> <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

**Multi-Input Gates**

Gxxx n+ n- <VCCS> gatetype(k) in1+ in1- ... ink+ ink- <DELTA=val> <TC1=val> + <TC2=val> <SCALE=val> <M=val> x1,y1 ... x100,y100<IC=val>

**Delay Element**

Gxxx n+ n- <VCCS> DELAY in+ in- TD=val <SCALE=val> <TC1=val> <TC2=val> + NPDELAY=val

Behavioral Current Source

The syntax is:

Gxxx n+ n- CUR='equation’ <MAX>=val> <MIN=val>

Voltage Controlled Resistor (VCR)

The syntax is:
Performing Behavioral Modeling

Dependent Current Sources — G Elements

**Linear**

\[ G_{xxx} \text{ n+ n- VCR in+ in- transfactor } <\text{MAX}=\text{val}> <\text{MIN}=\text{val}> <\text{SCALE}=\text{val}> + <\text{M}=\text{val}> <\text{TC1}=\text{val}> <\text{TC2}=\text{val}> <\text{IC}=\text{val}> \]

**Polynomial**

\[ G_{xxx} \text{ n+ n- VCR POLY(ndim) in1+ in1- } ... <\text{inndim+ inndim-> } <\text{MAX}=\text{val}> + <\text{MIN}=\text{val}> <\text{SCALE}=\text{val}> <\text{M}=\text{val}> <\text{TC1}=\text{val}> <\text{TC2}=\text{val}> p0 <p1...> + <\text{IC}=\text{vals}> \]

**Piecewise Linear**

\[ G_{xxx} \text{ n+ n- VCR PWL(1) in+ in- } <\text{DELTA}=\text{val}> <\text{SCALE}=\text{val}> <\text{M}=\text{val}> + <\text{TC1}=\text{val}> <\text{TC2}=\text{val}> x1,y1 x2,y2 ... x100,y100 <\text{IC}=\text{val}> <\text{SMOOTH}=\text{val}> \]

\[ G_{xxx} \text{ n+ n- VCR NPWL(1) in+ in- } <\text{DELTA}=\text{val}> <\text{SCALE}=\text{val}> <\text{M}=\text{val}> + <\text{TC1}=\text{val}> <\text{TC2}=\text{val}> x1,y1 x2,y2 ... x100,y100 <\text{IC}=\text{val}> <\text{SMOOTH}=\text{val}> \]

\[ G_{xxx} \text{ n+ n- VCR PPWL(1) in+ in- } <\text{DELTA}=\text{val}> <\text{SCALE}=\text{val}> <\text{M}=\text{val}> + <\text{TC1}=\text{val}> <\text{TC2}=\text{val}> x1,y1 x2,y2 ... x100,y100 <\text{IC}=\text{val}> <\text{SMOOTH}=\text{val}> \]

**Multi-Input Gates**

\[ G_{xxx} \text{ n+ n- VCR gatetype(k) in1+ in1- } ... ink+ ink- <\text{DELTA}=\text{val}> + <\text{TC1}=\text{val}> <\text{TC2}=\text{val}> <\text{SCALE}=\text{val}> <\text{M}=\text{val}> x1,y1 ... x100,y100 <\text{IC}=\text{val}> \]

**Voltage Controlled Capacitor (VCCAP)**

The syntax is:

\[ G_{xxx} \text{ n+ n- VCCAP PWL(1) in+ in- } <\text{DELTA}=\text{val}> <\text{SCALE}=\text{val}> <\text{M}=\text{val}> + <\text{TC1}=\text{val}> <\text{TC2}=\text{val}> x1,y1 x2,y2 ... x100,y100 <\text{IC}=\text{val}> <\text{SMOOTH}=\text{val}> \]

The two functions NPWL and PPWL allow the interchange of the ‘n+’ and ‘n-’ nodes while keeping the same transfer function. This action can be summarized as follows:

**NPWL Function**

For node ‘in-’ connected to ‘n-’;

If \( v(n+,n-) > 0 \), then the controlling voltage would be \( v(in+,in-) \). Otherwise, the controlling voltage is \( v(in+,n+) \)

For node ‘in-’ connected to ‘n+’;
If \( v(n^+, n^-) < 0 \), then the controlling voltage would be \( v(in^+, in^-) \). Otherwise, the controlling voltage is \( v(in^+, n^+) \).

**PPWL Function**

For node ‘\( \text{in-} \)’ connected to ‘\( \text{n-} \)’;

If \( v(n^+, n^-) < 0 \), then the controlling voltage would be \( v(in^+, in1^-) \). Otherwise, the controlling voltage is \( v(in^+, n^+) \)

For node ‘\( \text{in-} \)’ connected to ‘\( \text{n+} \)’;

If \( v(n^+, n^-) > 0 \), then the controlling voltage would be \( v(in^+, in^-) \). Otherwise, the controlling voltage is \( v(in^+, n^+) \)

**G Element Parameters**

- **ABS**
  
  Output is absolute value if \( \text{ABS}=1 \).

- **CUR=equation**
  
  Current output which flows from \( n^+ \) to \( n^- \). The “equation”, which is defined by the user, can be a function of node voltages, branch currents, \( \text{TIME} \), temperature \( \text{TEMPER} \), and frequency \( \text{HERTZ} \).

- **DELAY**
  
  Keyword for the delay element. The delay element is the same as voltage controlled current source except it is associated by a propagation delay \( TD \). This element facilitates the adjustment of propagation delay in the subcircuit model process. \( \text{DELAY} \) is a Star-Hspice keyword and should not be used as a node name.

- **DELTA**
  
  Used to control the curvature of the piecewise linear corners. Defaults to \( 1/4 \) of the smallest of the distances between breakpoints. The maximum is \( 1/2 \) of the smallest of the distances between breakpoints.

- **Gxxx**
  
  Voltage controlled element name. Must begin with “\( G \)”, which may be followed by up to 15 alphanumeric characters.
**gatetype(k)**

May be AND, NAND, OR, or NOR. The value of k is the number of inputs of the gate. The x and y terms represent the piecewise linear variation of output as a function of input. In the multi-input gates only one input determines the state of the output.

**IC**

Initial condition. The initial estimate of the value(s) of the controlling voltage(s). If IC is not specified, Default=0.0.

**in +/-**

Positive or negative controlling nodes. Specify one pair for each dimension.

**M**

Number of replications of the element in parallel

**MAX**

Maximum current or resistance value. The default is undefined, and sets no maximum value.

**MIN**

Minimum current or resistance value. The default is undefined, and sets no minimum value.

**n+/−**

Positive or negative node of controlled element

**NPDELAY**

Sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep. That is,

\[
NPDELAY_{default} = \max\left(\frac{\min(TD, tstop)}{tstep}, 10\right)
\]

The values of tstep and tstop are specified in the .TRAN statement.

**NPWL**

Models the symmetrical bidirectional switch or transfer gate, NMOS

**p0, p1 …**

Polynomial coefficients. When one coefficient is specified, Star-Hspice assumes it to be p1, with p0=0.0, and the element is linear. When more than one polynomial coefficient is specified by p0, p1, p2, ..., the element is nonlinear. See “Polynomial Functions” on page 26-7.
**POLY**

Polynomial dimension. If POLY(ndim) is not specified, a one-dimensional polynomial is assumed. Ndim must be a positive number.

**PWL**

Piecewise linear function keyword

**PPWL**

Models the symmetrical bidirectional switch or transfer gate, PMOS

**SCALE**

Element value multiplier

**SMOOTH**

For piecewise linear dependent source elements, SMOOTH selects the curve smoothing method.

A curve smoothing method simulates exact data points you provide. This method can be used to make Star-Hspice simulate specific data points that correspond to measured data or data sheets.

Choices for SMOOTH are 1 or 2. Specifying 1 selects the smoothing method prior to Release H93A. Specifying 2 selects the smoothing method that uses data points you provide. This is the default method starting with release H93A.

**TC1,TC2**

First- and second-order temperature coefficients. The SCALE is updated by temperature:

\[ SCALE_{eff} = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2) \]

**TD**

Time delay keyword

**transconductance**

Voltage to current conversion factor

**transfactor**

Voltage to resistance conversion factor

**VCCAP**

Keyword for voltage controlled capacitance element. VCCAP is a Star-Hspice keyword and should not be used as a node name.

**VCCS**

Keyword for voltage controlled current source. VCCS is a Star-Hspice keyword and should not be used as a node name.
Performing Behavioral Modeling  Dependent Current Sources — G Elements

VCR  
Keyword for voltage controlled resistor element. VCR is a Star-Hspice keyword and should not be used as a node name.

x1,....  
Controlling voltage across nodes in+ and in-. The x values must be in increasing order.

y1,....  
Corresponding element values of x

Example

Switch

A voltage controlled resistor represents a basic switch characteristic. The resistance between nodes 2 and 0 varies linearly from 10meg to 1m ohms when voltage across nodes 1 and 0 varies between 0 and 1 volt. Beyond the voltage limits, the resistance remains at 10meg and 1m ohms respectively.

Gswitch 2 0 VCR PWL(1) 1 0 0v,10meg 1v,1m

Switch-level MOSFET

A switch level n-channel MOSFET can be modelled by the N-piecewise linear resistance switch. The resistance value does not change when the node d and s positions are switched.

Gnmos d s VCR NPWL(1) g s LEVEL=1 0.4v,150g 1v,10meg 2v,50k + 3v,4k 5v,2k

Voltage Controlled Capacitor

The capacitance value across nodes (out,0) varies linearly from 1p to 5p when voltage across nodes (ctrl,0) varies between 2v and 2.5v. Beyond the voltage limits, the capacitance value remains constant at 1 picofarad and 5 picofarads respectively.

Gcap out 0 VCCAP PWL(1) ctrl 0 2v,1p 2.5v,5p

Zero Delay Gate

A two-input AND gate can be implemented using an expression and a piecewise linear table. The inputs are voltages at nodes a and b, and the output is the current flow from node out to 0. The current is multiplied by the SCALE value, which, in this example, is specified as the inverse of the load resistance connected across the nodes (out,0).
Dependent Current Sources — G Elements

Performing Behavioral Modeling

Gand out 0 AND(2) a 0 b 0 SCALE=’1/rload’ 0v,0a 1v,.5a + 4v,4.5a 5v,5a

Delay Element

A delay is a low-pass filter type delay similar to that of an opamp. A transmission line, on the other hand, has an infinite frequency response. A glitch input to a G delay is attenuated similarly to a buffer circuit. In this example, the output of the delay element is the current flow from node out to node 1 with a value equal to the voltage across nodes (in, 0) multiplied by SCALE value and delayed by TD value.

Gdel out 0 DELAY in 0 TD=5ns SCALE=2 NPDELAY=25

Diode Equation

A forward bias diode characteristic from node 5 to ground can be modelled with a run time expression. The saturation current is 1e-14 amp, and the thermal voltage is 0.025v.

Gdio 5 0 CUR=’1e-14*(EXP(V(5)/0.025)-1.0)’

Diode Breakdown

Diode breakdown region to forward region can be modelled. When voltage across diode goes beyond the piecewise linear limit values (-2.2v, 2v), the diode current remains at the corresponding limit values (-1a, 1.2a).

Gdiode 1 0 PWL(1) 1 0 -2.2v,-1a -2v,-1pa .3v,.15pa.6v,10ua + 1v,1a 2v,1.2a

Triodes

Both the following voltage controlled current sources implement a basic triode. The first uses the poly(2) operator to multiply the anode and grid voltages together and scale by .02. The next example uses the explicit behavioral algebraic description.

Gt i_anode cathode poly(2) anode,cathode grid,cathode 0 0 0 .02 Gt i_anode cathode cur=’20m*v(anode,cathode)*v(grid,cathode)’
Performing Behavioral Modeling

Dependent Voltage Sources – H Elements

Current Controlled Voltage Source (CCVS)

The syntax is:

**Linear**

Hxxx n+ n- <CCVS> vn1 transresistance <MAX=val> <MIN=val>
+ <SCALE=val> <TC1=val><TC2=val> <ABS=1> <IC=val>

**Polynomial**

Hxxx n+ n- <CCVS> POLY(ndim) vn1 <... vnndim> <MAX=val>MIN=val>
+ <TC1=val> <TC2=val> <SCALE=val> <ABS=1> p0 <p1...> <IC=vals>

**Piecewise Linear**

Hxxx n+ n- <CCVS> PWL(1) vn1 <DELTA=val> <SCALE=val> <TC1=val>
+ <TC2=val> x1,y1 ... x100,y100 <IC=val>

**Multi-Input Gates**

Hxxx n+ n- gatetype(k) vn1, ... vnk <DELTA=val> <SCALE=val> <TC1=val>
+ <TC2=val> x1,y1 ... x100,y100 <IC=val>

**Delay Element**

Hxxx n+ n- <CCVS> DELAY vn1 TD=val <SCALE=val> <TC1=val> <TC2=val>
+ <NPDELAY=val>

---

**Note:** E Elements with algebraics make CCVS elements obsolete. However, CCVS elements may still be used for the sake of backward compatibility.

---

H Element Parameters

**ABS**

Output is absolute value if ABS=1.

**CCVS**

Keyword for current controlled voltage source. CCVS is a Star-Hspice keyword and should not be used as a node name.

**DELAY**

Keyword for the delay element. The delay element is the same as a current controlled voltage source except it is associated by a propagation delay TD. This element
facilitates the adjustment of propagation delay in the subcircuit model process. DELAY is a Star-Hspice keyword and should not be used as a node name.

**DELTA**

Used to control the curvature of the piecewise linear corners. Defaults to 1/4 of the smallest of the distances between breakpoints. The maximum is 1/2 of the smallest of the distances between breakpoints.

**gatetype(k)**

May be AND, NAND, OR, or NOR. The value of k is the number of inputs of the gate. The x and y terms represent the piecewise linear variation of output as a function of input. In the multi-input gates only one input determines the state of the output.

**Hxxx**

Current controlled voltage source element name. Must begin with “H”, which may be followed by up to 15 alphanumeric characters.

**IC**

Initial condition. This is the initial estimate of the value(s) of the controlling current(s) in amps. Default=0.0.

**MAX**

Maximum voltage value. The default is undefined, which sets no maximum value.

**MIN**

Minimum voltage value. The default is undefined, which sets no minimum value.

**n+/-**

Positive or negative controlled source connecting nodes.

**NPDELAY**

Sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep. That is,

\[ NPDELAY_{default} = \max \left( \frac{\min \{ TD, tstop \} \times 10}{tstep} \right) \]

The values of tstep and tstop are specified in the .TRAN statement.
The polynomial coefficients. When one coefficient is specified, Star-Hspice assumes it to be p1, with p0=0.0, and the element is linear. When more than one polynomial coefficient is specified by p0, p1, p2, ..., the element is nonlinear. See “Polynomial Functions” on page 26-7.

**POLY**
Polynomial dimension. If POLY(ndim) is not specified, a one-dimensional polynomial is assumed. Ndim must be a positive number.

**PWL**
Piecewise linear function keyword.

**SCALE**
Element value multiplier.

**TC1, TC2**
First and second order temperature coefficients. The SCALE is updated by temperature:

\[
\text{SCALE}_{\text{eff}} = \text{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)
\]

**TD**
Time delay keyword.

**transresistance**
Current to voltage conversion factor.

**vn1...**
Names of voltage sources through which the controlling current flows. One name must be specified for each dimension.

**x1,...**
Controlling current through vn1 source. The x values must be in increasing order.

**y1,...**
Corresponding output voltage values of x.

**Example**

\[
\text{HX 20 10 VCUR MAX=+10 MIN=-10 1000}
\]

The example above selects a linear current controlled voltage source. The controlling current flows through the dependent voltage source called VCUR. The defining equation of the CCVS is:

\[
HX = 1000 \cdot VCUR
\]
The defining equation states that the voltage output of HX is 1000 times the value of current flowing through CUR. If the equation produces a value of HX greater than +10V or less than -10V, HX, because of the MAX= and MIN= parameters, would be set to either 10V or -10V, respectively. CUR is the name of the independent voltage source that the controlling current flows through. If the controlling current does not flow through an independent voltage source, a dummy independent voltage source must be inserted.

```
.PARAM CT=1000
HX 20 10 VCUR MAX=+10 MIN=-10 CT
HXY 13 20 POLY(2) VIN1 VIN2 0 0 0 0 1 IC=0.5, 1.3
```

The example above describes a dependent voltage source with the value:

\[ V = I(VIN1) \cdot I(VIN2) \]

This two-dimensional polynomial equation specifies FA1=VIN1, FA2=VIN2, P0=0, P1=0, P2=0, P3=0, and P4=1. The controlling current for flowing through VIN1 is initialized at 0.5mA. For VIN2, the initial current is 1.3mA.

The direction of positive controlling current flow is from the positive node, through the source, to the negative node of vnam (linear). The polynomial (nonlinear) specifies the source voltage as a function of the controlling current(s).
Current Dependent Current Sources — F Elements

Current Controlled Current Source (CCCS)

The syntax is:

**Linear**

Fxxx n+ n- <CCCS> vn1 gain <MAX=val> <MIN=val> <SCALE=val>  
+ <TC1=val> <TC2=val> <M=val> <ABS=1> <IC=val>

**Polynomial**

Fxxx n+ n- <CCCS> POLY(ndim) vn1 <... vnndim> <MAX=val> <MIN=val>  
+ <TC1=val><TC2=val> <SCALE=vals> <M=val> <ABS=1> p0 <p1…>  
+ <IC=vals>

**Piecewise Linear**

Fxxx n+ n- <CCCS> PWL(1) vn1 <DELTA=val> <SCALE=val><TC1=val>  
+ <TC2=val><M=val> x1,y1 ... x100,y100 <IC=val>

**Multi-Input Gates**

Fxxx n+ n- <CCCS> gatetype(k) vn1, ... vnk <DELTA=val> <SCALE=val>  
+ <TC2=val><M=val> <ABS=1> x1,y1 ... x100,y100 <IC=val>

**Delay Element**

Fxxx n+ n- <CCCS> DELAY vn1 TD=val <SCALE=val> <TC1=val><TC2=val>  
+ NPDELAY=val

**Note:** G Elements with algebraics make CCCS elements obsolete. However, CCCS elements may still be used for backward compatibility with existing designs.
F Element Parameters

ABS
Output is absolute value if ABS=1.

CCCS
Keyword for current controlled current source. CCCS is a Star-Hspice keyword and should not be used as a node name.

DELAY
Keyword for the delay element. The delay element is the same as a current controlled current source except it is associated by a propagation delay TD. This element facilitates the adjustment of propagation delay in the subcircuit model process. DELAY is a Star-Hspice keyword and should not be used as a node name.

DELTA
Used to control the curvature of the piecewise linear corners. Defaults to 1/4 of the smallest of the distances between breakpoints. The maximum is 1/2 of the smallest of the distances between breakpoints.

Fxxx
Current controlled current source element name. Must begin with “F”, which may be followed by up to 15 alphanumeric characters.

gain
Current gain.

gatetype(k)
May be AND, NAND, OR, or NOR. The value of k is the number of inputs of the gate. The x and y terms represent the piecewise linear variation of output as a function of input. In the multi-input gates only one input determines the state of the output. The above keywords should not be used as node names.

IC
Initial condition: the initial estimate of the value(s) of the controlling current(s) in amps. Default=0.0.

M
Number of replications of the element in parallel.

MAX
Maximum output current value. The default is undefined, and sets no maximum value.
\textbf{MIN} \hspace{1cm} \text{Minimum output current value. The default is undefined, and sets no minimum value.}

\textit{n+/-} \hspace{1cm} \text{Positive or negative controlled source connecting nodes.}

\textbf{NPDELAY} \hspace{1cm} \text{Sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep. That is,}

\[ NPDELAY_{default} = \max \left[ \frac{\min(TD, tstop)}{tstep}, 10 \right] \]

The values of tstep and tstop are specified in the .TRAN statement.

\textit{p0, p1 ...} \hspace{1cm} \text{The polynomial coefficients. When one coefficient is specified, Star-Hspice assumes it to be p1, with p0=0.0, and the element is linear. When more than one polynomial coefficient is specified by p0, p1, p2, ..., the element is nonlinear. See “Polynomial Functions” on page 26-7.}

\textbf{POLY} \hspace{1cm} \text{Polynomial dimension. If POLY(ndim) is not specified, a one-dimensional polynomial is assumed. Ndim must be a positive number.}

\textbf{PWL} \hspace{1cm} \text{Piecewise linear function keyword.}

\textbf{SCALE} \hspace{1cm} \text{Element value multiplier.}

\textit{TC1, TC2} \hspace{1cm} \text{First and second order temperature coefficients. The SCALE is updated by temperature:}

\[ SCALE_{eff} = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2) \]

\textbf{TD} \hspace{1cm} \text{Time delay keyword.}

\textit{vn1...} \hspace{1cm} \text{Names of voltage sources through which the controlling current flows. One name must be specified for each dimension.}
x1,... Controlling current through vn1 source. The x values must be in increasing order.

y1,... Corresponding output current values of x.

Example

$ Current controlled current sources - F Elements,
F1 13 5 VSENS MAX=+3 MIN=-3 5
F2 12 10 POLY VCC 1MA 1.3M
Fd 1 0 DELAY vin TD=7ns SCALE=5
Filim 0 out PWL(1) vsrc -1a,-1a 1a,1a

The first example describes a current controlled current source connected between nodes 13 and 5. The current that controls the value of the controlled source flows through the voltage source named VSENS (to use a current controlled current source, a dummy independent voltage source is often placed into the path of the controlling current). The defining equation is:

\[ I(F1) = 5 \cdot I(VSENS) \]

The current gain is 5, the maximum current flow through F1 is 3 A, and the minimum current flow is -3 A. If \( I(VSENS) = 2 \) A, \( I(F1) \) would be set to 3 amps and not 10 amps as would be suggested by the equation. A user-defined parameter may be specified for the polynomial coefficient(s), as shown below.

.PARAM VU = 5
F1 13 5 VSENS MAX=+3 MIN=-3 VU

The second example describes a current controlled current source with the value:

\[ I(F2)=1e-3 + 1.3e-3 \cdot I(VCC) \]

Current flow is from the positive node through the source to the negative node. The direction of positive controlling current flow is from the positive node through the source to the negative node of vnam (linear), or to the negative node of each voltage source (nonlinear).

The third example is a delayed current controlled current source.

The fourth example is a piecewise linear current controlled current source.
Modeling with Digital Behavioral Components

This section provides examples of how to model with digital behavioral components.

Behavioral AND and NAND Gates

In this example, a two-input AND gate is modeled by a G Element. A two-input NAND gate is modeled by an E Element.

Example

```plaintext
behave.sp and/nand gates using g, e Elements
.options post=2
.op
.tran .5n 20n
.probe v(in1) v(in2) v(andout) v(in1) v(in2) v(nandout)
g 0 andout and(2) in1 0 in2 0
+ 0.0 0.0ma
+ 0.5 0.1ma
+ 1.0 0.5ma
+ 4.0 4.5ma
+ 4.5 4.8ma
+ 5.0 5.0ma
*
e nandout 0 nand(2) in1 0 in2 0
+ 0.0 5.0v
+ 0.5 4.8v
+ 1.0 4.5v
+ 4.0 0.5v
+ 4.5 0.2v
+ 5.0 0.0v
*
vin1 in1 0 0 pwl(0,0 5ns,5)
vin2 in2 0 5 pwl(0,5 10ns,5 15ns,0)
rin1 in1 0 1k
rin2 in2 0 1k
rand andout 0 1k
rnand nandout 0 1k
.end
```
Behavioral D-Latch

In this example, a D flip-flop is modeled by one input NAND gates and NPWL/PPWL functions.
Example

dlatch.sp--- cmos d-latch
.option post
.tran .2n 60ns
.probe tran clock=v(clck) data=v(d) q=v(q)
.ic v(q)=0

Waveforms

vdata d 0 pulse(0,5 2n,1n,1n 19n,40n)
vclk clck 0 pulse(0,5 7n,1n,1n 10n,20n)
vclkn clckn 0 pulse(5,0 7n,1n,1n 10n,20n)
xdlatch d clck clckn q qb dlatch cinv=.2p
**Subcircuit Definitions for Behavioral N-Channel MOSFET**

* DRAIN GATE SOURCE
.SUBCKT nmos 1 2 3 capm=.5p
cd 1 0 capm
cs 3 0 capm
gn 3 1 VCR NPWL(1) 2 3
  + 0. 495.8840G
  + 200.00000M 456.0938G
  + 400.00000M 141.6902G
  + 600.00000M 7.0624G
  + 800.00000M 258.9313X
  + 1.00000 6.4866X
  + 1.20000 842.9467K
  + 1.40000 321.6882K
  + 1.60000 170.8367K
  + 1.80000 106.4944K
  + 2.00000 72.7598K
  + 2.20000 52.4632K
  + 2.40000 38.5634K
  + 2.60000 8.8056K
  + 2.80000 5.2543K
  + 3.00000 4.3553K
  + 3.20000 3.8407K
  + 3.40000 3.4950K
  + 3.60000 3.2441K
  + 3.80000 3.0534K
  + 4.00000 2.9042K
  + 4.20000 2.7852K
  + 4.40000 2.6822K
  + 4.60000 2.5K
  + 5.0 2.3K
.ENDS nmos

**Behavioral P-Channel MOSFET**

* DRAIN GATE SOURCE
.SUBCKT pmos 1 2 3 capm=.5p
cd 1 0 capm
cs 3 0 capm
gp 1 3 VCR PPWL(1) 2 3
  + -5.0000 2.3845K
  + -4.8000 2.4733K
Performing Behavioral Modeling

Modeling with Digital Behavioral Components

+ -4.6000 2.5719K
+ -4.4000 2.6813K
+ -4.2000 2.9415K
+ -3.8000 3.1116K
+ -3.6000 3.3221K
+ -3.4000 3.5895K
+ -3.2000 3.9410K
+ -3.0000 4.4288K
+ -2.8000 5.1745K
+ -2.6000 6.6041K
+ -2.4000 29.6203K
+ -2.0000 42.4517K
+ -2.0000 58.3239K
+ -1.8000 83.4296K
+ -1.6000 128.1517K
+ -1.4000 221.2640K
+ -1.2000 471.8433K
+ -1.0000 1.6359X
+ -800.00M 41.7023X
+ -600.00M 1.3394G
+ -400.00M 38.3449G
+ -200.00M 267.7325G
+ 0. 328.7122G
.ENDS pmos
*
.subckt tgate in out clk clkn ctg=.5p
xm in clk out nmos capm=ctg
xmp in clkn out pmos capm=ctg
.ends tgate

.SUBCKT inv in out capout=1p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(1) in 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
Behavioral Double-Edge Triggered Flip-Flop

In this example a double edged triggered flip-flop is modeled by using the D_LATCH subcircuit from previous example and several NAND gates.
Figure 26-5: Double-Edge Triggered Flip-Flop Schematic

Example

det_dff.sp--- double edge triggered flip-flop
.option post=2
.tran .2n 100ns
.probe tran clock=v(clk) data=v(d) q=v(q)

Waveforms

vdata d 0 pulse(0,5 2n,1n,1n 28n,50n)
vc1k clck 0 pulse(0,5 7n,1n,1n 10n,20n)

Main Circuit

xclkn clck clckn inv cinv=.1p
xd1 d clck clckn q1 qb1 dlatch cinv=.2p
xd2 d clckn clck q2 qb2 dlatch cinv=.2p
xnand1 clck qb2 n1 nand2 capout=.5p
xnand2 q1 n1 n2 nand2 capout=.5p
xnand3 q2 clck n3 nand2 capout=.5p
xnand4 n2 n3 q nand2 capout=.5p
xinv q qb inv capout=.5p
Subcircuit Definitions

*Note: Subcircuit definitions for NMOS, PMOS, and INV are given in the
* D-Latch examples; therefore they are not repeated here.

.SUBCKT nand2 in1 in2 out capout=2p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(2) in1 0 in2 0 scale=1
  0 4.90ma  
+ 0.25 4.88ma  
+ 0.5 4.85ma  
+ 1.0 4.75ma  
+ 1.5 4.42ma  
+ 3.5 1.00ma  
+ 4.000 0.50ma  
+ 4.5 0.2ma  
+ 5.0 0.1ma  
.ENDS nand2

.subckt dlatch data clck clckn q qb cinv=1p
xtg1 data a clck clckn tgate ctg='cinv/2'
xtg2 q ax clck clckn tgate ctg='cinv/2'
rx ax a 10
xinv1 a qb inv capout=cinv
xinv2 qb q inv capout=cinv
.ENDS dlatch
.END
Figure 26-6: Double Edge Triggered Flip-Flop Response
Calibrating Digital Behavioral Components

This section describes how to calibrate with digital behavioral components.

Building Behavioral Lookup Tables

The following simulation demonstrates an ACL family output buffer with 2 ns delay and 1.8 ns rise and fall time. The ground and VDD supply currents and the internal ground bounce due to the package are also shown.

Figure 26-7: ACL Family Output Buffer
Star-Hspice can automatically measure the datasheet quantities such as TPHL, risetime, maximum power dissipation, and ground bounce using the following commands.

```
.MEAS tphl trig v(D) val='5*vdd' rise=1
+ targ v(out) val='5*vdd' fall=1
.MEAS risetime trig v(out) val='1*vdd' rise=1
+ targ v(out) val='9*vdd' rise=1
.MEAS max_power max power
.MEAS bounce max v(xin.v_local)
```

The inverter is composed of capacitors, diodes, one-dimensional lookup table MOSFETs, and a special low-pass delay element. The low-pass delay element has the property that attenuates pulses that are narrower than the delay value.

**Figure 26-8: Inverter**

![Inverter Diagram]
Subcircuit Definition

```plaintext
.subckt inv in out v+ v-
cout+ out_l v+ 2p
cout- out_l v- 2p
xmp out_l inx v+ pmos
xmn out_l inx v- nmos
e inx v- delay in v- td=1n
din v- in dx
.model dx d cjo=2pf
chi in v+ .5pf
.ends inv
```

The behavioral MOSFETs are represented by one dimensional lookup tables. The equivalent n-channel lookup table is shown below.

**Behavioral N-Channel MOSFET**

**Drain Gate Source**

```plaintext
.subckt nmos 1 2 3
gn 3 1 VCR npwl(1) 2 3 scale=0.008
* VOLTAGE RESISTANCE
  + 0. 495.8840g
  + 200.00000m 456.0938g
  + 400.00000m 141.6902g
  + 600.00000m 7.0624g
  + 800.00000m 258.9313meg
  + 1.00000 6.4866meg
  + 1.20000 842.9467k
  + 1.40000 21.6882k
  + 1.60000 170.8367k
  + 1.80000 106.4944k
  + 2.00000 72.7598k
  + 2.20000 52.4632k
  + 2.40000 38.5634k
  + 2.60000 8.8056k
  + 2.80000 5.2543k
  + 3.00000 4.3553k
  + 3.40000 3.4950k
  + 3.80000 2.0534k
  + 4.20000 2.7852k
  + 4.60000 2.5k
```
The table above describes a voltage versus resistance table. It shows, for example, that the resistance at 5 volts is 2.3 kohm.

**Creating a Behavioral Inverter Lookup Table**

You can create an inverter lookup table in two simple steps. First simulate an actual transistor level inverter using a DC sweep of the input and print the output V/I for the output pullup and pulldown transistors. Next, copy the printed output into the volt controlled resistor lookup table element.

The following test file, `inv_vin_vout.sp` calculates RN (the effective pulldown resistor transfer function) and RP (the pullup transfer function).

RN is calculated as \( \frac{V_{out}}{I(mn)} \) where \( mn \) is the pulldown transistor. RP is calculated as \( \frac{(VCC-V_{out})}{I(mp)} \) where \( mp \) is the pullup transfer function.

The actual calculation uses a more accurate way of obtaining the transistor series resistance as follows:

**Figure 26-9: VIN versus VOUT**

```
Vdx     Rtot=(Vds-Vsx)/Ids
RD      For greater accuracy:
Vd      Rtot=RD +RS +(vd-vs)/Ids
Vd      RD =1/LV16(mn)
Vd      RS =1/LV17(mn)
Vd      (vd-vs) =LX3(mn)
Vd      Ids =LX4(mn)
```

The first graph below shows VIN versus VOUT and the second graph shows the computed transfer resistances RP and RN as a function of VIN.
The Star-Hspice file used to calculate RP and RN is

```
$ inv_vin_vout.sp sweep inverter vin versus vout, calculate rn and rp

* use dc sweep with 3 ranges; 0-1.5v, 1.6-2.5, 2.6 5
   .dc vin lin 8 0 2.0 lin 20 2.1 2.5 lin 6 2.75 5
   $$ rn=par('v(out)/i(x1.mn)')
   .print rn=
   + par('1/lv16(x1.mn)+1/lv17(x1.mn)+abs(lx3(x1.mn)/
     lx4(x1.mn)')
   .print rp=par('(-vcc+v(out))/i(x1.mp)')
   .param sigma=0 vcc=5
   .global vcc
   vcc vcc 0 vcc
   vin in 0 pwl 0,0 0.2n,5
```
Performing Behavioral Modeling

Calibrating Digital Behavioral Components

.xi in out inv
.macro inv in out
nn out in 0 0 nch w=10u l=1u
mp out in vcc vcc pch w=10u l=1u
.eom

The tabular listing produced by Star-Hspice is

****** dc transfer curves tnom= 25.000 temp= 25.000
******

<table>
<thead>
<tr>
<th>volt</th>
<th>rn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>3.312e+14</td>
</tr>
<tr>
<td>285.71429m</td>
<td>317.3503g</td>
</tr>
<tr>
<td>571.42857m</td>
<td>304.0682x</td>
</tr>
<tr>
<td>857.14286m</td>
<td>1.1222x</td>
</tr>
<tr>
<td>1.14286</td>
<td>107.6844k</td>
</tr>
<tr>
<td>1.42857</td>
<td>32.1373k</td>
</tr>
<tr>
<td>1.71429</td>
<td>14.6984k</td>
</tr>
<tr>
<td>2.00000</td>
<td>7.7108k</td>
</tr>
<tr>
<td>2.10000</td>
<td>5.8210k</td>
</tr>
<tr>
<td>2.12105</td>
<td>5.1059k</td>
</tr>
<tr>
<td>2.14211</td>
<td>3.2036k</td>
</tr>
<tr>
<td>2.16316</td>
<td>1.6906k</td>
</tr>
<tr>
<td>2.18421</td>
<td>1.4421k</td>
</tr>
<tr>
<td>2.20526</td>
<td>1.3255k</td>
</tr>
<tr>
<td>2.22632</td>
<td>1.2470k</td>
</tr>
<tr>
<td>2.24737</td>
<td>1.1860k</td>
</tr>
<tr>
<td>2.26842</td>
<td>1.1360k</td>
</tr>
<tr>
<td>2.28947</td>
<td>1.0935k</td>
</tr>
<tr>
<td>2.31053</td>
<td>1.0565k</td>
</tr>
<tr>
<td>2.33158</td>
<td>1.0238k</td>
</tr>
<tr>
<td>2.35263</td>
<td>994.3804</td>
</tr>
<tr>
<td>2.37368</td>
<td>967.7559</td>
</tr>
<tr>
<td>2.39474</td>
<td>943.4266</td>
</tr>
<tr>
<td>2.41579</td>
<td>921.0413</td>
</tr>
<tr>
<td>2.43684</td>
<td>900.3251</td>
</tr>
<tr>
<td>2.45789</td>
<td>881.0585</td>
</tr>
<tr>
<td>2.47895</td>
<td>863.0632</td>
</tr>
<tr>
<td>2.50000</td>
<td>846.1922</td>
</tr>
<tr>
<td>2.75000</td>
<td>701.5119</td>
</tr>
<tr>
<td>3.20000</td>
<td>560.6908</td>
</tr>
<tr>
<td>3.65000</td>
<td>479.8893</td>
</tr>
</tbody>
</table>
Optimizing Behavioral CMOS Inverter Performance

Calibrate behavioral models by running Star-Hspice on the full transistor version of a cell and then optimizing the behavioral model to this data.

**Figure 26-11: CMOS Inverter and its Equivalent Circuit**

In this example, Star-Hspice simulates the CMOS inverter using the LEVEL 3 MOSFET model. The input and output resistances are obtained by performing a .TF transfer function analysis (.TF V(out) Vin). The transfer function table of the inverter is obtained by performing the .DC analysis sweeping input voltage (.DC Vin 0 5 .1). This table is then used in the PWL element to represent the transfer function of the inverter. The rise and fall time of the inverter in the equivalent circuit is adjusted by a voltage controlled PWL capacitance across the output resistance. The propagation delay is obtained by the delay element across the output rc circuit. The input capacitance is adjusted by using the inverter in a ring.
oscillator. All the adjustment in this example is done using the Star-Hspice optimization analysis. The data file and the results are shown below.

Example

INVB_OP.SP---OPTIMIZATION OF CMOS MACROMODEL INVERTER
.OPTIONS POST PROBE NOMOD METHOD=GEAR
.GLOBAL VCC VCCM
.PARAM VCC=5 ROUT=2.5K CAPIN=.5P + TDELAY=OPTINV(1.0N,.5N,3N)
+ CAPL=OPTINV(.2P,.1P,.6P)
+ CAPH=OPTINV(.2P,.1P,.6P)
.TRAN .25N 120NS
+ SWEEP OPTIMIZE=OPTINV RESULTS=RISEX,FALLX,PROPFX,PROPRX
+ MODEL=OPT1
.MODEL OPT1 OPT ITROPT=30 RELIN=1.0E-5 RELOUT=1E-4
.MEAS TRAN PROPFM TRIG V(INM) VAL='.5*VCC' RISE=2
+ TARG V(OUTM) VAL='.5*VCC' FALL=2
.MEAS TRAN PROPFX TRIG V(IN) VAL='.5*VCC' RISE=2
+ TARG V(OUT) VAL='.5*VCC' FALL=2
+ GOAL='PROPFM' WEIGHT=0.8
.MEAS TRAN PROPRM TRIG V(INM) VAL='.5*VCC' FALL=2
+ TARG V(OUTM) VAL='.5*VCC' RISE=2
+ MEAS TRAN PROPRX TRIG V(IN) VAL='.5*VCC' FALL=2
+ TARG V(OUT) VAL='.5*VCC' RISE=2
+ GOAL='PROPRM' WEIGHT=0.8
.MEAS TRAN FALLM TRIG V(OUTM) VAL='.9*VCC' FALL=2
+ TARG V(OUTM) VAL='.1*VCC' FALL=2
.MEAS TRAN FALLX TRIG V(OUT) VAL='.9*VCC' FALL=2
+ TARG V(OUT) VAL='.1*VCC' FALL=2
+ GOAL='FALLM'
.MEAS TRAN RISEM TRIG V(OUTM) VAL='.1*VCC' RISE=2
+ TARG V(OUTM) VAL='.9*VCC' RISE=2
.MEAS TRAN RISEx TRIG V(OUT) VAL='.1*VCC' RISE=2
+ TARG V(OUT) VAL='.9*VCC' RISE=2
+ GOAL='RISEM'
.TRAN 0.5N 120NS
.PROBE V(out) V(outm)
VC VCC 0 VCC
VCCM VCCM 0 VCC
XI IN OUT INV
XIM INM OUTM INVM
VIN IN GND PULSE(0,5 1N,5N,5N 20N,50N)
VINM INM GND PULSE(0,5 1N,5N,5N 20N,50N)
Subcircuit Definition

```
.SUBCKT INV IN OUT
RIN IN 0 1E12
CIN IN 0 CAPIN
ET 1 0 PWL(1) IN 0
  + 1.00000 5.0
  + 1.50000 4.93
  + 2.00000 4.72
  + 2.40000 4.21
  + 2.50000 3.77
  + 2.60000 0.90
  + 2.70000 0.65
  + 3.00000 0.30
  + 3.50000 0.092
  + 4.00000 0.006
  + 4.60000 0.
RT 1 0 1K
GD 0 OUT DELAY 1 0 TD=TDELAY SCALE=1/ROUT
GCOUT OUT 0 VCCAP PWL(1) IN 0 1V,CAPL 2V,CAPH
ROUT OUT 0 ROUT
.ENDS
```

Inverter Using Model

```
.SUBCKT INVM IN OUT
XP1 OUT IN VCCM VCCM MP
XN1 OUT IN GND GND MN
.ENDS
.MODEL N NMOS LEVEL=3 TOX=850E-10 LD=.85U NSUB=2E16 VTO=1
  + GAMMA=1.4 PHI=.9 UO=823 VMAX=2.7E5 XJ=0.9U KAPPA=1.6
  + ETA=.1 THETA=.18 NFS=1.6E11 RSH=25 CJ=1.85E-4 MJ=.42 PB=.7
  + CJSW=6.2E-10 MJSW=.34 CGSO=5.3E-10 CGDO=5.3E-10
  + CGBO=1.75E-9
.MODEL P PMOS LEVEL=3 TOX=850E-10 LD=.6U
  + NSUB=1.4E16 VTO=-.86 GAMMA=.65 PHI=.76 UO=266
  + VMAX=.8E5 XJ=0.7U KAPPA=4 ETA=.25 THETA=.08 NFS=2.3E11
  + RSH=85 CJ=1.78E-4 MJ=.4 PB=.6 CJSW=5E-10 MJSW=.22
  + CGSO=5.3E-10 CGDO=5.3E-10 CGBO=.98E-9
SUBCKT MP 1 2 3 4
M1 1 2 3 4 P W=45U L=5U AD=615P AS=615P
  + PD=65U PS=65U NRD=.4 NRS=.4
```
Performing Behavioral Modeling
Calibrating Digital Behavioral Components

```
.ENDS MP
.SUBCKT MN 1 2 3 4
M1 1 2 3 4 N W=17U L=5U AD=440P AS=440P
+ PD=80U PS=80U NRD=.85 NRS=.85
.ENDS MN
.END
```

**Result**

OPTIMIZATION RESULTS

- RESIDUAL SUM OF SQUARES = 4.589123E-03
- NORM OF THE GRADIENT = 1.155285E-04
- MARQUARDT SCALING PARAMETER = 130.602
- NO. OF FUNCTION EVALUATIONS = 51
- NO. OF ITERATIONS = 15

OPTIMIZATION COMPLETED

MEASURED RESULTS < RELOUT= 1.0000E-04 ON LAST ITERATIONS

**Optimized Parameters OPTINV**

<table>
<thead>
<tr>
<th></th>
<th>%NORM-SEN</th>
<th>%CHANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>.PARAM TDELAY = 1.3251N</td>
<td>37.6164</td>
<td>-48.6429U</td>
</tr>
<tr>
<td>.PARAM CAPL  = 390.2613F</td>
<td>37.2396</td>
<td>60.2596U</td>
</tr>
<tr>
<td>.PARAM CAPH  = 364.2716F</td>
<td>25.1440</td>
<td>62.1922U</td>
</tr>
</tbody>
</table>

**Optimize Results Measure Names and Values**

- RISEX = 2.7018N
- FALLX = 2.5388N
- PROPFX = 2.0738N
- PROPRX = 2.1107N
Optimizing Behavioral Ring Oscillator Performance

To optimize behavioral ring oscillator performance, review the examples in this section.

Example Five-Stage Ring Oscillator
RING5BM.SP-5 STAGE RING OSCILLATOR--MACROMODEL CMOS INVERTER
.IC V(IN)=5 V(OUT1)=0 V(OUT2)=5 V(OUT3)=0
.IC V(INM)=5 V(OUT1M)=0 V(OUT2M)=5 V(OUT3M)=0
.GLOBAL VCCM
.OPTIONS NOMOD POST=2 PROBE METHOD=GEAR DELMAX=0.5N
.PARAM VCC=5 $ CAPIN=0.92137P
.PARAM TDELAY=1.32N CAPL=390.26F CAPH=364.27F ROUT=2.5K
+ CAPIN=OPTOSC(0.8P,0.1P,1.0P)
.TRAN 1NS 150NS UIC
+ SWEEP OPTIMIZE=OPTOSC RESULTS=PERIODX MODEL=OPT1
.MODEL OPT1 OPT RELIN=1E-5 RELOUT=1E-4 DFSIZ=.02 ITROPT=25
.MEAS TRAN PERIODM TRIG V(OUT3M) VAL=’.8*VCC’ RISE=2
+ TARG V(OUT3M) VAL=’.8*VCC’ RISE=3
.MEAS TRAN PERIODX TRIG V(OUT3) VAL=’.8*VCC’ RISE=2
+ TARG V(OUT3) VAL=’.8*VCC’ RISE=3
Performing Behavioral Modeling

Calibrating Digital Behavioral Components

+ GOAL='PERIODM'

.TRAN 1NS 150NS UIC
.PROBE V(OUT3) V(OUT3M)
X1 IN OUT1 INV
X2 OUT1 OUT2 INV
X3 OUT2 OUT3 INV
X4 OUT3 OUT4 INV
X5 OUT4 IN INV
CL IN 0 1P
VCCM VCCM 0 VCC
X1M INM OUT1M INVM
X2M OUT1M OUT2M INVM
X3M OUT2M OUT3M INVM
X4M OUT3M OUT4M INVM
X5M OUT4M INM INVM
CLM INM 0 1P
*Subcircuit definitions given in the previous example are not repeated here.

.END

Result

Optimization Results

RESIDUAL SUM OF SQUARES = 4.704516E-10
NORM OF THE GRADIENT = 2.887249E-04
MARQUARDT SCALING PARAMETER = 32.0000
NO. OF FUNCTION EVALUATIONS = 52
NO. OF ITERATIONS = 20

Optimization completed

MEASURED RESULTS < RELOUT= 1.0000E-04 ON LAST ITERATIONS

***** OPTIMIZED PARAMETERS OPTOSC
*
  %NORM-SEN  %CHANGE
.PARAM CAPIN = 921.4155F  $  100.0000  8.5740U

*** OPTIMIZE RESULTS MEASURE NAMES AND VALUES
  * PERIODX = 40.3180N

Figure 26-13: Ring Oscillator Response
Using Analog Behavioral Elements

The following components are examples of analog behavioral building blocks. Each demonstrates a basic Star-Hspice feature:

- **integrator**: ideal op-amp E Element source
- **differentiator**: ideal op-amp E Element source
- **ideal transformer**: ideal transformer E Element source
- **tunnel diode**: lookup table G Element source
- **silicon-controlled rectifier**: lookup table H Element source
- **triode vacuum tube**: algebraic G Element source
- **AM modulator**: algebraic G Element source
- **data sampler**: algebraic E Element source

**Behavioral Integrator**

The integrator circuit is modelled by an ideal op-amp and uses a VCVS to adjust the output voltage. The output of integrator is given by:

\[
V_{out} = -\frac{gain}{R_i \cdot C_i} \cdot \int_0^t V_{in} \cdot dt + V_{out}(0)
\]

**Figure 26-14: Integrator Circuit**
Example

Integ.sp integrator circuit

Control and Options

.TRAN 1n 20n
.OPTIONS POST PROBE DELMAX = .1n
.PROBE Vin=V(in) Vout=V(out)

Subcircuit Definition

.SUBCKT integ in out gain=-1 rval=1k cval=1p
  EOP out1 0 OPAMP in- 0
  Ri in in- rval
  Ci in- out1 cval
  Egain out 0 out1 0 gain
  Rout out 0 1e12
.ENDS

Circuit

Xint in out integ gain=-0.4
Vin in 0 PWL(0,0 5n,5v 15n,-5v 20n,0)
.END
Behavioral Differentiator

A differentiator is modelled by an ideal op-amp and a VCVS for adjusting the magnitude and polarity of the output. The differentiator response is given by:

\[ V_{out} = -\text{gain} \cdot R_d \cdot C_d \cdot \frac{d}{dt} V_{in} \]

For a high-frequency signal, the output of a differentiator can have overshoot at the edges. You can smooth this out using a simple RC filter.
**Figure 26-16: Differentiator Circuit**

Example

```
Diff.sp  differentiator circuit
* V(out)=Rval * Cval * gain * (dV(in)/dt)
```

Control and Options

```
.TRAN 1n 20n
.PROBE Vin=V(in)  Vout=V(out)
.OPTIONS PROBE POST
```

Differentiator Subcircuit Definition

```
.SUBCKT diff in out  gain=-1 rval=1k cval=1pf
EOP  out1  0  OPAMP  in-  0
Cd in in-  cval
Rd in- out1 rval
Egain out2 0 out1 0 gain
Rout out2 0 le12
*rc filter to smooth the output
R out2 out 75
C out 0 1pf
.ENDS
```

Circuit

```
Xdiff in out  diff rval=5k
Vin in 0 PWL(0,0 5n,5v 15n,-5v 20n,0)
.END
```
Figure 26-17: Response Of a Differentiator to a Triangle Waveform

Ideal Transformer

The following example uses the ideal transformer to convert 8 ohms impedance of a loudspeaker to 800 ohms impedance, which is a proper load value for a power amplifier, Rin=n² ⋅ RL.

MATCHING IMPEDANCE BY USING IDEAL TRANSFORMER
E OUT 0 TRANSFORMER IN 0 10
RL OUT 0 8
VIN IN 0 1
.OP
.END
Behavioral Tunnel Diode

In the following example, a tunnel diode is modeled by a PWL VCCS. The current characteristics are obtained for two DELTA values (50 µv and 10 µv). The IV characteristics corresponding to DELTA=10 µv have sharper corners. The derivative of current with respect to voltage (GD) is also displayed. The GD value around breakpoints changes in a linear fashion.

Example

tunnel.sp -- modeling tunnel diode characteristic by pwl vccs  
* pwl function is tested for two different delta values. The  
* smaller delta will create the sharper corners.  
.options post=2  
vin 1 0 pvd  
.dc pvd 0 550m 5m sweep delta poi 2 50mv 5mv  
.probe dc id=1x0(g) gd=1x2(g)  
g 1 0 pwl(1) 1 0 delta=delta  
  + -50mv,-5ma 50mv,5ma 200mv,1ma 400mv,.05ma  
  + 500mv,2ma 600mv,15ma  
.end
Behavioral Silicon Controlled Rectifier (SCR)

The silicon controlled rectifier (SCR) characteristic can be easily modeled using a PWL CCVS because there is a unique voltage value for any current value.

Example

pwl6.sp--- modeling SCR by pwl ccvs
*The silicon controlled rectifier (SCR) characteristic
*is modeled by a piecewise linear current controlled
*voltage source (PWL_CCVS), because for any current value
*there is a unique voltage value.
*
*use iscr as y-axis and v(1) as x-axis
*
.options post=2
  iscr 0 2 0
  vdm 2 1 0
Using Analog Behavioral Elements

Performing Behavioral Modeling

Behavioral Triode Vacuum Tube Subcircuit

The following example shows how to include the behavioral elements in a subcircuit to give very good triode tube action. The basic power law equation (current source $gt$) is modified by the voltage source $ea$ to give better response in saturation.

Example

triode.sp triode model family of curves using behavioral elements

Figure 26-20: Silicon Controlled Rectifier
Control and Options
.options post acct
.dc va 20v 60v 1v vg 1v 10v 1v
.probe ianode=i(xt.ra) v(anode) v(grid) eqn=lv6(xt.gt)
.print v(xt.int_anode) v(xt.i_anode) inode=i(xt.ra)
   eqn=lv6(xt.gt)

Circuit
vg grid 0 1v
va anode 0 20v
vc cathode 0 0v
xt anode grid cathode triode

Subcircuit Definition
.subckt triode anode grid cathode
   * 5 ohm anode resistance
   * ea creates saturation region conductance
   ra anode i_anode 5
   ea int_anode cathode pwl(1) i_anode cathode delta=.01
      + 20,0 27,.85 28,.95 29,.99 30,1 130,1.2
   gt i_anode cathode
   +
   cur=’20m*v(int_anode,cathode)*pwr(max(v(grid,cathode),0),1.5)’
   cga grid i_anode 30p
   cgc grid cathode 20p
   cac i_anode cathode 5p
.ends
*.

.end
Figure 26-21: Triode Family of Curves

Behavioral Amplitude Modulator

This example uses a G Element as an amplitude modulator with pulse waveform carrier.

Example

```
amp_mod.sp amplitude modulator with pulse waveform carrier
.OPTIONS POST
.TRAN .05m 40m
.PROBE V(1) V(2) V(3)
Vs 1 0 SIN(0,1,100)
Vc 2 0 PULSE(1,-1,0,1n,1n,.5m,1m)
Rs 1 0 1+
Rc 2 0 1
G 0 3 CUR='(1+.5*V(1))*V(2)'
Re 3 0 1
.END
```
Figure 26-22: Amplitude Modulator Waveforms

Behavioral Data Sampler

A behavioral data sample follows.

Example

sampling.sp sampling a sine wave.
.OPTIONS POST
.TRAN .05m 40m
.PROBE V(1) V(2) V(3)
Vc 1 0 SIN(0,5,100)
Vs 2 0 PULSE(0,1,0,1n,1n,.5m,1m)
Rc 1 0 1
Rs 2 0 1
E 3 0 VOL=’V(1)*V(2)’
Re 3 0 1
.END
Figure 26-23: Sampled Data
Using Op-Amps, Comparators, and Oscillators

This section describes the benefits of using Star-Hspice’s op-amps, comparators, and oscillators when performing simulation.

Star-Hspice Op-Amp Model Generator

Star-Hspice uses the model generator for the automatic design and simulation of both board level and IC op-amp designs. You can take the existing electrical specifications for a standard industrial operational amplifier, enter the specifications in the op-amp model statement, and Star-Hspice automatically generates the internal components of the op-amp to meet the specifications. You can then call the design from a library for a board level simulation.

The Star-Hspice op-amp model is a subcircuit that is about 20 times faster to simulate than an actual transistor level op-amp. You can adjust the AC gain and phase to within 20 percent of the actual measured values and set the transient slew rates accurately. This model does not contain high order frequency response poles and zeros and may significantly differ from actual amplifiers in predicting high frequency instabilities. Normal amplifier characteristics, including input offsets, small signal gain, and transient effects are represented in this model.

The op-amp subcircuit generator consists of two parts, a model and one or more elements. Each element is in the form of a subcircuit call. The model generates an output file of the op-amp equivalent circuit for collection in libraries. The file name is the name of the model (mname) with an .inc extension.

Once the output file is generated, other Star-Hspice input files may reference this subcircuit using a .SUBCKT call to the model name. The .SUBCKT call automatically searches the present directory for the file, then the directories specified in any .OPTION SEARCH = ‘directory_path_name’, and finally the directory where the DDL (Discrete Device Library) is located.

The amplifier element references the amplifier model.
Convergence

If DC convergence problems are encountered with op-amp models created by the model generator, use the .IC or .NODESET statement to set the input nodes to the voltage halfway between the VCC and VEE. This balances the input nodes and stabilizes the model.

Op-Amp Element Statement Format

COMP=0 (internal compensation)

The syntax is:
\[ xa1 \text{ in- in+ out vcc vee modelname AV=val} \]

COMP=1 (external compensation)

The syntax is:
\[ xa1 \text{ in- in+ out comp1 comp2 vcc vee modelname AV=val} \]

\[ in- \] the inverting input
\[ in+ \] the noninverting input
\[ out \] the output, single ended
\[ vcc \] the positive supply
\[ vee \] the negative supply
\[ modelname \] the subcircuit reference name

Op-Amp .MODEL Statement Format

The syntax is:
\[ .MODEL \text{ mname AMP parameter=value ...} \]

\[ mname \] model name. Elements reference the model by this name.
\[ AMP \] identifies an amplifier model
\[ parameter \] any model parameter described below
value value assigned to a parameter

Example

X0 IN– IN+ OUT0 VCC VEE ALM124
.MODEL ALM124 AMP
+ C2= 30.00P SRPOS= .5MEG SRNEG= .5MEG
+ IB= 45N IBOS= 3N VOS= 4M
+ FREQ= 1MEG DELPHS= 25 CMRR= 85
+ ROUT= 50 AV= 100K ISC= 40M
+ VOPOS= 14.5 VONEG= -14.5 PWR= 142M
+ VCC= 16 VEE= -16 TEMP= 25.00
+ PSRR= 100 DIS= 8.00E-16 JIS= 8.00E-16

Op-Amp Model Parameters

The model parameters for op-amps are shown below. The defaults for these parameters depend on the DEF parameter setting. Defaults for each of the three DEF settings are shown in the following table.

<table>
<thead>
<tr>
<th>Names (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV (AVD)</td>
<td>volt/volt</td>
<td></td>
<td>Amplifier gain in volts out per volt in. It is the DC ratio of the voltage in to the voltage out. Typical gains are from 25k to 250k. If the frequency comes out too low, try increasing the negative and positive slew rates or decreasing DELPHS.</td>
</tr>
<tr>
<td>AV1K</td>
<td>volt/volt</td>
<td></td>
<td>Amplifier gain at 1 kilohertz. This is a convenient method of estimating the unity gain bandwidth. The gain can be expressed in actual voltage gain or in dB. Decibel is now a standard unit conversion for Star-Hspice. If AV1K is set, then FREQ is ignored. A typical value for AV1K is AV1K=(unity gain freq)/1000.</td>
</tr>
</tbody>
</table>
### Using Op-Amps, Comparators, and Oscillators

#### Performing Behavioral Modeling

<table>
<thead>
<tr>
<th>Names (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>farad</td>
<td></td>
<td>Internal feedback compensation capacitance. If the amplifier is internally compensated and no capacitance value is given, assume 30 pF. If the gain is high (above 500k), the internal compensation capacitor is probably different (typically 10 pF). If the amplifier is externally compensated, (COMP=1) set C2 to about 0.5 pF as the residual internal capacitance.</td>
</tr>
<tr>
<td>CMRR</td>
<td>volt/ volt</td>
<td></td>
<td>Common mode rejection ratio. This is usually between 80 and 110 dB. This can be entered as 100 dB or as 100000.</td>
</tr>
<tr>
<td>COMP</td>
<td></td>
<td></td>
<td>Compensation level selector. This modifies the number of nodes in the equivalent to include external compensation nodes if set to one. See C2 for external compensation settings. COMP=0 internal compensation (Default) COMP=1 external compensation</td>
</tr>
<tr>
<td>DEF</td>
<td></td>
<td></td>
<td>Default model selector. Allows choice of three default settings. 0= generic (0.6 MHz bandwidth) (Default) 1= ua741 (1.2 MHz bandwidth) 2= mc4560 (3 MHz bandwidth)</td>
</tr>
</tbody>
</table>
Performing Behavioral Modeling Using Op-Amps, Comparators, and Oscillators

DELPHS deg Excess phase at the unity gain frequency. Also called the phase margin. DELPHS is measured in degrees. Typical excess phases range from 5° to 50°. To determine DELPHS, subtract the phase at unity gain from 90°; this gives the phase margin. Use the same chart as used for the FREQ determination above. DELPHS interacts with FREQ (or AV1K). Values of DELPHS tend to lower the unity gain bandwidth, particularly values greater than 20°. The model does not have enough poles to always give correct phase and frequency response. It is usually best to pick the DELPHS closest to measured value that does not reduce unity gain bandwidth more than 20%.

DIS amp 1e-16 Diode and BJT saturation current

FREQ (GBW,BW) Hz Unity gain frequency. Measured in hertz and typical frequencies range from 100 kHz to 3 MHz. If not specified, measure open loop frequency response at 0 dB voltage gain and the actual compensation capacitance. Typical compensation is 30 pF and single pole compensation configuration. If AV1K is greater than zero, the unity gain frequency is calculated from AV1K and FREQ is ignored.

IB amp Input bias current. The amount of current required to bias the input differential transistors. This is generally a fundamental electrical characteristic. Typical values are between 20 and 400 nA.
### IBOS amp
Input bias offset current, also called input offset current. This is the amount of unbalanced current between the input differential transistors. Generally a fundamental electrical characteristic. Typical values are 10% to 20% of the IB.

### ISC amp
Input short circuit current – not always specified. Typical values are between 5 and 25 mA. ISC can also be determined from output characteristics (current sinking) as the maximum output sink current. ISC and ROUT interact with each other, if ROUT is too large for a given value of ISC, ROUT is automatically reduced.

### JIS amp
JFET saturation current. Default=1e-16 and need not be changed.

### LEVIN
Input level type selector. Allows only BJT differential pair creation. LEVIN=1 BJT differential input stage.

### LEVOUT
Output level type selector. Allows only single-ended output stage creation. LEVOUT=1 single-ended output stage.

### MANU
Manufacturer's name. This can be added to the model parameter list to identify the source of the model parameters. The name is printed in the final equivalent circuit.

### PWR (PD) watt
Total power dissipation value for the amplifier. This includes the calculated value for the op-amp input differential pair. If high slew rate and very low power is specified a warning is issued and the input differential pair alone gives the power dissipation.

<table>
<thead>
<tr>
<th>Names (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBOS</td>
<td>amp</td>
<td></td>
<td>Input bias offset current, also called input offset current. This is the amount of unbalanced current between the input differential transistors. Generally a fundamental electrical characteristic. Typical values are 10% to 20% of the IB.</td>
</tr>
<tr>
<td>ISC</td>
<td>amp</td>
<td></td>
<td>Input short circuit current – not always specified. Typical values are between 5 and 25 mA. ISC can also be determined from output characteristics (current sinking) as the maximum output sink current. ISC and ROUT interact with each other, if ROUT is too large for a given value of ISC, ROUT is automatically reduced.</td>
</tr>
<tr>
<td>JIS</td>
<td>amp</td>
<td></td>
<td>JFET saturation current. Default=1e-16 and need not be changed.</td>
</tr>
<tr>
<td>LEVIN</td>
<td></td>
<td></td>
<td>Input level type selector. Allows only BJT differential pair creation. LEVIN=1 BJT differential input stage.</td>
</tr>
<tr>
<td>LEVOUT</td>
<td></td>
<td></td>
<td>Output level type selector. Allows only single-ended output stage creation. LEVOUT=1 single-ended output stage.</td>
</tr>
<tr>
<td>MANU</td>
<td></td>
<td></td>
<td>Manufacturer's name. This can be added to the model parameter list to identify the source of the model parameters. The name is printed in the final equivalent circuit.</td>
</tr>
<tr>
<td>PWR (PD)</td>
<td>watt</td>
<td></td>
<td>Total power dissipation value for the amplifier. This includes the calculated value for the op-amp input differential pair. If high slew rate and very low power is specified a warning is issued and the input differential pair alone gives the power dissipation.</td>
</tr>
</tbody>
</table>
Performing Behavioral Modeling Using Op-Amps, Comparators, and Oscillators

<table>
<thead>
<tr>
<th>Names (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAC (r0ac, roac)</td>
<td>ohm</td>
<td>High frequency output resistance. This typically is about 60% of ROUT. RAC usually ranges between 40 to 70 ohms for op-amps with video drive capabilities.</td>
<td></td>
</tr>
<tr>
<td>ROUT</td>
<td>ohm</td>
<td>Low frequency output resistance. This can be determined using the closed loop output impedance graph. The impedance at about 1kHz, using the maximum gain, is close to ROUT. Gains of 1,000 and above show the effective DC impedance, generally in the frequency region between 1k and 10 kHz. Typical values for ROUT are 50 to 100 ohms.</td>
<td></td>
</tr>
<tr>
<td>SRNEG (SRN)</td>
<td>volt</td>
<td>Negative going output slew rate. This is found from the graph of the voltage follower pulse response. This is generally a 4 or 5 volt output change with 10 to 20 volt supplies. Measures the negative going change in voltage and the amount of time for the change.</td>
<td></td>
</tr>
<tr>
<td>SRPOS (SRP)</td>
<td>volt</td>
<td>Positive going output slew rate. This is found from the graph of the voltage follower pulse response. This is generally a 4 or 5 volt output change with 10 to 20 volt supplies. Measures the positive going change in voltage and the amount of time for the change. Typical slew rates are in the range of 70k to 700k.</td>
<td></td>
</tr>
<tr>
<td>TEMP</td>
<td>°C</td>
<td>Temperature in degrees Celsius. This usually is set to the temperature at which the model parameters were measured, which typically is 25 °C.</td>
<td></td>
</tr>
<tr>
<td>VCC</td>
<td>volt</td>
<td>Positive power supply reference voltage for VOPOS. The amplifier VOPOS was measured with respect to VCC.</td>
<td></td>
</tr>
</tbody>
</table>
Op-Amp Model Parameter Defaults

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DEF=0</td>
</tr>
<tr>
<td>AV</td>
<td>Amplifier voltage gain</td>
<td>160k</td>
</tr>
<tr>
<td>AV1K</td>
<td>Amplifier voltage gain at 1 kHz</td>
<td>-</td>
</tr>
<tr>
<td>C2</td>
<td>Feedback capacitance</td>
<td>30 p</td>
</tr>
<tr>
<td>CMRR</td>
<td>Common mode rejection ratio</td>
<td>96 db</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63.1k</td>
</tr>
<tr>
<td>COMP</td>
<td>Compensation level selector</td>
<td>0</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Defaults</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>DEF</td>
<td>Default level selector</td>
<td>DEF=0 1 2</td>
</tr>
<tr>
<td>DELPHS</td>
<td>Delta phase at unity gain</td>
<td>25° 17° 52°</td>
</tr>
<tr>
<td>DIS</td>
<td>Diode saturation current</td>
<td>8e-16 8e-16 8e-16</td>
</tr>
<tr>
<td>FREQ</td>
<td>Unity gain frequency</td>
<td>600 k - -</td>
</tr>
<tr>
<td>IB</td>
<td>Input bias current</td>
<td>30 n 250 n 40 n</td>
</tr>
<tr>
<td>IBOS</td>
<td>Input bias offset current</td>
<td>1.5 n 0.7 n 5 n</td>
</tr>
<tr>
<td>ISC</td>
<td>Output short circuit current</td>
<td>25 mA 25 mA 25 mA</td>
</tr>
<tr>
<td>LEVIN</td>
<td>Input circuit level selector</td>
<td>1 1 1</td>
</tr>
<tr>
<td>LEVOUT</td>
<td>Output circuit level selector</td>
<td>1 1 1</td>
</tr>
<tr>
<td>MANU</td>
<td>Manufacturer’s name</td>
<td>- - -</td>
</tr>
<tr>
<td>PWR</td>
<td>Power dissipation</td>
<td>72 mW 60 mW 50 mW</td>
</tr>
<tr>
<td>RAC</td>
<td>AC output resistance</td>
<td>0 75 70</td>
</tr>
<tr>
<td>ROUT</td>
<td>DC output resistance</td>
<td>200 550 100</td>
</tr>
<tr>
<td>SRPOS</td>
<td>Positive output slew rate</td>
<td>450 k 1 meg 1 meg</td>
</tr>
<tr>
<td>SRNEG</td>
<td>Negative output slew rate</td>
<td>450 k 800 k 800 k</td>
</tr>
<tr>
<td>TEMP</td>
<td>Temperature of model</td>
<td>25 deg 25 deg 25 deg</td>
</tr>
<tr>
<td>VCC</td>
<td>Positive supply voltage for VOPOS</td>
<td>20 15 15</td>
</tr>
<tr>
<td>VEE</td>
<td>Negative supply voltage for VONEG</td>
<td>-20 -15 -15</td>
</tr>
<tr>
<td>VONEG</td>
<td>Maximum negative output</td>
<td>-14 -14 -14</td>
</tr>
<tr>
<td>VOPOS</td>
<td>Maximum positive output</td>
<td>14 14 14</td>
</tr>
<tr>
<td>VOS</td>
<td>Input offset voltage</td>
<td>0 0.3 m 0.5 m</td>
</tr>
</tbody>
</table>
Op-Amp Subcircuit Example

AUTOSTOP Option

This example uses the .OPTION AUTOSTOP option to shorten simulation time. Once Star-Hspice makes the measurements specified by the .MEASURE statement, the associated transient analysis and AC analysis stops whether or not the full sweep range for each has been covered.

AC Resistance

AC=10000G parameter in the Rfeed element statement installs a 10000 GΩ feedback resistor for the AC analysis in place of the 10 kΩ feedback resistor – used in the DC operating point and transient analysis – which is open-circuited for the AC measurements.

Simulation Results

The simulation results give the DC operating point analysis for an input voltage of 0 v and power supply voltages of 15 v. The DC offset voltage is 3.3021 mv, which is less than that specified for the original vos specification in the op-amp .MODEL statement. The unity gain frequency is given as 907.885 kHz, which is within 10% of the 1 MHz specified in the .MODEL statement with the parameter FREQ. The required time rate for a 1 volt change in the output (from the .MEASURE statement) is 2.3 µs (from the SRPOS simulation result listing) providing a slew rate of 0.434 Mv/s. This compares to within about 12% of the 0.5 Mv/s given by the SRPOS parameter in the .MODEL statement. The negative slew rate is almost exactly 0.5 Mv/s, which is within 1% of the slew rate specified in the .MODEL statement.

Example

$$\text{FILE ALM124.SP}$$

.$$\text{.OPTION NOMOD AUTOSTOP SEARCH=''}$$

.$$\text{.OP VOL}$$

.$$\text{.AC DEC 10 1HZ 10MEGHZ}$$

.$$\text{.MODEL PLOTDB PLOT XSCAL=2 YCAL=3}$$

.$$\text{.MODEL PLOTLOGX PLOT XSCAL=2}$$
Performing Behavioral Modeling Using Op-Amps, Comparators, and Oscillators

```plaintext
.GRAPH AC MODEL=PLOTDB VM(OUT0)
.GRAPH AC MODEL=PLOTLOGX VP(OUT0)
.TRAN 1U 40US 5US .15MS
.GRAPH V(IN) V(OUT0)
.MEASURE TRAN 'SRPOS' TRIG V(OUT0) VAL=2V RISE=1 + TARG V(OUT0) VAL=3V RISE=1
.MEASURE TRAN 'SRNEG' TRIG V(OUT0) VAL=-2V FALL=1 + TARG V(OUT0) VAL=-3V FALL=1
.MEASURE AC 'UNITFREQ' TRIG AT=1 + TARG VDB(OUT0) VAL=0 FALL=1
.MEASURE AC 'PHASEMARGIN' FIND VP(OUT0) + WHEN VDB(OUT0)=0
.MEASURE AC 'GAIN(DB)' MAX VDB(OUT0)
.MEASURE AC 'GAIN(MAG)' MAX VM(OUT0)
VCC VCC GND +15V
VEE VEE GND -15V
VIN IN GND AC=1 PWL 0US 0V 1US 0V 1.1US +10V 15US +10V + 15.2US -10V 100US -10V
.MODEL ALM124 AMP
+ C2= 30.00P SRPOS= 0.5MEG SRNEG= 0.5MEG
+ IB= 45N IBOS= 3N VOS= 4M
+ FREQ= 1MEG DELPHS= 25 CMRR= 85
+ ROUT= 50 AV= 100K ISC= 40M
+ VOPOS= 14.5 VONEG= -14.5 PWR= 142M
+ VCC= 16 VEE= -16 TEMP= 25.00
+ PSRR= 100 DIS= 8.00E-16 JIS= 8.00E-16
*

Unity Gain Resistor Divider Mode
*
Rfeed OUT0 IN- 10K AC=10000G
RIN IN IN- 10K
RIN+ IN+ GND 10K
X0 IN- IN+ OUT0 VCC VEE ALM124
ROUT0 OUT0 GND 2K
COUT0 OUT0 GND 100P
.END

***** OPERATING POINT STATUS IS VOLTAGE SIMULATION TIME IS 0.
```

Using Op-Amps, Comparators, and Oscillators

Performing Behavioral Modeling

741 Op-Amp from Controlled Sources

The µA741 op-amp is modeled by PWL controlled sources. The output is limited to ±15 volts by a piecewise linear CCVS (source “h”).

Figure 26-24: Op-Amp Circuit

Example

```
Op_AMP.sp --- operational amplifier
*
.options post=2
.tran .001ms 2ms
```
Performing Behavioral Modeling Using Op-Amps, Comparators, and Oscillators

Main Circuit
xamp input 0 output opamp
vin input 0 sin(0,1m,1k) ac 1
* subcircuit definitions
* input subckt
  .subckt opin in+ in- out
  rin in+ in- 2meg
  rin+ in+ 0 500meg
  rin- in- 0 500meg
  g 0 out pw(1) in+ in- -68mv,-68ma 68mv,68ma delta=1mv
c out 0 .136uf
r out 0 835k
.ends

RC Circuit With Pole At 9 MHz
  .subckt oprc in out
e out1 0 in 0 1
r1 out1 out2 168
r2 out2 out3 1.68k
r3 out3 out4 16.8k
r4 out4 out 168k
c1 out2 0 100p
c2 out3 0 10p
c3 out4 0 1p
c4 out 0 .1p
r out 0 1e12
.ends

Output Limiter to 15 v
  .subckt opout in out
eo out1 0 in 0 1
Using Op-Amps, Comparators, and Oscillators

Performing Behavioral Modeling

```plaintext
ro out1 out 75
evum out dum 0
h dum 0 pwl(1) v Dum delta=.01ma -.1ma,-15v .1ma,15v
.ends
* op-amp subckt
.subckt opamp in+ in- out
xin in+ in- out1 opin
xrc out1 out2 oprc
xout out2 out opout
.ends
.ends
```

Figure 26-25: AC Analysis Response
Inverting Comparator with Hysteresis

An inverting comparator is modelled by a piecewise linear VCVS.
Two reference voltages corresponding to volow and vohigh of Ecomp characteristic are:

\[ V_{reflow} = \frac{Volow \cdot Rb}{Rb + Rf} \]

\[ V_{refhigh} = \frac{Vohigh \cdot Rb}{Rb + Rf} \]

When Vin exceeds Vrefhigh, the output Vout goes to Volow. For Vin less than Vreflow, the output goes to Vohigh.

**Example**

Compar.sp Inverting comparator with hysteresis
.OPTIONS POST PROBE
.PARAM vohigh=5v volow=-2.5v rbval=1k rfval=9k
Ecomp out 0 PWL(1) a b -2u,vohigh 1u,volow
Rb b 0 rbval
Rf b out rfval
Cb b 0 1ff
Vin a 0 PWL(0,-4 1u,4 2u,-4)
.TRAN .1n 2u
.PROBE Vin=V(a) Vab=V(a,b) Vout=V(out)
.END
Voltage Controlled Oscillator (VCO)

In this example, a one-input NAND functioning as an inverter models a five stage ring oscillator. PWL capacitance is used to switch the load capacitance of this inverter from 1pF to 3 pF. As the simulation results indicate, the oscillation frequency decreases as the load capacitance increases.

Example

```verbatim
vcoob.sp voltage controlled oscillator using pwl functions
.OPTION POST
.GLOBAL ctrl
.TRAN 1n 100n
.IC V(in)=0 V(out1)=5
.PROBE TRAN V(in) V(out1) V(out2) V(out3) V(out4)
X1 in out1 inv
X2 out1 out2 inv
X3 out2 out3 inv
```
X4 out3 out4 inv
X5 out4 in inv
Vctrl ctrl 0 PWL(0,0 35n,0 40n,5)

**Subcircuit Definition**

.SUBCKT inv in out rout=1k
* The following G Element is functioning as PWL capacitance.
Gcout out 0 VCCAP PWL(1) ctrl 0 DELTA=.01
+ 4.5 1p
+ 4.6 3p
Rout out 0 rout
Gn 0 out NAND(1) in 0 SCALE='1.0k/rout'
+ 0. 5.00ma
+ 0.25 4.95ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.20ma
+ 5.0 0.05ma
.ENDS inv
*
.END
**LC Oscillator**

The capacitor is initially charged to 5 volts. The value of capacitance is the function of voltage at node 10. The value of capacitance becomes four times higher at time $t_2$. The frequency of this LC circuit is given by:

$$freq = \frac{1}{6.28 \cdot \sqrt{L \cdot C}}$$

At time $t_2$, the frequency must be halved. The amplitude of oscillation depends on the condition of the circuit when the capacitance value changes. The stored energy is:

$$E = (0.5 \cdot C \cdot V^2) + (0.5 \cdot L \cdot I^2)$$

$$E = 0.5 \cdot C \cdot Vm^2, I = 0$$
Using Op-Amps, Comparators, and Oscillators

Performing Behavioral Modeling

\[ E = 0.5 \cdot L \cdot I m^2, \ V = 0 \]

Assuming at time \( t2 \), when \( V=0 \), \( C \) changes to \( A \cdot C \), then:

\[ 0.5 \cdot L \cdot I m^2 = 0.5 \cdot V m^2 = 0.5 \cdot (A \cdot C) \cdot V m^2 \]

and from the above equation:

\[ V m' = \frac{V m}{\sqrt{A}} \]

\[ Q m' = \sqrt{A} \cdot V m \]

The second condition to consider is when \( V=Vin \), \( C \) changes to \( A \cdot C \). In this case:

\[ Q m = Q m' \]

\[ C \cdot V m = A \cdot C \cdot V m' \]

\[ V m' = \frac{V m}{A} \]

Therefore, the voltage amplitude is modified between \( V m/\sqrt{A} \) and \( V m/A \) depending on the circuit condition at the switching time. This example tests the CTYPE 0 and 1 results. The result for CTYPE=1 must be correct because capacitance is a function of voltage at node 10, not a function of the voltage across the capacitor itself.

Example

calg2.sp voltage variable capacitance
*
.OPTION POST
.IC v(1)=5 v(2)=5
C1 1 0 C='1e-9*V(10)' CTYPE=1
L1 1 0 1m
Performing Behavioral Modeling Using Op-Amps, Comparators, and Oscillators

*  
C2 2 0 C='1e-9*V(10)' CTYPE=0  
L2 2 0 1m  
*  
V10 10 0 PWL(0sec,1v t1,1v t2,4v)  
R10 10 0 1  
*  
.TRAN .1u 60u UIC SWEEP DATA=par  
.MEAS TRAN period1 TRIG V(1) VAL=0 RISE=1  
  + TARG V(1) VAL=0 RISE=2  
.MEAS TRAN period2 TRIG V(1) VAL=0 RISE=5  
  + TARG V(1) VAL=0 RISE=6  
.PROBE TRAN V(1) q1=LX0(C1)  
*  
.PROBE TRAN V(2) q2=LX0(C2)  
.DATA par t1 t2  
  15.65us 15.80us  
  17.30us 17.45us  
.ENDDATA  
.END
Figure 26-30: Correct Result Corresponding to CTYPE=1
Figure 26-31: Incorrect Result Corresponding to CTYPE=0
Using a Phase Locked Loop Design

Phase Detector Using Multi-Input NAND Gates

This circuit uses the behavioral elements to implement inverters, 2, 3, and 4 input NAND gates.

**Figure 26-32: Circuit Schematic of Phase Detector**

![Circuit Schematic of Phase Detector]

**Example**

```plaintext
pdb.sp phase detector using behavioral nand gates.
.option post=2
.tran .25n 50ns
*.graph tran v(r) v(v) v(u1)
*.graph tran v(r) v(v) v(u2) $ v(d2)
.probe tran v(r) v(v) v(u1)
.probe tran v(r) v(v) v(u2) $ v(d2)
xnr r u1 nr nand2 capout=.1p
xq1 nr q2 q1 nand2 capout=.1p
xq2 q1 n4 q2 nand2
```

Performing Behavioral Modeling Using a Phase Locked Loop Design

```
xq3 q4 n4 q3 nand2
xq4 q3 nv q4 nand2
xnv v d1 nv nand2
xul nr q1 n4 ul nand3
xd1 nv q4 n4 d1 nand3
xvn v vn inv
xu2 vn r u2 nand2
xd2 r v d2 nand2
xn4 nr q1 q4 nv n4 nand4
*
* waveform vv lags waveform vr
vr r 0 pulse(0,5,0n,1n,1n,15n,30n)
vv v 0 pulse(0,5,5n,1n,1n,15n,30n)
*
* waveform vr lags waveform vv
*vr r 0 pulse(0,5,5n,1n,1n,15n,30n)
*vv v 0 pulse(0,5,0n,1n,1n,15n,30n)
```

Subcircuit Definitions

```
.SUBCKT inv in out capout=.1p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(1) in 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS inv
*
.SUBCKT nand2 in1 in2 out capout=.15p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(2) in1 0 in2 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
```

+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS nand2

* .SUBCKT nand3 in1 in2 in3 out capout=.2p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(3) in1 0 in2 0 in3 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS nand3

* .SUBCKT nand4 in1 in2 in3 in4 out capout=.5p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(4) in1 0 in2 0 in3 0 in4 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS nand4
.end
Figure 26-33: Phase Detector Response
PLL BJT Behavioral Modeling

Figure 26-34: PLL Schematic

Example

A Phase Locked Loop (PLL) circuit synchronizes to an input waveform within a selected frequency range, returning an output voltage proportional to variations in the input frequency. It has three basic components: a voltage controlled oscillator (VCO), which returns an output waveform proportional to its input voltage, a phase detector which compares the VCO output to the input waveform and returns an output voltage depending on their phase difference, and a loop filter, which filters the phase detector voltage, returning an output voltage which forms the VCO input and the external voltage output of the PLL.
The following example shows a Star-Hspice simulation of a full bipolar implementation of a PLL; its transfer function shows a linear region of voltage vs. (periodic) time which is defined as the “lock” range. The phase detector is modeled behaviorally, effectively implementing a logical XNOR function. This model was then substituted into the full PLL circuit and resimulated. The behavioral model for the VCO was then substituted into the PLL circuit, and this behavioral PLL was then simulated. The results of the transient simulations (Figure 26-35) show minimal difference between implementations, but from the standpoint of run time statistics, the behavioral model shows a factor of five reduction in simulation time versus that of the full circuit.

Include the behavioral model if you use this PLL in a larger system simulation (for example, an AM tracking system) because it substantially reduces run time while still representing the subcircuit accurately.

**Figure 26-35: Behavioral (PLL_BVP Curve) vs. Bipolar (PLL_BJT Curve) Circuit Simulation**
Example

This is an example of a phase locked loop:

```
$ phase locked loop
   .option post probe acct
   .option relv=1e-5
$

$ wideband FM example, Grebene gives:
$ f0=1meg kf=250kHz/V
$ kd=0.1 V/rad
$ R=10K C=1000p
$ f_lock = kf*kd*pi/2 = 39kHz, v_lock = kd*pi/2 = 0.157
$ f_capture/f_lock ~= 1/sqrt(2*pi*R*C*f_lock)
$ v_capture ~= 0.63, v_capture ~= 0.100

*.ic v(out)=0 v(fin)=0
.tran .2u 500u
   .option delmax=0.01u interp
.probe v_in=v(inc,0) v_out=v(out,outb)
     .probe v(in) v(osc) v(mout) v(out)

Input

vin inc 0 pwl 0u,-0.2 500u,0.2
*vin inc 0 0
xin inc 0 in inb vco f0=1meg kf=125k phi=0 out_off=0
     out_amp=0.3
$vco
xvco e eb osc oscb vco f0=1meg kf=125k phi=0 out_off=-1
     out_amp=0.3

$ phase detector
xpd in inb osc oscb mout moutb pd kd=0.1 out_off=-2.5

$ filter
rf mout e 10k
cf e 0 1000p
rfb moutb eb 10k
cfb eb 0 1000p

$ final output
rout out e 100k
```
Performing Behavioral Modeling Using a Phase Locked Loop Design


cout out 0 100p
routb outb eb 100k
coutb outb 0 100p

.macro vco in inb out outb f0=100k kf=50k phi=0.0 out_off=0.0
out_amp=1.0
gs 0 s poly(2) c 0 in inb 0 ‘6.2832e-9*f0’ 0 0 ‘6.2832e-9*kf’
gc c 0 poly(2) s 0 in inb 0 ‘6.2832e-9*f0’ 0 0 ‘6.2832e-9*kf’
cs s 0 1e-9
cc c 0 1e-12
e1 s_clip 0 pwl(1) s 0 -0.1,-0.1 0.1,0.1
eout 0 s_clip 0 out_off vol=’10*out_amp’
eboutb 0 s_clip 0 out_off vol=’-10*out_amp’
.ic v(s)='sin(phi)' v(c)=’cos(phi)’
.eom

.macro pd in inb in2 in2b out outb kd=0.1 out_off=0
e1 clip1 0 pwl(1) in inb -0.1,-0.1 0.1,0.1
e2 clip2 0 pwl(1) in2 in2b -0.1,-0.1 0.1,0.1
e3 n1 0 poly(2) clip1 0 clip2 0 0 0 0 0 ‘78.6*kd’
e4 outb 0 n1 0 out_off 1
e5 out 0 n1 0 out_off -1
.eom

.end

This is an example of a BJT LEVEL Voltage Controlled Oscillator (VCO):

$ phase locked loop
.option post probe acct
.option relv=1e-5
$
$ wideband FM example, Grebene gives:
$ f0=1meg kf=250kHz/V
$ kd=0.1 V/rad
$ R=10K C=1000p
$ f_lock = kf*kd*pi/2 = 39kHz, v_lock = kd*pi/2 = 0.157
$ f_capture/f_lock =~ 1/sqrt(2*pi*R*C*f_lock)
$ = 0.63, v_capture =~ 0.100
$ .ic v(out)=0 v(fin)=0
.tran .2u 500u

26-95
Using a Phase Locked Loop Design

Performing Behavioral Modeling

```
.option delmax=0.01u interp
.probe v_in=v(inc,0) v_out=v(out,outb)
.probe v(in) v(osc) v(mout) v(out) v(e)

vcc vcc 0 6
vee vee 0 -6

$ input
vin inc 0 pw1 0u,-0.2 500u,0.2
xin inc 0 in inb vco f0=1meg kf=125k phi=0 out_off=0
out_amp=0.3

$ vco
xvcol e eb osc oscb 0 vee vcol
.ic v(osc)=-1.4 v(oscb)=-0.7
```

Figure 26-36: Voltage Controlled Oscillator Circuit
BJT Level Phase Detector

Example

$ phase detector
xpd1 in inb osc oscb mout moutb vcc vee pd1

Filter

rf mout e 10k
cf e 0 1000p
rfb mouth eb 10k
cfb eb 0 1000p

Final Output

rout out e 100k
cout out 0 100p
routb outb eb 100k
coutb outb 0 100p

.macro vco in inb out outb f0=100k kf=50k phi=0.0 out_off=0.0 out_amp=1.0
gs 0 s poly(2) c 0 in inb 0 '6.2832e-9*f0' 0 0 '6.2832e-9*kf'
 gc c 0 poly(2) s 0 in inb 0 '6.2832e-9*f0' 0 0 '6.2832e-9*kf'
 cs s 0 1e-9
 cc c 0 1e-9
e1 s_clip 0 pwl(1) s 0 -0.1,-0.1 0.1,0.1
e out 0 s_clip 0 out_off '10*out_amp'
eb outb 0 s_clip 0 out_off '-10*out_amp'
.ic v(s)='sin(phi)' v(c)='cos(phi)'
.eom

.macro pd in inb in2 in2b out outb kd=0.1 out_off=0
e1 clip1 0 pwl(1) in inb -0.1,-0.1 0.1,0.1
e2 clip2 0 pwl(1) in2 in2b -0.1,-0.1 0.1,0.1
e3 nl 0 poly(2) clip1 0 clip2 0 0 0 0 0 0 0 '78.6*kd'
e4 outb 0 nl 0 out_off 1
e5 out 0 nl 0 out_off -1
.eom

.macro vcol in inb e7 e8 vcc vee vco_cap=228.5p
gout vcc vcc b7 npn1
Using a Phase Locked Loop Design

Performing Behavioral Modeling

```
qoutb vcc vcc b8 npn1
rb vcc c0 5k $ 1ma
q0 c0 b0 vee npn1
q7 vcc b7 e7 npn1
r4 vcc b7 1k
i7 e7 0 1m
q8 vcc b8 e8 npn1
r5 vcc b8 1k
i8 e8 0 1m
q9 b7 e8 e9 npn1
q10 b8 e7 e10 npn1
c0 e9 e10 vco_cap
q11 e9 in 2 npn1 $ ic=i0
q12 e10 in 2 npn1 $ ic=i0
q15 c0 b0 npn1 $ ic=2*i0
q16 c0 b0 npn1 $ ic=2*i0
rx 2 3 8k
q13 vcc inb 3 npn1
q14 vcc inb 3 npn1
rt b0 vee 350 $ i=4*i0=2m
.eom

.model npn1 npn
+ eg=1.1 af=1 xcjc=0.95 subs=1
+ cjs=0 tf=5p
+ tr=500p cje=0.2p cjc=0.2p fc=0.8
+ vje=0.8 vjc=0.8 mje=0.33 mjc=0.33
+ rb=0 rbm=0 irb=10u
+ is=5e-15 ise=1.5e-14 isc=0
+ vaf=150 bf=100 ikf=20m
+ var=30 br=5 ikr=15m
+ rc=0 re=0
+ nf=1 ne=1.5 nc=1.2
+ tbf=8e-03

.macro pd1 in inb in2 in2b out outb vcc vee
r1 vcc n1 1k
rlb vcc n1b 1k
q3 n1 in c1 npn1
q4 n1b inb c1 npn1
q5 n1 inb c2 npn1
q6 n1b in c2 npn1
```

Performing Behavioral Modeling Using a Phase Locked Loop Design

```
q1 c1 in2 e npn1
c2 c1b in2b e npn1
ie e 0 0.5m
cl n1 0 lp
c1b n1b 0 lp
q7 vcc n1 e7 npn1
q8 vcc n1b e8 npn1
r1 e7 out 625
r2 out vee 300
r1b e8 outb 625
r2b outb vee 300
.eom
.end
```

**Figure 26-37: Phase Detector Circuit**

![Phase Detector Circuit Diagram]
References

Performing Pole/Zero Analysis

Pole/zero analysis is a useful method for studying the behavior of linear, time-invariant networks, and may be applied to the design of analog circuits, such as amplifiers and filters. It may be used for determining the stability of a design, and it may also be used to calculate the poles and zeroes for specification in a POLE statement as “Using Pole/Zero Analysis” on page 27-3 describes.

Pole/zero analysis is characterized by the use of the .PZ statement, as opposed to pole/zero and Laplace transfer function modeling, which employ the LAPLACE and POLE functions respectively. These are described in “Using Pole/Zero Analysis” on page 27-3.

This chapter covers these topics:

- Understanding Pole/Zero Analysis
- Using Pole/Zero Analysis
Understanding Pole/Zero Analysis

In pole/zero analysis, a network is described by its network transfer function which, for any linear time-invariant network, can be written in the general form:

\[
H(s) = \frac{N(s)}{D(s)} = \frac{a_0 s^m + a_1 s^{(m-1)} + \ldots + a_m}{b_0 s^n + b_1 s^{(n-1)} + \ldots + b_n}
\]

In the factorized form, the general function is:

\[
H(s) = \frac{a_0}{b_0} \cdot \frac{(s + z_1)(s + z_2)\ldots(s + z_i)(s + z_m)}{(s + p_1)(s + p_2)\ldots(s + p_j)(s + p_m)}
\]

The roots of the numerator N(s) (that is, \(z_i\)) are called the zeros of the network function, and the roots of the denominator D(s) (that is, \(p_j\)) are called the poles of the network function. S is a complex frequency\(^1\).

The dynamic behavior of the network depends upon the location of the poles and zeros on the network function curve. The poles are called the natural frequencies of the network. In general, you can graphically deduce the magnitude and phase curve of any network function from the location of its poles and zeros\(^2\).

The section “References” on page 27-20, lists a variety of source material addressing transfer functions of physical systems\(^3\), design of systems and physical modeling\(^4\), and interconnect transfer function modeling\(^5-6\).
Using Pole/Zero Analysis

Star-Hspice uses the Muller method to calculate the roots of polynomials \( N(s) \) and \( D(s) \). This method approximates the polynomial with a quadratic equation that fits through three points in the vicinity of a root. Successive iterations toward a particular root are obtained by finding the nearer root of a quadratic whose curve passes through the last three points.

In Muller’s method, the selection of the three initial points affects the convergence of the process and accuracy of the roots obtained. If the poles or zeros are spread over a wide frequency range, choose \((X_0R, X_0I)\) close to the origin to find poles or zeros at zero frequency first. Then find the remaining poles or zeros in increasing order. The values \((X_1R, X_1I)\) and \((X_2R, X_2I)\) may be orders of magnitude larger than \((X_0R, X_0I)\). If there are poles or zeros at high frequencies, \(X_1I\) and \(X_2I\) should be adjusted accordingly.

Pole/zero analysis results are based on the circuit’s DC operating point, so the operating point solution must be accurate. Consequently, the .NODESET statement (not .IC) is recommended for initialization to avoid DC convergence problems.

.PZ (Pole/Zero) Statement

The syntax is:

```
.PZ output input
```

- **PZ** Invokes the pole/zero analysis
- **input** Input source, which may be any independent voltage or current source name
- **output** Output variables, which may be any node voltage, \(V(n)\), or any branch current, \(I(element\ name)\)

Example

```
.PZ V(10) VIN
.PZ I(RL) ISORC
```
Using Pole/Zero Analysis

Performing Pole/Zero Analysis

.PZ  I1(M1)  VSRC

**Pole/Zero Control Options**

*CSCAL*  
Sets the capacitance scale. Capacitances are multiplied by CSCAL. Default=1e+12.

*FMAX*  
Sets the maximum pole and zero angular frequency value. Default=1.0e+12 rad/sec.

*FSCAL*  
Sets the frequency scale. Frequency is multiplied by FSCAL. Default=1e-9.

*GSCAL*  
Sets the conductance scale. Conductances are multiplied by GSCAL, and resistances are divided by GSCAL. Default=1e+3.

*ITLPZ*  
Sets the pole/zero analysis iteration limit. Default=100.

*LSCAL*  
Sets the inductance scale. Inductances are multiplied by LSCAL. Default=1e+6.

**Note:** The scale factors must satisfy the following relations.

\[
GSCAL = CSCAL \cdot FSCAL
\]

\[
GSCAL = \frac{1}{LSCAL \cdot FSCAL}
\]

If scale factors are changed, the initial Muller points, \((X0R, X0I), (X1R, X1I)\) and \((X2R, X2I)\), may have to be modified, even though internally the program multiplies the initial values by \((1e-9/GSCAL)\).

*PZABS*  
Sets absolute tolerances for poles and zeros. This option affects the low frequency poles or zeros. It is used as follows:

If \((|X_{real}| + |X_{imag}| < PZABS)\),

then \(X_{real} = 0\) and \(X_{imag} = 0\).

This option is also used for convergence tests. Default=1e-2.
Performing Pole/Zero Analysis

**PZTOL**
Sets the relative error tolerance for poles or zeros. Default=1.0e-6.

**RITOL**
Sets the minimum ratio value for (real/imaginary) or (imaginary/real) parts of the poles or zeros. Default 1.0e-6. RITOL is used as follows:

If $|X_{\text{imag}}| \leq \text{RITOL} \cdot |X_{\text{real}}|$, then $X_{\text{imag}} = 0$

If $|X_{\text{real}}| \leq \text{RITOL} \cdot |X_{\text{imag}}|$, then $X_{\text{real}} = 0$

$$(X0R,X0I)$$
the three complex starting trial points in the Muller
$$(x1R,X1I)$$
algorithm for pole/zero analysis. Defaults:
$$(X2R,X21)$$

$X0R=-1.23456e6 \quad X0I=0.0$
$X1R=1.23456e5 \quad X1I=0.0$
$X2R=+1.23456e6 \quad X21=0.0$

These initial points and FMAX are multiplied by FSCAL.

### Pole/Zero Analysis Examples

#### Example 1 – Low-Pass Filter

The following is an HSPICE input file for a low-pass prototype filter for pole/zero and AC analysis. This file can be found in `$installdir/demo/hspice/filters/flp5th.sp`.

**Fifth-Order Low-Pass Filter HSPICE File**

```plaintext
*FILE: FLP5TH.SP
5TH-ORDER LOW_PASS FILTER
****
* T = I(R2) / IIN
*   = 0.113*(S**2 + 1.6543)*(S**2 + 0.2632) /
*   (S**5 + 0.9206*S**4 + 1.26123*S**3 +
*    0.74556*S**2 + 0.2705*S + 0.09836)
****
.OPTIONS POST
.PZ I(R2) IN
.AC DEC 100 .001HZ 10HZ
```

Using Pole/Zero Analysis

Performing Pole/Zero Analysis

```
.PLOT AC IDB(R2) IP(R2)
IN 0 1 1.00 AC 1
R1 1 0 1.0
C3 1 0 1.52
C4 2 0 1.50
C5 3 0 0.83
C1 1 2 0.93
L1 1 2 0.65
C2 2 3 3.80
L2 2 3 1.00
R2 3 0 1.00
.END
```

Figure 27-1: Low-Pass Prototype Filter

Table 27-1 shows the magnitude and phase variation of the current output resulting from AC analysis. These results are consistent with the pole/zero analysis. The pole/zero unit is radians per second or hertz. The X-axis unit in the plot is in hertz.

Table 27-1: Pole/Zero Analysis Results for Low-Pass Filter

<table>
<thead>
<tr>
<th>Poles (rad/sec)</th>
<th>Poles (hertz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>Imag</td>
</tr>
<tr>
<td>-6.948473e-02</td>
<td>-4.671778e-01</td>
</tr>
<tr>
<td>-6.948473e-02</td>
<td>4.671778e-01</td>
</tr>
<tr>
<td>-1.182742e-01</td>
<td>-8.914907e-01</td>
</tr>
<tr>
<td>-1.182742e-01</td>
<td>8.914907e-01</td>
</tr>
<tr>
<td>-5.450890e-01</td>
<td>0.000000e+00</td>
</tr>
</tbody>
</table>
Table 27-1: Pole/Zero Analysis Results for Low-Pass Filter

<table>
<thead>
<tr>
<th>Zeros (rad/sec)</th>
<th>Zeros (hertz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>Imag</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>-1.286180e+00</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>-5.129892e-01</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>5.129892e-01</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>1.286180e+00</td>
</tr>
</tbody>
</table>

Constant Factor = 1.129524e-01

Figure 27-2: Fifth-Order Low-Pass Filter Response
Example 2 – Kerwin’s Circuit

The following is an HSPICE input file for pole/zero analysis of Kerwin’s circuit. This file can be found in $installdir/demo/hspice/filters/fkerwin.sp. Table 27-2 lists the results of the analysis.

Kerwin’s Circuit HSPICE File

```
*FILE: FKERWIN.SP
KERWIN'S CIRCUIT HAVING JW-AXIS TRANSMISSION ZEROS.
**
* T = V(5) / VIN
* = 1.2146 (S**2 + 2) / (S**2 + 0.1*S + 1)
* POLES = (-0.05004, +0.9987), (-0.05004, -0.9987)
* ZEROS = (0.0, +1.4142), (0.0, -1.4142)
*****
.PZ  V(5)  VIN
VIN  1  0  1
C1   1  2  0.7071
C2   2  4  0.7071
C3   3  0  1.4142
C4   4  0  0.3536
R1   1  3  1.0
R2   3  4  1.0
R3   2  5  0.5
E1   5  0  4  0  2.4293
.END
```
Figure 27-3: Design Example for Kerwin’s Circuit

Table 27-2: Pole/Zero Analysis Results for Kerwin’s Circuit

<table>
<thead>
<tr>
<th>Poles (rad/sec)</th>
<th>Poles (hertz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>Imag</td>
</tr>
<tr>
<td>-5.003939e-02</td>
<td>9.987214e-01</td>
</tr>
<tr>
<td>-5.003939e-02</td>
<td>-9.987214e-01</td>
</tr>
<tr>
<td>-1.414227e+00</td>
<td>0.000000e+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zeros (rad/sec)</th>
<th>Zeros (hertz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>Imag</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>-1.414227e+00</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>1.414227e+00</td>
</tr>
<tr>
<td>-1.414227e+00</td>
<td>0.000000e+00</td>
</tr>
</tbody>
</table>

Constant Factor = 1.214564e+00
Example 3 – High-Pass Butterworth Filter

The following is an HSPICE input file for pole/zero analysis of a high-pass Butterworth filter. This file can be found in $installdir/demo/hspice/filters/fhp4th.sp. The analysis results are shown in Table 27-3.

Fourth-Order High-Pass Butterworth Filter HSPICE File

*FILE: FHP4TH.SP
*****
* $T = \frac{V(10)}{VIN}$
* $= \frac{(S^4)}{((S^2 + 0.7653S + 1) \cdot (S^2 + 1.8477S + 1))}$
* 
* POLES, (-0.38265, +0.923895), (-0.38265, -0.923895)
* (-0.9239, +0.3827), (-0.9239, -0.3827)
* ZEROS, FOUR ZEROS AT (0.0, 0.0)
*****
.OPTIONS ITLPZ=200
.PZ V(10) VIN
VIN 1 0 1
C1 1 2 1
C2 2 3 1
R1 3 0 2.613
R2 2 4 0.3826
E1 4 0 3 0 1
C3 4 5 1
C4 5 6 1
R3 6 0 1.0825
R4 5 10 0.9238
E2 10 0 6 0 1
RL 10 0 1E20
.END
Figure 27-4: Fourth-Order High-Pass Butterworth Filter
Example 4 – CMOS Differential Amplifier

The following is an HSPICE input file for pole/zero analysis of a CMOS differential amplifier for pole/zero and AC analysis. The file can be found in $installdir/demo/hspice/apps/mcdiff.sp. The analysis results are shown in Table 27-4.

CMOS Differential Amplifier HSPICE File

FILE: MCDIFF.SP
CMOS DIFFERENTIAL AMPLIFIER
.OPTIONS PIVOT SCALE=1E-6 SCALM=1E-6 WL
.PZ V(5) VIN
VIN 7 0 0 AC 1
.AC DEC 10 20K 500MEG
.PRINT AC VDB(5) VP(5)
Performing Pole/Zero Analysis Using Pole/Zero Analysis

M1  4  0  6  6  MN  100  10  2  2
M2  5  7  6  6  MN  100  10  2  2
M3  4  4  1  1  MP  60  10  1.5  1.5
M4  5  4  1  1  MP  60  10  1.5  1.5
M5  6  3  2  2  MN  50  10  1.0  1.0
VDD  1  0  5
VSS  2  0  -5
VGG  3  0  -3
RIN  7  0  1

.MODEL MN NMOS LEVEL=5 VT=1 UB=700 FRC=0.05 DNB=1.6E16
+ XJ=1.2 LATD=0.7 CJ=0.13 PHI=1.2 TCV=0.003 TOX=800
$.

.MODEL MP PMOS LEVEL=5 VT=-1 UB=245 FRC=0.25 TOX=800
+ DNB=1.3E15 XJ=1.2 LATD=0.9 CJ=0.09 PHI=0.5 TCV=0.002
.END

Figure 27-5: CMOS Differential Amplifier
Example 5 – Simple Amplifier

The following is an HSPICE input file for pole/zero analysis of an equivalent circuit of a simple amplifier with Rs=Rpi=RL=1000 ohms, gm=0.04 mho, CMU=1.0e-11 farad, and CPI=1.0e-9 farad. The file can be found in $installdir/demo/hspice/apps/ampg.sp. The analysis results are shown in Table 27-5.

**Amplifier HSPICE File**

```
FILE: AMPG.SP
A SIMPLE AMPLIFIER.
* T = V(3) / VIN
* T = 1.0D6*(S - 4.0D9) / (S**2 + 1.43D8*S + 2.0D14)
* POLES = (-0.14D7, 0.0), (-14.16D7, 0.0)
* ZEROS = (+4.00D9, 0.0)
.PZ V(3) VIN
RS 1 2 1K
RPI 2 0 1K
```

**Table 27-4: Pole/Zero Analysis Results for CMOS Differential Amplifier**

<table>
<thead>
<tr>
<th>Poles (rad/sec)</th>
<th>Poles (hertz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>Imag</td>
</tr>
<tr>
<td>-1.798766e+06</td>
<td>0.000000e+00</td>
</tr>
<tr>
<td>-1.126313e+08</td>
<td>-6.822910e+07</td>
</tr>
<tr>
<td>-1.126313e+08</td>
<td>6.822910e+07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zeros (rad/sec)</th>
<th>Zeros (hertz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>Imag</td>
</tr>
<tr>
<td>-1.315386e+08</td>
<td>7.679633e+07</td>
</tr>
<tr>
<td>-1.315386e+08</td>
<td>-7.679633e+07</td>
</tr>
<tr>
<td>7.999613e+08</td>
<td>0.000000e+00</td>
</tr>
</tbody>
</table>

Constant Factor = 3.103553e-01
Performing Pole/Zero Analysis

Figure 27-6: Simple Amplifier

Table 27-5: Pole/Zero Analysis Results for Amplifier

<table>
<thead>
<tr>
<th>Poles (rad/sec)</th>
<th>Poles (hertz)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Real</td>
<td>Imag</td>
</tr>
<tr>
<td>-1.412555e+06</td>
<td>0.000000e+00</td>
</tr>
<tr>
<td>-1.415874e+08</td>
<td>0.000000e+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zeros (rad/sec)</th>
<th>Zeros (hertz)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Real</td>
<td>Imag</td>
</tr>
<tr>
<td>-1.456000e+09</td>
<td>0.000000e+00</td>
</tr>
</tbody>
</table>

Constant Factor = 1.000000e+06

Example 6—Active Low-Pass Filter

The following is an HSPICE input file for pole/zero analysis of an active ninth-order low-pass filter using the ideal op-amp element. AC analysis is performed. The file can be found in $installdir/demo/hspice/filters/flp9th.sp. The analysis results are shown in Table 27-6.
Ninth Order Low-Pass Filter HSPICE File

FILE: FLP9TH.SP
******
VIN IN 0 AC 1
.P2 V(OUT) VIN
.AC DEC 50 .1K 100K
.OPTIONS POST DCSTEP=1E3 XOR=-1.23456E+3 X1R=-1.23456E+2
+ X2R=1.23456E+3 FSCAL=1E-6 GSCAL=1E3 CSCAL=1E9 LSCAL=1E3
.PLOT AC VDB(OUT)
.SUBCKT OPAMP IN+ IN- OUT GM1=2 RI=1K CI=26.6U GM2=1.33333 RL=75
RII IN+ IN- 2MEG
RI1 IN+ 0 500MEG
RI2 IN- 0 500MEG
G1 1 0 IN+ IN- GM1
C1 1 0 CI
R1 1 0 RI
G2 OUT 0 1 0 GM2
RLD OUT 0 RL
.ENDS
.SUBCKT FDNR 1 R1=2K C1=12N R4=4.5K
RLX=75
R1 1 2 R1
C1 2 3 C1
R2 3 4 3.3K
R3 4 5 3.3K
R4 5 6 R4
C2 6 0 10N
XOP1 2 4 5 OPAMP
XOP2 6 4 3 OPAMP
.ENDS
*
RS IN 1 5.4779K
R12 1 2 4.44K
R23 2 3 3.2201K
R34 3 4 3.63678K
R45 4 OUT 1.2201K
C5 OUT 0 10N
X1 1 FDNR R1=2.0076K C1=12N R4=4.5898K
X2 2 FDNR R1=5.9999K C1=6.8N R4=4.25725K
X3 3 FDNR R1=5.88327K C1=4.7N R4=5.62599K
X4 4 FDNR R1=1.0301K C1=6.8N R4=5.808498K
.END
Figure 27-7: Linear Model of the 741C Op-Amp

Figure 27-8: The FDNR Subcircuit

Figure 27-9: Active Realization of the Low-Pass Filter
### Table 27-6: Pole/Zero Analysis Results for the Active Low-Pass Filter

<table>
<thead>
<tr>
<th>Poles (rad/sec)</th>
<th>Poles (hertz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>Imag</td>
</tr>
<tr>
<td>-4.505616e+02</td>
<td>-2.210451e+04</td>
</tr>
<tr>
<td>-4.505616e+02</td>
<td>2.210451e+04</td>
</tr>
<tr>
<td>-1.835284e+03</td>
<td>2.148369e+04</td>
</tr>
<tr>
<td>-1.835284e+03</td>
<td>-2.148369e+04</td>
</tr>
<tr>
<td>-4.580172e+03</td>
<td>1.944579e+04</td>
</tr>
<tr>
<td>-4.580172e+03</td>
<td>-1.944579e+04</td>
</tr>
<tr>
<td>-9.701962e+03</td>
<td>1.304893e+04</td>
</tr>
<tr>
<td>-9.701962e+03</td>
<td>-1.304893e+04</td>
</tr>
<tr>
<td>-1.353908e+04</td>
<td>0.000000e+00</td>
</tr>
<tr>
<td>-3.668995e+06</td>
<td>-3.669793e+06</td>
</tr>
<tr>
<td>-3.668995e+06</td>
<td>3.669793e+06</td>
</tr>
<tr>
<td>-3.676439e+06</td>
<td>-3.676184e+06</td>
</tr>
<tr>
<td>-3.676439e+06</td>
<td>3.676184e+06</td>
</tr>
<tr>
<td>-3.687870e+06</td>
<td>3.687391e+06</td>
</tr>
<tr>
<td>-3.687870e+06</td>
<td>-3.687391e+06</td>
</tr>
<tr>
<td>-3.695817e+06</td>
<td>-3.695434e+06</td>
</tr>
<tr>
<td>-3.695817e+06</td>
<td>+3.695434e+06</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zeroes (rad/sec)</th>
<th>Zeroes (hertz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>Imag</td>
</tr>
<tr>
<td>-3.220467e-02</td>
<td>-2.516970e+04</td>
</tr>
<tr>
<td>-3.220467e-02</td>
<td>2.516970e+04</td>
</tr>
<tr>
<td>2.524420e-01</td>
<td>-2.383956e+04</td>
</tr>
<tr>
<td>2.524420e-01</td>
<td>2.383956e+04</td>
</tr>
</tbody>
</table>
Table 27-6: Pole/Zero Analysis Results for the Active Low-Pass Filter

<table>
<thead>
<tr>
<th>Pole/Zero Analysis Results</th>
<th>1.637164e+00</th>
<th>2.981593e+04</th>
<th>2.605627e-01</th>
<th>4.745353e+03</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.981593e+04</td>
<td></td>
<td>-2.605627e-01</td>
<td>-4.745353e+03</td>
<td></td>
</tr>
<tr>
<td>4.88484e+00</td>
<td>4.852376e+04</td>
<td>7.780265e-01</td>
<td>7.722796e+03</td>
<td></td>
</tr>
<tr>
<td>-4.852376e+04</td>
<td></td>
<td>-7.780265e-01</td>
<td>-7.722796e+03</td>
<td></td>
</tr>
<tr>
<td>-3.641366e+06</td>
<td>-3.642634e+06</td>
<td>-5.795413e+05</td>
<td>-5.797342e+05</td>
<td></td>
</tr>
<tr>
<td>-3.649508e+06</td>
<td>3.642634e+06</td>
<td>-5.795413e+05</td>
<td>5.797342e+05</td>
<td></td>
</tr>
<tr>
<td>-3.649508e+06</td>
<td>-3.649610e+06</td>
<td>-5.808372e+05</td>
<td>-5.808535e+05</td>
<td></td>
</tr>
<tr>
<td>-3.649508e+06</td>
<td>3.649610e+06</td>
<td>-5.808372e+05</td>
<td>5.808535e+05</td>
<td></td>
</tr>
<tr>
<td>-3.683700e+06</td>
<td>3.683412e+06</td>
<td>-5.862790e+05</td>
<td>5.862333e+05</td>
<td></td>
</tr>
<tr>
<td>-3.683700e+06</td>
<td>-3.683412e+06</td>
<td>-5.862790e+05</td>
<td>-5.862333e+05</td>
<td></td>
</tr>
<tr>
<td>-3.693882e+06</td>
<td>3.693739e+06</td>
<td>5.878995e+05</td>
<td>5.878768e+05</td>
<td></td>
</tr>
<tr>
<td>-3.693882e+06</td>
<td>-3.693739e+06</td>
<td>5.878995e+05</td>
<td>-5.878768e+05</td>
<td></td>
</tr>
</tbody>
</table>

Constant Factor = 4.451586e+02
The top graph in Table 27-10 plots the bandpass response of the Pole/Zero Example 6 low-pass filter. The bottom graph shows the overall response of the low-pass filter.

**References**

References for this chapter are listed below.


Using Pole/Zero Analysis

Performing Pole/Zero Analysis
Chapter 28
Performing FFT Spectrum Analysis

Spectrum analysis is the process of determining the frequency domain representation of a time domain signal and most commonly employs the Fourier transform. The Discrete Fourier Transform (DFT) is used to determine the frequency content of analog signals encountered in circuit simulation, which deals with sequences of time values. The Fast Fourier Transform (FFT) is an efficient method for calculating the DFT, and Star-Hspice uses it to provide a highly accurate spectrum analysis tool.

The .FFT statement in Star-Hspice uses the internal time point values and, by default, through a second order interpolation, obtains waveform samples based on the user-specified number of points.

Note: New accuracy improvement feature added in the Hspice 99.4 release. The .option fft_accurate or .option accurate (which internally turns on the fft_accurate option) will force Hspice to dynamically adjust the time step so that each FFT point will be a real simulation point. This eliminates the interpolation error and provides the highest FFT accuracy with minimal overhead in simulation time.

Windowing functions, can be used to reduce the effects of truncation of the waveform on the spectral content. The .FFT command also allows you to specify the desired output format, to specify a frequency of interest, and to obtain any number of harmonics, as well as the total harmonic distortion (THD).

This chapter covers the following topics:
- Using Windows In FFT Analysis
- Using the .FFT Statement
Performing FFT Spectrum Analysis

- Examining the FFT Output
- AM Modulation
- Balanced Modulator and Demodulator
Using Windows In FFT Analysis

One problem with spectrum analysis in circuit simulators is that the duration of the signals is finite, although adjustable. Applying the FFT method to finite-duration sequences can produce inadequate results because of “spectral leakage,” due primarily to the periodic extension assumption underlying DFT.

The effect occurs when the finite duration of the signal does not result in a sequence that contains a whole number of periods. This is especially true when FFT is used for signal detection or estimation – that is, for detecting weak signals in the presence of strong signals or resolving a cluster of equal strength frequencies.

In FFT analysis, “windows” are frequency weighting functions applied to the time domain data to reduce the spectral leakage associated with finite-duration time signals. Windows are smoothing functions that peak in the middle frequencies and decrease to zero at the edges, thus reducing the effects of the discontinuities as a result of finite duration. Table 28-1 shows the windows available in Star-Hspice. Table 28-1 lists the common performance parameters for FFT windows available in Star-Hspice.

Figure 28-1: FFT Windows
Table 28-1: Window Weighting Characteristics in FFT Analysis

<table>
<thead>
<tr>
<th>Window</th>
<th>Equation</th>
<th>Highest Side-Lobe (dB)</th>
<th>Side-Lobe Roll-Off (dB/octave)</th>
<th>3.0-dB Bandwidth (1.0/T)</th>
<th>Worst Case Process Loss (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>$W(n)=1, \ 0 \leq n &lt; NP\dagger$</td>
<td>-13</td>
<td>-6</td>
<td>0.89</td>
<td>3.92</td>
</tr>
<tr>
<td>Bartlett</td>
<td>$W(n)=2n/(NP-1), \ 0 \leq n \leq (NP/2)-1$</td>
<td>-27</td>
<td>-12</td>
<td>1.28</td>
<td>3.07</td>
</tr>
<tr>
<td></td>
<td>$W(n)=2-2n/(NP-1), \ NP/2 \leq n &lt; NP$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hanning</td>
<td>$W(n)=0.5-0.5\cos(2\pi n/(NP-1)), \ 0 \leq n &lt; NP$</td>
<td>-32</td>
<td>-18</td>
<td>1.44</td>
<td>3.18</td>
</tr>
<tr>
<td>Hamming</td>
<td>$W(n)=0.54-0.46\cos(2\pi n/(NP-1)), \ 0 \leq n &lt; NP$</td>
<td>-43</td>
<td>-6</td>
<td>1.30</td>
<td>3.10</td>
</tr>
<tr>
<td>Blackman</td>
<td>$W(n)=0.42323-0.49755\cos(2\pi n/(NP-1)) +0.07922\cos(4\pi n/(NP-1)),$</td>
<td>-58</td>
<td>-18</td>
<td>1.68</td>
<td>3.47</td>
</tr>
<tr>
<td></td>
<td>$\ 0 \leq n &lt; NP$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blackman-Harris</td>
<td>$W(n)=0.35875-0.48829\cos(2\pi n/(NP-1)) +0.14128\cos(4\pi n/(NP-1)),$</td>
<td>-92</td>
<td>-6</td>
<td>1.90</td>
<td>3.85</td>
</tr>
<tr>
<td></td>
<td>$+0.01168\cos(6\pi n/(NP-1)),$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\ 0 \leq n &lt; NP$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td>$W(n)=\exp[-0.5a2(NP/2-1-n)^2/(NP)2], \ 0 \leq n \leq (NP/2)-1$</td>
<td>-42</td>
<td>-6</td>
<td>1.33</td>
<td>3.14</td>
</tr>
<tr>
<td></td>
<td>$W(n)=\exp[-0.5a2(n-NP/2)^2/(NP)2], \ NP/2 \leq n &lt; NP$</td>
<td>-55</td>
<td>-6</td>
<td>1.55</td>
<td>3.40</td>
</tr>
<tr>
<td></td>
<td>$a=2.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$a=3.0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$a=3.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kaiser-Bessel</td>
<td>$W(n)=I_0(x_2)/I_0(x_1)\ x_1=pa$</td>
<td>-46</td>
<td>-6</td>
<td>1.43</td>
<td>3.20</td>
</tr>
<tr>
<td></td>
<td>$x_2=x_1^2\sqrt{1-(2(NP/2-1-n)/NP)2}, \ 0 \leq n \leq (NP/2)-1$</td>
<td>-57</td>
<td>-6</td>
<td>1.57</td>
<td>3.38</td>
</tr>
<tr>
<td></td>
<td>$x_2=x_1^2\sqrt{1-(2(n-NP/2)/NP)2}, \ NP/2 \leq n &lt; NP$</td>
<td>-69</td>
<td>-6</td>
<td>1.71</td>
<td>3.56</td>
</tr>
<tr>
<td></td>
<td>$a=2.0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$a=2.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$a=3.0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$a=3.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*I0 is the zero-order modified Bessel function*
NP is the number of points used for the FFT analysis.

The most important parameters in Table 28-1 are the highest side-lobe level (to reduce bias, the lower the better) and the worst-case processing loss (to increase detectability, the lower the better). Some compromise usually is necessary to find a suitable window filtering for each application. As a rule, the window performance improves with functions of higher complexity (those listed lower in the table). The Kaiser window has an ALFA parameter that allows adjustment of the compromise between different figures of merit for the window.

The simple rectangular window produces a simple bandpass truncation in the classical Gibbs phenomenon. The Bartlett or triangular window has good processing loss and good side-lobe roll-off, but lacks sufficient bias reduction. The Hanning, Hamming, Blackman, and Blackman-Harris windows use progressively more complicated cosine functions that provide smooth truncation and a wide range of side-lobe level and processing loss. The last two windows in the table are parameterized windows that allow you to adjust the side-lobe level, the 3 dB bandwidth, and the processing loss.\(^1\)

The characteristics of two typical windows are shown in Figures 28-2 and 28-3.

---

**Figure 28-2: Bartlett Window Characteristics**

\(^1\)NP is the number of points used for the FFT analysis.
Figure 28-3: Kaiser-Bessel Window Characteristics, ALFA=3.0
Using the .FFT Statement

The general form of the .FFT statement is shown below. The parameters are described in Table 28-2.

Syntax

```
.FFT <output_var> <START=value> <STOP=value> <NP=value>
+ <FORMAT=keyword> <WINDOW=keyword> <ALFA=value> <FREQ=value>
+ <FMIN=value> <FMAX=value>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>output_var</td>
<td>Can be any valid output variable, such as voltage, current, or power</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>see Description</td>
<td>Specifies the beginning of the output variable waveform to be analyzed – Defaults to the START value in the .TRAN statement, which defaults to 0 s.</td>
</tr>
<tr>
<td>FROM</td>
<td>see START</td>
<td>An alias for START in .FFT statements</td>
</tr>
<tr>
<td>STOP</td>
<td>see Description</td>
<td>Specifies the end of the output variable waveform to be analyzed. Defaults to the TSTOP value in the .TRAN statement.</td>
</tr>
<tr>
<td>TO</td>
<td>see STOP</td>
<td>An alias for STOP in .FFT statements</td>
</tr>
<tr>
<td>NP</td>
<td>1024</td>
<td>Specifies the number of points used in the FFT analysis. NP must be a power of 2; if NP is not a power of 2, Star-Hspice automatically adjusts it to the closest higher number that is a power of 2.</td>
</tr>
<tr>
<td>FORMAT</td>
<td>NORM</td>
<td>Specifies the output format: NORM= normalized magnitude UNORM= unnormalized magnitude</td>
</tr>
</tbody>
</table>
Example

Below are four examples of valid .FFT statements.

```
.fft v(1)
.fft v(1,2) np=1024 start=0.3m stop=0.5m freq=5.0k window=kaiser
   alfa=2.5
.fft I(rload) start=0m to=2.0m fmin=100k fmax=120k
   format=unorm
.fft par('v(1) + v(2)') from=0.2u stop=1.2u window=harris
```
Only one output variable is allowed in an .FFT command. The following is an incorrect use of the command.

```
.fft v(1) v(2) np=1024
```

The correct use of the command is shown in the example below. In this case, an .ft0 and an .ft1 file are generated for the FFT of v(1) and v(2), respectively.

```
.fft v(1) np=1024
.fft v(2) np=1024
```
Examining the FFT Output

Star-Hspice prints the results of the FFT analysis in a tabular format in the .lis file, based on the parameters in the .FFT statement. The normalized magnitude values are printed unless you specify FORMAT= UNORM, in which case unnormalized magnitude values are printed. The number of printed frequencies is half the number of points (NP) specified in the .FFT statement.

If you specify a minimum or a maximum frequency using FMIN or FMAX, the printed information is limited to the specified frequency range. Moreover, if you specify a frequency of interest using FREQ, then the output is limited to the harmonics of this frequency, along with the percent of total harmonic distortion.

In the sample output below, notice that all the parameters used in the FFT analysis are defined in the header.

****** Sample FFT output extracted from the .lis file
fft test ... sine
****** fft analysis tnom= 25.000 temp=
25.000
******
fft components of transient response v(1)

Window: Rectangular
First Harmonic: 1.0000k
Start Freq: 1.0000k
Stop Freq: 10.0000k
dc component: mag(db)= -1.132D+02 mag= 2.191D-06 phase= 1.800D+02

<table>
<thead>
<tr>
<th>frequency index</th>
<th>frequency (hz)</th>
<th>fft_mag (db)</th>
<th>fft_mag (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.0000k</td>
<td>0.0000</td>
<td>-3.8093m</td>
</tr>
<tr>
<td>4</td>
<td>2.0000k</td>
<td>-125.5914</td>
<td>525.3264n</td>
</tr>
<tr>
<td>6</td>
<td>3.0000k</td>
<td>-106.3740</td>
<td>4.8007u</td>
</tr>
<tr>
<td>8</td>
<td>4.0000k</td>
<td>-113.5753</td>
<td>2.0952u</td>
</tr>
<tr>
<td>10</td>
<td>5.0000k</td>
<td>-112.6689</td>
<td>2.3257u</td>
</tr>
<tr>
<td>12</td>
<td>6.0000k</td>
<td>-118.3365</td>
<td>1.2111u</td>
</tr>
<tr>
<td>14</td>
<td>7.0000k</td>
<td>-109.8888</td>
<td>3.2030u</td>
</tr>
<tr>
<td>16</td>
<td>8.0000k</td>
<td>-117.4413</td>
<td>1.3426u</td>
</tr>
</tbody>
</table>
Performing FFT Spectrum Analysis

Examining the FFT Output

<table>
<thead>
<tr>
<th>Frequency Index</th>
<th>Frequency</th>
<th>fft_mag (dB)</th>
<th>fft_mag</th>
<th>fft_phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>9.0000k</td>
<td>-97.5293</td>
<td>13.2903u</td>
<td>70.0515</td>
</tr>
<tr>
<td>20</td>
<td>10.0000k</td>
<td>-114.3693</td>
<td>1.9122u</td>
<td>-12.5492</td>
</tr>
</tbody>
</table>

Total harmonic distortion = 1.5065m percent

The preceding example specifies a frequency of 1 kHz and THD up to 10 kHz, which corresponds to the first ten harmonics.

**Note:** The highest frequency shown in the Star-Hspice FFT output might not be exactly the same as the specified FMAX, due to adjustments made by Star-Hspice.

Table 28-3 describes the output of the Star-Hspice FFT analysis.

**Table 28-3: .FFT Output Description**

<table>
<thead>
<tr>
<th>Column Heading</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency Index</td>
<td>Runs from 1 to NP/2, or the corresponding index for FMIN and FMAX. Note that the DC component corresponding to the index 0 is displayed independently.</td>
</tr>
<tr>
<td>Frequency</td>
<td>The actual frequency associated with the index</td>
</tr>
<tr>
<td>fft_mag (dB), fft_mag</td>
<td>There are two FFT magnitude columns, the first in dB and the second in the units of the output variable. The magnitude is normalized unless UNORM format is specified.</td>
</tr>
<tr>
<td>fft_phase</td>
<td>The associated phase, in degrees</td>
</tr>
</tbody>
</table>

A .ft# file is generated, in addition to the listing file, for each FFT output variable. The .ft# file contains the graphical data needed to display the FFT analysis results in AvanWaves. The magnitude in dB and the phase in degrees are available for display.

**Notes:**

1. The following formula should be used as a guideline when specifying a frequency range for FFT output:

   \[
   \text{frequency increment} = \frac{1.0}{(\text{STOP} - \text{START})}
   \]
Each frequency index corresponds to a multiple of this increment. Hence, to obtain a finer frequency resolution you should maximize the duration of the time window.

2. FMIN and FMAX have no effect on the .fit0, .fit1, ..., .fitn files.
AM Modulation

This example input listing on the following page shows a 1 kHz carrier (FC) that is modulated by a 100 Hz signal (FM). The voltage at node 1, which is an AM signal, can be described by

\[ v(1) = s_a \cdot (\text{offset} + \sin(\omega_m (\text{Time} - td)))) \cdot \sin(\omega_c (\text{Time} - td)) \]

The preceding equation can be expanded as follows.

\[ v(1) = (s_a \cdot \text{offset} \cdot \sin(\omega_c (\text{Time} - td))) + 0.5 \cdot s_a \cdot \cos((\omega_c - \omega_m)(\text{Time} - td))) \]

\[ - 0.5 \cdot s_a \cdot \cos((\omega_c + \omega_m)(\text{Time} - td)) \]

where

\[ \omega_c = 2\pi f_c \]

\[ \omega_f = 2\pi f_m \]

The preceding equations indicate that \( v(1) \) is a summation of three signals with frequency \( f_c \), \( (f_c - f_m) \), and \( (f_c + f_m) \) — namely, the carrier frequency and the two sidebands.

Input Listing

AM Modulation

.OPTION post
.PARAM sa=10 offset=1 fm=100 fc=1k td=1m
VX 1 0 AM(sa offset fm fc td)
Rx 1 0
.TRAN 0.01m 52m
.FFT V(1) START=10m STOP=40m FMIN=833 FMAX=1.16K
.END
Output Listing

The relevant portion of the listing file is shown below.

*********
am modulation
***** fft analysis
25.000
*****
fft components of transient response v(1)
Window: Rectangular
Start Freq: 833.3333
Stop Freq: 1.1667k
dc component: mag(db) = -1.480D+02 mag = 3.964D-08 phase = 0.000D+00

<table>
<thead>
<tr>
<th>frequency index</th>
<th>frequency (hz)</th>
<th>fft_mag (db)</th>
<th>fft_mag (db)</th>
<th>fft_phase (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>833.3333</td>
<td>-129.4536</td>
<td>336.7584n</td>
<td>-113.0047</td>
</tr>
<tr>
<td>26</td>
<td>866.6667</td>
<td>-143.7912</td>
<td>64.6308n</td>
<td>45.6195</td>
</tr>
<tr>
<td>27</td>
<td>900.0000</td>
<td>-6.0206</td>
<td>500.0008m</td>
<td>35.9963</td>
</tr>
<tr>
<td>28</td>
<td>933.3333</td>
<td>-125.4909</td>
<td>531.4428n</td>
<td>112.6012</td>
</tr>
<tr>
<td>29</td>
<td>966.6667</td>
<td>-125.4909</td>
<td>531.4428n</td>
<td>112.6012</td>
</tr>
<tr>
<td>30</td>
<td>1.0000k</td>
<td>0.</td>
<td>1.0000</td>
<td>-90.0050</td>
</tr>
<tr>
<td>31</td>
<td>1.0333k</td>
<td>-132.4062</td>
<td>239.7125n</td>
<td>-9.0718</td>
</tr>
<tr>
<td>32</td>
<td>1.0667k</td>
<td>-152.0156</td>
<td>25.0738n</td>
<td>3.4251</td>
</tr>
<tr>
<td>33</td>
<td>1.1000k</td>
<td>-6.0206</td>
<td>499.9989m</td>
<td>143.9933</td>
</tr>
<tr>
<td>34</td>
<td>1.1333k</td>
<td>-147.0134</td>
<td>44.5997n</td>
<td>-3.0046</td>
</tr>
<tr>
<td>35</td>
<td>1.1667k</td>
<td>-147.7864</td>
<td>40.8021n</td>
<td>-4.7543</td>
</tr>
</tbody>
</table>

***** job concluded

Graphical Output

Figures 28-4 and 28-5 display the results. Figure 28-4 shows the time domain curve of node 1. Figure 28-5 shows the frequency domain components of the magnitude of node 1. Note the carrier frequency at 1 kHz, with two sideband frequencies 100 Hz apart. The third, fifth, and seventh harmonics are more than 100 dB below the fundamental, indicating excellent numerical accuracy. Since the time domain data contains an integer multiple of the period, no windowing is needed.
Performing FFT Spectrum Analysis

**AM Modulation**

**Figure 28-4: AM Modulation**

![AM Modulation Graph]

**Figure 28-5: AM Modulation Spectrum**

![AM Modulation Spectrum Graph]
Balanced Modulator and Demodulator

Demodulation, or detection, is the process of recovering a modulating signal from the modulated output voltage. The netlist below illustrates this process, using Star-Hspice behavioral models and FFT analysis to confirm the validity of the process in the frequency domain. The Laplace element is used in the low-pass filter. This filter introduces some delay in the output signal, which causes spectral leakage if no windowing is used in FFT. However, when window weighting is used to perform FFT, the spectral leakage is virtually eliminated. This can be verified from the THD of the two outputs shown in the output listing that follows. Since a 1 kHz output signal is expected, a frequency of 1 kHz is specified in the .FFT command. Additionally, specifying the desired FMAX provides the first few harmonics in the output listing for THD calculations.

Input Listing

Balanced Modulator & Demodulator Circuit

V1 mod1 GND sin(0 5 1K 0 0 0) $ modulating signal
r1 mod1 2 10k
r2 2 3 10k
r3 2 GND 10K
E1 3 GND OPAMP 2 GND $ buffered output of modulating signal
V2 mod2 GND sin(0 5 10K 0 0 0) $ modulated signal
E2 modout GND vol='(v(3)*v(mod2))/10.0' $ multiply to modulate
V3 8 GND sin(0 5 10K 0 0 0)
E3 demod GND vol='(v(modout)*v(8))/10.0' $ multiply to demodulate
* use a laplace element for filtering
E_filter lpout 0 laplace demod 0 67.11e6 / 66.64e6 6.258e3 1.0 $ filter out +the modulating signal
*
.tran 0.2u 4m
.fft v(mod1)
.fft v(mod2)
.fft v(modout)
.fft v(demod)
Performing FFT Spectrum Analysis

```
.fft v(lpout) freq=1.0k fmax=10k $ ask to see the first few harmonics
.fft v(lpout) window=harris freq=1.0k fmax=10k $ window should + reduce spectral leakage
.probe tran v(mod1) V(mod2) v(modout) v(demod) v(lpout)
.option acct post probe
.end
```

**Output Listing**

The relevant portion of the output listing is shown below to illustrate the effect of windowing in reducing spectral leakage and consequently, reducing the THD.

```
balanced modulator & demodulator circuit
  ***** fft analysis tnom= 25.000 temp= 25.000
  *****

fft components of transient response v(lpout)
Window: Rectangular
First Harmonic: 1.0000k
Start Freq: 1.0000k
Stop Freq: 10.0000k
dc component: mag(db) = 3.738D+01 mag = 1.353D-02 phase = 1.800D+02

<table>
<thead>
<tr>
<th>frequency index</th>
<th>frequency (hz)</th>
<th>fft_mag (db)</th>
<th>fft_mag (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.0000k</td>
<td>0.00</td>
<td>1.0000</td>
</tr>
<tr>
<td>8</td>
<td>2.0000k</td>
<td>-26.6737</td>
<td>46.3781m</td>
</tr>
<tr>
<td>12</td>
<td>3.0000k</td>
<td>-31.4745</td>
<td>26.6856m</td>
</tr>
<tr>
<td>16</td>
<td>4.0000k</td>
<td>-34.4833</td>
<td>18.8728m</td>
</tr>
<tr>
<td>20</td>
<td>5.0000k</td>
<td>-36.6608</td>
<td>14.6880m</td>
</tr>
<tr>
<td>24</td>
<td>6.0000k</td>
<td>-38.3737</td>
<td>12.0591m</td>
</tr>
<tr>
<td>28</td>
<td>7.0000k</td>
<td>-39.7894</td>
<td>10.2455m</td>
</tr>
<tr>
<td>32</td>
<td>8.0000k</td>
<td>-40.9976</td>
<td>8.9150m</td>
</tr>
<tr>
<td>36</td>
<td>9.0000k</td>
<td>-42.0524</td>
<td>7.8955m</td>
</tr>
<tr>
<td>40</td>
<td>10.0000k</td>
<td>-42.9888</td>
<td>7.0886m</td>
</tr>
</tbody>
</table>

  total harmonic distortion = 6.2269 percent
  *****
```

balanced modulator & demodulator circuit
Balanced Modulator and Demodulator

Performing FFT Spectrum Analysis

***** fft analysis
25.000
*****
fft components of transient response v(lpout)

Window: Blackman-Harris
First Harmonic: 1.0000k
Start Freq: 1.0000k
Stop Freq: 10.0000k
dc component: mag(db)= -8.809D+01 mag= 3.938D-05 phase= 1.800D+02

<table>
<thead>
<tr>
<th>frequency</th>
<th>fft_mag</th>
<th>fft_mag</th>
<th>fft_phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>(hz)</td>
<td>(db)</td>
<td>(deg)</td>
</tr>
<tr>
<td>4</td>
<td>1.0000k</td>
<td>0.</td>
<td>1.0000</td>
</tr>
<tr>
<td>8</td>
<td>2.0000k</td>
<td>-66.5109</td>
<td>472.5569u</td>
</tr>
<tr>
<td>12</td>
<td>3.0000k</td>
<td>-97.5914</td>
<td>13.1956u</td>
</tr>
<tr>
<td>16</td>
<td>4.0000k</td>
<td>-107.8004</td>
<td>4.0736u</td>
</tr>
<tr>
<td>20</td>
<td>5.0000k</td>
<td>-117.9984</td>
<td>1.2592u</td>
</tr>
<tr>
<td>24</td>
<td>6.0000k</td>
<td>-125.0965</td>
<td>556.1309n</td>
</tr>
<tr>
<td>28</td>
<td>7.0000k</td>
<td>-123.6795</td>
<td>654.6722n</td>
</tr>
<tr>
<td>32</td>
<td>8.0000k</td>
<td>-122.4362</td>
<td>755.4258n</td>
</tr>
<tr>
<td>36</td>
<td>9.0000k</td>
<td>-122.0336</td>
<td>791.2570n</td>
</tr>
<tr>
<td>40</td>
<td>10.0000k</td>
<td>-122.0388</td>
<td>790.7840n</td>
</tr>
</tbody>
</table>

Total harmonic distortion = 47.2763m percent

The signals and their spectral content are shown in Figures 28-6 through 28-14. The modulated signal contains only the sum and the difference of the carrier frequency and the modulating signal (1 kHz and 10 kHz). At the receiver end the carrier frequency is recovered in the demodulated signal, which also shows a 10 kHz frequency shift in the above signals (to 19 kHz and 21 kHz).

A low-pass filter is used to extract the carrier frequency using a second order Butterworth filter. Use of a Harris window significantly improves the noise floor in the filtered output spectrum and reduces THD in the output listing (from 9.23% to 0.047%). However, it appears that a filter with a steeper transition region and better delay characteristics is needed to suppress the modulating frequencies below the -60 dB level. The “Filtered Output Signal” waveform in Figure 28-9 is normalized.
Figure 28-6: Modulating and Modulated Signals

Figure 28-7: Modulated Signal
Balanced Modulator and Demodulator Performing FFT Spectrum Analysis

Figure 28-8: Demodulated Signal

Figure 28-9: Filtered Output Signal
Figure 28-10: Modulating and Modulated Signal Spectrum

Figure 28-11: Modulated Signal Spectrum
Figure 28-12: Demodulated Signal Spectrum

Figure 28-13: Filtered Output Signal (no window)
Figure 28-14: Filtered Output Signal (Blackman-Harris window)
Signal Detection Test Circuit

This example is a high frequency mixer test circuit, illustrating the effect of using a window to detect a weak signal in the presence of a strong signal at a nearby frequency. Two high frequency signals are added that have a 40 dB separation (that is, amplitudes are 1.0 and 0.01).

Input Listing

```
Signal Detection Test Circuit For FFT
v1 1 0 sin(0 1 1470.2Meg 0 0 90)  
r1 1 0 1
v2 2 0 sin(0 0.01 1560.25Meg 0 0 90)  
r2 2 0 1
E1 3 0 vol='v(1)+v(2)'
   r3 3 0 1
.tran 0.1n 102.4n  
.option post probe  
.fft v(3)
   .fft v(3) window=Bartlett fmin=1.2g fmax=2.2g
   .fft v(3) window=hanning fmin=1.2g fmax=2.2g
   .fft v(3) window=hamminn fmin=1.2g fmax=2.2g
   .fft v(3) window=blackman fmin=1.2g fmax=2.2g
   .fft v(3) window=harris fmin=1.2g fmax=2.2g
   .fft v(3) window=gaussian fmin=1.2g fmax=2.2g
   .fft v(3) window=kaiser fmin=1.2g fmax=2.2g
 .end
```

For comparison with the rectangular window in Figure 28-15, the spectra of the output for all of the FFT window types are shown in Figures 28-16 through 28-22. Without windowing, the weak signal is essentially undetectable due to spectral leakage.
Performing FFT Spectrum Analysis

Signal Detection Test Circuit

Figure 28-15: Mixer Output Spectrum, Rectangular Window

In the Bartlett window in Figure 28-16, notice the dramatic decrease in the noise floor over the rectangular window (from -55 to more than -90 dB). The cosine windows (Hanning, Hamming, Blackman, and Blackman-Harris) all produce better results than the Bartlett window. However, the degree of separation of the two tones and the noise floor is best with the Blackman-Harris window. The final two windows (Figures 28-21 and 28-22) are parameterized with ALFA=3.0, which is the default value in Star-Hspice. These two windows also produce acceptable results, especially the Kaiser-Bessel window, which gives sharp separation of the two tones and almost a -100-dB noise floor.

Such processing of high frequencies, as demonstrated in this example, shows the numerical stability and accuracy of the FFT spectrum analysis algorithms in Star-Hspice.
Figure 28-16: Mixer Output Spectrum, Bartlett Window

Figure 28-17: Mixer Output Spectrum, Hanning Window
Figure 28-18: Mixer Output Spectrum, Hamming Window

Figure 28-19: Mixer Output Spectrum, Blackman Window
Figure 28-20: Mixer Output Spectrum, Blackman-Harris Window

Figure 28-21: Mixer Output Spectrum, Gaussian Window
Performing FFT Spectrum Analysis Signal Detection Test Circuit

Figure 28-22: Mixer Output Spectrum, Kaiser-Bessel Window

References

Applying Kirchhoff’s laws to circuits containing energy storage elements results in simultaneous differential equations in the time domain that must be solved to analyze the circuit’s behavior. The solution of any equation of higher than first order can be difficult, and some driving functions cannot be solved easily by classical methods.

In both cases, the solution might be simplified using Laplace transforms to convert time domain equations containing integral and differential terms to algebraic equations in the frequency domain.

This chapter covers the following topics:

- Understanding Transient Modeling
- Using G and E Elements
- Laplace and Pole-Zero Modeling
- Modeling Switched Capacitor Filters
Understanding Transient Modeling

The Laplace transform method also provides an easy way of relating a circuit’s behavior in time and frequency-domains, facilitating simultaneous work in those domains.

The performance of the algorithm Star-Hspice uses for Laplace and pole/zero transient modeling is better than the performance of the Fast Fourier Transform (FFT) algorithm. Laplace and pole/zero transient modeling is invoked by using a LAPLACE or POLE function call in a source element statement.

Laplace transfer functions are especially useful in top-down system design, using ideal transfer functions instead of detailed circuit designs. Star-Hspice also allows you to mix Laplace transfer functions with transistors and passive components. Using this capability, a system may be modeled as the sum of the contributing ideal transfer functions, which can be progressively replaced by detailed circuit models as they become available. Laplace transfer functions are also conveniently used in control systems and behavioral models containing nonlinear elements.

Using Laplace transforms can reduce the long simulation times (as well as design time) of large interconnect systems, such as clock distribution networks, for which you can use methods such as asymptotic waveform evaluation (AWE) to create a Laplace transfer function model. The AWE model can represent the large circuit with just a few poles. You can input these poles through a Laplace transform model to closely approximate the delay and overshoot characteristics of many networks in a fraction of the original simulation time.

Pole/zero analysis is important in determining the stability of the design. The POLE function in Star-Hspice is useful when the poles and zeros of the circuit are provided, or they can be derived from the transfer function. (You can use the Star-Hspice .PZ statement to find poles and zeros. See “.PZ (Pole/Zero) Statement” on page 27-3 for information about the .PZ statement).
Frequency response, an important analog circuit property, is normally specified as a ratio of two complex polynomials (functions of complex frequencies) with positive real coefficients. Frequency response can be given in the form of the locations of poles and zeros or can be in the form of a frequency table.

Complex circuits are usually designed by interconnecting smaller functional blocks of known frequency response, either in pole/zero or frequency table form. For example, you can design a band-reject filter by interconnecting a low-pass filter, a high-pass filter, and an adder. The designer should study the function of the complex circuit in terms of its component blocks before designing the actual circuit. After testing the functionality of the component blocks, they can be used as a reference in using optimization techniques to determine the complex element’s value.
Using G and E Elements

This section describes how to use the G and E Elements.

Laplace Transform Function Call

Use the Star-Hspice G and E Elements (controlled behavioral sources) as linear functional blocks or elements with specific frequency responses in the following forms:

- Laplace Transform
- Pole-Zero Function
- Frequency Response Table

The frequency response is called the impulse response and is denoted by $H(s)$, where $s$ is a complex frequency variable ($s = j2\pi f$). In Star-Hspice, the frequency response is obtained by performing an AC analysis with AC=1 in the input source (the Laplace transform of an impulse is 1). The input and output of the G and E Elements with specified frequency response are related by the expression:

$$Y(j2\pi f) = Hj(2\pi f) \cdot Xj(2\pi f)$$

where $X$, $Y$ and $H$ are the input, the output, and the transfer function at frequency $f$.

For AC analysis, the frequency response is determined by the above relation at any frequency. For operating point and DC sweep analysis, the relation is the same, but the frequency is zero.

The transient analysis is more complicated than the frequency response. The output is a convolution of the input waveform with the impulse response $h(t)$:

$$y(t) = \int_{-\infty}^{\infty} x(\tau) \cdot h(t - \tau) \cdot d\tau$$
In discrete form, the output is

\[ y(k\Delta) = \Delta \sum_{m=0}^{k} x(m\Delta) \cdot h[(k-m)\cdot\Delta], \quad k = 0, 1, 2, ... \]

where the \( h(t) \) can be obtained from \( H(f) \) by the inverse Fourier integral:

\[ h(t) = \int_{-\infty}^{\infty} H(f) \cdot e^{j2\pi ft} \cdot df \]

The inverse discrete Fourier transform is given by

\[ h(m\Delta) = \frac{1}{N \cdot \Delta} \sum_{n=0}^{N-1} H(f_n) \cdot e^{j2\pi nm} \cdot e^{-j2\pi nm/N}, \quad m = 0, 1, 2, ..., N-1 \]

where \( N \) is the number of equally spaced time points and \( \Delta \) is the time interval or time resolution.

For the frequency response table form (FREQ) of the LAPLACE function, Star-Hspice’s performance-enhanced algorithm is used to convert \( H(f) \) to \( h(t) \). This algorithm requires \( N \) to be a power of 2. The frequency point \( f_n \) is determined by

\[ f_n = \frac{n}{N \cdot \Delta}, \quad n = 0, 1, 2, ..., N-1 \]

where \( n > N/2 \) represents the negative frequencies. The Nyquist critical frequency is given by

\[ f_c = f_{N/2} = \frac{1}{2 \cdot \Delta} \]

Since the negative frequencies responses are the image of the positive ones, only \( N/2 \) frequency points are required to evaluate \( N \) time points of \( h(t) \). The larger \( f_c \) is, the more accurate the transient analysis results are. However, for large \( f_c \), the
becomes smaller, and computation time increases. The maximum frequency of interest depends on the functionality of the linear network. For example, in a low-pass filter, $f_c$ can be set to the frequency at which the response drops by 60 dB (a factor of 1000).

$$|H(f_c)| = \frac{|H_{max}|}{1000}$$

Once $f_c$ is selected or calculated, then $\Delta$ can be determined by

$$\Delta = \frac{1}{2 \cdot f_c}$$

Notice the frequency resolution

$$\Delta f = f_1 = \frac{1}{N \cdot \Delta}$$

is inversely proportional to the maximum time ($N \cdot \Delta$) over which $h(t)$ is evaluated. Therefore, the transient analysis accuracy also depends on the frequency resolution or the number of points ($N$). You can specify the frequency resolution DELF and maximum frequency MAXF in the G or E Element statement. $N$ is calculated by $2 \cdot \text{MAXF}/\text{DELF}$. Then, $N$ is modified to be a power of 2. The effective DELF is determined by $2 \cdot \text{MAXF}/N$ to reflect the changes in $N$.

**Laplace Transform**

The syntax of the \texttt{LAPLACE} function is:

**Transconductance** $H(s)$:

\begin{verbatim}
Gxxx n_+ n_- LAPLACE in_+ in_- k_0, k_1, ..., k_n / d_0, + d_1, ..., d_m <SCALE=val> <TC1=val> <TC2=val> <M=val>
\end{verbatim}

**Voltage Gain** $H(s)$:

\begin{verbatim}
Exxx n_+ n_- LAPLACE in_+ in_- k_0, k_1, ..., k_n / d_0,
\end{verbatim}
H(s) is a rational function in the following form:

$$H(s) = \frac{k_0 + k_1 s + \ldots + k_n s^n}{d_0 + d_1 s + \ldots + d_m s^m}$$

All the coefficients $k_0, k_1, \ldots, d_0, d_1, \ldots$ can be parameterized.

### Example

**Glowpass**

```plaintext
Glowpass 0 out LAPLACE in 0 1.0 / 1.0 2.0 2.0 1.0
```

The Glowpass element statement describes a third-order low-pass filter with the transfer function

$$H(s) = \frac{1}{1 + 2s + 2s^2 + s^3}$$

**Ehipass**

```plaintext
Ehipass out 0 LAPLACE in 0 0.0,0.0,0.0,1.0 / 1.0,2.0,2.0,1.0
```

The Ehipass element statement describes a third-order high-pass filter with the transfer function

$$H(s) = \frac{s^3}{1 + 2s + 2s^2 + s^3}$$

### Pole-Zero Function

The syntax is:

**Transconductance H(s):**

```
Gxxx n+ n- POLE in+ in- a $\alpha_{z1}$, $f_{z1}$, \ldots, $\alpha_{zn}$, $f_{zn}$ / b,
+ $\alpha_{p1}$, $f_{p1}$, \ldots, $\alpha_{pm}$, $f_{pm}$ <SCALE=val> <TC1=val>
+ <TC2=val> <M=val>
```

**Voltage Gain H(s):**

```
```

---

*Star-Hspice Manual, Release 2001.2*
H(s) in terms of poles and zeros is defined by:

\[ H(s) = \frac{a \cdot (s + \alpha_{z1} - j2\pi f_{z1}) \cdots (s + \alpha_{zn} - j2\pi f_{zn})(s + \alpha_{zn} + j2\pi f_{zn})}{b \cdot (s + \alpha_{p1} - j2\pi f_{p1}) \cdots (s + \alpha_{pm} - j2\pi f_{pm})(s + \alpha_{pm} + j2\pi f_{pm})} \]

Notice the complex poles or zeros are in conjugate pairs. In the element description, only one of them is specified, and the program includes the conjugate. The a, b, \( \alpha \), and f values can be parameterized.

**Example**

Ghigh_pass 0 out POLE in 0 1.0 0.0,0.0 / 1.0 0.001,0.0
Elow_pass out 0 POLE in 0 1.0 / 1.0,0.0 0.5,0.1379

The Ghigh_pass statement describes a high-pass filter with transfer function

\[ H(s) = \frac{1.0 \cdot (s + 0.0 + j \cdot 0.0)}{1.0 \cdot (s + 0.001 + j \cdot 0.0)} \]

The Elow_pass statement describes a low-pass filter with transfer function

\[ H(s) = \frac{1.0}{1.0 \cdot (s + 1)(s + 0.5 + j2\pi \cdot 0.1379)(s + 0.5 - (j2\pi \cdot 0.1379))} \]

**Frequency Response Table**

The syntax is:

Transconductance H(s):

\[ \text{Gxxx n+ n- FREQ in+ in- f1, a1, \phi1, \ldots, f1, a1, \phi1} \]
+ \(<\text{DELF}=\text{val}> <\text{MAXF}=\text{val}> <\text{SCALE}=\text{val}> <\text{TC1}=\text{val}> \]
+ \(<\text{TC2}=\text{val}> <\text{M}=\text{val}> <\text{LEVEL}=\text{val}> \]
+ \(<\text{INTERPOLATION}=\text{val}> <\text{EXTRAPOLATION}=\text{val}> \]
+ \(<\text{ACCURACY}=\text{val}> \]
Voltage Gain $H(s)$:

```
Exxx n+ n- FREQ in+ in-  f_i,  a_i,  \phi_i, \ldots, f_i,  a_i,  \phi_i
+ <DELF=val>  <MAXF=val>  <SCALE=val>  <TC1=val>
+ <TC2=val>
```

Each $f_i$ is a frequency point in hertz, $a_i$ is the magnitude in dB, and $\phi_i$ is the phase in degrees. At each frequency the network response, magnitude, and phase are calculated by interpolation. The magnitude (in dB) is interpolated logarithmically as a function of frequency. The phase (in degrees) is interpolated linearly as a function of frequency.

\[
|H(j2\pi f)| = \left( \frac{a_i - a_k}{\log f_i - \log f_k} \right) (\log f - \log f_i) + a_i
\]

\[
\angle H(j2\pi f) = \left( \frac{\phi_i - \phi_k}{f_i - f_k} \right) (f - f_i) + \phi_i
\]

Example

```
Eftable output 0  FREQ  input 0
+ 1.0k  -3.97m  293.7
+ 2.0k  -2.00m  211.0
+ 3.0k  17.80m  82.45
+ ........
+ 10.0k  -53.20  -1125.5
```

The first column is frequency in hertz, the second is magnitude in dB, and third is phase in degrees. The LEVEL must be set to 1 for a high-pass filter, and the last frequency point must be the highest frequency response value that is a real number with zero phase. The frequency, magnitude, and phase in the table can be parameterized.

**Element Statement Parameters**

These keywords are common to the three forms, Laplace, pole-zero, and frequency response table described above.
**ACCURACY**

Used only in G element with frequency response table.

0: default. The most accurate control voltage at each time point is achieved by linear interpolation of closest 2 time points.

1: faster mode. The control voltage at each time point is determined from the simulated time point just before the target. The smaller the time step is set, the closer the result will be to the most accurate mode.

2: method used in release 2000.4 and prior.

**DELF, DELTA**

Frequency resolution $\Delta f$. The inverse of DELF is the time window over which $h(t)$ is calculated from $H(s)$. The smaller DELF is, the more accurate the transient analysis, and the longer the CPU time. The number of points, $N$, used in the conversion of $H(s)$ to $h(t)$ is $N = 2 \cdot \text{MAXF}/\text{DELF}$. Since $N$ must be a power of 2, the DELF is adjusted. The default is $1/T_{STOP}$. In G element with FREQ and ACCURACY = 0 or 1, circular convolution for periodic input will be done by limiting the period by setting DELF = $1/T : T < T_{STOP}$.

**EXTRAPOLATION**

Extrapolation scheme used only in G element with frequency response table.

0: default. Linear interpolation using the last two boundary points.

1: the last boundary point is used.

**FREQ**

Keyword to indicate that the transfer function is described by a frequency response table. Do not use FREQ as a node name in a G or E Element.

**INTERPOLATION**

Interpolation scheme used only in G element with frequency response table.

0: default. piece wise linear
1: piece wise step
2: b-spline curve fit

**LAPLACE**
Keyword to indicate the transfer function is described by a Laplace transform function. Do not use LAPLACE as a node name on a G or E Element.

**LEVEL**
Used only in elements with frequency response table. This parameter must be set to 1 if the element represents a high-pass filter.

**M**
G Element multiplier. This parameter is used to represent G G Elements in parallel. Default is 1.

**MAXF, MAX**
Maximum or the Nyquist critical frequency. The larger the MAXF, the more accurate the transient results and the longer is the CPU time. The default is $1024 \cdot DELF$. These parameters are applicable only when the FREQ parameter is also used.

**POLE**
Keyword to indicate the transfer function is described by the pole and zero location. Do not use POLE as a node name on a G or E Element.

**SCALE**
Element value multiplier

**TC1,TC2**
First- and second-order temperature coefficients. The default is zero. The SCALE is updated by temperature:

$$ SCALE_{eff} = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta) $$

**Notes**
- Pole/zero analysis is not allowed when the data file contains elements with frequency response specifications. If you include a MAXF=<$\text{par}$> specification in a G or Element Statement, Star-Hspice issues a warning that MAXF is ignored. This is normal.
- Interpolation, Extrapolation, and Accuracy options are only for G (with FREQ) element
The interpolation and extrapolation parameters noted above are available if ACCURACY = 0 or 1. If ACCURACY = 2, then interpolation/extrapolation is performed as mentioned earlier under “Frequency Response Table” on page 29-8.

Circular convolution is performed when G element ACCURACY = 0 or 1 (see Figure 29-7: )

Laplace Band-Reject Filter

This example models an active band-reject filter with 3-dB points at 100 and 400 Hz and greater than 35 dB of attenuation between 175 and 225 Hz. The band-reject filter is made up of low-pass and high-pass filters and an adder. The low-pass and high-pass filters are fifth order Chebyshev with a 0.5-dB ripple.

Figure 29-1: Band-Reject Filter

Example

BandstopL.sp band_reject filter
.OPTIONS PROBE POST=2
.AC DEC 50 10 5k
.PROBE AC VM(out_low) VM(out_high) VM(out)
.PROBE AC VP(out_low) VP(out_high) VP(out)
.TRAN .01m 12m
.PROBE V(out_low) V(out_high) V(out)
.GRAPH v(in) V(out)
Vin in 0 AC 1 SIN(0,1,250)
**Band_Reject Filter Circuit**

```
Elp3 out_low3 0 LAPLACE in 0
+ 1 / 1 6.729m 15.62988u 27.7976n
Elp out_low 0 LAPLACE out_low3 0
+ 1 / 1 0.364m 2.7482u
Ehp3 out_high3 0 LAPLACE in 0
+ 0,0,0,9.261282467p /
+ 1,356.608u,98.33419352n,9.261282467p
Eph out_high 0 LAPLACE out_high3 0
+ 0 0 144.03675n / 1 83.58u 144.03675n
Eadd out 0 VOL='V(out_low) - V(out_high)'
Rl out 0 1e6
.END
```

**Figure 29-2: Frequency Response of the Band-Reject Filter**
Figure 29-3: Transient Response of the Band-Reject Filter to a 250 Hz Sine Wave

Laplace Low-Pass Filter

This example simulates a third-order low-pass filter with a Butterworth transfer function, comparing the results of the actual circuit and the functional G Element with third-order Butterworth transfer function for AC and transient analysis.

Figure 29-4: Third-Order Active Low-Pass Filter
The third-order Butterworth transfer function that describes the above circuit is:

\[ H(s) = \frac{1.0}{1.0 \cdot (s + 1)(s + 0.5 + j2\pi \cdot 0.1379)(s + 0.5 - (j2\pi \cdot 0.1379))} \]

The following is the input listing of the above filter. Notice the pole locations are parameterized on the G Element. Also, only one of the complex poles is specified. The conjugate pole is derived by the program. The output of the circuit is node “out” and the output of the functional element is “outg”.

**Example**

This is an example of a third-order low-pass Butterworth filter:

```plaintext
Low_Pass.sp 3rd order low-pass Butterworth
.OPTIONS POST=2 PROBE INTERP=1 DCSTEP=1e8
.PARAM a=1.0 b=1.0 ap1=1.0 fp1=0.0 ap2=0.5 fp2=0.1379
.AC DEC 25 0.01 10
.PROBE AC VDB(out) VDB(outg) VP(out) VP(outg)
.TRAN .5 200
.PROBE V(in) V(outg) V(out)
.GRAPH V(outg) V(out)
VIN in 0 AC 1 PULSE(0,1,0,1,1,48,100)
* 3rd order low-pass described by G Element
Glow_pass 0 outg POLE in 0 a / b ap1,fp1 ap2,fp2
Rg outg 0 1
```

**Circuit Description**

```plaintext
R1 in 2 1
R2 2 3 1
R3 3 4 1
C1 2 0 1.392
C2 4 0 0.2024
C3 3 out 3.546
Eopamp out 0 OPAMP 4 out
.END
```
Figure 29-5: Frequency Response of Circuit and Functional Element
Circular Convolution Example

This example simulates a 30 degree phase shift filter using circular convolution. By setting DELF = 10MHz, the period of time domain response of the G element which will be obtained by IFFT based on input frequency table will be set to 100n seconds. FREQ G element performs convolution integral from t - T to t assuming that all the control voltage at t<0 is zero. Here, t is the target time point and T is the period of time domain response of the G element.

In this example, during time point is from 0 to 100n seconds, higher than 10MHz harmonics components due to input transition at t=0 is taken. So the circuit does not behave as a phase shift filter. After one period (t>100n seconds), circular convolution based on 100n seconds period will be performed and the transient result represents 30 degree phase shift for continuous periodic control voltage.
Notes

- V(ctrl): control voltage input
- V(expected): node which represents ideal 30 degree shifted wave of input
- V(test): output of the G element

30 Degree Phase Shift Circuit File

This example illustrates a 30 degree phase shift filter.

```
.TRAN 0.1n 300n
.OPTIONS POST=2 INGOLD=2 ACCURATE

Vctrl ctrl gnd sin (0 1 10e6)

Gtest gnd test freq ctrl gnd
  + 1.0e00 0 30
  + 1.0e01 0 30
  + 1.0e02 0 30
  + 1.0e03 0 30
  + 1.0e04 0 30
  + 1.0e05 0 30
  + 1.0e06 0 30
  + 1.0e07 0 30
  + 1.0e08 0 30
  + 1.0e09 0 30
  + 1.0e10 0 30
  + MAXF=1.0e9 DELF=10e6

Rtest test gnd 1

Iexpected gnd 3 sin (0 1 10e6 0 0 30)
Vmex 2 3 expected 0v
Rexpected expected gnd 1
.END
```
Figure 29-7: Transient Response of the 30 Degree Phase Shift Filter
Laplace and Pole-Zero Modeling

Laplace Transform (LAPLACE) Function

There are two forms of the Star-Hspice LAPLACE function call, one for transconductance and one for voltage gain transfer functions. See “Using G and E Elements” on page 29-4 for the general forms and “Element Statement Parameters” on page 29-9 for descriptions of the parameters.

General Form of the Transfer Function

To use the Star-Hspice LAPLACE modeling function, you must find the $k_0, \ldots, k_n$ and $d_0, \ldots, d_m$ coefficients of the transfer function. The transfer function is the s-domain (frequency domain) ratio of the output of a single-source circuit to the input, with initial conditions set to zero. The Laplace transfer function is represented by:

$$H(s) = \frac{Y(s)}{X(s)},$$

where $s$ is the complex frequency $j2\pi f$, $Y(s)$ is the Laplace transform of the output signal, and $X(s)$ is the Laplace transform of the input signal.

**Note:** In Star-Hspice, the impulse response $H(s)$ is obtained by performing an AC analysis, with AC=1 representing the input source. The Laplace transform of an impulse is 1. For an element with an infinite response at DC, such as a unit step function $H(s)=1/s$, Star-Hspice uses the value of the EPSMIN option (the smallest number possible on the platform) for the transfer function in its calculations.

The general form of the transfer function $H(s)$ in the frequency domain is:

$$H(s) = \frac{k_0 + k_1 s + \ldots + k_n s^n}{d_0 + d_1 s + \ldots + d_m s^m}.$$
The order of the numerator of the transfer function cannot be greater than the order of the denominator, except for differentiators, for which the transfer function $H(s) = ks$. All of the transfer function’s $k$ and $d$ coefficients can be parameterized in the Star-Hspice circuit descriptions.

**Finding the Transfer Function**

The first step in determining the transfer function of a circuit is to convert the circuit to the $s$-domain by transforming each element’s value into its $s$-domain equivalent form.

Tables 29-1 and 29-2 show transforms used to convert some common functions to the $s$-domain\(^{2,3}\). The next section provides examples of using transforms to determine transfer functions.

**Table 29-1: Laplace Transforms for Common Source Functions**

<table>
<thead>
<tr>
<th>$f(t)$, $t&gt;0$</th>
<th>Source Type</th>
<th>$\mathcal{L} { f(t) } = F(s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta(t)$</td>
<td>impulse</td>
<td>1</td>
</tr>
<tr>
<td>$u(t)$</td>
<td>step</td>
<td>$\frac{1}{s}$</td>
</tr>
<tr>
<td>$t$</td>
<td>ramp</td>
<td>$\frac{1}{s^2}$</td>
</tr>
<tr>
<td>$e^{-at}$</td>
<td>exponential</td>
<td>$\frac{1}{s + a}$</td>
</tr>
<tr>
<td>$\sin \omega t$</td>
<td>sine</td>
<td>$\frac{\omega}{s^2 + \omega^2}$</td>
</tr>
<tr>
<td>$\cos \omega t$</td>
<td>cosine</td>
<td>$\frac{s}{s^2 + \omega^2}$</td>
</tr>
</tbody>
</table>
Table 29-1: Laplace Transforms for Common Source Functions

<table>
<thead>
<tr>
<th>f(t), t&gt;0</th>
<th>Source Type</th>
<th>( \mathcal{L} { f(t) } = F(s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sin(\omega t + \theta) )</td>
<td>sine</td>
<td>( \frac{s \sin(\theta) + \omega \cos(\theta)}{s^2 + \omega^2} )</td>
</tr>
<tr>
<td>( \cos(\omega t + \theta) )</td>
<td>cosine</td>
<td>( \frac{s \cos(\theta) - \omega \sin(\theta)}{s^2 + \omega^2} )</td>
</tr>
<tr>
<td>( \sinh \omega t )</td>
<td>hyperbolic sine</td>
<td>( \frac{\omega}{s^2 - \omega^2} )</td>
</tr>
<tr>
<td>( \cosh \omega t )</td>
<td>hyperbolic cosine</td>
<td>( \frac{s}{s^2 - \omega^2} )</td>
</tr>
<tr>
<td>( te^{-\alpha t} )</td>
<td>damped ramp</td>
<td>( \frac{1}{(s + \alpha)^2} )</td>
</tr>
<tr>
<td>( e^{\alpha t} \sin \omega t )</td>
<td>damped sine</td>
<td>( \frac{\omega}{(s + \alpha)^2 + \omega^2} )</td>
</tr>
<tr>
<td>( e^{\alpha t} \cos \omega t )</td>
<td>damped cosine</td>
<td>( \frac{s + \alpha}{(s + \alpha)^2 + \omega^2} )</td>
</tr>
</tbody>
</table>

Table 29-2: Laplace Transforms for Common Operations

<table>
<thead>
<tr>
<th>f(t)</th>
<th>( \mathcal{L} { f(t) } = F(s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Kf(t) )</td>
<td>( KF(s) )</td>
</tr>
<tr>
<td>( f_1(t) + f_2(t) - f_3(t) + \ldots )</td>
<td>( F_1(s) + F_2(s) - F_3(s) + \ldots )</td>
</tr>
<tr>
<td>( \frac{d}{dt} f(t) )</td>
<td>( sF(s) - f(0-) )</td>
</tr>
</tbody>
</table>
Determining the Laplace Coefficients

The following examples describe how to determine the appropriate coefficients for the Laplace modeling function call in Star-Hspice.
LAPLACE Example 1 – Voltage Gain Transfer Function

To find the voltage gain transfer function for the circuit in Figure 29-8, convert the circuit to its equivalent s-domain circuit and solve for \( v_o / v_g \).

**Figure 29-8: LAPLACE Example 1 Circuit**

Use transforms from Table 29-2 to convert the inductor, capacitor, and resistors. \( \mathcal{L}\{f(t)\} \) represents the Laplace transform of \( f(t) \):

\[
\mathcal{L}\left\{ \frac{d}{dt}f(t) \right\} = L \cdot (sF(s) - f(0)) = 50 \times 10^{-3} \cdot (s - 0) = 0.05s
\]

\[
\mathcal{L}\left\{ \int_0^t f(t)dt \right\} = \frac{1}{C} \cdot \left( \frac{F(s)}{s} + \frac{f^{-1}(0)}{s} \right) = \frac{1}{10^6} \cdot \left( \frac{1}{s} + 0 \right) = \frac{10^6}{s}
\]

\[
\mathcal{L}\{R_1 \cdot f(t)\} = R_1 \cdot F(s) = R_1 = 1000 \ \Omega
\]

\[
\mathcal{L}\{R_2 \cdot f(t)\} = R_2 \cdot F(s) = R_2 = 250 \ \Omega
\]

To convert the voltage source to the s-domain, use the \( \sin \omega t \) transform from Table 29-1:

\[
\mathcal{L}\{2\sin 3t\} = 2 \cdot \frac{3}{s^2 + 9} = \frac{6}{s^2 + 9}
\]
Figure 29-9 displays the $s$-domain equivalent circuit.

**Figure 29-9: S-Domain Equivalent of the LAPLACE Example 1 Circuit**

Summing the currents leaving node n2:

$$\frac{v_o - v_g}{1000} + \frac{v_o}{250 + 0.05s} + \frac{v_o s}{10^6} = 0$$

Solve for $v_o$:

$$v_o = \frac{1000(s + 5000)v_g}{s^2 + 6000s + 25 \times 10^6}$$

The voltage gain transfer function is:

$$H(s) = \frac{v_o}{v_g} = \frac{1000(s + 5000)}{s^2 + 6000s + 25 \times 10^6} = \frac{5 \times 10^6 + 1000s}{25 \times 10^6 + 6000s + s^2}$$

For the Star-Hspice Laplace function call, use $k_n$ and $d_m$ coefficients for the transfer function in the form:
The coefficients from the voltage gain transfer function above are:

\[ k_0 = 5 \times 10^6 \quad k_1 = 1000 \]
\[ d_0 = 25 \times 10^6 \quad d_1 = 6000 \quad d_2 = 1 \]

Using these coefficients, a Star-Hspice Laplace modeling function call for the voltage gain transfer function of the circuit in Figure 29-8 is:

```
Example1 n1 n0 LAPLACE n2 n0 5E6 1000 / 25E6 6000 1
```

**LAPLACE Example 2 – Differentiator**

You can model a differentiator using either G or E Elements as shown in the following example.

In the frequency domain:

- **E Element**: \( V_{\text{out}} = k s V_{\text{in}} \)
- **G Element**: \( I_{\text{out}} = k s V_{\text{in}} \)

In the time domain:

- **E Element**: \( v_{\text{out}} = k \frac{dV_{\text{in}}}{dt} \)
- **G Element**: \( i_{\text{out}} = k \frac{dV_{\text{in}}}{dt} \)
For a differentiator, the voltage gain transfer function is:

\[ H(s) = \frac{V_{out}}{V_{in}} = ks \]

In the general form of the transfer function,

\[ H(s) = \frac{k_0 + k_1 s + \ldots + k_n s^n}{d_0 + d_1 s + \ldots + d_m s^m} \]

If you set \( k_1 = k \) and \( d_0 = 1 \) and the remaining coefficients are zero, then the equation becomes:

\[ H(s) = \frac{ks}{1} = ks \]

Using the coefficients \( k_1 = k \) and \( d_0 = 1 \) in the Laplace modeling, the Star-Hspice circuit descriptions for the differentiator are:

Edif out GND LAPLACE in GND 0 k / 1
Gdif out GND LAPLACE in GND 0 k / 1

**LAPLACE Example 3 – Integrator**

An integrator can be modeled by G or E Elements as follows:

In the frequency domain:

**E Element:** \( V_{out} = \frac{k}{s} V_{in} \)

**G Element:** \( I_{out} = \frac{k}{s} V_{in} \)
In the time domain:

**E Element:** $v_{out} = k \int V_{in} dt$

**G Element:** $i_{out} = k \int V_{in} dt$

For an integrator, the voltage gain transfer function is:

$$H(s) = \frac{V_{out}}{V_{in}} = \frac{k}{s}$$

In the general form of the transfer function:

$$H(s) = \frac{k_0 + k_1 s + ... + k_n s^n}{d_0 + d_1 s + ... + d_m s^m}$$

Like the previous example, if you make $k_0 = k$ and $d_1 = 1$, then the equation becomes:

$$H(s) = \frac{k + 0 + ... + 0}{0 + s + ... + 0} = \frac{k}{s}$$

**Laplace Transform POLE (Pole/Zero) Function**

This section describes the general form of the pole/zero transfer function and provides examples of converting specific transfer functions into pole/zero circuit descriptions.

**POLE Function Call**

The POLE function in Star-Hspice is useful when the poles and zeros of the circuit are available. The poles and zeros can be derived from the transfer function, as described in this chapter, or you can use the Star-Hspice .PZ statement to find them, as described in “.PZ (Pole/Zero) Statement” on page 27-3.
There are two forms of the Star-Hspice LAPLACE function call, one for transconductance and one for voltage gain transfer functions. See “Using G and E Elements” for the general forms and list of optional parameters.

To use the POLE pole/zero modeling function, find the $a$, $b$, $f$, and $\alpha$ coefficients of the transfer function. The transfer function is the $s$-domain (frequency domain) ratio of the output of a single-source circuit to the input, with initial conditions set to zero.

**General Form of the Transfer Function**

The general expanded form of the pole/zero transfer function $H(s)$ is:

$$H(s) = \frac{a(s + \alpha_{z1} + j2\pi f_{z1})(s + \alpha_{z1} - j2\pi f_{z1})...(s + \alpha_{zn} + j2\pi f_{zn})(s + \alpha_{zn} - j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})(s + \alpha_{p1} - j2\pi f_{p1})...(s + \alpha_{pm} + j2\pi f_{pm})(s + \alpha_{pm} - j2\pi f_{pm})}$$

The $a$, $b$, $\alpha$, and $f$ values can be parameterized.

**Example**

Ghigh_pass 0 out POLE in 0 1.0 0.0,0.0 / 1.0
0.001,0.0
Elow_pass 0 out POLE in 0 1.0 / 1.0, 1.0,0.0
0.5,0.1379

The Ghigh_pass statement describes a high pass filter with transfer function:

$$H(s) = \frac{1.0 \cdot (s + 0.0 + j \cdot 0.0)}{1.0 \cdot (s + 0.001 + j \cdot 0.0)}$$

The Elow_pass statement describes a low-pass filter with transfer function:

$$H(s) = \frac{1.0}{1.0 \cdot (s + 1)(s + 0.5 + j2\pi \cdot 0.1379)(s + 0.5 - (j2\pi \cdot 0.1379))}$$
To write a Star-Hspice pole/zero circuit description for an element, you need to know the element’s transfer function $H(s)$ in terms of the $a$, $b$, $f$, and $\alpha$ coefficients. Use the values of these coefficients in POLE function calls in the Star-Hspice circuit description.

First, however, simplify the transfer function, as described in the next section.

**Star-Hspice Reduced Form of the Transfer Function**

Complex poles and zeros occur in conjugate pairs (a set of complex numbers differ only in the signs of their imaginary parts):

$$ (s + \alpha_{pm} + j2\pi f_{pm})(s + \alpha_{pm} - j2\pi f_{pm}) \text{, for poles} $$

and

$$ (s + \alpha_{zn} + j2\pi f_{zn})(s + \alpha_{zn} - j2\pi f_{zn}) \text{, for zeros} $$

To write the transfer function in Star-Hspice pole/zero format, supply coefficients for one term of each conjugate pair and Star-Hspice provides the coefficients for the other term. If you omit the negative complex roots, the result is the reduced form of the transfer function, $\text{Reduced}(H(s))$. Find the reduced form by collecting all the general form terms with negative complex roots:

$$ H(s) = \frac{a(s + \alpha_{z1} + j2\pi f_{z1})\ldots(s + \alpha_{zn} + j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})\ldots(s + \alpha_{pm} + j2\pi f_{pm})} \cdot \frac{a(s + \alpha_{z1} - j2\pi f_{z1})\ldots(s + \alpha_{zn} - j2\pi f_{zn})}{b(s + \alpha_{p1} - j2\pi f_{p1})\ldots(s + \alpha_{pm} - j2\pi f_{pm})} $$

(1)

Then discard the right-hand term, which contains all the terms with negative roots. What remains is the reduced form:

$$ \text{Reduced}(H(s)) = \frac{a(s + \alpha_{z1} + j2\pi f_{z1})\ldots(s + \alpha_{zn} + j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})\ldots(s + \alpha_{pm} + j2\pi f_{pm})} \text{(2)} $$

For this function find the $a$, $b$, $f$, and $\alpha$ coefficients to use in a Star-Hspice POLE function for a voltage gain transfer function. The following examples show how to determine the coefficients and write POLE function calls for a high-pass filter and a low-pass filter.
POLE Example 1 – Highpass Filter

For a high-pass filter with a given transconductance transfer function, such as:

\[ H(s) = \frac{s}{(s + 0.001)} \]

Find the \( a \), \( b \), \( \alpha \), and \( f \) coefficients necessary to write the transfer function in the general form (1) shown previously, so that you can clearly see the conjugate pairs of complex roots. You only need to supply one of each conjugate pair of roots in the Laplace function call. Star-Hspice automatically inserts the other root.

To get the function into a form more similar to the general form of the transfer function, rewrite the given transconductance transfer function as:

\[ H(s) = \frac{1.0(s + 0.0)}{1.0(s + 0.001)} \]

Since this function has no negative imaginary parts, it is already in the Star-Hspice reduced form (2) shown previously. Now you can identify the \( a \), \( b \), \( f \), and \( \alpha \) coefficients so that the transfer function \( H(s) \) matches the reduced form. This matching process obtains the following values:

\[ n = 1, m = 1, \]
\[ a = 1.0, \quad \alpha_{z1} = 0.0, \quad f_{z1} = 0.0 \]
\[ b = 1.0, \quad \alpha_{p1} = 0.001, \quad f_{p1} = 0.0 \]

Using these coefficients in the reduced form provides the desired transfer function, \( \frac{s}{(s + 0.001)} \).

So the general transconductance transfer function POLE function call,

\[ \text{Gxxx n+ n- POLE in+ in- a } \alpha_{z1}, f_{z1} \ldots \alpha_{zn}, f_{zn} / b \alpha_{p1}, f_{p1} \ldots \alpha_{pm}, f_{pm} \]

for an element named \textit{Ghigh_pass} becomes:

\[ \text{Ghigh_pass gnd out POLE in gnd 1.0 0.0,0.0 / 1.0 0.001,0.0} \]
POLE Example 2 – Low-Pass Filter

For a low-pass filter with the given voltage gain transfer function:

\[ H(s) = \frac{1.0}{1.0(s + 1.0 + j2\pi \cdot 0.0)(s + 0.5 + j2\pi \cdot 0.15)(s + 0.5 - j2\pi \cdot 0.15)} \]

you need to find the \( a, b, \alpha, \) and \( f \) coefficients to write the transfer function in the general form, so that you can identify the complex roots with negative imaginary parts.

To separate the reduced form, \( Reduced\{H(s)\} \), from the terms with negative imaginary parts, rewrite the given voltage gain transfer function as:

\[ H(s) = \frac{1.0}{1.0(s + 1.0 + j2\pi \cdot 0.0)(s + 0.5 + j2\pi \cdot 0.15)} \cdot \frac{1.0}{(s + 0.5 - j2\pi \cdot 0.15)} = Reduced\{H(s)\} \cdot \frac{1.0}{(s + 0.5 - j2\pi \cdot 0.15)} \]

So:

\[ Reduced\{H(s)\} = \frac{1.0}{1.0(s + 1.0)(s + 0.5 + j2\pi \cdot 0.15)} \]

or:

\[ \frac{a(s + \alpha_{z1} + j2\pi f_{z1}) \cdots (s + \alpha_{zn} + j2\pi f_{zn})}{(s + \alpha_{p1} + j2\pi f_{p1}) \cdots (s + \alpha_{pm} + j2\pi f_{pm})} = \frac{1.0}{1.0(s + 1.0 + j2\pi \cdot 0.0)(s + 0.5 + j2\pi \cdot 0.15)} \]

Now assign coefficients in the reduced form to match the given voltage transfer function. The following coefficient values produce the desired transfer function:

\[ n = 0, m = 2, \]
\[ a = 1.0 \quad b = 1.0 \quad \alpha_{p1} = 1.0 \quad f_{p1} = 0 \quad \alpha_{p2} = 0.5 \quad f_{p2} = 0.15 \]
These coefficients can be substituted in the POLE function call for a voltage gain transfer function:

\[
\text{Exxx } n+ n- \text{ POLE in+ in- } a \alpha_{z1}, f_{z1} \ldots \alpha_{zn}, f_{zn} / b \\
\alpha_{p1}, f_{p1} \ldots \alpha_{pm}, f_{pm}
\]

for an element named \textit{Elow\_pass} to obtain the Star-Hspice statement:

\[
\text{Elow\_pass out GND POLE in 1.0 / 1.0 1.0,0.0 0.5,0.15}
\]

**RC Line Modeling**

Most RC lines can have very simple models, with just a single dominant pole. The dominant pole can be found by AWE methods, computed based on the total series resistance and capacitance\(^4\), or determined by the Elmore delay\(^5\).

The Elmore delay uses the value \((d1-k1)\) as the time constant of a single-pole approximation to the complete \(H(s)\), where \(H(s)\) is the transfer function of the RC network to a given output. The inverse Laplace transform of \(h(t)\) is \(H(s)\):

\[
\tau_{DE} = \int_0^\infty t \cdot h(t) dt
\]

Actually, the Elmore delay is the first moment of the impulse response, and so corresponds to a first order AWE result.

**Figure 29-10: Circuits for an RC Line**

![Circuits for an RC Line](image)
**RC Line Circuit File**

* Laplace testing RC line
. Tran 0.02ns 3ns
. Options Post Accurate List Probe
v1 1 0 PWL 0ns 0 0.1ns 0 0.3ns 5 1.3ns 5 1.5ns 0
r1 1 2 200
c1 2 0 0.6pF
r2 2 3 80
c2 3 0 0.8pF
r3 3 4 160
c3 4 0 0.7pF
r4 4 5 200
c4 5 0 0.8pF
e1 6 0 LAPLACE 1 0 1 / 1 1.16n
.Probe v(1) v(5) v(6)
.Print v(1) v(5) v(6)
.End

The output of the RC circuit shown in Figure 29-10 can be closely approximated by a single pole response, as shown in Figure 29-11.
Notice in Figure 29-11 that the single pole approximation has less delay: 1 ns compared to 1.1 ns for the full RC line model at 2.5 volts. The single pole approximation also has a lower peak value than the RC line model. All other things being equal, a circuit with a shorter time constant results in less filtering and allows a higher maximum voltage value. The single-pole approximation produces a lower amplitude and less delay than the RC line because the single pole neglects the other three poles in the actual circuit. However, a single-pole approximation still gives very good results for many problems.

**AWE Transfer Function Modeling**

Single-pole transfer function approximations can cause larger errors for low-loss lines than for RC lines since lower resistance allows ringing. Because circuit
ringing creates complex pole pairs in the transfer function approximation, at least one complex pole pair is needed to represent low-loss line response. Figure 29-12 shows a typical low-loss line, along with the transfer function sources used to test the various approximations. The transfer functions were obtained by asymptotic waveform evaluation⁶.

Figure 29-12: Circuits for a Low-Loss Line

Low-Loss Line Circuit File

* Laplace testing LC line Pillage Apr 1990
.Tran 0.02ns 8ns
.Options Post Accurate List Probe
v1 1 0 PWL 0ns 0 0.1ns 0 0.2ns 5
r1 1 2 25
L1 2 3 10nH
C2 3 0 1pF
L2 3 4 10nH
C3 4 0 1pF
L3 4 5 100nH
C4 5 0 1pF
r4 5 0 400
e3 8 0 LAPLACE 1 0 0.94 / 1.0 0.6n
Figure 29-13 shows the transient response of the low-loss line, along with E Element Laplace models using one, two, and four poles. Note that the single-pole model shows none of the ringing of the higher order models. Also, all of the E models had to adjust the gain of their response for the finite load resistance, so the models are not independent of the load impedance. The 0.94 gain multiplier in the models takes care of the 25 ohm source and 400 ohm load voltage divider. All of the approximations give good delay estimations.

While the two-pole approximation gives reasonable agreement with the transient overshoot, the four-pole model gives almost perfect agreement. The actual circuit has six poles. Scaling was used to bring some of the very small numbers in the Laplace model above the 1e-28 limit of Star-Hspice. The SCALE parameter multiplies every parameter in the LAPLACE specification by the same value, in this case 1.0E-20.
A low-loss line allows reflections between the load and source, while the loss of an RC line usually isolates the source from the load. So you can either incorporate the load into the AWE transfer function approximation or create a Star-Hspice model that allows source/load interaction. If you allow source/load interaction, the AWE expansions do not have to be done each time you change load impedances, allowing you to handle nonlinear loads and remove the need for a gain multiplier, as in the circuit file shown. You can use four voltage controlled current sources, or G Elements, to create a Y parameter model for a transmission line. The Y parameter network allows the source/load interaction needed. The next example shows such a Y parameter model for a low-loss line.
Y Parameter Line Modeling

A model that is independent of load impedance is more complicated. You can still use AWE techniques, but you need a way for the load voltage and current must be able to interact with the source impedance. Given a transmission line of 100 ohms and 0.4 ns total delay, as shown in Figure 29-14, compare the response of the line using a Y parameter model and a single-pole model.

The voltage and current definitions for a Y parameter model are shown in Figure 29-15.
The general network in Figure 29-15 is described by the following equations, which can be translated into G Elements:

\[ I_1 = Y_{11} V_{in} + Y_{12} V_{out} \]

\[ I_2 = Y_{21} V_{in} + Y_{22} V_{out} \]

A schematic for a set of two-port Y parameters is shown in Figure 29-16. Note that the circuit is essentially composed of G Elements.

![Figure 29-16: Schematic for the Y Parameter Network](image)

The Laplace parameters for the Y parameter model are determined by a Pade expansion of the Y parameters of a transmission line, as shown in matrix form in the following equation.

\[
Y = \frac{1}{Z_o} \begin{bmatrix}
  \coth(p) & -\csch(p) \\
  -\csch(p) & \coth(p)
\end{bmatrix},
\]

where \( p \) is the product of the propagation constant and the line length.\(^7\)
A Pade approximation contains polynomials in both the numerator and the denominator. Since a Pade approximation can model both poles and zeros and since \( \text{coth} \) and \( \text{csch} \) functions also contain both poles and zeros, a Pade approximation gives a better low order model than a series approximation. A Pade expansion of \( \text{coth}(p) \) and \( \text{csch}(p) \), with second order numerator and third order denominator, is given below:

\[
\text{coth}(p) \rightarrow \frac{1 + \frac{2}{5} \cdot p^2}{p + \frac{1}{15} \cdot p^3}
\]

\[
\text{csch}(p) \rightarrow \frac{1 - \frac{1}{20} \cdot p^2}{p + \frac{7}{60} \cdot p^3}
\]

When you substitute \((s \cdot \text{length} \cdot \sqrt{LC})\) for \(p\), you get polynomial expressions for each \(G\) Element. When you substitute 400 nH for \(L\), 40 pF for \(C\), 0.1 meter for length, and 100 for \(Z_o\) \((Z_o = \sqrt{LC})\) in the matrix equation above, you get values you can use in a circuit file.

The circuit file shown below uses all of the above substitutions. The Pade approximations have different denominators for \(\text{csch}\) and \(\text{coth}\), but the circuit file contains identical denominators. Although the actual denominators for \(\text{csch}\) and \(\text{coth}\) are only slightly different, using them would cause oscillations in the Star-Hspice response. To avoid this problem, use the same denominator in the \(\text{coth}\) and \(\text{csch}\) functions in the example. The simulation results may vary, depending on which denominator is used as the common denominator, because the coefficient of the third order term is changed (but by less than a factor of 2).
LC Line Circuit File
* Laplace testing LC line Pade
.Tran 0.02ns 5ns
.Options Post Accurate List Probe
v1 1 0 PWL 0ns 0 0.1ns 0 0.2ns 5
r1 1 2 25
r3 1 3 25
u1 2 0 5 0 wire1 L=0.1
r4 5 0 400
r8 8 0 400
e1 6 0 LAPLACE 1 0 1 / 1 0.4n
Gy11 3 0 LAPLACE 3 0 320016 0.0 2.048e-14 / 0.0 0.0128 0.0
 2.389e-22
Gy12 3 0 LAPLACE 8 0 -320016 0.0 2.56e-15 / 0.0 0.0128 0.0
  2.389e-22
Gy21 8 0 LAPLACE 3 0 -320016 0.0 2.56e-15 / 0.0 0.0128 0.0
  2.389e-22
Gy22 8 0 LAPLACE 8 0 320016 0.0 2.048e-14 / 0.0 0.0128 0.0
  2.389e-22
.model wire1 U Level=3 PLEV=1 ELEV=3 LLEV=0 MAXL=20
  + ZK=100 DELAY=4.0n
 Probe v(1) v(5) v(6) v(8)
.Print v(1) v(5) v(6) v(8)
.End

Figure 29-17 compares the output of the Y parameter model with that of a full transmission line simulation and with that obtained for a single pole transfer function. In the latter case, the gain was not corrected for the load impedance, so the function produces an incorrect final voltage level. As expected, the Y parameter model gives the correct final voltage level. Although the Y parameter model gives a good approximation of the circuit delay, it contains too few poles to model the transient details fully. However, the Y parameter model does give excellent agreement with the overshoot and settling times.
Comparison of Circuit and Pole/Zero Models

This example simulates a ninth order low-pass filter circuit and compares the results with its equivalent pole/zero description using an E Element. The results are identical, but the pole/zero model runs about 40% faster. The total CPU times for the two methods are shown in “Simulation Time Summary”. For larger circuits, the computation time saving can be much higher.

The input listings for each model type are shown below. Figures 29-18 and 29-19 display the transient and frequency response comparisons resulting from the two modeling methods.
Circuit Model Input Listing

low_pass9a.sp 9th order low_pass filter.
* Reference: Jiri Vlach and Kishore Singhal, "Computer Methods
for
* Circuit Analysis and Design", Van Nostrand Reinhold Co.,
1983,
* pages 142, 494-496.
*
.PARAM freq=100 tstop='2.0/freq'
*.PZ v(out) vin
.AC dec 50 .1k 100k
.OPTIONS dcsstep=1e3 post probe unwrap
.PROBE ac vdb(out) vp(out)
.TRAN STEP='tstop/200' STOP=tstop
.PROBE v(out)
vin in GND sin(0,1,freq) ac 1
.SUBCKT fdnr 1 r1=2k c1=12n r4=4.5k
r1 1 2 r1
c1 2 3 c1
r2 3 4 3.3k
r3 4 5 3.3k
r4 5 6 r4
c2 6 0 10n
eop1 5 0 opamp 2 4
eop2 3 0 opamp 6 4
.ENDS
*
rs in 1 5.4779k
r12 1 2 4.44k
r23 2 3 3.2201k
r34 3 4 3.63678k
r45 4 out 1.2201k
c5 out 0 10n
x1 1 fdnr r1=2.0076k c1=12n r4=4.5898k
x2 2 fdnr r1=5.9999k c1=6.8n r4=4.25725k
x3 3 fdnr r1=5.88327k c1=4.7n r4=5.62599k
x4 4 fdnr r1=1.0301k c1=6.8n r4=5.808498k
.END
Pole/Zero Model Input Listing

ninth.sp 9th order low_pass filter.
.PARAM twopi=6.2831853072
.PARAM freq=100 tstop='2.0/freq'
.AC dec 50 .1k 100k
.OPTIONS dcs=1e3 post probe unwrap
.PROBE ac vdb(outp) vp(outp)
.TRAN STEP='tstop/200' STOP=tstop
.PROBE v(outp)
vin in GND sin(0,1,freq) ac 1
Epole outp GND POLE in GND 417.6153
+ 0. 3.8188k
+ 0. 4.0352k
+ 0. 4.7862k
+ 0. 7.8903k / 1.0
+ '73.0669*twopi' 3.5400k
+ '289.3438*twopi' 3.4362k
+ '755.0697*twopi' 3.0945k
+ '1.5793k*twopi' 2.1105k
+ '2.1418k*twopi' 0.
repole outp GND 1e12
.END

Simulation Time Summary

Circuit model simulation times:

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Time (s)</th>
<th># Points</th>
<th>. iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv.iter</td>
<td>0.23</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>op point</td>
<td>0.47</td>
<td>151</td>
<td>151</td>
</tr>
<tr>
<td>ac analysis</td>
<td>0.75</td>
<td>201</td>
<td>226</td>
</tr>
<tr>
<td>transient</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rev= 0</td>
<td>0.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>readin</td>
<td>0.13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>errchk</td>
<td>0.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>setup</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>output</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total CPU time 1.98 seconds

Pole/zero model simulation times:
analysis | time | # points | tot. iter
--- | --- | --- | ---
conv.iter | | | |

df | 0.12 | 1 | 3
ac analysis | 0.22 | 151 | 151
transient | 0.40 | 201 | 222 | 111

rev= 0
readin | 0.23
errchk | 0.13
setup | 0.02
output | 0.00

total cpu time 1.23 seconds

Figure 29-18: Transient Responses of the Circuit and Pole/Zero Models
Figure 29-19: AC Analysis Responses of the Circuit and Pole/Zero Models
Modeling Switched Capacitor Filters

Switched Capacitor Network

It is possible to model a resistor as a capacitor and switch combination. The value of the equivalent is proportional to the frequency of the switch divided by the capacitance.

Construct a filter from MOSFETs and capacitors where the filter characteristics are a function of the switching frequency of the MOSFETs.

In order to quickly determine the filter characteristics, use ideal switches (voltage controlled resistors) instead of MOSFETs. The resulting simulation speedup can be as great as 7 to 10 times faster than a circuit using MOSFETs.

The model constructs an RC network using a resistor and a capacitor along with a switched capacitor equivalent network. Node RCOUT is the resistor/capacitor output, and VCROUT is the switched capacitor output.

The switches GVCR1 and GVCR2, together with the capacitance C3, model the resistor. The resistor value is calculated as:

\[ Res = \frac{T_{\text{switch}}}{C3} \]

where Tswitch is the period of the pulses PHI1 and PHI2.
Example

*FILE:VCR1.SP   A SWITCHED CAPACITOR RC CIRCUIT
.OPTIONS acct NOMOD POST

.IC V(SW1)=0 V(RCOUT)=0 V(VCROUT)=0
.TRAN 5U 200U
.GRAPH RC=V(RCOUT) SWITCH=V(VCROUT) (0,5)

VCC  VCC  GND  5V
C  RCOUT  GND  1NF
R  VCC  RCOUT  25K

C6  VCROUT  GND  1NF
* equivalent circuit for 25k resistor  r=12.5us/.5nf
VA  PHI1  GND  PULSE 0 5 1US .5US .5US 3US 12.5US
VB  PHI2  GND  PULSE 0 5 7US .5US .5US 3US 12.5US
Switched Capacitor Filter Example

This example is a fifth order elliptic switched capacitor filter with passband 0-1 kHz, loss less than 0.05 dB. It is realized by cascading linear, high_Q biquad, and low_Q biquad sections. The G Element models the switches with a resistance of 1 ohm when the switch is closed and 100 Megohm when it is open. The E Element models op-amps as an ideal op-amp. The transient response of the filter is provided for 1 kHz and 2 kHz sinusoidal input signal.

Figure 29-21: Linear Section
Figure 29-22: High_Q Biquad Section

Figure 29-23: Low_Q Biquad Section

Star-Hspice Input File for Switched Capacitor Filter

SWCAP5.SP Fifth Order Elliptic Switched Capacitor Filter.
.OPTIONS POST PROBE
.GLOBAL phi1 phi2
.TRAN 2u 3.2m UIC
*.GRAPH v(phi1) v(phi2) V(in)
.PROBE V(out)
*.PLOT v(in) v(phi1) v(phi2) v(out
*Iin 0 in SIN(0,1ma,1.0khz)
Iin 0 in SIN(0,1v,2khz)
Vphi1 phi1 0 PULSE(0,2 00u,.5u,.5u,7u,20u)
Vphi2 phi2 0 PULSE(0, 2 10u, .5u, .5u, 7u, 20u)
Rsrc in 0 1k
Rload out 0 1k
Xsh in out1 sh
Xlin out1 out2 linear
Xhq out2 out3 hqbiq
Xlq out3 out lqbiq

Sample and Hold
.SUBCKT sh in out
Gs1 in 1 VCR PWL(1) phi1 0 0.5v, 100meg 1.0v, 1.0
Eop1 out 0 OPAMP 1 out
Ch 1 0 1.0pf
.ENDS

Linear Section
.SUBCKT linear in out
Gs1 in 1 VCR PWL(1) phi1 0 0.5v, 100meg 1.0v, 1.0
Gs2 1 0 VCR PWL(1) phi2 0 0.5v, 100meg 1.0v, 1.0
Cs 1 2 1.0pf
Gs3 2 0 VCR PWL(1) phi1 0 0.5v, 100meg 1.0v, 1.0
Gs4 2 3 VCR PWL(1) phi2 0 0.5v, 100meg 1.0v, 1.0
Eop1 out 0 OPAMP 0 3
Ce 3 0 9.6725pf
Gs5 out 4 VCR PWL(1) phi2 0 0.5v, 100meg 1.0v, 1.0
Gs6 4 0 VCR PWL(1) phi1 0 0.5v, 100meg 1.0v, 1.0
Cd 4 2 1.0pf
Csh 3 0 0.5pf
.ENDS

High_Q Biquad Section
.SUBCKT hqbiq in out
Gs1 in 1 VCR PWL(1) phi2 0 0.5v, 100meg 1.0v, 1.0
Gs2 1 0 VCR PWL(1) phi1 0 0.5v, 100meg 1.0v, 1.0
C1 1 2 0.5pf
Gs3 2 0 VCR PWL(1) phi1 0 0.5v, 100meg 1.0v, 1.0
Gs4 2 3 VCR PWL(1) phi2 0 0.5v, 100meg 1.0v, 1.0
Eop1 4 0 OPAMP 0 3
Ca 3 4 7.072pf
Gs5 4 5 VCR PWL(1) phi1 0 0.5v, 100meg 1.0v, 1.0
Low_Q Biquad Section

.SUBCKT lqbiq in out
Gs1 in 1 VCR PWL(1) phi2 0 0.5v,100meg 1.0v,1.0
Gs2 1 2 VCR PWL(1) phi2 0 0.5v,100meg 1.0v,1.0
C1 2 3 0.9963pf
Gs3 2 0 VCR PWL(1) phi1 0 0.5v,100meg 1.0v,1.0
Gs4 3 0 VCR PWL(1) phi1 0 0.5v,100meg 1.0v,1.0
Gs5 3 4 VCR PWL(1) phi2 0 0.5v,100meg 1.0v,1.0
Ca 4 5 8.833pf
Eop1 5 0 OPAMP 0 4
Gs6 5 6 VCR PWL(1) phi1 0 0.5v,100meg 1.0v,1.0
Gs7 6 0 VCR PWL(1) phi2 0 0.5v,100meg 1.0v,1.0
C3 6 7 1.0558pf
Gs8 7 8 VCR PWL(1) phi2 0 0.5v,100meg 1.0v,1.0
Gs9 7 0 VCR PWL(1) phi1 0 0.5v,100meg 1.0v,1.0
Eop2 9 0 OPAMP 0 8
Cb 8 9 3.8643pf
Gs10 9 10 VCR PWL(1) phi2 0 0.5v,100meg 1.0v,1.0
Gs11 10 0 VCR PWL(1) phi1 0 0.5v,100meg 1.0v,1.0
C4 10 7 0.5pf
C2 10 3 0.5pf
Cl1 8 1 3.15425pf
Gs12 9 out VCR PWL(1) phi2 0 0.5v,100meg 1.0v,1.0
.ENDS
.END
Figure 29-24: Response to 1-kHz Sinusoidal Input

Figure 29-25: Response to 2-kHz Sinusoidal Input
References

References for this chapter are listed below.


Chapter 30

Timing Analysis Using Bisection

To analyze circuit timing violations, a typical methodology is to generate a set of operational parameters that produce a failure in the required behavior of the circuit. Then when a circuit timing failure occurs, you can identify a timing constraint that can lead to a design guideline. You must be able to perform an iterative analysis to define the violation specification.

Typical types of timing constraint violations include:
- Data setup time before clock
- Data hold time after clock
- Minimum pulse width required to allow a signal to propagate to the output
- Maximum toggle frequency of the component(s)

This chapter describes how to use the Star-Hspice bisection function in timing optimization. The general topic of optimization with Star-Hspice is covered in depth in “Statistical Analysis and Optimization” on page 13-1.

The following topics are covered in this chapter:
- Understanding Bisection
- Understanding the Bisection Methodology
- Using Bisection
- Setup Time Analysis
- Minimum Pulse Width Analysis
Understanding Bisection

Formerly, engineers built external drivers to submit multiple parameterized Star-Hspice jobs, with each job exploring a region of the operating envelope of the circuit. In addition, the driver needed to provide part of the analysis by post-processing the Star-Hspice results to deduce the limiting conditions.

Because characterization of circuits in this way is associated with small jobs, the individual analysis times are relatively small compared with the overall job time. This methodology is inefficient because of the overhead of submitting the job, reading and checking the netlist, and setting up the matrix. Efficiency in analyzing timing violations can be increased with more intelligent methods of determining the conditions causing timing failure. The bisection optimization method was developed to make cell characterization in Star-Hspice more efficient.

Star-Hspice bisection methodology saves time in three ways:

- Reduction of multiple jobs to a single characterization job
- Removal of post-processing requirements
- Use of accuracy-driven iteration

Figure 30-1 illustrates a typical analysis of setup time constraints. A cell is driven by clock and data input waveforms. There are two input transitions, rise and fall, that occur at times $T_1$ and $T_2$. The result is an output transition, when $V(\text{out})$ goes from low to high. The following relationship between times $T_1(\text{data})$ and $T_2(\text{clock})$ must be true in order for the $V(\text{out})$ transition to occur:

$T_2 > (T_1 + \text{setup time})$

The goal of the characterization, or violation analysis, is to determine the setup time. This is done by keeping $T_2$ fixed while repeating the simulation with different values of $T_1$ and observing which $T_1$ values produce the output transition and which values do not.

Previously, it was necessary to do very tight sweeps of the delay between the data setup and clock edge, looking for the value at which the transition fails to occur. This was done by sweeping a value that specifies how far the data edge precedes a fixed clock edge. This methodology is time consuming, and is not
accurate unless the sweep step is very small. The setup time value cannot be
determined accurately by linear search methods unless extremely small steps
from T1 to T2 are used to simulate the circuit at each point while monitoring the
outcome.

For example, even if it is known that the desired transition occurs during a
particular five nanosecond period, searching for the actual setup time to within
0.1 nanoseconds over that five nanosecond period takes as many as 50
simulations. Even after this, the error in the result can be as large as 0.05
nanoseconds.
The Star-Hspice bisection feature greatly reduces the amount of work and computational time required to find an accurate solution to this type of problem. The following pages show examples of using this feature to identify setup, hold, and minimum clock pulse width timing violations.
Understanding the Bisection Methodology

Bisection is a method of optimization that employs a binary search method to find the value of an input variable (target value) associated with a “goal” value of an output variable. The input and output variables may be of various types – for example, voltage, current, delay time or gain– related by some transfer function. In general, use a binary search to locate the output variable goal value within a search range of the input variable by iteratively halving that range to converge rapidly on the target value. At each iteration the “measured value” of the output variable is compared with the goal value. Bisection is employed in both the “pass/fail” method and the “bisection” method (see “Using Bisection”). The process is largely the same for either case.

The Star-Hspice bisection procedure involves two steps when solving the timing violation problem. First, the procedure detects whether the output transition occurred. Second, the procedure automatically varies the input parameter (T1 in Figure 30-1) to find the value for which the transition barely occurs. The Star-Hspice measurement and optimization features handle these two steps.

Measurement

Use the Star-Hspice MAX measurement function to detect success or failure of an output transition. In the case of a low-to-high output transition, a MAX measurement produces zero on failure, or approximately the supply voltage $V_{dd}$ on success. This measurement, using a goal of $V_{dd}$ minus a suitable small value to ensure a solution, is sufficient to drive the optimization.

Optimization

The bisection method is straightforward, given a single measurement with a goal and known upper and lower boundary values for the input parameter. The characterization engineer should be able to specify acceptable upper and lower boundary values.
Using Bisection

To use bisection, the following is required:

- A user-specified pair of upper and lower boundary input variable values. For a solution to be found, one of these values must result in an output variable result $|\text{goal value}|$ and the other in a result $< |\text{goal value}|$
- Specified goal value
- Error tolerance value. The bisection process stops when the difference between successive test values $\leq$ error tolerance. If the other criteria are met, see below.
- Related variables. Variables must be related by a monotonic transfer function, where a steadily progressing time (increase or decrease) results in a single occurrence of the “goal” value at the “target” input variable value.

The error tolerance is included in a relation used as a process-termination criterion.

Figure 30-2 shows an example of the binary search process used by the bisection algorithm. This example is of the “pass/fail” type, and is appropriate for a setup-time analysis that tests for the presence of an output transition as shown in Figure 30-1. Here, a long setup time $T_S (= T_2 - T_1)$ results in a VOUT transition (a “pass”), and a too-short setup time (where the latch has not stabilized the input data before the clock transition) results in a “fail.” A “pass” time value, for example, might be defined as any setup time $T_S$ that produces a VOUT output “minimum high” logic output level of $2.7 \, \text{V} –$ the “goal” value. The “target” value is that setup time that just produces the $V_{\text{OUT}}$ value of $2.7 \, \text{V}$. Since finding the exact value is impractical, if not impossible, an error tolerance is specified to give a solution arbitrarily close to the target value. The bisection algorithm performs tests for each of the specified boundary values to determine the direction in which to pursue the target value after the first bisection. In this example, the upper boundary value is a “pass” value, and the lower boundary value is a “fail” value.

To start the binary search, a lower boundary and upper boundary are specified. The program tests the point midway between the lower and upper boundaries (see Figure 30-2).
If the initial value passes the test, the target value must be less than the tested value (in this case), so the bisection algorithm moves the upper search limit to the value it just tested. If the test fails, the target value must greater than the tested value, so the bisection algorithm moves the lower limit to the value it just tested.

Then the algorithm tests a value midway between the new limits. The search continues in this manner, moving one limit or the other to the last midpoint, and testing the value midway between the new limits. The process stops when the difference between the latest test values is less than or equal to the user-specified error tolerance (normalized by multiplying by the initial boundary range).

**Examining the Command Syntax**

```
.MODEL <OptModelName> OPT METHOD=BISECTION ...
```

or

```
.MODEL <OptModelName> OPT METHOD=PASSFAIL ...
```

*OptModel-* The model to be used. Refer to “Statistical Analysis and Optimization” on page 13-1 for Name information on specification of optimization models in Star-Hspice.

**METHOD** Keyword to indicate which optimization method to use. For bisection, the method may be one of the following:

**BISECTION**

When the difference between the two latest test input values is within the error tolerance and the latest measured value exceeds the goal, bisection has succeeded, and stops. The process reports the optimized parameter that corresponded to the test value that satisfies this error tolerance, and this goal (passes).

**PASSFAIL**
When the difference between the two latest test input values is within the error tolerance and one of the values ≥ goal (passes) and the other fails, bisection has succeeded and stops. The process reports the value the input parameter value associated with the “pass” measurement.

**OPT**

Keyword to indicate optimization is to be performed

The parameters are passed in a normal optimization specification:

```
.PARAM <ParamName>=<OptParFun> (<Initial>, <Lower>, <Upper>)
```

In the BISECTION method, the measure results for <Lower> and <Upper> limits of <ParamName> must be on opposite sides of the GOAL value in the .MEASURE statement. For the PASSFAIL method, the measure must pass for one limit and fail for the other limit. The process ignores the value of the <Initial> field.

The error tolerance is a parameter in the model being optimized.

Note that the bisectional search is applied to only one parameter.

When the OPTLST option is set (.OPTION OPTLST=1), the process prints the following information for the BISECTION method:

```
  biseq-opt iter = <num_iterations>  xlo = <low_val>  xhi = <high_val>
  x = <result_low_val>  xnew = <result_high_val>
  err = <error_tolerance>
```

where x is the old parameter value and xnew is the new parameter value.

When .OPTION OPTLST=1, the process prints the following information for the PASSFAIL method:

```
  biseq-opt iter = <num_iterations>  xlo = <low_val>  xhi = <high_val>
  x = <result_low_val>  xnew = <result_high_val>
  measfail = 1
```

(measfail = 0 for a test failure for the x value).
Example: transient analysis .TRAN statement:

.TRAN <TranStep> <TranTime> SWEEP OPTIMIZE=<OptParFun>
+ RESULTS=<MeasureNames> MODEL=<OptModelName>

Example: transient .MEASURE statement:

.MEASURE TRAN <MeasureName> <MeasureClause> GOAL=<GoalValue>
Setup Time Analysis

This example uses a bisectional search to find the minimum setup time for a D flip-flop. The circuit for this example is `bisect/dff_top.sp` in the Star-Hspice `$installdir/demo/hspice` demonstration file directory. The files in Figures 30-2 and Figure 30-3 show the results of this demo. Note that setup time is not optimized directly, but is extracted from its relationship with the DelayTime parameter (the time preceding the data signal), which is the parameter being optimized.

Input listing

File: `$installdir/demo/hspice/bisect/dff_top.sp`
* DFF_top Bisection Search for Setup Time
* * PWL Stimulus
* v28 data gnd PWL
  + 0s 5v
  + 1n 5v
  + 2n 0v
  + Td = "DelayTime" $ Offsets Data from time 0 by DelayTime
v27 clock gnd PWL
  + 0s 0v
  + 3n 0v
  + 4n 5v
* * Specify DelayTime as the search parameter and provide the lower and upper limits.
* .PARAM DelayTime= Opt1 ( 0.0n, 0.0n, 5.0n )
* * Transient simulation with Bisection Optimization
* .TRAN 1n 8n Sweep Optimize = Opt1
  + Result = MaxVout $ Look at measure
  + Model = OptMod
*
Timing Analysis Using Bisection

Setup Time Analysis

* This measure finds the transition if it exists
* .MEASURE Tran MaxVout Max v(D_Output) Goal = 'v(Vdd)'
* This measure calculates the setup time value
* .MEASURE Tran SetupTime Trig v(Data) Val = 'v(Vdd)/2' Fall = 1
  + Targ v(Clock) Val = 'v(Vdd)/2' Rise = 1
* Optimization Model
* .MODEL OptMod Opt
  + Method = Bisection
  .OPTIONS Post Brief NoMod
*************************************
* AvanLink to Cadence Composer by Avant!
* Hspice Netlist
* May 31 15:24:09 1994
*************************************
 .MODEL nmos nmos LEVEL=2
 .MODEL pmos pmos LEVEL=2
 .Global vdd gnd
 .SUBCKT XGATE control in n_control out
  m0 in n_control out vdd pmos l=1.2u w=3.4u
  m1 in control out gnd nmos l=1.2u w=3.4u
 .ends
 .SUBCKT INV in out wp=9.6u wn=4u l=1.2u
  mb2 out in gnd gnd nmos l=1 w=wn
  mb1 out in vdd vdd pmos l=1 w=wp
 .ends
 .SUBCKT DFF c d nc ng
  Xi64 nc net46 c net36 XGATE
  Xi66 nc net38 c net39 XGATE
  Xi65 c ng nc net36 XGATE
  Xi62 c d nc net39 XGATE
  Xi60 net722 ng INV
  Xi61 net46 net38 INV
  Xi59 net36 net722 INV
  Xi58 net39 net46 INV
Setup Time Analysis

Timing Analysis Using Bisection

c20 net36 gnd c=17.09f
c15 net39 gnd c=15.51f
c12 net46 gnd c=25.78f
c4 nq gnd c=25.28f
c3 net722 gnd c=19.48f
c16 net38 gnd c=16.48f
.ENDS

*-------------------------------------------------------------
* Main Circuit Netlist:
*-------------------------------------------------------------
v14 vdd gnd dc=5
c10 vdd gnd c=35.96f
c15 d_output gnd c=21.52f
c12 dff_nq gnd c=11.73f
c11 net31 gnd c=42.01f
c14 net27 gnd c=34.49f
c13 net25 gnd c=41.73f
c8 clock gnd c=5.94f
c7 data gnd c=7.93f
Xi3  net25 net31 net27 dff_nq DFF l=1u wn=3.8u wp=10u
Xi6  data net31 INV
Xi5  net25 net27 INV
Xi4  clock net25 INV
Xi2  dff_nq d_output INV wp=26.4u wn=10.6u
.END
Figure 30-2: Bisection Example for Three Iterations

First bisection value is midway between specified boundaries. First test value passes because measured value ≥ goal value (2.7 V in this case).

Lower boundary $X_L$ - test fails

Target value

First test value passes, new upper test limit. Second test value is midway between new upper limit and lower boundary. Second test value fails.

$X_L = (X_1 + X_2)/2$

Second test value becomes new lower limit. Third test value is midway between new lower limit and current upper limit.

$X_L = (X_1 + X_2)/2$

Continue halving test region until the interval between successive test values meets the criterion:

$\delta = |X_{n} - X_{n-1}| \leq 0.01 \times |X_{n} - X_u|$

then report the value $X_{n}$ (associated with the measured value that passed). If you select the bisection method, the reported value must correspond with the condition:

measured value - goal ≥ 0

Results

The top plot in Figure 30-3 shows the relationship between the clock and data pulses that determine the setup time. The bottom plot shows the output transition.

Figure 30-3: Transition at Minimum Setup Time

Find the actual value for the setup time in the “Optimization, Results” section of the Star-Hspice listing file:

```
optimization completed, the condition
relin = 1.0000E-03 is satisfied
**** optimized parameters opt1
.PARAM DelayTime = 1.7188n
...
maxvout = 5.0049E+00 at= 4.5542E-09
from = 0.0000E+00 to= 8.0000E-09
setuptime = 2.8125E-10 targ= 3.5000E-09 trig= 3.2188E-09
```
This listing file excerpt shows that the optimal value for the setup time is 0.28125 nanoseconds.

The top plot in Figure 30-4 shows examples of early and late data transitions, as well as the transition at the minimum setup time. The bottom plot shows how the timing of the data transition affects the output transition. These results were produced with the following analysis statement:

```
* Sweep 3 values for DelayTime       Early  Optim  Late
* .TRAN 1n 8n Sweep DelayTime Poi 3 0.0n 1.7188n 5.0n
```

**Figure 30-4: Early, Minimum, and Late Setup and Hold Times**

This analysis produces the following results:

```
*** parameter DelayTime = .000E+00 *** $ Early
setup time = 2.0000E-09  targ= 3.5000E-09  trig= 1.5000E-09
*** parameter DelayTime = 1.719E-09 *** $ Optimal
setup time = 2.8120E-10  targ= 3.5000E-09  trig= 3.2188E-09
*** parameter DelayTime = 5.000E-09 *** $ Late
setup time = -3.0000E-09  targ= 3.5000E-09  trig= 6.5000E-09
```
Minimum Pulse Width Analysis

This example uses a pass/fail bisectional search to find a minimum pulse width required to allow the input pulse to propagate to the output of an inverter. The circuit for this example is \texttt{/bisect/inv_a.sp} in the \texttt{$installdir/demo/hspice} directory. The results of this demo are shown in Figure 30-5.

Input listing

File: \texttt{$installdir/demo/bisect/inv_a.sp}

$ Inv_a.sp testing bisectional search, cload=10p & 20p

* Parameters

* .PARAM Cload =10p \texttt{See end of deck for Alter}
* .PARAM Tpw =opt1(0,0,15n) \texttt{Used in Pulsed Voltage Source, v1}

* Transient simulation with PassFail Optimization

* .TRAN .1n 20n Sweep Optimize = Opt1
+ results = Tprop
+ Model = Optmod

* .MODEL OptMod Opt Method = PassFail
* .MEASURE Tran Tprop Trig v(in) Val=2.5 Rise=1
+ Targ v(out) val=2.5 Rall=1

* .OPTION nomod acct=3 post autostop
* .GLOBAL 1

* The Circuit

* vcc 1 0 5
vin in 0 pulse(0,5 1n 1n 1n Tpw 20n)
rin in 0 1e13
rout out 0 10k
cout out 0 cload
x1 in out inv

* .SUBCKT Inv in out
  mn out in 0 0 nch W=10u L=1u
  mp out in 1 1 pch W=10u L=1u

.ENDS
* Models

.PARAM
+ mult1=1 x1=0.06u xwn=0.3u xwp=0.3u
+ tox=200 delvton=0 delvtop=0 rsh=50
+ rshp=150

.MODEL nch nmos
+ LEVEL=28
+ lmlt=mult1 wmlt=mult1 wref=22u lref=4.4u
+ x1=x1 xw=xwn tox=tox delvto=delvton rsh=rshn
+ ld=0.06u wd=0.2u acm=2 ldif=0 hdif=2.5u
+ rs=0 rd=0 rsc=0 js=3e-04 jsw=9e-10
+ cj=3e-04 mj=.5 pb=.8 cjsw=3e-10 mjsw=.3 php=.8
+ fc=.5 capop=4 xqc=.4 meto=0.08u
+ tlev=1 cta=0 ctp=0 tlevc=0 nlev=0
+ trs=1.6e-03 bex=-1.5 tcv=1.4e-03

* dc model
+ x2e=0 x3e=0 x2ul=0 x2ms=0 x2u0=0 x2m=0
+ vfb0=-.5 phi0=0.65 k1=.9 k2=.1 eta0=0
+ muz=500 u00=.075 x3ms=15 u1=.02 x3ul=0
+ b1=.28 b2=.22 x33m=0.000000e+00
+ alpha=1.5 vcr=20 n0=1.6 wfac=15 wfacu=0.25
+ lvfb=0 lk1=.025 lk2=.05 lalpha=5

.MODEL pch pmos
+ LEVEL=28
+ lmlt=mult1 wmlt=mult1 wref=22u lref=4.4u
+ x1=x1 xw=xwp tox=tox delvto=delvtop rsh=rshp
+ ld=0.08u wd=0.2u acm=2 ldif=0 hdif=2.5u
+ rs=0 rd=0 rsc=0 rsh=rshp js=3e-04 jsw=9e-10
+ cj=3e-04 mj=.5 pb=.8 cjsw=3e-10 mjsw=.3 php=.8
+ fc=.5 capop=4 xqc=.4 meto=0.08u
+ tlev=1 cta=0 ctp=0 tlevc=0 nlev=0
+ trs=1.6e-03 bex=-1.5 tcv=1.7e-03

* dc model
+ x2e=0 x3e=0 x2ul=0 x2ms=0 x2u0=0 x2m=5
+ vfb0=-.1 phi0=0.65 k1=.35 k2=0 eta0=0
+ muz=200 u00=.175 x3ms=8 u1=0 x3ul=0.0
+ b1=.25 b2=.25 x33m=0.0 alpha=0 vcr=20
+ n0=1.3 wfac=12.5 wfacu=.2 lvfb=0 lk1=--.05
* Alter for second load value
* .ALTER $ repeat optimization for 20p load
 .PARAM Cload=20p
 .END

Results

Figure 30-5 shows the results of the pass/fail search for two different capacitive loads.

**Figure 30-5: Results of Bisectional Pass/Fail Search**
This chapter contains examples of basic file construction techniques, advanced features, and simulation tricks. Several Star-Hspice input files are listed and described.

The following topics are covered in this chapter:

- Using the Demo Directory Tree
- Running the Two-Bit Adder Demo
- Running the MOS I-V and C-V Plotting Demo
- Running the CMOS Output Driver Demo
- Running the Temperature Coefficients Demo
- Simulating Electrical Measurements
- Modeling Wide-Channel MOS Transistors
- Examining the Demonstration Input Files
Using the Demo Directory Tree

The last section of this chapter is a listing of demonstration files, which are designed as good training examples. These examples are included with most Star-Hspice distributions in the demo directory tree, where $installdir$ is the installation directory environment variable:

```
$installdir/demo/hspice  /alge    algebraic output
                     /apps    general applications
                     /behave  analog behavioral components
                     /bench   standard benchmarks
                     /bjt     bipolar components
                     /ccchar  cell characterization prototypes
                     /ciropt  circuit level optimization
                     /ddl     Discrete Device Library
                     /devopt  device level optimization
                     /fft     Fourier analysis
                     /filters filters
                     /mag     transformers, magnetic core components
                     /mos     MOS components
                     /pci     Intel Peripheral Component Interconnect
                     /rad     radiation effects (photocurrent)
                     /sources dependent and independent sources
                     /tline   filters and transmission lines
```
Running the Two-Bit Adder Demo

This two-bit adder demonstrates many techniques to improve circuit simulation efficiency, accuracy, and productivity. The adder in demonstration file $installdir/demo/hspice/apps/mos2bit.sp is composed of two-input NAND gates defined by the subcircuit NAND. CMOS devices are parameterized with length, width, and output loading. Descriptive names enhance the readability of this circuit.

The subcircuit ONEBIT defines the two half adders with carry in and carry out. The two-bit adder is created by two calls to ONEBIT. Independent piecewise linear voltage sources provide input stimuli. Complex waveforms are created by the “R” repeat function.

Figure 31-1: One-bit Adder Subcircuit

Figure 31-2: Two-bit Adder Circuit
Figure 31-3: 1-bit NAND Gate Binary Adder

MOS Two-Bit Adder Input File

*FILE: MOS2BIT.SP - ADDER - 2 BIT ALL-NAND-GATE BINARY ADDER
.OPTIONS ACCT NOMOD FAST autostop scale=1u gmindc=100n
.param lmin=1.25  hi=2.8v lo=.4v vdd=4.5
.global vdd
.TRAN .5NS 60NS
.graph TRAN V(c[0]) V(carry-out_1) V(c[1]) V(carry-out_2)
+ par('V(carry-in)/6 + 1.5')
+ par('V(a[0])/6 + 2.0')
+ par('V(b[0])/6 + 2.5') (0,5)
.MEAS PROP-DELAY TRIG V(carry-in) TD=10NS VAL='vdd*.5' RISE=1
+ TARG V(c[1]) TD=10NS VAL='vdd*.5' RISE=3
*
.MEAS PULSE-WIDTH TRIG V(carry-out_1) VAL='vdd*.5' RISE=1
+ TARG V(carry-out_1) VAL='vdd*.5' FALL=1
*
.MEAS FALL-TIME TRIG V(c[1]) TD=32NS VAL='vdd*.9' FALL=1
+ TARG V(c[1]) TD=32NS VAL='vdd*.1' FALL=1
VDD vdd gnd DC vdd
X1 A[0] B[0] carry-in C[0] carry-out_1 ONEBIT
Subcircuit Definitions

.subckt NAND in1 in2 out wp=10 wn=5
   M1 out in1 vdd vdd P W=wp L=imin ad=0
   M2 out in2 vdd vdd P W=wp L=imin ad=0
   M3 out in1 mid gnd N W=wn L=imin as=0
   M4 mid in2 gnd gnd N W=wn L=imin ad=0
   CLOAD out gnd 'wp*5.7f'
.ends

* switch model equivalent of the NAND. Gives a 10 times
* speedup over the MOS version.

.subckt NANDx in1 in2 out wp=10 wn=5
   G1 out vdd vdd in1 LEVEL=1 MIN=1200 MAX=1MEG 1.MEG -.5MEG
   G2 out vdd vdd in2 LEVEL=1 MIN=1200 MAX=1MEG 1.MEG -.5MEG
   G3 mid in1 gnd LEVEL=1 MIN=1200 MAX=1MEG 1.MEG -.5MEG
   G4 mid gnd in2 gnd LEVEL=1 MIN=1200 MAX=1MEG 1.MEG -.5MEG
   cout out gnd 300f
.ends

.subckt ONEBIT in1 in2 carry-in out carry-out
   X1 in1 in2 #1_nand NAND
   X2 in1 #1_nand 8 NAND
   X3 in2 #1_nand 9 NAND
   X4 8 9 10 NAND
   X5 carry-in 10 half1 NAND
   X6 carry-in half1 half2 NAND
   X7 10 half1 13 NAND
   X8 half2 13 out NAND
   X9 half1 #1_nand carry-out NAND
.ends ONEBIT

Stimuli

V1 carry-in gnd PWL(0NS,lo 1NS,hi 7.5NS,hi 8.5NS,lo 15NS lo R
V2 A[0] gnd PWL (0NS,hi 1NS,lo 15.0NS,lo 16.0NS,hi 30NS hi R
V3 A[1] gnd PWL (0NS,hi 1NS,lo 15.0NS,lo 16.0NS,hi 30NS hi R
V4 B[0] gnd PWL (0NS,hi 1NS,lo 30.0NS,lo 31.0NS,hi 60NS hi
V5 B[1] gnd PWL (0NS,hi 1NS,lo 30.0NS,lo 31.0NS,hi 60NS hi

Models
.MODEL N NMOS LEVEL=3 VTO=0.7 UO=500 KAPPA=.25 KP=30U
+ ETA=.01 THETA=.04 VMAX=2E5 NSUB=9E16 TOX=400 GAMMA=1.5
+ PB=0.6 JS=.1M XJ=0.5U LD=0.1U NFS=1E11 NSS=2E10
+ RSH=80 CJ=.3M MJ=0.5 CJSW=.1N MJSW=0.3
+ acm=2 capop=4

.MODEL P PMOS LEVEL=3 VTO=-0.8 UO=150 KAPPA=.25 KP=15U
+ ETA=.015 THETA=.04 VMAX=5E4 NSUB=1.8E16 TOX=400 GAMMA=.672
+ PB=0.6 JS=.1M XJ=0.5U LD=0.15U NFS=1E11 NSS=2E10
+ RSH=80 CJ=.3M MJ=0.5 CJSW=.1N MJSW=0.3
+ acm=2 capop=4

.END
Running the MOS I-V and C-V Plotting Demo

It is often necessary to review the basic transistor characteristics to diagnose a simulation or modeling problem. This demonstration file, $installdir/demo/hspice/mos/mosivcv.sp, is a template file that can be used with any MOS model. The example shows the easy input file creation and the complete graphical results display. The following features aid model evaluations:

- **SCALE=1u**
  Sets the element units to microns from meters since users generally think in microns rather than meters

- **DCCAP**
  Forces the voltage variable capacitors to be evaluated during a DC sweep

- **node names**
  Makes the circuit easy to understand. Up to 16 characters can be used in the symbolic name

- **.GRAPH**
  .GRAPH statements create high resolution plots. A graph model can be added to set additional characteristics

This template provides the ability to get plots of internal variables such as:

- **i(mn1)**
  i1, i2, i3, or i4 can specify the true branch currents for each transistor node

- **LV18(mn6)**
  Total gate capacitance (C-V plot)

- **LX7(mn1)**
  Gate transconductance GM. (LX8 specifies GDS, and LX9 specifies GMB)
Figure 31-4: MOS IDS Plot

Figure 31-5: MOS VGS Plot
Figure 31-6: MOS GM Plot

Figure 31-7: MOS C-V Plot
MOS I-V and C-V Plot Example Input File

*FILE: MOSIVCV.SP IDS, VGS, CV AND GM PLOTS

.OPTIONS SCALE=1U DCCAP
.DC VDDN 0 5.0 .1 $VBBN 0 -3 -3 sweep supplies
.PARAM ww=8 LL=2

$ ids-vds curves
.GRAPH 'I_VG=1' =I(MN1) 'I_VG=2' =I(MN2) 'I_VG=3' =I(MN3) + 'I_VG=4' =I(MN4)
.GRAPH 'I_VG=-1'=I(MP1) 'I_VG=-2'=I(MP2) 'I_VG=-3'=I(MP3) + 'I_VG=-4'=I(MP4)

$ ids-VGs curves
.GRAPH 'I_VD=.5'=I(MN6) 'I_VD=-.5'=I(MP6)
$ gate caps (cgs+cgd+cgb)
.GRAPH 'CG-TOT_N'=LX18(MN6) 'CG-TOT_P'= LX18(MP6)

$ gm
.GRAPH 'GM_N'=LX7(MN6) 'GM_P'=LX7(MP6)

VDDN vdd_n gnd 5.0
VBBN vbb_n gnd 0.0
EPD vdd_p gnd vdd_n gnd -1 $ reflect vdd for P devices
EPB vbb_p gnd vbb_n gnd -1 $ reflect vbb for P devices

V1 vg1 gnd 1
V2 vg2 gnd 2
V3 vg3 gnd 3
V4 vg4 gnd 4
V5 vddlow_n gnd .5
V-1 vg-1 gnd -1
V-2 vg-2 gnd -2
V-3 vg-3 gnd -3
V-4 vg-4 gnd -4
V-5 vddlow_p gnd -.5

MN1 vdd_n vg1 gnd vbb_n N W=ww L=LL
MN2 vdd_n vg2 gnd vbb_n N W=ww L=LL
MN3 vdd_n vg3 gnd vbb_n N W=ww L=LL
MN4 vdd_n vg4 gnd vbb_n N W=ww L=LL

MP1 gnd vg-1 vdd_p vbb_p P W=ww L=LL
Running Demonstration Files

Running the MOS I-V and C-V Plotting Demo

MP2  gnd vg-2 vdd_p vbb_p P W=ww L=LL
MP3  gnd vg-3 vdd_p vbb_p P W=ww L=LL
MP4  gnd vg-4 vdd_p vbb_p P W=ww L=LL

MN6  vddlown_vdd_n gnd vbb_n N W=ww L=LL
MP6  gnd vdd_p vddlown_p vbb_p P W=ww L=LL

.MODEL      N  NMOS LEVEL=3   VTO=0.7  UO=500    KAPPA=.25
+  KP=30U   ETA=.01 THETA=.04 VMAX=2E5 NSUB=9E16 TOX=400
+  GAMMA=1.5 PB=0.6 JS=.1M   XJ=0.5U  LD=0.1U NFS=1E11
+ NSS=2E10  RSH=80 CJ=.3M     MJ=0.5   CJSW=.1N  MJSW=0.3
+  acm=2 capop=4
*

.MODEL      P   PMOS LEVEL=3   VTO=-0.8 UO=150  KAPPA=.25
+  KP=15U   ETA=.015 THETA=.04 VMAX=5E4 NSUB=1.8E16 TOX=400
+  GAMMA=.67 PB=0.6 JS=.1M    XJ=0.5U  LD=0.15U NFS=1E11
+ NSS=2E10  RSH=80 CJ=.3M     MJ=0.5   CJSW=.1N  MJSW=0.3
+  acm=2 capop=4
.

.END
Running the CMOS Output Driver Demo

ASIC designers face the problem of integrating high performance IC parts onto a printed circuit board (PCB). The output driver circuit is most critical to the overall system performance. The demonstration file $installdir/demo/hspice/apps/asic1.sp shows the models for an output driver, the bond wire and leadframe, and a six inch length of copper transmission line.

This simulation demonstrates how to:

- Define parameters and measure test outputs
- Use the “LUMP5” macro to input geometric units and convert them to electrical units
- Use .MEASURE statements to calculate the peak local supply current, voltage drop, and power
- Measure RMS power, delay, rise times and fall times
- Simulate and measure an output driver under load. The load consists of:
  - Bondwire and leadframe inductance
  - Bondwire and leadframe resistance
  - Leadframe capacitance
  - Six inches of 6 mil copper on a FR-4 printed circuit board
  - Capacitive load at end of copper wire

The Star-Hspice strategy is to:

- Create a five-lump transmission line model for the copper wire
- Create single lumped models for leadframe loads
Figure 31-8: Noise Bounce

Figure 31-9: Asic1.sp Demo Local Supply Voltage
Running the CMOS Output Driver Demo

Running Demonstration Files

Figure 31-10: Asic1.sp Demo Local Supply Current

Figure 31-11: Asic1.sp Demo Input and Output Signals

CMOS Output Driver Example Input File

* FILE: ASIC1.SP
* SIMULATE AN OUTPUT DRIVER DRIVING 6 INCHES OF 6MIL PRINTED
* CIRCUIT BOARD COPPER WITH 25PF OF LOAD CAPACITANCE
* MEASURE PEAK TO PEAK GROUND VOLTAGE
* MEASURE MAXIMUM GROUND CURRENT
* MEASURE MAXIMUM SUPPLY CURRENT

GROUND BOUNCE FOR I/O CMOS DRIVER 1200/1.2 & 800/1.2 MICRONS
.OPTIONS POST=2 RELVAR=.05
.TRAN .25N 30N
.MEASURE IVDD_MAX MAX PAR('ABS(I(VD))')
.MEASURE IVSS_MAX MAX PAR('ABS(I(VS))')
.MEASURE PEAK_GNDV PP V(LVSS)
.MEASURE PEAK_IVD PP PAR(' ABS(I(VD)*V(VDD,OUT)) ')
.MEASURE PEAK_IVS PP PAR(' ABS(I(VS)*V(VSS,OUT)) ')
.MEASURE RMS_POWER RMS POWER
.MEASURE FALL_TIME TRIG V(IN) RISE=1 VAL=2.5V + TARG V(OUT) FALL=1 VAL=2.5V
.MEASURE RISE_TIME TRIG V(IN) FALL=1 VAL=2.5V + TARG V(OUT) RISE=1 VAL=2.5V
.MEASURE TLINE_DLY TRIG V(OUT) RISE=1 VAL=2.5V + TARG V(OUT2) RISE=1 VAL=2.5V

Input Signals

VIN IN LGND PWL(0N 0V, 2N 5V, 12N 5, 14N 0)

* OUTPUT DRIVER
MP1 LOUT IN LVDD LVDD P W=1400U L=1.2U
MN1 LOUT IN LVSS LVSS N W=800U L=1.2U
xout LOUT OUT LEADFRAME
*POWER AND GROUND LINE PARASITICS
Vd VDD GND 5V
xdd vdd lvdd leadframe
Vs VSS gnd 0v
xss vss lvss leadframe

*OUTPUT LOADING - 3 INCH FR-4 PC BOARD + 5PF LOAD +
*3 INCH FR-4 + 5PF LOAD
Running the CMOS Output Driver Demo

XLOAD1 OUT OUT1 GND LUMP5 LEN=3 WID=.006
CLOAD1 OUT1 GND 5PF
XLOAD2 OUT1 OUT2 GND LUMP5 LEN=3 WID=.006
CLOAD2 OUT2 GND 5PF

.macro leadframe in out
rframe in mid .01
lframe mid out 10n
cframe mid gnd .5p
.ends

*Transmission Line Parameter Definitions
.param rho=.6mho/sq cap=.55nf/in**2 ind=60ph/sq

*The 5-lump macro defines a parameterized transmission line
.macro lump5 in out ref len_lump5=1 wid_lump5=.1
.prot
.param reseff='len_lump5*rho/wid_lump5*5'
+ capeff='len_lump5*wid_lump5*cap/5'
+ indeff='len_lump5*ind/wid_lump5*5'
.r1 in 1 reseff
c1 1 ref capeff
.l1 1 2 indeff
.r2 2 3 reseff
c2 3 ref capeff
.l2 3 4 indeff
.r3 4 5 reseff
c3 5 ref capeff
.l3 5 6 indeff
.r4 6 7 reseff
c4 7 ref capeff
.l4 7 8 indeff
.r5 8 9 reseff
c5 9 ref capeff
.l5 9 out indeff
.unprot
.ends

Model Section

.MODEL N NMOS LEVEL=3 VTO=0.7 UO=500 KAPPA=.25 ETA=.03
+ THETA=.04 VMAX=2E5 NSUB=9E16 TOX=200E-10 GAMMA=1.5 PB=0.6 +
+ JS=.1M XJ=0.5U LD=0.0 NFS=1E11 NSS=2E10 capop=4
.MODEL P PMOS LEVEL=3 VTO=-0.8 UO=150 KAPPA=.25 ETA=.03 + THETA=.04 VMAX=5E4 NSUB=1.8E16 TOX=200E-10 GAMMA=.672 + PB=0.6 JS=1M XJ=0.5U LD=0.0 NFS=1E11 NSS=2E10 capop=4
.end
IVDD_MAX        =  0.1141        AT=  1.7226E-08
               FROM=  0.0000E+00    TO=  3.0000E-08
IVSS_MAX        =  0.2086        AT=  3.7743E-09
               FROM=  0.0000E+00    TO=  3.0000E-08
PEAK_GNDV  =  3.221     FROM=  0.0000E+00  TO=  3.0000E-08
PEAK_IVD    =  0.2929    FROM=  0.0000E+00  TO=  3.0000E-08
PEAK_IVS    =  0.3968    FROM=  0.0000E+00  TO=  3.0000E-08
RMS_POWER  =  0.1233    FROM=  0.0000E+00  TO=  3.0000E-08
FALL_TIME  = 1.2366E-09 TARG= 1.9478E-09  TRIG=  7.1121E-10
RISE_TIME  = 9.4211E-10 TARG= 1.4116E-08  TRIG=  1.3173E-08
TLINE_DLY  = 1.6718E-09 TARG= 1.5787E-08  TRIG=  1.4116E-08
Running the Temperature Coefficients Demo

SPICE-type simulators do not always automatically compensate for variations in temperature. The simulators make many assumptions that are not valid for all technologies. Star-Hspice has first-order and second-order temperature coefficients in many of the critical model parameters to assure accurate simulations. There are two methods to optimize these temperature coefficients.

The first method uses the DC sweep variable TEMP. All of the Star-Hspice analysis sweeps allow two sweep variables; one of these must be the optimize variable to do an optimization. Sweeping TEMP limits the component to a linear element such as resistor, inductor, or capacitor. The second method uses multiple components at different temperatures.

In the following example, demo file $installdir/demo/hspice/ciropt/opttemp.sp, three circuits of a voltage source and a resistor are simulated at -25, 0, and +25 °C from nominal using the DTEMP parameter for element delta temperatures. The resistors share a common model. Three temperatures are necessary to solve a second order equation. This simulation template can be easily extended to a transient simulation of nonlinear components, such as bipolar transistors, diodes, and FETs.

Some simulation shortcuts are used in this example. In the internal output templates for resistors, LV1 (resistor) is the conductance (reciprocal resistance) at the desired temperature, allowing the optimization to be done in the resistance domain. To optimize more complex elements, use the current or voltage domain with measured sweep data. Also, the error function is expecting a sweep on at least two points, requiring the data statement to have two duplicate points.

Optimized Temperature Coefficients Example Input File

*FILE OPTTEMP.SP   OPTIMIZE RESISTOR TC1 AND TC2
v-25 1 0 1v
v0   2 0 1v
v+25 3 0 1v
r-25 1 0 rmod dtemp=-25
r0   2 0 rmod dtemp=0
```
r+25 3 0 rmod dtemp=25
.model rmod R res=1k tclr=tclr_opt tc2r=tc2r_opt

Optimization Section

.model optmod opt
.dc data=RES_TEMP optimize=opt1
+     results=r@temp1,r@temp2,r@temp3
+     model=optmod
.param tclr_opt=opt1(.001, -.1,.1)
.param tc2r_opt=opt1(1u, -1m, 1m)
.meas r@temp1 err2 par(R_meas_t1) par('1.0 / lv1(r-25)')
.meas r@temp2 err2 par(R_meas_t2) par('1.0 / lv1(r0) ')
.meas r@temp3 err2 par(R_meas_t3) par('1.0 / lv1(r+25) ')
* * Output section *
.dc data=RES_TEMP
.print 'r1_diff'=par('1.0/lv1(r-25)')
+   'r2_diff'=par('1.0/lv1(r0) ')
+   'r3_diff'=par('1.0/lv1(r+25)')
.data RES_TEMP R_meas_t1 R_meas_t2 R_meas_t3
950 1000 1010
950 1000 1010
.enddata
.end
```
Simulating Electrical Measurements

In this example, Star-Hspice simulates the electrical measurements used to characterize devices for data sheet information. The demonstration file for this example is $installdir/demo/hspice/ddl/t2n2222.sp. The example automatically includes DDL models by reference using the DDLPATH environment variable, or through the .OPTION SEARCH="path". It also combines an AC circuit and measurement with a transient circuit and measurement.

The AC circuit measures the maximum Hfe, the small signal common emitter gain. The WHEN option of the .MEASURE statement allows calculation of the unity gain frequency and the phase at the frequency specified with WHEN. In the “Transient Measurements” section of the input file, a segmented transient statement is used to speed up the simulation and compress the output graph. Measurements include:

- **TURN ON** from 90% of input rising to 90% of output falling
- **OUTPUT FALL** from 90% to 10% of output falling
- **TURN OFF** from 10% of input falling to 10% of output rising
- **OUTPUT RISE** from 10% to 90% of output rising

**Figure 31-12: T2N2222 Optimization**
T2N2222 Optimization Example Input File

* FILE: T2N2222.SP
** assume beta=200 ft250meg at ic=20ma and vce=20v for 2n2222
.OPTION   nopage autostop search=' '
*** ft measurement
* the net command is automatically reversing the sign of the
* power supply current for the network calculations
.NET I(vce) IBASE ROUT=50 RIN=50
VCE   C 0 vce
IBASE 0 b   AC=1 DC=ibase
xqft c b 0 t2n2222
.ac dec 10 1 1000meg
.graph s21(m)  h21(m)
.measure 'phase @h21=0db' WHEN h21(db)=0
.measure 'h21_max'  max h21(m)
.measure 'phase @h21=0deg'  FIND h21(p) WHEN h21(db)=0
.param ibase=1e-4  vce=20 tauf=5.5e-10

Transient Measurements

**   vccf power supply for forward reverse step recovery time
**   vccr power supply for inverse reverse step recovery time
**   VPLUSF positive voltage for forward pulse generator
**   VPLUSr positive voltage for reverse pulse generator
**   Vminusf positive voltage for forward pulse generator
**   Vminusr positive voltage for reverse pulse generator
**   rloadf load resistor for forward
**   rloadr load resistor for reverse
.param vccf=30v
.param VPLUSF=9.9v
.param VMINUSF=-0.5v
.param rloadf=200
.TRAN 1N 75N 25N 200N 1N 300N 25N 1200N
.measure 'turn-on time 'trig par('v(inf)-0.9*vplusf') val=0
+ rise=1 targ par('v(outf)-0.9*vccf') val=0 fall=1
.measure 'fall time 'trig par('v(outf)-0.9*vccf') val=0
+ fall=1 targ par('v(outf)-0.1*vccf') val=0 fall=1
.measure 'turn-off time' trig par('v(inf)-0.1*vplusf) val=0
+ fall=1 targ par('v(outf)-0.1*vccf') val=0 rise=1
.measure 'rise time 'trig par('v(outf)-0.1*vccf') val=0
+ rise=1 targ par('v(outf)-0.9*vccf') val=0 rise=1
.graph V(INF) V(OUTF)
VCCF VCCF 0 vccf
RLOADF VCCF OUTF RLOADF
RINF INF VBASEF 1000
RPARF INF 0 58
XSCOPF OUTF 0 SCOPE
VINF INF 0 PL VMINUSF 0S VMINUSF 5NS
+ VPLUSF 7NS VPLUSF 207NS VMINUSF 209NS
* CCX0F VBASEF OUTF CCX0F
* CEX0F VBASEF 0 CEX0F
XQF OUTF VBASEF 0 t2n2222

.MACRO SCOPE VLOAD VREF
RIN VLOAD VREF 100K
CIN VLOAD VREF 12P
.EOM
.END
Modeling Wide-Channel MOS Transistors

Selecting an appropriate model for I/O cell transistors improves the accuracy of simulation. For wide-channel devices, model the transistor as a group of transistors connected in parallel with appropriate RC delay networks, rather than as one transistor, because of the delay introduced by the polysilicon gate. When scaling to higher speed technologies, the area of the polysilicon gate decreases, reducing the gate capacitance. However, if you scale the gate oxide thickness, it increases the capacitance per unit area, increasing the RC product. The following example illustrates the effect on delay due to this scaling. For example, for a device with

channel width = 100 microns
channel length = 5 microns
gate oxide thickness = 800 Angstroms

the resulting RC product for the polysilicon gate is

\[ R_{poly} = \frac{W}{L} \cdot 40 \]

\[ poly = \frac{E_{sio} \cdot nsi}{tox} \cdot L \cdot W \]

\[ R_{poly} = \frac{100}{5} \cdot 40 = 800, \quad C_o = \frac{3.9 \cdot 8.86}{800} \cdot 100 \cdot 5 = 215 \text{ fF} \]

\[ RC = 138 \text{ ps} \]

For a transistor with

channel width = 100 microns
channel length = 1.2 microns
gate oxide thickness = 250 Angstroms

\[ R_{poly} = \frac{\text{channel width}}{\text{channel length}} \cdot 40 \]
You can model the RC delay introduced in modern CMOS technologies by using a nine-stage ladder model.

**Example of Nine-Stage Ladder Model**

* FILE: ASIC3.SP Test of 9 Stage Ladder Model
  .subckt lrgtp drain gate source bulk
  m1 drain gate source bulk p w='wt/18' l=lt
  m2 drain q1 source bulk p w='wt/9' l=lt
  m3 drain q2 source bulk p w='wt/9' l=lt
  m4 drain q3 source bulk p w='wt/9' l=lt
  m5 drain q4 source bulk p w='wt/9' l=lt
  m6 drain q5 source bulk p w='wt/9' l=lt
  m7 drain q6 source bulk p w='wt/9' l=lt
  m8 drain q7 source bulk p w='wt/9' l=lt
m9 drain g8 source bulk p w='wt/9' l=lt
m10 drain g9 source bulk p w='wt/18' l=lt
r1 gate g1 'wt/lt*rpoly/9'
r2 g1 g2 'wt/lt*rpoly/9'
r3 g2 g3 'wt/lt*rpoly/9'
r4 g3 g4 'wt/lt*rpoly/9'
r5 g4 g5 'wt/lt*rpoly/9'
r6 g5 g6 'wt/lt*rpoly/9'
r7 g6 g7 'wt/lt*rpoly/9'
r8 g7 g8 'wt/lt*rpoly/9'
r9 g8 g9 'wt/lt*rpoly/9'
.ends lrgtp
.end pro
.end

Figure 31-14: Asic3 Single vs. Lumped Model
## Examining the Demonstration Input Files

### Algebraic Output Variable Examples

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alg.sp</td>
<td>demonstration of algebraic parameters</td>
</tr>
<tr>
<td>alg_fil.sp</td>
<td>magnitude response of behavioral filter model</td>
</tr>
<tr>
<td>alg_vco.sp</td>
<td>voltage controlled oscillator</td>
</tr>
<tr>
<td>alg_vf.sp</td>
<td>voltage-to-frequency converter behavioral model</td>
</tr>
</tbody>
</table>

### Applications of General Interest

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alm124.sp</td>
<td>AC, noise, transient op-amp analysis</td>
</tr>
<tr>
<td>alter2.sp</td>
<td>.ALTER examples</td>
</tr>
<tr>
<td>ampg.sp</td>
<td>pole/zero analysis of a G source amplifier</td>
</tr>
<tr>
<td>asic1.sp</td>
<td>ground bounce for I/O CMOS driver</td>
</tr>
<tr>
<td>asic3.sp</td>
<td>ten-stage lumped MOS model</td>
</tr>
<tr>
<td>bjt2bit.sp</td>
<td>BJT two-bit adder</td>
</tr>
<tr>
<td>bjt4bit.sp</td>
<td>four-bit, all NAND gate binary adder</td>
</tr>
<tr>
<td>bjtdiff.sp</td>
<td>BJT diff amp with every analysis type</td>
</tr>
<tr>
<td>bjtschmt.sp</td>
<td>bipolar Schmidt trigger</td>
</tr>
<tr>
<td>bjtsense.sp</td>
<td>bipolar sense amplifier</td>
</tr>
<tr>
<td>cellchar.sp</td>
<td>ASIC inverter cell characterization</td>
</tr>
<tr>
<td>crystal.sp</td>
<td>crystal oscillator circuit</td>
</tr>
<tr>
<td>gaasamp.sp</td>
<td>simple GaAsFET amplifier</td>
</tr>
<tr>
<td>grouptim.sp</td>
<td>group time delay example</td>
</tr>
<tr>
<td>inv.sp</td>
<td>sweep MOSFET -3 sigma to +3 sigma, use .MEASURE output</td>
</tr>
<tr>
<td>mcdiff.sp</td>
<td>CMOS differential amplifier</td>
</tr>
</tbody>
</table>
## Running Demonstration Files

### Examining the Demonstration Input Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mondc_a.sp</td>
<td>Monte Carlo of MOS diffusion and photolithographic effects</td>
</tr>
<tr>
<td>mondc_b.sp</td>
<td>Monte Carlo DC analysis</td>
</tr>
<tr>
<td>mont1.sp</td>
<td>Monte Carlo Gaussian, uniform, and limit function</td>
</tr>
<tr>
<td>mos2bit.sp</td>
<td>two-bit MOS adder</td>
</tr>
<tr>
<td>pll.sp</td>
<td>phase locked loop</td>
</tr>
<tr>
<td>sclopass.sp</td>
<td>switched capacitor low-pass filter</td>
</tr>
<tr>
<td>worst.sp</td>
<td>worst case skew models using .ALTER</td>
</tr>
<tr>
<td>xbjt2bit.sp</td>
<td>BJT NAND gate two-bit binary adder</td>
</tr>
</tbody>
</table>

### Behavioral Applications

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl.sp</td>
<td>acl gate</td>
</tr>
<tr>
<td>amp_mod.sp</td>
<td>amplitude modulator with pulse waveform carrier</td>
</tr>
<tr>
<td>behave.sp</td>
<td>AND/NAND gates using G, E Elements</td>
</tr>
<tr>
<td>calg2.sp</td>
<td>voltage variable capacitance</td>
</tr>
<tr>
<td>det_dff.sp</td>
<td>double edge triggered flip-flop</td>
</tr>
<tr>
<td>diff.sp</td>
<td>differentiator circuit</td>
</tr>
<tr>
<td>diode.sp</td>
<td>behavioral diode using a PWL VCCS</td>
</tr>
<tr>
<td>dlatch.sp</td>
<td>CMOS D-latch using behaviorals</td>
</tr>
<tr>
<td>galg1.sp</td>
<td>sampling a sine wave</td>
</tr>
<tr>
<td>idealop.sp</td>
<td>ninth-order low-pass filter</td>
</tr>
<tr>
<td>integ.sp</td>
<td>integrator circuit</td>
</tr>
<tr>
<td>invb_op.sp</td>
<td>optimization of CMOS macromodel inverter</td>
</tr>
<tr>
<td>ivx.sp</td>
<td>characterization of PMOS and NMOS as a switch</td>
</tr>
<tr>
<td>op_amp.sp</td>
<td>op-amp from Chua and Lin</td>
</tr>
<tr>
<td>pd.sp</td>
<td>phase detector modeled by switches</td>
</tr>
<tr>
<td>pdb.sp</td>
<td>phase detector using behavioral NAND gates</td>
</tr>
<tr>
<td>pwl10.sp</td>
<td>operational amplifier used as voltage follower</td>
</tr>
</tbody>
</table>

---

*Star-Hspice Manual, Release 2001.2*
### Examining the Demonstration Input Files

Running Demonstration Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pwl2.sp</td>
<td>PPW-VCCS with gain of 1 amp/volt</td>
</tr>
<tr>
<td>pwl4.sp</td>
<td>eight-input NAND gate</td>
</tr>
<tr>
<td>pwl7.sp</td>
<td>modeling inverter by a PWL VCVS</td>
</tr>
<tr>
<td>pwl8.sp</td>
<td>smoothing the triangle waveform by PWL CCCS</td>
</tr>
<tr>
<td>ring5bm.sp</td>
<td>five-stage ring oscillator – macromodel CMOS inverter</td>
</tr>
<tr>
<td>ringb.sp</td>
<td>ring oscillator using behavioral model</td>
</tr>
<tr>
<td>sampling.sp</td>
<td>sampling a sine wave</td>
</tr>
<tr>
<td>scr.sp</td>
<td>silicon controlled rectifier modelled with PWL CCVS</td>
</tr>
<tr>
<td>swcap5.sp</td>
<td>fifth-order elliptic switched capacitor filter</td>
</tr>
<tr>
<td>switch.sp</td>
<td>test for PWL switch element</td>
</tr>
<tr>
<td>swrc.sp</td>
<td>switched capacitor RC circuit</td>
</tr>
<tr>
<td>triode.sp</td>
<td>triode model family of curves using behavioral elements</td>
</tr>
<tr>
<td>triodex.sp</td>
<td>triode model family of curves using behavioral elements</td>
</tr>
<tr>
<td>tunnel.sp</td>
<td>modeling tunnel diode characteristic by PWL VCCS</td>
</tr>
<tr>
<td>vcob.sp</td>
<td>voltage controlled oscillator using PWL functions</td>
</tr>
</tbody>
</table>

#### Benchmarks

```bash
$installdir/demo/hspice/bench
```

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigmos1.sp</td>
<td>large MOS simulation</td>
</tr>
<tr>
<td>demo.sp</td>
<td>quick demo file to test installation</td>
</tr>
<tr>
<td>m2bit.sp</td>
<td>72-transistor two-bit adder – typical cell simulation</td>
</tr>
<tr>
<td>m2bitf.sp</td>
<td>fast simulation example</td>
</tr>
<tr>
<td>m2bitsw.sp</td>
<td>fast simulation example – same as m2bitf.sp but using behavioral elements</td>
</tr>
<tr>
<td>senseamp.sp</td>
<td>bipolar analog test case</td>
</tr>
</tbody>
</table>

#### Timing Analysis

```bash
$installdir/demo/hspice/bisect
```

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fig3a.sp</td>
<td>DFF bisection search for setup time</td>
</tr>
<tr>
<td>fig3b.sp</td>
<td>DFF early, optimum, and late setup times</td>
</tr>
<tr>
<td>inv.a.sp</td>
<td>inverter bisection pass-fail</td>
</tr>
</tbody>
</table>
## Running Demonstration Files

### Examining the Demonstration Input Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BJT and Diode Devices</strong></td>
<td>$\text{installdir/demo/hspice/bjt}$</td>
</tr>
<tr>
<td>bjtbeta.sp</td>
<td>plot BJT beta</td>
</tr>
<tr>
<td>bjtft.sp</td>
<td>plot BJT FT using s-parameters</td>
</tr>
<tr>
<td>bjtgm.sp</td>
<td>plot BJT Gm, Gpi</td>
</tr>
<tr>
<td>dpntun.sp</td>
<td>junction tunnel diode</td>
</tr>
<tr>
<td>snap.sp</td>
<td>convert SNAP to Star-Hspice</td>
</tr>
<tr>
<td>tun.sp</td>
<td>tunnel oxide diode</td>
</tr>
<tr>
<td><strong>Cell Characterization</strong></td>
<td>$\text{installdir/demo/hspice/cchar}$</td>
</tr>
<tr>
<td>dff.sp</td>
<td>DFF bisection search for setup time</td>
</tr>
<tr>
<td>inv3.sp</td>
<td>inverter characterization</td>
</tr>
<tr>
<td>inva.sp</td>
<td>inverter characterization</td>
</tr>
<tr>
<td>invb.sp</td>
<td>inverter characterization</td>
</tr>
<tr>
<td>load1.sp</td>
<td>inverter sweep, delay versus fanout</td>
</tr>
<tr>
<td>setupbsc.sp</td>
<td>setup characterization</td>
</tr>
<tr>
<td>setupold.sp</td>
<td>setup characterization</td>
</tr>
<tr>
<td>setuppas.sp</td>
<td>setup characterization</td>
</tr>
<tr>
<td>sigma.sp</td>
<td>sweep MOSFET -3 sigma to +3 sigma, use measure output</td>
</tr>
<tr>
<td>tdgtl.a2d</td>
<td>Viewsim A2D Star-Hspice input file</td>
</tr>
<tr>
<td>tdgtl.d2a</td>
<td>Viewsim D2A Star-Hspice input file</td>
</tr>
<tr>
<td>tdgtl.sp</td>
<td>two-bit adder using D2A Elements</td>
</tr>
<tr>
<td><strong>Circuit Optimization</strong></td>
<td>$\text{installdir/demo/hspice/ciropt}$</td>
</tr>
<tr>
<td>ampavl.sp</td>
<td>set unity gain frequency of BJT diff pair</td>
</tr>
<tr>
<td>ampopt.sp</td>
<td>optimize area, power, speed of MOS amp</td>
</tr>
<tr>
<td>asic2.sp</td>
<td>optimize speed, power of CMOS output buffer</td>
</tr>
<tr>
<td>asic6.sp</td>
<td>find best width of CMOS input buffer</td>
</tr>
<tr>
<td>delayopt.sp</td>
<td>optimize group delay of LCR circuit</td>
</tr>
</tbody>
</table>
### Examining the Demonstration Input Files

### Running Demonstration Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lpopt.sp</td>
<td>match lossy filter to ideal filter</td>
</tr>
<tr>
<td>opttemp.sp</td>
<td>find first and second temperature coefficients of resistor</td>
</tr>
<tr>
<td>rcopt.sp</td>
<td>optimize speed, power for RC circuit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DDL</th>
<th>$\text{installdir/demo/hspice/ddl}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ad8bit.sp</td>
<td>eight-bit A/D flash converter</td>
</tr>
<tr>
<td>alf155.sp</td>
<td>National JFET op-amp characterization</td>
</tr>
<tr>
<td>alf156.sp</td>
<td>National JFET op-amp characterization</td>
</tr>
<tr>
<td>alf157.sp</td>
<td>National JFET op-amp characterization</td>
</tr>
<tr>
<td>alf255.sp</td>
<td>National JFET op-amp characterization</td>
</tr>
<tr>
<td>alf347.sp</td>
<td>National JFET op-amp characterization</td>
</tr>
<tr>
<td>alf351.sp</td>
<td>National wide bandwidth JFET input op-amp char.</td>
</tr>
<tr>
<td>alf353.sp</td>
<td>National wide bandwidth dual JFET input op-amp char.</td>
</tr>
<tr>
<td>alf355.sp</td>
<td>Motorola JFET op-amp characterization</td>
</tr>
<tr>
<td>alf356.sp</td>
<td>Motorola JFET op-amp characterization</td>
</tr>
<tr>
<td>alf357.sp</td>
<td>Motorola JFET op-amp characterization</td>
</tr>
<tr>
<td>alf3741.sp</td>
<td></td>
</tr>
<tr>
<td>alm101a.sp</td>
<td></td>
</tr>
<tr>
<td>alm107.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm108.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm108a.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm118.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm124.sp</td>
<td>National low power quad op-amp characterization</td>
</tr>
<tr>
<td>alm124a.sp</td>
<td>National low power quad op-amp characterization</td>
</tr>
<tr>
<td>alm158.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm158a.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm201.sp</td>
<td>LM201 op-amp characterization</td>
</tr>
<tr>
<td>File Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>alm201a.sp</td>
<td>LM201 op-amp characterization</td>
</tr>
<tr>
<td>alm207.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm208.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm208a.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm224.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm258.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm258a.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm301a.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm307.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm308.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm308a.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm318.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm324.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm358.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm358a.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm725.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm741.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm747.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm747c.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm1458.sp</td>
<td>National dual op-amp characterization</td>
</tr>
<tr>
<td>alm1558.sp</td>
<td>National dual op-amp characterization</td>
</tr>
<tr>
<td>alm2902.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>alm2904.sp</td>
<td>National op-amp characterization</td>
</tr>
<tr>
<td>amc1458.sp</td>
<td>Motorola internally compensated high performance op-amp characterization</td>
</tr>
<tr>
<td>amc1536.sp</td>
<td>Motorola internally compensated high voltage op-amp characterization</td>
</tr>
<tr>
<td>amc1741.sp</td>
<td>Motorola internally compensated high performance op-amp characterization</td>
</tr>
<tr>
<td>File Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>amc1747.sp</td>
<td>Motorola internally compensated high performance op-amp characterization</td>
</tr>
<tr>
<td>ane5534.sp</td>
<td>TI low noise, high speed op-amp characterization</td>
</tr>
<tr>
<td>anjm4558.sp</td>
<td>TI dual op-amp characterization</td>
</tr>
<tr>
<td>anjm4559.sp</td>
<td>TI dual op-amp characterization</td>
</tr>
<tr>
<td>anjm4560.sp</td>
<td>TI dual op-amp characterization</td>
</tr>
<tr>
<td>aop04.sp</td>
<td>PMI op-amp characterization</td>
</tr>
<tr>
<td>aop07.sp</td>
<td>PMI ultra low offset voltage op-amp characterization</td>
</tr>
<tr>
<td>aop14.sp</td>
<td>PMI op-amp characterization</td>
</tr>
<tr>
<td>aop15b.sp</td>
<td>PMI precision JFET input op-amp characterization</td>
</tr>
<tr>
<td>aop16b.sp</td>
<td>PMI precision JFET input op-amp characterization</td>
</tr>
<tr>
<td>at094cns.sp</td>
<td>TI op-amp characterization</td>
</tr>
<tr>
<td>atl071c.sp</td>
<td>TI low noise op-amp characterization</td>
</tr>
<tr>
<td>atl072c.sp</td>
<td>TI low noise op-amp characterization</td>
</tr>
<tr>
<td>atl074c.sp</td>
<td>TI low noise op-amp characterization</td>
</tr>
<tr>
<td>atl081c.sp</td>
<td>TI JFET op-amp characterization</td>
</tr>
<tr>
<td>atl082c.sp</td>
<td>TI JFET op-amp characterization</td>
</tr>
<tr>
<td>atl084c.sp</td>
<td>TI JFET op-amp characterization</td>
</tr>
<tr>
<td>atl092cp.sp</td>
<td>TI op-amp characterization</td>
</tr>
<tr>
<td>atl094cn.sp</td>
<td>TI op-amp characterization</td>
</tr>
<tr>
<td>aupc358.sp</td>
<td>NEC general dual op-amp characterization</td>
</tr>
<tr>
<td>aupc1251.sp</td>
<td>NEC general dual op-amp characterization</td>
</tr>
<tr>
<td>j2n3330.sp</td>
<td>JFET 2n3330 I-V characteristics</td>
</tr>
<tr>
<td>mirf340.sp</td>
<td>IRF340 I-V characteristics</td>
</tr>
<tr>
<td>t2n2222.sp</td>
<td>BJT 2n2222 characterization</td>
</tr>
</tbody>
</table>

Device Optimization  $installdir/demo/hspice/devopt

| beta.sp    | LEVEL=2 beta optimization               |
Running Demonstration Files  

Examining the Demonstration Input Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bjtopt.sp</td>
<td>s-parameter optimization of 2n6604 BJT</td>
</tr>
<tr>
<td>bjtopt1.sp</td>
<td>2n2222 DC optimization</td>
</tr>
<tr>
<td>bjtopt2.sp</td>
<td>2n2222 Hfe optimization</td>
</tr>
<tr>
<td>d.sp</td>
<td>diode, multiple temperatures</td>
</tr>
<tr>
<td>dcopt1.sp</td>
<td>1n3019 diode I-V and C-V optimization</td>
</tr>
<tr>
<td>gaas.sp</td>
<td>JFET optimization</td>
</tr>
<tr>
<td>jopt.sp</td>
<td>300u/1u GaAs FET DC optimization</td>
</tr>
<tr>
<td>jopt2.sp</td>
<td>JFET optimization</td>
</tr>
<tr>
<td>joptac.sp</td>
<td>300u/1u GaAs FET 40 MHz–20 GHz s-parameter optimization</td>
</tr>
<tr>
<td>l3.sp</td>
<td>MOS LEVEL 3 optimization</td>
</tr>
<tr>
<td>l3a.sp</td>
<td>MOS LEVEL 3 optimization</td>
</tr>
<tr>
<td>l28.sp</td>
<td>LEVEL=28 optimization</td>
</tr>
<tr>
<td>ml2opt.sp</td>
<td>MOS LEVEL=2 I-V optimization</td>
</tr>
<tr>
<td>ml3opt.sp</td>
<td>MOS LEVEL=3 I-V optimization</td>
</tr>
<tr>
<td>ml6opt.sp</td>
<td>MOS LEVEL=6 I-V optimization</td>
</tr>
<tr>
<td>ml13opt.sp</td>
<td>MOS LEVEL=13 I-V optimization</td>
</tr>
<tr>
<td>opt_bjt.sp</td>
<td>2n3947 forward and reverse Gummel optimization</td>
</tr>
</tbody>
</table>

Fourier Analysis

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>am.sp</td>
<td>FFT analysis, AM source</td>
</tr>
<tr>
<td>bart.sp</td>
<td>FFT analysis, Bartlett window</td>
</tr>
<tr>
<td>black.sp</td>
<td>FFT analysis, Blackman window</td>
</tr>
<tr>
<td>dist.sp</td>
<td>FFT analysis, second harmonic distortion</td>
</tr>
<tr>
<td>exam1.sp</td>
<td>FFT analysis, AM source</td>
</tr>
<tr>
<td>exam3.sp</td>
<td>FFT analysis, high frequency signal detection test</td>
</tr>
<tr>
<td>exam4.sp</td>
<td>FFT analysis, small-signal harmonic distortion test</td>
</tr>
<tr>
<td>exp.sp</td>
<td>FFT analysis, exponential source</td>
</tr>
<tr>
<td>File Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>fft.sp</td>
<td>FFT analysis, transient, sweeping a resistor</td>
</tr>
<tr>
<td>fft1.sp</td>
<td>FFT analysis, transient</td>
</tr>
<tr>
<td>fft2.sp</td>
<td>FFT analysis on the product of three waveforms</td>
</tr>
<tr>
<td>fft3.sp</td>
<td>FFT analysis, transient, sweeping frequency</td>
</tr>
<tr>
<td>fft4.sp</td>
<td>FFT analysis, transient, Monte Carlo Gaussian distribution</td>
</tr>
<tr>
<td>fft5.sp</td>
<td>FFT analysis, data-driven transient analysis</td>
</tr>
<tr>
<td>fft6.sp</td>
<td>FFT analysis, sinusoidal source</td>
</tr>
<tr>
<td>gauss.sp</td>
<td>FFT analysis, Gaussian window</td>
</tr>
<tr>
<td>hann.sp</td>
<td>FFT analysis, Hamming window</td>
</tr>
<tr>
<td>harris.sp</td>
<td>FFT analysis, Blackman-Harris window</td>
</tr>
<tr>
<td>intermod.sp</td>
<td>FFT analysis, intermodulation distortion</td>
</tr>
<tr>
<td>kaiser.sp</td>
<td>FFT analysis, Kaiser window</td>
</tr>
<tr>
<td>mod.sp</td>
<td>FFT analysis, modulated pulse</td>
</tr>
<tr>
<td>pulse.sp</td>
<td>FFT analysis, pulse source</td>
</tr>
<tr>
<td>pw1.sp</td>
<td>FFT analysis, piecewise linear source</td>
</tr>
<tr>
<td>rect.sp</td>
<td>FFT analysis, rectangular window</td>
</tr>
<tr>
<td>rectan.sp</td>
<td>FFT analysis, rectangular window</td>
</tr>
<tr>
<td>sffm.sp</td>
<td>FFT analysis, single-frequency FM source</td>
</tr>
<tr>
<td>sine.sp</td>
<td>FFT analysis, sinusoidal source</td>
</tr>
<tr>
<td>swcap5.sp</td>
<td>FFT analysis, fifth-order elliptic switched capacitor filter</td>
</tr>
<tr>
<td>tri.sp</td>
<td>FFT analysis, rectangular window</td>
</tr>
<tr>
<td>win.sp</td>
<td>FFT analysis, window test</td>
</tr>
<tr>
<td>window.sp</td>
<td>FFT analysis, window test</td>
</tr>
<tr>
<td>winreal.sp</td>
<td>FFT analysis, window test</td>
</tr>
</tbody>
</table>

*Filters* 

$\text{installdir/demo/hspice/filters}$
### File Name Description

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fbp_1.sp</td>
<td>bandpass LCR filter measurement</td>
</tr>
<tr>
<td>fbp_2.sp</td>
<td>bandpass LCR filter pole/zero</td>
</tr>
<tr>
<td>fbpnet.sp</td>
<td>bandpass LCR filter s-parameters</td>
</tr>
<tr>
<td>fbprlc.sp</td>
<td>LCR AC analysis for resonance</td>
</tr>
<tr>
<td>fhp4th.sp</td>
<td>high-pass LCR fourth-order Butterworth filter</td>
</tr>
<tr>
<td>fkerwin.sp</td>
<td>pole/zero analysis of Kerwin’s circuit</td>
</tr>
<tr>
<td>flp5th.sp</td>
<td>low-pass fifth-order filter</td>
</tr>
<tr>
<td>flp9th.sp</td>
<td>low-pass ninth-order FNDR with ideal op-amps</td>
</tr>
<tr>
<td>micro1.sp</td>
<td>test of microstrip</td>
</tr>
<tr>
<td>micro2.sp</td>
<td>test of microstrip</td>
</tr>
<tr>
<td>tcoax.sp</td>
<td>test of RG58/AU coax</td>
</tr>
<tr>
<td>trans1m.sp</td>
<td>FR-4 printed circuit lumped transmission line</td>
</tr>
</tbody>
</table>

#### Magnetics

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aircore.sp</td>
<td>air core transformer circuit</td>
</tr>
<tr>
<td>bloop.sp</td>
<td>b-h loop nonlinear magnetic core transformer</td>
</tr>
<tr>
<td>mag2.sp</td>
<td>three primary, two secondary magnetic core transformer</td>
</tr>
<tr>
<td>magcore.sp</td>
<td>magnetic core transformer circuit</td>
</tr>
<tr>
<td>royerosc.sp</td>
<td>Royer magnetic core oscillator</td>
</tr>
</tbody>
</table>

#### MOSFET Devices

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bsim3.sp</td>
<td>LEVEL=47 BSIM3 model</td>
</tr>
<tr>
<td>cap13.sp</td>
<td>plot MOS capacitances LEVEL=13 model</td>
</tr>
<tr>
<td>cap_bp.sp</td>
<td>capacitances for LEVEL=13 model</td>
</tr>
<tr>
<td>cap_m.sp</td>
<td>capacitance for LEVEL=13 model</td>
</tr>
<tr>
<td>capop0.sp</td>
<td>plot MOS capacitances LEVEL=2</td>
</tr>
<tr>
<td>capop1.sp</td>
<td>plot MOS capacitances LEVEL=2</td>
</tr>
<tr>
<td>capop2.sp</td>
<td>plot MOS capacitances LEVEL=2</td>
</tr>
</tbody>
</table>
### File Name
<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plot MOS capacitances LEVEL=6</td>
</tr>
<tr>
<td>charge conservation test LEVEL=3</td>
</tr>
<tr>
<td>impact ionization current plot</td>
</tr>
<tr>
<td>plot temperature effects LEVEL=6</td>
</tr>
<tr>
<td>plot temperature effects LEVEL=13</td>
</tr>
<tr>
<td>s-parameters for LEVEL=13</td>
</tr>
<tr>
<td>plot I-V for LEVEL=13</td>
</tr>
<tr>
<td>plot I-V for LEVEL=27 SOSFET</td>
</tr>
<tr>
<td>plot I-V for user include file</td>
</tr>
<tr>
<td>plot I-V and C-V for LEVEL=3</td>
</tr>
<tr>
<td>charge conservation test LEVEL=6</td>
</tr>
<tr>
<td>charge conservation test LEVEL=6</td>
</tr>
<tr>
<td>automatic width and length model selector</td>
</tr>
<tr>
<td>LEVEL=6 gamma model</td>
</tr>
<tr>
<td>MOS LEVEL=34 EPFL, test DC</td>
</tr>
</tbody>
</table>

### Peripheral Component Interconnect

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Peripheral Component Interconnect demonstration</td>
</tr>
<tr>
<td>PCI Monte Carlo example</td>
</tr>
<tr>
<td>PCI worst-case modeling</td>
</tr>
</tbody>
</table>

### Radiation Effects

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bipolar radiation effects example</td>
</tr>
<tr>
<td>bipolar radiation effects example</td>
</tr>
<tr>
<td>bipolar radiation effects example</td>
</tr>
<tr>
<td>bipolar radiation effects example</td>
</tr>
<tr>
<td>bipolar radiation effects example</td>
</tr>
<tr>
<td>bipolar radiation effects example</td>
</tr>
<tr>
<td>File Name</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>drad1.sp</td>
</tr>
<tr>
<td>drad2.sp</td>
</tr>
<tr>
<td>drad4.sp</td>
</tr>
<tr>
<td>drad5.sp</td>
</tr>
<tr>
<td>drad6.sp</td>
</tr>
<tr>
<td>dradarb2.sp</td>
</tr>
<tr>
<td>jex1.sp</td>
</tr>
<tr>
<td>jex2.sp</td>
</tr>
<tr>
<td>jprad1.sp</td>
</tr>
<tr>
<td>jprad2.sp</td>
</tr>
<tr>
<td>jprad4.sp</td>
</tr>
<tr>
<td>jrad1.sp</td>
</tr>
<tr>
<td>jrad2.sp</td>
</tr>
<tr>
<td>jrad3.sp</td>
</tr>
<tr>
<td>jrad4.sp</td>
</tr>
<tr>
<td>jrad5.sp</td>
</tr>
<tr>
<td>jrad6.sp</td>
</tr>
<tr>
<td>mrad1.sp</td>
</tr>
<tr>
<td>mrad2.sp</td>
</tr>
<tr>
<td>mrad3.sp</td>
</tr>
<tr>
<td>mrad3p.sp</td>
</tr>
<tr>
<td>mrad3px.sp</td>
</tr>
<tr>
<td>rad1.sp</td>
</tr>
<tr>
<td>rad2.sp</td>
</tr>
<tr>
<td>rad3.sp</td>
</tr>
<tr>
<td>rad4.sp</td>
</tr>
</tbody>
</table>
### File Name

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rad5.sp</td>
<td>BJT photocurrent test circuit with an NPN transistor</td>
</tr>
<tr>
<td>rad6.sp</td>
<td>BJT secondary photocurrent effect which varies with R1</td>
</tr>
<tr>
<td>rad7.sp</td>
<td>BJT RLEV=6 example (semi-empirical model)</td>
</tr>
<tr>
<td>rad8.sp</td>
<td>JFET RLEV=1 example with Wirth-Rogers square pulse</td>
</tr>
<tr>
<td>rad9.sp</td>
<td>JFET stepwise increasing radiation source</td>
</tr>
<tr>
<td>rad10.sp</td>
<td>GaAs RLEV=5 example (semi-empirical model)</td>
</tr>
<tr>
<td>rad11.sp</td>
<td>NMOS E-mode LEVEL=8 with Wirth-Rogers square pulse</td>
</tr>
<tr>
<td>rad12.sp</td>
<td>NMOS 0.5x resistive voltage divider</td>
</tr>
<tr>
<td>rad13.sp</td>
<td>three-input NMOS NAND gate with non-EPI, EPI, and SOS examples</td>
</tr>
<tr>
<td>rad14.sp</td>
<td>GaAs differential amplifier circuit</td>
</tr>
<tr>
<td>rad14dc.sp</td>
<td>n-channel JFET DC I-V curves</td>
</tr>
</tbody>
</table>

### Sources

<table>
<thead>
<tr>
<th>Sources</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>amsrc.sp</td>
<td>amplitude modulation</td>
</tr>
<tr>
<td>exp.sp</td>
<td>exponential independent source</td>
</tr>
<tr>
<td>pulse.sp</td>
<td>test of pulse</td>
</tr>
<tr>
<td>pw1.sp</td>
<td>repeated piecewise linear source</td>
</tr>
<tr>
<td>pw110.sp</td>
<td>op-amp voltage follower</td>
</tr>
<tr>
<td>rtest.sp</td>
<td>voltage controlled resistor inverter chain</td>
</tr>
<tr>
<td>sffm.sp</td>
<td>single frequency FM modulation source</td>
</tr>
<tr>
<td>sin.sp</td>
<td>sinusoidal source waveform</td>
</tr>
<tr>
<td>vcr1.sp</td>
<td>switched capacitor network using G-switch</td>
</tr>
</tbody>
</table>

### Transmission Lines

<table>
<thead>
<tr>
<th>Transmission Lines</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fr4.sp</td>
<td>microstrip test FR-4 PC board material</td>
</tr>
<tr>
<td>fr4o.sp</td>
<td>optimizing model for microstrip FR-4 PC board material</td>
</tr>
<tr>
<td>fr4x.sp</td>
<td>FR4 microstrip test</td>
</tr>
<tr>
<td>hd.sp</td>
<td>ground bounce for I/O CMOS driver</td>
</tr>
</tbody>
</table>
## Running Demonstration Files

### Examining the Demonstration Input Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rcsnubts.sp</td>
<td>ground bounce for I/O CMOS driver at snubber output</td>
</tr>
<tr>
<td>rcsnubtt.sp</td>
<td>ground bounce for I/O CMOS driver</td>
</tr>
<tr>
<td>strip1.sp</td>
<td>two series microstrips (8 mil and 16 mil wide)</td>
</tr>
<tr>
<td>strip2.sp</td>
<td>two microstrips coupled together</td>
</tr>
<tr>
<td>t14p.sp</td>
<td>1400 mil by 140 mil, 50 ohm tline on FR-4 50 MHz - 10.05 GHz</td>
</tr>
<tr>
<td>t14xx.sp</td>
<td>1400 mil by 140 mil, 50 ohm tline on FR-4 optimization</td>
</tr>
<tr>
<td>t1400.sp</td>
<td>1400 mil by 140 mil, 50 ohm tline on FR-4 optimization</td>
</tr>
<tr>
<td>tcoax.sp</td>
<td>RG58/AU coax with 50 ohm termination</td>
</tr>
<tr>
<td>tfr4.sp</td>
<td>microstrip test</td>
</tr>
<tr>
<td>tfr4o.sp</td>
<td>microstrip test</td>
</tr>
<tr>
<td>tl.sp</td>
<td>series source coupled and shunt terminated transmission lines</td>
</tr>
<tr>
<td>transmis.sp</td>
<td>algebraics and lumped transmission lines</td>
</tr>
<tr>
<td>twin2.sp</td>
<td>twinlead model</td>
</tr>
<tr>
<td>xfr4.sp</td>
<td>microstrip test subcircuit expanded</td>
</tr>
<tr>
<td>xfr4a.sp</td>
<td>microstrip test subcircuit expanded, larger ground resistance</td>
</tr>
<tr>
<td>xfr4b.sp</td>
<td>microstrip test</td>
</tr>
<tr>
<td>xulump.sp</td>
<td>test 5-, 20-, and 100-lump U models</td>
</tr>
</tbody>
</table>
Appendix A
FAQ/Troubleshooting

This appendix contains frequently asked questions (FAQ) and their solutions for the following topics:

- Analysis
- Documentation
- Environment Variables
- Error Messages
- Input
- Installation Issues
- Licensing/Access Issues
- Limitations
- Miscellaneous
- Models
- MS Windows/PC Issues
- Netlist/Options
- Output
- W Element/Field Solver
- Waveform Viewing

Solution numbers at the end of each question and answer refer to the Avant! internal bug system solution number.
Analysis

Q: How do I make .TRAN and .TRAN SWEEP results agree?

A: If you get different results from transient simulations than when doing a parameterized transient sweep (and you are using transmission line elements), you should explicitly set the option RISTIME. For example:

```plaintext
.OPTION risetime=<rise time of fastest component being simulated>
```

Solution: 28

Q: Why do I get “Internal timestep too small” when I use UIC in the .TRAN statement?

A: If the UIC is used in the .TRAN statement, the DC operating point analysis is bypassed and a transient analysis is started using node voltages specified in an .IC statement.

During a simulation, DC operating point analysis is an important step. Most of the DC analysis algorithms, control options, initializations, and convergence also apply to transient analysis. As a result, although a transient analysis might provide a convergent DC solution, the transient analysis itself can still fail to converge if UIC is used. In transient analysis, the error message “internal timestep too small” indicates that the circuit failed to converge. The convergence failure might be due to stated initial conditions that are not close enough to the actual DC operating point values.

Therefore, the best solution is to comment out the UIC from the .TRAN statement and rerun the simulation. At that point, the DC operating point analysis is performed and the correct values are used in the transient analysis.

Solution: 383
Q: Is there a way to change the seed that pseudo-randomly generates the initial conditions in the Monte Carlo simulation?

A: Yes, you can set .OPTION seed=<value> where value is between 1 and 259200. This changes the random number generator starting seed.

Solution: 408

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
**Q:** Whom should I contact regarding Star-Hspice manuals and other documents?

**A:** To purchase the latest *Star-Hspice Manual*, call Ana Wagem at (510) 413-8383 or email:

ana_wageman@avanticorp.com

**Solution:** 41

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

**Q:** Are the demo files in the *Star-Hspice Manual* (versions 99.2 & 99.4) available anywhere on-line?

**A:** No, they are not on-line. However, they are in the directory where you install the Star-Hspice and AvanWaves executables:

1. From the directory where you install Star-Hspice (for example: `/hspice`), point to:

   `/hspice/99.4/demo/`

2. Two sub-directories appear:

   - `/hspice` directory, which contains several sub-directories, such as `alge`, `apps`, `bench`, `bisect`, `bjt`, etc. In each sub-directory, there are several demo netlists relative to the title of that sub-directory.
   - `/awaves` directory contains a sub-directory, named “tutorial” which contains AC, DC, and TRAN sub-directories. Each sub-directory contains one demo file relative to its title.

**Solution:** 376

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

**Q:** How do I get a copy of the *Star-Hspice Manual*?

**A:** First, check the Star-Hspice installation directory for a docs folder. If the manual isn’t there, check the Star-Hspice CD for the documents.
FAQ/Troubleshooting

For UNIX, they should be in a 200*.hspice_docs.tar.gz file, or something similar.

For Windows, rerun the installation. One of the options during the installation is to choose what items to install (Star-Hspice, AvanWaves, and documentation).

If none of these options work, send an email to:

hspice_nw@avanticorp.com

An ftp account can be set up to download the electronic documents.

For hardcopies, email:

hspice_nw@avanticorp.com

for the necessary contact information.

Solution: 460

Q: Are any demo files included with the PC version of the software?
A: Yes, during installation you have the option to install demo files.

Solution: 503
Environment Variables

Q: What does the environment variable META_QUEUE do?

A: If META_QUEUE is enabled, it allows licenses to be queued. For example, a customer has five Star-Hspice floating licenses; if all five licenses are checked out and META_QUEUE is enabled, then the next job submitted waits in queue until a license is available. If META_QUEUE is not enabled in this situation, you simply get an error message that no licenses are available.

**Note:** If you have more than one Star-Hspice token (INCREMENT line) and the version dates are different, only the first token in your license file is queued. FLEXlm queues for the first INCREMENT line that satisfies the request. If you have two INCREMENT lines with different versions, there are two license pools created on the server. When you issue the queuing request, the server tries to satisfy the request, and, if not possible, queues for the first INCREMENT line that would satisfy the request. Once it queues for that particular INCREMENT line, the server waits for it to become free. It does not continually look for any other line that might satisfy this request. This is normal operation for FLEXlm.

**Solution:** 367
FAQ/Troubleshooting

Error Messages

Q: I am getting “inconsistent license key” error messages when starting up lmgrd. Why? What should I do?

A: This happens when the encryption code that avantd calculates is different from the encryption key used to generate the license key. The following pieces of information from the license key are used: the host id of the server, the software expiration date, the increment name, and the software version.

If the user changes any of these fields, you will get an inconsistent license key error message. This can also happen when you are using different versions of avantd and lmgrd; however, it is very unlikely that this will cause a problem.

Avant! does not recommend using lmreread to start a new license. Shut the license server down using:

```
lmdown -c /path/to/license/file
```

and then start the new license server in the normal manner.

Solution: 134

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Q: I am getting a TCL error: “Cannot find config.status” when running the $installdir/bin/config script to configure Star-Hspice/AvanWaves. Why?

A: Your installation is probably fine. No matter what products you choose to install, the configuration script tries to display the config.status file. Unfortunately, not all installation options generate a “config.status” file.

You can verify that the configuration was successful by reading the automatically generated logfile. This is placed in /tmp/avanti_####/config.log, where #### are numbers specific to the date and time of the installation.

Solution: 137

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Q: I am getting the following error message:

```
** error**: iob_loads2:9612: Number of input voltage sources in IOB = 0
```

What does this mean?

A: Certain connections in IBIS elements must be connected to ideal sources (e.g., output buffers must have their input node connected to an ideal source). For more information on which nodes must be connected, please see Chapter 19, “Using IBIS Models”, in the *Star-Hspice Manual*.

**Solution:** 145

Q: Why is the simulate button grayed out in HSPUI, even if a simulation is not being performed?

A: This is usually caused by a computer crash during a simulation. To ungray the simulate button, type in a command prompt:

```
"del %tmp%\job.run"
```

Alternatively, you search for the file *job.run*, and delete it.

**Solution:** 293

Q: Why do I get the error “inductor/voltage source loop”?

A: This is normally caused by:

1. A voltage source with the inductor connected to the same point.
2. A voltage source with an inductor connected directly across its nodes.
3. Two or more inductors connected in a loop with nothing to limit the current.

The best way to solve this type of error is to connect a small series resistance (1 nano ohm or smaller usually works).

**Solution:** 464
Q: I get the error message: “Time: command terminated abnormally.” Why?

A: This means Star-Hspice stops unexpectedly, which can be due to a number of problems (such as convergence problems or a numerical overflow). Looking at the *.lis and *.st0 files can usually give clues to what happened. Also, try to find out if the simulation concluded using a different version of Star-Hspice or on other platforms. If it did, it could be a bug. Otherwise, it is most likely a problem with the designs or model cards.

The message, “Time: command terminated abnormally” is output by the UNIX utility TIME, not Star-Hspice itself. The TIME utility reports various system use information on the processes it monitors. When a process being monitored stops unexpectedly, TIME issues this warning message.

Solution: 520

Q: Why am I getting the following error in my license.log file when I try to start the avantd daemon:

ATTENTION ... “avantd” vendor daemon CAN NOT co-exist with monarc lockfile * error (-1) with “monarc”, program aborted... Please correct problem and restart daemons

A: This is caused by an earlier Avanti vendor daemon. Look in the /usr/tmp directory for a file named either lockmonarcd or lock.monarcd. Delete the lockfile and restart lmgrd. The avantd daemon will now be able to start.

Solution: 528

Q: Why do I get a Star-Hspice and AvanWaves “configure unsuccessful” error?

A: If $installdir is previously set, when you install a new version of Star-Hspice the path for the configuration will not be correct (it will try to install over your previous installation). This causes a “configure unsuccessful” error. To fix this, please set the correct path to your install directory in the configure utility.

Solution: 561
Q: Why do I get the error message: "**error**: system file descriptors may be used up"?

A: This is an error reported by your operating system. It happens when Star-Hspice tries to open more files than is permitted. If you have a large number of .ALTER statements or if you read in lots of files (.LIB or .INCLUDE), you will have to raise your system’s limit.

With sh/ksh/bash, you can use the “ulimit” built-in to display or change the limits for the current shell and all the processes you launch from this shell. Use:

```
$ ulimit -n
```

to display the current limit. For example, on my system, I get 64; this means that any process I launch from this shell cannot open more than 64 files at once. To raise this limit to 256, for instance, type:

```
$ ulimit -n 256
```

There is a “hard” limit that you cannot exceed (although the root can change the hard limit). To see what the hard limit is on your system, type:

```
$ ulimit -Hn
```

On csh/tcsh, use the “limit” built-in do this (type `limit` to see a list of limits). For example, on my system:

```
cputime unlimited
filesize unlimited
datasize 2097148 kbytes
stacksize 8192 kbytes
coredumpsize unlimited
descriptors 64
memorysize unlimited
```
To increase the number of open files (file descriptors) to 256, type:

   $ limit descriptors 256

To view the hard limits, type:

   $ limit -h

**Solution:** 613

********************************************************************************

**Q:** Why do I get this message: “**error** reference 0:nch not found” even when NCH appears in my library?

**A:** Star-Hspice has a feature called “automatic model selection,” which is probably used in your library. You can identify models that use this feature if the model name is followed by a period and a suffix, as in:

   .MODEL nch.1 nmos ...

It is likely that you have multiple .model nch.* statements that define the model’s parameters for different sizes of the transistor. These models use LMIN, LMAX, WMIN and WMAX to determine which model applies to a given transistor dimensions.

For instance, if model nch.1 includes LMIN=0.15u LMAX=0.25u WMIN=0.15u WMAX=1.0u, then the model is only applied to transistors with lengths between 0.15um and 0.25um and widths between 0.15um and 1.0um.

If you are getting a “reference not found” error, then your transistor dimensions (L and W) do not fall in the ranges specified by any of the models in your library.

**Solution:** 642

********************************************************************************

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Q: Why do I get a message: “input file contains no data”?
A: This occurs with either no .END or no return after the .END statement in the netlist. Ensure the netlist has an .END as the last statement, and do not forget to insert a return.

Solution: 44

Q: Why does an I/O Buffer used as an input have a much smaller input current than an input buffer?
A: The power clamp and ground clamp nodes need to be included in the calling card when using an I/O buffer as an input buffer.

Solution: 455
Installation Issues

Q: Whom should I contact regarding Star-Hspice installation?
A: To contact the Star-Hspice technical support team, submit a question to:

    hspice_nw@avanticorp.com

    To ensure Avant! has a thorough understanding of your question, please provide all pertinent details.

Solution: 41

Q: I am having trouble installing Star-Hspice on my Windows NT machine. Do you have any suggestions?
A: A common solution is setting Windows NT environment variables ($installdir, LM_LICENSE_FILE and PATH) in ControlPanel->System; these are not set by default.

Solution: 56

Q: I recently upgraded or installed a newer (post-97.4) Star-Hspice version; however, now I cannot simulate out of Analog Artist from Cadence. It is issuing an error message about not being able to find “permit.hsp.” Or perhaps Analog Artist is spawning the incorrect version of Star-Hspice. What do I do?
A: Most installations of Cadence Composer/Analog Artist create a script that is run to bring up the framework. Within this script, they are setting the $installdir environment variable. Edit this script and point the $installdir variable to the new Star-Hspice installation directory (the directory that contains the “bin” directory, which in turn contains the Star-Hspice and AvanWaves scripts).

Solution: 136
Q: I am having problems installing my new node-locked Star-Hspice license. The error I am receiving is: “invalid hostid”. Star-Hspice is reporting a hostid of 0, but I have the dongle installed! What should I do?

A: Node-locked versions of Star-Hspice on the PC require the use of a parallel port dongle. Under WinNT, you must install a driver to be able to use the dongle. The installation program is “setupx86.exe” in the %installdir% directory. Under Win98/95, it is not necessary to run the “setupx86.exe” sentinel driver.

Solution: 146

*******************************************************************************
Licensing/Access Issues

Q: Whom should I contact regarding Star-Hspice license files?
A: To request a new license file, send email to:

    license@avanticorp.com

    To ensure Avant! has a thorough understanding of your question, please provide all pertinent details.

Solution: 41

Q: My older FLEX license does not work with my new Star-Hspice. What should I do?
A: FLEXlm v5.0 needs a 4-digit year in the expiration date; you should get a new license.

Solution: 53

Q: How can I run older Star-Hspice versions without having to source the “cshrc.meta” file of the older version?
A: You have to edit the “versions” file in the latest version of Star-Hspice, e.g., ..installldir../99.2/bin/versions, and include the paths of the older versions in the “versions” file. Once that is set, simply type:

    hspice

    at the command line. After you enter the input and output file names, you are prompted for a version to run. Select the number (1,2,3...etc) that represents the version you would like to run.

    On PC’s, modify the “..installldir\99.2\versions” file in the same manner and then you can select the version you would like to run by modifying the last line on the HSPUI window.

Solution: 57

--------------------------------

Q: I am getting "inconsistent license key" error messages when starting up lmgrd.
A: See this solution in “Error Messages” on page 7.
Solution: 134

Q: I am getting a TCL error: “Cannot find config.status” when running the $installdir/bin/config script to configure Star-Hspice/AvanWaves.
A: See this solution in “Error Messages” on page 7.
Solution: 137

Q: How can I reserve licenses for a particular user, group, or host?
A: Flexlm allows you to reserve licenses using a configuration file. The configuration file is a plain ASCII file containing comments and statements. Lines starting with a # are considered comments. You may create groups of users using the statement:

```
GROUP <groupname> <user> [...] 
```

You may reserve a specific license to either a user, a group, a host or a display using one of:

```
RESERVE <num_to_reserve> <feature> USER <username>
RESERVE <num_to_reserve> <feature> GROUP <groupname>
RESERVE <num_to_reserve> <feature> HOST <hostname>
RESERVE <num_to_reserve> <feature> DISPLAY <displayname>
```

Reserving a license for a specific HOST does not prevent users from setting their DISPLAY environment variable to their local DISPLAY and running the tool. In fact, reserving a license for a HOST is very similar to using a node-locked license.

Reserving a license for a DISPLAY ensures that the DISPLAY environment variable is set to what you have specified.

Solution: 654
Limitations

Q: What are the known limitations of Metaencryption?

A: The known limitations are:

■ The filenames of the encrypted files must have the same name as the subcircuit definition they contain (with a .inc extension).

■ The encrypted files must not be used with a .LIB or .INCLUDE in the netlist. Instead, they must be included through the automatic search path (.OPTION search=...) by referencing the subckt name in a subckt instance.

■ No more than 80 characters are allowed in a line; semi colons (;) are not allowed.

■ For Star-Hspice 1998.4 and earlier versions, dummy models should be inserted before using the LEVEL 49 model.

■ For every .PROTECT you should have an .UNPROTECT statement.

■ It is always good to have .PROTECT and .UNPROTECT in the following order:

<table>
<thead>
<tr>
<th>Recommended</th>
<th>Not Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>.SUBCKT</td>
<td>.SUBCKT</td>
</tr>
<tr>
<td>.PROTECT</td>
<td>.PROTECT</td>
</tr>
<tr>
<td>~~~</td>
<td>~~~</td>
</tr>
<tr>
<td>.UNPROTECT</td>
<td>.ENDS</td>
</tr>
<tr>
<td>.ENDS</td>
<td>.UNPROTECT</td>
</tr>
</tbody>
</table>

Solution: 40

*************************************************************************************************
**Miscellaneous**

Q: Should I include the gate edge of the source and drain when specifying the PD and PS (periphery of the drain and source respectively, used for calculating parasitic diodes)?

A: For ACM=0 or 2, you do include the gate edge of the source and drain when calculating PS and PD. For ACM=3, you do not include the source or drain edges when specifying PS and PD. PS and PD are not used for ACM=1.

**Solution:** 133

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Q: Is there a difference between parameters defined on the .SUBCKT line and those defined in the subcircuit body?

A: Parameters defined on the .subckt line are in the same namespace as parameters defined within the body of the subcircuit.

The following two subcircuits are equivalent:

```
.SUBCKT sub1 in out l=10u w=5u
.ENDS
```

Or:

```
.SUBCKT sub2 in out
.PARAM l=10u w=5u
.ENDS
```

**Note:** Both forms of the subcircuit accept the parameters “l” and “w” on element instance lines.

**Solution:** 139

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Q: How can I simplify repetitive complex measurements?
A: Encapsulate the functionality of your measurements/probes in subcircuits. Subcircuits do not have to contain elements. They can contain complex measurement routines to make measurements less tedious. If you make judicious choices for the instance names, parsing the output becomes relatively easy. For example, calculate AC transfer functions for arbitrary nodes in your netlist:

```
.PARAM mag(a,b) = 'sqrt(a*a+b*b)'
.SUBCKT Xfer in out
.PROBE AC xfer
par('mag(vr(in)*vr(out)+vi(in)*vi(out),vi(in)*vr(out)-vi(out)*vr(in))/mag(vr(out),vi(out))')
*xfer = mag(in/out) (note the voltages at in and out are complex)
.ENDS
```

Solution: 142

*******************************************************************************
Models FAQ/Troubleshooting

Models

Q: Why doesn’t .OPTION SCALE work in my simulation?

A: Check to see if your MOSFET and model card are wrapped inside a .SUBCKT call. For example:

```
.SUBCKT pmos d g s b w=1 l=1 m=1 mp d g s b p w=w l=l m=m
.MODEL p pmos LEVEL=.........
.ENDS
```

If so, check the model parameters for any dependencies on width or length. These parameters, in this case, are not scaled. .OPTION SCALE only affects width and length at the time they’re passed into the model card as the physical length and width. The model parameter is using width and length before they get scaled.

Solution: 209

******************************************************************************

Q: What is the process to get proprietary MOSFET models?

A: The following MOSFET model level numbers are vendor proprietary and require a special license to use:

- **AMD** LEVEL 32
- **Dallas Semi** DALLAS LEVEL 19
- **Hitachi** HIT LEVEL 43
- **IBM** IBM LEVEL 48
- **Lucent** ATT LEVEL 45
- **Motorola** MOT LEVEL 31
- **MOTSSIM** LEVEL 29
- **National** NAT LEVEL 33
- **SGS Thomson** SGSTHOM LEVEL 46
As of July 12, 2000, these are the contact people:

<table>
<thead>
<tr>
<th>Company</th>
<th>Contact Person</th>
<th>Phone Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD</td>
<td>Barbara Vervenne</td>
<td>(512) 602-5783</td>
</tr>
<tr>
<td>Dallas Semi</td>
<td>Roy Hensley</td>
<td>(214) 450-3858</td>
</tr>
<tr>
<td>Hitachi</td>
<td>Len Mizrah</td>
<td>(408) 456-2081</td>
</tr>
<tr>
<td>IBM</td>
<td>Cal Bittner</td>
<td>(902) 769-6528</td>
</tr>
<tr>
<td>Lucent</td>
<td>Roberta Kulp</td>
<td>(610) 712-2614</td>
</tr>
<tr>
<td>Motorola</td>
<td>Ralph Sokel</td>
<td>(512) 933-5264</td>
</tr>
<tr>
<td>National</td>
<td>Chen Young Tao</td>
<td>(408) 721-5665</td>
</tr>
<tr>
<td>SGS Thomson</td>
<td>Claude Frank</td>
<td>33 476 92 63 47 (Tel)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33 476 08 96 52 (Fax)</td>
</tr>
<tr>
<td>Sharp</td>
<td>Sigenobu Kiyoi</td>
<td>81 7 4365 4273</td>
</tr>
<tr>
<td>TI</td>
<td>Jeff Brunson</td>
<td>(972) 480-2481</td>
</tr>
<tr>
<td>Vitesse</td>
<td>Aubrey Grey</td>
<td>(805) 388-7517</td>
</tr>
</tbody>
</table>

For a customer to receive a license from Avant! (the vendors do not cut the licenses), he/she must contact the vendor and request that an authorization letter is submitted to Avant! Once Avant! receives the letter, a license file can be processed.

Solution: 263

***********************************************************************************
Q: Why doesn’t BJT LV5 - FT agree with net analysis results?

A: FT is a simple estimate for a cut-off frequency of the internal transistor:

\[ ft = \frac{\text{abs}(g_m-g_o)}{(2\pi \times \text{max}(1.0d-18, C_{be}+C_{bc}+C_{bx}))} \]

The parameters for the equation are from the equivalent circuit for BJT (see Figure 16-5: Vertical Transistor, BJT AC Analysis on page 16-24).

FT does not include parasitic elements (Rb, Rc, Re for example), and does not include the external circuit. It just gives a simple estimate for cut-off frequency of the internal transistor and it cannot replace results of actual simulation.


**Solution:**

Q: What is the importance of the following excerpt from the attached .lis file:

```
*****************************************************************
**warning BSIM3v3 VERSION parameter does not appear in the model card!!!
Failure to explicitly set VERSION may result in inaccurate results!
*****************************************************************
```

A: Avant! implemented the BSIM3v3 models released from UC Berkeley in the forms of LEVEL 49 and LEVEL 53. Since implementation, UC Berkeley has released patches and Avant! has implemented them as 3.1 (default for 98.2), 3.2 (default for 98.2 and 98.4) and 3.22 (default for 99.2, 99.4 and 2000.2).

If your models were developed for the VERSION = 3.1 release and this parameter is not present in your models/libraries, then Star-Hspice defaults to 3.22 (in Star-Hspice 2000.2 release), which may give you unexpected results.

**Solution:**
MS Windows/PC Issues

Q: My Star-Hspice license will not work on my PC. Why?
A: Problems often occur on PC’s with truncated lines in license files; repair the license or contact the Star-Hspice technical support team by submitting a question to:

   hspice_nw@avanticorp.com

   To ensure Avant! has a thorough understanding of your question, please provide all pertinent details.

Solution: 55

********************************************************************************

Q: I am having trouble installing Star-Hspice on my Windows NT machine.
A: Windows NT environment variables (installdir, LM_LICENSE_FILE and PATH) need to be set in ControlPanel->System; these are not set by default.

Solution: 56

********************************************************************************

Q: How can I run older Star-Hspice versions without having to source the “cshrc.meta” file of the older version?
A: See this solution in “Licensing/Access Issues” on page 15.

Solution: 57

********************************************************************************

Q: Why do I get the message, “Bad date found in the running machine,” when starting Star-Hspice on Win NT?
A: To get a license to check out, make sure the dates on the Windows NT machine are consistent with the current time (i.e., no future creation or modification times). The files with bad dates must be changed.
However, Windows NT does not come with a utility that can easily change file dates. To change these dates requires a command such as the UNIX “touch” command which Windows NT does not have. There is a third party software package that does have this “touch” command. “UnixDos Toolkit 4.3d” has numerous UNIX commands for the Window NT environment. It can be downloaded from software sites. After installation of this utility, one can “touch” the files and change their dates to the current date. However “touch” does not work with folders. To update the folder’s dates, just copy the folder’s contents to a new folder (example: folder xxx to folder xxx1), delete the old folder, and rename the folder to the original name (back to xxx).

Star-Hspice works when the files and directories in the “WINNT” folder are changed to the current date. Other files, not in the “WINNT” folder, with bad (future) dates do not cause any problems. An example of this is the MS Office files and folders. They can have bad dates but they do not affect the ability of Star-Hspice to run.

Solution: 341

Q: In Windows, the path to license.dat is correct, but still Star-Hspice cannot find it. What do I do?

A: If you use Windows Explorer, it will, by default, not show the extension of a file. So if, by mistake, your license file has an additional extension besides .dat, then you will see it in the Explorer view, but Star-Hspice cannot find it.

To remedy the problem, open a “DOS prompt” and go to the directory where you have your license file and type (without quotes) “dir”. Find your license file, and see if it has an additional extension such as .txt or .flx. To fix it, type “rename license.dat.?? license.dat”. This should rename the file to license.dat.

Solution: 439
Q: How do I load network files into AvanWaves on a PC?
A: To be able to view network drives in AvanWaves, you need to know the path to the network resource. To access a file on server in your home directory, you would use the path:

\Server\home\user

This has to be typed in the “open” dialog box in AvanWaves. AvanWaves will 'remember' this path throughout the current session.

Note: You have to be logged into the server you are trying to access, and have permission to read. AvanWaves may crash unexpectedly otherwise.

Solution: 448

Q: How do you run Star-Hspice from a command prompt in Windows?
A: Use the form:

hspice -i input.sp -o output.lis

This opens the Star-Hspice output window (shows the status of the .sw0 file) and runs the input file. When Star-Hspice is done, the window closes and the command prompt is updated.

Solution: 457

Q: How do I get a copy of the Star-Hspice manuals (for Windows)?
A: See this solution in “Documentation” on page 4.

Solution: 460
Q: If I downloaded Star-Hspice from Windows NT, will it work on all Windows operating systems?
A: Yes, Star-Hspice can be run on Windows 95/98 and NT 3.5 and 4.0.

Solution: 502

Q: Are there any demo files included with the PC version of the software?
A: Yes, during installation you have the option to install demos.

Solution: 503
Netlist/Options

Q: Determining the actual values used for simulation.
A: To determine the actual options used for a simulation, include the following line in your netlist:

```
.OPTION OPTS
```

This forces Star-Hspice to print out the options and corresponding values used in the simulation to stdout. Emphasis is generally redirected to a .lis file.

**Note:** If you are looking for the analysis-specific options, you must perform the type of analysis you are interested in within your netlist.

If you would like to find the actual element parameters used in the simulation, then you can use the element templates as described in Chapter 8, “Element Template Output” on page 8-35 in the 2000.4 Star-Hspice Manual.

**Solution:** 143

-----------------------------

Q: My netlists, which contain both IBIS elements and .ALTER statements, do not always complete successfully or the answers are odd.
A: This was corrected in Star-Hspice v1999.4 release and after.

**Solution:** 147

-----------------------------
Q: Can I use more than one interface option in my netlist?
A: No, the first interface option will be overwritten by the second. If you first set:

```
.OPTION CSDF
```
and later in the netlist set:

```
.OPTION POST
```

Star-Hspice first writes all the outputs in Viewlogic format and later the outputs are overwritten in AvanWaves format. You cannot see the output in Viewlogic but you can see them in AvanWaves.

Solution: 208

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Q: Why doesn’t .OPTION SCALE work in my simulation?
A: See this solution in “Models” on page 20.

Solution: 209

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Q: Why does my extracted netlist fail in Star-Hspice?
A: To use an extracted netlist, do a global search for the colon symbol (:). Replace this character globally with some unique pattern that is not likely to be in the netlist already. “_c_” is a possible replacement for the colon symbol.

Solution: 342

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Q: What is the magnitude difference between the NORM and UNORM options when using windowing in the .FFT statement?
A: If you use the option UNORM instead of the default NORM setting in your .FFT statement, you will see a difference in the signal magnitude. The reason for this is because of the equations that govern the different windows. The equations are given in Table 28-1 on page 28-4 in the 2000.4 Star-Hspice Manual.
The difference in magnitude is due to “Coherent Power Gain.” The definition for Coherent Power Gain is that it is a measure of the reduction in signal power due to the window function suppressing a coherent signal at the end of the measurement interval. This value is really given for each window in the equation column (it is not obvious in all cases). Here is a table to show what the coherent power gain is for each case.

<table>
<thead>
<tr>
<th>Window</th>
<th>Coherent Power Gain</th>
<th>Coherent Power Gain (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Bartlett</td>
<td>0.5</td>
<td>-6.021</td>
</tr>
<tr>
<td>Hanning</td>
<td>0.5</td>
<td>-6.021</td>
</tr>
<tr>
<td>Hamming</td>
<td>0.54</td>
<td>-5.352</td>
</tr>
<tr>
<td>Blackman</td>
<td>0.42323</td>
<td>-7.468</td>
</tr>
<tr>
<td>Blackman - Harris</td>
<td>0.35875</td>
<td>-8.904</td>
</tr>
<tr>
<td>Gaussian (a=3)</td>
<td>0.72</td>
<td>-2.853</td>
</tr>
<tr>
<td>Kaiser - Bessel (a=3)</td>
<td>0.4</td>
<td>-7.959</td>
</tr>
</tbody>
</table>

Solution: 351

Q: How can I find out what components are connected to the same node?

A: The option you probably need is: OPTION NODE. This option creates a node cross-reference table that can be printed. The table lists each node and all the elements connected to it. The terminal is indicated by a code, separated from the element name with a colon (:). The codes are:

- + Diode anode
- - Diode cathode
- B BJT base
- B MOSFET or JFET bulk
- C BJT collector
Q: In PETL, what is the significance of GRIDFACTOR?
A: The program uses the predefined number of segments according to the specified accuracy mode. In some cases, users may want higher accuracy than they can specify using the accuracy mode. Then they can double or triple the number of the predefined segments by setting GRIDFACTOR to 2 or 3.

Solution: 392

Q: Which integration method (TRAP or GEAR) is more accurate?
A: The TRAP (trapezoidal) method is faster and more accurate than GEAR. The GEAR method is used to remove the oscillation that can occur due to the TRAP algorithm. Circuits that are nonconvergent with TRAP will often converge with GEAR. The GEAR method is suitable for the circuits that are inductive in nature, such as switching regulators.

Solution: 409

Q: Why doesn’t losstangent have any affect on my W Element?
A: For the dielectric loss matrix to be computed and included in the simulation, “.FSOPTIONS computegd” has to be set in the SPICE deck.

Solution: 505
Q: I have 64 .ALTER statements but only get 36 .mtx files. Why?
A: By default, Star-Hspice only generates 36 unique output files for .tr*, .mt* etc. After 36 .ALTER statements, Star-Hspice starts from the beginning and writes over *.mt0 on up. To change this you can use either:

```
.OPTION ALT999
```
or:

```
.OPTION ALT9999
```

The first option lets Star-Hspice write 999 files before going back to the 0th file and alt9999 lets you write 9999 output files.

Solution: 511

```
******************************************************************************
```

Q: What does the option MU do? The manual just says it is a trapezoidal integration parameter.
A: The option MU acts as both a flag and a parameter. The equations for trapezoidal integration and backward-Euler integration are very similar to each other. The difference between these two is the parameter MU. By setting MU to 0.5 (the default), Star-Hspice uses trapezoidal integration. If MU=0, then Star-Hspice uses backward-Euler integration. If you set MU to a value between 0 and 0.5, then you get a blend of both methods.

Here is the order of the three integration methods available in Star-Hspice (trapezoidal, backward-Euler, Gear) from most accurate to most stable:

- Trapezoidal (most accurate)
- Backward-Euler
- Gear (most stable)

Solution: 556

```
******************************************************************************
```
Output

Q: How can I reduce the size of output files?
A: By adding .OPTION PROBE to the netlist, only those nodes explicitly referenced with .PROBE, .PRINT, .PLOT, or .GRAPH have output values displayed in one of the output files.

Solution: 43

Q: The output of my simulation is not what I expected at all. Also, the solution shows instability or extreme sensitivity to parameter or conductance values.
A: Try .OPTION PIVOT=1 (or .OPTION PIVOT=11). The default pivoting algorithm in Star-Hspice does not pick a pivot element. This can lead to instability when there are large conduction ratios (typically when small conductances exist).

Solution: 135

Q: I would like .GRAPH to only write to a file, and not send the job to the printer automatically.
A: You can accomplish this by editing your $installdir/meta.cfg. Comment out all the blocks that refer to “PLOTSETUP” and “PRTDEFAULT.” Then enter:

```
PRTDEFAULT = PS
PLOTSETUP PS PSCRIPT
PLOTFILE = /tmp/hspice.ps
END
```

Solution: 138
Q: Can I use more than one interface option in my netlist (and how does this affect my output)?

A: See this solution in “Netlist/Options” on page 27.

Solution: 208

---

Q: Can PRINTDATA in PETL only have the YES and NO option? When I set PRINTDATA=YES, it did not create RLGC file.

A: If RLGCFILE is specified, it prints to that file; otherwise, it prints to the standard error output, which can be redirected with the standard output using >& in UNIX. Please make sure to specify .PRINTDATA=YES in .FSOPTIONS and RLGCFILE in .MODEL cards.

Solution: 390

---
W Element/Field Solver

Q: How do I define an RLGC for a W Element without R and G?
A: Since the R and G matrix elements are optional, they can be left empty or 0 when defining RLGC parameters in W Element.

Solution: 458

Q: Why is the field solver not giving me accurate results?
A: To receive more accurate results when using the field solver, please use Star-Hspice 1999.4 or later. There have been many improvements to the W Element and the field solver.

Solution: 459

Q: Why doesn’t losstangent have any affect on my W Element?
A: See this solution in “Netlist/Options” on page 27.

Solution: 505
Waveform Viewing

Q: How can one view a waveform on an old waveform viewer (e.g., HSPLOT) or using third-party tools (e.g. Viewdraw)?

A: Add the following card to SPICE deck:

```
.OPTION POST_VERSION = 9007
```

Solution: 42

Q: I am performing analysis at different operating temperatures. However, my waveform/measurements are not changing when I change the temperature of the simulation. What is going on?

A: Most elements’ temperature coefficients default to zero. You must specify non-zero coefficients to derate components in the simulation. You can do this either on the element instance line or in the .MODEL card for the elements. For information on element-specific temperature coefficients, please see Star-Hspice Manual, Volume II.

Solution: 155

Q: Can I use more than one interface option in my netlist (how does this affect my waveform viewing)?

A: See this solution in “Netlist/Options” on page 27.

Solution: 208

Q: How do I create eye diagrams with Star-Hspice?

A: Here is an example of an eye diagram circuit in the Star-Hspice distribution:

```
<installdir>/demo/awaves/demo/eyediag*
```
As a quick reference, an eye diagram simply imitates the behavior of an oscilloscope (the waveform is displayed starting from the left end of the window and moving to the right at a given rate). When the electron beam reaches the right end of the window, it is moved quickly to the left end of the window where another “cycle” can be displayed. This type of display is quite useful in observing cyclic patterns that vary slightly from one cycle to another.

To get these results in Star-Hspice, you need to include these two statements in your netlist:

```
.PARAM width=5ns phase=0ns
.PROBE TRAN TIME2=par('TIME+phase-int((TIME+phase)/width)*width')
```

This creates a new TIME2 output variable. You may adjust the width of the window (in ns) with the parameter “width”. You may also adjust the phase of the waveform with the parameter “phase.” To get these results in AvanWaves:

- Open the resulting .tr# file
- In the “Results Browser” window, under “Types:” select “Params”
- Under “Curves:” select “time2”
- Under “Current X-Axis,” click on “Apply”
- Under “Types:” again, select “Voltages,” “Currents” or any other type of output variable you want to display
- Double click on the desired signals you want to display
- In the viewing window, click on the right mouse button; select “Monotonic Plot ...”

Solution: 611
Appendix B

Interfaces For Design Environments

Star-Hspice simulation is supplemented by the following design interface products:

■ AvanLink to Cadence Composer™ and Cadence Analog Artist™ interfaces Star-Hspice with Cadence Design Systems’ Design Framework II to support data interchange and cross-probing with Analog Artist and Composer.

■ AvanLink to Mentor Design Architect™ interfaces Star-Hspice with Mentor Graphics’ Falcon Framework to support cross-probing with Design Viewpoint Editor and data interchange with Design Architect (DA) and Design Viewpoint Editor (DVE).

Overviews of these interface products are presented in the following sections:

■ AvanLink to Cadence Composer and Analog Artist
■ AvanLink for Design Architect
■ Viewlogic Links
AvanLink to Cadence Composer and Analog Artist

AvanLink to Cadence Composer and Analog Artist provides a netlister, a symbol library, cross-probing, and backannotation capabilities. AvanLink is an Avant! interface product that links the Star-Hspice circuit simulator with the following products from Cadence:

- Composer, Versions 4.2.1a, 4.2.2, 4.3.2, and 4.3.3
- Analog Artist, Versions 4.2.1a, 4.2.2, 4.3.2, and 4.3.3

Features

AvanLink has the following functionality:

- Provides hierarchical Netlisting (HNL), with incremental capability (IHNL), through the CadenceLink netlister and the Cadence simulation environment
- Provides Avant!’s metaLib class-based library to support all of the Star-Hspice elements. The metaLib library uses Cadence’s standard Component Description Format (CDF).
- Supports cross-probing for both voltage and current from both Composer and Analog Artist schematics to Avant!’s AvanWaves waveform display tool
- Supports hierarchical parameter passing and algebraic function operations
- Preserves Star-Hspice hierarchical path names through a transparent net name mapping function
- Supports the Parameter Storage Format (PSF) output program for displaying waveforms with Analog Artist
- Provides an automatic library translation program for the Cadence sample and analogLib libraries and user-defined libraries
- Supports backannotation of operating points and element parameters back to the schematic
- Supports graphical setup of simulation initial conditions
Figure B-1 shows how AvanLink fits into the design process.

**Figure B-1: AvanLink Architecture for Design Framework II with Composer**

![Diagram showing AvanLink architecture](image)

**Environment**

AvanLink provides an integrated environment that allows configuration of design, simulation, and display tools to support the chosen design flow. Develop the design using Composer, simulate it using Star-Hspice, and then view it using AvanWaves. Access AvanLink by selecting options on menus, or by using specific keys assigned to these options. Figure B-2 shows the operating environment for AvanLink.
Figure B-2: Star-Hspice AvanLink in a Cadence Composer Environment

Figure B-2 is an overview of the process involved in creating a design with AvanLink.

AvanLink Design Flow

Figure B-3 is an overview of the process involved in creating a design with AvanLink.
Schematic Entry and Library Operations

AvanLink helps you design your schematic using the metaLib Component Description Format (CDF) library. This is a class-based library that includes all the Star-Hspice elements. Each element is described by a set of parameters that are organized in classes and subclasses. An example of a metaLib class-based CDF library structure for a source element is shown in Figure B-4.

As each element in the design is instantiated, a form is opened, automatically displaying the CDF parameters for that element. Modifications can be made to
these parameters as necessary. AvanLink guarantees that Star-Hspice generates an accurate and syntactically correct netlist for simulation.

**Figure B-4: Example metaLib Library Structure for a Source Element**

<table>
<thead>
<tr>
<th>Library</th>
<th>Category</th>
<th>Element</th>
<th>Class</th>
<th>Subclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>metaLib</td>
<td>sources</td>
<td>vsrc</td>
<td>tran</td>
<td>pulse</td>
</tr>
<tr>
<td></td>
<td></td>
<td>isrc</td>
<td>dc</td>
<td>SIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PWL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PL</td>
</tr>
</tbody>
</table>

Use the library interchange program supplied by AvanLink to convert personal or customized symbol libraries to incorporate *metaLib* CDF parameters. Figure B-5 diagrams this procedure.

**Figure B-5: AvanLink Library Interchange Function**

**Netlist Generation**

AvanLink uses the Cadence Open Simulation System (OSS) to integrate Star-Hspice into the Cadence Simulation Environment (SE). The AvanLink netlister includes customized functions, in addition to those in the OSS Hierarchical Netlister, in order to provide the following features:
Preservation of the pin names in the design
Maintenance of the original design signal names
Preservation of design hierarchy
Time savings when dealing with large designs that require few modifications

Simulation

The simulation is run in a user-designated simulation directory. The Cadence simulation environment initializes this directory and copies the basic Star-Hspice control files into it. You can modify these files to add personal Star-Hspice commands as desired. All Star-Hspice analysis types, including advanced analyses such as Optimization and Monte Carlo, can be specified with the control file.

Waveform Display

Analyze and display output waveforms using the AvanWaves program. The cross-probing feature allows probing a net in the schematic window and viewing its signal in the AvanWaves display window. Cross-probing is supported for signals generated by transient, DC, or AC analysis. The cross-probing feature also allows pins to be probed for branch currents.

AvanLink also generates waveform format (PSF) for displaying in the Analog Artist environment and supports cross-probing and backannotation of the Analog Artist display.
AvanLink for Design Architect

AvanLink for Design Architect is an Avant! interface product that links the Star-Hspice circuit simulator with the following products from Mentor Graphics:

- Design Architect
- Design Viewpoint Editor (DVE)

AvanLink-DA provides a netlister, a symbol library, and cross-probing and backannotation capabilities.

Features

AvanLink-DA features include the following:

- Hierarchical netlisting, provided through the AvanLink-DA netlister and simulation environment
- Avant!’s metaLib class-based library, supporting all of the Star-Hspice elements, with advanced Star-Hspice properties and element syntax
- Cross-probing from Design Viewpoint Editor schematics to Avant!’s waveform display tool, AvanWaves
- Hierarchical parameter passing and algebraic function operations
- Facility to preserve Star-Hspice hierarchical path names through a transparent net name mapping function
- Utilities for migrating customers’ component libraries to AvanLink-DA
- Backannotation of operating point voltages back to the Mentor Graphics Design Viewpoint Editor
- Ability to set up and probe terminal current values
- Ability to specify initial condition (IC) and nodeset values
- Support for element parameter backannotation onto the schematic within DVE
- Support for Star-Hspice legacy libraries

Figure B-6 shows how AvanLink-DA fits into the design process.
AvanLink-DA provides an integrated environment for circuit design, simulation, and waveform display. Develop a design using the Design Architect, simulate it using Star-Hspice, and then view the results using AvanWaves. Access AvanLink-DA by selecting options on menus or by clicking buttons on the palette in the Mentor Graphics Design Architect window corresponding to these options. Figure B-7 shows the operating environments for AvanLink-DA.
AvanLink-DA Design Flow

Figure B-8 provides an overview of the processes involved in creating and simulating a design with AvanLink-DA.

Figure B-8: Design Flow for AvanLink for Design Architect

Schematic Entry and Library Operations

AvanLink-DA helps schematic design using metaLib. Avant!’s metaLib is a class-based library that includes all of the Star-Hspice elements. Each element is described by a set of parameters that are organized in classes and subclasses. An example of a metaLib class-based library structure for a voltage source element is shown in Figure B-9.
After each design element is instantiated, you can display and edit the element’s parameters. Modify these parameters as necessary. AvanLink-DA guarantees that Star-Hspice generates an accurate and syntactically correct netlist for simulation.

The library migration program supplied by AvanLink-DA to convert personal or customized symbol libraries to incorporate metaLib attributes. Figure B-10 diagrams this procedure.

**Figure B-9: Example metaLib Library Structure for a Source Element**

<table>
<thead>
<tr>
<th>Library</th>
<th>Element</th>
<th>Class</th>
<th>Subclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>metaLib</td>
<td>vsrc</td>
<td>tran</td>
<td>pulse</td>
</tr>
<tr>
<td></td>
<td>isrc</td>
<td>dc</td>
<td>SIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EXP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PWL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PL</td>
</tr>
</tbody>
</table>

**Figure B-10: AvanLink-DA Library Migration Function**

Custom Library → Library Migration Program → Customized Library with AvanLink-DA Attributes
Netlist Generation

The AvanLink-DA Netlister provides the following features:

- Creation of a complete hierarchical Star-Hspice netlist
- Preservation of the original design pin names
- Maintenance of the original design signal names
- Preservation of the design hierarchy

Simulation

Run the simulation in a self-designated simulation run directory. The AvanLink-DA simulation environment initializes this directory and copies the basic Star-Hspice control files into it. Modify these files to add personal Star-Hspice commands as desired. Specify all Star-Hspice analysis types, including advanced analyses such as Optimization and Monte Carlo, using the control file.

Waveform Display

Display output waveforms using the waveform display tool, AvanWaves. The cross-probing feature allows you to probe on a net in the schematic window inside DVE and view its signal in the AvanWaves display window. Cross-probing is supported for signals generated by transient, DC, or AC analysis. The cross-probing feature also allows pins to be probed for branch currents.
**Viewlogic Links**

Users of Viewlogic have access to Star-Hspice through the Viewlogic Powerview framework. These tools (provided by Viewlogic) include a netlister, a Star-Hspice option CSDF for waveform display in viewtrace, and cross-probing. Additionally, Viewlogic provides a Star-Hspice/Madssim mixed-signal simulation solution.
A transmission line delivers an output signal at a distance from the point of signal input. Any two conductors can make up a transmission line. The signal that is transmitted from one end of the pair to the other end is the voltage between the conductors. Power transmission lines, telephone lines, and waveguides are examples of transmission lines. Traces on printed circuit boards and multichip modules (MCMs) in integrated circuits are other electrical elements that are examples of transmission lines.

With current technologies that use high-speed active devices on both ends of most circuit traces, all of the following transmission line effects must be considered during circuit analysis:

- Time delay
- Phase shift
- Power, voltage, and current loss
- Distortion
- Reduction of frequency bandwidth
- Coupled line crosstalk

Star-Hspice provides accurate modeling for all kinds of circuit connections, including both lossless (ideal) and lossy transmission line elements.

This chapter describes:

- Selecting Wire Models
- Performing Interconnect Simulation
- Understanding the Transmission Line Theory
- References
Selecting Wire Models

A transmission line or interconnect is one of the following:

- Wire
- Trace
- Conductor
- Line

Many applications model electrical properties of interconnections between nodes by their equivalent circuits, and integrate them into the system simulation to accurately predict system performance.

An electrical model that simulates the behavior of interconnect must consider all of the following:

- Physical nature or electrical properties of the interconnect
- Bandwidth or risetime and source impedance of signals of interest
- Interconnect’s actual time delay
- Complexity and accuracy of the model, and the corresponding effects on the amount of CPU time required for simulations

You can choose from the following circuit models for interconnects:

- No model at all. Use a common node to connect two elements.
- Lumped models with R, L, and C Elements, as described in “Statistical Analysis and Optimization” on page 13-1. These include a series resistor (R), a shunt capacitor (C), a series inductor and resistor (RL), and a series resistor and a shunt capacitor (RC).
- Transmission line models such as an ideal transmission line (T Element) or a lossy transmission line (U Element)
Choosing the simplest model that adequately simulates the required performance minimizes sources of confusion and error during analysis.

Generally, to simulate both low and high frequency electrical properties of interconnects, select the U Element transmission line model. For compatibility with conventional versions of SPICE, use one a discrete lumped model or the T Element.

Following are the factors that determine the best choice of a transmission line:

**Source properties**
- $t_{\text{rise}}$ = source risetime
- $R_{\text{source}}$ = source output impedance

**Interconnect properties**
- $Z_0$ = characteristic impedance
- $T_D$ = time delay of the interconnection

or
- $R$ = equivalent series resistance
- $C$ = equivalent shunt capacitor
- $L$ = equivalent series inductance

**Figure C-1** shows you how to select a model based on source and interconnect properties.
Selecting Wire Models

Use the U model with either the ideal T Element or the lossy U Element. You can also use the T Element alone, without the U model. Star-Hspice offers both, a flexible definition of the conventional SPICE T Element and an accurate U Element lossy simulation.
The T and U Elements do not support the \texttt{<M=val> multiplier function.} If a U or T Element is used in a subcircuit and an instance of the subcircuit has a multiplier applied, the results are inaccurate.

A warning message similar to the following is issued in both the status file (.st0) and the output file (.lis) if the smallest transmission line delay is less than TSTOP/10e6:

\begin{verbatim}
**warning**: the smallest T-line delay (TD) = 0.245E-14 is too small Please check TD, L and SCALE specification
\end{verbatim}

This feature is an aid to finding errors that cause excessively long simulations.

**Using Ground and Reference Planes**

All transmission lines have a ground reference for the signal conductors. In this manual, the ground reference is called the reference plane so that it is not confused with SPICE ground. The reference plane is the shield or the ground plane of the transmission line element. The reference plane nodes may or may not be connected to SPICE ground.

**Selecting Ideal or Lossy Transmission Line Element**

The ideal and lossy transmission line models each have particular advantages, and they may be used in a complementary manner. Both model types are fully
functional in AC analysis and transient analysis. Some of the comparative advantages and uses of each type of model are listed in Table C-1.

Table C-1: Ideal versus Lossy Transmission Line

<table>
<thead>
<tr>
<th>Ideal Transmission Line</th>
<th>Lossy Transmission Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>lossless</td>
<td>includes loss effects</td>
</tr>
<tr>
<td>used with voltage sources</td>
<td>used with buffer drivers</td>
</tr>
<tr>
<td>no limit on input risetime</td>
<td>prefiltering necessary for fast rise</td>
</tr>
<tr>
<td>less CPU time for long delays</td>
<td>less CPU time for short delays</td>
</tr>
<tr>
<td>differential mode only</td>
<td>supports common mode simulation</td>
</tr>
<tr>
<td>no ground bounce</td>
<td>includes reference plane reactance</td>
</tr>
<tr>
<td>single conductor</td>
<td>up to five signal conductors allowed</td>
</tr>
<tr>
<td>AC and transient analysis</td>
<td>AC and transient analysis</td>
</tr>
</tbody>
</table>

The ideal line is modeled as a voltage source and a resistor. The lossy line is modeled as a multiple lumped filter section, as shown in Figure C-3.

Figure C-3: Ideal versus Lossy Transmission Line Model

Because the ideal element represents the complex impedance as a resistor, the transmission line impedance is constant, even at DC values. On the other hand, you must prefilter the lossy element if ideal piecewise linear voltage sources are used to drive the line.
Selecting U Models

The U model allows three different description formats: geometric/physical, precomputed, and electrical. It provides equally natural description of vendor parts, physically described shapes, and parametric input from field solvers. The description format is specified by the required model parameter ELEV, as follows:

- **ELEV=1** – geometric/physical description such as width, height, and resistivity of conductors. It is used by board designers dealing with physical design rules.
- **ELEV=2** – precomputed parameters. These are available with some commercial packaging, or as a result of running a field solver on a physical description of commercial packaging.
- **ELEV=3** – electrical parameters such as delay and impedance, available with purchased cables. It only allows one conductor and ground plane for PLEV = 1.

The U model explicitly supports transmission lines with several types of geometric structures. The geometric structure type is indicated by the PLEV model parameter, as follows:

- **PLEV=1** – Selects planar structures, such as microstrip and stripline, which are the usual conductor shapes on integrated circuits and printed-circuit boards.
- **PLEV=2** – Selects coax, which frequently is used to connect separated instruments.
- **PLEV=3** – Selects twinlead, which is used to connect instruments and to suppress common mode noise coupling.
Using Transmission Lines - Example

The following Star-Hspice file fragment is an example of how both T Elements and U Elements can be referred to a single U model as indicated in Figure C-2. The file specifies a 200 millimeter printed circuit wire implemented as both a U Element and a T Element. The two implementations share a U model that is a geometric description (ELEV=1) of a planar structure (PLEV=1).

```
T1 in gnd t_out gnd micro1 L=200m
U1 in gnd u_out gnd micro1 L=200m
.model micro1 U LEVEL=3 PLEV=1 ELEV=1 wd=2m ht=2m th=0.25m KD=5
```
The next section provides details of element and model syntax where:

- `T1, U1` are element names
- `micro1` is the model name
- `in, gnd, t_out, and u_out` are nodes
- `L` is the length of the signal conductor
- `wd, ht, th` are dimensions of the signal conductor and dielectric, and
- `KD` is the relative dielectric constant
Performing Interconnect Simulation

This section provides details of the requirements for T line or U line simulation.

Using the Ideal Transmission Line

The ideal transmission line element contains the element name, connecting nodes, characteristic impedance ($Z_0$), and wire delay (TD), unless $Z_0$ and TD are obtained from a U model. In that case, it contains a reference to the U model.

![Figure C-5: Ideal Element Circuit](image)

The input and output of the ideal transmission line have the following relationships:

\[
V_{\text{in}}|_{\text{t}} = V(\text{out} - \text{refout})|_{\text{t}} - \text{TD} + (i_{\text{out}} \times Z_0)|_{\text{t}} - \text{TD}
\]

\[
V_{\text{out}}|_{\text{t}} = V(\text{in} - \text{refin})|_{\text{t}} - \text{TD} + (i_{\text{in}} \times Z_0)|_{\text{t}} - \text{TD}
\]

The signal delays for ideal transmission lines are given by:

\[T_{\text{Deff}} = TD - L\]

If TD is given, or

\[T_{\text{Deff}} = NL/F\]

if NL and F are given, or

\[T_{\text{Deff}} = TD\]
if a U model is used.

The ideal transmission line only delays the difference between the signal and the reference. Some applications, such as a differential output driving twisted pair cable, require both differential and common mode propagation. Use a U Element, if you need the full signal and reference. You can use approximately two T Elements (as shown in Figure C-6). In this figure, the two lines are completely uncoupled, so that only the delay and impedance values are correctly modeled.

Figure C-6: Use of Two T Elements for Full Signal and Reference

You cannot implement coupled lines with the T Element, so use U Elements for applications requiring two or three coupled conductors.

Star-Hspice uses a transient timestep that does not exceed half the minimum line delay. Very short transmission lines (relative to the analysis time step) cause long simulation times. You can replace very short lines with a single R, L, or C Element (see Figure C-1).

Lossy U Element Statement

Star-Hspice uses a U Element to model single and coupled lossy transmission lines for various planar, coaxial, and twinlead structures. When a U Element is included in your netlist, Star-Hspice creates an internal network of R, L, C, and G Elements to represent up to five lines and their coupling capacitances and inductances. For more information, see “Using Passive Device Models” on page 14-1.
You can specify interconnect properties in three ways:

- Specify the R, L, C, and G (conductance) parameters in a matrix form (ELEV = 2).
- Provide common electrical parameters, such as characteristic impedance and attenuation factors (ELEV = 3).
- Specify the geometry and the material properties of the interconnect (ELEV = 1).

This section initially describes how to use the third method.

The U model provided with Star-Hspice has been optimized for typical geometries used in ICs, MCMs, and PCBs. The model's closed form expressions are optimized via measurements and comparisons with several different electromagnetic field solvers.

The Star-Hspice U Element geometric model handles one to five uniformly spaced transmission lines, all at the same height. Also, the transmission lines can be on top of a dielectric (microstrip), buried in a sea of dielectric (buried), have reference planes above and below them (stripline), or have a single reference plane and dielectric above and below the line (overlay). Thickness, conductor resistivity, and dielectric conductivity allow for calculating loss as well.

The U Element statement contains the element name, the connecting nodes, the U model reference name, the length of the transmission line, and, optionally, the number of lumps in the element. You can create two kinds of lossy lines: lines with a reference plane inductance (LRR, controlled by the model parameter LLEV) and lines without a reference plane inductance. Wires on integrated circuits and printed circuit boards require reference plane inductance.

The reference ground inductance and the reference plane capacitance to SPICE ground are set by the HGP, CMULT, and optionally, the CEXT parameters.

**Lossy U Model Statement**

The schematic for a single lump of the U model, with LLEV=0, is shown in Figure C-7. If LLEV is 1, the schematic includes inductance in the reference path as well as capacitance to HSPICE ground. See the section, “Reference Planes...”
and HSPICE Ground” on page C-18 for more information about LLEV=1 and reference planes.

**Figure C-7: Lossy Line with Reference Plane**

The following describes Star-Hspice netlist syntax for the U model. Tables C-2 and C-3 list the model parameters.

**U Model Syntax**

The syntax is:

```
.MODEL mname U LEVEL=3 ELEV=val PLEV=val <DLEV=val>
+ <LLEV=val> + <Pname=val> ...
```

- **LEVEL=3** Selects the lossy transmission line model
- **ELEV=val** Selects the electrical specification format including the geometric model
  - val=1
- **PLEV=val** Selects the transmission line type
- **DLEV=val** Selects the dielectric and ground reference configuration
- **LLEV=val** Selects the use of reference plane inductance and capacitance to HSPICE ground.
- **Pname=val** Specifies a physical parameter, such as NL or WD (see Table C-2) or a loss parameter, such as RHO or NLAY (see Table C-3).

**Figure C-8** shows the three dielectric configurations for the geometric U model. You use the DLEV switch to specify one of these configurations. The geometric U model uses ELEV=1.
In Figure C-8 a) sea, DLEV=0, b) microstrip, DLEV=1, c) stripline, DLEV=2 and d) overlay, DLEV=3

Planar Geometric Models Lossy U Model Parameters

Common Planar Model Parameters

The parameters for U models are shown in Table C-2.

Table C-2: U Element Physical Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td></td>
<td>required (=3) required for lossy transmission lines model</td>
<td></td>
</tr>
<tr>
<td>ELEV</td>
<td></td>
<td>required Electrical model (=1 for geometry)</td>
<td></td>
</tr>
<tr>
<td>DLEV</td>
<td></td>
<td>Dielectric model (=0 for sea, =1 for microstrip, =2 stripline, =3 overlay; default is 1)</td>
<td></td>
</tr>
<tr>
<td>PLEV</td>
<td></td>
<td>required Transmission line physical model (=1 for planar)</td>
<td></td>
</tr>
<tr>
<td>LLEV</td>
<td></td>
<td>Omit or include the reference plane inductance (=0 to omit, =1 to include; default is 0)</td>
<td></td>
</tr>
<tr>
<td>NL</td>
<td></td>
<td>Number of conductors (from 1 to 5)</td>
<td></td>
</tr>
<tr>
<td>WD</td>
<td>m</td>
<td>Width of each conductor</td>
<td></td>
</tr>
<tr>
<td>HT</td>
<td>m</td>
<td>Height of all conductors</td>
<td></td>
</tr>
</tbody>
</table>
There are two parametric adjustments in the U model: XW, and CORKD. XW adds to the width of each conductor, but does not change the conductor pitch (spacing plus width). XW is useful for examining the effects of conductor etching. CORKD is a multiplier for the dielectric value. Some board materials vary more than others, and CORKD provides an easy way to test tolerance to dielectric variations.

**Physical Parameters**

The dimensions for one and two-conductor planar transmission lines are shown in Figure C-9.

### Table C-2: U Element Physical Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TH</td>
<td>m</td>
<td></td>
<td>Thickness of all conductors</td>
</tr>
<tr>
<td>THB</td>
<td>m</td>
<td></td>
<td>Reference plane thickness</td>
</tr>
<tr>
<td>TS</td>
<td>m</td>
<td></td>
<td>Distance between reference planes for stripline (default for DLEV=2 is 2 HT + TH. TS is not used when DLEV=0 or 1)</td>
</tr>
<tr>
<td>SP</td>
<td>m</td>
<td></td>
<td>Spacing between conductors (required if NL &gt; 1)</td>
</tr>
<tr>
<td>KD</td>
<td></td>
<td></td>
<td>Dielectric constant</td>
</tr>
<tr>
<td>XW</td>
<td>m</td>
<td></td>
<td>Perturbation of conductor width added (default is 0)</td>
</tr>
<tr>
<td>CEXT</td>
<td>F/m</td>
<td></td>
<td>External capacitance between reference plane and ground. Only used when LLEV=1, this overrides the computed characteristic.</td>
</tr>
<tr>
<td>CMULT</td>
<td></td>
<td>1</td>
<td>Dielectric constant of material between reference plane and ground (default is 1 – only used when LLEV=1)</td>
</tr>
<tr>
<td>HGP</td>
<td>m</td>
<td></td>
<td>Height of the reference plane above HSPICE ground. Used for computing reference plane inductance and capacitance to ground (default is 1.5*HT – HGP is only used when LLEV=1).</td>
</tr>
<tr>
<td>CORKD</td>
<td></td>
<td></td>
<td>Perturbation multiplier for dielectric (default is 1)</td>
</tr>
<tr>
<td>WLUMP</td>
<td>20</td>
<td></td>
<td>Number of lumps per wavelength for error control</td>
</tr>
<tr>
<td>MAXL</td>
<td>20</td>
<td></td>
<td>Maximum number of lumps per element</td>
</tr>
</tbody>
</table>
Loss Parameters

Loss parameters for the U model are shown in Table C-3.

**Table C-3: U Element Loss Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHO</td>
<td>ohm·m</td>
<td>Conductor resistivity (default is rho of copper, 1.7E-9 ohm·m)</td>
</tr>
<tr>
<td>RHOB</td>
<td>ohm·m</td>
<td>Reference plane resistivity (default value is for copper)</td>
</tr>
<tr>
<td>NLAY</td>
<td></td>
<td>Number of layers for conductor resistance computation (=1 for DC resistance or core resistance, =2 for core and skin resistance at skin effect frequency)</td>
</tr>
<tr>
<td>SIG</td>
<td>mho/m</td>
<td>Dielectric conductivity</td>
</tr>
</tbody>
</table>

Losses have a large impact on circuit performance, especially as clock frequencies increase. RHO, RHOB, SIG, and NLAY are parameters associated with losses. Time domain simulators, such as SPICE, cannot directly handle losses that vary with frequency. Both the resistive skin effect loss and the effects of dielectric loss create loss variations with frequency. NLAY is a switch that turns on skin effect calculations in Star-Hspice. The skin effect resistance is proportional to the conductor and backplane resistivities, RHO and RHOB.

The dielectric conductivity is included through SIG. The U model computes the skin effect resistance at a single frequency and uses that resistance as a constant.
The dielectric SIG is used to compute a fixed conductance matrix, which is also constant for all frequencies. A good approximation of losses can be obtained by computing these resistances and conductances at the frequency of maximum power dissipation. In AC analysis, resistance increases as the square root of frequency above the skin-effect frequency, and resistance is constant below the skin effect frequency.

**Geometric Parameter Recommended Ranges**

The U Element analytic equations compute quickly, but have a limited range of validity. The U Element equations were optimized for typical IC, MCM, and PCB applications. Table C-4 lists the recommended minimum and maximum values for U Element parameter variables.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>KD</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>WD/HT</td>
<td>0.08</td>
<td>5</td>
</tr>
<tr>
<td>TH/HT</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TH/WD</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SP/HT</td>
<td>0.15</td>
<td>7.5</td>
</tr>
<tr>
<td>SP/WD</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

The U Element equations lose their accuracy when you use values outside the recommended ranges. Because the single-line formula is optimized for single lines, you will notice a difference between the parameters of single lines and two coupled lines at a very wide separation. The absolute error for a single line parameter is less than 5% when used within the recommended range. The main line error for coupled lines is less than 15%. Coupling errors can be as high as 30% in cases of very small coupling. Since the largest errors occur at small coupling values, actual waveform errors are kept small.
Reference Planes and HSPICE Ground

Figure C-10 shows a single lump of a U model, for a single line with reference plane inductance. When LLEV=1, the reference plane inductance is computed, and capacitance from the reference plane to HSPICE ground is included in the model. The reference plane is the ground plane of the conductors in the U model.

![Figure C-10: Schematic of a U Element Lump when LLEV=1](image)

The model reference plane is not necessarily the same as HSPICE ground. For example, a printed circuit board with transmission lines can have a separate reference plane above a chassis. Star-Hspice uses either HGP, the distance between the reference plane and HSPICE ground, or $C_{ext}$ to compute the parameters for the ground-to-reference transmission line.

When HGP is used, the capacitance per meter of the ground-to-reference line is computed based on a planar line of width $(NL+2)(WD+SP)$ and the height of HGP above the SPICE ground. CMULT is used as the dielectric constant of the ground-to-reference transmission line. If $C_{ext}$ is given, then $C_{ext}$ is used as the capacitance per meter for the ground-to-reference line. The inductance of the ground-to-reference line is computed from the capacitance per meter and an assumed propagation at the speed of light.

Estimating the Skin Effect Frequency

Most of the power in a transmission line is dissipated at the clock frequency. As a first choice, Star-Hspice estimates the maximum dissipation frequency, or skin effect frequency, from the risetime parameter. The risetime parameter is set with the .OPTION statement (for example, .OPTION RISETIME=0.1ns).
Some designers use $0.35/trise$ to estimate the skin effect frequency. This estimate is good for the bandwidth occupied by a transient, but not for the clock frequency, at which most of the energy is transferred. In fact, a frequency of $0.35/trise$ is far too high and results in excessive loss for almost all applications. Star-Hspice computes the skin effect frequency from $1/(15\times trise)$. If you use precomputed model parameters (ELEV = 2), compute the resistance matrix at the skin effect frequency.

When the risetime parameter is not given, Star-Hspice uses other parameters to compute the skin effect frequency. Star-Hspice examines the .TRAN statement for tstep and delmax and examines the source statement for trise. If you set any one of the parameters tstep, delmax, and trise, Star-Hspice uses the maximum of these parameters as the effective risetime.

In AC analysis, the skin effect is evaluated at the frequency of each small-signal analysis. Below the computed skin effect frequency (ELEV=1) or FR1(ELEV=3), the AC resistance is constant. Above the skin effect frequency, the resistance increases as the square root of frequency.

**Number of Lumped-Parameter Sections**

The number of sections (lumps) in a transmission line model also affects the transmission line response. Star-Hspice computes the default number of lumps from the line delay and the signal risetime. There should be enough lumps in the transmission line model to ensure that each lump represents a length of line that is a small fraction of a wavelength at the highest frequency used. It is easy to compute the number of lumps from the line delay and the signal risetime, using an estimate of $0.35/trise$ as the highest frequency.

For the default number of lumps, Star-Hspice uses the smaller of 20 or $1+(20\times TDeff/trise)$, where TDeff is the line delay. In most transient analysis cases, using more than 20 lumps gives a negligible bandwidth improvement at the cost of increased simulation time. In AC simulations over many decades of frequency with lines over one meter long, more than 20 lumps may be needed for accurate simulation.
**Ringing**

Sometimes a transmission line simulation shows ringing in the waveforms, as in Figure C-27. If the ringing is not verifiable by measurement, it might be due to an incorrect number of lumps in the transmission line models or due to the simulator integration method. Increasing the number of lumps in the model or changing the integration method to Gear reduces the amount of ringing due to simulation errors. The default Star-Hspice integration method is TRAP (trapezoidal), but you can change it to Gear with the statement .OPTION METHOD=GEAR.

See “Oscillations Due to Simulation Errors” on page C-61 for more information on the number of lumps and ringing.

The next section covers parameters for geometric lines. Coaxial and twinlead transmission lines are discussed in addition to the previously described planar type.

**Geometric Parameters (ELEV=1)**

Geometric parameters provide a description of a transmission line in terms of the geometry of its construction and the physical constants of each layer, or other geometric shape involved.

**PLEV=1, ELEV=1 Geometric Planar Conductors**

Planar conductors are used to model printed circuit boards, packages, and integrated circuits. The geometric planar transmission line is restricted to:

- One conductor height (HT or HT1)
- One conductor width (WD or WD1)
- One conductor thickness (TH or TH1)
- One conductor spacing (SP or SP12)
- One dielectric conductivity (SIG or SIG1)
- One or two relative dielectric constants (KD or KD1, and KD2 only if DLEV=3)

Common planar conductors include:
- DLEV=0 – microstrip sea of dielectric. This planar conductor has a single reference plane and a common dielectric surrounding conductor (Figure C-11).
- DLEV=1 – microstrip dual dielectric. This planar conductor has a single reference plane and two dielectric layers (Figure C-12).
- DLEV=2 – stripline. This planar conductor has an upper and lower reference plane (Figure C-13). Both symmetric and asymmetric spacing are available.
- DLEV=3 – overlay dielectric. This planar conductor has a single reference plane and an overlay of dielectric material covering the conductor (Figure C-14).

Figure C-11: Planar Transmission Line, DLEV=0, Sea of Dielectric
Figure C-12: Planar Transmission Line, DLEV=1, Microstrip

Figure C-13: Planar Transmission Line, DLEV=2, Stripline
Figure C-14: Planar Transmission Line, DLEV=3, Overlay Dielectric
## ELEV=1 Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| DLEV         |       | 1.0     | 0: microstrip sea of dielectric  
               |               | 1: microstrip layered dielectric  
               |               | 2: stripline |
| NL           |       | 1       | Number of conductors |
| NLAY         |       | 1.0     | Layer algorithm:  
               |               | 1: DC cross section only  
               |               | 2: skindepth cross section on surface plus DC core |
| HT(HT1)      | m     | req     | Conductor height |
| WD(WD1)      | m     | req     | Conductor width |
| TH(TH1)      | m     | req     | Conductor thickness |
| THK1         | m     | HT      | Dielectric thickness for DLEV=3 |
| THK2         | m     | 0.0     | Overlay dielectric thickness for DLEV=3  
               |               | 0≤THK2<3·HT (see Note) |
| THB          | m     | calc    | Reference conductor thickness |
| SP(SP12)     | m     | req     | Spacing: line 1 to line 2 required for nl > 1 |
| XW           | m     | 0.0     | Difference between drawn and realized width |
| TS           | m     | calc    | Height from bottom reference plane to top reference plane  
               |               | TS=TH+2·HT (DLEV=2, stripline only) |
| HGP          | m     | HT      | Height of reference plane above spice ground – LLEV=1 |
| CMULT        |       | 1.0     | Multiplier (used in defining CPR) for dielectric constant of material between shield and SPICE ground when LLEV=1 and CEXT is not present |
| CEXT         | F/m   | und     | External capacitance from reference plane to circuit ground point – used only to override HGP and CMULT computation |
Note: If THK2 is greater than three times HT, simulation accuracy decreases. A warning message is issued to indicate this. A reference plane is a ground plane, but it is not necessarily at SPICE ground potential.

### Lossy U Model Parameters for Geometric Coax (PLEV=2, ELEV=1)

**Figure C-15: Geometric Coaxial Cable**

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHO</td>
<td>ohm-m</td>
<td>17E-9</td>
<td>Resistivity of conductor material – defaults to value for copper</td>
</tr>
<tr>
<td>RHOB</td>
<td>ohm-m</td>
<td>rho</td>
<td>Resistivity of reference plane material</td>
</tr>
<tr>
<td>SIG1(SIG)</td>
<td>mho/m</td>
<td>0.0</td>
<td>Conductivity of dielectric</td>
</tr>
<tr>
<td>KD1(KD)</td>
<td></td>
<td>4.0</td>
<td>Relative dielectric constant of dielectric</td>
</tr>
<tr>
<td>KD2</td>
<td>KD</td>
<td></td>
<td>Relative dielectric constant of overlay dielectric for DLEV=3</td>
</tr>
<tr>
<td>CORKD</td>
<td></td>
<td>1.0</td>
<td>Correction multiplier for KD</td>
</tr>
</tbody>
</table>
## Geometric Coax Parameters

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA</td>
<td>m</td>
<td>req</td>
<td>Outer radius of inner conductor</td>
</tr>
<tr>
<td>RB</td>
<td>m</td>
<td>req</td>
<td>Inner radius of outer conductor (shield)</td>
</tr>
<tr>
<td>RD</td>
<td>m</td>
<td>ra+rb</td>
<td>Outer radius of outer conductor (shield)</td>
</tr>
<tr>
<td>HGP</td>
<td>m</td>
<td>RD</td>
<td>Distance from shield to SPICE ground</td>
</tr>
<tr>
<td>RHO</td>
<td>ohm·m</td>
<td>17E-9</td>
<td>Resistivity of conductor material – defaults to value for copper</td>
</tr>
<tr>
<td>RHOB</td>
<td>ohm·m</td>
<td>rho</td>
<td>Resistivity of shield material</td>
</tr>
<tr>
<td>SIG</td>
<td>mho/m</td>
<td>0.0</td>
<td>Conductivity of dielectric</td>
</tr>
<tr>
<td>KD</td>
<td>—</td>
<td>4.0</td>
<td>Relative dielectric constant of dielectric</td>
</tr>
<tr>
<td>CMULT</td>
<td>—</td>
<td>1.0</td>
<td>Multiplier (used in defining CPR) for dielectric constant of material between shield and SPICE ground when LLEV=1 and CEXT is not present</td>
</tr>
<tr>
<td>CEXT</td>
<td>F/m</td>
<td>und.</td>
<td>External capacitance from shield to SPICE ground – used only to override HGP and CMULT computation</td>
</tr>
<tr>
<td>SHTHK</td>
<td>m</td>
<td>2.54E-4</td>
<td>Coaxial shield conductor thickness</td>
</tr>
</tbody>
</table>
Lossy U Model Parameters Geometric Twinlead (PLEV=3, ELEV=1)

Figure C-16: Geometric Embedded Twinlead, DLEV=0, Sea of Dielectric

Figure C-17: Geometric Twinlead, DLEV=1, with Insulating Spacer
### Geometric Twinlead Parameters (ELEV=1)

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| DLEV         |       | 0.0     | 0: embedded twinlead  
1: spacer twinlead  
2: shielded twinlead |
| RA1          | m     | req.    | Outer radius of each conductor |
| D12          | m     | req.    | Distance between the conductor centers |
| RHO          | ohm·m | 17E-9   | Resistivity of first conductor material – defaults to value for copper |
| KD           |       | 4.0     | Relative dielectric constant of dielectric |
| SIG          | mho/m | 0.0     | Conductivity of dielectric |
| HGP          | m     | d12     | Distance to reference plane |
| CMULT        |       | 1.0     | Multiplier {used in defining CPR} for dielectric constant of material between reference plane and SPICE ground when LLEV=1 and CEXT is not present. |
| CEXT         | F/m   | undef.  | External capacitance from reference plane to SPICE ground point (overrides LRR when present) |
| TS1          | m     | req.    | Insulation thickness on first conductor |
The following parameters apply to shielded twinlead:

<table>
<thead>
<tr>
<th>Name (Alias)</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS2</td>
<td>m</td>
<td>TS1</td>
<td>Insulation thickness on second conductor</td>
</tr>
<tr>
<td>TS3</td>
<td>m</td>
<td>TS1</td>
<td>Insulation thickness of spacer between conductor</td>
</tr>
<tr>
<td>RHOB</td>
<td>ohm·m</td>
<td>rho</td>
<td>Resistivity of shield material (if present)</td>
</tr>
<tr>
<td>OD1</td>
<td>m</td>
<td>req.</td>
<td>Maximum outer dimension of shield</td>
</tr>
<tr>
<td>SHTHK</td>
<td>m</td>
<td>2.54E-4</td>
<td>Twinlead shield conductor thickness</td>
</tr>
</tbody>
</table>
Precomputed Model Parameters (ELEV=2)

Precomputed parameters allow the specification of up to five signal conductors and a reference conductor. These parameters may be extracted from a field solver, laboratory experiments, or packaging specifications supplied by vendors. The parameters supplied include:

- Capacitance/length. Each conductor has a capacitance to all other conductors.
- Conductance/length. Each conductor has a conductance to all other conductors due to dielectric leakage.
- Inductance/length. Each conductor has a self inductance and mutual inductances to all other conductors in the transmission line.
- Resistance/length. Each conductor has two resistances, high frequency resistance due to skin effect and bent wires and DC core resistance.

Figure C-19 identifies the precomputed components for a three-conductor line with a reference plane. The Star-Hspice names for the resistance, capacitance, and conductance components for up to five lines are shown in Figure C-20.

**Figure C-19: Precomputed Components for Three Conductors and a Reference Plane**

![Precomputed Components Diagram](image)
Figure C-20: ELEV=2 Model Keywords for Conductor PLEV=1

<table>
<thead>
<tr>
<th>Ref. plane</th>
<th>line 1</th>
<th>line 2</th>
<th>line 3</th>
<th>line 4</th>
<th>line 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSPICE ground</td>
<td>CPR</td>
<td>CP1</td>
<td>CP2</td>
<td>CP3</td>
<td>CP4</td>
</tr>
<tr>
<td></td>
<td>GPR</td>
<td>GP1</td>
<td>GP2</td>
<td>GP3</td>
<td>GP4</td>
</tr>
<tr>
<td>Ref. plane</td>
<td>RRR</td>
<td>CR1</td>
<td>CR2</td>
<td>CR3</td>
<td>CR4</td>
</tr>
<tr>
<td></td>
<td>LRR</td>
<td>GR1</td>
<td>GR2</td>
<td>GR3</td>
<td>GR4</td>
</tr>
<tr>
<td>line 1</td>
<td>R11</td>
<td>L11</td>
<td>C12</td>
<td>C13</td>
<td>C14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>G12</td>
<td>G13</td>
<td>G14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L12</td>
<td>L13</td>
<td>L14</td>
</tr>
<tr>
<td>line 2</td>
<td>R22</td>
<td>L22</td>
<td>C23</td>
<td>C24</td>
<td>C25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>G23</td>
<td>G24</td>
<td>G25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L23</td>
<td>L24</td>
<td>L25</td>
</tr>
<tr>
<td>line 3</td>
<td>R33</td>
<td>L33</td>
<td>C34</td>
<td>C35</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>G34</td>
<td>G35</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L34</td>
<td>L35</td>
<td></td>
</tr>
<tr>
<td>line 4</td>
<td>R44</td>
<td>L44</td>
<td>C45</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>G45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>line 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L55</td>
</tr>
</tbody>
</table>

All precomputed parameters default to zero except CEXT, which is not used unless it is defined. The units are standard MKS in every case, namely:
- capacitance F/m
- inductance H/m
- conductance mho/m
- resistance ohm/m
Three additional parameters, LLEV (which defaults to 0), CEXT, and GPR are described below.

- LLEV=0. The reference plane conductor is resistive only (the default).
- LLEV=1. Reference plane inductance is included, as well as common mode inductance and capacitance to SPICE ground for all conductors.
- CEXT. External capacitance from the reference plane to SPICE ground. When CEXT is specified, it overrides CPR.
- GPR. Conductance to circuit ground; is zero except for immersion in a conductive medium.

Conductor Width Relative to Reference Plane Width

For the precomputed lossy U model (ELEV=2), the conductor width must be smaller than the reference plane width, which makes the conductor inductance smaller than the reference plane inductance. If the reference plane inductance is greater than the conductor inductance, Star-Hspice reports an error.

Alternative Multiconductor Capacitance/Conductance Definitions

Three different definitions of capacitances and conductances between multiple conductors are currently used. In this manual, relationships are written explicitly only for various capacitance formulations, but they apply equally well to corresponding conductance quantities, which are electrically in parallel with the capacitances. The symbols used in this section, and where one is likely to encounter these usages, are:

- CXY: branch capacitances, Star-Hspice input, and circuit models.
- Cjk: Maxwell matrices for capacitance, multiple capacitor stamp for MNA (modified nodal admittance) matrix, which is a SPICE (and Star-Hspice) internal. Also the output of some field solvers.
- CX: capacitance with all conductors except X grounded. The output of some test equipment.
- GXY, Gjk, GX: conductances corresponding to above capacitances
The following example uses a multiple conductor capacitance model, a typical Star-Hspice U model transmission line. The U Element supports up to five signal conductors plus a reference plane, but the three conductor case, Figure C-21, demonstrates the three definitions of capacitance. The branch capacitances are given in Star-Hspice notation.

**Figure C-21: Single-Lump Circuit Capacitance**

The branch and Maxwell matrices are completely derivable from each other. The “O.C.G.” (“other conductors grounded”) matrix is derivable from either the Maxwell matrix or the branch matrix. Thus:

\[
CX = \sum_{X \neq Y} C_{XY}
\]

\[
C_{jk} = CX \text{ on diagonal} = -C_{XY} \text{ off diagonal}
\]
The matrices for the example given above provide the following “O.C.G.” capacitances:

\[
C_1 = CR_1 + C_{12} + C_{13} \\
C_2 = CR_2 + C_{12} + C_{23} \\
C_3 = CR_3 + C_{13} + C_{23} \\
CR = CR_1 + CR_2 + CR_3 + CPR
\]

Also, the Maxwell matrix is given as:

\[
C_{jk} = \begin{bmatrix}
CR & -CR_1 & -CR_2 & -CR_3 \\
-CR_1 & C_1 & -C_{12} & -C_{13} \\
-CR_2 & -C_{12} & C_2 & -C_{23} \\
-CR_3 & -C_{13} & -C_{23} & C_3
\end{bmatrix}
\]

The branch capacitances also may be obtained from the Maxwell matrices. The off-diagonal terms are the negative of the corresponding Maxwell matrix component. The branch matrix terms for capacitance to circuit ground are the sum of all the terms in the full column of the maxwell matrix, with signs intact:

\[
CPR = \sum (C_{jk}), j=R, k=R:3 \\
CP_1 = \sum (C_{1k}), j=1, k=R:3 \\
CP_2 = \sum (C_{2k}), j=2, k=R:3 \\
CP_3 = \sum (C_{3k}), j=3, k=R:3
\]

CP1, ... CP5 are not computed internally with the Star-Hspice geometric (ELEV=1) option, although CPR is. This, and the internally computed inductances, are consistent with an implicit assumption that the signal conductors are completely shielded by the reference plane conductor. This is true, to a high degree of accuracy, for stripline, coaxial cable, and shielded twinlead, and to a fair degree for MICROSTRIP. If accurate values of CP1 and so forth are available from a field solver, they can be used with ELEV=2 type input.
If the currents from each of the other conductors can be measured separately, then all of the terms in the Maxwell matrix may be obtained by laboratory experiment. By setting all voltages except that on the first signal conductor equal to 0, for instance, you can obtain all of the Maxwell matrix terms in column 1.

\[
\begin{bmatrix}
  IR \\
  I1 \\
  I2 \\
  I3
\end{bmatrix} =
\begin{bmatrix}
  CR & -CR1 & -CR2 & -CR3 \\
  -CR1 & C1 & -C12 & -C13 \\
  -CR2 & -C12 & C2 & -C23 \\
  -CR3 & -C13 & -C23 & C3
\end{bmatrix} \cdot jw \cdot
\begin{bmatrix}
  0.0 \\
  1.0 \\
  0.0 \\
  0.0
\end{bmatrix} = jw \cdot
\begin{bmatrix}
  -CR1 \\
  C1 \\
  -C12 \\
  -C13
\end{bmatrix}
\]

The advantage of using branch capacitances for input derives from the fact that only one side of the off-diagonal matrix terms are input. This makes the input less tedious and provides fewer opportunities for error.

**Measured Parameters (ELEV=3)**

When measured parameters are specified in the input, the program calculates the resistance, capacitance, and inductance parameters using TEM transmission line theory with the LLEV=0 option. If redundant measured parameters are given, the program recognizes the situation, and discards those which are usually presumed to be less accurate. For twinlead models, PLEV=3, the common mode capacitance is one thousandth of that for differential-mode, which allows a reference plane to be used.

The ELEV=3 model is limited to one conductor and reference plane for PLEV=1.
### Basic ELEV=3 Parameters

<table>
<thead>
<tr>
<th>Name/Alias</th>
<th>Units</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| PLEV       |       | 1: planar  
             |       | 2: coax     
             |       | 3: twinlead |
| ZK         | ohm   | calc    | Characteristic impedance |
| VREL       |       | calc    | Relative velocity of propagation (delen / (delay * clight)) |
| DELAY      | sec   | calc    | Delay for length delen |
| CAPL       |       | 1.0     | Linear capacitance in length clen |
| AT1        |       | 1.0     | Attenuation factor in length atlen. Use dB scale factor when specifying attenuation in dB. |
| DELEN      | m     | 1.0     | Unit of length for delay (for example, ft.) |
| CLEN       | m     | 1.0     | Unit of length for capacitance |
| ATLEN      | m     | 1.0     | Unit of length for attenuation |
| FR1        | Hz    | req.    | Frequency at which AT1 is valid. Resistance is constant below FR1, and increases as √(frequency) above FR1. |

### Parameter Combinations

You can use several combinations of measured parameters to compute the L and C values used internally. The full parameter set is redundant. If you input a redundant parameter set, the program discards those that are presumed to be less accurate. Table C-5 on page 37 shows how each of seven possible parameter combinations are reduced, if need be, to a unique set and then used to compute C and L.
Three different delays are used in discussing Star-Hspice transmission lines:

**DELAY**  
U model input parameter that is the delay required to propagate a distance “dlen”

**TD**  
T Element input parameter signifying the delay required to propagate one meter

**TD\_eff**  
internal variable, which is the delay required to propagate the length of the transmission line T Element or U Element.

### Table C-5: Lossless Parameter Combinations

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Basis of Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ZK, \text{DELAY}, \text{DELEN}, \text{CAPL}, \text{CLEN})</td>
<td>redundant. Discard CAPL and CLEN.</td>
</tr>
<tr>
<td>(ZK, \text{VREL}, \text{CAPL}, \text{CLEN})</td>
<td>redundant. Discard CAPL and CLEN.</td>
</tr>
<tr>
<td>(ZK, \text{DELAY}, \text{DELEN})</td>
<td>(\text{VREL}=\text{DELEN}/(\text{DELAY} \cdot \text{CLIGHT}))</td>
</tr>
<tr>
<td>(ZK) and (\text{VREL})</td>
<td>(C=1/(ZK \cdot \text{VREL} \cdot \text{CLIGHT}))</td>
</tr>
<tr>
<td></td>
<td>(L=ZK/(\text{VREL} \cdot \text{CLIGHT}))</td>
</tr>
<tr>
<td>(ZK, \text{CAPL}, \text{CLEN})</td>
<td>(C=\text{CAPL}/\text{CLEN})</td>
</tr>
<tr>
<td></td>
<td>(L=C \cdot ZK^2)</td>
</tr>
<tr>
<td>(\text{CAPL}, \text{CLEN}, \text{DELAY}, \text{DELEN})</td>
<td>(\text{VREL}=\text{DELEN}/(\text{DELAY} \cdot \text{CLIGHT}))</td>
</tr>
<tr>
<td>(\text{CAPL}, \text{CLEN}, \text{VREL})</td>
<td>(L(C)=\text{CAPL}/\text{CLEN})</td>
</tr>
<tr>
<td></td>
<td>(L(L)=1/(C \cdot \text{VREL}^2 \cdot \text{CLIGHT}^2))</td>
</tr>
</tbody>
</table>

### Loss Factor Input

The attenuation per unit length may be specified either as an attenuation factor or as a decibel attenuation. In order to allow for the fact that the data may be available either as input/output or output/input, decibels greater than 0, or factors greater than 1 are assumed to be input/output. The following example shows the four ways that one may specify that an input of 1.0 is attenuated to an output of 0.758.
The attenuation factor is used to compute the exponential loss parameter and linear resistance.

\[
\alpha = \frac{\ln(v_{in}/v_{out})}{AT_{lin}}
\]

\[
LR = 2 \cdot \alpha \cdot \sqrt{LL/LC}
\]

**Table C-6: Input Attenuation Variations**

<table>
<thead>
<tr>
<th>AT1 Input</th>
<th>Computation of attenuation factor and linear resistance</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT1 = -2.4dB</td>
<td>(v_{out}/v_{in} = 0.758 = 10^{(-AT1/20)}) (for dB &lt; 0)</td>
</tr>
<tr>
<td>AT1 = +2.4dB</td>
<td>(v_{out}/v_{in} = 0.758 = 10^{(+AT1/20)}) (for dB &gt; 0)</td>
</tr>
<tr>
<td>AT1 = 1.318</td>
<td>(v_{out}/v_{in} = 0.758 = 1/AT1) (for AT1 &lt; 1)</td>
</tr>
<tr>
<td>AT1 = 0.758</td>
<td>(v_{out}/v_{in} = 0.758 = AT1) (for AT1 &gt; 1)</td>
</tr>
</tbody>
</table>

U Element Examples

The following examples show the results of simulating a stripline geometry using the U model in a PCB scale application and in an IC scale application.

**Three Coupled Lines, Stripline Configuration**

Figure C-22 shows three coupled lines in a stripline configuration on an FR4 printed circuit board. A simple circuit using three coupled striplines is shown in Figure C-23.
The Star-Hspice input file for the simulation is shown below.

```
* Stripline circuit
.Tran 50ps 7.5ns
.Options Post NoMod Accurate Probe Method=Gear
VIN 12 0 PWL 0 0v 250ps 0v 350ps 2v
L1 14 11 2.5n
C1 14 0 2p
Tin 14 0 10 0 ZO=50 TD=0.17ns
Tfix 13 0 11 0 ZO=45 TD=500ps
RG 12 13 50
RLD1 7 0 50
C2 1 0 2p
U1 3 10 2 0 5 1 4 0 USTRIP L=0.178
T6 2 0 6 0 ZO=50 TD=0.17ns
T7 1 0 7 0 ZO=50 TD=0.17ns
```
Performing Interconnect Simulation

Figures C-24, C-25, and C-26 show the main line and crosstalk responses. The rise time and delay of the waveform are sensitive to the skin effect frequency, since losses reduce the slope of the signal rise. The main line response shows some differences between simulation and measurement. The rise time differences are due to layout parasitics and the fixed resistance model of skin effect. The differences between measured and simulated delays are due to errors in the estimation of dielectric constant and the probe position.

**Figure C-24: Measured versus Computed Through-Line Response**

![Graph showing measured versus computed through-line response]
The gradual rise in response between 3 ns and 4 ns is due to skin effect. During this period, the electric field driving the current penetrates farther into the conductor so that the current flow increases slightly and gradually. This affects the measured response as shown for the period between 3 ns and 4 ns.

Figure C-25 shows the backward crosstalk response. The amplitude and delay of this backward crosstalk are very close to the measured values. The risetime differences are due to approximating the skin effect with a fixed resistor, while the peak level difference is due to errors in the LC matrix solution for the coupled lines.

**Figure C-25: Measured Versus Computed Backward Crosstalk Response**
Figure C-26 shows the forward crosstalk response. This forward crosstalk shows almost complete signal cancellation in both measurement and simulation. The forward crosstalk levels are about one tenth the backward crosstalk levels. The onset of ringing of the forward crosstalk has reasonable agreement between simulation and measurement. However, the trailing edge of the measured and simulated responses differ. The measured response trails off to zero after about 3 ns, while the simulated response does not trail down to zero until 6 ns. Errors in simulation at this voltage level can easily be due to board layout parasitics that have not been included in the simulation.
Simulation methods can have a significant effect on the predicted waveforms. Figure C-27 shows the main line response at Node 7 of Figure C-23 as the integration method and the number of lumped elements change. With the recommended number of lumps, 20, the Trapezoidal integration method shows a fast risetime with ringing, while the Gear integration method shows a fast risetime and a well damped response. When the number of lumped elements is changed to 3, both Trapezoidal and Gear methods show a slow risetime with ringing. In this situation, the Gear method with 20 lumps gives the more accurate simulation.

Figure C-27: Computed Responses for 20 Lumps and 3 Lumps, Gear and Trapezoidal Integration Methods
**Three Coupled Lines, Sea of Dielectric Configuration**

This example shows the U Element analytic equations for a typical integrated circuit transmission line application. Three 200µm-long aluminum wires in a silicon dioxide dielectric are simulated to examine the through-line and coupled line response.

The Star-Hspice U model uses the transmission line geometric parameters to generate a multisection lumped-parameter transmission line model. Star-Hspice uses a single U Element statement to create an internal network of three 20-lump circuits.

Figure C-28 shows the IC-scale coupled line geometry.

**Figure C-28: Three Coupled Lines with One Reference Plane in a Sea of Dielectric (IC Scale)**

![Diagram of coupled lines](image)

Figure C-29 shows one lump of the lumped-parameter schematic for the three-conductor stripline configuration of Figure C-28. This is the internal circuitry Star-Hspice creates to represent one U Element instantiation. The internal elements are described in “Star-Hspice Output” on page C-49.
Figure C-29: Schematic for Three Coupled Lines with One Reference Plane

Figure C-30 shows a schematic using the U Element of Figure C-29. In this simple circuit, a pulse drives a three-conductor transmission line source terminated by 50 Ω resistors and loaded by 1pF capacitors.

Figure C-30: Schematic Using the Three Coupled Lines U Model

The Star-Hspice input file for the U Element solution is shown below.
.Tran 0.1ns 20ns
.Options Post Accurate NoMod Brief Probe
Vss Vss 0 0v
Rss Vss 0 1x
vIn1 In1 Vss Pwl 0ns 0v 11ns 0v 12ns 5v 15ns 5v 16ns 0v
rIn1 In1 In10 50
rIn2 Vss In20 50
rIN3 Vss In30 50
u1 In10 In20 In30 Vss Out1 Out2 Out3 Vss IcWire L=200um
cIn1 Out1 Vss 1pF
cIn2 Out2 Vss 1pF
cIn3 Out3 Vss 1pF
.Probe v(Out1) v(Out2) v(Out3)
.Model IcWire U LEVEL=3 Dlev=0 Nl=3 Nlay=2 Plev=1 Elev=1
+ Llev=0 Ht=2u Wd=5u Sp=15u Th=1u Rho=2.8e-8 Kd=3.9
.End

The Star-Hspice U Element uses the conductor geometry to create length-independent RLC matrices for a set of transmission lines. You can then input any length, and Star-Hspice computes the number of circuit lumps that are required.

Figure C-31, Figure C-32, and Figure C-33 show the through and coupled responses computed by Star-Hspice using the U Element equations.
Figure C-31: Computed Through-Line Response

Figure C-32 shows the nearest coupled line response. This response only occurs during signal transitions.
Figure C-32: Computed Nearest Coupled-Line Response

Figure C-33 shows the third coupled line response. The predicted response is about 1/100000 of the main line response.
By default, Star-Hspice prints the model values, including the LCRG matrices, for the U Element. All of the LCRG parameters printed by Star-Hspice are identified in the following section.

**Star-Hspice Output**

The listing below is part of the Star-Hspice output from a simulation using the HSPICE input deck for the IcWire U model. Descriptions of the parameters specific to U Elements follows the listing. (Parameters not listed in this section are described in Tables C-2 and C-3.)
IcWire Output Section

*** model name: 0:icwire ****

<table>
<thead>
<tr>
<th>names</th>
<th>values</th>
<th>units</th>
</tr>
</thead>
<tbody>
<tr>
<td>ml</td>
<td>20.00</td>
<td></td>
</tr>
<tr>
<td>pl</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>nl</td>
<td>3.00</td>
<td></td>
</tr>
</tbody>
</table>

--- u-model control parameters ---

maxl= 20.00  #lumps
plev= 1.00
nlay= 2.00
nl= 3.00

--- begin type specific parameters ---

dlev= 0.  #
sig= 0.  mho/m
xw= 0.  meter
th= 1.00u meter
skin= 10.31u meter
th2= 1.00u meter
ht3= 2.00u meter
cr1= 170.28p f/m
g12= 649.53f f/m
g23= 0.  mho/m
r1s= 5.60k ohm/m
r3s= 5.60k ohm/m

--- two layer (skin and core) parameters ---

rrs= 1.12k ohm/m
r1c= 0.  ohm/m
r2c= 0.  ohm/m
r3c= 0.  ohm/m

U Element Parameters

$C_{ij}$ coupling capacitance from conductor $i$ to conductor $j$ (positive)

$C_{rj}$ self capacitance/m of conductor $j$ to the reference plane

$C_{pr}$ capacitance of the reference plane to the HSPICE ground

$G_{ij}$ conductance/m from conductor $i$ to conductor $j$ (zero if $\text{sig}=0$)

$G_{rj}$ conductance/m from conductor $j$ to the reference plane (0 if $\text{sig}=0$)

$G_{pr}$ conductance from the reference plane to the HSPICE ground, always=0

$h_{ti}$ height of conductor $i$ above the reference plane (only $h_{t}$ is input, all heights are the same)
Ideal and Lumped Transmission Lines

Performing Interconnect Simulation

The total conductor resistance is indicated by $r_{ij}$ when NLAY = 1, or by $r_{is} + r_{ic}$ when NLAY = 2.

As shown in the next section, some difference between HSPICE and field solver results is to be expected. Within the range of validity shown in Table C-4 for the Star-Hspice model, Star-Hspice comes very close to field solver accuracy. In fact, discrepancies between results from different field solvers can be as large as

$l_{ri}$ inductance/m from conductor $i$ to the reference plane

$l_{rr}$ inductance/m of the reference plane

$l_{ij}$ inductance/m from conductor $i$ to conductor $j$

$l_{jj}$ self inductance/m of conductor $j$

$r_{rc}$ core resistance/m of the reference plane (if NLAY = 2, zero of skin depth > 90% of thb)

$r_{rr}$ resistance/m of the reference plane (if NLAY = 1)

$r_{rs}$ skin resistance/m of the reference plane (if NLAY = 2)

$r_{is}$ skin resistance/m of conductor $i$ (if NLAY = 2)

$r_{ic}$ core resistance/m of conductor $i$ (if NLAY = 2, zero if skin depth > 50% of th)

$r_{jj}$ resistance/m of conductor $j$ (if NLAY = 1)

$skin$ skin depth

$skinb$ skin depth of the reference plane

$spi$ spacing between conductor $i$ and conductor $i+1$ (only sp is input, all spacings are the same)

$thi$ thickness of conductor $i$ (only th is input, all thicknesses are the same)

$wdi$ width of conductor $i$ (only wd is input – all widths are the same)
their discrepancies with Star-Hspice. The next section compares some Star-Hspice physical models to models derived using field solvers.

**Capacitance and Inductance Matrices**

Star-Hspice places capacitance and inductance values for U Elements in matrix form, for example:

\[
\begin{bmatrix}
C_{ii} & \ldots & C_{ji} \\
\vdots & \ddots & \vdots \\
C_{ij} & \ldots & C_{jj}
\end{bmatrix}
\]

Figure C-7 shows the capacitance and inductance matrices for the three-line, buried microstrip IC-scale example shown in Figure C-28.

**Table C-7: Capacitance and Inductance Matrices for the Three-Line, IC-Scale Interconnect System**

<table>
<thead>
<tr>
<th>Capacitance (pF/m)</th>
<th>176</th>
<th>-5.02</th>
<th>-0.65</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-5.02</td>
<td>178</td>
<td>-5.02</td>
</tr>
<tr>
<td></td>
<td>-0.65</td>
<td>-5.02</td>
<td>176</td>
</tr>
<tr>
<td>Inductance (nH/m)</td>
<td>246</td>
<td>6.97</td>
<td>1.11</td>
</tr>
<tr>
<td></td>
<td>6.97</td>
<td>243</td>
<td>6.97</td>
</tr>
<tr>
<td></td>
<td>1.11</td>
<td>6.97</td>
<td>246</td>
</tr>
</tbody>
</table>

The capacitance matrices in Table C-7 are based on the admittance matrix of the capacitances between the conductors. The negative values in the capacitance matrix are due to the sign convention for admittance matrices. The inductance matrices are based on the impedance matrices of the self and mutual inductance of the conductors. Each matrix value is per meter of conductor length. The actual lumped values used by Star-Hspice would use a conductor length equal to the total line length divided by the number of lumps.

The above capacitance matrix can be related directly to the Star-Hspice output of Example 2. Star-Hspice uses the branch capacitance matrix for internal
calculations. For the three-conductors in this example, Figure C-34 shows the equivalent capacitances, in terms of Star-Hspice parameters.

**Figure C-34: Conductor Capacitances for Example 2**

The capacitances of Figure C-34 are those shown in Figure C-29. The HSPICE nodal capacitance matrix of Figure C-7 is shown below, using the capacitance terms that are listed in the HSPICE output.

\[
\begin{bmatrix}
C_{R1} + C_{12} & -C_{12} & -C_{13} \\
-C_{12} & C_{R2} + C_{12} & -C_{23} \\
-C_{13} & -C_{23} & C_{R3} + C_{13}
\end{bmatrix}
\]

The off-diagonal terms are the negative of the coupling capacitances (to conform to the sign convention). The diagonal terms require some computation, for example,

\[C_{11} = C_{R1} + C_{12} + C_{13}\]
\[= 170.28 + 5.02 + 0.65\]
\[= 175.95, \text{ or } 176 \text{ pF/m}\]
Note that the matrix values on the diagonal in Figure C-7 are large: they indicate self-capacitance and inductance. The diagonal values show close agreement among the various solution methods. As the coupling values become small compared to the diagonal values, the various solution methods give very different results. Third-line coupling capacitances of 0.65 pF/m, 1.2 pF/m, and 0.88 pF/m are shown in Table C-7. Although the differences between these coupling capacitances seem large, they represent a negligible difference in waveforms because they account for only a very small amount of voltage coupling. Table C-7 represents very small coupling because the line spacing is large (about seven substrate heights).

Table C-8 shows the parameters for the stripline shown in Figure C-35.

**Figure C-35: Stripline Geometry Used in MCM Technology**

![Stripline Geometry](image)

**Table C-8: Capacitance and Inductance for the Single Line MCM-Scale Stripline**

<table>
<thead>
<tr>
<th>Capacitance (pF/m)</th>
<th>164.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inductance (nH/m)</td>
<td>236.5</td>
</tr>
</tbody>
</table>

**Five Coupled Lines, Stripline Configuration**

This example shows a five-line interconnect system in a PCB technology. Table C-9 shows the matrix parameters for the line configuration of Figure C-36.
Figure C-36: Five Coupled Lines on a PCB

Table C-9: Capacitance and Inductance for the Five-Line PCB-Scale Interconnect System

<table>
<thead>
<tr>
<th>Capacitance (pF/m)</th>
<th>59</th>
<th>-19</th>
<th>-2.5</th>
<th>-0.8</th>
<th>-0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-19</td>
<td>69</td>
<td>-18</td>
<td>-2.2</td>
<td>-0.8</td>
</tr>
<tr>
<td></td>
<td>-2.5</td>
<td>-18</td>
<td>69</td>
<td>-18</td>
<td>-2.5</td>
</tr>
<tr>
<td></td>
<td>-0.8</td>
<td>-2.2</td>
<td>-18</td>
<td>69</td>
<td>-19</td>
</tr>
<tr>
<td></td>
<td>-0.4</td>
<td>-0.8</td>
<td>-2.5</td>
<td>-19</td>
<td>57</td>
</tr>
<tr>
<td>Inductance (nH/m)</td>
<td>676</td>
<td>309</td>
<td>179</td>
<td>116</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>309</td>
<td>641</td>
<td>297</td>
<td>174</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>179</td>
<td>297</td>
<td>637</td>
<td>297</td>
<td>179</td>
</tr>
<tr>
<td></td>
<td>116</td>
<td>174</td>
<td>297</td>
<td>641</td>
<td>309</td>
</tr>
<tr>
<td></td>
<td>81</td>
<td>116</td>
<td>179</td>
<td>309</td>
<td>676</td>
</tr>
</tbody>
</table>

U Model Applications

This section gives examples of use, and then explains some of the aspects of ringing (impulse-initiated oscillation) in real and simulated transmission line circuits.

Data Entry Examples

Coax Geometry Entry (ELEV=1,PLEV=2) with ground reference (LLEV=1) and skin effect (NLAY=2)

uc in1 3 out1 4 wire2 l=1
.model wire2 u LEVEL=3 nlay=2 plev=2 elev=1 Llev=1
Performing Interconnect Simulation Ideal and Lumped Transmission Lines

+ ra=1m rb=7.22m hgp=20m rho=1.7e-8 kd=2.5

Matrix Entry (ELEV=2)

u1 In1 In2 In3 Vss Out1 Out2 Out3 Vss Wire3 L=0.01
.model Wire3 U LEVEL=3 NL=3 Elev=2 Llev=0
+ rrr=1.12k r11=5.6k r22=5.6k r33=5.6k c13=0.879pF
+ cr1=176.4pF cr2=172.6pF cr3=176.4pF c12=4.7pF c23=4.7pF
+ L11=237nH L22=237nH L33=237nH L12=5.52nH L23=5.52nH
+ L13=1.34nH

Coax Measured Data Entry (ELEV=3, PLEV=2)

u10 1 0 2 0 rg58 l=12
.model rg58 u LEVEL=3 plev=2 elev=3
+ zk=50 capl=30.8pF clen=1ft vrel=0.66
+ frl=100meg atl=5.3db atlen=100ft

Printed Circuit Board Models

Figure C-37 illustrates a small cross section of a six-layer printed circuit board. The top and bottom signal layers require a microstrip U model (DLEV=1), while the middle signal layers use a stripline U model (DLEV=2). Important aspects of such a circuit board are the following:

- Trace impedance is difficult to control because of etch variation
- 6 mil effective trace widths
- 8 mil drawn widths
- 10 mil insulator thickness
- 1 ounce copper 1.3 mil thick
- Microstrip model TOP used for top and bottom
- Stripline model MID used for middle signal layers

Example

Top and bottom layer model:

*.MODEL TOP U LEVEL=3 ELEV=1 PLEV=1 TH=1.3mil HT=10mil KD=4.5 DLEV=1
+ WD=8mil XW=-2mil
Middle layer model:

```
.MMODEL MID U LEVEL=3 ELEV=1 PLEV=1 TH=1.3mil HT=10mil KD=4.5 DLEV=2
 + WD=8mil XW=-2mil TS=32mil
```

**Figure C-37: Six-Layer Printed Circuit Board**

Coax Models

The following examples are for standard coax. These are obtained from commonly available tables\(^1\). (The parameter fr1 is the frequency at which a specific amount of attenuation, at1, occurs for a specified length of coax, atlen.) Star-Hspice accepts dB (decibel) and ft. (foot) units.

**Example**

```
.model rg9/u u LEVEL=3 plev=2 elev=3
 + Zk=51 vrel=.66
 + fr1=100meg at1=2.1db atlen=100ft
*
.model rg9b/u u LEVEL=3 plev=2 elev=3
 + Zk=50 vrel=.66
 + fr1=100meg at1=2.1db atlen=100ft
*
.model rg11/u u LEVEL=3 plev=2 elev=3
 + Zk=75 vrel=.78
 + fr1=100meg at1=1.5db atlen=100ft
*
.model rg11a/u u LEVEL=3 plev=2 elev=3
 + Zk=75 vrel=.66
 + fr1=100meg at1=1.9db atlen=100ft
```
Performing Interconnect Simulation

Ideal and Lumped Transmission Lines

* .model rg54a/u u LEVEL=3 plev=2 elev=3
+ Zk=58 vrel=.66
+ frl=100meg atl=3.1db atlen=100ft
*

.model rg15/u u LEVEL=3 plev=2 elev=3
+ Zk=53.5 vrel=.66
+ frl=100meg atl=4.1db atlen=100ft
*

.model rg53/u u LEVEL=3 plev=2 elev=3
+ Zk=53.5 vrel=.66
+ frl=100meg atl=4.1db atlen=100ft
*

.model rg58a/u u LEVEL=3 plev=2 elev=3
+ Zk=50 vrel=.66
+ frl=100meg atl=5.3db atlen=100ft
*

.model rg58c/u u LEVEL=3 plev=2 elev=3
+ Zk=50 vrel=.66
+ frl=100meg atl=5.3db atlen=100ft
*

.model rg59b/u u LEVEL=3 plev=2 elev=3
+ Zk=75 vrel=.66
+ frl=100meg atl=3.75db atlen=100ft
*

.model rg62/u u LEVEL=3 plev=2 elev=3
+ Zk=93 vrel=.84
+ frl=100meg atl=3.1db atlen=100ft
*

.model rg62b/u u LEVEL=3 plev=2 elev=3

Twinlead Models

Example

.model tw/sh u LEVEL=3 plev=3 elev=3
* Shielded TV type twinlead
+ Zk=300 vrel=.698
+ frl=57meg atl=1.7db atlen=100ft
*

.model tw/un u LEVEL=3 plev=3 elev=3
* Unshielded TV type twinlead
+ Zk=300 vrel=.733
+ frl=100meg atl=1.4db atlen=100ft

Two Coupled Microstrips

Figure C-38 shows two metal lines formed of the first aluminum layer of a modern CMOS process. The microstrip model assumes that the metal strips sit on top of a dielectric layer that covers the reference plane.

Figure C-38: Two Coupled Microstrips Geometrically Defined as LSI Metallization
Example of Two Microstrips Coupled Together

* file strip2.sp Two microstrips coupled together
*.... The tests following use geometric/physical model
.option acct post list
.print tran V(no1) V(no2) V(noref)

Signal Source
.tran 1ps 300ps
.PARAM Rx=54
* excitation voltage + prefilter
V1 np1 0 PWL 0.0s 0v 50ps 0v 60ps 1v
xrI1 NP1 NI1 rcfilt rflt=rx tdflt=1ps

Circuit Definition
Ue1 NI1 NI2 0 NO1 NO2 NORef u1 L=5.0m
RI2 NI2 0 Rx
RO1 NO1 NORef Rx
RO2 NO2 NORef Rx
rref noref 0 1
* ...MODEL DEFINITION -- metal layer1 (sea of dielectric)
.MODEL u1 U LEVEL=3 plev=1 elev=1 nl=2
+ KD=3.5 xw=0.1u rho=17e-9 rhob=20e-9
+ wd=1.5u ht=1.0u th=0.6u sp=1.5u
+ llev=1 dlev=0 maxl=50
*
.END

Solving Ringing Problems with U Elements

Ringing oscillations at sharp signal edges may be produced by:

- Oscillations due to the simulator
- Oscillations due to lossy approximation of a transmission line (U Element)
- Signal reflections due to impedance mismatch

The primary reason for using a circuit simulator to measure high speed transmission line effects is to calculate how much transient noise the system contains and to determine how to reduce it to acceptable values.
Oscillations Due to Simulation Errors

The system noise results from the signal reflections in the circuit. It may be masked by noises from the simulator. Simulator noise must be eliminated in order to obtain reliable system noise estimates. The following sections describes ways to solve problems with simulator noise.

Timestep Control Error

The default method of integrating inductors and capacitors is trapezoidal integration. While this method gives excellent results for most simulations, it can lead to what is called trapezoidal ringing. This is numerical oscillations that look like circuit oscillations, but are actually timestep control failures. In particular, trapezoidal ringing can be caused by any discontinuous derivatives in the nonlinear capacitance models, or from the exponential charge expressions for diodes, BJTs, and JFETs.

Set the .OPTION METHOD=GEAR to change the integration method from trapezoidal to Gear. The gear method does not ring and, although it typically gives a slightly less accurate result, is still acceptable for transient noise analysis.

Incorrect Number of Element Lumps

It is important to use the right number of lumps in a lossy transmission line element. Too few lumps results in false ringing or inaccurate signal transmission, while too many lumps leads to an inordinately long simulation run. Sometimes, as in verification tests, it is necessary to be able to specify the number of lumps in a transmission line element directly. The number of lumps in an Star-Hspice lossy transmission line element may be directly specified, defaulted to an accuracy and limit based computation, or computed with altered accuracy and limit and risetime parameters.
**Default Computation**

In the default computation, LUMPS=1 until a threshold of total delay versus risetime is reached:

\[
T_{Deff} = \frac{RISETIME}{20}
\]

where: 
- \(T_{Deff}\) = total end-to-end delay in the transmission line element
- \(RISETIME\) = the duration of the shortest signal ramp, as given in the statement

\[.OPTION\ RISETIME = \text{value}\]

At the threshold, two lumps are used. Above the threshold, the number of lumps is determined by:

\[
\text{number of lumps} = \text{minimum of 20 or } [1+(T_{Deff}/RISETIME)*20]
\]

The upper limit of 20 is applied to enhance simulation speed.

If the standard accuracy-based computation does not provide enough lumps, or if it computes too many lumps for simulation efficiency, you can use one of several methods to change the number of lumps on one or more elements:

1. Specify LUMPS=value in the element statement.
2. Specify MAXL=value and WLUMP=value in the .MODEL statement.
3. Specify a different RISETIME=value in the .OPTION statement.

**Specify LUMPS=value (direct specification)**

Direct specification overrides the model and limit based computation, applying only to the element specified in the element statement, as in the example below:

```
U35 n1 gnd n2 oref    model lumps=31 L=5m
```

where 31 lumps are specified for an element of length 5 mm.
Specify MAXL and WLUMP (altered accuracy and limit parameters)

You can alter the default computation for all the elements that refer to a particular model by specifying the model parameters “MAXL” and “WLUMP” (which would otherwise default to 20). In the nondefault case the number of lumps, the threshold, and the upper limit all would be changed:

\[
lumps = \min\{\text{MAXL}, [1+(T_{\text{Deff}}/\text{RISETIME})\times\text{WLUMP}]\}
\]

Threshold: \(T_{\text{Deff}} = \frac{\text{RISETIME}}{\text{WLUMP}}\)

Upper lim: MAXL

Specify a different “RISETIME” parameter in the .OPTION statement

You can change the threshold and number of lumps computed for all elements of all models, reduce or increase the analysis parameter “RISETIME”. Note that care is required if RISETIME is decreased, because the number of lumps may be limited by MAXL in some cases where it was not previously limited.

Using a Multistage RC Filter to Prevent Ringing

Artificial sources such as pulse and piecewise linear sources often are used to simulate the action of real output buffer drivers. Since real buffers have a finite cutoff frequency, a multistage filter can be used to give the ideal voltage source reasonable impedance and bandwidth.

You can place a multistage RC filter, shown below, between the artificial source and any U Element to reduce the unrealistic source bandwidth and, consequently, the unrealistic ringing. In order to provide as much realism as possible, the interposed RC filter and the PWL (piecewise linear) source must be designed together to meet the following criteria:

- Reduce the ringing to acceptable levels
- Preserve the realistic bandwidth of the source signal
- Provide a driver with any chosen impedance
- Provide accurately timed transient signals
Example

.MACRO RCFNEW in out gnd_ref RFLT=50 TDFLT=100n
* 
.PROT
* Begin RCFILT.inc (RC filter) to smooth and match pulse
* sources to transmission lines. User specifies impedance
* (RFLT) and smoothing interval (TDFLT). TDFLT is usually
* specified at about .1*risetime of pulse source.
* 
* cutoff freq, total time delay; and frequency dependent
* impedance and signal voltage at "out" node:
* 
* smoothing period = TDFLT.....(equivalent boxcar filter)
* delay= TDFLT
* fc=2/(pi*TDFLT)...............(cuttoff frequency)
* Zo~ RFLT*(.9 + .1/sqrt( 1 + (f/fc)^2 ))
* V(out)/V(in)= [1 / sqrt( 1 + (f/fc)^2 )]^4
* 
.PARAM TD1S='TDFLT/4.0'
RF1 in n1 '.00009*RFLT'
RF2 n1 n2 '.0009*RFLT'
RF3 n2 n3 '.009*RFLT'
RF4 n3 n4 '.09*RFLT'
Rout n4 out '.90*RFLT'
*
.PARAM CTD='TD1S/(.9*RFLT)'
CF1 n1 0 '10000*CTD'
CF2 n2 0 '1000*CTD'
CF3 n3 0 '100*CTD'
CF4 n4 0 '10*CTD'

Figure C-39: Circuit Diagram of an RC Filter
From the comments embedded in the macro, the output impedance varies from RFLT in the DC limit to 3% less at FC and 10% less in the high frequency limit.

\[
R_{FLT|_{DC}} = 0.99999 \cdot R_{FLT} \\
R_{FLT|_{FC}} = 0.97 \cdot R_{FLT}
\]

Therefore, setting RFLT to the desired driver impedance gives a reasonably good model for the corrected driver impedance. TDFLT is generally set to 40% of the voltage source risetime.

* excitation voltage + prefilter
V1 np1 0 PWL 0.0s 0v 50ps 0v 60ps 1v
xrI1 NP1 NI1 RCFNEW rflt=rx tdflt=4ps

**Note:** RCFNEW is an automatic include file named $installdir/parts/behave/rcfilt.inc, where $installdir is the HSPICE installation directory.

Figure C-40 shows the input and output voltages for the filter.
Signal Reflections Due to Impedance Mismatch

The effect of impedance mismatch is demonstrated in the following example. This circuit has a 75 ohm driver, driving 3 inches of 8 mil wide PCB (middle layer), then driving 3 inches of 16 mil wide PCB.

The operational characteristics of such a circuit are shown in Figures C-41, C-42, and C-43. The first steady value of impedance is 75 ohms, which is the impedance of the first transmission line section. The input impedance falls to 56 ohms after about 2.5ns, when the negative reflection from the nx1 node reaches nil. This TDR displays one idiosyncrasy of the U Element. The high initial value of Zin(TDR) is due to the fact that the input element of the U Element is inductive. The initial TDR spike can be reduced in amplitude and duration by simply using a U Element with a larger number of lumped elements.
Figure C-41: Mathematical Transmission Line Structure

Figure C-42: Waveforms in Mismatched Transmission Line Structure
Input File for Impedance Mismatch Example

* file strip1.sp
* Two series microstrips (8mil wide & 16mil) drive 3 inches of 8mil
* middle layer PCB, series connected to 3 inches of 16mil wide
* middle layer PCB, 500ns risetime driver. The tests following use
* geometric/physical model

.option acct post list

Signal Source

.tran 20ps 4ns
.probe tdr=par(v(ni1)/i(ue1))
.PARAM R8mil=75 r16mil=56
* excitation voltage + prefilter
V1 np1 0 PWL 0.0s 0v 500ps 0v 1n 1v
xrI1 NP1 NI1 rcfilt rflt=r8mil tdflt=50ps
Circuit Definition
Ue1 NI1 0 NX1 0 u1 L=3000mil
Ue2 NX1 0 NO2 0 u2 L=3000mil
RO2 NO2 0 r16mil
* ...MODEL DEFINITION -- 8mil middle metal layer of a copper PCB
  .MODEL u1 U LEVEL=3 plev=1 elev=1 nl=1
  + th=1.3mil ht=10mil ts=32mil kd=4.5 dlev=0
  + wd=8mil xw=-2mil
* ...MODEL DEFINITION -- 16mil middle metal layer of a copper PCB
  .MODEL u2 U LEVEL=3 plev=1 elev=1 nl=1
  + th=1.3mil ht=10mil ts=32mil kd=4.5 dlev=0
  + wd=16mil xw=-2mil
.END
Understanding the Transmission Line Theory

This section:

■ Discusses how the discrete lumped model of the U Element transmission line explains characteristic impedance and transmission velocity
■ Uses the concepts of self and mutual inductance to explain crosstalk
■ Describes rules of thumb for various types of clock pulses
■ Discusses the sources of transmission line attenuation

Lossless Transmission Line Model

As a signal propagates down the pair of conductors, each new section acts electrically as a small lumped circuit element. In its simplest form, called the lossless model, the equivalent circuit of a transmission line has just inductance and capacitance. These elements are distributed uniformly down the length of the line, as shown in Figure C-44.

Figure C-44: Equivalent Circuit Model of a Lossless Transmission Line

From this electrical circuit model, the two important terms that characterize a transmission line can be derived: the velocity of a signal ($v$) and the characteristic impedance ($Z_0$).

$$v = \frac{1}{\sqrt{L_C C_L}}$$

and

$$Z_0 = \frac{L_C}{\sqrt{C_L}}$$
Ideal and Lumped Transmission Lines

Understanding the Transmission Line Theory

\[ L_L = \text{inductance per length} \]
\[ C_L = \text{capacitance per length} \]

This is the basis for the T Element used in Star-Hspice. It accounts for a characteristic impedance \((Z_0)\) and a time delay \((TD)\). The time delay depends on the distance \((d)\) between the two ends of the transmission line:

\[ TD = \frac{d}{v} \]

**Lossy Transmission Line Model**

When loss is significant, the effects of the series resistance \((R)\) and the dielectric conductance \((G)\) should be included. Figure C-45 shows the equivalent circuit model of a lossy transmission line, with distributed “lumps” of \(R, L,\) and \(C\) Elements.

**Figure C-45: Equivalent Circuit Model of a Lossy Transmission Line**

The U Element used in Star-Hspice is the equivalent circuit model for the lossy transmission line. In a transient simulation, the U Element automatically accounts for frequency-dependent characteristic impedance, dispersion (frequency dependence in the velocity), and attenuation.

The most common types of transmission line cross sections are microstrip, stripline, coax, wire over ground, and twisted pair. There is no direct relationship between cross section, velocity of propagation, and characteristic impedance.

In a balanced transmission line, the two conductors have similar properties and are electrically indistinguishable. For example, each wire of a twisted pair has the same voltage drop per length down the line. The circuit model for each wire has the same resistance, capacitance, and inductance per length.
Understanding the Transmission Line Theory

This is not the case with a microstrip line or a coaxial cable. In those structures, the signal conductor has a larger voltage drop per length than the other conductor. The wide reference plane in a microstrip or the larger diameter shield in a coax have lower resistance per length and lower inductance per length than the signal line. The equivalent circuit model for unbalanced lines typically assumes the resistance and inductance per length of the ground path is zero and all the voltage drop per length is on the signal conductor. Even though the inductance of the reference plane is small, it can play a significant role when there are large transient currents.

**Impedance**

The impedance of a device ($Z$) is defined as the instantaneous ratio of the voltage across the device ($V$) to the current through it:

$$ Z = \frac{V}{I} $$

**Impedance of Simple Lumped Elements**

The impedance of a device can be thought of as the quality of the device that causes it to transform a current through it into a voltage across it:

$$ V = ZI $$

The admittance ($Y$) is less often used to characterize a device. It is the inverse of the impedance:

$$ Y = \frac{1}{Z} = \frac{I}{V} $$

There are three ideal circuit elements used to describe passive components: a resistor, a capacitor, and an inductor. They are defined by how they interact with voltage across them and current though them:

Resistor, with resistance ($R$):

$$ V = IR $$
Capacitor, with capacitance (C):

\[ I = C \frac{dV}{dt} \]

Inductor with inductance (L):

\[ V = L \frac{dI}{dt} \]

When the voltage or current signals are time dependent, the impedance of a capacitive or inductive element is a very complicated function of time. You can simulate it with Star-Hspice, but it is difficult to build an intuitive model.

The impedance of a capacitor rotates the phase of the current 90° in the negative or direction to generate the voltage across the capacitor. The impedance of an inductor rotates the current 90° in the positive direction to generate the voltage across the inductor. For a resistor, the current and voltage have the same phase.

In the frequency domain, when all signals are sine waves in the time domain, the impedance of a capacitor and an inductor is frequency dependent, decreasing with frequency for a capacitor and increasing with frequency for an inductor. The impedance of a resistor is constant with frequency.

In the real world of finite dimensions and engineered materials, ideal circuit elements have parasitics associated with them, which cause them to behave in complex ways that are very apparent at high frequencies.

**Characteristic Impedance**

A controlled impedance transmission line is a pair of conductors that have a uniform cross section and uniform distribution of dielectric materials down their length. A short segment, \( \Delta x \), of the transmission line has a small capacitance associated with it, \( \Delta C \), which is the capacitance per length, \( C_L \), times the \( \Delta x \):

\[ \Delta C = C_L \Delta x \]

When a voltage signal is introduced at one end, the voltage between the conductors induces an electric that propagates the length of the line at the speed of light in the dielectric. As the voltage signal moves down the line, each new
section of line charges up. The new section of line, $\Delta x$, is charged up in a time $\Delta t$:

$$\Delta t = \frac{\Delta x}{v}$$

If the voltage ($V$) moves down the line at a constant speed and the capacitance per length is uniform throughout the line, then the constant voltage applied to the front end draws a constant charging current ($I$):

$$I = \frac{\Delta Q}{\Delta t} = \frac{\Delta CV}{\Delta t} = \frac{C_L \Delta x V}{\Delta t} = C_L v V$$

This constant voltage with constant current has the behavior of a constant impedance ($Z$):

$$Z = \frac{V}{I} = \frac{V}{C_L v V} = \frac{1}{C_L v}$$

The impedance is determined by the speed of the signal and the capacitance per length of the pair of conductors, both intrinsic properties of the line. This intrinsic impedance is termed the characteristic impedance of the line ($Z_0$).

If a measurement is made at one end of the line in a short time compared to the round trip time delay, the line behaves like a resistor with a resistance equal to the characteristic impedance of the line. Transmission line effects are only important when rise times are comparable or shorter than the round trip time delay.

For example, if the rise time of a device is 1 ns, and it drives an interconnect trace in FR4 which is longer than three inches, the load on the device during the risetime is purely resistive. For CMOS devices, which are used to drive high resistance loads, the typical 50 ohm resistance they see initially can significantly distort the waveform from what is expected.

It is only during the initial surge of the voltage that a transmission line behaves as a constant impedance, with a value equal to its characteristic impedance. For this reason the characteristic impedance of a line is also called the surge impedance. The surge time during which the impedance is constant is the round
trip time of flight, or twice the time delay. Reflections from the far end complicate the electrical behavior of the line after the surge time.

The instantaneous impedance measured at the front end of a transmission line is a complicated function of time. It depends on the nature of the terminations at the far end. When the line is shunted to ground with a resistor of value equal to the characteristic impedance of the line, there is no reflection back, and the front end of the line behaves as a resistive load. When the termination at the far end is open, the impedance at the front end starts out at the characteristic impedance and eventually, after multiple reflections, approaches an infinite impedance. During some periods the instantaneous impedance may be zero. These transient effects are fully simulated with T Elements and U Elements in Star-Hspice.

**Inductance**

**Mutual Inductance and Self Inductance**

The most confusing, subtle and important parameter in high-speed packaging and interconnect design is inductance. It plays a key role in the origin of simultaneous switching noise, also called common ground inductance, and a key role in crosstalk between transmission line structures.

**Operational Definition of Inductance**

Consider an inductor to be any section of circuit element which carries current: an interconnect trace, a ground plane, a TAB lead frame, a lead in a DIP package, the lead of a resistor or a pin in a connector. An inductor does not have to be a closed circuit path, but can be a small section of a circuit path.

A changing current passing through an inductor generates a voltage drop. The magnitude of the voltage drop (ΔV) for an inductance (L) and change in current (dI/dt) is:

$$\Delta V = L \frac{dI}{dt}$$

This definition can always be used to evaluate the inductance of a section of a circuit. For example, with two long parallel wires, each of radius (r) and a center-
to-center separation (s), you can measure the voltage drop per length for one of the wires when a changing current dI/dt flows through one wire and back through the other. The induced voltage per length on one of the wires is:

\[ V_L = \frac{\mu_0}{2\pi} \ln\left(\frac{s - r}{s}\right) \frac{dI}{dt} \text{ [V in mV/inch, I in mA, t in ns]} \]

From this expression, the effective inductance per length of one wire is found to be:

\[ L_L = \frac{\mu_0}{2\pi} \ln\left(\frac{s - r}{s}\right) \approx 5\ln\left(\frac{s}{r}\right) (s > r) \text{ [nH/inch]} \]

**Mutual Inductance**

A second effect also is important: the induced voltage from currents that are adjacent to, but not in, the same circuit path. This is caused by the mutual inductance between two current elements. A section of conductor in a circuit, labeled 1, may have an induced voltage generated across it because of currents not in circuit 1, but from circuits 2, 3, and 4.

The voltage generated across the section of circuit 1, \( V_1 \), is given in its general form by:

\[ V_1 = L_{11} \frac{dI_1}{dt} + L_{12} \frac{dI_2}{dt} + L_{13} \frac{dI_3}{dt} + L_{14} \frac{dI_4}{dt} \]

The notation for mutual inductance \( (L_{ab}) \) is related to the induced voltage on circuit element a, from the current element, b. In some texts, the symbol used is \( M \), rather than \( L \). The special case of the induced voltage on a circuit element from its own current \( (L_{aa}) \) is called self inductance.

Mutual inductance relates to the magnitude of induced voltage from an adjacent current. The magnitude of this voltage depends on the flux linkages between the two circuit elements.
**Self Inductance**

The self inductance of an isolated single trace is a well-defined, absolute mathematical quantity, but not a measurable physical quantity. There is always a return current path somewhere, and the mutual inductance from this return current path induces a voltage on the circuit element that subtracts from the self inductance. Self inductance can never be measured or isolated, independent of a mutual inductance of a return current path.

In the example above of two long parallel wires, the measured inductance per length \( (L_L) \) of one wire is neither the self inductance nor the mutual inductance of the wire. It is a combination of these two terms. If the universe contained just the two wires, the measured voltage drop per length would be:

\[
V_L = L_{11} \frac{dI_1}{dt} - L_{12} \frac{dI_1}{dt} = (L_{11} - L_{12}) \frac{dI_1}{dt} = L_{L} \frac{dI_1}{dt}
\]

The minus sign reflects the opposite directions of the currents \( I_1 \) and \( I_2 \). Operationally, when the inductance per length of one wire is measured, what is really being measured is the difference between its self inductance and the mutual inductance of the return path. Because of this effect, it is clear that the nature of the return path greatly influences the measured inductance of a circuit element.

**Reference Plane Return Paths**

The capacitance per length \( (C_L) \) of any planar transmission line is:

\[
C_L = \frac{85}{Z_0 \sqrt{\varepsilon_r}} \ [\text{pF/inch}]
\]

The inductance per length of the signal line \( (L_L) \) is:

\[
L_L = 0.085Z_0 \sqrt{\varepsilon_r} \ [\text{nH/inch}]
\]

This is the self inductance of the signal line, minus the mutual inductance of the return current in the reference plane.
For example, the inductance per length of a transmission line with characteristic impedance of 50 ohms in an FR4 printed circuit board is 9.6 nH/inch. The capacitance per length is 3.8 pF/inch. In the equivalent circuit of a lossless transmission line, the series inductance per length is 9.6 nH/inch, and the shunt capacitance to ground is 3.8 pF/inch.

In the notation of the U Element used in Star-Hspice, for an ELEV=2 (RCLK equivalent model) and a PLEV=1 (microstrip cross section), the parameters to model this lossless transmission line are

\[ C_{11} = 3.8 \text{ pF/inch}, \quad L_{11} = 9.6 \text{ nH/inch} \]

In the LLEV=0 parameter, Star-Hspice simplifies the inductance problem by automatically calculating the inductance of the line as the difference between the self inductance of the line and the mutual inductance of the return signal path.

In many texts, the term L11 is generically used as the self inductance. For LLEV=1, Star-Hspice assumes a circuit ground point separate from the reference plane of the transmission line. Thus the LLEV=1 option includes an approximation to the self inductance of both the signal conductor and the reference plane, while LLEV=0 assumes a reference plane return current.

**Crosstalk in Transmission Lines**

When there are adjacent transmission lines, for instance line 2 and line 3, the coupling capacitance and inductance between them and the quiet line, line 1, lead to crosstalk.

In the notation of Star-Hspice, the voltage per length on transmission line 1, \( V_1 \), including the mutual inductance to lines 2 and 3 is:

\[
V_1 = L_{11} \frac{dI_1}{dt} + L_{12} \frac{dI_2}{dt} + L_{13} \frac{dI_3}{dt}
\]

In the LLEV=0 case, Star-Hspice simplifies the inductance analysis by automatically including the effects of the return current path. The first inductance term (L11) is the inductance per length of the transmission line (1) including the self inductance of the line and the mutual inductance of the return ground path, as discussed in the previous section.
The second term, the coupling inductance of the second transmission line (L12), includes the mutual inductance of the second signal line and the mutual inductance of the return current path of the second line. Because these two currents are in opposite directions, the mutual inductance of the pair is much less than the mutual inductance of just the second signal trace alone.

The third term (LL13) includes the mutual inductance of the signal path in the third transmission line and the mutual inductance of its return path through the reference plane.

It is important to keep in mind that the coupling inductances (L12 and L13) include the mutual inductances of the adjacent signal lines and their associated return paths. They are more than the mutual inductance of the adjacent traces. In this sense, Star-Hspice deals with operational inductances, those that could be measured with a voltmeter and a dI/dt source.

**Risetime, Bandwidth, and Clock Frequency**

In the time domain, a clock waveform can be described in terms of its period (T\text{clock}), its frequency (F\text{clock}), and a risetime (τ\text{edge}). Figure C-46 illustrates these features.

![Figure C-46: Clock Waveform](image)

The risetime is typically defined by the time between the 10% to 90% points.

To describe this waveform in terms of sine wave components, the highest sine wave frequency required (the bandwidth, BW) depends on the risetime. As the bandwidth increases and higher sine wave frequency components are introduced, the risetime of the reconstructed waveform decreases. The
bandwidth of a waveform is determined by the fastest risetime it contains. The risetime and bandwidth are related by:

\[ \text{BW} = \frac{0.35}{\tau_{\text{edge}}} \quad \text{or} \quad \tau_{\text{edge}} = \frac{0.35}{\text{BW}} \]

The risetime of a clock waveform and the clock period are only indirectly related. The risetime of a system is determined by the output driver response and the characteristics of the packaging and interconnect. In general, the risetime is made as long as possible without degrading the clock period.

Without specific information about a system, it is difficult to know precisely what the risetime is, given just the clock frequency or period. In a fast system such as an oscillator with only one gate, the period might be two times the risetime:

\[ T_{\text{period}} \approx 2 \tau_{\text{edge}} \]

For a complex system such as a microprocessor board, the period might be 15 times as long as the risetime:

\[ T_{\text{period}} \approx 15 \tau_{\text{edge}} \]

In each case, the bandwidth is always related to the risetime by the first expression in this section, and the clock frequency and clock period are always related by:

\[ F_{\text{clock}} = \frac{1}{T_{\text{clock}}} \]

The example of the microprocessor would give the worst case of the shortest risetime for a given clock period. Combining these expressions shows the relationship between clock frequency and bandwidth:

\[ F_{\text{clock}} = \frac{1}{T_{\text{clock}}} = \frac{1}{15 \tau_{\text{edge}}} = \frac{\text{BW}}{15 \cdot 0.35} \approx \frac{1}{5} \text{BW} \quad \text{or} \quad \text{BW} \approx 5 F_{\text{clock}} \]

In general, the highest sine wave frequency component contained in a clock waveform is five times the clock frequency. The important assumption is that there are about 15 risetimes in one period. If the risetime is actually faster than
this assumption, the bandwidth is higher. To provide a safety margin, a package or interconnect is characterized or simulated at a bandwidth of about 10 to 20 times the clock frequency, which corresponds to roughly two to four times the bandwidth of the signal.

Definitions of Transmission Line Terms

Table C-10: Transmission Line Terms

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{\text{clock}}$</td>
<td>Clock frequency in units of frequency such as Hz</td>
</tr>
<tr>
<td>$T_{\text{clock}}$</td>
<td>Clock period in units of time such as secs or ns</td>
</tr>
<tr>
<td>$t_{\text{edge}}$</td>
<td>Rise or fall time in units of time such as sec or ns</td>
</tr>
<tr>
<td>BW</td>
<td>Bandwidth in units of frequency such as Hz or MHz</td>
</tr>
<tr>
<td>F</td>
<td>A repetitive frequency in units of Hz or MHz</td>
</tr>
<tr>
<td>f</td>
<td>A sine wave frequency in units of Hz or MHz</td>
</tr>
<tr>
<td>$\omega$</td>
<td>An angular frequency in units of radians/sec</td>
</tr>
<tr>
<td>t</td>
<td>Time or a conductor thickness, units of sec or length</td>
</tr>
<tr>
<td>V(t)</td>
<td>Instantaneous voltage in units of volts</td>
</tr>
<tr>
<td>I(t)</td>
<td>Instantaneous current in units of amps or mA</td>
</tr>
<tr>
<td>Z(t)</td>
<td>Instantaneous impedance in units of ohms</td>
</tr>
<tr>
<td>Z($\omega$)</td>
<td>Frequency domain, complex impedance in units of ohms</td>
</tr>
<tr>
<td>C</td>
<td>Capacitance in units of Farads or microFarads</td>
</tr>
<tr>
<td>R</td>
<td>Resistance in units of ohms</td>
</tr>
<tr>
<td>L</td>
<td>Inductance in units of Henrys or nanoHenries</td>
</tr>
<tr>
<td>v</td>
<td>Speed of light in the medium in units of length/time</td>
</tr>
<tr>
<td>d</td>
<td>A length in units such as inches</td>
</tr>
<tr>
<td>TD</td>
<td>Time delay in units of time such as sec or nsec</td>
</tr>
<tr>
<td>$C_L$</td>
<td>Capacitance per length in units such as pF/inch</td>
</tr>
<tr>
<td>$L_L$</td>
<td>Inductance per length in units such as nH/inch</td>
</tr>
<tr>
<td>$R_L$</td>
<td>Resistance per length in units such as ohms/inch</td>
</tr>
<tr>
<td>$\varepsilon_r$</td>
<td>Relative dielectric constant, $\varepsilon/\varepsilon_0$, dimensionless</td>
</tr>
</tbody>
</table>
Understanding the Transmission Line Theory

Ideal and Lumped Transmission Lines

Table C-10: Transmission Line Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>Reflection coefficient, dimensionless</td>
</tr>
<tr>
<td>DZ</td>
<td>A small change in characteristic impedance</td>
</tr>
<tr>
<td>GL</td>
<td>Conductance per length units of Mhos (Siemens)/length</td>
</tr>
<tr>
<td>tan(δ)</td>
<td>Dissipation factor of a material, dimensionless</td>
</tr>
<tr>
<td>d</td>
<td>Skin depth of a conductor, units of length such as meter</td>
</tr>
<tr>
<td>ρ</td>
<td>Resistivity of a conductor, units of ohm-length</td>
</tr>
<tr>
<td>μ₀</td>
<td>Permeability of free space = 4π x 10⁻⁷ Henry/meter</td>
</tr>
<tr>
<td>α</td>
<td>Attenuation per length, units of dB/len or nepers/len</td>
</tr>
<tr>
<td>W</td>
<td>A conductor width</td>
</tr>
<tr>
<td>n</td>
<td>Number of squares in a planar conductor, dimensionless</td>
</tr>
<tr>
<td>Rₛq</td>
<td>Sheet resistance of a planar conductor, units: ohms/sq</td>
</tr>
<tr>
<td>ε₀</td>
<td>Permittivity of free space=0.225 pF/inch=0.0885 pF/cm</td>
</tr>
<tr>
<td>εₑᶠᶠ</td>
<td>Effective dielectric constant due to mixed dielectrics</td>
</tr>
<tr>
<td>h</td>
<td>A dielectric thickness in units of length such as mils</td>
</tr>
<tr>
<td>R</td>
<td>Resistance in ohms</td>
</tr>
</tbody>
</table>

Relationships and Rules of Thumb

Time and Frequency Relationships

\[ F_{\text{clock}} = \frac{1}{T_{\text{clock}}} \]

\[ BW = \frac{0.35}{\tau_{\text{edge}}} \quad \text{or} \quad \tau_{\text{edge}} = \frac{0.35}{BW} \]

\[ T_{\text{period}} \approx 15\tau_{\text{edge}} \]

\[ F_{\text{clock}} \approx \frac{1}{5BW} \quad \text{or} \quad BW \approx 5F_{\text{clock}} \]
Transmission Line Effects

Transmission line analysis recommended for:

\[
\text{BW} > \frac{\nu}{10d} = \frac{1}{10 \cdot \text{TD}}
\]

\[
\tau_{\text{edge}} < \frac{5d}{\nu} = 5 \cdot \text{TD}
\]

On FR4 material,

\[
\text{BW} > \frac{600}{d} \quad \text{[BW in MHz, d in inches]}
\]

\[
F_{\text{clock}} > \frac{120}{d} \quad \text{[F_{clock} in MHz, d in inches]}
\]

\[
\tau_{\text{edge}} < \frac{d}{7.5} \quad \text{[\tau_{\text{edge}} in ns, d in inches]}
\]

Intrinsic Properties

\[
Z_0 = \frac{1}{\nu C_L} = \frac{\sqrt{L_L}}{\sqrt{C_L}} \quad \text{[C_L is in pF/inch}}
\]

\[
\nu = \frac{1}{\sqrt{L_L C_L}}
\]

\[
C_L = \frac{1}{\nu Z_0} = \frac{\sqrt{\epsilon_r}}{c Z_0} = \frac{85}{Z_0 \sqrt{\epsilon_r}} \quad \text{[pF/inch]}
\]

\[
L_L = \frac{1}{C_L \nu^2} = \frac{7.3 \epsilon_r}{C_L} \quad \text{[L_L in nH/inch, C_L in pF/inch]}
\]
Typical polymers

$\text{TD} = \frac{d}{6}$ [TD in ns, d in inches]

$C_L = \frac{170}{Z_0}$ [pF/inch]

$L_L = 0.172Z_0$ [nH/inch]

50 ohm lines

$C_L = 1.7\sqrt{\varepsilon_r}$ [pF/inch = 3.4 pF/inch (typical polymer)]

$L_L = 4.3\sqrt{\varepsilon_r}$ [nH/inch = 8.6 nH/inch (typical polymer)]

Transmission line of length $d$

$C = C_L d = \frac{\text{TD}}{Z_0}$

$L = L_L d = Z_0\text{TD}$

Reflections

Reflection coefficient from $Z_1$ to $Z_2$:

$r = \frac{Z_2 - Z_1}{Z_2 + Z_1}$

Reflection from a $\Delta Z$, of short length with time delay, TD, $\tau_{\text{edge}} > \text{TD}$:

$r = \frac{\Delta Z}{2Z_0\left(\frac{\text{TD}}{\tau_{\text{edge}}}\right)}$

Reflection from a series lumped L surrounded by a transmission line:
Reflection from a lumped C load to ground, on a transmission line:

\[ r = \frac{CZ_0}{2\tau_{\text{edge}}} \]

**Loss and Attenuation**

Skin depth of a conductor:

\[ \delta = \sqrt{\frac{\rho}{\pi \mu_0 f}} = 500 \cdot \sqrt{\frac{\rho}{f}} \]

[ \delta \text{ in meters, } \rho \text{ in ohm-meter, } f \text{ in Hz: in copper, at } f = 1e+9 \text{ and } \rho = 1.7e-8 \text{ ohm-meter, } \delta = 2.0e-6] \]

Low loss approximation for attenuation per length:

\[ \alpha = \frac{1}{2} \left( \frac{R_L}{Z_0} + G_L Z_0 \right) \quad \text{[nepers/length]} \]

\[ \alpha = 4.34 \left( \frac{R_L}{Z_0} + G_L Z_0 \right) \quad \text{[dB/length]} \]

Attenuation per length due to just dielectric loss:

\[ \alpha = 2.3 f \tan (\delta) \sqrt{\varepsilon_r} \quad \text{[dB/inch, } f \text{ in GHz]} \]

\[ \alpha \approx 0.05 \quad \text{[dB/inch for FR4 at 1 GHz]} \]

Attenuation per length due to metal, \( t < \delta \):

\[ \alpha = 43.4 \frac{\rho}{twZ_0} \quad \text{[dB/inch, with } \rho \text{ in ohm-meter, } t \text{ in microns, } w \text{ in mils]} \]
For 1 ounce copper microstrip, 5 mils wide, 50 ohm:
\[ \alpha = 0.01 \quad [\text{dB/inch}] \]

Attenuation per length due to metal, 50 ohm line, skin depth limited, \( t > \delta \):
\[ \alpha = 0.55 \sqrt{\frac{\rho f}{w}} \quad [\text{dB/inch, with } \rho \text{ in ohm-meter, } w \text{ in mils, } f \text{ in GHz}] \]

For 1 ounce copper microstrip, 5 mils wide, 50 ohm, at 1 GHz:
\[ \alpha = 0.15 \quad [\text{dB/inch}] \]

**Physical Design Quantities**

For a planar interconnect:

Sheet resistance:
\[ R_{sq} = \frac{\rho}{t} \quad [\text{ohm/sq}] \]

Number of squares:
\[ n = \frac{d}{w} \]

Resistance:
\[ R_L = n \cdot R_{sq} \quad [\text{ohm/inch}, R_{sq} \text{ in ohm/sq}] \]

Resistance per length:
\[ R_L = 10 \frac{\rho}{w} \quad [\text{ohm/inch, } \rho \text{ in ohm-meter, } t \text{ in microns, } w \text{ in mils}] \]

Parallel plate, no fringe fields:
\[ C_L = 0.225 \varepsilon_r \left( \frac{w}{h} \right) \quad [\text{pF/inch}] \]
Microstrip capacitance per length good to ~20%:

\[ C_L = 0.45 \varepsilon_r \left( \frac{w}{h} \right) \quad \text{[pF/inch]} \]

Microstrip capacitance per length, good to ~5%:

\[ C_L = \frac{1.41 \varepsilon_{\text{eff}}}{\ln \left( \frac{8h}{w} + \frac{w}{4h} \right)} \]

\[ \varepsilon_{\text{eff}} = \left( \frac{\varepsilon_r + 1}{2} \right) + \left( \frac{\varepsilon_r - 1}{2} \right) \left( 1 + \frac{10h}{w} \right)^{-\frac{1}{2}} \]

Stripline capacitance per length good to ~20%:

\[ C_L = 0.675 \varepsilon_r \left( \frac{w}{h} \right) \quad \text{[pF/inch]} \]

Stripline capacitance per length, good to ~5%:

\[ C_L = \frac{0.9 \varepsilon_r}{\ln \left( 1 + \frac{2h}{w} \right)} \quad \text{[pF/inch]} \]

Inductance per length of one wire in a pair of two parallel wires:

(r = radius, s = center to center spacing, s \gg r)

\[ L_L = 5 \ln \left( \frac{s}{r} \right) \quad \text{[nH/inch]} \]

Inductance per length for a circular loop:

\[ L_L = 26 \quad \text{[nH/inch of perimeter]} \]
Inductance per length of controlled impedance line, when the return line is a reference plane:

\[ L_L = 0.086 Z_0 \sqrt{\epsilon_r} = \frac{7.3 \epsilon_r}{C_L} \text{ [nH/inch with } C_L \text{ in pF/inch]} \]

Conductance per length:

\[ G_L = \omega \tan(\delta) C_L \]

**Attenuation in Transmission Lines**

The T Element used in Star-Hspice and common to most Berkeley-compatible SPICE tools uses the lossless model for a transmission line. This model adequately simulates the dominant effects related to transmission behavior: the initial driver loading due to a resistive impedance, reflections from characteristic impedance changes, reflections introduced by stubs and branches, a time delay for the propagation of the signal from one end to the other and the reflections from a variety of linear and nonlinear termination schemes.

In systems with risetimes are on the order of 1 ns, transmission line effects dominate interconnect performance.

In some high-speed applications, the series resistance seriously effects signal strength and should be taken into account for a realistic simulation.

The first-order contribution from series resistance is an attenuation of the waveform. This attenuation decreases the amplitude and the bandwidth of the propagating signal. As a positive result, reflection noise decreases, so that a lossless simulation is a worst case. As a negative result, the effective propagation delay is longer because the risetimes are longer. A lossless simulation shows a shorter interconnect related delay than a lossy simulation.

The second-order effects introduced by series resistance are a frequency dependence to the characteristic impedance and a frequency dependence to the speed of propagation, often called dispersion. Both the first order and second order effects of series resistance generic to lossy transmission lines are simulated using the Star-Hspice U model.
The Physical Basis of Loss

The origin of loss is the series resistance of the conductors and the dielectric loss of the insulation. Conductor resistance is considered in two parts, the DC resistance and the resistance when skin depth plays a role. The dielectric loss of the insulation, at low frequency, is described by the material conductivity ($\sigma$) (SIG in Star-Hspice), and at high frequency, by the dissipation factor, $\tan(\delta)$. These material effects contribute a shunt conductance to ground ($G$).

In a planar interconnect such as a microstrip or stripline, the resistance per length of the conductor $R_L$ is

$$ R_L = \frac{\rho}{w t} $$

$\rho$ = bulk resistivity of the conductor  
$w$ = the line width of the conductor  
$t$ = thickness of the conductor

**Example 1: FR4, 5 mil wide line, half ounce copper:**

$R_L = 0.24$ ohm/inch

**Example 2: Cofired ceramic, 4 mil wide, 0.75 mils thick Tungsten:**

$R_L = 2.7$ ohm/inch

**Example 3: Thin film copper, 1 mil wide, 5 microns thick:**

$R_L = 3.6$ ohm/inch

**Skin Depth**

At high frequency, the component of the electric field along the conductor, which drives the current flow, does not penetrate fully into the depth of the conductor. Rather, its amplitude falls off exponentially. This exponential decay length is called the skin depth ($\delta$). When the signal is a sine wave, the skin depth depends on the conductor’s resistivity ($\rho$), and the sine wave frequency of the current ($f$):
A real signal has most of its energy at a frequency of $1/\tau_{\text{edge}}$, where $\tau_{\text{period}}$ is the average period. Because most of the loss occurs at this frequency, as a first approximation $\delta$ should be used to compute the skin depth. In practice, as mentioned previously, approximating $\tau_{\text{period}}$ by $15(\tau_{\text{edge}})$ works well. With a 1 ns rise time, the skin depth of copper is 8 microns, assuming $15(\tau_{\text{edge}})$ is used for $1/\tau_{\text{skin}}$. For half ounce copper, where the physical thickness is 15 microns, the skin depth thickness should be used in place of the conductor thickness to estimate the high frequency effects.

For thin film substrates with a physical thickness of the order of 5 microns or less, the effects of skin depth can, to the first order, be ignored. In cofired ceramic substrates, the skin depth for 1 ns edges with tungsten paste conductors is 27.6 microns. This is also comparable to the 19 micron physical thickness, and to the first order, the effects of skin depth can be ignored. At shorter rise times than 1 ns, skin depth plays an increasingly significant role.

**Dielectric Loss**

Two separate physical mechanisms contribute to conductivity in dielectrics, which results in loss: DC conduction and high frequency dipole relaxation. As illustrated in the following section, the effects from dielectric loss are in general negligible. For most practical applications, the dielectric loss from the DC conductivity and the high frequency dissipation factor can be ignored.

To be cautious, estimate the magnitude of the conductance of the dielectric and verify that, for a particular situation, it is not a significant issue. Exercise care in using these material effects in general application.

The bulk conductivity of insulators used in interconnects ($\sigma$), typically specified as between $10^{-12}$ and $10^{-16}$ siemens/cm, is often an upper limit, rather than a true value. It is also very temperature and humidity sensitive. The shunt conductance per length ($G_L$) depends on the geometrical features of the conductors in the same way as the capacitance per length ($C_L$). It can be written as:

$$\delta = \sqrt{\frac{\rho}{\pi \mu_0 f}} = 500 \cdot \sqrt{\frac{\rho}{f}} \quad [\delta \text{ in meters, } \rho \text{ in ohm-meter, } f \text{ in Hz}]$$
At high frequencies, typically over 1 MHz, dipole relaxations begin to dominate the conduction current and cause it to be frequency dependent. This effect is described by the dissipation factor of a material, which ranges from 0.03 for epoxies down to 0.003 for polyimides and less than 0.0005 for ceramics and Teflon. The effective conductivity of a dielectric material at high frequency is:

\[ \sigma = 2\pi f \varepsilon_0 \varepsilon_r \tan(\delta) \]

The shunt conductance per length of an interconnect, when dipole relaxation dominates, is:

\[ G_L = 2\pi f \varepsilon_0 \varepsilon_r \tan(\delta) C_L \]

As a worst case, the frequency corresponding to the bandwidth of the signal can be used to estimate the high frequency conductivity of the material.

**The Lossy Transmission Line Model**

In the lossless transmission line model, only the distributed capacitance (C) and inductance (L) of the interconnect is considered:

**Figure C-47: Lossless Transmission Line Model**

In the lossy transmission line model, the series resistance and dielectric conductance are introduced into the equivalent circuit model:
These four circuit elements, normalized per unit length, can be used to describe all the high frequency properties of a transmission line. When the equivalent circuit equation is solved in the frequency domain, the characteristic impedance is modified to:

\[
Z_0 = \sqrt{\frac{R_L + j\omega L}{G_L + j\omega C}}
\]

and the propagation phase term, \( \gamma \), is:

\[
\gamma = \alpha + j\beta = \sqrt{(R_L + j\omega L)(G_L + j\omega C)}
\]

In the propagation phase term, \( \beta \) is related to the phase velocity by:

\[
v = \frac{\omega}{\beta}
\]

To first order, when \( R_L \ll \omega L \) and \( G_L \ll \omega C \), the characteristic impedance and phase velocity \( (v) \) are unchanged from their lossless values. However, a new term, the attenuation per length \( (\alpha) \) is introduced.

The attenuation per length is approximately:

\[
\alpha = \frac{1}{2} \left( \frac{R_L}{Z_0} + G_L Z_0 \right) \text{[nepers/length]}
\]

\[
\alpha = 4.34 \left( \frac{R_L}{Z_0} + G_L Z_0 \right) \text{[dB/length]}
\]
The total attenuation ($\alpha d$) determines the fraction of the signal amplitude that remains after propagating the distance ($d$). When $\alpha$ has the units of dB/length, the fraction of signal remaining is:

$$\left(\frac{-\alpha d}{20}\right)$$

It is useful to keep in mind that a 2 dB attenuation in a signal corresponds to a final amplitude of 80% of the original and 6 dB attenuation corresponds to a final amplitude of 50% of the original. Attenuation on the order of 6 dB significantly changes the signal integrity.

**Attenuation Due to Conductor Resistance**

In the typical case of a 50 ohm transmission line, the attenuation per length due to just the series resistance is

$$\alpha = 0.09 R_L \quad [\text{dB/length}]$$

When the resistance per length is of the order of 0.2 ohm/inch or less, as is the case in typical printed circuit boards, the attenuation per length is about 0.02 dB/inch. Typical interconnect lengths of 10 inches yields only 0.2 dB, which would leave about 98% of the signal remaining. Using the lossless T Element to approximate most applications provides a good approximation.

However, in fine line substrates, as the examples in the previous section illustrated, the resistance per length can be on the order of 2 ohms/inch. In such a case, the attenuation is on the order of 0.2 dB/inch. So a 10 inch interconnect line then has an attenuation on the order of 2 dB, which would leave only about 80% of the signal. This is large enough that its effects should be included in a simulation.

**Attenuation Due to the Dielectric**

When the dielectric completely surrounds the conductors, the attenuation due to just the conductance per length of the dielectric is:

$$\alpha_{\text{dielectric}} = 2.3 f \tan(\delta) \sqrt{\varepsilon_{\text{eff}}} \quad [\text{dB/inch}, f \text{ in GHz}]$$
The worst case and highest attenuation per length is exhibited by FR4 boards, with \( \tan(\delta) \) of the order of 0.02 and a dielectric constant of 5. The attenuation at 1 GHz is about 0.1 dB/inch. For an interconnect 10 inches long, this is 1 dB of attenuation, which would leave about 90% of the signal remaining, comparable to the attenuation offered by a conductor with 1 ohm/inch resistance.

When the resistance per length is larger than 1 ohm/inch— for example in cofired ceramic and thin film substrates, and the dissipation factor is less than 0.005, the attenuation from the conductor losses can be on the order of 10 times greater than dielectric loss. In these applications, the dielectric losses can be ignored.

**Integrating Attenuation Effects in Star-Hspice**

All of the first-order effects of attenuation are automatically simulated with the U Element in Star-Hspice.

With ELEV=1, the inputs can be the cross sectional geometry and the material properties of the conductor, bulk resistivity (RHO), the relative dielectric constant of the insulation (KD), and the conductivity of the dielectric (SIG). From these features, the equivalent capacitance per length, inductance per length, series resistance per length, and conductance per length are calculated by Star-Hspice.

With ELEV=2, the equivalent capacitance per length, inductance per length, series resistance per length, and conductance per length are input directly using estimates, measurements or third-party modeling tools.

Star-Hspice automatically generates a model for the specified net composed of a series of lumped elements that resembles the model for a lossy transmission line. The parameter WLUMPS controls the number of lumped elements included per wavelength, based on the estimated rise time of signals in the simulation.

The attenuation effects previously described are a natural consequence of this model. The U Element allows realistic simulations of lossy transmission lines in both the AC and the transient domain.
References

Appendix D

Finding Device Libraries

For libraries with multiple models of a given element, you need to be able to automatically find the proper model for each transistor size. Use the Star-Hspice automatic model selector to accomplish this.

This chapter describes how to use the model selector, then provides listings of device libraries that you can use with Star-Hspice.

The following topics are covered in this chapter:

- Selecting Models Automatically
- Examining the Library Listings
Selecting Models Automatically

The model selector uses the following criteria:

\[
L_{MIN} + XL_{REF} \leq L + XL < L_{MAX} + XL_{REF}
\]

\[
W_{MIN} + XW_{REF} \leq W + XW < W_{MAX} + XW_{REF}
\]

(If \(XL_{REF}\) is not specified, \(XL_{REF}\) is set to \(XL\). If \(XW_{REF}\) is not specified, \(XW_{REF}\) is set to \(XW\).)

The model selector syntax is based on a common model root name, with a unique extension for each model.

**Note:** The preceding does not apply to JFETs.

The following is an example of HSPICE syntax for models:

M1 drain gate source bulk NJ W=2u L=1u

```
.MODEL NJ4 NJF WMIN=1.5u WMAX=3u LMIN=.8u LMAX=2u
.MODEL NJ5 NJF WMIN=1.5u WMAX=3u LMIN=2u LMAX=6u
```

Figure D-1 illustrates the model selection method.
For this example, there are several pch.x models, with varying drawn channel lengths and widths, in the model library. (The model root name is pch and the extensions are 1, 2, ..., 6). The NJ4 instance of the NJ Element (W=2 µ, L=1 µ) requires a model for which 1.5 µ ≤ channel width ≤ 3 µ, and 0.8 µ ≤ channel length ≤ 2 µ. The automatic model selector chooses the pch.4 model since that model satisfies these requirements. Similarly, the NJ5 transistor requires a model with 1.5 µ ≤ channel width ≤ 3 µ, and 2 µ ≤ channel length ≤ 6 µ. The pch.5 model satisfies these requirements. If a device size is out of range for all the models that exist, an error message is issued.

If a model within a subcircuit cannot be found, the automatic model selector searches the top level. If the automatic model selector fails to find a model, HSPICE terminates.

The following combination of conditions causes the automatic model selector to fail and terminates HSPICE:

For this example, there are several pch.x models, with varying drawn channel lengths and widths, in the model library. (The model root name is pch and the extensions are 1, 2, ..., 6). The NJ4 instance of the NJ Element (W=2 µ, L=1 µ) requires a model for which 1.5 µ ≤ channel width ≤ 3 µ, and 0.8 µ ≤ channel length ≤ 2 µ. The automatic model selector chooses the pch.4 model since that model satisfies these requirements. Similarly, the NJ5 transistor requires a model with 1.5 µ ≤ channel width ≤ 3 µ, and 2 µ ≤ channel length ≤ 6 µ. The pch.5 model satisfies these requirements. If a device size is out of range for all the models that exist, an error message is issued.

If a model within a subcircuit cannot be found, the automatic model selector searches the top level. If the automatic model selector fails to find a model, HSPICE terminates.

The following combination of conditions causes the automatic model selector to fail and terminates HSPICE:
1. In the element statement, a model name is used which contains a period (.)

2. The model library was not designed for use with the HSPICE automatic model selector.

3. Either a multisweep specification or a .TEMP temperature analysis statement is included in the HSPICE input.

The following example illustrates how a period in a model name can cause automatic model selection problems.

Case 1:
M1 d g s b N.CHN W=10u L=5u* Element statement
.MODEL N.CHN LMIN=1u LMAX=4u WMIN=2u WMAX=100u*
.MODEL statement

Case 2:
.TEMP 25
.M1 d g s b N.CHN W=10u L=5u* Element statement
.MODEL N.CHN LMIN=1u LMAX=4u WMIN=2u WMAX=100u*
.MODEL statement

In Case 1, since there is no multisweep or temperature analysis specified, the HSPICE model selector feature is not invoked, so the N.CHN model is used with no problems.

In Case 2, however, the presence of the .TEMP statement invokes the model selector feature. The model selector tries to find a model named N.\textit{nnn} that fits within the length and width ranges given in the element statement. Because the length given in the element statement (5 \(\mu\)m) is not within the 1 to 4 \(\mu\)m range specified in the .MODEL statement, the model selector cannot find a model that matches the element statement, and HSPICE issues a “device ‘N’ not found” error message.
Examining the Library Listings

The names of models provided with Star-Hspice are listed in the following sections. Each type of model is stored in a directory that has a name indicating the type of models it contains, such as dio for diodes and bjt for bipolar junction transistors. The path to the directory is shown for each model type. This path can be specified in a .OPTION SEARCH statement, such as:

`.OPTION SEARCH \"$installdir/96/parts/dio\"`

where $installdir is the environment variable set to the path to the HSPICE software installation directory and 96 is the HSPICE release number. All model directories are under the parts directory.

Analog Device Models

Search path: $installdir/parts/ad

Model names:

AD581  AD581J  AD581K  AD581L  AD581S
AD581T AD581U  AD584  AD584J  AD584K
AD584L AD584S  AD584T  AD587  AD587J
AD587K AD587L  AD587S  AD587T  AD587U
AD600  AD602J  AD602  AD620
AD620A AD620B  AD620S  AD624  AD624A
AD624B AD624C  AD624S  AD630  AD630A
AD630B AD630J  AD630K  AD630S  AD633
AD633J AD645  AD645A  AD645B  AD645J
AD645K AD645S  AD704  AD704A  AD704B
AD704J AD704K  AD704T  AD705  AD705A
AD705B AD705J  AD705K  AD705T  AD706
AD706A AD706B  AD706J  AD706K  AD706T
AD711  AD711A  AD711B  AD711C  AD711J
AD711K AD711S  AD711T  AD712  AD712A
AD712B AD712C  AD712J  AD712K  AD712S
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AD712T</td>
<td>AD713</td>
<td>AD713A</td>
<td>AD713B</td>
<td>AD713J</td>
</tr>
<tr>
<td>AD713K</td>
<td>AD713S</td>
<td>AD713T</td>
<td>AD734</td>
<td>AD734A</td>
</tr>
<tr>
<td>AD734B</td>
<td>AD734S</td>
<td>AD743</td>
<td>AD743A</td>
<td>AD743B</td>
</tr>
<tr>
<td>AD743J</td>
<td>AD743K</td>
<td>AD743S</td>
<td>AD744</td>
<td>AD744A</td>
</tr>
<tr>
<td>AD744B</td>
<td>AD744C</td>
<td>AD744J</td>
<td>AD744K</td>
<td>AD744S</td>
</tr>
<tr>
<td>AD744T</td>
<td>AD745</td>
<td>AD745A</td>
<td>AD745B</td>
<td>AD745J</td>
</tr>
<tr>
<td>AD745K</td>
<td>AD745S</td>
<td>AD746</td>
<td>AD746A</td>
<td>AD746B</td>
</tr>
<tr>
<td>AD746J</td>
<td>AD746S</td>
<td>AD780</td>
<td>AD780A</td>
<td>AD780B</td>
</tr>
<tr>
<td>AD780S</td>
<td>AD797</td>
<td>AD797A</td>
<td>AD797B</td>
<td>AD797S</td>
</tr>
<tr>
<td>AD810</td>
<td>AD810A</td>
<td>AD810S</td>
<td>AD811</td>
<td>AD812</td>
</tr>
<tr>
<td>AD812A</td>
<td>AD813</td>
<td>AD813A</td>
<td>AD817</td>
<td>AD817A</td>
</tr>
<tr>
<td>AD818</td>
<td>AD818A</td>
<td>AD820</td>
<td>AD826</td>
<td>AD826A</td>
</tr>
<tr>
<td>AD828</td>
<td>AD828A</td>
<td>AD829</td>
<td>AD829A</td>
<td>AD829J</td>
</tr>
<tr>
<td>AD829S</td>
<td>AD830</td>
<td>AD830A</td>
<td>AD830J</td>
<td>AD830S</td>
</tr>
<tr>
<td>AD840</td>
<td>AD840J</td>
<td>AD840K</td>
<td>AD840S</td>
<td>AD843</td>
</tr>
<tr>
<td>AD843A</td>
<td>AD843B</td>
<td>AD843J</td>
<td>AD843K</td>
<td>AD843S</td>
</tr>
<tr>
<td>AD844</td>
<td>AD844A</td>
<td>AD844B</td>
<td>AD844S</td>
<td>AD845</td>
</tr>
<tr>
<td>AD845A</td>
<td>AD845B</td>
<td>AD845J</td>
<td>AD845K</td>
<td>AD845S</td>
</tr>
<tr>
<td>AD846</td>
<td>AD846A</td>
<td>AD846B</td>
<td>AD846S</td>
<td>AD847</td>
</tr>
<tr>
<td>AD847A</td>
<td>AD847J</td>
<td>AD847S</td>
<td>AD848</td>
<td>AD848A</td>
</tr>
<tr>
<td>AD848J</td>
<td>AD848S</td>
<td>AD9617</td>
<td>AD9618</td>
<td>AD9621</td>
</tr>
<tr>
<td>AD9622</td>
<td>AD9623</td>
<td>AD9624</td>
<td>AD9630</td>
<td>ADG411</td>
</tr>
<tr>
<td>ADG411B</td>
<td>ADG411T</td>
<td>ADG412</td>
<td>ADG412B</td>
<td>ADG412T</td>
</tr>
<tr>
<td>ADG413</td>
<td>ADG413B</td>
<td>ADG413T</td>
<td>AMP01</td>
<td>AMP02</td>
</tr>
<tr>
<td>BUF04</td>
<td>MAT02</td>
<td>MAT03</td>
<td>MAT04</td>
<td>MLT04</td>
</tr>
<tr>
<td>MLT04G</td>
<td>OP160</td>
<td>OP160A</td>
<td>OP160F</td>
<td>OP160G</td>
</tr>
<tr>
<td>OP176</td>
<td>OP176G</td>
<td>OP177</td>
<td>OP177A</td>
<td>OP177B</td>
</tr>
<tr>
<td>OP177E</td>
<td>OP177F</td>
<td>OP177G</td>
<td>OP20</td>
<td>OP200</td>
</tr>
<tr>
<td>OP200A</td>
<td>OP200E</td>
<td>OP200F</td>
<td>OP200G</td>
<td>OP20B</td>
</tr>
<tr>
<td>OP20C</td>
<td>OP20F</td>
<td>OP20G</td>
<td>OP20H</td>
<td>OP21</td>
</tr>
<tr>
<td>OP213</td>
<td>OP215</td>
<td>OP215A</td>
<td>OP215B</td>
<td>OP215C</td>
</tr>
<tr>
<td>OP21F</td>
<td>OP21G</td>
<td>OP21H</td>
<td>OP220</td>
<td>OP220A</td>
</tr>
</tbody>
</table>
### Finding Device Libraries

#### Examining the Library Listings

<table>
<thead>
<tr>
<th>Device</th>
<th>Device</th>
<th>Device</th>
<th>Device</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP220C</td>
<td>OP220E</td>
<td>OP220F</td>
<td>OP220G</td>
<td>OP221</td>
</tr>
<tr>
<td>OP221A</td>
<td>OP221B</td>
<td>OP221C</td>
<td>OP221E</td>
<td>OP221G</td>
</tr>
<tr>
<td>OP249</td>
<td>OP249A</td>
<td>OP249E</td>
<td>OP249F</td>
<td>OP249G</td>
</tr>
<tr>
<td>OP260</td>
<td>OP27</td>
<td>OP275</td>
<td>OP275G</td>
<td>OP27A</td>
</tr>
<tr>
<td>OP27B</td>
<td>OP27C</td>
<td>OP27E</td>
<td>OP27F</td>
<td>OP27G</td>
</tr>
<tr>
<td>OP282</td>
<td>OP282G</td>
<td>OP283</td>
<td>OP285</td>
<td>OP285G</td>
</tr>
<tr>
<td>OP290</td>
<td>OP290A</td>
<td>OP290E</td>
<td>OP290F</td>
<td>OP290G</td>
</tr>
<tr>
<td>OP292</td>
<td>OP295</td>
<td>OP297</td>
<td>OP297A</td>
<td>OP297E</td>
</tr>
<tr>
<td>OP297F</td>
<td>OP297G</td>
<td>OP37</td>
<td>OP37A</td>
<td>OP37B</td>
</tr>
<tr>
<td>OP37C</td>
<td>OP37E</td>
<td>OP37F</td>
<td>OP37G</td>
<td>OP400</td>
</tr>
<tr>
<td>OP400A</td>
<td>OP400E</td>
<td>OP400F</td>
<td>OP400G</td>
<td>OP400H</td>
</tr>
<tr>
<td>OP41</td>
<td>OP41A</td>
<td>OP41B</td>
<td>OP41E</td>
<td>OP41F</td>
</tr>
<tr>
<td>OP41G</td>
<td>OP42</td>
<td>OP420</td>
<td>OP420B</td>
<td>OP420C</td>
</tr>
<tr>
<td>OP420F</td>
<td>OP420G</td>
<td>OP420H</td>
<td>OP421</td>
<td>OP421B</td>
</tr>
<tr>
<td>OP421C</td>
<td>OP421F</td>
<td>OP421G</td>
<td>OP421H</td>
<td>OP42A</td>
</tr>
<tr>
<td>OP42E</td>
<td>OP42F</td>
<td>OP42G</td>
<td>OP43</td>
<td>OP43A</td>
</tr>
<tr>
<td>OP43B</td>
<td>OP43E</td>
<td>OP43F</td>
<td>OP43G</td>
<td>OP44</td>
</tr>
<tr>
<td>OP467</td>
<td>OP467G</td>
<td>OP470</td>
<td>OP482</td>
<td>OP482G</td>
</tr>
<tr>
<td>OP490</td>
<td>OP490A</td>
<td>OP490E</td>
<td>OP490F</td>
<td>OP490G</td>
</tr>
<tr>
<td>OP492</td>
<td>OP497</td>
<td>OP497A</td>
<td>OP497B</td>
<td>OP497C</td>
</tr>
<tr>
<td>OP497F</td>
<td>OP497G</td>
<td>OP61</td>
<td>OP64</td>
<td>OP77</td>
</tr>
<tr>
<td>OP77A</td>
<td>OP77B</td>
<td>OP77E</td>
<td>OP77F</td>
<td>OP77G</td>
</tr>
<tr>
<td>OP80</td>
<td>OP80B</td>
<td>OP80E</td>
<td>OP80F</td>
<td>OP80G</td>
</tr>
<tr>
<td>OP90</td>
<td>OP90A</td>
<td>OP90E</td>
<td>OP90F</td>
<td>OP90G</td>
</tr>
<tr>
<td>OP97</td>
<td>OP97A</td>
<td>OP97E</td>
<td>OP97F</td>
<td>PM1012</td>
</tr>
<tr>
<td>REF01</td>
<td>REF01A</td>
<td>REF01C</td>
<td>REF01E</td>
<td>REF01H</td>
</tr>
<tr>
<td>REF02</td>
<td>REF02A</td>
<td>REF02C</td>
<td>REF02D</td>
<td>REF02E</td>
</tr>
<tr>
<td>REF02H</td>
<td>REF05</td>
<td>REF05A</td>
<td>REF05B</td>
<td>REF10</td>
</tr>
<tr>
<td>REF10A</td>
<td>REF10B</td>
<td>SSM2017</td>
<td>SSM2017P</td>
<td>SSM2131</td>
</tr>
<tr>
<td>SSM2210</td>
<td>SSM2220</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Behavioral Device Models

Required element syntax: Xyyyyy in- in+ out vcc vee modelname

Search path: $installdir/parts/behave

Optional parameters: vos=value, ibos=value, av=value

Model names:

<table>
<thead>
<tr>
<th>AD4BIT</th>
<th>AD8BIT</th>
<th>ALF155</th>
<th>ALF156</th>
<th>ALF157</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALF255</td>
<td>ALF347</td>
<td>ALF351</td>
<td>ALF353</td>
<td>ALF355</td>
</tr>
<tr>
<td>ALF356</td>
<td>ALF357</td>
<td>ALF3741</td>
<td>ALM101A</td>
<td>ALM107</td>
</tr>
<tr>
<td>ALM108</td>
<td>ALM108A</td>
<td>ALM111</td>
<td>ALM118</td>
<td>ALM124</td>
</tr>
<tr>
<td>ALM124A</td>
<td>ALM139A</td>
<td>ALM1458</td>
<td>ALM1558</td>
<td>ALM158</td>
</tr>
<tr>
<td>ALM158A</td>
<td>ALM201A</td>
<td>ALM207</td>
<td>ALM208</td>
<td>ALM208A</td>
</tr>
<tr>
<td>ALM224</td>
<td>ALM258</td>
<td>ALM258A</td>
<td>ALM2901</td>
<td>ALM2902</td>
</tr>
<tr>
<td>ALM2904</td>
<td>ALM301A</td>
<td>ALM307</td>
<td>ALM308</td>
<td>ALM308A</td>
</tr>
<tr>
<td>ALM318</td>
<td>ALM324</td>
<td>ALM3302</td>
<td>ALM339</td>
<td>ALM358</td>
</tr>
<tr>
<td>ALM358A</td>
<td>ALM725</td>
<td>ALM741</td>
<td>ALM747</td>
<td>ALM747C</td>
</tr>
<tr>
<td>AMC1458</td>
<td>AMC1536</td>
<td>AMC1741</td>
<td>AMC1747</td>
<td>ANE5534P</td>
</tr>
<tr>
<td>ANJM4558</td>
<td>ANJM4559</td>
<td>ANJM4560</td>
<td>AOP04</td>
<td>AOP07</td>
</tr>
<tr>
<td>AOP14</td>
<td>AOP15B</td>
<td>AOP16B</td>
<td>AT094CN5</td>
<td>ATL071C</td>
</tr>
<tr>
<td>ATL072C</td>
<td>ATL074C</td>
<td>ATL081C</td>
<td>ATL082C</td>
<td>ATL084C</td>
</tr>
<tr>
<td>ATL092CP</td>
<td>ATL094CN</td>
<td>AUPC1251</td>
<td>AUPC358</td>
<td>GA201</td>
</tr>
<tr>
<td>RCFILT</td>
<td>TLINE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bipolar Transistor Models

Required element syntax: Xyyyy coll base emit modelname

Search path: $installdir/parts/bjt

Optional parameters: betaf=value, tauf=value

Model names:

- T2N1132A
- T2N2102
- T2N2219A
- T2N2222
- T2N2222A
- T2N2369
- T2N2369A
- T2N2501
- T2N2605
- T2N2642
- T2N2857
- T2N2894
- T2N2904
- T2N2904A
- T2N2905
- T2N2905A
- T2N3013
- T2N3227
- T2N3250
- T2N3250A
- T2N3251
- T2N3251A
- T2N3467
- T2N3501
- T2N3546
- T2N3637
- T2N3742
- T2N3743
- T2N3866
- T2N3904
- T2N3906
- T2N3946
- T2N3947
- T2N3962
- T2N4261
- T2N4449
- T2N5058
- T2N5059
- T2N5179
- T2N6341
- T2N6438
- T2N706
- T2N708
- T2N869
- T2N869A
- T2N918
- T2N930
- T2SA1015
- T2SA950
- T2SA965
- T2SA970
- T2SC1815
- T2SC1923
- T2SC2120
- T2SC2235
- T2SC2669
- TMP6595
- TNE741
- TNE901

Burr-Brown Devices

Search path: $installdir/parts/burr_brn

Model names:

- INA101
- INA101E
- INA102
- INA102E
- INA103
- INA103E
- INA105
- INA105E
- INA106
- INA106E
- INA110
- INA110E
- INA117
- INA117E
- INA120
- INA120E
- ISO120X
- ISO121X
- OPA101
- OPA1013
- OPA102
- OPA102E
- OPA111
- OPA111E
- OPA121
- OPA121E
- OPA128
- OPA128E
- OPA177
- OPA177E
- OPA2107
- OPA2107E
- OPA2111
Comlinear Device Models

Search path: $installdir/parts/comline

Model names:

<table>
<thead>
<tr>
<th>CLC109</th>
<th>CLC111</th>
<th>CLC400</th>
<th>CLC401</th>
<th>CLC402</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLC404</td>
<td>CLC406</td>
<td>CLC409</td>
<td>CLC410</td>
<td>CLC414</td>
</tr>
<tr>
<td>CLC415</td>
<td>CLC420</td>
<td>CLC425</td>
<td>CLC426</td>
<td>CLC428</td>
</tr>
<tr>
<td>CLC430</td>
<td>CLC431</td>
<td>CLC432</td>
<td>CLC501</td>
<td>CLC502</td>
</tr>
<tr>
<td>CLC505</td>
<td>CLC520</td>
<td>CLC522</td>
<td>CLC532</td>
<td></td>
</tr>
</tbody>
</table>

Diode Models

Required element syntax: Xyyyyy anode cathode modelname

Search path: $installdir/parts/dio

Optional parameters: isat=value, tt=value

Model names:

<table>
<thead>
<tr>
<th>D12BG11</th>
<th>D12BH11</th>
<th>D12DG11</th>
<th>D12DH11</th>
<th>D12FG11</th>
</tr>
</thead>
<tbody>
<tr>
<td>D12FH11</td>
<td>D12GG11</td>
<td>D12GH11</td>
<td>D12JG11</td>
<td>D12JH11</td>
</tr>
<tr>
<td>D1N3016</td>
<td>D1N3017</td>
<td>D1N3018</td>
<td>D1N3019</td>
<td>D1N3020</td>
</tr>
<tr>
<td>D1N3021</td>
<td>D1N3022</td>
<td>D1N3023</td>
<td>D1N3024</td>
<td>D1N3025</td>
</tr>
<tr>
<td>D1N3026</td>
<td>D1N3027</td>
<td>D1N3028</td>
<td>D1N3029</td>
<td>D1N3030</td>
</tr>
</tbody>
</table>
### Finding Device Libraries

### Examining the Library Listings

<table>
<thead>
<tr>
<th>Library Listings</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1N3031</td>
</tr>
<tr>
<td>D1N3036</td>
</tr>
<tr>
<td>D1N3041</td>
</tr>
<tr>
<td>D1N3046</td>
</tr>
<tr>
<td>D1N3051</td>
</tr>
<tr>
<td>D1N3825</td>
</tr>
<tr>
<td>D1N3830</td>
</tr>
<tr>
<td>D1N4005</td>
</tr>
<tr>
<td>D1N4150</td>
</tr>
<tr>
<td>D1N4447</td>
</tr>
<tr>
<td>D1N4730</td>
</tr>
<tr>
<td>D1N4735</td>
</tr>
<tr>
<td>D1N4740</td>
</tr>
<tr>
<td>D1N4745</td>
</tr>
<tr>
<td>D1N4750</td>
</tr>
<tr>
<td>D1N4755</td>
</tr>
<tr>
<td>D1N4760</td>
</tr>
<tr>
<td>D1N5221</td>
</tr>
<tr>
<td>D1N5226</td>
</tr>
<tr>
<td>D1N5231</td>
</tr>
<tr>
<td>D1N5236</td>
</tr>
<tr>
<td>D1N5241</td>
</tr>
<tr>
<td>D1N5246</td>
</tr>
<tr>
<td>D1N5251</td>
</tr>
<tr>
<td>D1N5256</td>
</tr>
<tr>
<td>D1N5261</td>
</tr>
<tr>
<td>D1N5266</td>
</tr>
<tr>
<td>D1N5271</td>
</tr>
<tr>
<td>D1N5336</td>
</tr>
<tr>
<td>D1N5341</td>
</tr>
<tr>
<td>D1N5346</td>
</tr>
<tr>
<td>D1N5351</td>
</tr>
<tr>
<td>D1N5356</td>
</tr>
</tbody>
</table>
Examining the Library Listings

Finding Device Libraries

D1N5361   D1N5362   D1N5363   D1N5364   D1N5365
D1N5366   D1N5367   D1N5368   D1N5369   D1N5370
D1N5371   D1N5372   D1N5373   D1N5374   D1N5375
D1N5376   D1N5377   D1N5378   D1N5379   D1N5380
D1N5381   D1N5382   D1N5383   D1N5384   D1N5385
D1N5386   D1N5387   D1N5388   D1N5817   D1N5818
D1N5819   D1N5913   D1N5914   D1N5915   D1N5916
D1N5917   D1N5918   D1N5919   D1N5920   D1N5921
D1N5922   D1N5923   D1N5924   D1N5925   D1N5926
D1N5927   D1N5928   D1N5929   D1N5930   D1N5931
D1N5932   D1N5933   D1N5934   D1N5935   D1N5936
D1N5937   D1N5938   D1N5939   D1N5940   D1N5941
D1N5942   D1N5943   D1N5944   D1N5945   D1N5946
D1N5947   D1N5948   D1N5949   D1N5950   D1N5951
D1N5952   D1N5953   D1N5954   D1N5955   D1N5956
D1N746    D1N747    D1N748    D1N749    D1N750
D1N751    D1N752    D1N753    D1N754    D1N755
D1N756    D1N757    D1N758    D1N759    D1N914
D1N957    D1N958    D1N959    D1N960    D1N961
D1N962    D1N963    D1N964    D1N965    D1N966
D1N967    D1N968    D1N969    D1N970    D1N971
D1N972    D1N973    D1N974    D1N975    D1N976
D1N977    D1N978    D1N979    D1N980    D1N981
D1N982    D1N983    D1N984    D1N985    D1N986
D1S1585   D1S1586   D1S1587   D1S1588   D1SV147
D1SV149   DMBR115P   DMBR120P   DMBR130P   DMBR140P
DSK4A3

**FET Models**

Required element syntax: Xyyyy drain gate source modelname

Search path:  $installdir/parts/fet

Optional parameters: vt = value, betaf = value
### Model names:

<table>
<thead>
<tr>
<th>J108</th>
<th>J109</th>
<th>J110</th>
<th>J111</th>
<th>J112</th>
</tr>
</thead>
<tbody>
<tr>
<td>J113</td>
<td>J2N3330</td>
<td>J2N3460</td>
<td>J2N3824</td>
<td>J2N4391</td>
</tr>
<tr>
<td>J2N5463</td>
<td>J2N5465</td>
<td>J309</td>
<td>J511</td>
<td>J557</td>
</tr>
<tr>
<td>JSJ74</td>
<td>JSK170</td>
<td>M2N6755</td>
<td>M2N6756</td>
<td>M2N6757</td>
</tr>
<tr>
<td>M2N6758</td>
<td>M2N6759</td>
<td>M2N6760</td>
<td>M2N6761</td>
<td>M2N6762</td>
</tr>
<tr>
<td>M2N6763</td>
<td>M2N6764</td>
<td>M2N6765</td>
<td>M2N6766</td>
<td>M2N6767</td>
</tr>
<tr>
<td>M2N6768</td>
<td>M2N6769</td>
<td>M2N6770</td>
<td>M2N6787</td>
<td>M2N6788</td>
</tr>
<tr>
<td>M2N6789</td>
<td>M2N6790</td>
<td>M2N6791</td>
<td>M2N6792</td>
<td>M2N6793</td>
</tr>
<tr>
<td>M2N6794</td>
<td>M2N6795</td>
<td>M2N6796</td>
<td>M2N6797</td>
<td>M2N6798</td>
</tr>
<tr>
<td>M2N6799</td>
<td>M2N6800</td>
<td>M2N6801</td>
<td>M2N6802</td>
<td>MBUZ10</td>
</tr>
<tr>
<td>MBUZ20</td>
<td>MBUZ23</td>
<td>MBUZ24</td>
<td>MBUZ32</td>
<td>MBUZ35</td>
</tr>
<tr>
<td>MBUZ36</td>
<td>MBUZ42</td>
<td>MBUZ45</td>
<td>MBUZ46</td>
<td>MBUZ60</td>
</tr>
<tr>
<td>MBUZ63</td>
<td>MBUZ64</td>
<td>MBUZ71</td>
<td>MBUZ72A</td>
<td>MBUZ74</td>
</tr>
<tr>
<td>MBUZ76</td>
<td>MIRF120</td>
<td>MIRF121</td>
<td>MIRF122</td>
<td>MIRF123</td>
</tr>
<tr>
<td>MIRF130</td>
<td>MIRF131</td>
<td>MIRF132</td>
<td>MIRF133</td>
<td>MIRF140</td>
</tr>
<tr>
<td>MIRF141</td>
<td>MIRF142</td>
<td>MIRF143</td>
<td>MIRF150</td>
<td>MIRF151</td>
</tr>
<tr>
<td>MIRF152</td>
<td>MIRF153</td>
<td>MIRF220</td>
<td>MIRF221</td>
<td>MIRF222</td>
</tr>
<tr>
<td>MIRF223</td>
<td>MIRF230</td>
<td>MIRF231</td>
<td>MIRF232</td>
<td>MIRF233</td>
</tr>
<tr>
<td>MIRF240</td>
<td>MIRF241</td>
<td>MIRF242</td>
<td>MIRF243</td>
<td>MIRF250</td>
</tr>
<tr>
<td>MIRF251</td>
<td>MIRF252</td>
<td>MIRF253</td>
<td>MIRF320</td>
<td>MIRF321</td>
</tr>
<tr>
<td>MIRF322</td>
<td>MIRF323</td>
<td>MIRF330</td>
<td>MIRF331</td>
<td>MIRF332</td>
</tr>
<tr>
<td>MIRF333</td>
<td>MIRF340</td>
<td>MIRF341</td>
<td>MIRF342</td>
<td>MIRF343</td>
</tr>
<tr>
<td>MIRF350</td>
<td>MIRF351</td>
<td>MIRF352</td>
<td>MIRF353</td>
<td>MIRF420</td>
</tr>
<tr>
<td>MIRF421</td>
<td>MIRF422</td>
<td>MIRF423</td>
<td>MIRF430</td>
<td>MIRF431</td>
</tr>
<tr>
<td>MIRF432</td>
<td>MIRF433</td>
<td>MIRF440</td>
<td>MIRF441</td>
<td>MIRF442</td>
</tr>
<tr>
<td>MIRF443</td>
<td>MIRF450</td>
<td>MIRF451</td>
<td>MIRF452</td>
<td>MIRF453</td>
</tr>
<tr>
<td>MIRF510</td>
<td>MIRF511</td>
<td>MIRF512</td>
<td>MIRF513</td>
<td>MIRF520</td>
</tr>
<tr>
<td>MIRF521</td>
<td>MIRF522</td>
<td>MIRF523</td>
<td>MIRF530</td>
<td>MIRF531</td>
</tr>
<tr>
<td>MIRF532</td>
<td>MIRF533</td>
<td>MIRF540</td>
<td>MIRF541</td>
<td>MIRF542</td>
</tr>
<tr>
<td>MIRF543</td>
<td>MIRF610</td>
<td>MIRF611</td>
<td>MIRF612</td>
<td>MIRF613</td>
</tr>
</tbody>
</table>
### Linear Technology Device Models

Search path: $\textit{installdir/parts/lin\_tech}$

Model names:

<table>
<thead>
<tr>
<th>113090</th>
<th>LF155</th>
<th>LF155A</th>
<th>LF156</th>
<th>LF156A</th>
</tr>
</thead>
<tbody>
<tr>
<td>LF355</td>
<td>LF355A</td>
<td>LF356</td>
<td>LF356A</td>
<td>LF412</td>
</tr>
<tr>
<td>LF412A</td>
<td>LM101A</td>
<td>LM107</td>
<td>LM108</td>
<td>LM108A</td>
</tr>
<tr>
<td>LM10C</td>
<td>LM118</td>
<td>LM301A</td>
<td>LM307</td>
<td>LM308</td>
</tr>
<tr>
<td>LM308A</td>
<td>LM318</td>
<td>LT1001</td>
<td>LT1001A</td>
<td>LT1006</td>
</tr>
<tr>
<td>LT1006A</td>
<td>LT1006S8</td>
<td>LT1007</td>
<td>LT1007A</td>
<td>LT1008</td>
</tr>
<tr>
<td>LT1012</td>
<td>LT1012A</td>
<td>LT1012D</td>
<td>LT1012S8</td>
<td>LT1013</td>
</tr>
<tr>
<td>LT1013A</td>
<td>LT1013D</td>
<td>LT1022</td>
<td>LT1022A</td>
<td>LT1028</td>
</tr>
<tr>
<td>LT1028A</td>
<td>LT1037</td>
<td>LT1037A</td>
<td>LT1055</td>
<td>LT1055A</td>
</tr>
<tr>
<td>LT1055S8</td>
<td>LT1056</td>
<td>LT1056A</td>
<td>LT1056S8</td>
<td>LT1057</td>
</tr>
</tbody>
</table>
Finding Device Libraries

Exercising the Library Listings

Intel PCI Speedway Models

Search path: \$installdir/parts/pci

Model names:

<table>
<thead>
<tr>
<th>Model names</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI_II_B</td>
</tr>
<tr>
<td>PCI_II_T</td>
</tr>
<tr>
<td>PCI_II_W</td>
</tr>
<tr>
<td>PCI_IN_W</td>
</tr>
<tr>
<td>TRACE</td>
</tr>
</tbody>
</table>

Signetics Device Models

Search path: \$installdir/parts/signet

Model names:

<table>
<thead>
<tr>
<th>Model names</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC109EQ</td>
</tr>
<tr>
<td>AC240EQ</td>
</tr>
<tr>
<td>AC833OD</td>
</tr>
<tr>
<td>AC86EQ</td>
</tr>
<tr>
<td>ACTINPUT</td>
</tr>
<tr>
<td>CL10X10</td>
</tr>
<tr>
<td>CL30X10</td>
</tr>
<tr>
<td>DIP14</td>
</tr>
<tr>
<td>DIP16</td>
</tr>
<tr>
<td>DIP20</td>
</tr>
<tr>
<td>DIP24</td>
</tr>
<tr>
<td>DIP28</td>
</tr>
<tr>
<td>DIP8</td>
</tr>
<tr>
<td>DIP80001</td>
</tr>
<tr>
<td>SO16</td>
</tr>
<tr>
<td>SO20</td>
</tr>
<tr>
<td>SO24</td>
</tr>
<tr>
<td>SO28</td>
</tr>
<tr>
<td>SO8</td>
</tr>
</tbody>
</table>
**Texas Instruments Device Models**

Search path: `$installdir/parts/ti`

Model names:

<table>
<thead>
<tr>
<th>ICL7652</th>
<th>LF347</th>
<th>LF351</th>
<th>LF353</th>
<th>LF411C</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM308</td>
<td>LM318</td>
<td>LM324</td>
<td>LM348</td>
<td>LM358</td>
</tr>
<tr>
<td>LT1001</td>
<td>LT1008</td>
<td>LT1012</td>
<td>LT1013</td>
<td>LT1028</td>
</tr>
<tr>
<td>LT1037</td>
<td>LTC1052</td>
<td>MC1458</td>
<td>MC3403</td>
<td>NE5534</td>
</tr>
<tr>
<td>OP-27G</td>
<td>OP-37A</td>
<td>RC4136</td>
<td>RC4558</td>
<td>RC4559</td>
</tr>
<tr>
<td>TL022C</td>
<td>TL031</td>
<td>TL032</td>
<td>TL034</td>
<td>TL044C</td>
</tr>
<tr>
<td>TL051</td>
<td>TL052</td>
<td>TL054</td>
<td>TL060</td>
<td>TL061</td>
</tr>
<tr>
<td>TL062</td>
<td>TL064</td>
<td>TL066</td>
<td>TL070</td>
<td>TL071</td>
</tr>
<tr>
<td>TL072</td>
<td>TL074</td>
<td>TL075</td>
<td>TL080</td>
<td>TL081</td>
</tr>
<tr>
<td>TL082</td>
<td>TL083</td>
<td>TL084</td>
<td>TL085</td>
<td>TL087</td>
</tr>
<tr>
<td>TL088</td>
<td>TL136</td>
<td>TL287</td>
<td>TL288</td>
<td>TL321</td>
</tr>
<tr>
<td>TL322</td>
<td>TLC1078</td>
<td>TLC1079</td>
<td>TLC2201</td>
<td>TLC251H</td>
</tr>
<tr>
<td>TLC251L</td>
<td>TLC251M</td>
<td>TLC252C</td>
<td>TLC254C</td>
<td>TLC25L2C</td>
</tr>
<tr>
<td>TLC25L4C</td>
<td>TLC25M2C</td>
<td>TLC25M4C</td>
<td>TLC2652</td>
<td>TLC2654</td>
</tr>
<tr>
<td>TLC271H</td>
<td>TLC271L</td>
<td>TLC271M</td>
<td>TLC272</td>
<td>TLC274</td>
</tr>
<tr>
<td>TLC277</td>
<td>TLC279</td>
<td>TLC27L2</td>
<td>TLC27L4</td>
<td>TLC27L7</td>
</tr>
<tr>
<td>TLC27L9</td>
<td>TLC27M2</td>
<td>TLC27M4</td>
<td>TLC27M7</td>
<td>TLC27M9</td>
</tr>
<tr>
<td>TLE2021</td>
<td>TLE2022</td>
<td>TLE2024</td>
<td>TLE2061</td>
<td>TLE2062</td>
</tr>
<tr>
<td>TLE2064</td>
<td>TLE2161</td>
<td>UA741</td>
<td>UA747</td>
<td>UA748</td>
</tr>
</tbody>
</table>
Transmission Line Models

Search path:  $installdir/parts/tline

Model names:

<table>
<thead>
<tr>
<th>RCFILT</th>
<th>RG11_U</th>
<th>RG11A_U</th>
<th>RG15_U</th>
<th>RG180B_U</th>
</tr>
</thead>
<tbody>
<tr>
<td>RG188A_U</td>
<td>RG53_U</td>
<td>RG54A_U</td>
<td>RG58A_U</td>
<td>RG58C_U</td>
</tr>
<tr>
<td>RG59B_U</td>
<td>RG62_U</td>
<td>RG62B_U</td>
<td>RG71_U</td>
<td>RG71B_U</td>
</tr>
<tr>
<td>RG9_U</td>
<td>RG9B_U</td>
<td>TW_SH_U</td>
<td>TW_UN_U</td>
<td></td>
</tr>
</tbody>
</table>

Xilinx Device Models

Search path:  $installdir/parts/xilinx

Model names:

<table>
<thead>
<tr>
<th>FOUTPUT</th>
<th>OUTPUT</th>
<th>XC7236A</th>
<th>XC7272A</th>
<th>XC7336A</th>
</tr>
</thead>
<tbody>
<tr>
<td>XIL_IOB</td>
<td>XIL_IOB4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix E

Performing Library Encryption

Library encryption allows you to distribute proprietary Star-Hspice models, parameters and circuits to other people without revealing your company’s sensitive information. Recipients of an encrypted library can run simulations that use your libraries, but Star-Hspice does not print encrypted parameters, encrypted circuit netlists, or internal node voltages. Your library user sees the devices and circuits as “black boxes,” which provide terminal functions only.

Use the library encryption scheme primarily to distribute circuit blocks with embedded transistor models, such as ASIC library cells and I/O buffers. Star-Hspice uses subcircuit calls to read encrypted information. To distribute device libraries only, create a unique subcircuit file for each device.

This chapter describes Avant!’s Library Encryptor and how to use it to protect your intellectual property. The following topics are covered in the chapter:

- Understanding Library Encryption
- Knowing the Encryption Guidelines
- Installing and Running the Encryptor
- Understanding Metaencrypt Features
Understanding Library Encryption

The library encryption algorithm is based on that of a five-rotor Enigma machine. The encryption process allows the user to specify which portions of subcircuits are encrypted. The libraries are encrypted using a key value that Star-Hspice reconstructs for decryption.

Controlling the Encryption Process

To control the beginning and end of the encryption process, insert .PROTECT and .UNPROTECT statements around text to be encrypted in an Star-Hspice subcircuit. The encryption process produces an ASCII text file in which all text that follows .PROTECT and precedes .UNPROTECT is encrypted.

Note: The Star-Hspice .PROTECT and .UNPROTECT statements often are abbreviated to .PROT and .UNPROT, respectively. Either form may be used in Star-Hspice input files.

Library Structure

The requirements for encrypted libraries of subcircuits are the same as the requirements for regular subcircuit libraries. Subcircuit library structure requirements are described in “Subcircuit Library Structure” on page 3-52. Refer to an encrypted subcircuit by using its subcircuit name in a subcircuit element line of the Star-Hspice netlist.

The following example provides the description of an encrypted I/O buffer library subcircuit. This subcircuit is constructed of several subcircuits and model statements that you need to protect with encryption. Figure E-1 shows the organization of subcircuits and models in libraries used in this example.
The following input file fragment from the main circuit level selects the *Fast* library and creates two instances of the *iobuf* circuit.

```plaintext
... .Option Search='<LibraryDir>/Fast' $ Corner Spec x1 drvin drvout *iobuf* Cload=2pF $ Driver u1 drvout 0 recvin 0 PCBModel... $ Trace x2 recvin recvout *iobuf* $ Receiver ...```

---

**Figure E-1: Encrypted Library Structure**

The image shows a diagram of the encrypted library structure, illustrating the design view and file system organization, with specific file paths and library components.
The file `<LibraryDir>/Fast/iobuf.inc` contains:

```
.Subckt iobuf Pin1 Pin2 Cload=1pF
* * iobuf.inc - model 2001 improved iobuf
* .PROTECT
cPin1 Pin1 0 1pF $ Users can’t change this!
x1 Pin1 Pin2 ioinv $ Italics here means encrypted
.Model pMod pmos Level=28 Vto=... $ <FastModels>
.Model nMod nmos Level=28 Vto=... $ <FastModels>
.UNPROTECT
cPin2 Pin2 0 Cload $ give you some control
.Ends
```

The file `<LibraryDir>/Fast/ioinv.inc` contains:

```
.Subckt ioinv Pin1 Pin2
.PROTECT
mp1 Vcc Pin1 Pin2 Vcc pMod... $ Italics=Protected
mn1 Pin2 Pin1 Gnd Gnd nMod... $ Italics=Protected
.UNPROTECT
.Ends
```

After encryption, the basic layout of the subcircuits is the same. However, the text between .PROTECT and .UNPROTECT statements is unreadable, except by Star-Hspice.

The protection statements also suppress printouts of encrypted model information from Star-Hspice. Only Star-Hspice knows how to decrypt the model.
Knowing the Encryption Guidelines

In general, there are no differences between using the encrypted models and using regular models. However, you must test your subcircuits before encryption. You will not be able to see what has gone wrong after encryption because of the protection offered by Star-Hspice.

Use any legal Star-Hspice statement inside your subcircuits to be encrypted. Refer to “.SUBCKT or .MACRO Statement” on page 3-12 for further information on subcircuit construction. You must take care when structuring your libraries. If your library scheme requires that you change the name of a subcircuit, you must encrypt that circuit again.

Placement of the .PROTECT and .UNPROTECT statements allows your customers to see portions of your subcircuits. If you protect only device model statements in your subcircuits, your users can set device sizes or substitute different subcircuits for lead frames, protection circuits, and so on. This requires your users to know the circuits, but it reduces the library management overhead for everyone.

---

**Note:** If you are running any version of the encryptor prior to Star-Hspice Release H93A.03, there is a bug that prevents Star-Hspice from correctly decrypting a subcircuit if that subcircuit contains any semicolon (;) characters, even in comments.

---

In the following example, the subcircuit `badsemi.dat` is encrypted into `badsemi.inc`.

```plaintext
* Sample semicolon bug
.SubCkt BadSemi A B
.PROT
* Semicolons (;) cause problems!
  r1 A B 1k
.UNPROT
.Ends
```
Star-Hspice responds with the following message:

**reading include file=badsemi.inc**
**error**: .ends card missing at readin
>error  ***difficulty in reading input

To solve this problem, remove the semicolon from *badsemi.dat* and encrypt the file again.

Prior to the 2001.2 version, Metaencrypt cannot encrypt files with lines longer than 80 characters. Metaencrypt version 2001.2 can encrypt files with lines longer than 80 characters but cannot correctly encrypt files with lines longer than 254 characters. Avant! strongly recommends that all files to be encrypted be limited to a 254-character line length and use version 2001.2 and above to encrypt any files with lines longer than 80 characters. Star-Hspice can recognize all correctly encrypted files.

You cannot gather the individually-encrypted files into a single file or include them directly in the Star-Hspice netlist. Place them in a separate directory pointed to by the `.OPTION SEARCH = <dir> named <sub>.inc` for correct decryption by Star-Hspice.
Installing and Running the Encryptor

This section describes how to install and run the Encryptor.

Installing the Encryptor

If Star-Hspice is already installed on your system, place the Encryptor in the directory $installdir/bin to install it. Add the lines that allow the Encryptor to operate to your permit.hsp file in the $installdir/bin directory.

If Star-Hspice is not installed on your system, first install Star-Hspice according to the installation guide and Star-Hspice Release Notes included in your Star-Hspice package, and then follow the instructions in the previous paragraph.

Note: If you are running a floating license server, you must stop and restart the server to see the changes to the permit file.

Running the Encryptor

The Encryptor requires three parameters for each subcircuit encrypted: <InFileName>, <OutFileName>, and the key type specifier, Freelib. Enter the following line to encrypt a file.

```
metaencrypt -i <InfileName> -o <OutFileName> -t Freelib
```

As the Encryptor reads the input file, it looks for .PROT/.UNPROT pairs and encrypts the text between them. You can encrypt only one file at a time.
To encrypt many files in a directory, use the following shell script to encrypt the files as a group. This script produces a .inc encrypted file for each .dat file in the current directory. The procedure assumes that the unencrypted files are suffixed with .dat.

```
#!/bin/sh
for i in *.dat
  do
    Base=`basename $i .dat`
    metaencrypt -i $Base.dat -o $Base.inc -t Freelib
  done
.SUBCKT ioinv Pin1 Pin2
.PROT FREELIB $ Encryption starts here ...
X34%43*27@#^3rx*34%%^1
^(*^!^HJHD(*@H$!:&*$
dFE2341&*&)(@3 $ ... and stops here (.UNPROT is
encrypted!)
.ENDS
Understanding Metaencrypt Features

This section describes new features and enhancements to the metaencrypt functionality supported in version 2001.2 and later.

New 8-Byte Key Encryption

A new 8-byte key encryption feature based on a 56-bit DES is now available in metaencrypt with versions 2001.2 and above. You can insert data into an include file and encrypt this file using 8-byte key encryption. The encrypted data is in binary format.

Syntax

```bash
metaencrypt -i <infileName> -o <outfileName> -t <keyname>
```

**Note:** The keyname can be a combination of English characters and numbers and should be limited to 8 bytes.

Example

```bash
metaencrypt -i example.dat -o example.inc -t fGi85H9b
```

Encryption Structure

In version 2001.2 and later, .inc encryption is supported when using 8-byte key encryption. Insert the data to be encrypted into an include file and encrypt this file.

In the following example, example.dat contains data to be encrypted.

```bash
metaencrypt -i example.dat -o example.inc -t h78Gbvni
```

*example.sp*

......
Understanding Metaencrypt Features

Performing Library Encryption

.inc example.inc $ example.inc contains encrypted data
......
.end

Consider the following rules when employing 8-byte key encryption:
■ 8-byte key encryption only supports .inc encryption
■ 8-byte key encryption does NOT support .lib, .load or .option search encryption in the 2001.2 version
■ .prot and .unpr should NOT be included with the data while performing 8-byte key encryption
■ if keyname=ddl1, ddl2, custom, freelib, then metaencrypt will user a former algorithm to perform the encryption
■ if keyname is an 8-byte string (combination of characters and numbers), then metaencrypt will perform the 8-byte key encryption

Supporting the .sp File Encryption

You can encrypt a .sp file in Star-Hspice version 2001.2. The data between .prot and .unpr in a .sp file can be encrypted so that Star-Hspice will recognize it.

Note: When performing the .sp encryption, the encrypted data should not contain .inc, .lib or .load to include another file.

Example
*sampole.sp*
......
.inc sample1.inc

.prot
...... $ data to be encrypted
...... $ .inc .lib .load should not be contained in
encrypted data
.unpr

.inc sample2.inc
......
.end

Supporting .lib File Encryption

Any important information can be placed into a .lib file and can be encrypted.

You can place parallel .lib statements into one library file. Each .lib can then be encrypted separately. However, .prot and .unpr must be placed between each pair of .lib and .endl.

---

*Note:*. .prot and .unpr must be placed between .lib and .endl since Hspice will first find the library name in the .lib file.

---

**Example**

*sample.sp*

......
.lib `./sample.lib’ test1
.lib `./sample.lib’ test2
.lib `./sample.lib’ test3
......
.end

*sample.lib*

.lib test1 $ .prot , .unpr should be put between .lib and .endl
.prot
...... $ data to be encrypted
.unpr
... $ data not to be encrypted
.end1 test1

.lib test2 $ .prot , .unpr should be put between .lib and .end1
.prot
... $ data to be encrypted
.unpr
.end1 test2

.lib test3
... $ data
.end1 test3

### Supporting .inc File Encryption

You can insert any important data into a .inc file and encrypt the file. There are no restrictions on the placement of .prot and .unpr.

**Example**

*sample.sp*

......
.inc sample.inc
......
.end

*sample.inc*

.prot $ no restriction on the placement of .prot
......$ data to be encrypted should not contain .inc, .lib or .load
.unpr
......
Performing Library Encryption
Understanding Metaencrypt Features

Supporting .load Encryption
You can encrypt a .ic file just like a .inc file.

Supporting 80+ Columns Encryption
Star-Hspice version 2001.2 and later can support 1-255 column encryption.

Statements Not Supported
- Embedded .lib encryption as this will cause confusion in decryption
- Embedded .inc encryption as this will cause confusion in decryption
- Data should not contain .inc, .lib or .load statements to include another .inc, .lib or .ic file

Additional Recommendations for Encryption
Using .option search for encrypting models and subcircuits is not recommended. This method was supported using the old metaencryption functionality. Subcircuits and model libraries can be encrypted directly using the .inc and .lib encryption method.

Complete Encryption Structure Example
The following complete example illustrates the encryption structure.

file enc.sp:

*test .inc .lib .load encryption

.inc "mm.inc"
.load "xx.ic"
.lib ‘kk.lib’ pch

.options post list
.tran 2ns 400ns
Understanding Metaencrypt Features
Performing Library Encryption

.file mm.inc:

.prot CUSTOM
-hs#yIB)*7[
+t’Y=O$[t0]ajL
+C :Nx:$.=(*X:$.pP=020#ZWP=020x[K:[1:898
y[-x:$.tRr0($x#4://U$<K:I[U$<J<9 :P#ZQ
6%P2V7D6;41/0#:IXj0#ZWP=020#ZWP/[U$=J++bZ
3[7D6:BXHpg8
/C902P73+26
mh$y#D:bX/$KwI)U-OR#=-ib+\{
a$0) :P.##)< :P.#to)\7*K-I1M$#’;-[Xz:9qpy
eMDv0%WuoxZ>izzW*-(3_;W6x.*P!uW.]a+P0.h:n=O>1q+H(J0
o.H#-/B+(;$W Me*0x<6#9[UqpH/2h97%;/-B+T35Q
$m;’_he[uE$%H] 5a:ZxRW9x=*77w$2]=*P!RW%.ahT3VQ
H0[I:

.file xx.ic:

.prot FREELIB
59yUH\$=‘x.3k77}*\8AT\8
<:7-(:9CV+7x15Xj+h’x=5Xj+(2 +4)8
<:7_D:\[2x9Y>/.7q
59y3\#D$ *y2k=u]PIq:97jH=ulw5Xj+x6
92k<2FW0’k772<xBU677Q
59y3\#s# r21$,29b72[4’/RW72wd#$0.0
0sv%5$; [4sv; 9=zV7[WFW[(g8#/’]=AH%T5:7Z
[$%C999A2P!8
<:X2o60$ 06($_#pel:pX8
Performing Library Encryption
Understanding Metaencrypt Features

<5ax/toC n90;<0dw0]23G%C z9$Dh#Sw5a90
ZM*2!M0
o729!=PAy73x(/l:6[
+ 0%2UT%8
_:-x*$X+q
$9P2y73x(/l:L
T#;*9A27!j+(/z
$$o#(:/b0
o7ZW-9 PxJ+y
a9[$0;\n90;<0dw0]23G%C z9$Dh#Sw5a90
Zr;6

file kk.lib:

.LIB NCH
.prot FREELIB
.HO. T,# %fXz>MZWF*-(3_;w6X.*p!Uw.]A+p0.H:N=o>1Q+h(j0
o.H#-/B+(S;W Me*0x<6#9[UqpH/2h97%;-/B+T35Q
$\m;'_-_he[uE$%H] 5a:ZxRW9x=*77w$2]=*P!RW%.ahT3VQ
H0[I:
. ENDL
.LIB PCH
.prot FREELIB
.HO. T,# %t%fXz>MZWF*-(3_;w6X.*p!Uw.]A+p0.H:N=o>1Q+h(j0
o.H#-/B+(S;W Me*0x<6#9[UqpH/2h97%;-/B+T35Q
$\m;'_-_he[uE$%H] 5a:ZxRW9x=*77w$2]=*P!RW%.ahT3VQ
H0[I:
. ENDL
Appendix F

Full Simulation Examples

The examples in this chapter display the basic text and post-processor output for a sample input netlist. The first example uses AvanWaves to view results and the second example uses Cosmos-Scope.

This chapter includes the following sections:

- Full Simulation Example with AvanWaves
- Full Simulation Example with Cosmos-Scope
Full Simulation Example with AvanWaves

Input Netlist and Circuit

The input netlist for the linear CMOS amplifier is shown below. The individual sections of the netlist are indicated using comment lines. Please refer to “Specifying Simulation Input and Controls” on page 3-1 for information about the individual commands.

* Example HSPICE netlist using a linear CMOS amplifier

* netlist options
    .option post probe brief nomod

* defined parameters
    .param analog_voltage=1.0

* global definitions
    .global vdd

* source statements
    Vinput in gnd SIN ( 0.0v analog_voltage 10x )
    Vsupply vdd gnd DC=5.0v

* circuit statements
    Rinterm in gnd 51
    Cin_cap in infilt 0.001
    Rdamp infilt clamp 100
    Dlow gnd clamp diode_mod
    Dhigh clamp vdd diode_mod
    Xinv1 clamp invlout inverter
    Rpull clamp invlout 1x
    Xinv2 invlout inv2out inverter
    Routerm inv2out gnd 100x

* subcircuit definitions
    .subckt inverter in out
    Mpmos out in vdd vdd pmos_mod l=1u w=6u
    Mnmos out in gnd gnd nmos_mod l=1u w=2u
    .ends
The following is the circuit diagram for the linear CMOS amplifier that is described in the circuit portion of the netlist. The two sources shown in the diagram are also in the netlist. Please note that the inverter symbols shown in the circuit diagram are constructed from two complementary MOSFET elements. Also, the diode and MOSFET models in the netlist were not given any non-default parameter values, except to specify Level 3 MOSFET models (empirical model).

![Figure F-1: Circuit Diagram for Linear CMOS Inverter](image)
Execution and Output Files

The following section displays the various output files from a Star-Hspice simulation of the amplifier shown in the previous section. The simulation was executed by entering:

```
  hspice example.sp > example.lis
```

where the input netlist was named `example.sp` and the output listing was named `example.lis`. The following output files were created with a brief explanation of their content.

- `example.ic`: initial conditions for the circuit
- `example.lis`: text simulation output listing
- `example.mt0`: post-processor output for MEASURE statements
- `example.pa0`: subcircuit path table
- `example.st0`: run-time statistics
- `example.tr0`: post-processor output for transient analysis

The following subsections show the text files in their entirety for the amplifier simulation performed using Star-Hspice 1998.4 on a Sun workstation. The two post-processor output files cannot be shown because they are in binary format.

Example.ic

```
* "simulator" "HSPICE"
* "version" "98.4 (981215) "
* "format" "HSP"
* "rundate" "13:58:43  01/08/1999"
* "netlist" "example.sp "
* "runtime" "* example hspice netlist using a linear cmos amplifier "
* time =  0.
* temperature =  25.0000
*** BEGIN: Saved Operating Point ***
.option
+ gmindc= 1.0000p
.node
```
Example.lis

Using: /net/sleepy/l0/group/hspice/98.4beta/sol4/hspice

****** Star-HSPICE -- 98.4 (981215) 13:58:43 01/08/1999 solaris
Copyright (C) 1985-1997 by Avant! Corporation.
Unpublished-rights reserved under US copyright laws.
This program is protected by law and is subject to the
terms and conditions of the license agreement found in:
/afs/rtp.avanticorp.com/product/hspice/current/license.txt
Use of this program is your acceptance to be bound by this
license agreement. Star-HSPICE is the trademark of Avant! Corporation.
Input File: example.sp
lic:
lic: FLEXlm:v5.12 USER:hspicuser HOSTNAME:hspicuser HOSTID:8086420f PID:1459
lic: Using FLEXlm license file:
lic: /afs/rtp/product/distrib/bin/license/license.dat
lic: Checkout hspice; Encryption code: AC34CE559E01F6E05809
lic: License/Maintenance for hspice will expire on 14-apr-1999/1999.200
lic: 1(in_use)/10 FLOATING license(s) on SERVER hspiceserv
lic:
******
* example hspice netlist using a linear cmos amplifier

******
* netlist options
.option post probe brief nomod

* defined parameters
Opening plot unit= 15
file=./example.pao

****** Star-HSPICE -- 98.4 (981215) 13:58:43 01/08/1999 solaris******
* example hspice netlist using a linear cmos amplifier

****** transient analysis 
tnom= 25.000 temp= 25.000 ******

*** parameter analog_voltage = 1.000E+00 ***

node =voltage node =voltage node =voltage
Full Simulation Example with AvanWaves

+0:clamp = 2.6200 0:in = 0. 0:infilt = 2.6200
+0:inv1out = 2.6200 0:inv2out = 2.6199 0:vdd = 5.0000

Opening plot unit= 15
file=./example.tr0

**warning** negative-mos conductance = 1:mnmos
vds,vgs,vbs = 2.45 2.93 0.
gm,gds,gmb,ids = -3.636E-05 1.744E-04 0. 1.598E-04
******
* example hspice netlist using a linear cmos amplifier
****** transient analysis tnom= 25.000 temp= 25.000 ******
falltime= 3.9149E-08 targ= 7.1916E-08 trig= 3.2767E-08

****** Star-HSPICE -- 98.4 (981215) 13:58:43 01/08/1999 solaris ******
* example hspice netlist using a linear cmos amplifier
****** transient analysis tnom= 25.000 temp= 25.000 ******

*** parameter analog_voltage = 2.000E+00 ***
node =voltage node =voltage node =voltage

+0:clamp = 2.6200 0:in = 0. 0:infilt = 2.6200
+0:inv1out = 2.6200 0:inv2out = 2.6199 0:vdd = 5.0000

******
* example hspice netlist using a linear cmos amplifier
****** transient analysis tnom= 25.000 temp= 25.000 ******
falltime= 1.5645E-08 targ= 5.7994E-08 trig= 4.2348E-08

****** Star-HSPICE -- 98.4 (981215) 13:58:43 01/08/1999 solaris ******
* example hspice netlist using a linear cmos amplifier
****** transient analysis tnom= 25.000 temp= 25.000 ******

*** parameter analog_voltage = 3.000E+00 ***
node =voltage node =voltage node =voltage

+0:clamp = 2.6200 0:in = 0. 0:infilt = 2.6200
+0:inv1out = 2.6200 0:inv2out = 2.6199 0:vdd = 5.0000
* example hspice netlist using a linear cmos amplifier

*** parameter analog_voltage = 4.000E+00 ***

node  =voltage  node  =voltage  node  =voltage

+0:clamp  =  2.6200  0:in  =  0.  0:infilt  =  2.6200
+0:invlout  =  2.6200  0:inv2out  =  2.6199  0:vdd  =  5.0000

* example hspice netlist using a linear cmos amplifier

*** parameter analog_voltage = 5.000E+00 ***

node  =voltage  node  =voltage  node  =voltage

+0:clamp  =  2.6200  0:in  =  0.  0:infilt  =  2.6200
+0:invlout  =  2.6200  0:inv2out  =  2.6199  0:vdd  =  5.0000

meas_variable = falltime
mean  =  16.0848n  varian  =  1.802e-16
sigma  =  13.4237n  avgdev  =  9.2256n
max  =  39.1488n  min  =  6.1706n

***** job concluded
Full Simulation Example with AvanWaves

Full Simulation Examples

****** Star-HSPICE -- 98.4 (981215) 13:58:43 01/08/1999 solaris ******
* example hspice netlist using a linear cmos amplifier
****** job statistics summary tnom= 25.000 temp= 25.000 ******
total memory used 155 kbytes
# nodes = 8 # elements= 14
# diodes= 2 # bjts = 0 # jfets = 0 # mosfets = 4
analysis time # points tot. iter conv.iter
op point 0.04 1 23
transient 4.71 505 9322 2624 rev= 664
readin 0.03
errchk 0.01
setup 0.01
output 0.01
total cpu time 4.84 seconds
job started at 13:58:43 01/08/1999
job ended at 13:58:50 01/08/1999

HSPICE job example.sp completed.
Fri Jan  8 13:58:50 EST 1999

Example.pa0
1 xinv1.
2 xinv2.

Example.st0
****** Star-HSPICE -- 98.4 (981215) 13:58:43 01/08/1999 solaris
Input File: example.sp
lic:
lic: Release hspice token(s)
HSPICE job example.sp completed.
Fri Jan  8 13:58:50 EST 1999
establish matrix -- done, cpu clock= 2.24E+00 memory= 146 kb
re-order matrix -- done, cpu clock= 2.24E+00 memory= 146 kb
init: end setup matrix, cpu clock= 2.25E+00 memory= 154 kb
sweep: parameter parameter1 begin, #sweeps= 5
parameter: analog_voltage = 1.00E+00
dcop: begin dcop, cpu clock= 2.25E+00
dcop: end dcop, cpu clock= 2.27E+00 memory= 154 kb  tot_iter= 11
output: ./example.mt0
sweep: tran tran1 begin, stop_t= 1.00E-06 #sweeps= 101 cpu clock= 2.28E+00
tran: time= 1.03750E-07 tot_iter= 78 conv_iter= 24
tran: time= 2.03750E-07 tot_iter= 179 conv_iter= 53
tran: time= 3.03750E-07 tot_iter= 280 conv_iter= 82
tran: time= 4.03750E-07 tot_iter= 381 conv_iter= 111
tran: time= 5.03750E-07 tot_iter= 482 conv_iter= 140
tran: time= 6.03750E-07 tot_iter= 583 conv_iter= 169
tran: time= 7.03750E-07 tot_iter= 684 conv_iter= 198
tran: time= 8.03750E-07 tot_iter= 785 conv_iter= 227
tran: time= 9.03750E-07 tot_iter= 886 conv_iter= 256
tran: time= 1.00000E-06 tot_iter= 987 conv_iter= 285
sweep: tran tran1 end, cpu clock= 2.82E+00 memory= 155 kb
parameter: analog_voltage = 2.00E+00
dcop: begin dcop, cpu clock= 2.83E+00
dcop: end dcop, cpu clock= 2.83E+00 memory= 155 kb  tot_iter= 14
output: ./example.mt0
sweep: tran tran2 begin, stop_t= 1.00E-06 #sweeps= 101 cpu clock= 2.83E+00
tran: time= 1.00313E-07 tot_iter= 143 conv_iter= 42
tran: time= 2.01211E-07 tot_iter= 340 conv_iter= 100
tran: time= 3.01850E-07 total_iter= 539 conv_iter= 156
tran: time= 4.02192E-07 total_iter= 729 conv_iter= 211
tran: time= 5.01997E-07 total_iter= 917 conv_iter= 265
tran: time= 6.01801E-07 total_iter= 1088 conv_iter= 314
tran: time= 7.01800E-07 total_iter= 1221 conv_iter= 351
tran: time= 8.01801E-07 total_iter= 1362 conv_iter= 392
tran: time= 9.02387E-07 total_iter= 1515 conv_iter= 435
sweep: tran tran2 end, cpu clock= 3.71E+00 memory= 155 kb  tot_iter= 17
parameter: analog_voltage = 3.00E+00
dcop: begin dcop, cpu clock= 3.71E+00
dcop: end dcop, cpu clock= 3.72E+00 memory= 155 kb  tot_iter= 17
output: ./example.mt0
sweep: tran tran3 begin, stop_t= 1.00E-06 #sweeps= 101 cpu clock= 3.72E+00
tran: time= 1.00313E-07 total_iter= 143 conv_iter= 42
tran: time= 2.01211E-07 total_iter= 340 conv_iter= 100
tran: time= 3.01850E-07 total_iter= 539 conv_iter= 156
tran: time= 4.02192E-07 total_iter= 729 conv_iter= 211
tran: time= 5.01997E-07 total_iter= 917 conv_iter= 265
tran: time= 6.01801E-07 total_iter= 1088 conv_iter= 314
tran: time= 7.01800E-07 total_iter= 1221 conv_iter= 351
tran: time= 8.01801E-07 total_iter= 1362 conv_iter= 392
tran: time= 9.02387E-07 total_iter= 1515 conv_iter= 435
Simulation Graphical Output in AvanWaves

The following plots show the six different post-processor outputs from the simulation of the example netlist, as seen in AvanWaves, the graphical waveform viewer available from Avant! These plots are postscript output from the actual data.
Figure F-2: Plot of Voltage on Node *in*

Figure F-3: Plot of Voltage on Node *clamp vs. Time*
Figure F-4: Plot of Voltage on Node \textit{inv1out} vs. Time

Figure F-5: Plot of Voltage on Node \textit{inv2out} vs. Time
Figure F-6: Plot of Current through Diode $d_{low}$ vs. Time

Figure F-7: Plot of User-defined Measured Variable $falltime$ vs. Amplifier Input Voltage
This example demonstrates the basic steps to perform simulation output and view the waveform results using Avant!’s Cosmos-Scope Waveform Viewer.

Input Netlist and Circuit

The input netlist for a BJT diff amplifier is given under Syntax. The individual sections of the netlists are indicated using comment lines. Please refer to “Specifying Simulation Input and Controls” on page 3-1 for information about the individual commands.

Syntax

```plaintext
*file: bjtdiff.sp bjt diff amp with every analysis type
**
** # ANALYSIS: ac dc tran tf noise new four sens pz disto temp
** # OPTIONS: list node ingold=2 measdgt=5 numdgt=8 probe post
** # TEMPERATURE: 25
*
* netlist options
.OPTIONS list node ingold=2 measdgt=6 numdgt=8 probe post
* defined parameters
.PARAM rb1x=aunif(20k,1k,30k) rb2x=aunif(20k,1k,30k)

* analysis specifications
.TF v(5) vin
.DC vin -0.20 0.20 0.01 sweep monte=3
.AC dec 10 100k 10meghz
.NOISE v(4) vin 20
.NET v(5) vin rout=10k
.PZ v(5) vin
.DISTRO rcl 20 .9 1m 1.0
.SENS v(4)
.TRAN 5ns 200ns
.FOUR 5meg v(5) v(15)
.TEMP -55 150

* output specifications
.MEAS qa_propdly trig v(1) val=0.09 rise=1
```
Full Simulation Example with Cosmos-Scope

+                                targ v(5) val=6.8  rise=1
.MEAS qa_magnitude max v(5)
.MEAS qa_rmspower rms power
.MEAS qa_avgv5 avg v(5)
.MEAS ac qa_bandwidth trig at=100k targ vdb(5) val=36 fall=1
.MEAS ac qa_phase find vp(5) when vm(5)=52.12
.MEAS ac qa_freq when vm(5)=52.12
.PROBE dc v(4) v(5) v(14) v(15)
.PROBE ac vm(5) vp(5) vm(15) vp(15)
.PROBE ac vt(5) vt(15)
.PROBE noise onoise(m) inoise(m)
.PROBE ac z11(m) z12(m) z22(m) zin(m)
.PROBE disto hd2 hd3 sim2 dim2 dim3
.PROBE tran v(4) v(5) v(14) v(15)
.PROBE tran p(vcc) p(vee) p(vin) power

* source statements
VIN 1 0 sin(0 0.1 5meg) ac 1
VCC 8 0 12
VEE 9 0 -12

* circuit statements
q1  4  2  6 qnl
q11 14 12 16 qpl
q2  5  3  6 qnl
q21 15 13 16 qpl
rs1  1  2 1k
rs11 1 12 1k
rs2  3  0 1k
rs12 13  0 1k
rc1  4  8 10k
rc11 14  9 10k
rc2  5  8 10k
rc12 15  9 10k
q3  6  7  9 qnl
q13 16 17  8 qpl
q4  7  7  9 qnl
q14 17 17  8 qpl
rb1  7  8 rb1x
rb2  17  9 rb2x
Try using the previous example (linear CMOS amp) to draw a circuit diagram for this BJT diff amplifier. Also, specify parameter values.

Execution and Output Files

This section displays the various output files from a Star-Hspice simulation of the BJT diff amplifier example. The simulation was executed by entering:

```
hspice bjtdiff.sp > bjtdiff.lis
```

where the input file is bjtdiff.sp and the output file is bjtdiff.lis. This is a list of the output files that were created, with a brief explanation of their content.

- bjtdiff.ic: Initial conditions for the circuit
- bjtdiff.lis: Text simulation output listing
- bjtdiff.mt0: Post-processor output for MEASURE statements
- bjtdiff.st0: Run-time statistics
- bjtdiff.tr0: Post-processor output for transient analysis
- bjtdiff.sw0: Post-processor output for DC analysis
- bjtdiff.ac0: Post-processor output for AC analysis
- bjtdiff.ma0: Post-processor output for AC analysis measurements

Using Cosmos-Scope to View Star-Hspice Results

The following steps show how to use the Avant! Cosmos-Scope Waveform Viewer to view the AC, DC, and transient analysis results from the BJT diff amplifier simulation. Refer to previous examples to see a sample of .lis, .ic, and .st0 files.
Viewing Star-Hspice Transient Analysis Waveforms

1. Invoke Cosmos-Scope.

Scope can be invoked from a UNIX command: `cscope`. On a
Windows-NT system, Scope can be opened by choosing the
Programs > (user_install_location)> Cosmos-Scope.

2. Open the Open Plotfiles dialog box: File > Open > Plotfiles.

3. In the Open Plotfiles dialog box, in the Files of Type fields, select the
Hspice Transient (*.tr*) item.

4. You will see bjtdiff.tr0 in the menu. Click on it and click Open. The Signal
Manager and bjtdiff Plot File windows are displayed.

5. While holding down the Ctrl key, select the v(4), v(5), and
ITPOWERD(power) signals.

6. Click on Plot from the bjtdiff Plot File window. You will see three cascaded
plots. To see three signals in one plot, right-click on the name of top-most
signal to get a Signal Menu.

7. From the Signal Menu, select Stack Region > Analog 0.

8. Repeat Step 6 for next top-most signal.

9. You will see a plot similar to one below, Figure F-8.
Viewing Star-Hspice AC Analysis Waveforms

1. Close the transient waveforms. From the Signal Manager dialog box, select bjtdiff(1) and click on Close Plotfiles. This closes all transient plots.

2. In the Signal Manager, click on Open Plotfiles.

3. In the Open Plotfiles dialog box, in the Files of Type fields, select the Hspice AC (*.ac*) item.

4. You will see bjtdiff.ac0 in the menu. Click on it and click Open. The bjtdiff Plot File windows are displayed.

5. While holding down the Ctrl key, select the dim2(mag) and dim3(mag) signals.
6. Click on **Plot** from the bjtdiff Plot File window. You will see two cascaded plots. To see two signals in one plot, right-click on signal dim2(mag) to get a **Signal Menu**.

7. From the **Signal Menu**, select **Stack Region > Analog 0**. You will see a plot similar to the one below, Figure F-9.

**Figure F-9: AC Analysis Result: Plot of dim2(mag) and dim3(mag)**
*from bjtdiff.ac0*

---

**Viewing Star-Hspice DC Analysis Waveforms**

1. Close the AC waveforms. From the Signal Manager dialog box, select bjtdiff(1) and click on **Close Plotfiles**. This closes all AC plots.

2. In the Signal Manager, click on **Open Plotfiles**.

3. In the Open Plotfiles dialog box, in the Files of Type fields, select the Hspice DC (*.sw*) item.
4. You will see bjtdiff.sw0 in the menu. Click on it and click Open. The bjtdiff Plot File windows are displayed.

5. While holding down the Ctrl key, select all signals.

6. Click on Plot from the bjtdiff Plot File window. You will see four cascaded plots. To see four signals in one plot, right-click on the name of the top-most signal to get a Signal Menu.

7. From the Signal Menu, select Stack Region > Analog 0.

8. Repeat Steps 6 and 7 for next two top-most signals. You will see a plot similar to one below, Figure F-10.

Figure F-10: DC Analysis Result: Plot of v(14), v(15), v(4), and v(5) from bjtdiff.sw0

The Cosmos-Scope User’s and Reference Manual includes a full tutorial, information on the various Scope tools, and reference information on the
Measure tool. For more information, please see the Scope Manual or visit the Avant! website at:

http://www.avanticorp.com
Index — All Volumes

Symbols

!GND node 3-17
$ comment delimiter 3-9
$installdir installation directory 3-50
* comment delimiter 3-9

A

A model parameter 14-19
A2D
  function 5-50, 26-4
  model parameter 5-50
  output model parameters 5-54
  See also mixed mode
.a2d file 3-55, 3-57, 5-50
AB model parameter 15-35
ABS element parameter 26-12, 26-18, 26-23, 26-28
abs(x) function 7-9
ABSH option 9-20, 9-38, 10-36, 11-15, 12-7
ABSI option 9-20, 10-35, 10-37
  KCLTEST setting 9-21, 10-26
ABSMOS option 9-20, 10-35, 10-37
  KCLTEST setting 9-21, 10-26
absolute
  power function 7-9
  value function 7-9
  value parameter 26-12, 26-18, 26-23, 26-28
ABSTOL option 9-21, 10-22
ABSV option 9-38, 10-35, 11-15
ABSVAR option 9-43, 9-46, 11-15, 11-18, 11-27
ABSVDC option 9-21, 10-38
AC

analysis 8-3
  circuit and pole/zero models 29-47
  distortion 12-9
  MOSFETs 20-24
  noise 12-11
  RC network 2-2
  control options 12-7
  magnitude calculation 9-5, 9-39, 12-7
  network analysis 12-14
  optimization 12-4
  output
    calculation method 9-5, 9-39, 12-7
    variables analysis 8-29
    phase calculation 9-5, 9-39, 12-7
    resistance 12-3
    parameter 26-74
    transmission lines C-19
    small signal analysis 9-38
    flow 12-2
    sources 5-5
.AC statement 13-5, 13-39
  external data 3-22
  keywords 12-5
  parameters 12-5
  uses 12-4
.ac# file 3-55, 3-57
accessing customer support vi
ACCT option 8-16, 9-5
ACCURACY
  element parameter 29-10
  accuracy
    control options 10-36
    simulation time 10-35
    tolerance 10-34
ACCURATE option 9-38, 11-16, 11-27
ACM 17-3
  model parameter 11-27, 17-3, 20-27
  MOS diode 20-31, 20-34, 20-37, 20-41
  parameter 20-9, 20-26
  equations 17-22
acos(x) function 7-9
ACOUT option 8-30–8-32, 9-5, 9-39, 12-7
adder
  circuit 31-3
  demo 31-3
  NAND gate binary 31-4
  subcircuit 31-3
admittance
  AC input 8-34
  AC output 8-34
  Y parameters 8-29
AF model parameter 12-13
AGAUSS keyword 13-16
ALFA, .FFT parameter 28-8
algebraic
  equations, example 24-11
  expressions 7-8
  models 11-27
algorithms
  Damped Pseudo Transient algorithm 10-48
  DVDT 9-43, 9-48, 11-13, 11-15, 11-33, 11-34
  GEAR 9-48, 11-13, 11-29
  integration 11-29
  iteration count 11-33
  Levenberg-Marquardt 13-53
local
  truncation error 9-41, 9-48, 11-13, 11-19,
    11-33, 11-34
  timestep 9-40, 11-18
Muller 27-3
pivoting 9-24, 9-26, 10-27
timestep control 9-44, 11-12, 11-32, 11-33, 11-35
transient analysis timestep 9-48, 11-13
TRAP 9-48, 11-13
trapezoidal integration 9-49, 11-14, 11-29
ALL keyword 10-6, 10-13
ALPHA model parameter 14-19

.ALTERT
  blocks 3-41–3-42
  statement 3-40
  failure 8-18
  limit 8-18
  multiple .ALTER’s 3-42
  PCI modeling 25-44
  with .DEL LIB 3-42
alternate saturation
  model parameters
    LEVELs 6, 7 21-57
AM
  behavioral 26-62
  modulation frequency 28-13
  source function 5-22, 5-22
AMD models 20-5, 20-6
AMI gate capacitance model 20-94
AMP model parameter 26-66
amplifiers, pole/zero analysis 27-12, 27-14
analog
  behavioral
    elements 26-53
    modeling 26-4
    circuit simulation of a digital system 25-4
    device models D-5
Analog Artist interface 9-11
  output data format 9-12
  See also Artist
Analysis
  FAQ A-2
analysis
  AC 8-3
  accuracy 10-34–10-35
  circuit model 29-47
  data driven 13-2, 13-3, 24-9
  DC 8-3
  distortion 12-9
  element template 8-4
  FFT 11-44
  example
    AM modulation 28-13
    modulator/demodulator 28-16
    test circuit 28-24
    windows 28-3
    Fourier 11-37
initialization 10-3
inverter 2-7
.MEASURE statement 8-4
Monte Carlo 13-3, 13-14, 13-14–13-34, 18-52
MOSFETs
   AC 20-24
      noise 20-25
      transient 20-23
network 12-14
noise 16-58
optimization 13-39
parametric 8-4
pole/zero 10-21, 27-1
   example
      active low-pass filter 27-15
      CMOS differential amplifier 27-12
      high-pass Butterworth filter 27-10
      Kerwin’s circuit 27-8
      low-pass filter 27-5
      simple amplifier 27-14
      model 29-47
      overview 27-2
      using Muller method 27-3
pulse width 30-16
RC network
   AC 2-2
   transient 2-5
setup time 30-10
spectrum 28-1
statistical 13-8–13-34
Taguchi 13-2
temperature 13-2, 13-5
timing 30-1
transfer function (.TF) 26-46
transient 8-3, 11-3
   worst case 13-2, 13-8–13-34
   yield 13-2
arccos(x) function 7-9
arcsin(x) function 7-9
arctan(x) function 7-9
area
   calculation method See ACM
   JFETs and MESFETs
      equations 17-22–17-24
      units 17-22, 17-24
   AREA capacitor parameter 15-6
   arithmetic operators 7-9
   ARTIST option 9-11, 9-12
   ASCII output data 9-11, 9-12
   ASIC
      device libraries 3-51
      vendor libraries 3-51
   asin(x) function 7-9
   ASPEC
      AMI model 20-5
      compatibility 9-13, 20-3, 20-37, 21-86
      option 9-13, 20-11
   asterisk comment delimiter 3-9
   asymptotic waveform evaluation 29-2
      transfer function 29-35
   AT keyword 8-42
   AT1 model parameter C-36
   atan(x) function 7-9
   ATEM characterization system 3-50
   ATLEN model parameter C-36
   AUNIF keyword 13-16
   AURORA user’s group vi
   autoconvergence 10-42
      algorithm 9-31, 10-39
      disabling 9-31, 10-39
   automatic model selection
      failure D-3
      multisweep or .TEMP effect D-4
   See also model selection
   AUTOSTOP option 9-41, 11-20, 11-25, 24-9, 26-74
   AV model parameter 26-67
   AV1K model parameter 26-67
   AvanLink
      Cadence products B-2
      DA
         design flow B-10
         environment B-9
      Netlister B-12
      schematic B-10
      simulation B-12
      design flow B-4
      environment B-3
      library operations B-5
      Mentor Graphics products B-8
netlist B-6
schematic B-5
simulation B-7
Avant! web site vi
AvanWaves waveform display  B-7, B-12
AVD model parameter 26-67
average deviation 13-3
average value, measuring 8-46
AVG keyword 8-47
AWE. See asymptotic waveform evaluation

B

B# node name in CSOS 3-19
backslash continuation character 3-3, 7-8
double 7-8
in input files 3-3
BADCHR option 3-3, 9-15
bandwidth C-79
BART FFT analysis keyword 28-8
Bartlett FFT analysis window 28-4, 28-5, 28-26
base
charge equations 16-32
collector capacitance 16-36
resistance equations 16-32
base-collector junction 16-5
base-emitter
   capacitance equations 16-34
   junction 16-5
basic model parameters
   BJTs 16-6
   JFETs TOM model 17-46
   MOSFETs
   LEVEL 1 21-2
   LEVEL 13 21-107
   LEVEL 2 21-7-21-8
   LEVEL 27 21-138
   LEVEL 28 21-152
   LEVEL 3 21-19
   LEVEL 38 21-164
   LEVEL 39 21-179
   LEVEL 40 21-207
   LEVEL 47 22-2
   LEVEL 49, 53 22-34
   LEVEL 5 21-34
   LEVEL 50 22-56
   LEVEL 57 22-117
   LEVEL 58 22-134
   LEVEL 8 21-89
   LEVELs 6, 7 21-52-21-54
behavioral
   741 op-amp 26-76
   amplitude modulator 26-62
   AND and NAND gates 26-31
   BJTs
      modeling 26-92
      phase detector model 26-97
   CMOS inverter 26-46
   comparator 26-79
   components 26-40
   current source 5-34, 26-16
   data sampler 26-63
   device models D-8
   differentiator 26-55
   D-Latch 26-32
   double-edge triggered flip-flop 26-36
   elements
      analog 26-53
      using 26-3
   flip-flop 26-36
   gates 26-31
   integrator 26-53
   LC oscillator 26-83
   look-up tables 26-40
   n-channel MOSFETs 26-34, 26-42
   p-channel MOSFETs 26-34
   phase detector model 26-88
   phased locked loop 26-88
   ring oscillator 26-50
   silicon controlled rectifier 26-59
   transformer 26-57
   triode vacuum tube 26-60
   tunnel diode 26-58
   VCO model 26-81
   voltage source 5-29, 26-11
   VVCAP model 26-84
Berkeley
   BSIM3-SOI model 22-114
   junction model 22-24
NonQuasi-Static (NQS) model
beta degradation

BETA keyword

Biaschk

BiCMOS
  circuits
  devices

bidirectional circuits, wire RC model

binary
  output data
  search

Bipolar Junction Transistors. See BJTs
bisect
  data, printing
  error tolerance
  function
    syntax
    methodology
    overview
    pass-fail method
    requirements
    violation analysis

BISECTION model parameter

BJTs
  base
    charge equations
    push-out effects
    width modulation model parameters
  base-collector
    depletion capacitance equations
    diffusion capacitance equations
  base-emitter
    depletion capacitance equations
    diffusion capacitance equations
  beta
degradation
  temperature equations
  capacitance temperature equations
  conductance
  current
    equations
    flow
  DC
    models
      equations

parameters

element template listings

elements, names

energy gap temperature equations

equations

equivalent circuits

excess phase equation

EXPLI

geometric

high current Beta degradation parameters

junction capacitance
  equations
  model parameters
  junction capacitor

LEVEL
  model parameters
  temperature equations

LEVEL, model parameters

LEVEL HiCUM
  parameters

LEVEL HiCUM

low current Beta degradation parameters

.MODEL statement

models
  constants (table)
  convergence
  names
  parameters
  quasi-saturation
  statement
  transistor D
  variables (table)

noise

equations

model parameters

summary printout

npn identifier

parasitic capacitance

parasitics
  capacitance model parameters
  resistance model parameters
  resistor temperature equations

pnp identifier

power dissipation

quasi-saturation model

resistor
saturation temperature equations 16-42–16-46
scaling 16-20
S-parameters, optimization 13-60
subcircuits, scaled 16-55
substrate
  capacitance equations 16-39
  current equation 16-31
temperature
  capacitance equations 16-46
  compensation equations 16-42
  effect parameters 16-14
  parasitic resistor 16-49
  saturation equations 16-44
  transit time 16-6
  model parameters 16-12
  variable base resistance equations 16-32
BKPSIZ option 9-41, 11-20
BLACK FFT analysis keyword 28-8
Blackman FFT analysis window 28-4, 28-27
Blackman-Harris FFT analysis window 28-4, 28-28
bond wire example 31-12
branch current
  error 9-20, 9-21, 10-22, 10-37
  output 8-22
branch matrix C-33
breakpoint table
  reducing size 11-36
  size 9-41, 11-20
BRIEF
  keyword 10-6
  option 9-2, 9-6, 9-7, 9-8, 9-9
Broyden update data, printing 9-8
BSIM model 21-101
  equations 21-112
  LEVEL 13 3-31, 20-5
  example 21-121
  VERSION parameter effects 21-111
BSIM2 LEVEL 39 model 3-31
BSIM2 model 21-179
  equations 21-186
  LEVEL 39 20-6
  VERSION parameter effects 21-193
BSIM3 model
  equations 22-10
  Leff/Weff 22-9
  LEVEL 47 20-6
  SOI FD 22-147
BSIM3 SOI FD
  parameters 22-149
  template output 22-157
BSIM3 Version 2 MOS model 22-2
BSIM3v3 model
  MOS 22-19
  with Star-Hspice 22-32
BSIM4 model 22-66
  parameters 22-67
buffer 4-27, 19-6
  differential pins 19-36
ECL
  input 19-19
  input/output 19-23
  output 19-20
  tristate 19-21
IBIS 19-1
  input 19-6, 19-7
  example 19-42
  input/output 19-16
  open drain 19-18
  open sink 19-18
  open source 19-19
  syntax 19-15
  open
drain 19-18
sink 19-18
source 19-18
output 19-8, 19-9
  example 19-42
  scaling strength 19-38
  tristate 19-11, 19-12
bulk
  charge effect 20-3
  semiconductor devices 17-2
  transconductance, MOSFETs 20-23
BULK wire model parameters 14-4
Burr-Brown devices D-9
bus notation 5-68
BUS wire RC model 14-2
Butterworth filter pole/zero analysis 27-10
BV diode model parameter 15-4
BW model parameter 26-69
BYPASS option 9-42, 11-11, 20-11, 20-13
BYTOL option 9-42, 11-16

C

C2 model parameter 26-68
Cadence
   Analog Artist
      See Artist, Analog Artist
      with AvanLink B-2
   Composer, with AvanLink B-2
   Opus 9-11
   WSF format 9-11

capacitance
   base collector 16-36
   CAPOP model selector 20-7
   charge tolerance, setting 9-39, 11-16
   circuit C-33
   control options 20-68
   CSHUNT node-to-ground 9-39, 11-12
   DCAP 15-5, 16-2
   DCCAP 15-5, 16-2
   diode, Fowler-Nordheim 15-49
   distribution for wire RC model 14-2
   effective 14-10
   element parameter 4-5
   equations
      BJTs 16-34, 16-35, 16-36, 16-37
diode, Fowler-Nordheim 15-49
effective 14-10
   equations
      BJTs 16-34, 16-35, 16-36, 16-37
deployment 15-26
diffusion 15-26
diode 15-25
metal and poly 15-27
MOSFETs 20-46
GMIN 16-2
GMINDC 16-2
equations 20-66
input-output ratio 14-2
JFETs 17-6
JFETs and MESFETs 17-6, 17-25–17-30
capacitance 26-84
gate capacitance 15-12
gate to drain 17-5
model 14-9
selection 20-57
MOSFETs
   AC gate 20-97
   BSIM model 21-117
diodes 20-46
   equations 20-106
diode 15-26
   equations 20-74–20-98
gate capacitance 20-66
   example 20-64
   length/width 20-97
   models 20-57
   SPICE Meyer 20-74
   Meyer model 20-57, 20-71
   models 20-57
   multiconductor C-32
   overlap 20-73
   parameters 14-9
   BJTs 16-11
   junction 15-12
   Meyer 20-71
   MOSFETs
   Cypress 21-169
   levels 21-36
   LEVEL 38 21-169
   LEVEL 49 and 53 22-39
   LEVEL 5 21-36
   LEVEL 57 22-124
   parasitic 14-2, 16-12
   pins 24-6
   marking 20-66
   scale factor, setting 9-35, 10-31
   substrate 16-38
   table of values 9-28, 10-22
   temperature 14-11
   equations 14-11
   BJTs 16-46
   voltage variable 26-84
   wire, equations 14-6
   capacitance-voltage plots, generating 9-28, 10-23
   capacitor
      BJT 16-6
conductance requirement 10-47
current flow 8-23
DC sweep evaluation 16-34
device
equations 14-10
model 14-9
element 4-4, 14-9
template listings 8-55
equation selector option 15-5
equations 17-6
models 3-30
gate 20-57
list 20-7
name 4-5
SPICE 17-5
parameters
ejunction 16-11
metal and poly 15-13
switched 29-48
temperature, equations 14-8
transcapacitance 20-59
voltage controlled 5-35, 5-40
CAPL model parameter C-36
XPART 20-72
XQC 20-72
CAPTAB option 9-28, 10-22
cascoding, example 20-55
CASMOS
GEC model 20-5
GTE model 20-5
Rutherford model 20-6
CBD model parameter 20-27
CBS model parameter 20-27
CCCS
element parameter 26-28
syntax 5-46
CCVS 5-42
element parameter 26-23
syntax 5-42
CDB model parameter 20-28
cell characterization 13-2, 24-9
CENDIF optimization parameter 13-41
CEXT model parameter C-15, C-24, C-26, C-28
channel length modulation, MOSFETs
equations
LEVEL 2 21-15
LEVEL 3 21-26
LEVEL 38 21-175
LEVEL 5 21-48
LEVEL 6 21-80
LEVEL 8 21-98
parameters
LEVEL 6 21-80
LEVEL 8 21-93
characteristic impedance C-73
characterization of models 10-14
charge conservation 20-91
capacitance 20-87
charge tolerance, setting 9-39, 11-16
CHGTOL option 9-39, 11-16, 11-34
circles, defining 18-39
circuits
adder 31-3
BiCMOS 16-2
BJT 16-20, 16-21
design
nonplanar and planar technologies 20-15
wave processes 20-15
ECL 16-2
inverter, MOS 2-7
model 29-43
AC analysis 29-47
transient responses 29-46
nonconvergent 10-51
RC
line 29-33
network 2-2
reusable 3-47
simulating
with Signetics drivers 25-18
with Xilinx FPGAs 25-22
subcircuit numbers 3-18
temperature 13-5, 13-6
test, FFT analysis 28-24
TTL 16-2
See also subcircuits
CJ model parameter 20-28

I-8 Star-Hspice Manual, Release 2000.4
CJA model parameter 20-28
CJBR model parameter 15-36
CJGATE model parameter 20-28
CJGR model parameter 15-36
CJP model parameter 20-28
CJSR model parameter 15-36
CJSW model parameter 20-28
CLEN model parameter C-36
CLOAD model parameter 5-54
clock
  frequency C-79
  skew 25-6
CLOSE optimization parameter 13-41
CMI
  function calling protocol 23-36
  models, simulations 23-4
  supported platforms 23-5
  testing 23-12
  variables 23-18
CMI_AssignInstanceParm 23-23
CMI_AssignModelParm 23-22
CMI_Conclude 23-35
CMI_DiodeEval 23-27
CMI_Evaluate 23-26
CMI_FreeInstance 23-32
CMI_FreeModel 23-32
CMI_Noise 23-29
CMI_PrintModel 23-31
CMI_ResetInstance 23-22
CMI_ResetModel 23-20
CMI_SetupInstance 23-25
CMI_SetupModel 23-24
CMI_Start 23-35
CMI_WriteError 23-33
CMOS
  differential amplifier, pole/zero analysis 27-12
  output driver demo 31-12
  tristate buffer, optimization 13-55
CMRR
  model parameter 26-68
  specification 26-5
CMULT model parameter C-15, C-24, C-26, C-28
CO option 8-14, 9-6

coax
  models C-57
  See also transmission lines
coefficients
  Laplace 29-23
  transfer function 29-21
  collector-substrate junction 16-5
  column laminated data 3-28
  comlinear device models D-10
commands
  Hspice 3-58–3-62
    arguments 3-59
    examples 3-62
    options 3-60
  limit descriptors 8-18
  output 8-2
  comment line (digital vector files) 5-62
  comments 3-9
common
  emitter gain 31-20
  model interface (CMI) 23-1
Common Simulation Data Format 9-11
COMP model parameter 26-66, 26-68
comparators
  behavioral models 26-65
  model 26-79
complex poles and zeros 29-30
compression of input files 3-2
computer platforms for HSpice 1-6
concatenated data files 3-26
conductance
  BJTs 16-21, 16-22
  current source, initialization 9-33, 10-40
  diodes 15-17
  for capacitors 10-47
  GMIN 15-5
  GMINDC 15-5
  JFETs 17-6
  JFETs and MESFETs 17-6, 17-9
  matrices 18-4
  minimum, setting 9-39, 11-21
  models 9-31, 10-24
  MOSFETs 20-22, 20-54
  nodes 9-33, 10-40

Star-Hspice Manual, Release 2000.4
multiconductor C-32
negative, logging 9-15
node-to-ground 9-34, 9-39, 10-25, 11-12
output, negative, preventing 21-121, 21-193

pn junction 10-54
scale, setting 9-35, 10-31
sweeping 9-33, 10-24

configuration file 3-54
continuation character, parameter strings 7-8
continuation of line (digital vector file) 5-62
control characters in input 3-3
control options 17-6
accuracy 9-20, 10-36
AC 9-38, 9-38–9-41
defaults 11-35
algorithm 9-47–9-49
algorithm selection 10-22
analysis 9-13–9-15
ASPEC 20-11
BYPASS 20-11
capacitance 20-68
convergence 9-29–9-35, 10-36, 15-5
DC 10-22
DC convergence 10-22
DC operating point analysis 10-22
defaults 9-2
error 9-15
FAQ A-27
initialization 10-22
input and output 9-5–9-10, 9-28, 9-50
interface options 9-11–9-13
keyword table 9-3
limit 11-20
matrix-related 9-23–9-27
method 11-11
pole/zero 9-35–9-37, 27-4
analysis 10-31–10-33
printing 8-14
values of 9-9
setting 9-2, 15-4
speed 9-41–9-43
table 9-5
timestep 9-43–9-47
tolerance 11-15
transient analysis

limit 11-20–11-23
method 11-11–11-14
tolerance 11-15–11-20

version 9-15
controlled sources 5-24, 26-4, 26-6
element statement 5-24, 26-7
conventions
bias polarity 23-42
source-drain reversal conventions 23-43
CONVERGE option 9-29, 9-32, 10-38, 10-42, 10-48

convergence
BJT model 16-3
control options 10-36
ensuring 10-23
for optimization 13-43
increasing iterations 10-23
JFETs and MESFETs 17-6
MOSFETs diodes 20-26
problems 10-49
analyzing 10-49
autoconverge process 10-42
causes 10-51
by BYPASS 9-42, 11-11
changing integration algorithm 9-49, 11-14
CONVERGE option 10-48
solutions 9-29, 9-32, 10-38
DCON setting 9-30, 10-39, 10-44
decreasing the timestep 9-44, 11-21
diagnosing 10-49–10-54
diagnostic tables 10-49
diodes 15-5
floating point overflow 10-48
GMINDC ramping 10-44
internal timestep too small 9-48, 11-13
.NODESET statement 10-10
nonconvergent node listing 9-31, 10-39
op-amp models 26-66
operating point Debug mode 10-7
pole/zero analysis 27-3
reducing 10-45
setting DCON 9-31, 10-39
steady state 9-33, 10-24

core
Jiles-Atherton 14-12
ferromagnetic 14-19
model parameters 14-16
magnetic 14-12
element output 14-16
model parameters 14-14
model parameters 14-16
A 14-19
ALPHA 14-19
MS 14-19
CORKD model parameter C-15, C-25
cos(x) function 7-9
cosh(x) function 7-9
coupled line noise simulation example 25-27
CPTIME option 9-11
CPU time
limiting 9-11
reducing 9-8
setting maximum 9-11
critical frequency 29-5
CROSS keyword 8-41
crossstalk C-78
CSB model parameter 20-28
CSCAL option 9-35, 10-31, 27-4
CSDF option 9-11
CSHDC option 9-29, 10-23
CSHUNT option 9-39, 11-12
CUR element parameter 26-18
current
ABSMOS floor value for convergence 9-22, 10-
branch 8-23
current error 9-20, 10-22, 10-37
controlled
current sources 5-46, 26-6, 26-7, 26-27
element template listings 8-57
elements 26-6
sources 5-24
voltage sources 5-42, 26-6, 26-23
element template listings 8-57
convention
BJTs 16-20
JFETs 17-7
MOSFETs 20-21
diodes 15-16
epitaxial 16-51
in HSPICE elements 8-23–??
operating point table 10-6
output 8-21
element branches 8-22
sources 5-34
CURRENT keyword 10-6
Curtice model 17-3, 17-21, 17-22
customer support vi
CUT optimization parameter 13-42
C-V plots 31-7
generating 9-28, 10-23
CVTOL option 9-18
Cypress model 20-5, 21-162

D

D12 model parameter C-28
D2A
function 5-50, 26-4
input model parameters 5-51
model parameter 5-50
See also mixed mode
d2a file 5-50
Dallas Semiconductor model 20-5
Damped Pseudo Transient algorithm 10-48
data
driven
analysis 13-2, 13-3, 24-9
PWL source function
PV parameter 5-19
TIME parameter 5-19
encryption E-1
files, disabling printout 9-6, 9-8
flow, overview 1-9
sampler, behavioral 26-63
sheet parameters 24-1, 24-2
DATA keyword 3-21, 10-15, 11-5, 12-5
.DATA statement 3-21, 24-9
data-driven analysis 3-21
datanames 3-22
external file 3-25, 3-27
for sweep data 3-22

Star-Hspice Manual, Release 2000.4
inline data 3-23
inner sweep example 3-24
outer sweep 3-25
outer sweep example 3-25
sweep 3-25
datanames 3-22
db(x) function 7-10

DC
analysis 8-3, 10-19–10-21
capacitor conductances 10-47
decade variation 10-16
initialization 9-30, 10-22, 10-23
iteration limit 9-23, 10-25
linear variation 10-16
list of points 10-16
octave variation 10-16
sensitivity 10-19
transfer function 10-20

BJT 16-6
convergence control options 10-22
current 20-45
equations
BJTs 16-29
JFETs and MESFETs 17-31–17-34
ers, reducing 10-45
operating point
analysis 10-6
bypassing 11-3
initial conditions file 3-54
See also operating point
optimization 10-15
parameters
JFETs
LEVEL 1 17-19
LEVEL 2 17-19–17-20
LEVEL 3 17-21–17-22
MOSFETs 20-27
LEVEL 57 22-118
sensitivity analysis 10-19
small-signal analysis 10-20
sources 5-5
sweep 10-14
transfer function 10-20
.DC statement 10-14, 13-5, 13-39
external data with .DATA 3-22
DCAP 15-5, 16-2
equation selector 16-3
option 15-5
JFETs and MESFETs
capacitance 17-6
models 17-6
overriding in BJTs models 16-3
DCCAP 15-5, 16-2
option 9-28, 10-23, 15-5, 16-34, 31-7
JFETs and MESFETs capacitance 17-6
DCFOR option 9-29, 10-23
DCHOLD option 9-30, 10-23
DCON option 9-30, 10-39, 10-42
DCSTEP option 9-31, 10-24, 10-47
DCTRAN option 9-32, 10-39
.DCVOLT statement 10-9
DDL 3-50, 26-5
DDLPATH variable 3-51
models 31-20
DDLPATH environment variable 3-51, 31-20
DEBUG keyword 10-7
DEC keyword 10-16, 11-6, 12-6
decibel function 7-10
decoupling methods 25-6
DEF model parameter 26-68
DEFAD option 9-18, 20-11
DEFAS option 9-18, 20-11
DEFAULT_INCLUDE variable 3-54
DEFI option 9-18, 20-11
DEFNRD option 9-18, 20-11
DEFNRS option 9-18, 20-11
DEFPD option 9-18, 20-11
DEFPS option 9-18
DEFW option 7-16, 9-18, 20-11
.DEL LIB statement 3-6
in .ALTER blocks 3-41
with .ALTER 3-42
with .LIB 3-42
with multiple .ALTER statements 3-42
DELAY
element parameter 26-12, 26-18, 26-24, 26-28
model parameter C-36
delays

Star-Hspice Manual, Release 2000.4
causes 25-3
derating curve 24-4
element example 5-40
Elmore 29-33
group 8-33, 9-51, 12-8
measuring 8-39
plotting 24-5
problems and solutions 25-3
simulation example 24-2, 24-9
time (TD) 8-33
transmission lines C-37
types 25-6
DELEN model parameter C-36
DELF element parameter 29-6, 29-10
DELMAX option 9-43, 9-46, 11-20, 11-23, 11-26, 11-35, 11-38
ideal delay elements 11-26
oscillator circuits 11-26
DELPHS 26-69
DELTA
element parameter 26-10, 26-12, 26-18, 26-24, 26-26, 26-58, 29-10
internal timestep 9-43, 11-20
See also timestep
DELVTO model parameter 13-9
demo files
2n2222 BJTs transistor characterization 31-32
2n3330 JFETs transistor characterization 31-32
A2D 31-29, 31-30
AC
analysis 31-26
resonance analysis 31-35
acl gate 31-27
A/D flash converter 31-30
adders
72-transistor two-bit 31-28
BJT NAND gate two-bit 31-27
BJT two-bit 31-26
D2A 31-29
MOS two-bit 31-27
NAND gate four-bit binary 31-26
air core transformer 31-35
algebraic
output variables 31-26
parameters 31-26
transmission lines 31-39
.ALTERR statement 31-26
AM source 31-38
amplifier 31-29
amplitude modulator 31-27
analog 31-28
AND gate 31-27
automatic model selection program 31-36
behavioral applications 31-27–31-28
behavioral models 31-28
diode 31-27
D-latch 31-27
filter 31-26
NAND gate 31-27
ring oscillator 31-28
triode 31-28
voltage to frequency converter 31-26
benchmarks 31-28
bisection
pass-fail 31-28
search 31-28, 31-29
BJTs
analog circuit 31-28
beta plot 31-29
differential amplifier 31-26, 31-29
diodes 31-29
ft plot 31-29
gm, gpi plots 31-29
photocurrent 31-38
Schmidt trigger 31-26
sense amplifier 31-26
BSIM3 model, LEVEL=47 31-35
capacitances, MOS models
LEVEL=13 31-35
LEVEL=2 31-35
LEVEL=6 31-36
cell characterization 31-26, 31-27, 31-29
charge conservation, MOS models
LEVEL=3 31-36
LEVEL=6 31-36
circuit optimization 31-29–31-30
CMOS
differential amplifier 31-26
input buffer 31-29
inverter macro 31-28
I/O driver ground bounce 31-26, 31-38
output buffer 31-29
coax transmission line 31-39
crystal oscillator 31-26
current controlled
current source 31-28
voltage source 31-28
D2A 31-29
DC analysis, MOS model LEVEL=34 31-36
DDL 31-30–31-32
delay 31-26, 31-29
versus fanout 31-29
device optimization 31-32–31-33
differential amplifier 31-26
differentiator 31-27
diffusion effects 31-27
diode photocurrent 31-37
D-latch 31-27
E Element 31-27
edge triggered flip-flop 31-27
exponential source 31-38
FFT
AM source 31-33
analysis 31-33–31-34
Bartlett window 31-33
Blackman window 31-33
Blackman-Harris window 31-34
data-driven transient analysis 31-34
exponential source 31-33
Gaussian window 31-34
Hamming window 31-34
Hanning window 31-34
harmonic distortion 31-33
high frequency detection 31-33
intermodulation distortion 31-34
Kaiser window 31-34
modulated pulse source 31-34
Monte Carlo, Gaussian distribution 31-34
product of waveforms 31-34
pulse source 31-34
PWL 31-34
rectangular window 31-34
single-frequency FM source 31-34
sinusoidal source 31-34
small-signal distortion 31-33
switched capacitor 31-34
transient 31-34
sweep 31-34
window tests 31-34
filter matching 31-30
filters 31-34–31-35
behavioral 31-26
fifth-order
elliptical switched capacitor 31-28
low-pass 31-35
fourth-order Butterworth 31-35
Kerwin’s circuit 31-35
LCR bandpass 31-35
matching lossy to ideal 31-30
ninth-order low-pass 31-27, 31-35
switched capacitor low-pass 31-27
FR-4 microstrip transmission line 31-35, 31-38
G Element 31-26, 31-27
GaAsFET amplifier 31-26
gamma model LEVEL=6 31-36
general applications 31-26–31-27
ground bounce 31-26, 31-38
group time delay 31-26
impact ionization plot 31-36
input 31-26
installation test 31-28
integrator 31-27
inverter 31-27, 31-28
characterization 31-26, 31-29
sweep 31-29
IRF340 NMOS transistor characterization 31-32
I-V
and C-V plots, LEVEL=3 31-36
plots
MOSFETs model LEVEL=13 31-36
SOSFET’s model LEVEL=27 31-36
JFETs photocurrent 31-38
junction tunnel diode 31-29
LCR circuit 31-29
lumped
MOS model 31-26
transmission lines 31-35, 31-39
magnetic core transformer 31-35
magnetics 31-35
microstrip transmission lines 31-35, 31-39
coupled 31-39
optimization 31-39
series 31-39
Monte Carlo
    analysis 31-27
    DC 31-27
    Gaussian distribution 31-27
    limit function 31-27
    uniform distribution 31-27
MOS
    amplifier 31-29
    simulation 31-28
    MOSFETs 31-35–31-36
    sigma sweep 31-29
    sweep 31-26
    NAND gate 31-27, 31-28
    NMOS E-mode model, LEVEL=8 31-38
    noise analysis 31-26
    op-amp 31-26, 31-27
    characterization 31-30–31-32
    voltage follower 31-27, 31-38
    optimization 31-27
    2n3947 Gummel model 31-33
    DC 31-33
    diode
        I-V and C-V 31-33
        temperature 31-33
    GaAs
        FET 31-33
        s-parameter 31-33
        JFETs 31-33
    group delay 31-29
    Hfe 31-33
    I-V 31-33
    JFETs 31-33
    LEVEL=2 model beta 31-32
    LEVEL=2 31-33
    MOS 31-33
        LEVEL=13 I-V 31-33
        LEVEL=2 I-V 31-33
        LEVEL=3 I-V 31-33
        LEVEL=6 I-V 31-33
    s-parameter 31-33
    speed, power, area 31-29
    width 31-29
    parameters 31-26
PCI
    Monte Carlo analysis 31-36
    worst-case modeling 31-36
    peripheral component interconnect
    See PCI
    phase
        detector 31-27
        locked loop 31-27
        photocurrent 31-36–31-38
        GaAs device 31-38
        photolithographic effects 31-27
        piecewise linear source
            See demo files, PWL
        pll 31-27
        pole/zero analysis 31-26, 31-35
        pulse source 31-38
        PWL 31-38
            CCCS 31-28
            CCVS 31-28
            switch element 31-28
            VCCS 31-27, 31-28
            VCO 31-28
            VCVS 31-28
        radiation effects 31-36–31-38
            bipolar devices 31-36–31-37
            DC I-V, JFETs 31-38
            GaAs differential amplifier 31-38
            JFETs devices 31-37
            MOSFETs devices 31-37
            NMOS
                NAND 31-38
                voltage divider 31-38
        RC circuit optimization 31-30
        resistor temperature coefficients 31-30
        RG58/AU coax test 31-35
        ring oscillator 31-28
        Royer magnetic core oscillator 31-35
        Schmidt trigger 31-26
        sense amplifier 31-26
        series source coupled transmission lines 31-39
        setup
            characterization 31-29
            time search 31-28, 31-29
            shunt terminated transmission lines 31-39
            silicon controlled rectifier 31-28
            sine wave sampling 31-27, 31-28
            single-frequency FM source 31-38
            sinusoidal source 31-38
            skew models 31-27
SNAP to HSPICE conversion 31-29
sources 31-38
s-parameters 31-29, 31-35
    LEVEL=13 31-36
sweep 31-26
switch 31-27
    characterization 31-27
switched capacitor 31-27, 31-38
RC circuit 31-28
temperature effects
    LEVEL=13 31-36  
    LEVEL=6 31-36
timing analysis 31-28
total radiation dose 31-37
transient analysis 31-26
transistor characterization 31-32
transmission lines 31-38–31-39
triode model 31-28
tunnel diodes 31-28, 31-29
twinlead transmission line model 31-39
U models 31-39
unity gain frequency 31-29
Viewsim
    A2D input 31-29
    D2A input 31-29
voltage
    controlled
        current source 31-27, 31-28
        oscillator 31-26, 31-28
        resistor inverter 31-38
        voltage source 31-28
follower 31-27
    to frequency converter 31-26
    variable capacitor 31-27
waveform smoothing 31-28
worst case skew model 31-27
depletion
capacitance
    DCAP equation selector 15-5, 16-3, 17-6
    equations 15-26
MOS devices 20-3
DERIVATIVE keyword 8-49
derivative, measuring 8-49
design
    high speed, problems and solutions 25-3
name 3-53
partitioning 25-7
stability 29-2
time, reducing 29-2
Design Architect
    with AvanLink B-8
Design Viewpoint Editor (DVE)
    with AvanLink B-8
deviation, average 13-3
device
    capacitor, equations 14-10
    characterization 3-50
    DDL 3-50
    inductor, equations 14-17
    models 14-1
DFT 28-1
DI control option 9-21, 9-39, 10-39, 11-16, 12-8
DIAGNOSTIC option 9-15
diagnostic tables 10-49–10-50
dielectric
    loss modeling 18-12
dielectric
    constant 14-7
    thickness, wire model parameters 14-5
differentiator, behavioral 26-55
diffusion
    capacitance equations 15-26
    layer process parameters, MOSFETs LEVEL 13
    21-106
DIFSIZ optimization parameters 13-42
digital
    files 5-50, 26-31
    input 5-50
    vector file 5-60
DIM2
    distortion measure 12-9
    parameter 8-35
DIM3
    distortion measure 12-9
    parameter 8-35
diodes
    barrier 15-1
    breakdown example 5-41
    capacitance 15-25
calculations 15-5
  equations 15-25, 20-46
depletion capacitance 15-26
diffusion capacitance 15-26
LEVEL 3 metal 15-27
LEVEL 3 poly 15-27
temperature 20-106
Fowler-Nordheim 15-49
CMI_DiodeEval 23-27
conductance 15-17
control options 15-4
convergence problems 15-5
current 15-16
  flow 8-24
DC parameters 17-15
  elements
    template listings 8-58
  equations 15-22–15-32
    example 5-40
    Fowler-Nordheim diodes 15-49
equivalent circuits 15-17, 15-17–15-18
EXPLI 9-19
Fowler-Nordheim 15-3, 15-48
  equations 15-49
junction 15-1
  DC equations 15-23
  equations 15-22
  geometric 15-3
  model 15-7
    capacitance parameters (table) 15-12
    DC parameters (table) 15-9
  parameters 15-8
  periphery 4-13
  temperature 15-19
  parameters 15-19
Junction Cap Model 15-33
metal model capacitance parameters (table) 15-13
  models 3-30, D-10
  define 15-16
  levels 15-4
  names 4-12
  noise parameters (table) 15-13
  regions 15-23
  statements 15-4
  using 15-1
variables (table) 15-22
MOSFETs 20-9
  capacitance equations 20-46
  equations 20-45
  model selector 20-9
  models 20-9, 20-26
  resistance temperature 20-111
noise equations 15-28
nongeometric junction 15-3
nonvolatile memory 15-1
poly model capacitance parameters (table) 15-13
polysilicon capacitor length 4-13
power dissipation 8-26
scaling
  options 15-4
  parameters 15-14–15-16
Schottky, barrier 15-1
series resistance units 15-11
temperature
  compensation equations 15-28
  effects 15-19
  equations
    breakdown voltage 15-29
    contact potential 15-30
    energy gap 15-28
    grading coefficient 15-32
    junction capacitance 15-31
    leakage current 15-29
    resistance 15-32
    transit time 15-29
types 15-3
Zener 15-1, 15-4
See also junction diodes
directories
  installation directory 3-50
tmp 1-12
DIS 26-69
Discrete Fourier Transform 28-1
.DISTO statement 12-9, 12-10
distortion
  analysis 8-35, 12-9
  measures 12-9
DLAT model parameter 20-30
DLEV model parameter C-13, C-14, C-21, C-24, C-28
DNB model parameter 20-28
document conventions iv
documentation
FAQ A-4
documentation issues iv
dollar sign ($) comment delimiter 3-9
drain current equation, MOSFETs LEVEL 47 22-15
drain-to-source current, convergence error tolerance 9-20, 10-37
DTA model parameter 15-35
DTEMP
   element parameter 13-5
   model parameter 13-5, 13-6, 31-18
DV
   calculation 9-30, 10-39
   option 9-32, 10-24, 10-42
DVDT
   algorithm 9-43, 9-46, 11-15, 11-18, 11-29, 11-33
   option 9-44, 9-48, 11-12, 11-13, 11-33, 11-34
   timestep control 11-26, 11-35
DVTR option 9-47, 11-21
DW model parameter 20-30
dynamic timestep algorithm 11-34

E

E Elements 26-6, 26-11, 29-26–29-33
   applications 26-6
   controlling voltage 26-14
   data
      points 26-13
      sampler application 26-63
   element
      name 26-12
      value multiplier 26-13
   gate type 26-12
   initial conditions 26-12
   integrator application 26-54
   Laplace transform 29-4
   maximum voltage 26-13
   minimum voltage 26-13
   NAND gate 26-31
   parameters 26-12
      value multiplier 29-11
   polynomial
   coefficients 26-13
   dimension 26-13
   SCALE parameter 29-37
   temperature coefficients 26-13
time delay keyword 26-13
   transformer application 26-57
   triode application 26-61
turns ratio 26-12
   voltage gain 26-12
E elements
   syntax statements 5-29
   Early voltage 21-5
   Ebers-Moll model 16-1, 16-5
   ECL
      circuits 16-2
devices 16-2
effective capacitance
calculation 14-10
effective channel length, MOSFETs
   equations
      LEVEL 1 21-5
      LEVEL 13 21-112
      LEVEL 28 21-156
      LEVEL 39 21-189
      LEVEL 49 22-49, 22-52
      LEVEL 5 21-38
      LEVEL 6 21-61
      LEVEL 8 21-94
parameters
      LEVEL 1 21-3
      LEVEL 2 21-8–21-9
      LEVEL 3 21-20, 21-24
      LEVEL 38 21-165
      LEVEL 5 21-35
      LEVEL 8 21-90
      LEVELs 6, 7 21-54–21-55
effective channel width, MOSFETs
   equations
      LEVEL 1 21-5
      LEVEL 13 21-112
      LEVEL 2 21-12
      LEVEL 28 21-156
      LEVEL 3 21-24
      LEVEL 39 21-189
      LEVEL 49 22-49, 22-52
LEVEL 5 21-38
LEVEL 6 21-61
parameters
LEVEL 1 21-3
LEVEL 2 21-8
LEVEL 3 21-20–21-21
LEVEL 38 21-165
LEVEL 5 21-35
LEVEL 8 21-90–21-91, 21-94
effective mobility
MOSFETs, equations
LEVEL 28 21-157
LEVEL 3 21-26
LEVEL 6 21-75
LEVEL 8 21-96
parameters, LEVEL 6 21-75
EFPS option 20-11
EISA standard 25-33
electrical measurements
simulating 31-20
ELEMENT
statements 3-50
element
active
BJTs 4-14
diodes 4-12
JFETs 4-16
MESFETs 4-16
MOSFETs 4-18
analog behavioral 26-53
capacitors 14-9
checking, suppression of 9-8
current controlled 26-6
IC parameter 10-9
ideal 5-24
independent source 5-2, 5-7
linear inductors 14-12
markers, mutual inductors 4-10
names 3-17
OFF parameter 9-35, 10-4, 10-27
parameters See element parameters 4-1
passive
capacitors 4-4
inductor 4-6
mutual inductor 4-9
resistors 4-2
statement, current output 8-22
statements 3-10, 20-4
capacitor 14-9
G Elements 26-16–26-22
linear inductor 14-12
op-amps 26-66
transconductance
Laplace 29-6
pole/zero 29-7
voltage gain
Laplace 29-6
pole/zero 29-7
subcircuits 3-14
temperature 13-6
templates 8-35–8-58
analysis 8-4
BJTs 8-59
capacitor 8-55
current-controlled
current source 8-57
voltage source 8-57
function 7-10
independent
current source 8-58
voltage source 8-57
inductor 8-55
JFETs 8-61
MOSFETs 8-62, 22-128, 22-144
mutual inductor 8-56
printout 20-62
resistor 8-55
saturable core
element 8-65
winding 8-65
voltage-controlled
current source 8-56
transmission line 4-21, 4-23, 4-25
voltage-controlled 26-6
current source 5-24
voltage source 5-24
ELEMENT parameters
.ALTER blocks 3-41
element parameters
BJTs 4-15
model names 4-15
node names 4-15
capacitors 4-5–4-6, 14-9–14-11
DELTA 26-10
diodes 15-6
AREA 15-14
M 15-14
DTEMP 13-5
E Elements 26-12–26-14
F Elements 26-28–26-29
G and E Elements 29-10–29-11
G Elements 26-18–26-21
H Elements 26-23–26-25
IBIS buffers 4-27
independent sources 5-2–5-3
data driven PWL function 5-18
PULSE function 5-7, 5-10, 5-13, 5-15
SFFM function 5-20
inductors 4-7–4-8
coefficients 4-8
JFETs and MESFETs 4-16–4-17
linear inductors 4-7
M 15-14
MOSFETs 4-18–4-20
GEO 20-42
mutual inductors, Kxxx 4-9
POLY 26-7
PWL
PV 5-19
R 5-16
TIME 5-19
range limits 22-52
resistors 4-2–4-3
DTEMP 4-3
scaling 20-12
transmission lines
T Element 4-24
U Element 4-25
W Element 4-21, 4-21–4-22
U Elements 4-20–C-51
element statements
BJT 16-20
independent sources 5-2
elements
templates
MOSFET 22-157
ELEV model parameter C-7, C-13, C-14
Elmore delay 29-33
Empirical model 21-19
equations 21-22
example 21-30
enable (digital vector file) 5-72
encrypted data E-2
encryption
file organization E-6
FREELIB keyword E-7
guidelines E-5
permit file changes E-7
semicolon bug E-5
Encryption structure example E-13
Encryption, 8-byte key E-9
.END statement
for multiple HSPICE runs 3-45
in libraries 3-34
location 3-45
missing 3-2
with .ALTER 3-42
ENDDATA keyword 3-23, 3-25, 3-27
.ENDL statement 3-32, 3-33
.ENDS statement 3-14
energy gap temperature equations
BJTs 16-42
JFETs and MESFETs 17-41
MOSFETs 20-105
environment variables
DDLPATH 3-51, 31-20
FAQ A-6
.EOM statement 3-14
EPFL-EKV MOSFETs model 22-85
epitaxial
charge 16-52
current source 16-51
EPSMIN option 9-11, 29-20
equations
ACM parameters 17-22
BJTs
DC models 16-29
noise 16-40
parasitic resistor temperature 16-49
temperature 16-42, 16-44
BSIM LEVEL 13 21-112
capacitance
depletion 15-26
diffusion 15-26
diode 15-25
effective 14-10
metal and poly 15-27
MOS diode 20-106
MOSFETs 20-46
overlap 20-73
temperature 14-11, 16-46
capacitor 17-6
device 14-9, 14-10
DC
JFETs and MESFETs 17-31
junction 15-23
device
capacitor 14-10
inductor 14-17
diodes
capacitance 15-25, 20-46
Fowler-Nordheim 15-49
junction 15-22
MOSFETs 20-45
error 8-52
evaluation 8-45
Frohman-Bentchkowski 21-77
impact ionization 20-53
inductor
device 14-12, 14-17
temperature 14-17
MOSFET models
LEVEL 61 22-178
LEVEL 62 22-186
MOSFETs
channel length modulation 20-111
diode 20-45
resistance 20-111
impact ionization 20-53
mobility temperature 20-111
model parameters 20-51
models
BSIM2 21-186
BSIM3 22-10
Leff/Weff 22-9
Cypress 21-169
Empirical 21-22
EPFL-EKV 22-92, 22-107
NQS 22-106
HP a-Si TFT 21-211
IDS
LEVEL 5 21-37, 21-43
LEVEL 6 21-60, 21-71
LEVEL 8 21-93
LEVELs 49 and 53 22-52
modified BSIM LEVEL 28 21-155
quasi-static 22-103
Schichman-Hodges 21-5, 21-11
noise 20-99
surface potential temperature 20-109
temperature 20-105
threshold voltage 20-110
voltage 20-50
noise 17-35, 20-99
BJTs 16-40
junction diode 15-28
MOSFETs LEVEL 55 22-107
Normal Field 21-78
resistor
device 14-2
model 14-5
noise 14-7
temperature 14-8
Star-Hspice ΔL 21-85
substrate current 16-31
temperature
BJTs 16-44
parasitic resistor 16-49
capacitance 14-11, 16-46
compensation 15-28, 16-42
energy gap 20-105
inductor 14-17
JFETs and MESFETs 17-38
MOS diode capacitance 17-38
MOSFETs 20-105
saturation current 20-106
transmission lines 18-3
variable
definitions 15-22
names 16-25
names and constants 17-11
voltage 20-50
Wang’s 21-84
wire
capacitance 14-6
resistance 14-5
equivalent
circuit
BJTs 16-21
JFETs 17-7, 17-8
MESFETs 17-7
MOSFETs 20-20, 20-22
AC analysis 20-24
AC noise analysis 20-25
transient analysis 20-23
variables and constants 20-20
ERR function 8-51, 8-52
ERR1
function 8-51, 8-53
keyword 13-37
ERR2 function 8-51, 8-53
ERR3 function 8-51, 8-53
errors
bulk node not specified 14-7
cannot open
input file 3-58
output spool file 8-18
control options A-27
current exceeding MAXAMP 9-39, 11-17
DC 10-45
digital file has blank first line 5-50
environment variables A-6
file open 1-13
functions 8-51-8-53
input A-12, A-32
installation A-13
internal timestep too small 9-34, 9-39, 9-46, 9-48, 10-5, 10-25, 10-53, 11-3, 11-12, 11-13, 11-23
licensing A-15
messages 19-46, A-7
missing .END statement 3-2
models A-20
negative slope for magnetization level 14-19
netlist A-27
no input data 1-12
optimization goal 8-40
parameter name conflict 8-38
PC A-23
reference to undefined inductor 14-18
special characters in input 3-3
system resource inaccessible 8-18
timestep control error in transmission lines C-61
tolerances
ABSMOS 9-20, 10-37
branch current 9-20, 9-21, 10-22, 10-37
optimization by bisection 30-8
pole/zero analysis 9-36, 10-32
relative change 9-22, 9-40, 10-41, 11-18
REL MOS 9-20, 10-37
voltage 9-22, 9-23, 9-40, 10-41, 11-18
EXA model parameter 20-28
example
AC analysis
Hspice vs. SPICE methods 8-31
RC network 2-2
capacitance, MOSFETs 20-64
cascoding 20-55
.DATA
inner sweep 3-24
outer sweep
digital vector file 5-62
experiments 1-8
.FFT statement 11-41
field solver 18-45
gate capacitance 20-64
Hspice vs. SPICE methods
AC analysis 8-31
hysteresis 14-23
input buffer 19-42
.MODELS CARDS NMOS model 22-53
Monte Carlo 13-18, 13-26
MOSFETs
BSIM LEVEL 13 21-121
data fitting 21-227
Empirical 21-30
gate capacitance 20-64
IDS LEVEL 5 21-50
IDS LEVEL 7 21-66, 21-68
network analysis, bipolar transistor 12-21
optimization 13-44
.OPTIONS SEARCH 3-36
output buffer 19-42
parameter extraction 14-24
PMOS model 22-55
S parameter 6-8
simulation PCI 25-46
subcircuit 3-15
MULTI model 3-15
subcircuit test 3-13
transient analysis
inverter 2-7
RC network 2-5
transmission line 19-42, C-8
worst case 13-26
excess phase 16-39
EXD model parameter 20-28
EXJ model parameter 20-28
EXP model parameter 20-28
EXP source function
fall time
constant 5-13
delay 5-13
initial value 5-13
pulsed value 5-13
rise time
constant 5-13
delay 5-13
exp(x) function 7-10
experiment 1-8
experimental methods
using Star-Hspice 1-8
EXPLI option 9-19
EXP MAX option 9-11
exponential
function 7-10
source function 5-13
expressions
algebraic 7-8
EXS model parameter 20-28
external data files 3-7, 3-22
EXTRAPOLATION
element parameter 29-10

F Elements 26-6, 26-27
applications 26-7
controlling voltage 26-30
current gain 26-28
data points 26-28
gate type 26-28
initial conditions 26-28
maximum current 26-28
minimum current 26-29
multiply parameter 26-28
name 26-28
parameters 26-28
polynomial
coefficients 26-29
dimension 26-29
temperature coefficients 26-29
time delay keyword 26-29
value multiplier 26-29
F elements
syntax statements 5-46
FALL keyword 8-41
fall time
example 24-9
EXP source function 5-13
simulation example 24-2
fanout, plotting 24-5
FAQ
analysis A-2
control options A-27
environment variables A-6
error messages A-7
field solver A-34
input A-12
installation A-13
known limitations A-17
licensing A-15
manuals A-4
models A-20
MS Windows/PC A-23
netlist A-27
output A-32
W Element A-34
waveform viewing A-35
Fast Fourier Transform

See FFT

FAST option 9-42, 11-17, 11-25
FC model parameter 20-28
ferromagnetic core 14-12
model 14-19

FFT

analysis

alpha control parameter 28-8
example
AM modulation 28-13
modulator/demodulator 28-16
test circuit 28-24
frequency
maximum 28-8
minimum 28-8
of interest 28-10
range 28-10, 28-11
specification 28-8
harmonic distortion 28-16
number of points 28-7
output 11-44
magnitude format 28-7
results 28-10–28-12
spectral leakage 28-16
start point 28-7
statement syntax 28-7–28-9
stop point 28-7
window type 28-8
output 28-10
windows 28-3–28-5
Bartlett 28-4
Blackman 28-4
Blackman-Harris 28-4
Gaussian 28-4
Hamming 28-4
Hanning 28-4
Kaiser-Bessel 28-4
rectangular 28-4

.FFT statement 28-1, 28-7
field effect transistor 20-16
geometry 20-16
models D-12
See also MOSFETs, JFETs

field programmable gate arrays 25-22

field solver
accuracy updates 18-12
examples 18-45
FAQ A-34
statement syntax 18-36
W Element 18-24

FIL keyword 3-22
file descriptors limit 8-18
files
.a2d 3-55, 3-57, 5-50
AC analysis
measurement 3-57
results 3-57
.ac# 3-55
column lamination 3-28
collapsed data files 3-26
.d2a 5-50
DC analysis
measurements 3-56
results 3-56
design.ac0 24-1
design.mt0 24-1
design.sw0 24-1
external data 3-7, 3-21
filenames 3-22
.ft# 3-55, 3-57, 28-11
.gr# 3-55
graph data 1-9, 3-57
hspice.ini 9-12
creation 3-51
.ic 3-55, 10-4
include files 3-6, 3-29, 3-35
including 3-54
initialization 3-54
input 1-9
file name 3-59
limit on number 8-18
.lis 3-55
.m# 3-55
.MEASURE output 24-1
.ms# 3-55
.mt# 3-55
multiple simulation runs 3-45
names 3-53, 3-59–3-61
output
listings 3-56
names 3-61
status 3-57
version number 3-60

,.pa# 3-55

scratch files 1-12
,.st# 3-55
subcircuit node cross-listing 3-57
,.sw# 3-55
,.tr# 3-55
transient analysis
measurement 3-56
results 3-56
transition data 26-4

filing a documentation bug iv

filters
30 degree phase shift 29-17
active low-pass 27-15
band reject 29-12
low-pass 29-14
pole/zero analysis
Butterworth 27-10
high-pass 27-10
low-pass 27-5, 27-15
transfer functions 29-31

FIND keyword 8-43

flicker noise 20-99

floating point overflow
CONVERGE setting 9-29, 10-38
setting GMINDC 9-33, 10-40

Fluke-Mosaid model 20-5

FMAX
,.FFT parameter 28-8
option 9-35, 10-31, 27-4

FMIN .FFT parameter 28-8

FORMAT .FFT parameter 28-7

Fourier
analysis 11-37
coefficients 11-39
equation 11-39
integral 29-5
transfer function H(f) 29-5
transform 29-5

,.FOURIER statement 11-38

Fowler-Nordheim diodes 15-3, 15-6
capacitances 15-49
equations 15-49
model parameters 15-48
using 15-48

FPGA’s 25-22
FR1 model parameter C-36
FREELIB keyword E-7

FREQ
element parameter 29-10
function
transconductance element statement 29-8
voltage gain element statement 29-9
keyword 29-5
model parameter 8-13, 26-69

frequency
analysis 28-1
Nyquist 29-11
poles and zeros 29-11
complex 29-4, 29-20
critical 29-5
domain to time domain 29-1
domain, transfer function 29-20
low model parameters 16-5
maximum 29-6
frequency analysis 29-11
setting 9-35, 10-31
ratio 12-10
resolution 29-6
Fourier analysis 29-6
frequency analysis 29-10
response 29-4
analysis 29-3
table 29-8, 29-10
parameters 29-9
setting scale 9-35, 10-31
sweep 12-6
table 29-3
variable 7-11
weighing functions 28-3

Frequency Table Model 6-4
frequency-domain model 6-2
Frohman-Bentchkowski equations 21-77
FROM
,.FFT parameter 28-7
keyword 8-52

FS
    keyword 12-13
    option 9-44, 9-46, 11-21, 11-23, 11-35
    FSCAL option 9-35, 9-37, 10-31, 10-33, 27-4
    FT option 9-44, 9-46, 11-21, 11-23, 11-35
timestep control 11-35
    .f# file 3-55, 3-57, 28-11
functions
    A2D 5-50, 26-4
    bisection 30-1
    built-in 7-9–7-11
    D2A 5-50, 26-4
    DERIVATIVE 8-48
    ERR 8-51
    INTEG 8-48
    LAPLACE 27-1, 29-2, 29-6
    NPWL 5-35
    OPTIMIZE 26-3
    POLE 27-1, 29-2, 29-7
    PPWL 5-36
    redefining 7-11
    table 7-9
    user-defined 7-11
See also independent sources
FV function value 26-7

G Elements 26-6, 26-16, 29-26–29-31
    amplitude modulator application 26-62
    AND gate 26-31
    applications 26-6
    controlling voltages 26-19, 26-21
curve smoothing 26-20
data points 26-19
    DELTA parameter 26-58
    element value multiplier 26-20
gate type 26-19
    initial conditions 26-19
    Laplace transform 29-4
    maximum
        current 26-19
        resistance 26-19
    minimum
current 26-19
    resistance 26-19
    multiply parameter 26-19, 29-11
    names 26-18
    parameter value multiplier 29-11
    polynomial
        coefficients 26-19
dimension 26-20
temperature coefficients 26-20
time delay keyword 26-20
transconductance 26-20
triode application 26-61
tunnel diode application 26-58
voltage to resistance factor 26-20
G elements
    syntax statements 5-34
GaAsFET model DC optimization 13-67
gain, calculating 8-31
gate
capacitance 20-94
    AMI 20-94
    charge sharing coefficient 20-72
    equations, JFETs and MESFETs 17-25–17-30
    length/width 20-97
    model (CAPOP=39) 21-194
modeling 21-225
MOSFETs
    example 20-64
    LEVEL 39 21-192, 21-197
parameters
    JFETs 17-17–17-18
    MOSFETs 20-69
    models 20-57
plotting 20-66
SPICE 20-74
temperature equations, JFETs and MES-
FETs 17-42
diode DC parameters, JFETs 17-15–17-16
GAUSS
FFT analysis
    keyword 28-8
    window 28-4, 28-28
functions 13-20
keyword 13-16
parameter distribution 13-14
GBW model parameter 26-69
GEAR algorithm 9-48, 11-13, 11-29
GE-CRD Franz model 20-5
GE-Intersil model 20-6
GENK option 9-19
general model parameter 16-6
geometry
   ACM 17-22
   JFETs and MESFETs parameters 17-7
   MOSFETs model parameters 20-30
   SCALE 15-5
scaling 20-12
SCALM 15-5
substrate diode 16-28
transistor field effect 20-16
getting customer support vi
global
   node names 3-20
   parameters 7-14
   scaling 20-12
   overriding 17-7
   in models 15-4
   .GLOBAL statement 3-19
GMAX option 9-33, 10-40
GMIN 15-5, 16-2
   option 9-39, 10-54, 11-21, 17-6, 20-11
   stepping 9-33, 10-40
GMINDC 15-5, 16-2
   option 9-33, 10-40, 10-42, 10-54, 20-11, 20-26
   JFETs and MESFETs conductance 17-6
GND node 3-17
GOAL keyword 8-46, 13-37
golden parser (IBIS) 19-1
   .gr# file 3-55, 3-57
   maximum version number 8-11
   GRAD optimization parameter 13-42
   gradient data, printing 9-8
GRAMP
   calculation 9-30, 10-39
   conductance option 17-6
   option 9-33, 10-24, 10-40, 10-42, 10-46
   graph data
   file 1-9
   Viewlogic format 9-11
See also .gr# file
   .GRAPH statement 8-2, 31-7
   simulation results 8-10, 8-19
ground
   bounce 25-24
   node name 3-17
   plane, transmission lines C-5
   group delay, calculating 9-51, 12-8
   Grove-Frohman model 20-4
GSCAL
   multiplier 9-36, 10-32
   option 9-35, 10-31, 27-4
GSHUNT option 9-34, 9-39, 10-25, 11-12
   Gummel-Poon model 16-1, 16-50
   Gxxx element parameters 26-18

H

H Elements 26-6, 26-23
   applications 26-6
   controlling voltage 26-25
   data points 26-24
   element
      name 26-24
      value multiplier 26-25
      gate type 26-24
      initial conditions 26-24
      maximum current 26-24
      minimum current 26-24
      polynomial
         coefficients 26-25
         dimension 26-25
         SCR application 26-60
         temperature coefficients 26-25
         time delay keyword 26-25
         transresistance 26-25
   H elements
      syntax statements 5-42
   H parameters 12-21
   H9007 option 9-15
   HAMM FFT analysis keyword 28-8
   Hamming FFT analysis window 28-4, 28-27
   HANN FFT analysis keyword 28-8

Star-Hspice Manual, Release 2000.4
Hanning FFT analysis window 28-4, 28-26
harmonic distortion 12-9, 28-1
HARRIS FFT analysis keyword 28-8
HD2
distortion measure 12-9
parameter 8-35
HD3
distortion measure 12-9
parameter 8-35
HDIF model parameter 20-30
hertz variable 7-11
HGP model parameter C-15, C-24, C-26, C-28
HiCUM model
BJT
parameters 16-95
HiCUM model, BJT 16-93
hierarchical designs, flattened 3-6
hold time 30-4
HP a-Si TFT model 21-207
equations 21-211
topology 21-217
HSPICE
ground for transmission lines C-18
installation directory 3-50
integration method C-20
job statistics report 8-16–8-18
model enhancements 21-218
output, redirecting 3-61
starting 1-12
version
95.3 compatibility 11-35
H9007 compatibility 9-15
specifying 3-59
VERSION parameter 3-31, 21-111
hspace command 3-58–3-62
arguments 3-59
Hspice junction diode model 22-24
hspace.ini file 9-12
creating 3-51
HT model parameter C-14, C-24
HT1 model parameter C-24
hybrid (H) parameters 8-29
hysteresis, Jiles-Atherton example 14-23
IB model parameter 26-69
IBIS
buffers 4-27
conventions 19-3
differential pins 19-36
enhancements 19-44
keywords
optional 19-26
required 19-26
limitations/restrictions 19-5
models 19-1
references 19-49
terminology 19-4
IBOS model parameter 26-70
IC
element parameter 10-9, 26-12, 26-19, 26-24, 26-28
keyword 10-12
parameter 10-9
.ic file 3-55, 10-4
.IC statement 10-3, 10-4, 10-9, 27-3
balancing input nodes 26-66
from .SAVE 10-12
ICSWEEP option 9-34, 10-25
ideal
current sources 10-46
delay elements 26-6, 26-7
DELMAX setting 11-26
op-amp 5-29, 5-32, 26-6, 26-11, 26-13, 26-14
transformer 5-30, 5-33, 26-6, 26-12, 26-15, 26-57
transmission line C-6, C-10
idelay (digital vector file) 5-73
IDS
Cypress depletion model 20-6
equations
LEVEL 1 21-5
LEVEL 13 21-113
LEVEL 2 21-11
LEVEL 3 21-22
LEVEL 38 21-169
LEVEL 5 21-37
LEVEL 6 21-60, 21-71
LEVEL 8 21-93
LEVEL 38 Cypress model 21-162
LEVEL 5
  equations 21-43
  example 21-50
  model 21-34
LEVEL 6
  equations 21-60, 21-71
  example 21-66, 21-68
LEVEL 7 model 21-88
LEVEL 8
  equations 21-93
  model 21-89
IGNOR keyword 8-52
IKF model parameter 16-2
IKR model parameter 16-2
imaginary
  part of AC voltage 8-31
  vs. real component ratio 10-32
imaginary vs. real component ratio 9-36
IMAX option 9-45, 9-47, 11-21, 11-33
  default 11-33
IMIN option 9-45, 9-48, 11-22, 11-33
  default 11-33
impact ionization
  BSIM2 21-197
MOSFETs 20-53
  equations 20-23, 20-53
  LEVEL 39 21-197
  LEVEL 55 22-90
impedance C-72
  AC 8-34
  versus time 25-9
  Z parameters 8-29
impulse response h(t) 29-5
inactive devices
  See latent devices
  .inc file encryption E-12
  include files 3-29, 3-35, 3-54
  .INCLUDE statement 3-6, 3-29, 3-51, 3-52
  and .ALTER blocks 3-42
independent sources
  AC 5-5
    magnitude parameter 5-3
    phase parameter 5-3
    AM function 5-22
    current
      element template listings 8-58
      source name 5-3
      data driven PWL function 5-18
    DC 5-5
      source value parameter 5-3
      elements
        parameters 5-2
        statements 5-2
      EXP function 5-13
      functions 5-7
      mixed types 5-6
      PULSE function 5-7
      PWL function 5-15
      SFFM function 5-20
      SIN function 5-10
      transient 5-5
        source function 5-3
        types 5-7
      voltage
        element template listings 8-57
        source name 5-3
      See also sources
    individual element temperature 13-5
inductance C-75
  matrixes C-52
  mutual 14-18
  scale 9-36, 10-32
  temperature equation 14-17
inductors
  core models 14-12
  coupling 14-18
  current flow 8-23
  device
    equations 14-17
    model 14-12
    element 4-6
    template listings 8-55
    GENK 9-19
    KLIM 9-19
    linear 14-12
      branch relations 14-18
      mutual models 3-30
node names 4-7
  temperature, equation 14-17
INGOLD option 9-6, 9-7
initial conditions 10-3
  file 3-54
  saving and reusing 9-34, 10-25
  statement 10-9
  transient 11-7
initialization 9-35, 10-3, 10-4, 10-27
  file 3-54
  saved operating point 10-11
inline data 3-23
inner sweep 3-25
INOISE parameter 8-35
input
  admittance 8-34
  buffer 19-6, 19-7
    ECL 19-19
    example 19-42
data
  adding library data 3-42
  column laminated 3-28
  concatenated data files 3-26
  deleting library data 3-43
  external, with .DATA statement 3-22
  filenames on networks 3-29
  for data driven analysis 3-21
  formats 3-24, 3-27, 3-28
  include files 3-29
  printing 9-7
  suppressing printout 9-7
FAQ A-12
files
  character case 3-3
  compression 3-2
  configuration file 3-54
  control characters 3-3
  DC operating point 3-54
  demonstration 31-26
  initialization 3-54
  names 3-53, 3-59
  netlist 3-2
  structure 3-6
  table of components 3-7
  unprintable characters 3-3
impedance 8-34
netlist
  file 1-9
  composition 3-9
  delimiters 3-4
  .END statement requirement 3-45
  expressions 3-5
  format 3-2
  guidelines 3-2
  hierarchy paths 3-5
  instance names 3-4
  names 3-3
  nodes 3-4
  numbers 3-5
  parameters 3-5
  schematic vetlists 3-6
  sections and chapter references 3-7
  structure 3-6
See also input files
simulation F-2
input/output
  cell modeling 31-23
  digital vector file 5-70
  drivers 25-18
  ECL buffer 19-23
  open
    drain buffer 19-18
    sink buffer 19-18
    source buffers 19-18
installation
  FAQ A-13
  installation directory $installdir 3-50
  insulation breakdown devices 15-48
int(x) function 7-10
INTEG keyword 8-47, 8-48
integer function 7-10
integration
  algorithms 11-29, C-43, C-61
  backward Euler method 9-48, 11-14
  method C-20
  order of 9-48, 11-14
integrator, behavioral 26-53
Intel PCI Speedway models D-15
interconnect
  analyzing 25-2
intermodulation distortion 12-9
internal
nodes, referencing 3-18
routines 23-38
INTERP option 9-50, 11-12
INTERPOLATION
  element parameter 29-10
interstage gain 8-31
intrinsic model parameters
  limits 22-109
inverse Laplace transform 29-33
inverter
  analysis, transient 2-7
  circuit, MOS 2-7
  gate 26-15
  lookup table 26-43
ion-implanted devices 20-3
IS model parameter 20-27
ISA standard 25-33
ISC model parameter 26-70
isoplanar
  MOSFETs
    construction 20-18
    width cut 20-19
    silicon gate transistor 20-17
ISPICE LEVEL 6 model 21-72
iterations
  algorithm 11-31
  count algorithm 11-33
  extra 9-34, 10-26
  limit 9-23, 10-25
  maximum number of 9-46, 11-23
  number 13-54
ITL1
  calculation 9-30, 10-39

option 9-23, 10-25
ITL2 option 10-26
ITL3 option 9-45, 11-22
ITL4 option 9-46, 11-22
ITL5 option 9-46, 11-23
ITLPZ option 9-42, 10-31, 27-4
ITROPT optimization parameter 13-42
ITRPRT option 9-50, 11-13
I-V and C-V plotting demo 31-7

J

Jacobian data, printing 9-8
JFETs
capacitance
  equations 17-6, 17-25
  parameters 17-29
CAPE=2 model parameters 17-29
control options 17-6
convergence 17-6
current convention 17-7
current flow 8-24
DC
  model
    equation selector 17-13
    LEVEL 1 parameters 17-19
    LEVEL 2 parameters 17-19–17-20
    LEVEL 3 parameters 17-21–17-22
voltage equations 17-31
element
  names 4-17
  template listings 8-61
equivalent circuits 17-7, 17-8
gate
  capacitance parameters 17-17–17-18
  diode DC parameters 17-15–17-16
length 4-17
model
  constants 17-11
  names 17-13
  parameters 17-6, 17-13, 17-14
  specifying 17-3
  statements 17-13
  variables 17-11
n-channel specification 17-13
noise
  equations 17-35
  parameters 17-35
  summary printout 17-37
output conductance 17-9
overview 17-2
p-channel specification 17-13
power dissipation 8-27
scaling 17-7
temperature
  equations 17-38, 17-41–17-44
  parameters 17-38, 17-39
TOM model parameters 17-46
transconductance 17-8
width 4-17
Jiles-Atherton
  example
    hysteresis 14-23
    parameter extraction 14-24
    varying parameters 14-20
  model 14-12, 14-19
  examples 14-20
  parameters 14-16
JIS model parameter 26-70
JS model parameter 20-27
JSDBR model parameter 15-35
JSDGR model parameter 15-35
JSDSR model parameter 15-35
JSGBR model parameter 15-35
JSGGR model parameter 15-35
JSGSR model parameter 15-35
JSGBR model parameter 15-35
JSGGR model parameter 15-35
JSW model parameter 20-27
Juncap diode
  electrical variable 15-39
  internal variable 15-40
  leakage 15-44
Juncap diodes
  capacitance charge 15-38
  current voltage 15-37
  equations 15-38
  model parameters 15-35
JUNCAP model parameters 22-62
  junction
    capacitance 16-6
  capacitor parameters 16-11
DC 15-23
diodes
  element parameters 15-6
  equations 15-22
  geometric 15-6
  line-to-line 18-8
  nongeometric 15-6
  parameters 15-6
  Silicon diffused 15-1
  temperature 15-19
model
  parameters 15-8
  statement 15-7
parameters
  MOSFET's LEVELs 49 and 53 22-44
  setting
    capacitance 15-12
    DC LEVEL 1 and 3 15-9
Junction Cap Model 15-33
Junction diode
  ON/OFF 15-41
K
  model parameters 14-20
KAISER FFT analysis keyword 28-8
Kaiser-Bessel FFT analysis window 28-4, 28-6, 28-29
KCLTEST 10-26
  option 9-21
KD model parameter C-15, C-25, C-26, C-28
KD1 model parameter C-25
KD2 model parameter C-25
Kerwin's circuit, pole/zero analysis 27-8
keywords
  .AC statement parameters 12-5
  analysis statement syntax 13-39
  BART 28-8
  command syntax 30-8
  DATA statement parameters 3-21
  .DC statement parameters 10-15
  DTEMP 13-5
  ERR1 13-37
  .FFT statement parameters 28-7
  FREELIB E-7

Star-Hspice Manual, Release 2000.4
FREQ 29-5
FS 12-13
GOAL 13-37
IBIS optional 19-26
LAST 8-42, 8-43, 8-45
.MEASUREMENT statement parameters 8-47
.MODEL statement parameters 8-12
MONTE 13-14
optimization syntax 13-38
PAR 7-8
power output 8-26
PP 8-46, 8-47
RECT 28-8
source functions 5-2
target syntax 8-42
.TRAN statement parameters 11-5
UNORM 28-7
weight 8-47
KF model parameter 12-13
Kirchhoff’s Current Law (KCL) test 9-21, 10-26
LAST keyword 8-42, 8-43, 8-45
LATD model parameter 20-30
latent devices
  BYPASS option 9-42, 11-11, 11-16, 20-13
  bypassing 20-13
  BYTOL option 9-42, 11-16
  excluding 9-42, 11-17
  MBYPASS option 9-42, 11-16
  removing from simulation 9-42, 11-16
  VNTOL option 9-42, 11-16
Lattin-Jenkins-Grove model 20-5
layer stacks, transmission line 18-37
LC oscillator model 26-83
LD model parameter 20-30
LDAC parameter 20-98
LDIF model parameter 20-30
leadframe example 31-12
LENGTH model parameter 13-22
LENNAM option 9-6
LEVEL
  diode parameter 15-4
  element parameter 29-11
  model
    parameter C-13, C-14
    selector 17-3
  optimization parameter 13-42
levels, MOSFETs models 20-3, 20-4
Levenberg-Marquardt algorithm 13-53
LEVIN model parameter 26-70
LEVOOUT model parameter 26-70
LG model parameter 15-35
.LIB
  call statement 3-32
  definition statement 3-33
  building libraries 3-33
  statement 3-6, 3-52
  in .ALTER blocks 3-32, 3-41
  nesting 3-33
  with .DEL LIB 3-42
  with multiple .ALTER statements 3-42
.lib file encryption E-11

L

L capacitor parameter 15-6
LAM keyword 3-22, 3-29
lambda equations, MOSFETs LEVEL 6 21-61, 21-63
laminated data 3-28
LAPLACE
  element parameter 29-11
  function 27-1, 29-2, 29-5, 29-6, 29-24–29-28
  modeling 29-20
  transconductance element statement 29-6
  voltage gain element statement 29-6
Laplace
  band-reject filter 29-12
  low-pass filter 29-14
  parameters 29-9
  transfer function 29-2, 29-20
  transform 29-2, 29-6, 29-21, 29-22
  frequency
    analysis 29-11
    response table 29-8
  function call 29-4
  inverse 29-33

modeling 29-20
  POLE (pole/zero) function 29-28
LAST keyword 8-42, 8-43, 8-45
LATD model parameter 20-30
latency option 20-13
latent devices
  BYPASS option 9-42, 11-11, 11-16, 20-13
  bypassing 20-13
  BYTOL option 9-42, 11-16
  excluding 9-42, 11-17
  MBYPASS option 9-42, 11-16
  removing from simulation 9-42, 11-16
  VNTOL option 9-42, 11-16
Lattin-Jenkins-Grove model 20-5
layer stacks, transmission line 18-37
LC oscillator model 26-83
LD model parameter 20-30
LDAC parameter 20-98
LDIF model parameter 20-30
leadframe example 31-12
LENGTH model parameter 13-22
LENNAM option 9-6
LEVEL
  diode parameter 15-4
  element parameter 29-11
  model
    parameter C-13, C-14
    selector 17-3
  optimization parameter 13-42
levels, MOSFETs models 20-3, 20-4
Levenberg-Marquardt algorithm 13-53
LEVIN model parameter 26-70
LEVOOUT model parameter 26-70
LG model parameter 15-35
.LIB
  call statement 3-32
  definition statement 3-33
  building libraries 3-33
  statement 3-6, 3-52
  in .ALTER blocks 3-32, 3-41
  nesting 3-33
  with .DEL LIB 3-42
  with multiple .ALTER statements 3-42
.lib file encryption E-11
libraries
    adding with .LIB 3-42
    ASIC cells 3-51
    AvanLink B-5
    building
        rules 3-34
        with .LIB definition 3-33
    configuring 7-17
    creating parameters 7-14
    DDL 3-50
    accessing 3-36
    defining macros 3-33
    deleting 3-42
    device D-1
    duplicated parameter names 7-14
    encryption E-1
    .END statement 3-34
    integrity 7-13
    listings D-5
    private 3-40
    protecting 3-40, E-1
    search
        order 3-51
        path 3-51
        selecting 3-37
        subcircuits 3-52
        vendor 3-51
    licenses issues, FAQ A-15
    limit descriptors command 8-18
    LIMIT keyword 13-17
    limitations, known A-17
    LIMPTS option 9-13
    LIMTIM option 9-11
    LIN keyword 10-16, 11-6, 12-6
    linear
        region equations, MOSFETs LEVEL 47 22-14
        technology device models D-14
        .lis file 3-55, 3-56
    LIST option 9-7
    listing
        page width 8-14
        suppressing 3-40
    LLEV model parameter C-13, C-14
    LM capacitor parameter 15-6
    LMAX model parameter 1-5
    LMIN model parameter 1-5
    .LOAD statement 10-11
    local
        parameters 7-14
        truncation error algorithm 9-41, 9-48, 11-13, 11-19, 11-33, 11-34
        timestep 9-40, 11-18
    log(x) function 7-10
    log10(x) function 7-10
    logarithm function 7-10
    lossless
        parameter combinations C-37
        transmission lines C-6
        model C-70
    lossy
        transmission lines C-6, C-11
        model C-71
        U model C-13
    low-frequency large-signal characteristics
        using Ebers-Moll model 16-5
    LP capacitor parameter 15-6
    LRD model parameter 20-29
    LRS model parameter 20-29
    LS model parameter 15-35
    LSCAL option 9-36, 10-32, 27-4
    Lsim models, calibrating 24-1
    lumped parameters C-19
    LV 8-36
    LV18 model parameter 31-7
        timestep control 11-35
    LX 8-36
    LX7 model parameter 31-7
    LX8 model parameter 31-7
    LX9 model parameter 31-7

M

M capacitor parameter 15-6
    element parameter 15-14, 26-19, 26-28, 29-11
multiplier parameter
U and T Elements C-5
.ma# file 3-55, 3-57
mA741 op-amp 26-76
.MACRO statement 3-12
macros
  defining with .LIB definition 3-33
deleing library data 3-42
magnetic core 14-12
element outputs 14-16
models 3-30
  ferromagnetic core 14-19
  names 14-12
  parameters 14-14, 14-16
  saturable core 14-12
  statement 14-12
outputs (table) 14-16
magnitude
  AC voltage 8-31
calculating 8-31
MANU model parameter 26-70
manufacturing tolerances 13-21
Marquardt scaling parameter 13-53
mask (digital vector file) 5-73
matrix
  calculations 9-23, 10-26
  conductance 18-4
  minimum pivot values 9-25, 10-29
  parameters 12-14
  properties 18-4
  resistance 18-4
  row/matrix ratio 9-25, 10-29
  size limitation 9-25, 10-28
MAX
  element parameter 26-13, 26-19, 26-24, 26-28, 29-11
  function 30-5
  keyword 8-47
  optimization parameter 13-42
max(x,y) function 7-10
MAXAMP option 9-22, 9-39, 10-26, 11-17, 12-8
MAXD model parameter C-15
MAXF
  element parameter 29-11
parameter 29-6
MAXFLD keyword 12-14
maximum
  number size 9-11
  value, measuring 8-46
MAXORD option 9-48, 11-14
Maxwell matrix C-34
MBYPAS option 20-11
MBYPASS option 9-42, 9-43, 11-16, 11-17, 20-13
mean, statistical 13-3
MEASDGT option 9-6, 9-7
MEASFAIL option 9-50
measfail value 30-8
MEASOUT option 9-12
measure
  data output formatting 9-7
  parameter types 8-38
.MEASURE statement 8-2, 8-3, 8-38, 9-7, 9-12
  failure message 8-37
  parameters 7-7
  PCI modeling 25-41
  performance 8-3
  TDR files 25-9
Mentor interface 9-12
MENTOR option 9-12
MESFETs
  capacitance equations 17-6, 17-25
  CAPOP=2 model parameters 17-29
  control options 17-6
  convergence 17-6
  DC
    model equation selector 17-13
    voltage equations 17-31
  element names 4-16
  equivalent circuits 17-7
  models
    constants 17-11
    names 17-13
    parameters 17-6, 17-13, 17-14
    specifying 17-3
    statements 17-13
    variables 17-11
  n-channel specification 17-13
noise equations 17-35
parameters 17-35
summary printout 17-37
overview 17-2
p-channel specification 17-13
scaling 17-7
temperature equations 17-38, 17-41–17-44
messages
pivot change 9-25, 9-27, 10-28, 10-31
See also errors, warnings
Metaencrypt features E-9
Metaencrypt, character length restrictions E-6
metal and poly capacitance equations 15-27
metal contact 20-17
METHOD option 9-49, 11-14
Meyer
and Charge Conservation Model parameters 8-64
capacitance
modified 20-81
parameters 20-71
gate capacitances
modified 20-77
model 20-57
microstrip transmission line C-21
MIN
element parameter 26-13, 26-19, 26-24, 26-29
keyword 8-47
min(x,y) function 7-10
minimum
number size 9-11
value, measuring 8-46
MINVAL keyword 8-46, 8-52
mixed
sources 5-6
mixed mode
simulation 5-50, 26-4
See also D2A, A2D
mixed-signal simulation
See mixed mode
MJ model parameter 20-28
MJSW model parameter 20-28
mobility

equations, MOSFETs LEVEL 47 22-12
parameters
curve fitting 21-11
MOSFETs
LEVEL 2 21-10–21-11
LEVEL 3 21-22
LEVEL 38 21-167
LEVEL 5 21-35–21-36
LEVEL 8 21-92–21-93
reduction equations, MOSFETs
LEVEL 2 21-14
LEVEL 38 21-174
LEVEL 5 21-47
temperature equations, JFETs and MESFETs 17-44
.MODEL
syntax 6-4, 18-31
model
parameters See model parameters diodes
model analysis options 9-16, 9-16–9-18
BJTs, EXPLI 9-19
DCAP 9-16
diodes, EXPLI 9-19
inductors
GENK 9-19
KLIM 9-19
MODSRH 9-16
MOSFETs 9-18
SCALM 9-17
TNOM 9-17
MODEL keyword 10-15, 13-39
model names, periods in D-3
model parameters
A2D 5-50
ACM 20-9
.ALTER blocks 3-41
basic 21-2–21-4
BISECTION 30-7
BJTs 16-6–16-13
base width modulation 16-9
beta degradation 16-5
high current Beta degradation 16-10
junction capacitance 16-11
LEVEL 2 16-13
LEVEL 8, HiCUM 16-95
low current Beta degradation 16-9
model name 16-4
noise 16-13
parasitics
capacitance 16-12
resistance 16-10
temperature
effects 16-14–16-19
parameters 16-14
transistor 16-4
transit time 16-12
BV 15-4
capacitance 14-9
capacitance distribution 13-24
D2A 5-50, 5-51–5-52
CHI 5-51
DCAP 17-6
DELVTO 13-9
diodes 15-6
junction 15-7, 15-8
AREA 15-9
BV 15-11
capacitance 15-12
DC 15-9–15-11
diodes
model 15-7, 15-10
LEVEL 15-7, 15-10
model names 15-7
level 15-4
noise 15-13
scaling 15-15
temperature 15-19–15-21
DTEMP 13-5
Ebers-Moll 16-5
Fowler-Nordheim diodes 15-48
geometry, JFETs and MESFETs 17-7
GRAPH statement parameters 8-13
IKF 16-2
IKR 16-2
intrinsic limits 22-109
JFETs 17-6, 17-29, 17-35, 17-46
ACM 17-22
capacitance 17-29
DC
LEVEL 1 17-19
LEVEL 2 17-19–17-20
LEVEL 3 17-21–17-22
gate capacitance 17-17–17-18
gate diode DC 17-15–17-16
noise 17-35
temperature 17-39–17-41
JFETs and MESFETs 17-14
DCAP 17-6, 17-25
GMIN 17-6
LEVEL 17-13
level selector 17-3
NIF 17-13
PIF 17-13
See also model parameters, JFETs
metal and poly capacitors 15-13
MONO 8-13
MOSFETs 20-14, 21-76, 21-79, 21-81, 21-82
LEVEL 59 22-149
LEVEL 61 22-176
LEVEL 62 22-184
op-amps 26-67, 26-67–26-72
default values 26-72
output 8-13
PHOTO 13-22
range limit 22-49
RSH 13-9
scaling 20-12
sigma deviations, worst case analysis 13-9
skew 13-8
SUBS 16-2
suppressing printout of 9-8
TEMP 3-20, 13-5
temperature analysis 13-5
TIC 8-13
TOM model 17-45, 17-46
TOX 13-9
transmission lines
generic
coax models C-26
models C-24
  twinlead models C-28
loss parameters C-16
measured parameters C-36
planar models C-14
precomputed parameters C-30
TREF 13-4, 13-5, 13-6
U models C-14
  basic ELEV parameters C-36
  common planar parameters C-24
  ELEV = 1 parameters C-28
geometric
coaX parameters C-26
twinlead parameters C-15
measured parameters C-35
wire models 14-2, 14-4-14-5
table 14-5
XPHOTO 13-22
model selection
  automatic D-1-D-4
  failure to find a model D-3
  program 20-9, D-3
  See also automatic model selection
  syntax 20-14
.MODEL statement 3-30, 13-5, 15-6, 20-4
  BISECTION syntax 30-7
BJTs 16-3, 16-4
BSIM models 21-111
capacitance 14-9
diode junction 15-7
examples, NMOS model 22-53
ferromagnetic cores 14-12
for .GRAPH 8-11
HSPICE version parameter 3-31
magnetic core 14-12
model name 3-30
MOSFETs 20-14
op-amp 26-66
OPT method 30-7
optimization syntax 13-40
PASSFAIL syntax 30-7
U model C-13
VERSION parameter 21-111
wire RC 14-2
.MODEL statements and MESFETs 17-13
models
  ACM selector 17-3
  algebraic 11-27
  automatic selection 20-9
BJTs 3-30
  HiCUM 16-93
  quasi-saturation 16-50
  statement 16-4
BSIM LEVEL 13 3-31
BSIM2 LEVEL 39 3-31
bulk charge effect 20-3
capacitance 14-9, 20-7
capacitors 3-30
model selector 17-3
characterization 10-14
Curtice 17-3
depletion MOS devices 20-3
device
  capacitor 14-9
  inductor 14-12
  resistor 14-2
diode 3-30, 15-4
define 15-16
junction 15-7
scaling 15-14
statements 15-4
Ebers-Moll 16-1, 16-5
equations
  LEVEL 61 22-178
  LEVEL 62 22-186
examples, Jiles-Atherton 14-20
FAQ A-20
ferromagnetic cores 14-19
Gummel-Poon 16-1
IBIS 19-1
ion-implanted devices 20-3
JFETs 3-30
JFETs and MESFETs
capacitor 17-5
  DC models 17-1
Jiles-Atherton core 14-19
devices
  parameters 15-8
  statement 15-7
LV18 31-7
LX7, LX8, LX9 31-7
magnetic core 3-30, 14-12
Monte Carlo
  analysis 13-14
  example 13-26
  parameter distribution 13-18
MOS compare 21-218
MOSFETs 3-30
  Berkeley
    BSIM3-SOI 22-114
    junction 22-24
  BSIM 21-101
    equations 21-112
    LEVEL 13 example 21-121
  BSIM2 21-179
    equations 21-186
  BSIM3 22-2
    equations 22-10
    Left/wet 22-9
  BSIM3-SOI DD 22-161
  BSIM3-SOI FD 22-147, 22-149, 22-157
  BSIM3v3
    MOS 22-19
    NQS 22-23
  BSIM4 22-66, 22-67
  Cypress 21-162, 21-169
  Empirical 21-19
    equations 21-22
  EPFL-EKV 22-85
    equations 22-92
    NQS equations 22-106
  Frohman-Bentchkowski, equations 21-77
  HP aSi TFT 21-207
    equations 21-211
  Hspice junction 22-24
  IDS
    LEVEL 5 21-34
      equations 21-43
      example 21-50
    LEVEL 6
      equations 21-60, 21-71
      example 21-66, 21-68
    LEVEL 6 and LEVEL 7 21-52
    LEVEL 7 21-88
    LEVEL 8 21-89
      equations 21-93
    LEVEL 55, updates 22-111
    levels 20-3, 20-4
    LEVELs 49 and 53, equations 22-52
    modified BSIM LEVEL 28 21-147
      equations 21-155
    MOS 21-33
    Philips MOS9 22-56
    quasi-static equations 22-103
    RPI aSi TFT 22-175, 22-176
    RPI Pol-Si TFT 22-182
    Schichman-Hodges 21-2, 21-7
      equations 21-5, 21-11
    SOSFET’s LEVEL 27 21-136
    UFSOI 22-133
  mutual inductors 3-30
  names 3-30
  National Semiconductor 15-50, 16-55
  op-amps 3-30
    creating 26-65
    optimization 3-30
  parameters
    DTEMP 31-18
    Fowler-Nordheim 15-48
  passive device 14-1
  plot 3-30
  private 3-40
  protecting 3-40
  quasi-saturation 16-50
  reference temperature 13-5
  resistor equations 14-5
  scaling 20-12
    diode 15-14
    selecting 17-3
    silicon-on-sapphire 20-3
  simulator access 3-35
  SOSFET’s 20-3
  specifying 3-51, 20-14
  SPICE capacitor 17-5
  Statz 17-3
  subcircuit MULTI 3-15
  transient 16-5
  types 3-30
  typical set 13-12–13-13
  VBIC bipolar transistor 16-57
  wire
    parameters 14-4
RC 14-2
modified BSIM LEVEL 28
equations 21-155
models 21-147
MONO model parameter 8-13
Monte Carlo
AC analysis 12-5
analysis 13-2, 13-3, 13-26–13-34
demo files 31-27
distribution options 13-16–13-17
DC analysis 10-15
time analysis 11-4
MONTE keyword 11-5, 12-5, 13-14
MOS
diodes 20-9
inverter circuit 2-7
model 21-33
op-amp optimization 13-70
MOS2 model 20-4
MOS3 model 20-4
MOSFETs
Berkeley
BSIM3-SOI DD 22-161
BEX factor 21-127
bulk capacitance 20-23
charge
capacitance 
effective length and width 20-97
equations 20-74–20-98
Meyer model 20-57
models 20-57
scaling parameters 20-68
CAPOP 20-69
capacitance selector 20-74–20-98
channel length modulation temperature 
equations 20-111
drain diffusion area 4-19
effective length and width 20-97
output conductance 20-54
element 
names 4-18
template listings 8-62
energy gap temperature equations 20-105
EPFL-EKV NQS equations 22-106
equation variables and constants 20-20
equivalent circuits 20-22
AC analysis 20-24
AC noise analysis 20-25
transient analysis 20-23
drain and source resistance 20-33, 20-36, 20-41, 20-44
saturation current 20-33, 20-44
equation 20-45
GEO element parameter 20-42
gate
impact ionization equations 20-23, 20-53
initial conditions 4-20

Star-Hspice Manual, Release 2000.4
isoplanar
construction 20-18
silicon gate 20-17
width cut 20-19
level parameter 20-14
LEVELs 6, 7 UPDATE selector 21-57
Meyer capacitance model parameters 20-71
mobility temperature equations 20-111
model
RPI a-Si TFT 22-175
model analysis options 9-18–9-19
model parameters
A0 22-150
A1 22-150
A2 22-150
AGIDL 22-150
AGS 22-150
ALPHA0 22-150
ASD 22-154
AT 22-156
B0 22-150
B1 22-150
BGIDL 22-150
BGID 22-150
BSIM3-SOI FD 22-149
BSIM4 22-67
CAPMOD 22-149
CDSC 22-150
CDSCB 22-150
CDSCD 22-151
CF 22-154
CGDL 22-154
CGDO 22-155
CGSL 22-155
CGSO 22-155
change conservation 20-71
CIT 22-151
CJSWG 22-155
CKAPPA 22-155
CLC 22-155
CLE 22-155
CSDESW 22-155
CSDMIN 22-155
CTH0 22-156
DELT A 22-151
DLC 22-155
DROUT 22-151
DSUB 22-151
DVT0 22-151
DVTOW 22-151
DVT1 22-151
DVT1W 22-151
DVT2 22-151
DVT2W 22-151
DWB 22-152
DWC 22-155
DWG 22-152
effective width/length 21-9
ETA0 22-152
ETAB 22-152
gate capacitance
basic 20-69
Meyer 20-71
overlap 20-70
impact ionization 20-53
ISBJT 22-152
ISDIF 22-152
ISR EC 22-152
ISTUN 22-152
K1 22-152
K2 22-152
K3 22-152
KB1 22-152
KETA 22-152
KT 22-156
KT1 22-156
KT2 22-156
KTIL 22-156
LEVEL 22-149
Level 60 22-164
LINT 22-153
MJSWG 22-155
MOBMOD 22-149
NCH 22-149
NDIO 22-153
NFACTOR 22-153
NGATE 22-149
NGIDL 22-153
NLX 22-153
NOIMOD 22-149
noise 20-99
NSS 22-154
NSUB 22-149
Star-Hspice Manual, Release 2000.4
NTUN 22-153
PBSWG 22-156
PCLM 22-153
PDBLCL1 22-153
PDBLCL2 22-153
PT 22-156
PRWB 22-153
PRWG 22-153
PVAG 22-153
RBODY 22-153
RBSH 22-153
RDSW 22-153
RSH 22-153
RTHO 22-156
SHMOD 22-149
SIH0 22-150
SIH 22-150
SIH2 22-151
SIHD 22-151
TBOX 22-149
temperature 20-102
threshold voltage 20-50
TNOM 22-156
TOX 22-149
TSI 22-149
TT 22-156
U0 22-153
UA 22-154
UA1 22-156
UB 22-154
UB1 22-156
UC 22-154
UC1 22-157
UTE 22-157
VBSA 22-154
VERSION 22-20
VOFF 22-154
VSAT 22-154
VSDFB 22-156
VSDTH 22-156
VTH0 22-154
WINT 22-154
WR 22-154
XBJT 22-157
XDIF 22-157
XPART 22-156

See also Chapter 20
models

AMD 20-5, 20-6
AMI-ASPEC 20-5
ASPEC-AMI 20-5
Berkeley
BSIM3-SOI 20-7, 22-114
BSIM3-SOI DD 22-161
junction 22-24
BSIM 20-5, 21-101
equations 21-112
LEVEL 13 example 21-121
BSIMv3
MOS 22-19
NQS 22-23
BSIM2 20-6, 21-179
equations 21-186
BSIM3 20-6, 22-2
equations 22-10
BSIM3-SOI FD 22-147
BSIM3-SOI
BSIM2 20-6, 21-179
equations 21-186
BSIM3 20-6, 22-2
equations 22-10
GEC 20-5
model (GTE style) 20-5
Rutherford 20-6
Cypress 20-5, 21-162, 21-169
Dallas Semiconductor 20-5
Empirical 21-19
equations 21-22
EPFL-EKV 20-7, 22-85
equations 22-92
Fluke-Mosaid 20-5
Frohman-Bentchkowski, equations 21-77
GE-CRD-Franz 20-5
GE-Intersil 20-6
Grove-Frohman 20-4
HP a-Si TFT 21-207
equations 21-211
Hspice junction 22-24
HSPICE PC version 20-4
IDS
Cypress depletion 20-6
LEVEL 5 21-34
equations 21-43
example 21-50
LEVEL 6
equations 21-60, 21-71
example 21-66, 21-68
LEVEL 7 21-88
LEVEL 8 21-89
equations 21-93
Lattin-Jenkins-Grove 20-5
LEVEL 61 circuit 22-178
LEVEL 62 22-186
levels 20-4
LEVELs 49 and 53 equations 22-52
modified BSIM LEVEL 28 21-147
equations 21-155
MOS 21-33
compare 21-218
MOS2 20-4
MOS3 20-4
MOS9 22-56
Motorola 20-6
National Semiconductor 20-6
Philips MOS9 22-56
quasi-static, equations 22-103
RPI a-Si TFT 22-176
RPI Poli-Si parameters 22-184
RPI Poli-Si TFT 22-182
Schichman-Hodges 20-4, 21-2, 21-7
equations 21-5, 21-11
SGS-Thomson 20-6
Sharp 20-6
Siemens 20-5, 20-6
Sierra 1 20-5
Sierra 2 20-6
Siliconix 20-5
SOSFET’s 20-6
LEVEL 27 21-136
statement 20-14
STC-ITT 20-5
Taylor-Huang 20-5
TI 20-6
University of Florida SOI 20-7, 22-133
user defined 20-5
VTI 20-6
n-channel specification 20-14
node names 4-18
noise 20-99
model equations 20-99
model parameters 20-99
summary printout 20-101
p-channel specification 20-14
perimeter
drain junction 4-19
source junction 4-19
power dissipation 8-28
saturation current temperature equations 20-106
SCALM 9-18
sensitivity factors 21-154
source
diffusion area 4-19
drain sharing selector 4-20
SPICE compatibility 20-4
squares
per drain diffusion 4-19
per source diffusion 4-19
surface potential temperature equations 20-109
temperature
coefficient model parameters 20-104
differential 4-20
effects parameters 20-102
parameters 20-102
template input
BSIM3-SOI FD 22-157
threshold voltage
model parameters 20-50
temperature equations 20-110
transconductance 20-22
UPDATE parameter usage 21-128
WIC ids current selector 21-73
WL 9-18
zero-bias voltage threshold shift 4-20
Motorola model 20-6
MS model parameter 14-19
.ms# file 3-55, 3-56
MSPICE simulator interface 9-12
.mt# file 3-55, 3-56
MU option 9-49, 11-17
Muller algorithm 27-3
starting points 9-37, 10-33
MULTI 3-15
multiconductor capacitance/conductance C-32
multiconductor systems 18-3
Multi-Level Gamma model, example 21-66, 21-68
multiple .ALTER statements 3-42
multiplier
  function, U and T Elements C-5
  G and E Element values 29-11
  GSCAL 9-36, 10-32
multiply parameter 3-48, 4-3, 5-3
  subcircuits 3-15
multipoint experiment 1-8
  examples 1-8
multi-terminal network 6-1
multi-threading 3-63
mutual
  inductance C-76
  inductor
coupling coefficient 14-18
  element 4-9
  template listings 8-56

N

N model parameter 20-27
NAND gate adder 31-4
narrow width effect 21-13
National Semiconductor model 16-55, 20-6
  converting 15-50
natural
  frequency 27-2
  log function 7-10
NB model parameter 15-36, 20-28
n-channel, MOSFET’s models 3-30
ND model parameter C-14
NDIM 5-24
NDS model parameter 20-27
negative conductance, logging 9-15
nested library calls 3-33
.NET statement 12-14
netlist 3-6
  AvanLink B-6
  AvanLink-DA B-12
  encrypting E-1
  FAQ A-27
  flat 3-6
input files 3-2
  schematic 3-6
network
  analysis 12-14
  filenames 3-29
  multi-terminal 6-1
  output 8-34, 12-15
  switched capacitor 29-48
  variable specification 12-18
NEWTOL option 9-34, 10-26
NG model parameter 15-36
NL model parameter C-24
NLAY model parameter C-16, C-24
nodal voltage output 8-21, 8-30
NODE option 9-7
  suppressing listing 3-40
nodes
  connection requirements 3-17
  cross-reference table 9-7
  floating supply 3-17
  global versus local 3-20
  internal 3-18
  MOSFET’s substrate 3-17
  names 3-16, 3-17, 3-19, 31-7
    automatic generation 3-19
    ground node 3-17
    period in 3-17
    subcircuits 3-17, 3-18
    zeros in 3-19
  numbers 3-16, 3-17
  phase or magnitude difference 8-31
  printing 9-7
  shorted 10-47
  terminators 3-17
  voltages, encrypting E-1
NODESET keyword 10-12
.NODESET statement 9-29, 10-3, 10-23, 27-3
  balancing input nodes 26-66
  DC operating point initialization 10-10
  from .SAVE 10-12
  node-to-element list 9-25, 9-27, 10-28, 10-31
NOELCK option 9-8
Noise 22-43
  noise
analysis 16-58  
BJTs 16-6  
calculations 12-12  
CMII Noise 23-29  
coupled line  
  example 25-27  
  noise 25-27  
equations  
  BJTs 16-40  
  JFETs and MESFETs 17-35  
  junction diode 15-28  
  resistor 14-7  
folding 12-13  
input 8-35  
JFETs and MESFETs  
equations 17-35  
summary printout 17-37  
MOSFETs 20-99  
equivalent circuits 20-25  
models  
  EPFL-EKV equations 22-107  
numerical 9-39, 11-12  
output 8-35, 12-12  
parameters 15-13, 16-13  
  BJTs 16-13  
  JFETs and MESFETs 17-35  
  MOSFETs  
    BSIM3v3 22-43  
    LEVEL 55 22-91  
problems and solutions 25-3  
quota 25-6, 25-8  
resistor 14-7  
equation 14-7  
sampling 12-13  
sources 25-3, 25-5  
thermal 14-7  
.NOISE statement 12-11  
NOISENP  
frequency Table Model 6-4  
NOMOD option 9-8  
NONE keyword 10-7, 10-13  
nonlinear elements 29-2  
NonQuasi-Static (NQS) model 22-23  
equations 22-106  
parameters 22-45  
nonvolatile memory diodes 15-1  
NOPAGE option 9-8  
NOPIV option 9-23, 10-26  
NORM FFT analysis keyword 28-7  
norm of the gradient 13-53  
Normal Field equations 21-78  
NOTOP option 9-8  
NOWARN option 9-15  
NP_ FFT parameter 28-7  
NPDELAY element parameter 26-13, 26-19, 26-24, 26-29  
npn BJT models 3-30  
NPWL  
  element parameter 26-19  
  function 5-35, 26-17  
  NMOS transistor function 26-10  
NS model parameter 15-36  
NSUB model parameter 20-28  
NT, FAQ A-23  
numbers  
  formatting 9-6, 9-7  
  maximum size 9-11  
  minimum size 9-11  
NUMDGT option 9-8  
numerical  
  integration  
    algorithms 9-49, 11-14  
    order of 9-48, 11-14  
    noise 9-39, 11-12  
    problems 9-34, 9-39, 10-25, 11-12  
NUMF keyword 12-13  
NXX option 9-6, 9-8  
Nyquist critical frequency 29-5, 29-11  

obtaining customer support vi  
O.C.G. matrix C-33  
OCT keyword 10-16, 11-6, 12-6  
OD1 model parameter C-29  
odelay (digital vector file) 5-73  
OFF  
  element parameter 9-35, 10-4, 10-27
option 9-35, 10-4, 10-27
one-dimensional function 5-25
ON/OFF Condition 15-41
ONOISE parameter 8-35
.OP statement 10-5, 10-6, 11-3
.OP statement parameters 10-6
OPAMP element parameter 26-13
op-amps
  automatic generation 26-5
  behavioral models 26-65
  characterization 24-7
  common mode rejection ratio 26-68
  compensation level selector 26-68
  diode and BJT saturation current 26-69
  element statement 26-66
  excess phase parameter 26-69
  gain parameter 26-67
  input
    bias current 26-69
    bias offset current 26-70
    level type selector 26-70
    offset current 26-70
    offset voltage 26-72
    short circuit current 26-70
  internal feedback compensation capacitance 26-68
  JFETs saturation current 26-70
  mA741 model 26-76
  manufacturer’s name 26-70
  model
    generator 26-65
    names 3-30
    parameters 26-67
    defaults 26-72
    selector 26-68
  .MODEL statement 26-66
  open loops 10-46
  optimization 13-70
  output
    level type selector 26-70
    resistance ROAC 26-71
    slew rate 26-71
    voltage 26-72
    phase margin 26-69
    power
dissipation 26-70
  supply voltage 26-71
  subcircuit example 26-74
  subcircuit generator 26-5
temperature parameter 26-71
  unity gain frequency 26-69
operating point
capacitance
  printout 20-61
  values table 9-28, 10-22
  Early voltage 22-108
  estimate 10-5, 11-3
  .IC statement initialization 10-9
  initial conditions 3-54
  .NODESET statement initialization 10-10
  numerical values of model internal variables 22-108
  Overdrive voltage 22-108
  pole/zero analysis 27-3
  restoring 10-13
  saturation / non-saturation flag 22-109
  saturation voltage 22-108
  saving 3-19, 10-12
  solution 9-35, 10-3, 10-4, 10-27
  SPICE-like threshold voltage 22-108
  transconductance efficiency factor 22-108
  transient 11-3
  voltage table 10-7
operating systems, HSPICE 1-6
operators
  arithmetic 7-9
  Laplace transforms 29-22
OPT keyword 13-38, 30-8
optimization
  AC analysis 12-4, 13-60
  algorithm 13-42
  analysis statements 13-39
  behavioral models 26-46, 26-50
  bisection method 30-5, 30-7
  CMOS tristate buffer 13-55
  control 13-36
  convergence options 13-36
  curve-fit 13-37
  cv 31-33
  data-driven vs. s-parameters 13-61
error function 8-40
example 13-44, 31-24
goal 13-37
incremental 13-63
iterations 13-42
lengths and widths 13-70
MODEL keyword 13-39
.MODEL statement 13-40
models 3-30
MOS
  LEVEL 13 13-48
  op-amp 13-70
network 13-60
RC 13-51
.PARAM statement 13-39
parameters 13-60
  magnitude and phase 13-61
  measured vs. calculated 13-61
.MODEL statement 13-41–13-43
results
  function evaluations 13-53
  iterations 13-54
  Marquadt scaling parameter 13-53
  norm of the gradient 13-53
  residual sum of squares 13-52
S parameters 13-60
simulation accuracy 13-36
simultaneous 13-56, 13-67, 13-70
statements 13-38
syntax 13-38
TDR
  packaging 25-9
  procedure 25-11
time
  analysis 11-5, 13-37
  required 13-36, 13-41
OPTIMIZE
  function 26-3
  keyword 10-15, 13-38
.OPTION 10-29, 10-31
  ACCT, summary of job statistics 8-16
  ALT999 or ALT9999, for output file name extension 8-14
  AUTOSTOP 26-74
CO, for printout width 8-14
DCAP 15-5, 16-2, 16-3
DCSTEP 10-47
DI 9-21, 9-39, 10-39, 11-16, 12-8
GMIN 15-5, 16-2
GMINDC 15-5, 16-2
GRAMP 16-2
INGOLD, for printout numerical format 8-15
keyword application table 9-3
MBYPAS 20-13
POST, for high resolution graphics 8-15
RISETIME 18-12, 18-13
SCALE 15-4, 15-14
SCALM 15-4, 15-14
SHRINK 15-14
statement 9-2, 10-31
.OALTERT blocks 3-41
options
  control, setting 15-4
  convergence 15-5
  EPSMIN 29-20
  FAQ A-27
  OPTLST 30-8
.OPTIONS SEARCH statement 3-36
OPTLST option 9-8, 30-8
  printing bisection results 30-8
OPTS option 9-9
OPTxxx parameter 13-37, 13-38
Opus 9-11
oscillation
  eliminating 9-49, 11-14
  from simulation errors C-61
oscillators
  behavioral models 26-65
  DELMAX option setting 11-26
  LC 26-83
  VCO 26-81
out (digital vector file) 5-76
outer sweep 3-25
output
  admittance 8-34
  buffer 19-8, 19-9
  example 19-42
  commands 8-2
conductance 20-54
current 8-22
data
  format 9-6, 9-7, 9-12
  limiting 9-12, 9-50, 11-12
  number format 9-6
  redirecting 3-61
  significant digits specification 9-8
  specifying 9-12, 9-13
  storing 9-12
driver example 31-12
ECL buffer 19-20
FAQ A-32
.FFT results 28-10
files 3-55
  names 3-53, 3-61
  redirecting 3-53
  reducing size of 9-12, 9-15
  version number, specifying 3-60
graphing 8-10
impedance 8-34
magnetic core 14-16
LX1 - LX7 14-16
measfail value 30-8
.MEASURE results 8-37
network 8-34
nodal voltage, AC 8-30
noise 8-35, 12-12
parameters 8-21
plotting 8-8–8-9
  specifying variables 8-10
power 8-25
printing 8-6–8-20
printout format 8-15
saving 8-9
statements 8-2
variables 8-3
  AC formats 8-32
  function 7-11
  printing 9-50, 11-13
  probing 8-10
  specifying significant digits for 9-8
  voltage 8-21
outz (digital vector file) 5-76
overlap capacitors 20-73
overlay dielectric transmission line C-21
overview
  data flow 1-9
  simulation process 1-10
.P
  .pa file 3-55, 3-57
  packed input files 3-2
  Pade approximation 29-40
  page eject, suppressing 9-8
PAR
  keyword 7-4, 7-8
  parameter, example 24-11
.PARAM statement 3-15–3-20, 3-37, 8-38, 13-2
  in .ALTER blocks 3-41
  optimization 13-39
parameters
  AC sweep 12-4
  ACM 11-27
  admittance (Y) 8-29
  AF 12-13
  algebraic 7-9
  expressions 7-8
  analysis 7-6
  AREA 15-6
  assignment 7-3
  base width 16-9
  BJT LEVEL 2 16-13
  capacitance 14-9
  capacitor
    junction 16-11
    metal and poly 15-13
  CAPOP 11-27
  cell geometry 7-13
  constants 7-3
data
  driven analysis 3-21
  type 7-2
DC
  model 16-5
  sweep 10-14
  defaults 7-17
  defining 7-2, 7-14
DIM2 8-35
DIM3 8-35
Ebers-Moll 16-5
encrypting E-1
evaluation order 7-3
extraction example 14-24
.FFT statement 28-7–28-8
frequency response table 29-9
HD2 8-35
HD3 8-35
hierarchical 3-48, 7-13, 8-38–8-39
high-current beta degradation 16-10
hybrid (H) 8-29
IC 10-9
impedance (Z) 8-29
inheritance 7-16, 7-17
INOISE 8-35
input netlist file 3-5
junction
capacitor 16-5
diode 15-6
setting
capacitance 15-12
DC LEVEL 1 and 3 15-9

KF 12-13
Laplace 29-9
libraries 7-14–7-17
limit checking
capacitor device 14-10
magnetic core 14-17
low-current beta degradation 16-9
M 3-48, 7-7
matrix 12-14
measurement 7-7
model
A2D 5-54
BJT LEVEL 2 16-13
D2A 5-51
Fowler-Norheim 15-48
Jiles-Atherton core 14-16
Juncap 15-35
junction 15-8
magnetic core 14-14
wire 14-4
modifying 3-21
multiply 7-7
names 3-31
noise 15-13, 20-99
BJTs 16-13
JFETs and MESFETs 17-35
ONOISE 8-35
optimization 7-13
OPTxxx 13-37, 13-38
output 8-21
overriding
assignments 7-15
default values 7-5
PAR keyword 7-4
PARHIER option 7-17
passing 7-13–7-19
example 24-11
order 7-3
problems 7-19
Release 95.1 and earlier 7-19
pole/zero 29-9
repeated 8-38
resistance 16-10
scattering (S) 8-29
scope 7-13–7-14, 7-19
SIM2 8-35
simple 7-3
simulator access 3-35
skew, assigning 3-35
subcircuit 3-48, 7-5
temperature, JFETs and MESFETs 17-38
time sweep 11-4
transient model 16-5
transit time 16-5, 16-12
transmission lines 18-3, 18-34
UIC 10-10
user-defined 7-4
UTRA 10-45
varying example 14-20
VBIC 16-57
voltage 20-50
See also model parameters, optimization
parameters
parametric analysis 8-4
parasitic
capacitance 16-12
BJTs 16-6
parameters 16-12
RC wire model 14-2
diode, MOSFETs LEVEL 39 21-198
generation 20-37
MOSFETs LEVEL 13 21-126
resistance
parameters, BJTs 16-10
temperature equations
BJTs 16-49
JFETs and MESFETs 17-44
PARHIER option 7-17
PARMIN optimization parameter 13-43
PASSFAIL keyword 30-8
passive device models 14-1
path names 9-9
subcircuits 3-18
path numbers, printing 9-9
PATHNUM option 9-9
PB model parameter 15-36, 20-28
PC, FAQ A-23
p-channel
JFETs models 3-30
MOSFET’s models 3-30
PCI 31-36
demo files 25-37
driver selection 25-42
Hspice file 25-37
modeling 25-33
simulation
eexample files 25-46
process 25-45
Speedway 25-35
circuit schematic 25-43
models D-15
subsections 25-43
PD model parameter 26-70
peak-to-peak value, measuring 8-46
period (digital vector file) 5-71
peripheral component interconnect
See PCI
permit.hsp file, encryption capability E-7
PG model parameter 15-37
PHA model parameter 15-28
phase
AC voltage 8-31–8-32
calculating 8-31
detector model 26-88, 26-97
locked loop, BJT model 26-92
PHD model parameter 20-28
Philips MOS9 model 22-56
PHOTO model parameter 13-22
PHP model parameter 20-28
PHS model parameter 20-28
piecewise linear sources See PWL
pin capacitance, plotting 24-6
pivot
algorithm, selecting 9-24, 9-26, 10-27
change message 9-25, 9-27, 10-28, 10-31
matrix calculations 9-23, 10-26
reference 9-25, 10-28
selection 11-24
PIVOT option 9-24, 9-26, 10-27, 11-24
PIVREF option 9-25, 10-28
PIVREL option 9-25, 10-29
PIVTOL option 9-24, 9-25, 9-27, 10-28, 10-29, 10-30
PJ capacitor, parameter 15-6
planar conductor parameters C-20
platforms for Hspice 1-6
PLEV model parameter C-7, C-13, C-14, C-36
PLIM option 9-9
PLL See phase locked loop
plot
limits 8-8
models 3-30
value calculation method 9-5, 9-39, 12-7
PLOT keyword 8-12
.PLOT statement 8-2
in .ALTER block 3-40
simulation results 8-8, 8-19
PMOS model
examples 22-55
pn junction conductance 10-54
pnp BJT models 3-30
POI keyword 10-16, 11-6, 12-6
POLE
element parameter 29-11
function 27-1, 29-2, 29-7, 29-28–29-33
call 29-28
highpass filter 29-31
low-pass filter 29-32
transconductance element statement 29-7
voltage gain element statement 29-7
poles and zeros, complex 29-30
pole/zero
analysis 27-1, 29-2
absolute tolerance 9-36, 10-32, 27-4
capacitance scale 27-4
conductance scale 27-4
example
active low-pass filter 27-15
CMOS differential amplifier 27-12
high-pass Butterworth filter 27-10
Kerwin’s circuit 27-8
low-pass filter 27-5
simple amplifier 27-14
frequency
maximum 9-35, 10-31
scale 27-4
imaginary to real ratio 27-5
inductance scale 27-4
iteration limit 27-4
maximum
frequency 27-4
number of iterations 9-42, 10-31
Muller algorithm 27-3
operating point 27-3
overview 27-2
.PZ statement 27-3
real to imaginary ratio 9-36, 10-32, 27-5
relative error tolerance 27-5
starting points, Muller algorithm 9-37, 10-33
conjugate pairs 29-8
function, Laplace transform 29-7
models 29-43
AC analysis 29-47
transient responses 29-46
parameters 29-9
transfer function 29-28
transient modeling 29-2
POLY 5-24
element parameter 26-7, 26-13, 26-20, 26-25, 26-29
polygon, defining 18-40
polynomial function 5-24, 26-7
FV function value 26-7
one-dimensional 5-25
three-dimensional 5-27
two-dimensional 5-26
polysilicon 20-17
POST option 1-9, 9-12
POST_VERSION option 9-9
pow(x,y) function 7-9
power
dissipation 8-25–8-29
subcircuits 8-25
function 7-10
operating point table 10-6
output 8-25
stored 8-25
POWER keyword 8-26
PP keyword 8-46, 8-47
PPWL
element parameter 26-20
function 5-36, 26-17
PMOS transistor function 26-10
PRD model parameter 20-29
precision numbers 7-8
print
CMI_PrintModel 23-31
control options 8-14
.PRINT statement 8-2
in .ALTER 3-40
simulation results 8-5, 8-19
printed circuit boards
models 25-2
printer, device specification 8-10
printout
columns, number 9-6
disabling 9-6, 9-8
suppressing 3-40
page ejects 9-8
value calculation method 9-5, 9-39, 12-7
PROBE option 9-12
.PROBE statement 8-2
simulation results 8-9, 8-19
program structure 1-7
.PROTECT statement 3-40, E-2
protected data 3-40
PRS model parameter 20-29
.PRTDEFAULT printer 8-10
PS model parameter 15-37
PSF option 9-12
PSRR specification 26-5
PULSE source function 5-7, 5-10, 5-13, 5-15
delay time 5-8
initial value 5-8
onset ramp duration 5-8
plateau value 5-8
recovery ramp duration 5-8
repetition period 5-8
width 5-8
pulse width 30-16
PUTMEAS option 8-38, 9-51
PWL
current controlled gates 26-6, 26-7
data driven 5-18
element parameter 26-13, 26-20, 26-25, 26-29
functions 5-28, 26-7, 26-10
gates 26-6
NMOS and PMOS transistor functions 26-10
output values 5-16
parameters 5-15
repeat parameter 5-16
segment time values 5-16
simulation time 11-36
sources, data driven 5-18
voltage controlled
   capacitors 26-6
gates 26-6
See also data driven PWL source
PWR model parameter 26-70
pwr(x,y) function 7-10
.PZ statement 10-19, 27-3, 29-2
PZABS option 9-36, 10-32, 27-4
PZTOL option 9-36, 10-32, 27-5
Q
quality assurance 13-2

quasi-saturation BJT model 16-50
quasi-static model equations 22-103

R

RA model parameter C-26
RA1 model parameter C-28
RAC model parameter 26-71
radix (digital vector file) 5-68
random limit parameter distribution 13-14
RB model parameter C-26
RC
   line model 29-33
   network
      analysis
         AC 2-2
         transient 2-5
circuit 2-2
   optimization 13-51
   wire model 14-2
cells, reusing 7-14
RD model parameter 20-29, C-26
RDC model parameter 20-29
real part of AC voltage 8-31
real vs. imaginary component ratio 9-36, 10-32
RECT FFT analysis keyword 28-8
rectangle, defining 18-38
rectangular FFT window 28-4
reference
   IBIS 19-49
   plane, transmission lines C-5
temperature 3-20, 13-5
regions charge equations, MOSFETs LEVEL 13 21-118
related documents iii
RELH option 9-22, 9-40, 10-40, 11-18, 12-8
RELI option 9-22, 9-40, 10-35, 10-41, 11-18
KCLTEST setting 9-21, 10-26
RELIN optimization parameter 13-43
RELMOS option 9-20, 9-22, 10-35, 10-37, 10-41, 11-27
KCLTEST setting 9-21, 10-26
RELOUT optimization parameter 13-43
RELQ option 9-40, 11-18, 11-34
RELTO option 9-39, 9-40, 11-16, 11-18
RELV option 9-22, 9-40, 9-42, 10-35, 10-41, 11-17, 11-18
multiplier for 9-43, 11-17
RELVAR option 9-46, 11-18, 11-27
RELVDC option 9-22, 9-23, 10-41
repeat function 31-3
residual sum of squares 13-52
Resistance
AC 12-3
matrices 18-4
minimum 9-35, 10-29
MOSFETs model parameters 20-29
parameters 16-10
Resistor
BJTs 16-6
current flow 8-23
device model 14-2
element 4-2
element template listings 8-55
length parameter 4-3
model
equations 14-5
name 4-2
node to bulk capacitance 4-3
noise 14-7
equation 14-7
temperature 14-8
equations 14-8
voltage controlled 5-35
width parameter 4-3
wire model parameters 14-5
RESMIN option 9-35, 10-29
RESULTS keyword 10-15
RHO model parameter C-16, C-25, C-26, C-28
RHOB model parameter C-16, C-25, C-26, C-29
RIN keyword 12-15
ripple calculation 24-3
rise and fall times 8-39
RISE keyword 8-41
rise time C-79
simulation example 24-2, 24-9
RISETIME option 9-40, 18-12, 18-13
RITOL option 9-36, 10-32, 27-5
RL model parameter 20-29
RLGC
element syntax 18-27
model syntax 18-28
RLOAD model parameter 5-54
RMAX option 9-46, 11-23, 11-35
RMIN option 9-46, 11-23, 11-35
RMS keyword 8-47
rms value, measuring 8-46
roac model parameter 26-71
ROUT keyword 12-15
ROUT model parameter 26-71
row/matrix ratio 9-25, 10-29
RPI
a-Si TFT model 22-175
circuit 22-178
parameters 22-176
Poli-Si model
parameters 22-184
Poli-Si TFT model 22-182
circuit 22-186
RS model parameter 20-29
RSC model parameter 20-29
RSH model parameter 13-9, 20-29
runtime statistics 9-5
S
S Element 6-2
Frequency Table Model 6-4
syntax 6-3
transmission line 6-8
S parameter
example 6-8
model type 3-30
S Element 6-2
transmission line 6-10
S19NAME model parameter 5-55
S19VHI model parameter 5-55
S19VLO model parameter 5-55
S1NAME model parameter 5-54
S1VHI model parameter 5-55
S1VLO model parameter 5-55
SIVLO model parameter 5-55
.SAMPLE statement 12-13
sampling noise 12-13
saturable core
  elements 8-65
    names 4-10
    winding names 4-10
  models 14-12
    names 4-10
    winding names 8-65
saturation
  carrier velocity 21-83
  current temperature equations
    JFETs and MESFETs 17-41
    MOSFETs 20-6
  temperature equations, BJTs 16-42
  voltage (vdsat), BSIM LEVEL 13 21-115
voltage equations, MOSFETs
  LEVEL 1 21-6
  LEVEL 13 21-115
  LEVEL 2 21-14
  LEVEL 28 21-158
  LEVEL 3 21-25
  LEVEL 38 21-173
  LEVEL 47 22-12
  LEVEL 5 21-47
  LEVEL 6 21-65, 21-70
  LEVEL 8 21-95
.SAVE statement 10-11
.SAVE statement parameter 10-13
SCALE 15-4
  element parameter 26-13, 26-20, 26-25, 26-29, 29-11
  option 14-5, 14-10, 15-4, 15-14, 31-7
  parameter 29-37
scale
  diode parameters 15-15
  factor 4-3
  JFETs and MESFETs 17-6
  parameters 17-7
SCALE option 20-11, 20-12
scaling 20-12
  BJTs 16-20
  diode model 15-14
  effect on delays 31-23
global SCALM override 20-12
global vs model 15-4, 20-12
  JFETs 17-6
  MOSFETs capacitance parameters 20-68
  options 15-4
SCALM 15-4
  option 9-18, 14-5, 14-10, 15-4, 15-14, 20-11
  JFETs and MESFETs scaling 17-6
  parameter
    global scaling 20-12
    in a diode model statement 15-4
    overriding in a model 20-12
    scaling by model 20-12
scattering (S) parameters 8-29
  See also S parameter 6-2
schematic
  AvanLink B-5
  AvanLink-DA B-10
  netlists 3-6
Schichman-Hodges model 20-4, 21-2, 21-7
  equations 21-11
  Schmitt trigger example 10-18
  Schottky barrier diodes 15-1
  scope of parameters 7-14
  scratch files 1-12
  SDA option 9-12
  s-domain 29-20
    equivalent circuit 29-25
SEARCH option 3-36, 3-52, 9-9, 31-20
  search path, setting 3-37
SEED option 9-15
self inductance C-77
.SENS statement 10-19
sensitivity
  analysis 10-19
  factors, MOSFETs LEVEL 28 21-154
setup time 30-4
  analysis 30-6, 30-10
SFFM source function
  carrier frequency 5-20
  modulation index 5-20
  output amplitude 5-20
  output offset 5-20
  signal frequency 5-20
sgn(x) function 7-10
SGS-Thomson MOS model 20-6
shape
circle 18-39
defining 18-38
polygon 18-41
rectangle 18-38
strips 18-40
Sharp model 20-6
short-channel effect 21-13
shorted nodes 10-47
SHRINK
model parameter 14-5, 14-10
option 15-14
SHTHK model parameter C-26, C-29
Siemens model 20-5, 20-6
Sierra 1 model 20-5
Sierra 2 model 20-6
SIG
conductivity C-89
model parameter C-16, C-25, C-26, C-28
SIG1 model parameter C-25
sigma model parameter, sweeping 24-4
sign
function 7-10
transfer function 7-10
(x,y) function 7-10
signal integrity
problems 25-3
SIGNAME element parameter 5-54
signed power function 7-10
Signetics
device models D-15
FAST I/O drivers 25-18
silicon
controlled rectifier, behavioral 26-59
gate transistor 20-17
Siliconix model 20-5
silicon-on-sapphire devices 20-3
bulk node name 3-19
SIM2 21-179
distortion measure 12-9
parameter 8-35
Simpson Integration 20-86
simulation
accuracy 9-38, 9-48, 11-13, 11-16, 11-25, 13-36, 21-4
improvement 9-44, 11-12
models 11-27
option 11-28
defaults 11-35
reduced by BYPASS 9-42, 11-11
timestep 11-26
tolerances 10-34, 10-35
AvanLink B-7
AvanLink-DA B-12
circuits
with Signetics drivers 25-18
with Xilinx FPGAs 25-22
electrical measurements 31-20
example F-1
graphical output F-10
ground bounce example 25-24
interconnect C-10
multiple analyses, .ALTER statement 3-40
multiple runs 3-45
PCI 25-45
example 25-46
performance, multithreading 3-63
preparation 25-2
process, overview 1-10
results
graphing 8-10
output variables 9-12
plotting 8-8–8-9
printing 8-6–8-20
specifying 8-37–8-39
storing 9-12
speed 9-8, 11-25
structure 1-7
time 21-4
ABSVAR option 11-35
with TRTOL 9-41, 11-19
RELVAR option 11-35
title 3-9
SIN source function 5-11
sin(x) function 7-9
single point experiment 1-8
single-frequency FM source function 5-20
sinh(x) function 7-9
sinusoidal source function 5-10
skew
   file 13-12
   parameters 13-8
   assigning 3-35
Skin 18-13
skin effect
   imaginary term handling 18-13
skin effect frequency C-18
slope (digital vector file) 5-74
SLOPETOL option 9-47, 11-19
   simulation time 11-36
   timestep control 11-35
small-signal
   DC sensitivity 10-19
   distortion analysis 12-9
   transfer function 10-20
SMOOTH element parameter 26-20
SOI model 21-142
SONAME model parameter 5-54
SOSFET’s model 20-3, 20-6
   LEVEL 27 21-136
source
   AC sweep 12-4
   controlled 26-4, 26-6
   data driven 5-18
   DC sweep 10-14
   element types 26-6
   keywords 3-2
   Laplace transforms 29-21
   statements 3-10
   time sweep 11-4
   See also independent sources
SOVHI model parameter 5-54
SOVLO model parameter 5-54
   .sp file encryption E-10
SP model parameter C-15, C-24
SP12 model parameter C-24
SPARSE option 9-26, 10-30
spatial extent of leading edge C-83
spectral leakage 28-3, 28-16
spectrum analysis 28-1
SPICE
   compatibility 9-14, 17-22
   AC
      output calculations 9-5, 9-39, 12-7
      output variables 8-30–8-32
   BSIM model 21-123
diodes 20-37
   models 21-218
   MOSFETs
      LEVEL 13 parameters 21-123
      LEVEL 3 21-29
      LEVEL 39 21-182, 21-191
      models 20-4
      numeric format 9-6
      plot
      limits 8-8
      size 9-9
   UTRA model parameter 21-11
depletion capacitor model 17-5
   Meyer gate capacitances 20-74
   model parameters, MOSFETs LEVEL 39 21-182
   option 9-14
sqrt(x) function 7-9
square root function 7-9
SRN model parameter 26-71
SRNEG model parameter 26-71
SRP model parameter 26-71
SRPOS model parameter 26-71
   .st# file 3-55, 3-57
   stacked devices 20-41
   Star-Hspice $\Delta L$ equation 21-85
START .FFT parameter 28-7
START keyword 11-6
statements
   .AC 12-4, 13-5, 13-39
   .ALTER 3-40
   call
   subcircuit 3-14
   .DATA 3-21, 24-9
   external file 3-25, 3-27
inline 3-23
  inner sweep example 3-24
  outer sweep example 3-25
  DC 10-14, 13-5, 13-39
  DCVOLT 10-9, 10-10
  .DEL LIB 3-42
  .DISTO 12-9
  element 3-10
  .END 3-45
  .ENDL 3-32, 3-33
  .ENDS 3-14
  .EOM 3-14
  .FFT 11-41, 28-7
  .FOUR 11-38
  .GLOBAL 3-20
  .GRAPH 8-2, 8-11, 8-19
  .IC 10-9, 10-10
  .INCLUDE 3-29
  initial conditions 10-9
  .LIB 3-32, 3-33
    call 3-32
    nesting 3-33
  .LOAD 10-11, 10-13
  .MACRO 3-12
  .MEASURE 8-2, 8-3, 8-37, 9-7, 9-12
  .MODEL 3-30, 8-12, 13-5, 13-41
  model 14-1
    BJT's 16-4
    junction 15-7
  .NET 12-14
  .NODESET 9-29, 10-10, 10-23
  .NOISE 12-11
  .OP 10-6
  .OPTION 9-2, 10-31
    ACCT 8-16
    ALT999(9) 8-14
    CO 8-14
    INGOLD 8-15
    POST 8-15
  .OPTIONS SEARCH 3-36
  .PARAM 3-37, 13-39
  .PLOT 8-2, 8-8, 8-19
  .PRINT 8-2, 8-5, 8-19
  .PROBE 8-2, 8-10, 8-19
  .PROTECT 3-40
  .PZ 10-21, 27-3
  .SAMPLE 12-13
  .SAVE 10-11, 10-12
  .SENS 10-19
  SHAPE 18-38
  source 3-10
  .SUBCKT 3-12, 8-38, 26-65
  .TEMP 3-20, 13-5, 13-6
  .TF 10-20
  .TITLE 3-9
  .TRAN 11-4, 13-5, 13-39
  .UNPROTECT 3-40
  .WIDTH 8-14
  statistical analysis 13-8–13-34
  statistics
    calculations 13-3
    listing 8-16
    report 9-5
  Statz model 17-2, 17-3, 17-22
    capacitance equations 17-28
  STC-ITT model 20-5
    stimulus input files 26-4
  STOP .FFT parameter 28-7
  strip, defining 18-40
  stripline transmission line C-21
  structure simulation 1-7
  subcircuits
    adder 31-3
    BJT's 16-55
    call statement 3-14
    calling 3-14
    calling tree 3-18
    changing in .ALTER blocks 3-41
    creating reusable circuits 3-47
    element names 3-14
    generator 26-5
    global versus local nodes 3-20
    hierarchical parameters 3-48
    library structure 3-52
    model names 3-14
    multiply parameter 3-15
    multiplying 3-48
    names 3-13
    node names 3-14, 3-17
    abbreviated 3-18
    node numbers 3-13
output printing 8-19
parameter 3-13
   example 24-11
   passing 3-14, 3-15
path names 3-18
power dissipation computation 8-25
   .PRINT and .PLOT statements 3-49
printing path numbers 9-9
search order 3-49
test example 3-13
zero prefix 3-19
.SUBCKT statement 3-12, 8-38
   search order 26-65
SUBS model parameter 16-2
substrate
   capacitance equations 16-39
current equations 16-31
diodes 16-19
subthreshold current equations, MOSFETs
   LEVEL 13 21-115
   LEVEL 2 21-17
   LEVEL 3 21-28
   LEVEL 38 21-175
   LEVEL 5 21-48
   LEVEL 6 21-73
   LEVEL 8 21-99
surface potential equations 20-109
.sw# file 3-55, 3-56
sweep
   data 3-25
   storing 9-12
   frequency 12-6
   inner 3-25
   outer 3-25
   variables 31-18
Sweep keyword 10-16, 11-6, 12-6
switch example 5-39
switched capacitor 29-48
   filter example 29-50
switch-level MOSFET’s example 5-39
Synopsys models, calibrating 24-1
systems, simulating 25-2

T

T Elements C-2
   lossless transmission line model C-88
   multiplier function M C-5
   transient effects modeling C-75
tabular data 5-61, 5-64
Taguchi analysis 13-2
tan(x) function 7-9
tanh(x) function 7-9
TARG keyword 8-40
target specification 8-40
Taylor-Huang model 20-5
TC1, TC2 element parameters 26-13, 26-20, 26-25,
   26-29, 29-11
TD
   element parameter 26-13, 26-20, 26-25, 26-29
   keyword 8-33, 8-43
tdelay (digital vector file) 5-73
TDR
   optimization 25-10
   packaging 25-9
   procedure 25-11
time domain reflectometry 25-9
TEM transmission lines 18-3
TEMP
   keyword 10-16, 12-6
   model parameter 3-20, 13-5, 26-71
   sweep variable 31-18
   .TEMP statement 3-20–3-21, 13-5, 13-6–13-7
temper variable 7-11
temperature
   AC sweep 12-4
BJTs
   beta equations 16-42–16-46
capacitance equations 16-46
   energy gap equations 16-42
   LEVEL 2 equations 16-49
   parameters 16-14
   parasitic resistor equations 16-49
   saturation equations 16-42–16-46
capacitor equations 14-8
circuit 13-4, 13-6
default 13-5
coefficients 4-2
demo 31-18
G and E Elements 29-11
compensation
equations 15-28
MOSFETs LEVEL 47 22-17
example 21-30
MOSFETs BSIM LEVEL 13 21-127
DC sweep 10-14, 10-17
derating 3-20, 3-21, 13-5, 13-7
diodes 15-19
effect
BSIM LEVEL 13 21-116
MOSFETs LEVEL 39 21-196
parameters
BJTs 16-14
junction diodes 15-19
element 13-5, 13-6
equations
BJTs 16-42
LEVEL 2 16-49
breakdown voltage 15-29
capacitance 15-31
contact potential 15-30
energy gap 15-28
FJET’s and MESFETs 17-38
grading coefficient 15-32
leakage 15-29
MOSFETs 20-105
resistance 15-32
resistor 14-8
transit time 15-29
inductor 14-17
JFETs
equations 17-38, 17-41–17-44
TLEV parameter 17-38
TLEV parameter 17-38
junction diodes 15-19
MESFETs equations 17-38, 17-41–17-44
MOSFETs
channel length modulation 20-111
diode
capacitance 20-106
resistance 20-111
equations 20-105
mobility 20-111
parameters 20-102
surface potential 20-109
threshold voltage 20-110
optimizing coefficients 31-18
parameters 17-38
JFETs 17-38
JFETs and MESFETs 17-39
MOSFETs 20-102
LEVEL 13 21-109
LEVEL 28 21-154
LEVEL 49 22-49
LEVEL 55 22-90
LEVEL 57 22-126
LEVELs 49 and 53 22-41
reference 3-20, 13-5
model parameters 14-5
resistor equations 14-8
sweeping 31-18
time sweep 11-4
variable 7-11
Temperature Variation Analysis 13-2
Texas Instruments device models D-16
.TF statement 10-19, 26-46
tfall (digital vector file) 5-75
TH model parameter C-15, C-24
TH1 model parameter C-24
THB model parameter C-15, C-24
THD (total harmonic distortion) 28-1
The 3-40
thermal noise 14-7
THK1 model parameter C-24
THK2 model parameter C-24
three-dimensional function 5-27
threshold
temperature equations, JFETs and MESFETs 17-44
voltage
BSIM LEVEL 13 21-114
equations, MOSFETs 20-50
LEVEL 1 21-6
LEVEL 13 21-114
LEVEL 2 21-13
LEVEL 28 21-156
LEVEL 3 21-25
temperature 20-110
parameters, MOSFETs 20-50
LEVEL 1 21-4
LEVEL 2 21-9–21-10
LEVEL 3 21-21–21-22
LEVEL 8 21-91–21-92
LEVEL 38 21-165
LEVEL 5 21-35
LEVEL 6, 7 21-56–21-57

time 10-7
delay 8-33
domain
algorithm 11-30
reflectometry 25-9
to frequency domain 29-1
maximum 29-6
resolution 29-5
sweep 11-4
variable 7-11
versus impedance 25-9
See also CPU time

TimpMill models, calibrating 24-1
TIMES option 9-47, 11-19
TIMESCALE model parameter 5-55

timestep
algorithms 9-44, 11-12, 11-33
calculation for DVDT=3 9-44, 11-21
changing size 9-40, 11-18
control 9-41, 9-44, 9-46, 11-18, 11-19, 11-21, 18-11
algorithms 11-32–11-35
DELMAX 11-35
error C-61
FS 11-35
FT 11-35
minimum internal timestep 11-36
Minimum Timestep Coefficient 11-36
options 11-26, 11-35

CHGTOL 11-34
IMAX 11-33
IMIN 11-33
RELQ 11-34
TRTOL 11-34
RMAX 11-35
RMIN 11-35
TSTEP 11-36
control error C-61
default control algorithm 11-29
DVDT algorithm 11-34
internal 9-43, 11-20
local truncation error algorithm 11-34
maximum 9-45, 9-46, 9-47, 11-21, 11-22, 11-23
minimum 9-45, 9-46, 9-48, 11-22, 11-23
reversal 11-34
definition 9-43, 11-15
setting initial 9-43, 11-20
transient analysis algorithm 9-48, 11-13
variation by HSPICE 9-43, 11-20

TIMESTEP model parameter 5-55
timing
analysis 30-1
constraints 30-1
failures 30-2
hold time 30-4
setup time 30-4
violation analysis 30-2
title for simulation 3-9
.TITLE statement 3-9
tmp directory 1-12
TMPDIR environment variable 1-12
TNOM option 3-20, 13-5
TO .FFT parameter 28-7
TO keyword 8-46, 8-52
TOL 12-13
tolerance options 10-22
TOM model 17-45
LEVEL 3 parameters 17-45
parameters 17-46
See also TriQuint model
TOP keyword 10-13
topology
check, suppressing 9-8
supported 23-40
TOX model parameter 13-9
TR model parameter 15-35
.tr# file 3-55, 3-56
.TRAN statement 13-5, 13-39
transcapacitance
transconductance
FREQ function 29-8
JFETs and MESFETs 17-8
LAPLACE function 29-6
MOSFETs 20-22
POLE function 29-7
transfer function 10-20
algebraic 26-3
analysis 26-46
coefficients 29-30
filters 29-31
frequency domain 27-2, 29-20
general form 29-20, 29-29
inverse Laplace transform 29-33
poles 27-2
reduced form 29-30
roots 27-2
voltage gain 29-24
zeros 27-2
transfer sign function 7-10
TRANSFORMER element parameter 26-14
transformer, behavioral 26-57
transforms, Fourier 29-5
transient
analysis 8-3
accuracy 9-38
Fourier analysis 11-38
initial conditions 10-9, 11-3
inverter 2-7
MOSFETs, equivalent circuit 20-23
number of iterations 9-46, 11-23
RC network 2-5
sources 5-5
lateral 16-22
modeling 29-2
output variables 8-21
vertical 16-23
transistors

BJTs
AC analysis 16-23, 16-24
AC noise analysis 16-24, 16-25
transient analysis 16-22, 16-23
field effect 20-16
isoplanar silicon gate 20-17
lateral 16-23, 16-24, 16-28
ground 16-28, 16-51
process parameters, MOSFETs
LEVEL 13 21-102–21-106
LEVEL 28 21-147
LEVEL 49 22-34
substrate diodes 16-19
ground 16-28
vertical 16-24, 16-25, 16-28
ground 16-28, 16-50
VBIC 16-57
transit time
BJTs 16-6
parameters 16-12
transition files 26-4
Transmission line
S Element 6-8
transmission line
resistive termination 6-8
transmission lines
AC resistance C-19
analysis guidelines C-83
attenuation C-88
calculations C-85
effects C-94
bandwidth C-79
branch capacitances C-33, C-34
capacitance
definitions C-32
matrix C-52
to ground C-18
CEXT parameter C-32
characteristic impedance C-70, C-73
Cjk symbol C-32
clock frequency C-79
corax
example C-57
models C-25
common ground inductance C-75
conductance definitions C-32
conductor
  length C-32
  resistance C-93
  width C-32
coupled
  lines C-11
  microstrips example C-59
crosstalk C-75, C-78
CX symbol C-32
CXY symbol C-32
delay too small warning C-5
delays C-37
dielectric loss C-90
dispersion C-88
dissipation C-91
  factor C-89
dual dielectric C-21
effects 25-7
elements C-2
  selection C-4
equations 18-3
example 19-42, 31-12, C-8
geometric parameters C-17
Gjk symbol C-32
GPR parameter C-32
ground plane C-5
GX symbol C-32
GXY symbol C-32
HSPICE
  ground C-18
  reference plane C-5
ideal C-10
  properties C-6
impedance C-72
  characteristic C-73
  lumped elements C-72
  mismatch C-66
inductance C-75
  matrix C-52
  mutual C-76
  self C-77
integration
  algorithms C-43
  method C-61
interconnect properties C-3
layer stacks 18-37
LLEV parameter C-32
loss
  calculations C-85
  factor C-37
lossless
  model C-70
lossy
  model C-12, C-71, C-91
  properties C-6
  selector C-13
  statement C-13
U Element 4-25
lumped
  parameters C-19
RC model C-6
RLC model C-71
matrixes C-33, C-34, C-52–C-54
measured parameters C-35
microstrip C-21
models C-2, D-17
  selection C-4
multiple
  conductors C-32
  stripline, example C-39
mutual inductance C-76
number of lumps C-46, C-61
  computation C-62
  default C-62
  specification C-62
oscillation problems C-60
overlay dielectric C-21
parameter 18-3
  error C-17
  extracting 18-34
  physical C-15
planar
  conductors C-20
  models C-14
precomputed parameters C-30
  keywords C-31
printed circuit boards
  example C-56
  propagating a voltage step 18-6
  properties C-1
reflection calculations C-84
ringing C-20, C-60–C-65
time C-79
sea of dielectric C-21
example C-44
self inductance C-77
SIG conductivity parameter C-89
signal reflections C-66
simulation 25-2
skin depth C-89
calculation C-85
skin effect frequency C-18
source properties C-3
stripline C-21
switching noise C-75
system model 18-7
theory C-70
timestep control errors C-61
total conductor resistance C-51
transient impedance effects C-75
twinlead
example C-58
models C-27, C-58
U Element C-71
using 18-1
W Element input properties 18-14
wave propagation 18-5
transverse electromagnetic (TEM) waves 18-3
TRAP algorithm
See trapezoidal integration
trapezoidal integration
algorithm 9-48, 11-13, 11-29
coefficient 9-49, 11-17, 11-20
TREF model parameter 13-5, 13-6
TRIG keyword 8-40
trigger specification 8-40
triode tube
behavioral 26-60
example 5-41
TriQuint model 17-2
extensions 17-45
LEVEL 3 parameters 17-45
See also TOM model
trise (digital vector file) 5-75
tristate
buffer 19-11, 19-12
ECL buffer 19-21
triz (digital vector file) 5-76
troubleshooting A-1
TRTOL option 9-41, 11-19, 11-34
truncation algorithm 11-33
TS model parameter C-15, C-24
TS1 model parameter C-28
TS2 model parameter C-29
TS3 model parameter C-29
tskip (digital vector file) 5-71
TSTEP
multiplier 9-46, 11-23
setting 9-46, 11-23
option 9-46, 11-23
timestep control 11-36
TT model parameter 20-28
TTL
circuits 16-2
devices 16-2
tunit (digital vector file) 5-70
wave propagation 18-5
tunnel diode behavioral 26-58
twinlead
models C-58
twinlead.See transmission lines
two-dimensional function 5-26
U
U Elements 5-50, C-2
attributes effects C-94
digital input 5-50
examples C-38
lossy transmission line model C-71, C-88
multiplier function M C-5
parameters C-50
physical C-14
ringing problems C-60
statement C-11
transient effects modeling C-75
transmission line
loss parameters C-16
model 3-30, C-78
U models
applications C-55
conductor
  height C-14
  resistivity C-16
  spacing C-15
  thickness C-15
  width C-14
dielectric
  conductivity C-16
  configurations C-14
  constant C-15
electrical specification format C-13
ELEV parameter C-7, C-20, C-24
  external capacitance C-15
model selector C-13, C-14
  .MODEL statement C-13
number of
  conductors C-14
  layers C-16
  lumps C-15, C-19
parameters C-13, C-14–C-38
  geometric coax C-25
  geometric twinlead C-27
perturbation
  multiplier C-15
  parameter C-15
PLEV parameter C-7
reference plane
  configuration C-13
  height C-15
  resistivity C-16
  separation distance C-15
  thickness C-15
selection C-7
statement C-12
syntax C-13
transmission line type selector C-13
UIC
  analysis parameter 10-5
  keyword 11-7
  parameter 10-10
  transient analysis parameter 11-3
UNIF keyword 13-16
  uniform parameter distribution 13-14
unity gain frequency 31-20
Universal Field mobility reduction 21-79
University of California, Berkeley SOI model
  (BSIM3-SOI) 22-114
University of Florida SOI model 22-133
UNORM FFT analysis keyword 28-7
unprintable characters in input 3-3
  .UNPROTECT statement 3-40, E-2
UNWRAP option 9-51, 12-8
UPDATE parameter 21-128
  MOSFETs LEVEL 13 21-128
  MOSFETs LEVELs 6, 7 21-57
URL for Avant! vi
UTRA model parameter restriction 10-45
variables
  AC formats 8-32
  changing in .ALTER blocks 3-41
  DEFAULT_INCLUDE 3-54
Hspice-specific 7-11
  output 8-3
    AC 8-29
    AC analysis 8-29
    DC 8-21
    transient 8-21
  plotting 31-7
  sweeping 31-18
  TMPDIR 1-12
variance, statistical 13-3
VB model parameter 15-36
VBIC
  model (vertical bipolar inter-company) 16-57
  noise analysis 16-58
  parameters 16-57
VCC model parameter 26-71
VCCAP 5-35
  element parameter 26-20
  syntax 5-35
VCCS See voltage controlled current source
VCO See voltage controlled oscillator
VCR See voltage controlled resistor
VCVS See voltage controlled voltage source
VDBR model parameter 15-36
VDGR model parameter 15-36
VDSR model parameter 15-36
vector patterns 5-61, 5-67
VEE model parameter 26-72
vendor libraries 3-51
VERIFY option 9-7, 9-10
Verilog
  models, calibrating 24-1
  value format 5-66
version
  95.3 compatibility 11-35
  H9007 compatibility 9-15
  HSPICE 3-31
  specifying 3-59
VERSION parameter 21-111
VFLOOR option 9-28, 11-23
VHDL models, calibrating 24-1
Viewlogic 5-50
  graph data file 9-11
  links B-13
Viewsim simulator 5-50
vih (digital vector file) 5-77
vil (digital vector file) 5-77
vname (digital vector file) 5-68
VNDS model parameter 20-27
Vnn node name in CSOS 3-19
VNTOL option 9-41, 9-42, 11-16, 11-17, 11-20
voh (digital vector file) 5-79
vol (digital vector file) 5-79
VOL keyword 5-33, 26-15
voltage
  controlled
    capacitor 5-35, 5-40, 26-17
    current 26-18
    example 5-40
    current source 5-34, 5-46, 8-56, 26-6, 26-16
    element parameter 26-20
    element template listings 8-56
    syntax 5-34
    elements 26-6
    oscillator 5-33, 26-15, 26-81
    example 26-15
    model 26-81
    resistor 5-35, 26-6, 26-16
    element parameter 26-21
  sources 5-24
  voltage source 5-29, 8-56, 26-6, 26-11
  element parameter 26-14
  inverting comparator application 26-79
  syntax 5-29
error tolerance
  DC analysis 9-22, 9-23, 10-41
  transient analysis 9-40, 11-18
failure 10-49
gain
  FREQ function 29-9
  LAPLACE function 29-6
  POLE function 29-7
initial conditions 10-9
iteration-to-iteration change 9-32, 10-24
JFETs and MESFETs DC models 17-31
limiting, in transient analysis 9-47, 11-21
maximum change 9-43, 11-15
minimum
  DC analysis 9-21, 10-38
  listing 9-28, 11-23
  transient analysis 9-38, 9-41, 11-20
MOSFETs model parameters 20-50
nodal output DC 8-21
operating point table 10-7
relative change, setting 9-46, 11-18
sources 5-29, 5-42
  current output 8-21
  summer 5-32, 26-14
tolerance
  BYTOL option 9-42, 11-16
  MBYPASS multiplier 9-43, 11-20
value for BYPASS 9-42, 11-16
variable capacitance model 26-84
VOLTAGE keyword 10-7
VON model parameter 26-72
VONEG model parameter 26-72
VOP model parameter 26-72
VOPOS model parameter 26-72
VOS model parameter 26-72
Vos specification 26-5
VR model parameter 15-35
vref (digital vector file) 5-78
VREL model parameter C-36
vth (digital vector file) 5-78
VTI model 20-6
VVVCAP model 26-84
Vxxx source element statement 5-2

W capacitor parameter 15-6
W Element
   accuracy update 18-12
   FAQ A-34
   RLG
      model 18-27, 18-28
      model definition 6-11
      skin effect 18-13
      transmission line
         properties inputs 18-14
W Elements 4-21, 18-10
   field solver model 18-24
   RLG
      file 18-24
      model 18-16
      transmission line
      model 3-30
U models 18-20
Wang’s equation 21-84
warnings 19-46
   all nodes connected together 10-47
   capacitance too large 14-10, 14-17
   floating power supply nodes 3-17
   invalid value for CRATIO 14-3
   IS parameter too small 15-10
   limiting repetitions 9-15
   MAXF ignored 29-11
   misuse of VERSION parameter 3-31
   resistance smaller than RESMIN 14-6
   suppressing 9-15
   T-line delay too small 11-5
   zero diagonal value detected 10-49
WARNLIMIT option 9-15
wave propagation, transmission lines 18-5
waveform
   characteristics 5-61
   modifying 5-73
   display

AvanWaves B-7, B-12
FAQ A-35
WD model parameter C-14, C-24
WD1 model parameter C-24
WDAC parameter 20-98
WDEL model parameter 20-30
WEIGHT keyword 8-47, 8-52
WHEN keyword 8-43, 31-20
WIC weak inversion current selector 21-73
width of printout 8-14
.WIDTH statement
   print control options 8-14
WINDOW FFT parameter 28-8
Windows (MS), FAQ A-23
wire C-2
   capacitance 14-6
   critical length 25-7
   model
      capacitance
         calculation 14-6
         distribution 14-2
      effective length and width 14-5
      parameters 14-2, 14-5
      resistance calculation 14-5
      selection C-2, C-69
         chart C-4
      model RC 14-2
      resistance 14-5
      types 25-7
   See also transmission lines
WL option 9-18, 20-11
WLUMP model parameter C-15
WM capacitor parameter 15-6
WMAX model parameter 1-5
WMIN model parameter 1-5
WMLT model parameter 20-30
worst case analysis 13-8–13-34
   example 13-26, 24-9
Worst Case Corners Analysis 13-2
WP capacitor parameter 15-6
WRD model parameter 20-29
WRS model parameter 20-29
WSF output data 9-11
X

XGRID model parameter 8-13
Xilinx device models D-17
XJ model parameter 20-30
XL model parameter 13-9
XMAX model parameter 8-13
XMIN model parameter 8-13
XMU option 9-49, 11-17, 11-20
XnR, XnI option 9-37, 10-33
XPART CAPOP model parameter 20-72
XPHOTO model parameter 13-22
XQC CAPOP model parameter 20-72
XSCALE model parameter 8-13
XW model parameter 13-9, 20-30, C-15, C-24

Y

Y parameter
  line model 29-43
  modeling 29-39
  network 29-40
Y parameters 6-2
YGRID model parameter 8-13
yield analysis 13-2
YIN keyword 8-34, 12-16
YMAX
  keyword 8-52
  model parameter 8-13
YMIN
  keyword 8-52
  model parameter 8-13
YOUT keyword 8-34, 12-16
YSCALE model parameter 8-13

Z

Zener diodes 15-4
zero delay
  gate example 5-40
  inverter gate 5-33

www site for Avant! vi

ZIN keyword 8-34, 12-16
ZK model parameter C-36
ZOUT keyword 8-34, 12-16
ZUKEN option 9-13