Avant!

Star-Hspice Manual

Release 2000.4
December 2000

Avant!

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 III 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:

Convention	Description
menuName > commandName	Indicates the name of the menu and the command name. For example: *Cell > Open* refers to the *Open* command in the Cell menu.
Tool: menuName > commandName	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: *Data Prep: Pin Solution > Via* refers to the *Via* command on the Pin Solution menu, which you access by selecting *Data Prep* from the Tools menu in Apollo.
`courier`	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.

Convention	Description
`courier italic`	Arguments appear in this font when the value of an argument is a string. The string must be enclosed with quotation marks.
times italic	Indicates commands, functions, arguments, file names, and variables within a line of text.
	When a variable is included in italicized text, the variable is enclosed by angle brackets (<>). For example, "the name of the technology file is <libraryName>.tf, where <libraryName> is the name of the library."
[ ]	Denotes optional arguments, such as: *pin1* [*pin2*, ...*pinN*] In this example, you must enter at least one pin name, the other arguments are optional.
({*instanceName orientation*} ...)	Indicates that you can repeat the construction enclosed in braces.
. . .	Indicates that text was omitted.
'(item1 item2)	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.

Convention	Description
|	Separates items in a list of choices. For example, on | off.
\	Indicates the continuation of a command line.

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 8:30 AM through 5:30 PM Pacific Standard Time (PST) at:

    ```
    1-510-413-8000
    ```

    or

    ```
    1-800-369-0080
    ```

    Ask the receptionist for customer support.

- Faxing a description of the problem to the Avant! Corporate office at:

    ```
    1-510-413-8080
    ```

    Ensure that you write "Attn.: Customer Support Service Center" somewhere on the cover letter so the FAX can be properly routed.

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).

Avant!

Contents

Contents

Contents

Avant!

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 Features

Figure 1-1: Star-Hspice Design Features
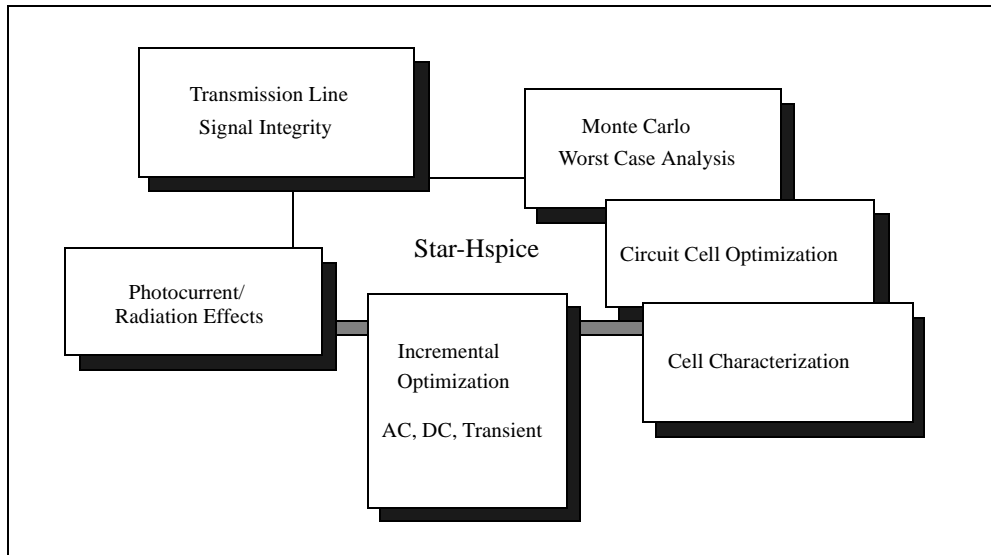
Star-Hspice is compatible with most SPICE variations, and has the following additional features:

- 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

- 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
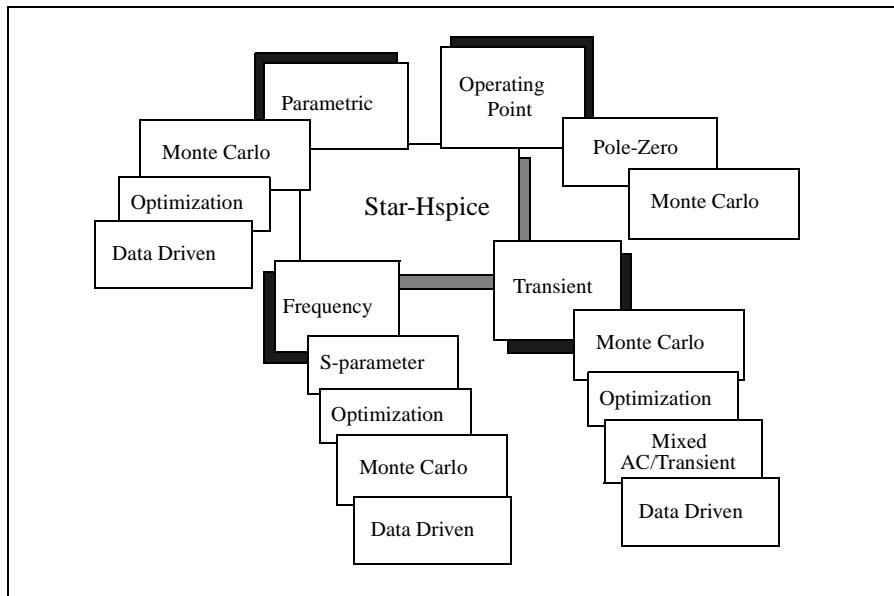
Figure 1-3: Star-Hspice Modeling Technologies
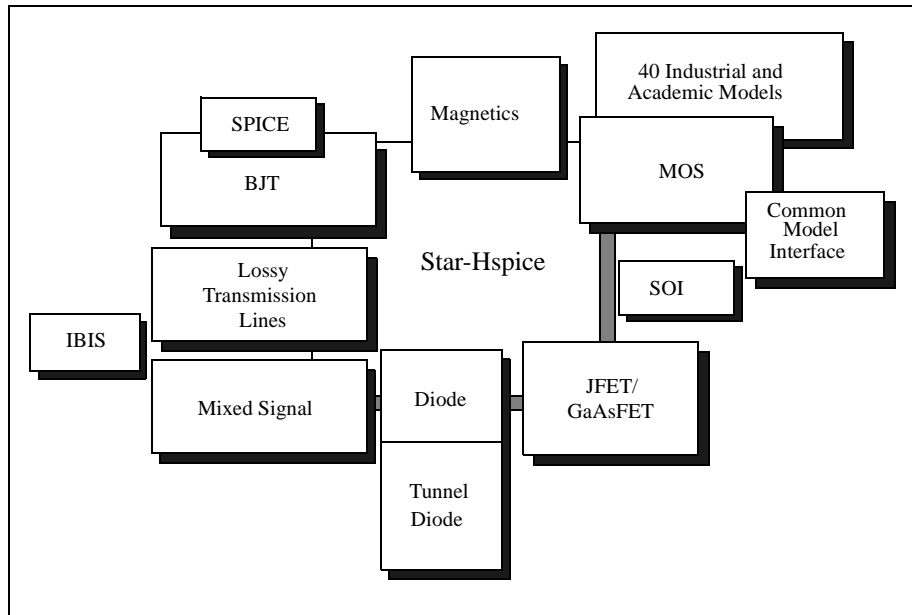
Simulation at the integrated circuit level and at the system level requires careful planning of the organization and interaction between transistor models and subcircuits. Methods that worked for small circuits might have too many limitations when applied to higher-level simulations.

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:

Platform	Operating System
Sun Sparc and Ultra	Solaris 5.5, 5.6 and 5.7, SunOS 4.1 +
HP PA	UX 10.20 +
IBM RS6000	AIX 4.1 +
DEC Alpha	OSF 4.0 +
SGI	IRIX 6.5
PC	Windows 95, 98, and NT
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
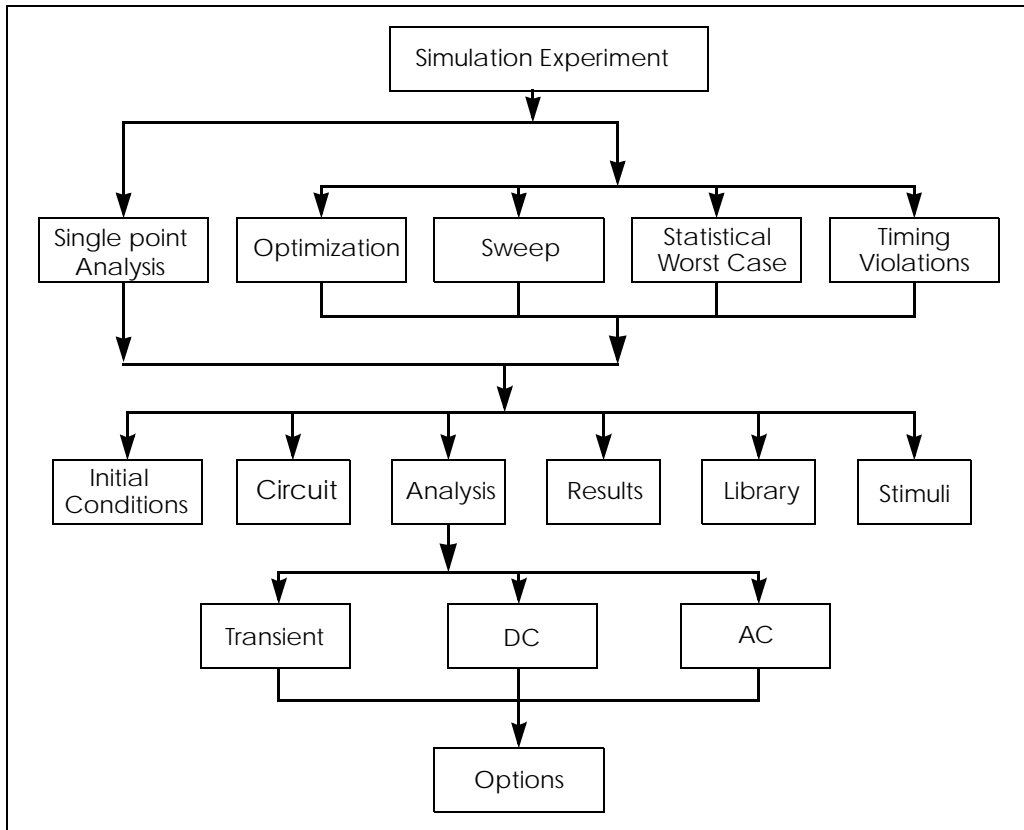
Analysis and verification of complex designs are typically organized around a series of experiments. These experiments are simple sweeps or more complex Monte Carlo, optimization, and setup and hold violation analyses that analyze DC, AC, and transient conditions.

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.

Figure 1-5: Overview of Star-Hspice Data Flow
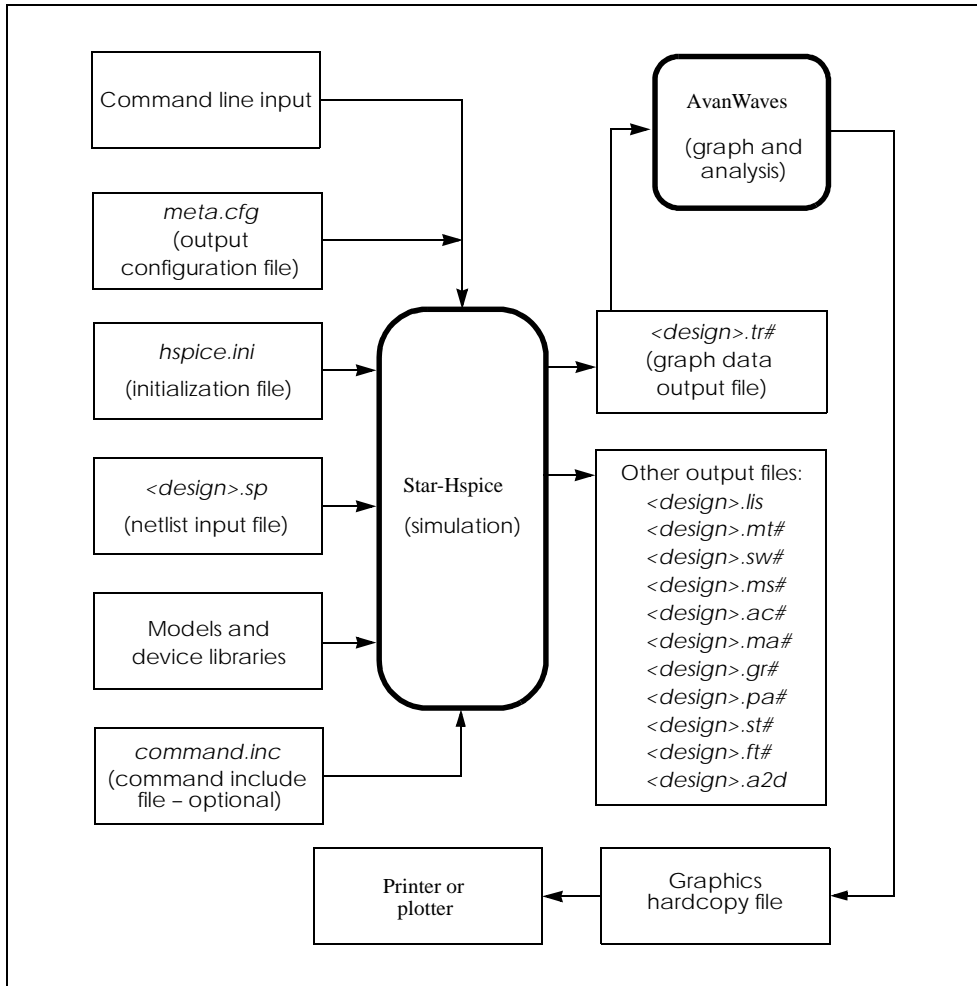
Simulation Process Overview

Figure 1-6 is a diagram of the Star-Hspice simulation process. The following section summarizes the steps in a typical simulation.

Figure 1-6: Star-Hspice Simulation Process

1. Invocation	*hspice -i demo.ip -o demo.lis*
2. Run script	Select version Select best architecture Run Star-Hspice program
3. Licensing	Find license file in *LM_LICENSE_FILE* Get FLEX*lm* license token
4. Simulation configuration	Read *~/meta.cfg* or Read *<installdir>/meta.cfg*
5. Design input	Read input file: *demo.sp* Open temp. files in *$tmpdir* Open output file Read *hspice.ini* file
6. Library input	Read .INCLUDE statement files Read .LIB Read implicit include (*.inc*) files
7. Operating point Initialization	Read *.ic* file (optional) Find operating point Write *.ic* file (optional)
8. Multipoint analysis	Open measure data files *.mt0* Initialize outer loop sweep Set analysis temperature
9. Single point analysis	Open graph data file *.tr0* Perform analysis sweep
10. Worst case .ALTER	Process library delete/add Process parameter and topology changes
11. Clean up	Close all files Release all tokens

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 FLEX*lm* licensing management system. With FLEX*lm* 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.

Avant!

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.

Figure 2-1: RC Network Circuit
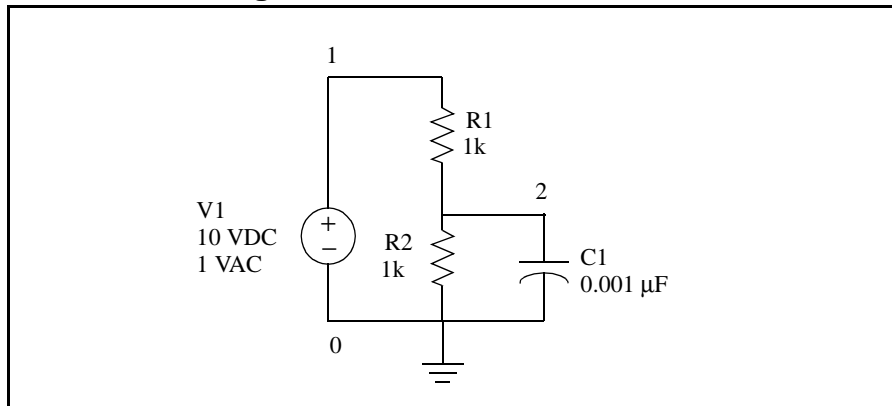
The Star-Hspice netlist for the RC network circuit is:

```
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.ic
   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
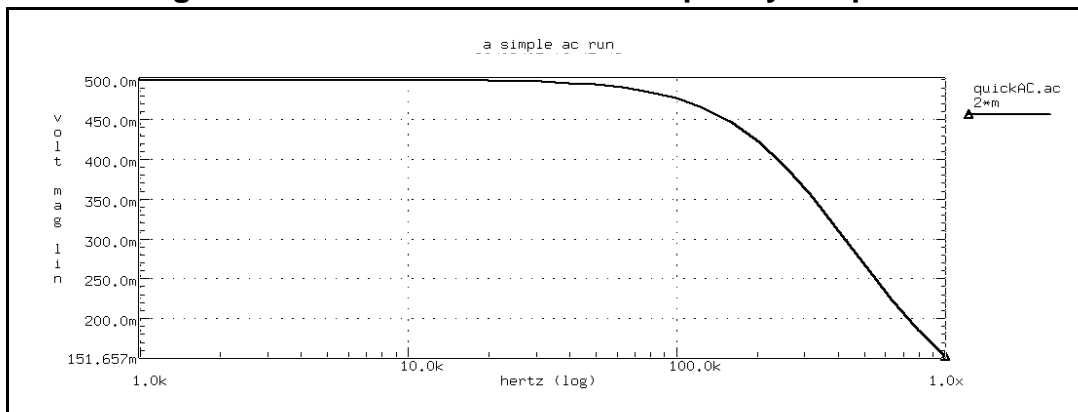
The file *quickAC.lis* displays the input netlist, details about the elements and topology, operating point information, and the table of requested data as the input is swept from 1 kHz to 1 MHz. The files *quickAC.ic* and *quickAC.st0* contain information about the DC operating point conditions and 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
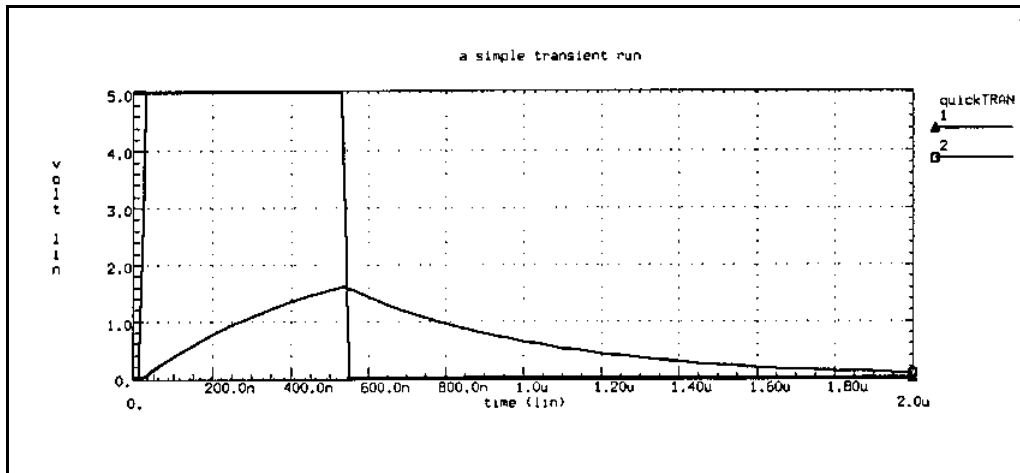
Transient Analysis of an Inverter

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

Figure 2-4: MOS Inverter Circuit
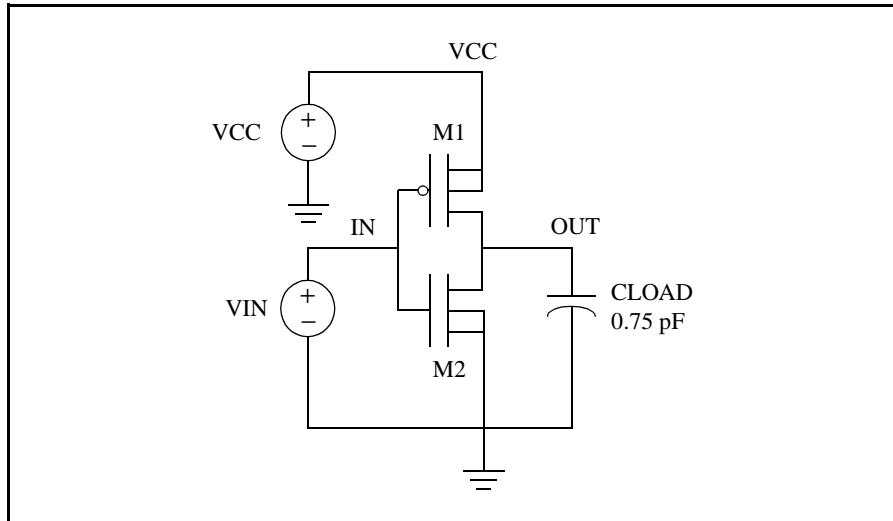
1. Type the following netlist data into a file named *quickINV.sp*.

```
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

Avant!
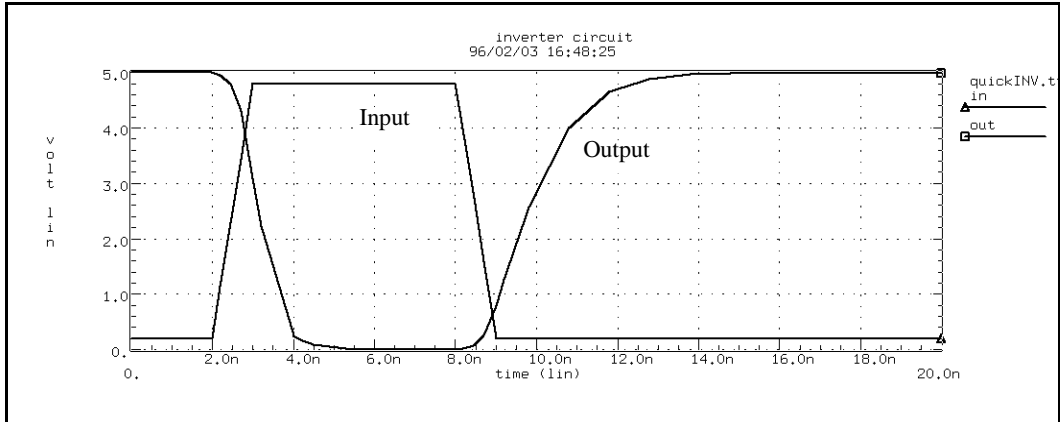
Specifying Simulation Input and Controls

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 alphabetic characters are ignored in node numbers. For example, node 1A is the same as node 1.
- 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 netlisters 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

Sections	Examples	Ch	Definition
Title	.TITLE	3	The first line is the input netlist file title
Set-up	.OPTIONS	9	Sets conditions for simulation
	.IC or .NODESET	10	Initial values in circuit and subcircuit
	.PARAM	7	Set parameter values in the netlist
	.GLOBAL	7	Set node name globally in netlist
Sources	Sources (I or V) and digital inputs	5	Sets input stimuli
Netlist	Circuit elements	3-4	Circuit for simulation
	.SUBKCT, .ENDS	3	Subcircuit definitions
Analysis	.DC, .TRAN, .AC, etc.	10-12	Statements to perform analyses
	.SAVE and .LOAD	10	Save and load operating point info
	.DATA	3	Create table for data-driven analysis
	.TEMP	3	Set analysis temperature
Output	.PRINT, .PLOT, .GRAPH, .PROBE	8	Statements to output variables
	.MEASURE	8	Statement to evaluate and report user-defined functions of a circuit

Sections	Examples	Ch	Definition
Library, Model and File Inclusion	.INCLUDE	3	General include files
	.MODEL	3, 8	Element model descriptions
	.LIB	3	Library
	.<UN>PROTECT	3	Control printback to output listing
Alter blocks	.ALTER	3	Sequence for in-line case analysis
	.DELETE LIB	3	Removes previous library selection
End of netlist	.END	3	Required statement to end the netlist

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

```
elname <node1 node2 ... nodeN> <mname>
+ <val1 val2 ... valn>
```

where:

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

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>
mname	Model reference name is required for all elements except passive devices.
pname1 ...	Element parameter name used to identify the parameter value that follows this name.
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

```
*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
*
```

```
.MODEL DA D  CJA = CAJA CJP = CAJP VRB = -20 IS = 7.62E-18
+   PHI = .5 EXA = .5 EXP = .33
*
.PARAM CAJA = 2.535E-16 CAJP = 2.53E-16
.END
```

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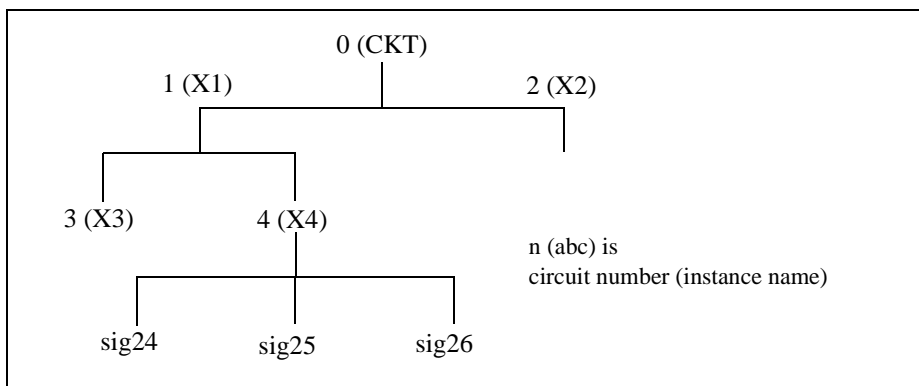
The path name of the node named *sig25* in subcircuit X4 in Figure 3-1 is *X1.X4.sig25.* You can use this path in Star-Hspice statements such as:

```
.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*nn*, where *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
```
.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.
```
.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
```
.TEMP t1 <t2 <t3 ...>>
```

where:

t1 t2 ...	Specifies temperatures, in °C, at which the circuit is to be 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.

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, 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).

.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 pnam1 <pnam2 pnam3 ... pnamxxx >
+   pval1<pval2  pval3 ... pvalxxx>
+   pval1' <pval2' pval3' ... pvalxxx'>
.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.

```
.DATA datanm MER
FILE = 'filename1' pname1 = colnum <panme2 = colnum
...>
<FILE = 'filename2' pname1 = colnum <pname2 = 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,

File A	*File B*	*File C*
a a a	b b b	c c c
a a a	b b b	c c c
a a a		

the data appears as follows:

```
a a a
a a a
a a a
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' pname1 = colnum <panme2 = colnum
...>
<FILE = 'filename2' pname1 = colnum <pname2 = colnum
...>>
...
<OUT = 'fileout'>
.ENDDATA
```

where:

datanm	Specifies the data name referred to in the .TRAN, .DC or .AC statement
LAM	Specifies column laminated (parallel 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.

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,

File D	*File E*	*File F*
d1 d2 d3	e4 e5	f6
d1 d2 d3	e4 e5	f6
d1 d2 d3	e4 e5	f6

the laminated data appears as follows:

```
d1 d2 d3 e4 e5 f6
d1 d2 d3 e4 e5 f6
d1 d2 d3 e4 e5 f6
```

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:

S*tar-Hspice release*	*Version number*
9007B	9007.02
9007D	9007.04
92A	92.01
92B	92.02
93A	93.01
93A.02	93.02
95.3	95.3
96.1	96.1

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

```
.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.

.LIB Library 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.
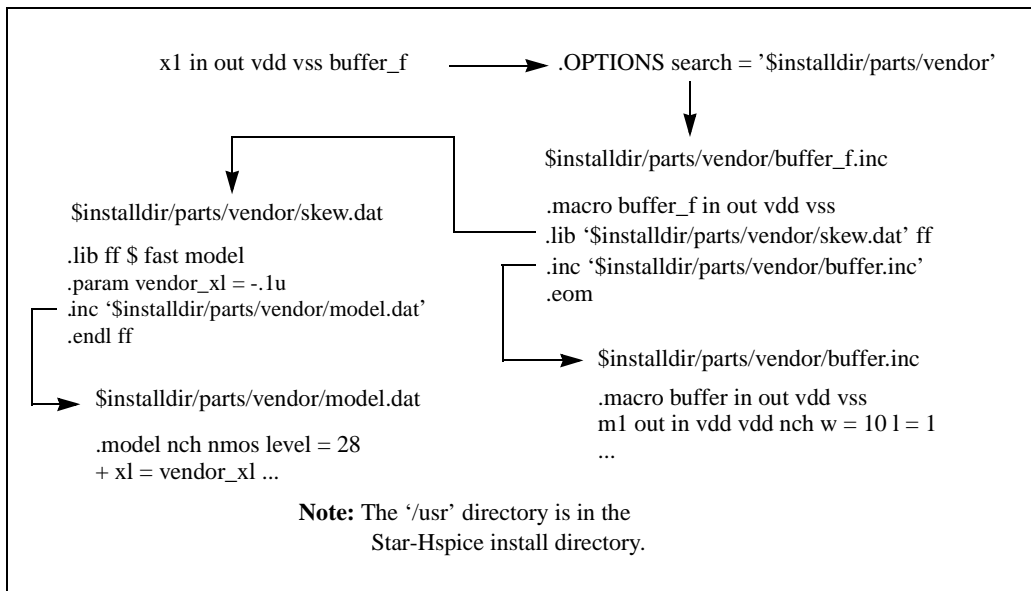
Automatic Library Selection

Automatic library selection allows a search order for up to 40 directories. The *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 .OPTIONS SEARCH statements. When Star Hspice encounters a subcircuit call, the search order is as follows:

1. Read the input file for a .SUBCKT or .MACRO with call name.

2. Read any .INC files or .LIB files for a .SUBCKT or .MACRO with the call name.

3. Search the directory that the input file was in for a file named *call_name.inc*.

4. Search the directories in the .OPTIONS SEARCH list.

By using the Star-Hspice library search and selection features you can, for example, simulate process corner cases, using .OPTIONS SEARCH = '*<libdir>*' to target different process directories. If you have an input/output buffer subcircuit stored in a file named *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 *iobuf.inc*) in separate directories.

.PARAM Statement

Use the .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. 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)')
```

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 8-36.

.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:

- 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.

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
.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
.PLOT V(3)  V(2)
.ALTER
.DEL LIB 'MOS.LIB' NORMAL
.TF V(3) VIN              $DC small-signal transfer function
*
```

```
 .MACRO INV 1  2   3          $change data within subcircuit def
 M1  4 2 1 1 P 100U 100U      $change channel length,width,also
                              $topology
 M2  4 2 0 0 N 6U   8U        $change topology
 R4  4  3  100                $add the new element
 C3  3  0  10P                $add the new element
 .LIB 'MOS.LIB' SLOW          $set slow model library
 $.INC 'MOS2.DAT'             $not allowed to be used inside
                              $subcircuit allowed outside
                              $subcircuit
 .EOM INV
 *
 .END
```

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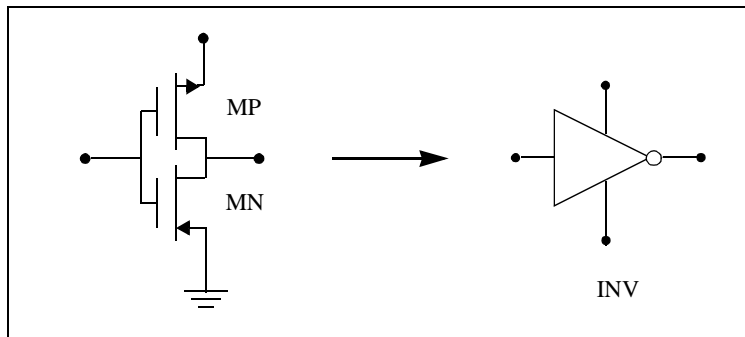
The following Star-Hspice input creates an instance named X1 of the INV cell macro, which consists of two MOSFETs, MN and MP:

```
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
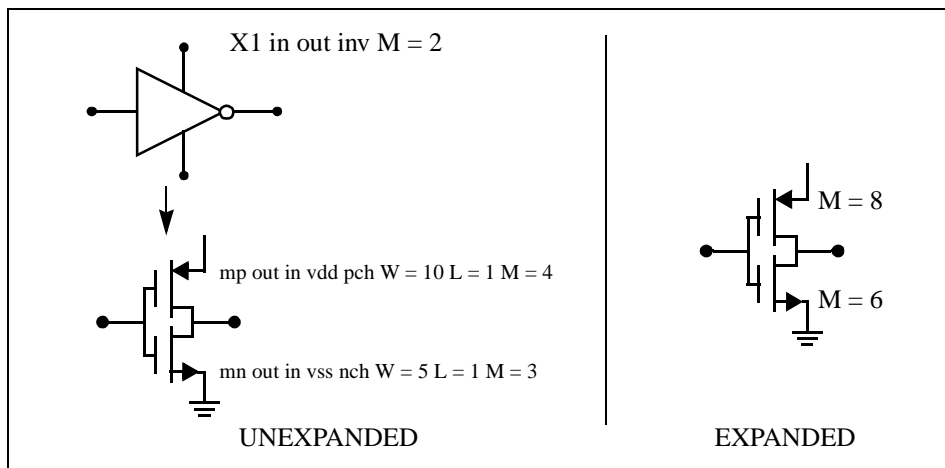
Example

```
X1 D Q Qbar CL CLBAR dlatch flip = 0
macro dlatch
+ D Q Qbar CL CLBAR flip = vcc
.nodeset v(din) = flip
xinv1 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
```

Figure 3-5: D Latch with Nodeset
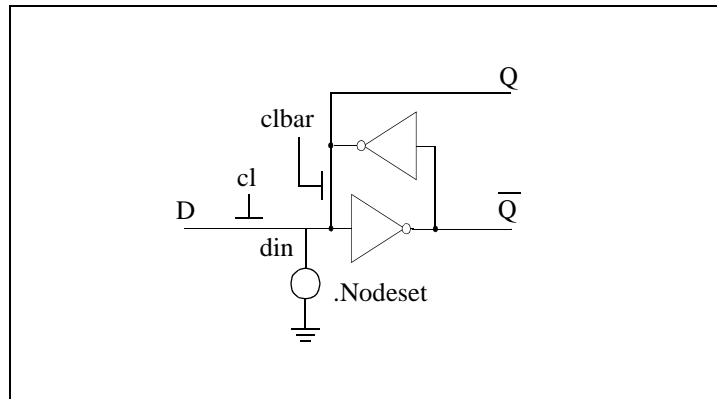
There is no limit to the size or complexity of subcircuits; they may contain subcircuit references and any model or element statement. To specify subcircuit nodes in .PRINT or .PLOT statements, give the full subcircuit path and node name.

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.

Figure 3-6: Vendor Library Usage

x1 in out vdd vss buffer_f ⟶ .OPTION search = '/usr/lib/vendor'

/usr/lib/vendor/buffer_f.inc

/usr/lib/vendor/skew.dat

.macro buffer_f in out vdd vss
.lib '/usr/lib/vendor/skew.dat' ff
.inc '/usr/lib/vendor/buffer.inc'
.eom

.lib ff $ fast model
.param vendor_xl = -.1u
.inc '/usr/lib/vendor/model.dat'
.endl ff

/usr/lib/vendor/buffer.inc

/usr/lib/vendor/model.dat

.macro buffer in out vdd vss
m1 out in vdd vdd nch w = 10 l = 1
...

.model nch nmos level = 28
+ xl = vendor_xl ...

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 (*<subname>.inc*) resides in,
and then reference the name *<subname>* in each subcircuit call.

Component naming conventions require that each subcircuit be of the form
<subname>.inc and stored in a directory that is accessible through the
.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 3-1: Star-Hspice Input Files

Input File Type	**File Name**
Output configuration file	*meta.cfg*
Initialization file	*hspice.ini*
DC operating point initial conditions file	*<design>.ic*
Input netlist file	*<design>.sp*
Library input file	*<library_name>*
Analog transition data file	*<design>.d2a*

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

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

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:
    ```
    The default is to use the standard system priority.
    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

Option	Description
-a <arch>	Platform architecture. Choices are: • sun4, sol4 (SparcStation, Ultra) • sgi (SGI) • pc (Windows 95/NT) • pa (HP 700/800/9000) • alpha (DEC ALPHA) • rs (IBM RS6000)
-i <input_file>	Name of the input netlist file. If no extension is given, *.sp* is assumed.
-m <mem_needed>	Amount of memory requested for the simulation, in 8-byte words (only required for Star-Hspice releases prior to Release H93A.01)
-n <number>	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).
-o <output_file>	Name of the output file. If no extension is given, *.lis* is assigned.
-r <remote_host>	Name of the machine on which to run the simulation
-v <version>	Star-Hspice version. Choices are determined at the time of installation by the Star-Hspice installation script.
-x	Displays the Star-Hspice script on the window as it runs

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:

```
hspice -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:

```
Tmt = Tserial + Tparallel/Ncpu + Toverhead
```

where:

Tserial	Represents the Star-Hspice calculations that are not threaded
Tparallel	Represents the threaded Star-Hspice calculations
Ncpu	The number of CPUs used. *Toverhead* 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, *Tparallel* is about 80% of T1cpu (the CPU time associated with a single CPU), if you run with two threads on a multi-CPU machine. Ideally, assuming *Toverhead = 0*, you can achieve a speedup of:

```
T1cpu/(0.2T1cpu + 0.8T1cpu/2cpus) = 1.67
```

For six CPUs the speedup is:

```
T1cpu/(0.2T1cpu + 0.8T1cpu/6cpus) = 3.0
```

The typical value of *Tparallel* is 0.6 to 0.7 for moderate to large circuits.

Avant!

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

Passive Elements

Resistors

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

General form:

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.

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.

```
.PARAMETER 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-dependant 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.

CTYPE	Determines capacitance charge calculation for elements with capacitance equations. If capacitance equation is a function of v(n1,n2), set CTYPE = 1. This setting must be used correctly to ensure proper capacitance calculations and hence simulation results. Default = 0.
POLY	Keyword to specify capacitance given by a polynomial.
c0 c1...	Coefficients of a polynomial in voltage describing the capacitor value. c0 represents the magnitude of the 0th order term, c1 represents the magnitude of the 1st order term, and so on. Note that the coefficients can not be parameterized.

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 = 'lu*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.

L = 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 = '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 = turns	Number representing the number of turns of an inductive magnetic winding.

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.

```
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:*
>
> > Kxxx Lyyy Lzzz <K = >coupling
>
> *Mutual Core form:*
>
> > Kaaa Lbbb <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.

Lzzz	Name of the second of two coupled inductors.
K = coupling	Coefficient of mutual coupling. K is a unitless number with magnitude greater than 0 and less than or equal to 1. If K is negative, the direction of coupling is reversed. This reversal is equivalent to reversing the polarity of either of the coupled inductors. The K = coupling syntax should be used when using a parameterized value or an equation.
Kaaa	Saturable core element name. Must begin with "K", which can be followed by up to 1023 alphanumeric characters.
Lbbb, Lccc, Lddd	The names of the windings about the Kaaa core. One winding element is required, and each winding element must have the magnetic winding syntax.
mname	Saturable core model name. See "Using Passive Device Models" on page 14-1 for model information.
MAG = magnetization	Initial magnetization of the saturable core. Can be set to +1, 0 and -1, where +/- 1 refer to positive and negative values of the model parameter BS (see "Using Passive Device Models" on page 14-1).

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.

```
K1 Lin Lout 0.9
```

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

```
Kxfmr Lhigh Llow K = 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.

```
K1 L1 L2 0.98
K2 L2 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:

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

or

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

Fowler-Nordheim (LEVEL = 2) form:

    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

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.

W	Width of the diode in meters (diode model LEVEL=3 only)
L	Length of the diode in meters (diode model LEVEL = 3 only)

Example

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

```
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.

```
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.

```
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:

> Qxxx nc nb ne <ns> mname <area> <OFF> <IC = vbeval,vceval>
> + <M = val> <DTEMP = val>

or

> Qxxx nc nb ne <ns> mname <AREA = area> <AREAB = val>
> + <AREAC = val> <OFF> <VBE = vbeval> <VCE = vceval>
> <M = val>
> + <DTEMP = val>

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

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.

```
Q1 1 2 3 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:

> Jxxx nd ng ns <nb> mname <<<AREA> = area | <W = val>
> <L = val>>
> + <OFF> <IC = vdsval,vgsval> <M = val> <DTEMP = val>

or

> Jxxx nd ng ns <nb> mname <<<AREA> = area> | <W = val>
> <L = val>>
> + <OFF> <VDS = vdsval> <VGS = vgsval> <M = val>
> <DTEMP = 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

ng	Gate terminal node name
ns	Source terminal node name
nb	Bulk terminal node name, which is optional.
mname	JFET or MESFET model name reference
area, AREA = area	Area multiplying factor that affects the BETA, RD, RS, IS, CGS and CGD model parameters. Default = 1.0 in units of square meters.
W	FET gate width in meters
L	FET gate length in meters
OFF	Sets initial condition to OFF for this element in DC analysis. Default = ON.
IC = vdsval, vgsval, VDS, VGS	Initial internal drain-source voltage (vdsval) and gate-source voltage (vgsval). These are used when UIC is present in the .TRAN statement and is overridden by the .IC statement.
M	Multiplier to simulate multiple JFETs or MESFETs 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.

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
```

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

L	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.
AD	Drain diffusion area. Overrides DEFAD in the OPTIONS statement. Default = DEFAD only when the MOSFET model parameter ACM = 0.
AS	Source diffusion area. Overrides DEFAS in the OPTIONS statement. Default = DEFAS only when the MOSFET model parameter ACM = 0.
PD	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.
PS	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.
NRD	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.
NRS	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.
RDC	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.

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.

```
M1 1 2 3 model_1
```

MOSFET Mopamp1 with drain, gate, and source connected to and 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.

```
Mopamp1 d1 g3 s2 b 1stage L = 2u W = 10u
```

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.

```
Mdrive driver in output bsim3v3 W = 3u L = 0.25u 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 xth transmission line (in1 is required).
refin	Ground reference for input signal
outx	Signal output node for the xth transmission line (each input port must have a corresponding output port).
refout	Ground reference for output signal.

RLGCfile = fname	File name reference for file containing the RLGC information for the transmission lines (see "W Element Transmission Line Properties Inputs" on page 18-14 for syntax).
N	Number of conductors (excluding the reference conductor).
L	Physical length of the transmission line in units of meters.
Umodel = mname	U-model lossy transmission-line model reference name. A lossy transmission line model, used here to represent the characteristics of the W-element transmission line.
FSmodel = mname	Internal field solver model name. References the PETL internal field solver as the source of the transmission-line characteristics (see "W Element Transmission Line Properties Inputs" on page 18-14 for syntax).

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 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 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 i1 i2 i3 i4 i5 gnd o1 gnd o3 gnd o5 gnd FSmodel = board1
+ 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 `RLGCfile,FSmodel` or `Umodel` 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
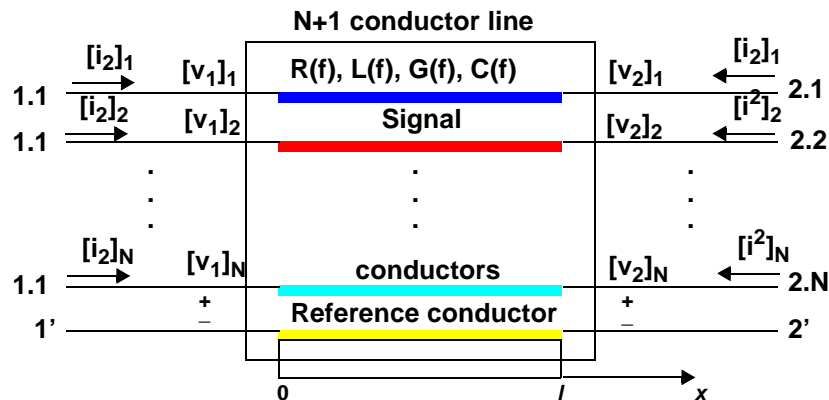
T Element Statement

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

General form:

> Txxx in refin out refout Z0 = val TD = val <L = val>
> <IC = v1,i1,v2,i2>

or

> Txxx in refin out refout Z0 = val F = val <NL = val>
> <IC = v1,i1,v2,i2>

U-model form:

> 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.

```
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 xth transmission line (in1 is required).
refin	Ground reference for input signal
outx	Signal output node for the xth 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

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.

Avant!

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

```
Vxxx n+ n- <<DC=> dcval> <tranfun> <AC=acmag, <acphase>>
```
or
```
Iyyy n+ n- <<DC=> dcval> <tranfun> <AC=acmag, <acphase>>
+ <M=val>
```

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 1while 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.

```
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.

```
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.

```
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).

```
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.

```
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.

```
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 *per* from onset to onset.

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

General form:

```
Vxxx n+ n- PU<LSE> <(>v1 v2 <td <tr <tf <pw <per>>>>> <)>
```

or

```
Ixxx n+ n- PU<LSE> <(>v1 v2 <td <tr <tf <pw <per>>>>> <)>
```

The arguments are defined as:

Vxxx, Ixxx	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:

Time	Value
0	*v1*
td	*v1*
td + tr	*v2*
td + tr + pw	*v2*
td + tr + pw + tf	*v1*
tstop	*v1*

Intermediate points are determined by linear interpolation.

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

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.

```
vIN 3 0 PULSE (-1 1 2NS 2NS 2NS 50NS 100NS)
```

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

```
V1 99 0 PU lv hv tdlay tris tfall tpw 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
```

Figure 5-1: Pulse Source Function
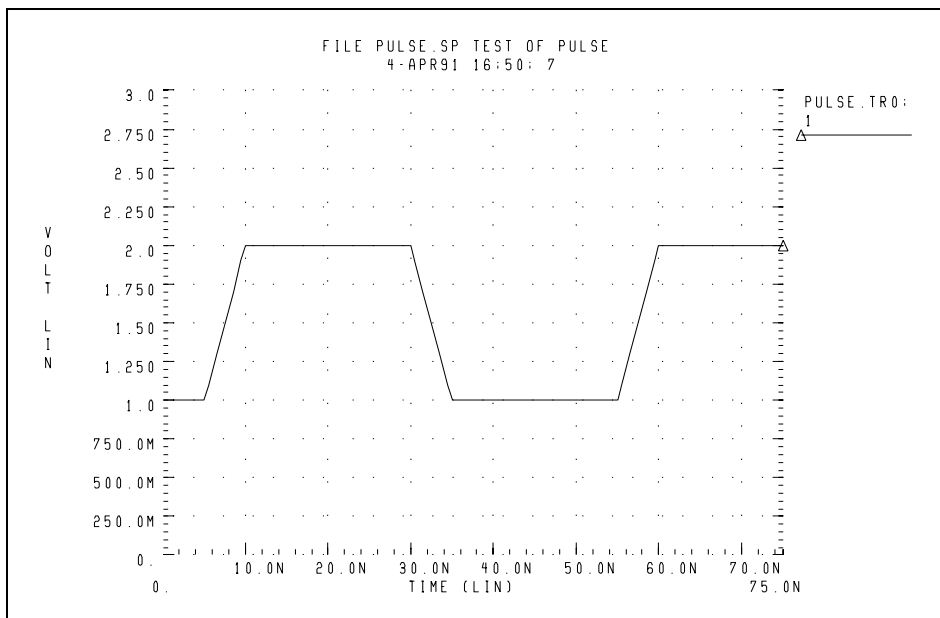
Sinusoidal Source Function

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:

```
Vxxx n+ n- SIN <(> vo va <freq <td <θ <φ>>>> <)>
```

or

```
Ixxx n+ n- SIN <(> vo va <freq <td <θ <φ>>>> <)>
```

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:

Time	*Value*
0 to td	$vo + va \cdot SIN\left(\dfrac{2 \cdot \Pi \cdot \varphi}{360}\right)$
td to tstop	$vo + va \cdot Exp[-(Time - td) \cdot \theta] \cdot$ $SIN\left\{2 \cdot \Pi \cdot \left[freq \cdot (time - td) + \dfrac{\varphi}{360}\right]\right\}$

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
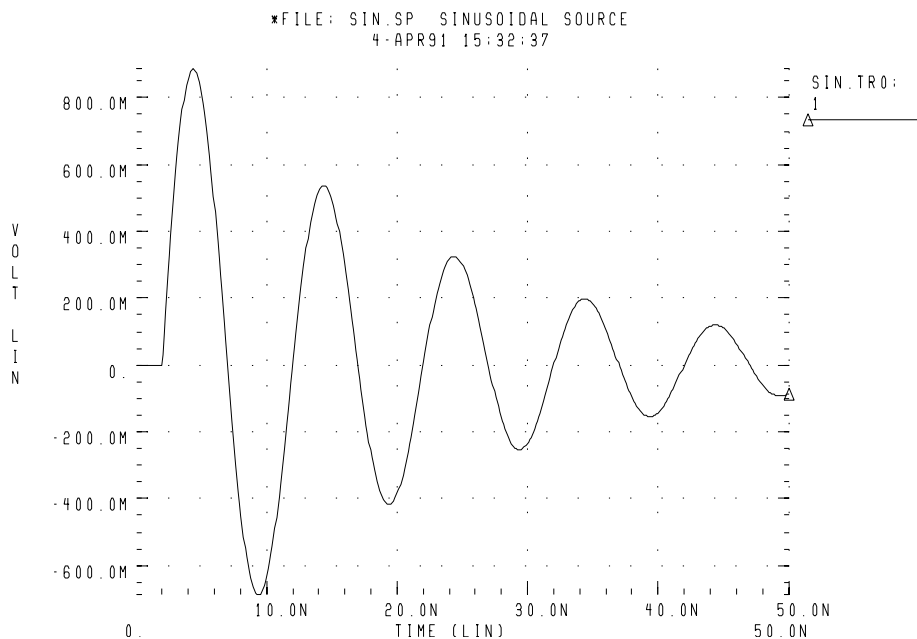

```
*File: SIN.SP THE SINUSOIDAL WAVEFORM
*<decay envelope>
.OPTIONS POST
.PARAM V0=0 VA=1 FREQ=100MEG DELAY=2N THETA=5E7
+PHASE=0
V  1  0  SIN (V0 VA FREQ DELAY THETA PHASE)
R  1  0  1
.TRAN .05N 50N
.END
```

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:

```
    Vxxx n+ n- EXP <(> v1 v2 <td1 <τ1 <td2 <τ2>>>> <)>
```

or

```
    Ixxx n+ n- EXP <(> v1 v2 <td1 <τ1 <td2 <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:

Time	*Value*
0 to td1	$v1$
td1 to td2	$v1 + (v2 - v1) \cdot \left[1 - Exp\left(-\dfrac{Time - td1}{\tau_1} \right) \right]$

td2 to tstop $v1 + (v2 - v1) \cdot \left[1 - Exp\left(-\dfrac{td2 - td1}{\tau_1} \right) \right] \cdot$

$$Exp\left[\dfrac{-(Time - td2)}{\tau_2} \right]$$

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
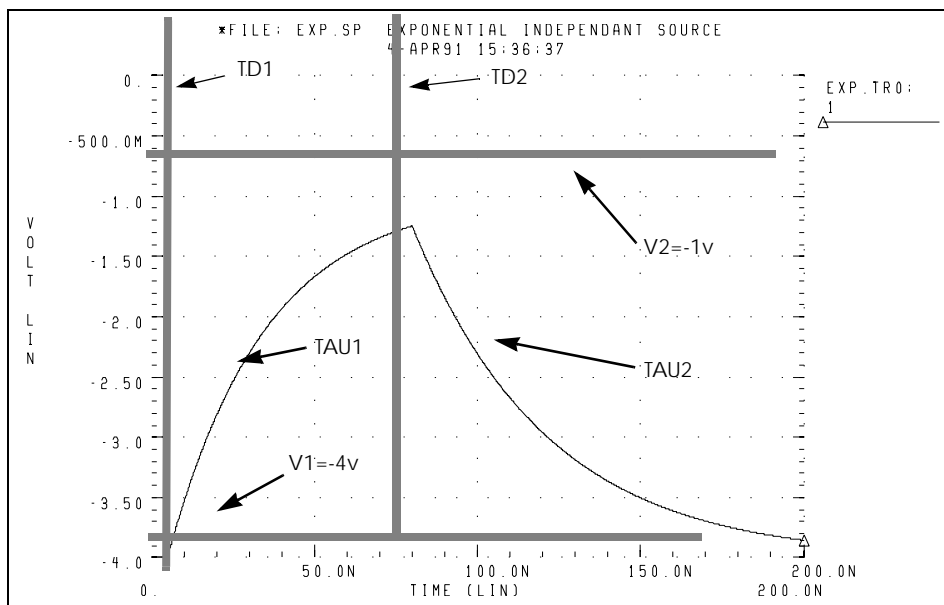
```
*FILE: EXP.SP THE EXPONENTIAL WAVEFORM
.OPTIONS POST
.PARAM V1=-4 V2=-1 TD1=5N TAU1=30N TAU2=40N TD2=80N
V 1 0 EXP (V1 V2 TD1 TAU1 TD2 TAU2)
R 1 0 1
.TRAN .05N 200N
.END
```

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 8-1for 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.

Figure 5-4: Results of Using the Repeat Function
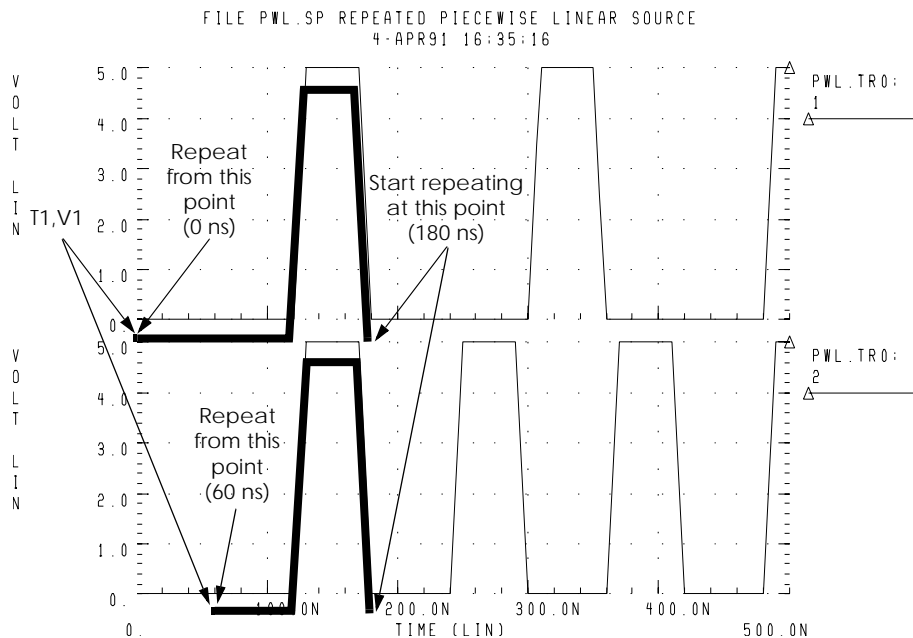
Data Driven Piecewise Linear Source Function

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
.. ..
.ENDDATA
.TRAN DATA=datanam
```

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:

```
Vxxx n+ n- SFFM <(> vo va <fc <mdi <fs>>> <)>
```

or

```
Ixxx n+ n- SFFM <(> vo va <fc <mdi <fs>>> <)>
```

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:

$$sourcevalue = vo + va \cdot SIN[2 \cdot \pi \cdot fc \cdot Time + mdi \cdot SIN(2 \cdot \pi \cdot fc \cdot Time)]$$

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
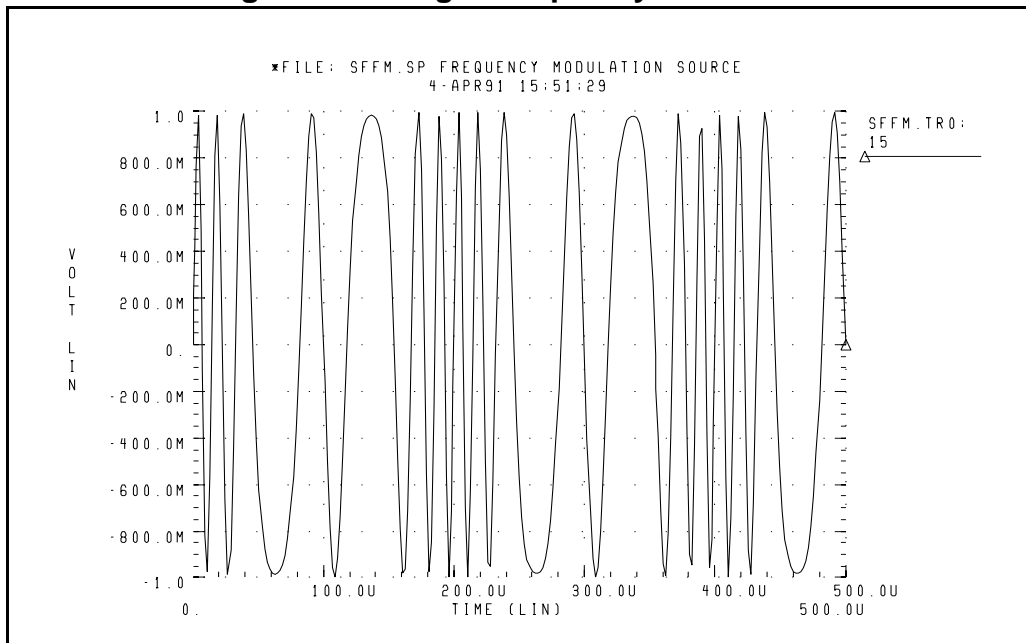
Amplitude Modulation Source Function

The general syntax for including a single-frequency frequency-modulated source in an independent voltage or current source is:

General form:

```
Vxxx n+ n- AM < (> sa oc fm fc <td> <)>
```

or

```
Ixxx 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
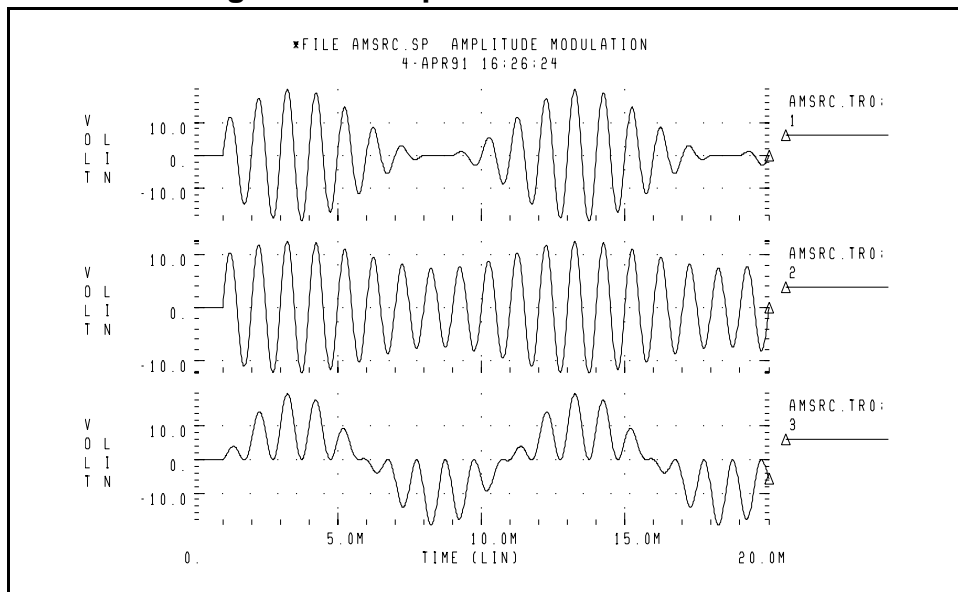
Using Voltage and Current Controlled Elements

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:

$$FV = P0 + (P1 \cdot FA) + (P2 \cdot FA^2) + \\ (P3 \cdot FA^3) + (P4 \cdot FA^4) + (P5 \cdot FA^5) + \ldots$$

FV	Controlled voltage or current from the controlled source
P0. . .PN	Coefficients of polynomial equation
FA	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) 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 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 FA=V(3,2). The dependent source statement then specifies that P0=1 and P1=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), FV is determined by the following expression:

$$FV = P0 + (P1 \cdot FA) + (P2 \cdot FB) + (P3 \cdot FA^2) + (P4 \cdot FA \cdot FB) + (P5 \cdot FB^2)$$
$$+ (P6 \cdot FA^3) + (P7 \cdot FA^2 \cdot FB) + (P8 \cdot FA \cdot FB^2) + (P9 \cdot FB^3) + \dots$$

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 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:

$$
\begin{aligned}
FV = \ & P0 + (P1 \cdot FA) + (P2 \cdot FB) + (P3 \cdot FC) + (P4 \cdot FA^2) \\
& + (P5 \cdot FA \cdot FB) + (P6 \cdot FA \cdot FC) + (P7 \cdot FB^2) + (P8 \cdot FB \cdot FC) \\
& + (P9 \cdot FC^2) + (P10 \cdot FA^3) + (P11 \cdot FA^2 \cdot FB) + (P12 \cdot FA^2 \cdot FC) \\
& + (P13 \cdot FA \cdot FB^2) + (P14 \cdot FA \cdot FB \cdot FC) + (P15 \cdot FA \cdot FC^2) \\
& + (P16 \cdot FB^3) + (P17 \cdot FB^2 \cdot FC) + (P18 \cdot FB \cdot FC^2) \\
& + (P19 \cdot FC^3) + (P20 \cdot FA^4) + \ldots
\end{aligned}
$$

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:

$$
\begin{aligned}
FA &= V(3,2) \\
FB &= V(7,6) \\
FC &= V(9,8) \\
P1 &= 3 \\
P7 &= 4 \\
P19 &= 5
\end{aligned}
$$

Substituting these values into the voltage controlled voltage source statement yields the following:

$V(1, 0)$ POLY(3) 3 2 7 6 9 8 0 3 0 0 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 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 DELTA is provided to control the curvature of the characteristic at the corners. The smaller the DELTA, the sharper the corners are. The maximum DELTA is limited to half of the smallest breakpoint distance. If the breakpoints are quite separated, specify the 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

```
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- ... ink+ ink- <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>
```

Behavioral 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
```

Parameter Definitions

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 macro-modelling 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 one-fourth of the smallest breakpoint distances. The maximum is limited to one-half of the smallest breakpoint distances.
Exxx	Voltage controlled element name. The parameter must begin with an "E" followed by up to 1023 alphanumeric characters.
gain	Voltage 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.
IC	Initial condition: the initial estimate of the value(s) of the controlling voltage(s). If IC is not specified, the default=0.0.
in +/-	Positive or negative controlling nodes. Specify one pair for each dimension.

k	Ideal transformer turn ratio: $$V(in+,in-) = k \cdot V(n+,n-)$$ or, number of gates input
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
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, $$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$ The values of tstep and tstop are specifed in the .TRAN statement.
OPAMP	The keyword for ideal op-amp element. OPAMP is a reserved word and should not be used as a node name.
P0, P1 …	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).
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: $$\mathrm{SCALEeff} \ = \ \mathrm{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$
TD	Time delay keyword
TRANSFORMER	Keyword for ideal transformer. TRANS is a reserved word and should not be used as a node name.
VCVS	Keyword for voltage controlled voltage source. VCVS is a reserved word 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 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.

```
Eopamp 2 3 14 1 MAX=+5 MIN=-5  2.0
```

A user-defined parameter can 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 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.

```
Einv out 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- <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

Syntax

```
Gxxx n+ n- CUR='equation' <MAX>=val> <MIN=val> <M=val>
+     <SCALE=val>
```

Voltage Controlled Resistor (VCR)

Syntax

Linear

Gxxx n+ n- VCR in+ in- transfactor <MAX=val> <MIN=val> <SCALE=val> <M=val>
+　　　<TC1=val> <TC2=val> <IC=val>

Polynomial

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

Gxxx n+ n- VCR PWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> <TC1=val>
+　　　<TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Gxxx n+ n- VCR NPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> <TC1=val>
+　　　<TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Gxxx n+ n- VCR PPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> <TC1=val>
+　　　<TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Multi-Input Gates

Gxxx n+ n- VCR gatetype(k) in1+ in1- ... ink+ ink- <DELTA=val>
+　　　<TC1=val> <TC2=val> <SCALE=val> <M=val> x1,y1 ... x100,y100 <IC=val>

Voltage Controlled Capacitor (VCCAP)

Syntax (Piecewise Linear)

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

ABS	Output is absolute value if ABS=1.
CUR, VALUE	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).
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 macromodel process. **Note:** Because DELAY is a Star-Hspice keyword, it 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.
Gxxx	Voltage controlled element name. This parameter must begin with a "G" followed by up to 1023 alphanumeric characters.

gatetype(k)	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.
IC	Initial condition. The initial estimate of the value(s) of the controlling voltage(s). If IC is not specified, the default=0.0.
in +/-	Positive or negative controlling nodes. Specify one pair for each dimension.
M	Number of 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
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, $$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop \rangle}{tstep}, 10\right]$$ The values of tstep and tstop are specified in the .TRAN statement.
NPWL	Models the symmetrical bidirectional switch or transfer gate, NMOS

P0, P1 …	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).
POLY	Polynomial keyword function
PWL	Piecewise linear keyword function
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, 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.
TC1,TC2	First and second order temperature coefficients. The SCALE is updated by temperature: $SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$
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 reserved word and should not be used as a node name.
VCCS	Keyword for voltage controlled current source. VCCS is a reserved word and should not be used as a node name.
VCR	Keyword for voltage controlled resistor element. VCR is a reserved word 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 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.

```
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.

```
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).

```
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.

```
gt i_anode cathode poly(2) anode,cathode grid,cathode 0 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

ABS	Output is absolute value if ABS=1.
CCVS	Keyword for current controlled voltage source. CCVS is a reserved word 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 macromodel process. 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.
gatetype(k)	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.
Hxxx	Current controlled voltage source element name. The parameter must begin with an "H" followed by up to 1023 alphanumeric characters.
IC	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.
MAX	Maximum voltage value. The default is undefined and sets no maximum value.
MIN	Minimum voltage 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, $$NPDELAY_{default} \ = \ max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$ The values of tstep and tstop are specifed 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: $$\text{SCALEeff} \ = \ \text{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$
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> 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> <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

ABS	Output is absolute value if ABS=1.
CCCS	Keyword for current controlled current source. Note that CCCS is a reserved word 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 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,

$$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$

The values of tstep and tstop are specifed 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: $SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$ |
| 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 VSENS MAX=+3 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.

```
.PARAM VU = 5
F1 13 5 VSENS MAX=+3 MIN=-3 VU
```

The next example describes a current controlled current source with the value:

$$I(F2)=1e\text{-}3 + 1.3e\text{-}3 \cdot I(VCC)$$

```
F2 12 10 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).

```
Fd 1 0 DELAY vin TD=7ns SCALE=5
```

The above example is a delayed current controlled current source.

```
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.

```
...
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:

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

Names (Alias)	Units	Default	Description
CLO	farad	0	Capacitance to low level node
CHI	farad	0	Capacitance to high level node
S0NAME			State "0" character abbreviation, can be a string of up to four alphanumerical characters.
S0TSW	sec		State "0" switching time
S0RLO	ohm		State "0" resistance to low level node

Names (Alias)	Units	Default	Description
S0RHI	ohm		State "0" resistance to high level node
S1NAME			State "1" character abbreviation, can be a string of up to four alphanumerical characters.
S1TSW	sec		State "1" switching time
S1RLO	ohm		State "1"resistace to low level node
S1RHI	ohm		State "1" resistance to high level node
S19NAME			State "19" character abbreviation, can be a string of up to four alphanumerical characters.
S19TSW	sec		State "19" switching time
S19RLO	ohm		State "19" resistance to low level node
S19RHI	ohm		State "19" resistance to high level node
TIMESTEP	sec		Digital input file step size (digital files only)

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
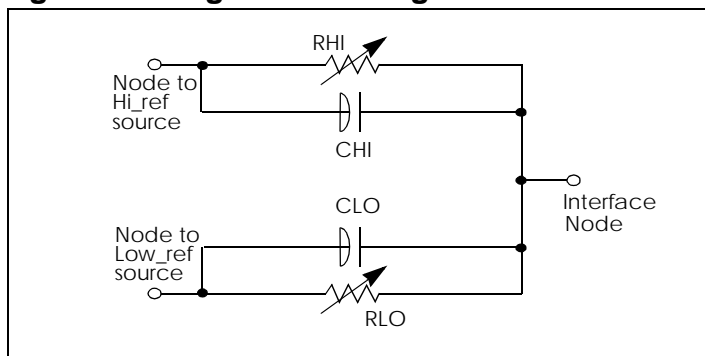
Example

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=0 S0TSW=1NS S0RLO = 15, S0RHI = 10K,
+ S2NAME=x S2TSW=3NS S2RLO = 1K, S2RHI = 1K
+ S3NAME=z S3TSW=5NS S3RLO = 1MEG,S3RHI = 1MEG
+ S4NAME=1 S4TSW=1NS S4RLO = 10K, 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<name> interface reference mname SIGNAME = 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 mname U LEVEL=4 <parameters...>
```

Analog-to-Digital Output Model Parameters

Name (Alias)	Units	Default	Description
RLOAD	ohm	1/gmin	Output resistance.
CLOAD	farad	0	Output capacitance.
S0NAME			State "0" character abbreviation, can be a string of up to four alphanumerical characters.
S0VLO	volt		State "0" low level voltage.
S0VHI	volt		State "0" high level voltage.
S1NAME			State "1" character abbreviation, can be a string of up to four alphanumerical characters.

Name (Alias)	Units	Default	Description
S1VLO	volt		State "1" low level voltage.
S1VHI	volt		State "1" high level voltage.
S19NAME			State "19" character abbreviation, can be a string of up to four alphanumerical characters.
S19VLO	volt		State "19" low level voltage.
S19VHI	volt		State "19" high level voltage.
TIMESTEP	sec	1E-9	Digital input file step size.
TIMESCALE			Scale factor for time.

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
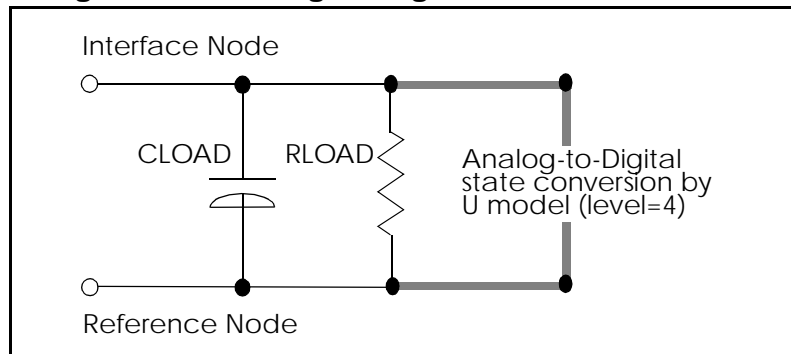
Replacing Sources With Digital Inputs

Figure 5-9: Digital File Signal Correspondence

Traditional voltage pulse sources ...

```
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
```

... become D2A drivers ...

```
UC carry-in VLD2A VHD2A D2A SIGNAME①1 IS=0
UA[0] A[0] VLD2A VHD2A D2A SIGNAME②2 IS=1
UA[1] A[1] VLD2A VHD2A D2A SIGNAME③3 IS=1
UB[0] B[0] VLD2A VHD2A D2A SIGNAME④4 IS=1
UB[1] B[1] VLD2A VHD2A D2A SIGNAME⑤5 IS=1
```

... that get their input from
    the Digital stimulus file ...
      <designname>.d2a

Signalname list

Time (in model time units)

Statechange: Signal list

```
①②③④⑤
0 1:1 0:2 0:3 0:4 0:5
75 0:1
150 1:1 1:2 1:3
225 0:1
300 1:1 0:2 0:3 1:4 1:5
375 0:1
450 1:1 1:2 1:3
525 0:1
600 1:1 0:2 0:3 0:4 0:5
```

Example

```
* EXAMPLE OF U-ELEMENT DIGITAL OUTPUT
VOUT carry_out GND PWL 0N 0V 10N 0V 11N 5V 19N 5V 20N 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)
```

```
.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
*
.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
X2 A[1] B[1] carry-out_1 C[1] carry-out_2 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
UA[1] A[1] VLD2A VHD2A D2A SIGNAME=3 IS=1
UB[0] B[0] VLD2A VHD2A D2A SIGNAME=4 IS=1
UB[1] B[1] VLD2A VHD2A D2A SIGNAME=5 IS=1
```

```
*
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
```

```
VREFA2D VREFA2D 0 DC 0.0V
.END
```

Figure 5-10: Digital Stimulus File Input
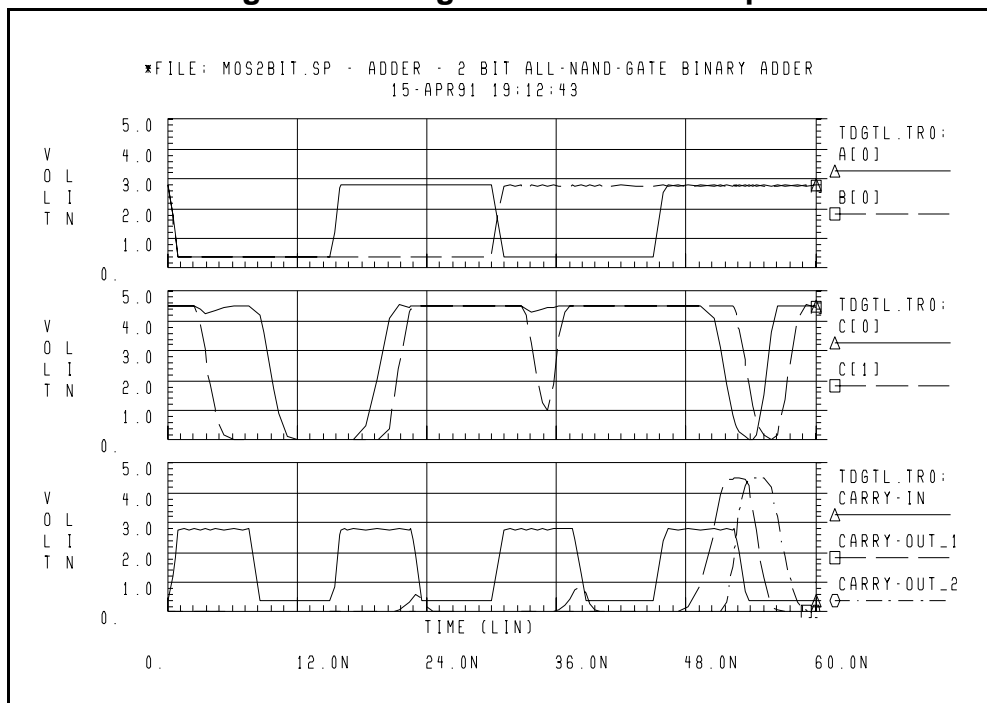
Specifying a Digital Vector File

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:

```
radix 1111 1111
vname a b c d e f g h
io iiii iiii
tunit ns
```

Keywords such as *radix*, *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:

```
trise 0.3 137F 0000
tfall 0.5 137F 0000
vih 5.0 137F 0000
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 *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:

```
time1 signal1_value1 signal2_value1 signal3_value1...
time2 signal1_value2 signal2_value2 signal3_value2...
time3 signal1_value3 signal2_value3 signal3_value3...
.
.
```

The set of values for a particular signal over all times is a *vector*, a vertical column in the tabular data and vector table. Thus, the set of all *signal1_valuex* constitute one vector. Signal values may have the legal states described in the following section.

Rows in the tabular data section must appear in chronological order because row placement carries sequential timing information.

Example

```
10.0 1000 0000
15.0 1100 1100
20.0 1010 1001
30.0 1001 1111
```

This example feature eight signals and therefore eight vectors. The first signal (starting from the left) has a vector [1 1 1 1]; the second has a vector [0 1 0 0]; and so on.

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:

```
...
; 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

```
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 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 iiii iiii
tunit 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.

# bits	Radix	Number System	Valid Digits
1	2	Binary	0, 1
2	4	–	0 – 3
3	8	Octal	0 – 7
4	16	Hexadecimal	0 – F

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
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:

```
radix 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.

```
radix 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:

```
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.

```
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 *tdelay* 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.

```
radix 1 1 4 1234 11111111
io i i o iiib iiiiiiii
vname V1 V2 VX[3:0] V4 V5[1:0] V6[0:2] V7[0:3]
+V8 V9 V10 V11 V12 V13 V14 V15
tdelay 1.0
tdelay -1.2 0 1 1 0000 00000000
tdelay 1.5 0 0 0 1370 00000000
idelay 2.0 0 0 0 000F 00000000
odelay 3.0 0 0 0 000F 00000000
```

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
tfall 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 overrule 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:

```
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*:

```
v1 V1 0 pwl(......)
```

as will *v2*, *v3*, *v4*, *v5*, *v6*, and *v7*. However, *v8* will be realized by

```
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 *vth* statements have no effect on the input signals.

Voh Statement

The *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 *voh* statement, Star-Hspice assumes 3.3. If you apply more than one *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 *voh* statements have no effect on input signals.

Note: *If both voh and vol are not defined, Star-Hspice uses vth (default or defined).*

Vol Statement

The *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 *vol* statement, Star-Hspice assumes 0.0. If you apply more than one *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-Hspice uses vth (default or defined)*

Avant!

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`), 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, DC, steady-state frequency-domain (`.SSFD`) and RF AC/noise (`.RF_AC`, `.RF_NOISE` and `.NOISENPT`) analyses.

In particular, the S parameter in S Element represents the generalized scattering parameter $\mathbf{S}$ for a multi-terminal network, which is defined as:

$$\mathbf{v_{ref}} = \mathbf{S} \cdot \mathbf{v_{inc}} \ .$$

where boldface lower-case and upper-case symbols denote vectors and matrices, respectively. $\mathbf{v_{inc}}$ and $\mathbf{v_{ref}}$ 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:

$$\mathbf{Y} = \mathbf{Y_r}(\mathbf{I} - \mathbf{S})(\mathbf{I} + \mathbf{S})^{-1} .$$

where $\mathbf{Y_r}$ is the characteristic admittance matrix of the reference system, which is related to the characteristic impedance matrix $\mathbf{Z_r}$ by:

$$\mathbf{Y_r} = \mathbf{Z_r}^{-1} .$$

Similarly, Y parameter can be converted to S parameter as follows:

$$\mathbf{S} = (\mathbf{Y_r} + \mathbf{Y})^{-1}(\mathbf{Y_r} - \mathbf{Y}) .$$

Figure 6-1: Terminal Node Notation
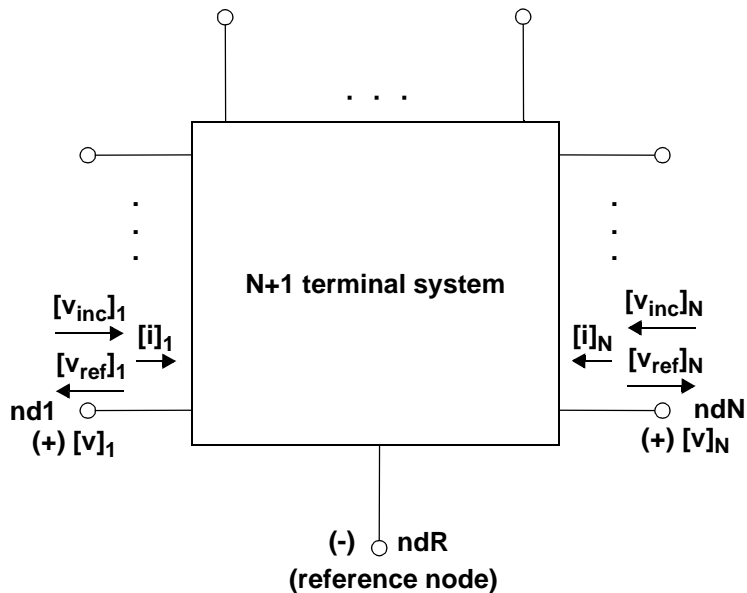
Syntax

The syntax of the S Element is:

```
Sxxx nd1 nd2 ... ndN ndR FQMODEL=name [TYPE=val Zo=val Zof=name]
```

nd1 nd2 ... ndN	N signal nodes (see Figure 6-1).
ndR	Reference node.
FQMODEL	.MODEL statement of type sp, which defines the frequency behavior of S or Y parameter.
TYPE	Parameter type: S: scattering parameter (default) 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 Ω.

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:

```
.MODEL 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	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.
VALTYPE	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.
INFINITY	Data point at infinity. Typically real-valued. Data format must be consistent with MATRIX and VALTYPE specifications. This point is not counted in npts.
DC	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:
	STEP: piecewise step. Default.
	LINEAR: piecewise linear

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:

```
.MODEL name sp SPACING=LIN [N=dim]
+ FSTART=f0 DF=f1 DATA=npts d1 d2 ...
```

Syntax for OCT or DEC spacing:

```
.MODEL name sp SPACING=DEC or OCT
+ [N=dim] FSTART=f0
+ NI=n_per_intval DATA=npts d1
+ d2 ...
```

Syntax for POI spacing:

```
.MODEL name sp SPACING=NONUNIFORM
+ [N=dim] DATA=npts f1 d1 f2 d2
+ ...
```

`DATAFILE`	This option allows users to specify data points in an external file. The content of this file must be only raw numbers without any suffixes, comments or continuation letters. The order of data must follow the `DATA` statement.
	This feature is useful as this data file does not have limitation on line length so users can enter a large number of data points.

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

Four-conductor line

R_o, L, G_o, C, R_s, G_d

v_1

Reference conductor

l
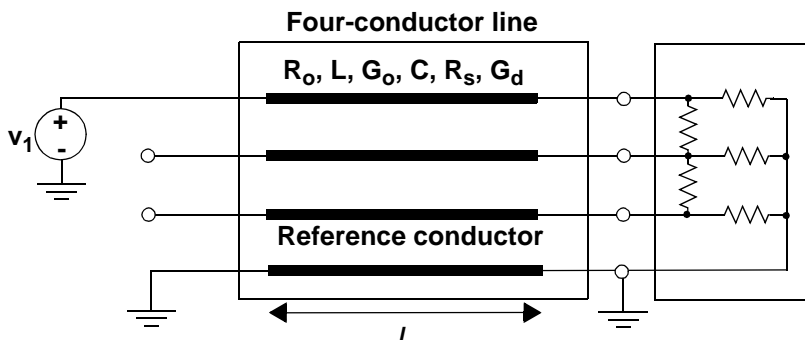
Table 6-1: Input File Listing

| Header, options, and sources | ```
*S parameter ex1: x-line with a resistive
+ termination
.OPTIONS POST
V1 i1 0 ac=1v
``` |
| Analysis | ```
.AC lin 500 0Hz 30MegHz
.DC v1 0v 5v 1v
``` |
| Transmission line (W element) | ```
W1 i1 i2 i3 0 o1 o2 o3 0 RLGCMODEL=wrlgc N=3 L=0.97
.MODEL wrlgc sp W 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
+ -0.270166 0.0
+  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

| Frequency model definition | ```
.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
``` |
|---|---|
| Using capacitive elements | ```
.SUBCKT terminator n1 n2 n3 ref
 C1  n1 ref 10pF
 C2  n2 ref 10pF
 C3  n3 ref 10pF
 C12 n1 n2  2pF
 C23 n2 n3  2pF
.ENDS terminator
``` |

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

| Header, options and sources | ```
*S parameter ex3: modeling x-line using
+ S parameter
.OPTIONS POST
vin in0 0  ac=1
``` |
| Analysis | ```
.AC lin 100 0 1000meg
.DC vin 0 1v 0.2v
``` |
| Transmission line | ```
X1  in1 in2 out1 out2 0 x-line
``` |
| Termination | ```
R1  in0   in1   28
R2  in2   0     28
R3  out1  0     28
R4  out2  0     28
``` |
| W Element RLGC model definition | ```
.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
``` |
| Frequency model definition | ```
.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 0 0.996135 0 0 0 0.00386491 0
+ -0.0492864 -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
+ ...
``` |

Equivalent S parameter element	``` .SUBCKT terminator n1 n2 n3 ref S1 n1 n2 n3 ref FQMODEL=SM2 .ENDS terminator ```

Avant!

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:

Simple assignment	.PARAM <SimpleParam> = 1e-12
Algebraic definition	.PARAM <AlgebraicParam> = 'SimpleParam*8.2'
User-defined function	.PARAM <MyFunc(x, y)> = 'Sqrt((x*x)+(y*y))'
Subcircuit default	.SUBCKT <SubName> <ParamDefName> = + <Value>
Predefined analysis function	.PARAM <mcVar> = Agauss(1.0,0.1) (see "Statistical Analysis and Optimization" on page 13-1)
.MEASURE statement	.MEASURE <DC | AC | TRAN> result TRIG ... + TARG ... <GOAL = val> <MINVAL = val> + <WEIGHT = val> <MeasType> <MeasParam> (see "Specifying User-Defined Analysis (.MEASURE)" on page 8-35)

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

.OPTION PARHIER = GLOBAL	.OPTION PARHIER = LOCAL
Analysis sweep parameters	Analysis sweep parameters
.PARAM statement (library)	.SUBCKT call (instance)
.SUBCKT call (instance)	.SUBCKT definition (symbol)
.SUBCKT definition (symbol)	.PARAM statement (library)

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 = 1g
  rTerm Bit0 0 TermValue
```

```
    rTerm Bit1 0 TermValue
     ...
```

Expression Example
```
.PARAM Pi = '355/113'
.PARAM Pi2 = '2*Pi'

.PARAM npRatio = 2.1
.PARAM nWidth = 3u
.PARAM pWidth = 'nWidth * npRatio'
Mp1 ... <pModelName> W = pWidth
Mn1 ... <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
...
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
...
```

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
```

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 8-35.

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 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[1]

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

HSPICE Form	Function	Class	Description
sin(x)	sine	trig	Returns the sine of x (radians)
cos(x)	cosine	trig	Returns the cosine of x (radians)
tan(x)	tangent	trig	Returns the tangent of x (radians)
asin(x)	arc sine	trig	Returns the inverse sine of x (radians)
acos(x)	arc cosine	trig	Returns the inverse cosine of x (radians)
atan(x)	arc tangent	trig	Returns the inverse tangent of x (radians)
sinh(x)	hyperbolic sine	trig	Returns the hyperbolic sine of x (radians)
cosh(x)	hyperbolic cosine	trig	Returns the hyperbolic cosine of x (radians)
tanh(x)	hyperbolic tangent	trig	Returns the hyperbolic tangent of x (radians)
abs(x)	absolute value	math	Returns the absolute value of x: |x|
sqrt(x)	square root	math	Returns the square root of the absolute value of x: sqrt(-x) = -sqrt(|x|)
pow(x,y)	absolute power	math	Returns the value of x raised to the integer part of y: $x^{(\text{integer part of } y)}$

Table 7-2: Star-Hspice Built-in Functions

HSPICE Form	Function	Class	Description
pwr(x,y)	signed power	math	Returns the absolute value of x raised to the y power, with the sign of x: (sign of x)$\|x\|^y$
log(x)	natural logarithm	math	Returns the natural logarithm of the absolute value of x, with the sign of x: (sign of x)log($\|x\|$)
log10(x)	base 10 logarithm	math	Returns the base 10 logarithm of the absolute value of x, with the sign of x: (sign of x)$\log_{10}(\|x\|)$
exp(x)	exponential	math	Returns *e* raised to the power x: e^x
db(x)	decibels	math	Returns the base 10 logarithm of the absolute value of x, multiplied by 20, with the sign of x: (sign of x)$20\log_{10}(\|x\|)$
int(x)	integer	math	Returns the largest integer less than or equal to x
sgn(x)	return sign	math	Returns -1 if x is less than 0, 0 if x is equal to 0, and 1 if x is greater than 0
sign(x,y)	transfer sign	math	Returns the absolute value of x, with the sign of y: (sign of y)$\|x\|$
min(x,y)	smaller of two args	control	Returns the numeric minimum of x and y
max(x,y)	larger of two args	control	Returns the numeric maximum of x and y
lv(<Element>) or lx(<Element>)	element templates	various	Returns various element values during simulation. See "Element Template Output" in Chapter 7 for more information.

Table 7-2: Star-Hspice Built-in Functions

HSPICE Form	Function	Class	Description
v(<Node>), i(<Element>)...	circuit output variables	various	Returns various circuit values during simulation. See "DC and Transient Output Variables" on page 8-19 for more information.

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

HSPICE Form	Function	Class	Description
time	current simulation time	control	Parameterizes the current simulation time during transient analysis.
temper	current circuit temperature	control	Parameterizes the current simulation temperature during transient/ temperature analysis.
hertz	current simulation frequency	control	Parameterizes the frequency during AC analysis.

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

```
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) = POW(a,2)+a*b g(d) = 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 *Tau* as a parameter to control one or more

subcircuits in their library. Now suppose that the second vendor uses *Tau* to control a different aspect of their library. Setting a global parameter named *Tau* to control one library also modifies the behavior of the second library, which might not be the intent.

When the *scope* of a higher level parameter is *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 *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!
```

```
x2 A Y2 Inv Wid = 1u           $ Incorrect! Both x1 and x2
                               $ simulate with mp1 = 10u and
                               $ mn1 = 5u instead of 2u and 1u.
```

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 *Wid* overrides the
subcircuit default.

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

Method	Parameter Location	Pros	Cons
Local	On a .SUBCKT definition line	The library is protected from global circuit parameter definitions unless the user wishes to override. Single location for default values.	Cannot be used with releases of Star-Hspice prior to Release 95.1.
Global	At the global level and on .SUBCKT definition lines	Works with older Star-Hspice versions	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.
Special	.OPTION DEFW statement	Simple to do	Third party libraries or other sections of the design might depend on the option DEFW.

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


```
TEST OF PARHIER
.OPTION list node post = 2
+ ingold = 2
+ parhier = <Local|Global>
.PARAM Val = 1
x1 n0 0 Sub1
.SubCkt Sub1 n1 n2 Val = 1
    r1 n1 n2 Val
    x2 n1 n2 Sub2
.Ends Sub1
.SubCkt Sub2 n1 n2 Val = 2
    r2 n1 n2 Val
    x3 n1 n2 Sub3
.Ends Sub2
.SubCkt Sub3 n1 n2 Val = 3
    r3 n1 n2 Val
.Ends Sub3
```

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

Element	PARHIER = Local	PARHIER = Global
r1	1.0	1.0
r2	2.0	1.0
r3	3.0	1.0

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.

Avant!

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.

Output Statement	Description
.PRINT	Prints numeric analysis results in the output listing file (and post-processor data if .OPTION POST is used).
.PLOT	Generates low-resolution (ASCII) plots in the output listing file (and post-processor data if .OPTION POST is used).
.GRAPH	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.
.PROBE	Outputs data to post-processor output files but not to the output listing (used with .OPTION PROBE to limit output).
.MEASURE	Prints to output listing file the results of specific user-defined analyses (and post-processor data if .OPTION POST is used).

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.

Example

```
.PRINT 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.

```
.PRINT 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.
ov1 …	Output variables to plot. These are voltage, current, or element template variables from a DC, AC, TRAN, NOISE, or DISTO analysis. See the following sections for syntax.
plo1,phi1 …	Lower and upper plot limits. Each output variable is plotted using the first set of plot limits following the output variable. Output variables following a plot limit should have a new plot limit. For example, to plot all output variables with the same scale, specify one set of plot limits at the end of the PLOT statement. Setting the plot limits to (0,0) causes Star-Hspice to set the plot limits.

Example

In the following example, PAR invokes the plot of the ratio of the collector current and the base current of the transistor Q1.

```
.PLOT DC V(4) V(5) V(1) PAR(`I1(Q1)/I2(Q1)')
.PLOT TRAN V(17,5) (2,5) I(VIN) V(17) (1,9)
.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.

```
.PLOT AC ZIN YOUT(P) S11(DB) S12(M) Z11(R)
.PLOT DISTO HD2 HD3(R) SIM2
.PLOT TRAN V(5,3) V(4) (0,5) V(7) (0,10)
.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. Pnam*n* is one of the model parameters of a .GRAPH statement, and val*n* is the value of pnam*n*. Val*n* can be a number of parameter.

Example

```
.GRAPH DC  cgb = 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

Name (Alias)	Default	Description
FREQ	0.0	Plots symbol frequency. Value 0 suppresses plot symbol generation; a value of n generates a plot symbol every n points.
MONO	0.0	Monotonic option. MONO = 1 automatically resets x-axis if any change in x direction.
TIC	0.0	Shows tick marks
XGRID, YGRID	0.0	Setting to 1.0 turns on the axis grid lines
XMIN, XMAX	0.0	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.

XSCAL	1.0	Scale for the x-axis. Two common axis scales are:
		    Linear(LIN) (XSCAL = 1)
		    Logarithm(LOG) (XSCAL = 2)

YMIN, YMAX	0.0	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.

YSCAL	1.0	Scale for the y-axis. Two common axis scales are:
		    Linear(LIN) (YSCAL = 1)
		    Logarithm(LOG) (YSCAL = 2)

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:

```
F = 1e-15        M = 1e-3
P = 1e-12        K = 1e3
N = 1e-9         X = 1e6
U = 1e-6         G = 1e9
```

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 INGOLD = 1 or 2 with an .OPTION statement.

INGOLD = 0 ***(default)***	Engineering Format	1.234K 123M
INGOLD = 1	G Format (fixed and exponential)	1.234e+03 .123
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.

POST = 0,1,BINARY   Output format is binary

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 (default)	Is the same as ACCT with no argument
.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.

SETUP	Constructs a sparse matrix pointer system
TOTAL JOB TIME	Total amount of CPU time required to process the simulation. This is not the length of actual (clock) time that was taken, and may differ slightly from run to run, even if the runs are identical.

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

```
.GLOBAL vdd vss
X1 1 2 3 nor2
X2 3 4 5 nor2
.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.

```
   .PRINT v(1) v(X1.A) $ Print statement 2
```

The print statement above specifies two ways of printing the voltage on input A of instance X1.

```
   .PRINT v(3) v(X1.Y) v(X2.A) $ Print statement 3
```

This print statement specifies three different ways of printing the voltage at output Y of instance X1. (input A of instance X2).

```
   .PRINT v(X2.N1) $ Print statement 4
```

The print statement above prints out the voltage on the internal node N1 of instance X2.

```
   .PRINT  i(X1.M1) $ Print statement 5
```

The print statement above prints out the drain-to-source current through MOSFET M1 in instance X1.

Example 2
```
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)	Local node to subcircuit definition YYY, called by subcircuit X1
V(X1.X2.56)	Local node to subcircuit definition 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 sub-circuits 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 8-12.

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:

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

Example

```
.PLOT TRAN I(VIN)
.PRINT DC I(X1.VSRC)
.PLOT DC I(XSUB.XSUBSUB.VY)
```

Current: Element Branches

Syntax

```
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)

Figure 8-2: Capacitor (node1, node2); Inductor (node 1, node2)

Figure 8-3: Diode (node1, node2)

I1 (D1)   node1 (anode, P-type, + node)

I2 (D2)   node2 (cathode, N-type, -node)

Figure 8-4: JFET (node1, node2, node3) - n-channel

node1 (drain node)
I1 (J1)

node2
(gate node)
I2 (J1)

node3 (source node)
I3 (J1)

Figure 8-5: BJT (node1, node2, node3, node4) - npn

node1 (drain node)

I1 (M1)

node4 (substrate node)

node2 (gate node)

I2 (M1)   I4 (M1)

node3 (source node)

I3 (M1)

Figure 8-6: MOSFET (node1, node2, node3, node4) - n-channel

Power Output

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

```
.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

$$Pd = Vpp' \cdot (Ido + Icap) + Vp'n \cdot Ido$$

Pd	Power dissipated in diode
Ido	DC component of the diode current
Icap	Capacitive component of the diode current
Vp'n	Voltage across the junction
Vpp'	Voltage across the series resistance RS

BJT Power Dissipation

Vertical

$$Pd = Vc'e' \cdot Ico + Vb'e' \cdot Ibo + Vcc' \cdot Ictot + Vee' \cdot Ietot + \\ Vsc' \cdot Iso - Vcc' \cdot Istot$$

Lateral

$$Pd = Vc'e' \cdot Ico + Vb'e' \cdot Ibo + Vcc' \cdot Ictot + Vbb' \cdot Ibtot + Vee' \cdot Ietot + Vsb' \cdot Iso - Vbb' \cdot Istot$$

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 RB
Vc'e'	Voltage across the collector-emitter terminals
Vcc'	Voltage across the series collector resistance RC
Vee'	Voltage across the series emitter resistance RE
Vsb'	Voltage across the substrate-base junction
Vsc'	Voltage across the substrate-collector junction

JFET Power Dissipation

$$Pd = Vd's' \cdot Ido + Vgd' \cdot Igdo + Vgs' \cdot Igso + Vs's \cdot (Ido + Igso + Icgs) + Vdd' \cdot (Ido - Igdo - Icgd)$$

Icgd	Capacitive component of the gate-drain junction current
Icgs	Capacitive component of the gate-source junction current
Ido	DC component of the drain current
Igdo	DC component of the gate-drain junction current
Igso	DC component of the gate-source junction current
Pd	Power dissipated in JFET
Vd's'	Voltage across the internal drain-source terminals
Vdd'	Voltage across the series drain resistance RD
Vgd'	Voltage across the gate-drain junction
Vgs'	Voltage across the gate-source junction
Vs's	Voltage across the series source resistance RS

MOSFET Power Dissipation

$$Pd = Vd's' \cdot Ido + Vbd' \cdot Ibdo + Vbs' \cdot Ibso +$$
$$Vs's \cdot (Ido + Ibso + Icbs + Icgs) + Vdd' \cdot (Ido - Ibdo - Icbd - Icgd)$$

Ibdo	DC component of the bulk-drain junction current
Ibso	DC component of the bulk-source junction current
Icbd	Capacitive component of the bulk-drain junction current
Icbs	Capacitive component of the bulk-source junction current
Icgd	Capacitive component of the gate-drain current

Icgs	Capacitive component of the gate-source current
Ido	DC component of the drain current
Pd	Power dissipated in the MOSFET
Vbd'	Voltage across the bulk-drain junction
Vbs'	Voltage across the bulk-source junction
Vd's'	Voltage across the internal drain-source terminals
Vdd'	Voltage across the series drain resistance RD
Vs's	Voltage across the series source resistance RS

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 8-1: AC Output Variable Types

Type Symbol	Variable Type
DB	decibel
I	imaginary part
M	magnitude

Table 8-1: AC Output Variable Types

Type Symbol	Variable Type
P	phase
R	real part
T	group delay

Specify real or imaginary parts, magnitude, phase, decibels, and group delay for voltages and currents.

Nodal Voltage

Syntax

```
Vx (n1,<,n2>)
```

where:

x	Specifies the voltage output type (see Table 8-1)
n1, n2	Specifies node names. If n2 is omitted, ground (node 0) is assumed.

Example

```
.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.

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) =  REAL [V(N1,0)] - REAL [V(N2,0)]
VI(N1,N2) =  IMAG [V(N1,0)] - IMAG [V(N2,0)]
```

Magnitude:
```
VM(N1,0) =  [VR(N1,0)² + VI(N1,0)²]⁰·⁵
```
$$VM(N1,0) = [VR(N1,0)^2 + VI(N1,0)^2]^{0.5}$$
$$VM(N2,0) = [VR(N2,0)^2 + VI(N2,0)^2]^{0.5}$$
```
VM(N1,N2) =  VM(N1,0) - VM(N2,0)
```

Phase:
```
VP(N1,0) =  ARCTAN[VI(N1,0)/VR(N1,0)]
VP(N2,0) =  ARCTAN[VI(N2,0)/VR(N2,0)]
VP(N1,N2) =  VP(N1,0) - VP(N2,0)
```

Decibel:
```
VDB(N1,N2) =  20·LOG10(VM(N1,0)/VM(N2,0))
```

SPICE Method (ACOUT = 0)

Real and imaginary:
```
VR(N1,N2) =  REAL [V(N1,0) - V(N2,0)]
VI(N1,N2) =  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) = ARCTAN[VI(N1,N2)/VR(N1,N2)]

Decibel:

   VDB(N1,N2) = 20·LOG10[VM(N1,N2)]

Current: Independent Voltage Sources

Syntax

Iz (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

Izn (Wwww)

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.
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, IM3(X1.Wwww).

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

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

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

Syntax
```
.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
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
```
ovar <(z)>
```

where:

ovar	Noise and distortion analysis parameter. It can be either ONOISE (output noise), or INOISE (equivalent input noise) or any of the distortion analysis parameters (HD2, HD3, SIM2, DIM2, DIM3).
z	Output type (only for distortion). If z is omitted, the magnitude of the output variable is output.

Example
```
.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).
```
.PLOT NOISE INOISE ONOISE
```

> *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
```

Elname	Name of the element
Property	Property name of an element, such as a user-input parameter, state variable, stored charge, capacitance current, capacitance, or derivative of a variable

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
.PRINT TRAN X2.M1:CGGBO M1:CGDBO X2.M1:CGSBO
```

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 9-48 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.

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 = length/6 = 100u
```

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 $ERRfun = (GOAL - result)/GOAL$.
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.
TARG	Indicates the beginning of the target signal specification

targ_val	Specifies the value of the *targ_var* at which the counter for crossing, rises, or falls is incremented by one
targ_var	Name of the output variable whose propagation delay is determined with respect to the *trig_var*
time_delay	Amount of simulation time that must elapse before the measurement is enabled. The number of crossings, rises, or falls is counted only after *time_delay* value. The default trigger delay is zero.
CROSS = c *RISE = r* *FALL = f*	The numbers indicate which occurrence of a CROSS, FALL, or RISE event causes a measurement to be performed. For RISE = r, the WHEN condition is met and measurement is performed when the designated signal has risen *r* rise times. For FALL = f, measurement is performed when the designated signal has fallen *f* fall times. A crossing is either a rise or a fall, so for CROSS = c, measurement is performed when the designated signal has achieved a total of *c* crossing times, as a result of either rising or falling. For TARG, the last event is specified with the LAST keyword.
LAST	Measurement is performed when the last CROSS, FALL, or RISE event occurs. For CROSS = LAST, measurement is performed the last time the WHEN condition is true for either a rising or falling signal. For FALL = LAST, measurement is performed the last time the WHEN condition is true for a falling signal. For RISE = LAST, measurement is performed the last time the WHEN condition is true for a rising signal. LAST is a reserved word and cannot be chosen as a parameter name in the above .MEASURE statements.

> $AT = val$ Special case for trigger specification. The "val" is the time for TRAN analysis, the frequency for AC analysis, or the parameter for DC analysis, at which measurement is to start.

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.

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 8-4 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

CROSS = c *RISE = r* *FALL = f*	The numbers indicate which occurrence of a CROSS, FALL, or RISE event causes a measurement to be performed. For RISE = r, the WHEN condition is met and measurement is performed when the designated signal has risen *r* rise times. For FALL = f, measurement is performed when the designated signal has fallen *f* fall times. A crossing is either a rise or a fall, so for CROSS = c, measurement is performed when the designated signal has achieved a total of *c* crossing times, as a result of either rising or falling.
<DC/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 $ERRfun = (GOAL - result)/GOAL.$
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 the last time the WHEN condition is true for a falling signal. For RISE = LAST, measurement is performed the last time the WHEN condition is true for a rising signal. LAST is a reserved word and cannot be chosen as a parameter name in the above .MEASURE statements.
MINVAL	If the absolute value of GOAL is less than MINVAL, the GOAL value is replaced by MINVAL in the denominator of the ERRfun expression. Default = 1.0e-12.

out_var(1,2,3)	Variables used to establish conditions at which measurement is to take place
result	Name associated with the measured value in the Star-Hspice output
TD	Identifies the time at which measurement is to start
WEIGHT	Calculated error is multiplied by the weight value. Default = 1.0.
WHEN	Selects the WHEN function

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'
+ <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

```
.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 $ERRfun = (GOAL - result)/GOAL$
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_va*r 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.

```
.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 *RISE = r* *FALL = f*	The numbers indicate which occurrence of a CROSS, FALL, or RISE event causes a measurement to be performed. For RISE = r, the WHEN condition is met and measurement is performed when the designated signal has risen *r* rise times. For FALL = f, measurement is performed when the designated signal has fallen *f* fall times. A crossing is either a rise or a fall, so for CROSS = c, measurement is performed when the designated signal has achieved a total of *c* crossing times, as a result of either rising or falling.
<DC|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 $\text{ERRfun} = (\text{GOAL} - \text{result}) / \text{GOAL}.$

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:

```
.MEAS TRAN slew DERIV v(1) WHEN v(1) = '0.90*vdd'
```

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

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

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

.MEASURE <DC|AC|TRAN> result ERRfun meas_var calc_var <MINVAL = val>
+ < IGNORE | YMIN = val> <YMAX = val> <WEIGHT = val> <FROM = val>
+ <TO = val>

where:

<DC|AC|TRAN>	Specifies the analysis type of the measurement. If omitted, the last analysis mode requested is assumed.
result	Name associated with the measured result in the output
ERRfun	ERRfun indicates which error function to use: ERR, ERR1, ERR2, or ERR3.
meas_var	Name of any output variable or parameter in the data statement. M denotes the *meas_var* in the error equation.
calc_var	Name of the simulated output variable or parameter in the .MEASURE statement to be compared with *meas_var*. C denotes the *calc_var* in the error equation.

IGNOR|YMIN	If the absolute value of *meas_var* is less than IGNOR value, then this point is not considered in the ERRfun calculation. Default = 1.0e-15.
FROM	Specifies the beginning of the ERRfun calculation. For transient analysis, the from value is in units of time. Defaults to the first value of the sweep variable.
WEIGHT	The calculated error is multiplied by the weight value. Default = 1.0.
YMAX	If the absolute value of *meas_var* is greater than the YMAX value, then this point is not considered in the ERRfun calculation. Default = 1.0e+15.
TO	Specifies the end of the ERRfun calculation. Defaults to the last value of the sweep variable.
MINVAL	If the absolute value of *meas_var* is less than MINVAL, the *meas_var* value is replaced by MINVAL in the denominator of the ERRfun expression. Default = 1.0e-12.

Error Equations

ERR

ERR sums the squares of (M-C)/max (M, 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.

$$ERR = \left[\frac{1}{NPTS} \cdot \sum_{i=1}^{NPTS} \left(\frac{M_i - C_i}{\max(MINVAL, M_i)} \right)^2 \right]^{1/2}$$

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).

$$ERR1_i = \frac{M_i - C_i}{\max(MINVAL, M_i)}, i = 1, NPTS$$

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.

$$ERR2_i = \left| \frac{M_i - C_i}{\max(MINVAL, M_i)} \right|, i = 1, NPTS$$

The returned value printed for ERR2 is

$$ERR = \frac{1}{NPTS} \cdot \sum^{NPTS} ERR2_i$$

ERR3

$$ERR3_i = \frac{\pm \log \left| \frac{M_i}{C_i} \right|}{\left| \log[\max(MINVAL, |M_i|)] \right|}, i = 1, NPTS$$

The + and - signs correspond to a positive and negative M/C ratio, respectively.

> ***Note:*** *If the measured value M is less than MINVAL, the MINVAL is used instead. Also, if the absolute value of M is less than the IGNOR | YMIN value or greater than the YMAX value, then this point is not considered in the error calculation.*

Element Template Listings

Resistor

Name	Alias	Description
G	LV1	Conductance at analysis temperature
R	LV2	Resistance at reference temperature
TC1	LV3	First temperature coefficient
TC2	LV4	Second temperature coefficient

Capacitor

Name	Alias	Description
CEFF	LV1	Computed effective capacitance
IC	LV2	Initial condition
Q	LX0	Charge stored in capacitor
CURR	LX1	Current flowing through capacitor
VOLT	LX2	Voltage across capacitor
–	LX3	Capacitance (not used in Star-Hspice releases after 95.3)

Inductor

Name	Alias	Description
LEFF	LV1	Computed effective inductance
IC	LV2	Initial condition
FLUX	LX0	Flux in the inductor

Inductor

Name	Alias	Description
VOLT	LX1	Voltage across inductor
CURR	LX2	Current flowing through inductor
–	LX4	Inductance (not used in Star-Hspice releases after 95.3)

Mutual Inductor

Name	Alias	Description
K	LV1	Mutual inductance

Voltage-Controlled Current Source

Name	Alias	Description
CURR	LX0	Current through the source, if VCCS
R	LX0	Resistance value, if VCR
C	LX0	Capacitance value, if VCCAP
CV	LX1	Controlling voltage
CQ	LX1	Capacitance charge, if VCCAP
DI	LX2	Derivative of source current with respect to control voltage
ICAP	LX2	Capacitance current, if VCCAP
VCAP	LX3	Voltage across capacitance, if VCCAP

Voltage-Controlled Voltage Source

Name	Alias	Description
VOLT	LX0	Source voltage

Voltage-Controlled Voltage Source

Name	Alias	Description
CURR	LX1	Current through source
CV	LX2	Controlling voltage
DV	LX3	Derivative of source voltage with respect to control current

Current-Controlled Current Source

Name	Alias	Description
CURR	LX0	Current through source
CI	LX1	Controlling current
DI	LX2	Derivative of source current with respect to control current

Current-Controlled Voltage Source

Name	Alias	Description
VOLT	LX0	Source voltage
CURR	LX1	Source current
CI	LX2	Controlling current
DV	LX3	Derivative of source voltage with respect to control current

Independent Voltage Source

Name	Alias	Description
VOLT	LV1	DC/transient voltage
VOLTM	LV2	AC voltage magnitude

Independent Voltage Source

Name	Alias	Description
VOLTP	LV3	AC voltage phase

Independent Current Source

Name	Alias	Description
CURR	LV1	DC/transient current
CURRM	LV2	AC current magnitude
CURRP	LV3	AC current phase

Diode

Name	Alias	Description
AREA	LV1	Diode area factor
AREAX	LV23	Area after scaling
IC	LV2	Initial voltage across diode
VD	LX0	Voltage across diode (VD), excluding RS (series resistance)
IDC	LX1	DC current through diode (ID), excluding RS. Total diode current is the sum of IDC and ICAP
GD	LX2	Equivalent conductance (GD)
QD	LX3	Charge of diode capacitor (QD)
ICAP	LX4	Current through diode capacitor. Total diode current is the sum of IDC and ICAP.
C	LX5	Total diode capacitance
PID	LX7	Photo current in diode

BJT

Name	Alias	Description
AREA	LV1	Area factor
ICVBE	LV2	Initial condition for base-emitter voltage (VBE)
ICVCE	LV3	Initial condition for collector-emitter voltage (VCE)
MULT	LV4	Number of multiple BJTs
FT	LV5	FT (Unity gain bandwidth)
ISUB	LV6	Substrate current
GSUB	LV7	Substrate conductance
LOGIC	LV8	LOG 10 (IC)
LOGIB	LV9	LOG 10 (IB)
BETA	LV10	BETA
LOGBETAI	LV11	LOG 10 (BETA) current
ICTOL	LV12	Collector current tolerance
IBTOL	LV13	Base current tolerance
RB	LV14	Base resistance
GRE	LV15	Emitter conductance, 1/RE
GRC	LV16	Collector conductance, 1/RC
PIBC	LV18	Photo current, base-collector
PIBE	LV19	Photo current, base-emitter
VBE	LX0	VBE
VBC	LX1	Base-collector voltage (VBC)
CCO	LX2	Collector current (CCO)
CBO	LX3	Base current (CBO)
GPI	LX4	$g_\pi = {}^1ib / {}^1vbe$, constant vbc

BJT

Name	Alias	Description
GU	LX5	$g_\mu = {}^1ib\,/{}^1vbc$, constant vbe
GM	LX6	$g_m = {}^1ic\,/{}^1vbe + {}^1ic\,/{}^1vbe$, constant vce
G0	LX7	$g_0 = {}^1ic\,/{}^1vce$, constant vbe
QBE	LX8	Base-emitter charge (QBE)
CQBE	LX9	Base-emitter charge current (CQBE)
QBC	LX10	Base-collector charge (QBC)
CQBC	LX11	Base-collector charge current (CQBC)
QCS	LX12	Current-substrate charge (QCS)
CQCS	LX13	Current-substrate charge current (CQCS)
QBX	LX14	Base-internal base charge (QBX)
CQBX	LX15	Base-internal base charge current (CQBX)
GXO	LX16	1/Rbeff Internal conductance (GXO)
CEXBC	LX17	Base-collector equivalent current (CEXBC)
–	LX18	Base-collector conductance (GEQCBO) (not used in Star-Hspice releases after 95.3)
CAP_BE	LX19	cbe capacitance (C Π)
CAP_IBC	LX20	cbc internal base-collector capacitance (Cμ)
CAP_SCB	LX21	csc substrate-collector capacitance for vertical transistors csb substrate-base capacitance for lateral transistors
CAP_XBC	LX22	cbcx external base-collector capacitance
CMCMO	LX23	^{1}(TF*IBE) $/{}^1vbc$
VSUB	LX24	Substrate voltage

JFET

Name	Alias	Description
AREA	LV1	JFET area factor
VDS	LV2	Initial condition for drain-source voltage
VGS	LV3	Initial condition for gate-source voltage
PIGD	LV16	Photo current, gate-drain in JFET
PIGS	LV17	Photo current, gate-source in JFET
VGS	LX0	VGS
VGD	LX1	Gate-drain voltage (VGD)
CGSO	LX2	Gate-to-source (CGSO)
CDO	LX3	Drain current (CDO)
CGDO	LX4	Gate-to-drain current (CGDO)
GMO	LX5	Transconductance (GMO)
GDSO	LX6	Drain-source transconductance (GDSO)
GGSO	LX7	Gate-source transconductance (GGSO)
GGDO	LX8	Gate-drain transconductance (GGDO)
QGS	LX9	Gate-source charge (QGS)
CQGS	LX10	Gate-source charge current (CQGS)
QGD	LX11	Gate-drain charge (QGD)
CQGD	LX12	Gate-drain charge current (CQGD)
CAP_GS	LX13	Gate-source capacitance
CAP_GD	LX14	Gate-drain capacitance
–	LX15	Body-source voltage (not used in Star-Hspice releases after 95.3)
QDS	LX16	Drain-source charge (QDS)

JFET

Name	Alias	Description
CQDS	LX17	Drain-source charge current (CQDS)
GMBS	LX18	Drain-body (backgate) transconductance (GMBS)

MOSFET

Name	Alias	Description
L	LV1	Channel length (L)
W	LV2	Channel width (W)
AD	LV3	Area of the drain diode (AD)
AS	LV4	Area of the source diode (AS)
ICVDS	LV5	Initial condition for drain-source voltage (VDS)
ICVGS	LV6	Initial condition for gate-source voltage (VGS)
ICVBS	LV7	Initial condition for bulk-source voltage (VBS)
–	LV8	Device polarity: 1 = forward, -1 = reverse (not used in Star-Hspice releases after 95.3)
VTH	LV9	Threshold voltage (bias dependent)
VDSAT	LV10	Saturation voltage (VDSAT)
PD	LV11	Drain diode periphery (PD)
PS	LV12	Source diode periphery (PS)
RDS	LV13	Drain resistance (squares) (RDS)
RSS	LV14	Source resistance (squares) (RSS)
XQC	LV15	Charge sharing coefficient (XQC)
GDEFF	LV16	Effective drain conductance (1/RDeff)
GSEFF	LV17	Effective source conductance (1/RSeff)

MOSFET

Name	Alias	Description
IDBS	LV18	Drain-bulk saturation current at -1 volt bias
ISBS	LV19	Source-bulk saturation current at -1 volt bias
VDBEFF	LV20	Effective drain bulk voltage
BETAEFF	LV21	BETA effective
GAMMAEFF	LV22	GAMMA effective
DELTAL	LV23	ΔL (MOS6 amount of channel length modulation) (only valid for LEVELs 1, 2, 3 and 6)
UBEFF	LV24	UB effective (only valid for LEVELs 1, 2, 3 and 6)
VG	LV25	VG drive (only valid for LEVELs 1, 2, 3 and 6)
VFBEFF	LV26	VFB effective
–	LV31	Drain current tolerance (not used in Star-Hspice releases after 95.3)
IDSTOL	LV32	Source diode current tolerance
IDDTOL	LV33	Drain diode current tolerance
COVLGS	LV36	Gate-source overlap capacitance
COVLGD	LV37	Gate-drain overlap capacitance
COVLGB	LV38	Gate-bulk overlap capacitance
VBS	LX1	Bulk-source voltage (VBS)
VGS	LX2	Gate-source voltage (VGS)
VDS	LX3	Drain-source voltage (VDS)
CDO	LX4	DC drain current (CDO)
CBSO	LX5	DC source-bulk diode current (CBSO)
CBDO	LX6	DC drain-bulk diode current (CBDO)

MOSFET

Name	Alias	Description
GMO	LX7	DC gate transconductance (GMO)
GDSO	LX8	DC drain-source conductance (GDSO)
GMBSO	LX9	DC substrate transconductance (GMBSO)
GBDO	LX10	Conductance of the drain diode (GBDO)
GBSO	LX11	Conductance of the source diode (GBSO)
Meyer and Charge Conservation Model Parameters		
QB	LX12	Bulk charge (QB)
CQB	LX13	Bulk charge current (CQB)
QG	LX14	Gate charge (QG)
CQG	LX15	Gate charge current (CQG)
QD	LX16	Channel charge (QD)
CQD	LX17	Channel charge current (CQD)
CGGBO	LX18	$\mathrm{CGGBO} = \partial Qg/\partial Vgb$ = CGS + CGD + CGB
CGDBO	LX19	$\mathrm{GDBO} = \partial Qg/\partial Vdb,$ (for Meyer CGD = -CGDBO)
CGSBO	LX20	$\mathrm{GSBO} = \partial Qg/\partial Vsb,$ (for Meyer CGS = -CGSBO)
CBGBO	LX21	$\mathrm{BGBO} = \partial Qb/\partial Vgb,$ (for Meyer CGB = -CBGBO)
CBDBO	LX22	$\mathrm{BDBO} = \partial Qb/\partial Vdb$
CBSBO	LX23	$\mathrm{BSBO} = \partial Qb/\partial Vsb$
QBD	LX24	Drain-bulk charge (QBD)
–	LX25	Drain-bulk charge current (CQBD) (not used in Star-Hspice releases after 95.3)

MOSFET

Name	Alias	Description
QBS	LX26	Source-bulk charge (QBS)
–	LX27	Source-bulk charge current (CQBS) (not used in Star-Hspice releases after 95.3)
CAP_BS	LX28	Bulk-source capacitance
CAP_BD	LX29	Bulk-drain capacitance
CQS	LX31	Channel charge current (CQS)
CDGBO	LX32	$\text{DGBO} = \partial Qd/\partial Vgb$
CDDBO	LX33	$\text{DDBO} = \partial Qd/\partial Vdb$
CDSBO	LX34	$\text{DSBO} = \partial Qd/\partial Vsb$

Saturable Core Element

Name	Alias	Description
MU	LX0	Dynamic permeability (mu) Weber/(amp-turn-meter)
H	LX1	Magnetizing force (H) Ampere-turns/meter
B	LX2	Magnetic flux density (B) Webers/meter^2

Saturable Core Winding

Name	Alias	Description
LEFF	LV1	Effective winding inductance (Henry)
IC	LV2	Initial condition
FLUX	LX0	Flux through winding (Weber-turn)
VOLT	LX1	Voltage across winding (Volt)

Avant!

<div align="right">

Chapter 9

Specifying Simulation Options

</div>

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

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 ...>
```

opt1 ...   Specifies any of the input control options. Many options are in the form *<opt>* = *x*, where *<opt>* is the option name and "x" is the value assigned to that option. All options are described in this section.

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

GENERAL CONTROL OPTIONS		MODEL ANALYSIS	DC OPERATING POINT, DC SWEEP, and POLE/ZERO		TRANSIENT and AC SMALL SIGNAL ANALYSIS	
Input, Output	*Interfaces*	*General*	*Accuracy*	*Convergence*	*Accuracy*	*Timestep*
ACCT	ARTIST	DCAP	ABSH	CONVERGE	ABSH	ABSVAR
ACOUT	CDS	SCALE	ABSI	CSHDC	ABSV, VNTOL	DELMAX
ALT999 ALT9999	CSDF	TNOM	ABSMOS	DCFOR	ACCURATE	DVDT
	MEASOUT		ABSTOL	DCHOLD	ACOUT	FS
BRIEF	MENTOR	*MOSFETs*	ABSVDC	DCON	CHGTOL	FT
CO	POST	CVTOL	DI	DCSTEP	CSHUNT, GSHUNT	IMIN, ITL3
INGOLD	PROBE	DEFAD	KCLTEST	DCTRAN		IMAX, ITL4
LENNAM	PSF	DEFAS	MAXAMP	DV	DI	ITL5
LIST	SDA	DEFL	RELH	GMAX	GMIN	RELVAR
MEASDGT	ZUKEN	DEFNRD	RELI	GMINDC	GSHUNT, CSHUNT	RMAX
NODE		DEFNRS	RELMOS	GRAMP		RMIN
NOELCK	*Analysis*	DEFPD	RELV	GSHUNT	MAXAMP	SLOPETOL
NOMOD	ASPEC	DEFPS	RELVDC	ICSWEEP	RELH	TIMERES
NOPAGE	LIMPTS	DEFW		NEWTOL	RELI	
NOTOP	PARHIER	SCALM	*Matrix*	OFF	RELQ	*Algorithm*
NUMDGT	SPICE	WL	ITL1	RESMIN	RELTOL	DVTR
NXX	SEED		ITL2		RISETIME	IMAX
OPTLST		*Inductors*	NOPIV	*Pole/Zero*	TRTOL	IMIN
OPTS	*Error*	GENK	PIVOT, SPARSE	CSCAL	VNTOL, ABSV	LVLTIM
PATHNUM	BADCHR	KLIM		FMAX	*Speed*	MAXORD
PLIM	DIAGNOSTIC		PIVREF	FSCAL	AUTOSTOP	METHOD

Table 9-1: .OPTION Keyword Application Table

GENERAL CONTROL OPTIONS		MODEL ANALYSIS	DC OPERATING POINT, DC SWEEP, and POLE/ZERO		TRANSIENT and AC SMALL SIGNAL ANALYSIS	
POST_VERSION	NOWARN	*BJTs*	PIVREL	GSCAL	BKPSIZ	MU, XMU
SEARCH	WARNLIMIT	EXPLI	PIVTOL	LSCAL	BYPASS	
VERIFY			SPARSE, PIVOT	PZABS	BYTOL	*Input, Output*
CPU	*Version*	*Diodes*		PZTOL	FAST	INTERP
CPTIME	H9007	EXPLI		RITOL	ITLPZ	ITRPRT
EPSMIN			*Input, Output*	XnR, XnI	MBYPASS	UNWRAP
EXPMAX			CAPTAB	NEWTOL		
LIMTIM			DCCAP			
			VFLOOR			

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-61N = 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:

+	Diode anode
-	Diode cathode
B	BJT base
B	MOSFET or JFET bulk
C	BJT collector
D	MOSFET or JFET drain
E	BJT emitter

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).
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 = IE-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.
SCALM	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 $1e^{-4}$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 $1e^{-4}$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 1e15amp/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 node voltage error tolerance for DC and transient analysis. 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:

$$\left| \Sigma i_b \right| < RELI \cdot \Sigma \left| i_b \right| + ABSI$$

where the i_bs 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   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. 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:

$$DV = max\left(0.1, \frac{V_{max}}{50}\right), \ if \ DV = 1000$$

$$GRAMP = max\left(6, \log_{10}\left(\frac{I_{max}}{GMINDC}\right)\right)$$

$$ITL1 = ITL1 + 20 \cdot GRAMP$$

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.

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.

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 $k\Omega$).

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:

$$GSCA = CSCAL \cdot FSCAL$$

$$GSCAL = \frac{1}{LSCAL} \cdot FSCAL$$

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 $(\lvert X_{real}\rvert + \lvert X_{imag}\rvert < PZABS)$,

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 $\lvert X_{imag}\rvert \le RITOL \cdot \lvert X_{real}\rvert$, then $X_{imag} = 0$

If $\lvert X_{real}\rvert \le RITOL \cdot \lvert X_{imag}\rvert$, 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), *(X2R,X2I)*	X0R = -1.23456e6 X0I = 0.0 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:

$$MIN\left[20,\ 1 + \left(\frac{TDeff}{RISETIME}\right) \cdot 20\right]$$

where *TDeff* 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.

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 MBYPASS**x**VNTOL.
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:

BYTOL = MBYPASS**x**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 11-24. This means that the initial DELMAX value shown in the Star-Hspice output listing is generally not the value used for simulation.

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

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.

$$Delta \ = \ FS \times [MIN(TSTEP, DELMAX, BKPT)]$$

where DELMAX is specified and BKPT is related to the breakpoint of the source. TSTEP is set in the .TRAN statement. 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.

IMIN = x, *ITL3 = x*	Determines the timestep in the algorithms used for transient analysis siimulations. 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: DELMAX = TSTEPxRMAX The default value equals 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 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.

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.

Algorithm

DVTR   Allows the use of voltage limiting in transient analysis. The default value equals 1000.

IMAX = x,   Determines the maximum timestep in the timestep
ITL4 = x   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.
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.

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.

Avant!

<div align="right">

Chapter 10

</div>

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**

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.

Figure 10-2: Equivalent Voltage Source: NODESET and .IC

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

- ■ DEBUG
  Usually only invoked by the program in the event of a nonconvergent simulation. Debug prints back the nonconvergent nodes with the new voltage, old voltage, and the tolerance (degree of nonconvergence). It also prints back the nonconvergent elements with their tolerance values.

- ■ NONE
  Inhibits node and element printouts but allows additional analysis specified to be performed

- ■ VOLTAGE
  Voltage table only

Note: *The preceding keywords are mutually exclusive; use only one at a time.*

---

time   Parameter that is placed directly following ALL, VOLTAGE, CURRENT, or DEBUG and specifies the time at which the report is printed.

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.

```
.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.

```
.OP
```

Output

```
***** OPERATING POINT INFORMATION     TNOM = 25.000
TEMP = 25.000
***** OPERATING POINT STATUS IS ALL     SIMULATION TIME IS 0.
  NODE      VOLTAGE       NODE     VOLTAGE      NODE     VOLTAGE
+ 0:2     =   0.          0:3   = 437.3258M     0:4  = 455.1343M
+ 0:5     = 478.6763M     0:6   = 496.4858M     0:7  = 537.8452M
+ 0:8     = 555.6659M     0:10  =   5.0000      0:11 = 234.3306M
 **** VOLTAGE SOURCES
SUBCKT
ELEMENT     0:VNCE         0:VN7          0:VPCE          0:VP7
  VOLTS      0.            5.00000        0.             -5.00000
   AMPS    -2.07407U   -405.41294P      2.07407U    405.41294P
  POWER      0.            2.02706N       0.             2.02706N
   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.             0.             0.             0.
   POWER    5.39908N     875.09107N      2.27712U       4.78896U
   BETAD-987.65432M   -987.65432M    -987.65432M   - 987.65432M
   GM       0.             0.             0.             0.
   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.             0.             0.             0.
   CCS    800.00000P    800.00000P     800.00000P    800.00000P
   BETAAC   0.             0.             0.             0.
   FT       0.             0.             0.             0.
```

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.

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*.

level_keyword	Circuit level at which the operating point is saved. The level can be one of the following. Default = ALL.

- ■ ALL
  All nodes from the top to the lowest circuit level are saved. This option provides the greatest improvement in simulation time.
- ■ TOP
  Only nodes in the top-level design are saved. No subcircuit nodes are saved.
- ■ NONE
  The operating point is not saved.

save_time	Time during transient analysis at which the operating point is saved. A valid transient analysis statement is required to successfully save a DC operating point. Default = 0.

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.

.DC Statement—DC Sweeps

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:

DATA = datanm	Datanm is the reference name of a .DATA statement.
incr1 …	Voltage, current, element, model parameters, or temperature increment values
MODEL	Specifies the optimization reference name used in the .MODEL OPT statement used in an optimization analysis
MONTE = val	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.*
np	Number of points per decade or per octave, or just number of points depending on the preceding keyword.
OPTIMIZE	Specifies the parameter reference name used for optimization in the .PARAM statement
RESULTS	Specifies the measure name used for optimization in the .MEASURE statement

start1 …	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.
stop1 …	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
var1 …	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 can not 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 pwl(0,0 2.5u,12 5u,0)
cb1 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.

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	GRAMP	NOPIV
CAPTAB	GSHUNT	OFF
CSHDC	ICSWEEP	PIVOT
DCCAP	ITL1	PIVREF
DCFOR	ITL2	PIVREL
DCHOLD	KCLTEST	PIVTOL
DCSTEP	MAXAMP	RESMIN
DV	NEWTOL	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.

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:

$$\left| \Sigma i_b \right| < RELI \cdot \Sigma \left| i_b \right| + ABSI$$

where the i_bs 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 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.

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 $(\lvert X_{real} \rvert + \lvert X_{imag} \rvert < PZABS)$,

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 $\lvert X_{imag} \rvert \leq RITOL \cdot \lvert X_{real} \rvert$, then $X_{imag} \ = \ 0$

If $\lvert X_{real} \rvert \leq RITOL \cdot \lvert X_{imag} \rvert$, then $X_{real} \ = \ 0$

Default = 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.

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.

$$| Vn^k - Vn^{k-1} | <\ = accuracy\ tolerance$$

where

Vn^k is the solution at timepoint n and iteration k

Vn^{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 10-1: Absolute and Relative Accuracy Tolerances

Type	Option	Default
Nodal Voltage Tolerances	ABSVDC	50 μv
	RELVDC	.001

Table 10-1: Absolute and Relative Accuracy Tolerances

Type	Option	Default
Current Element Tolerances	ABSI	1 nA
	RELI	.01
	ABSMOS	1 uA
	RELMOS	.05

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 10-2: RELV vs. Accuracy and Simulation Time for 2 Bit Adder

RELVDC	Iteration	Delay (ns)	Period (ns)	Fall time (ns)
.001	540	31.746	14.336	1.2797
.005	434	31.202	14.366	1.2743
.01	426	31.202	14.366	1.2724
.02	413	31.202	14.365	1.3433

Table 10-2: RELV vs. Accuracy and Simulation Time for 2 Bit Adder

RELVDC	Iteration	Delay (ns)	Period (ns)	Fall time (ns)
.05	386	31.203	14.365	1.3315
.1	365	31.203	14.363	1.3805
.2	354	31.203	14.363	1.3908
.3	354	31.203	14.363	1.3909
.4	341	31.202	14.363	1.3916
.4	344	31.202	14.362	1.3904

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 11-23.

Convergence Control Option Descriptions

The options listed below are described in this section.

ABSH	DCON	RELH
ABSI	DCTRAN	RELI
ABSMOS	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.

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 µV).

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.

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.

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

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, ABSMOS. 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.

2. Setting .OPTIONS CONVERGE = -1 disables step 4.

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

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

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:

conductance = capacitance/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.

Figure 10-5: Conductance Insertion

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

subckt element model	xinv21 21:mphc1 0:p1	xinv22 22:mphc1 0:p1	xinv23 23:mphc1 0:p1	xinv23 23:mnch1 0:n1	xinv24 24: mphc1 0:p1
id	27.5809f	140.5646u	1.8123p	1.7017m	5.5132u
ibs	205.9804f	3.1881f	31.2989f	0.	200.0000f
ibd	0.	0.	0.	-168.7011f	0.
vgs	4.9994	-4.9992	69.9223	4.9998	-67.8955
vds	4.9994	206.6633u	69.9225	-64.9225	2.0269
vbs	4.9994	206.6633u	69.9225	0.	2.0269
vth	-653.8030m	-745.5860m	-732.8632m	549.4114m	-656.5097m
tolds	114.8609	82.5624	155.9508	104.5004	5.3653
tolbd	0.	0.	0.	0.	0.
tolbs	3.534e-19	107.1528m	0.	0.	0.

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
.Xff data_in[1] out[1] out[1]/ strobe LATCH Qset = vdd
```

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, BJTs)

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.

Avant!

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

Figure 11-1 illustrates the transient analysis simulation flow for Star-Hspice.

Figure 11-1: Transient Analysis Simulation Flow

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>
```

Optimization:

```
.TRAN DATA = datanm OPTIMIZE = opt_par_fun
+ RESULTS = measnames MODEL = optmod
```

Transient sweep specifications can include the following keywords and parameters:

DATA = datanm	Data name referred to in the .TRAN statement
MONTE = val	Produces a number *val* of randomly generated values that are used to select parameters from a distribution. The distribution can be *Gaussian*, *Uniform*, or *Random Limit.*
np	Number of points or number of points per decade or octave, depending on the preceding keyword
param_expr...	User-specified expressions—for example, param_expr1...param_exprN
pincr	Voltage, current, element or model parameter, or temperature increment value ***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

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

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.

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
   ```

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
   ```

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:

Method	**Tolerance**		**Limit**	
BYPASS	ABSH	RELH	AUTOSTOP	ITL3
CSHUNT	ABSV	RELI	BKPSIZ	ITL4
DVDT	ABSVAR	RELQ	DELMAX	ITL5
GSHUNT	ACCURATE	RELTOL	DVTR	RMAX
INTERP	BYTOL	RELV	FS	RMIN
ITRPRT	CHGTOL	RELVAR	FT	VFLOOR
LVLTIM	DI	SLOPETOL	GMIN	
MAXORD	FAST	TIMERES		
METHOD	MBYPASS	TRTOL		
	MAXAMP	VNTOL		
	MU	XMU		

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.*

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. Default = 0.
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.

	Note: *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.
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.

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

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.
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. Default = 50 (microvolts).
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. Default = 0.5 (volts).

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: BYTOL = MBYPASS x 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.

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. Default = 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 percent change in current from the value calculated at the previous timepoint. Default = 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. Default = 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. Default = 1e-3.
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. Default = 0.30 (30%).

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. Default = 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. Default = 1 ps.
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. Default = 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. Default = 50 (microvolts).
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.

Limit Options

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. Default = 5000.
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 11-24. This means that the initial DELMAX value shown in the Star-Hspice output listing is generally not the value used for simulation.

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 siimulations. 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.
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.

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 = TSTEPxRMAX 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 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. 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 11-9.

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:

    DVDT = 4   LVLTIM = 1   RMAX = 5   SLOPETOL = 0.75
    FT = FS = 0.25BYPASS = 1 BYTOL = MBYPASSxVNTOL = 0.100m

> *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 11-30.

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.

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

Figure 11-3: Iteration Algorithm

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 11-32.

- 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:

IMAX	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.
IMIN	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.

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:

min[(TSTOP/50), DELMAX, (TSTEP**x**RMAX)]

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

.FOUR Statement

.FFT Statement

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:

$$THD = \frac{1}{R1} \cdot \left(\sum_{m=2}^{9} R_m^2 \right)^{1/2} \cdot 100\%$$

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

$$C_m = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} g(t) \cdot \cos(m \cdot t) \cdot dt$$

$$D_m = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} g(t) \cdot \sin(m \cdot t) \cdot dt$$

$$g(t) = \sum_{m=0}^{9} C_m \cdot \cos(m \cdot t) + \sum_{m=0}^{9} D_m \cdot \sin(m \cdot t)$$

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
******
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   125.2127m    41.1043m   157.7399    -18.7987
  8       160.0000x    25.6916m     8.4339m   172.9579     -3.5807
  9       180.0000x    47.7347m    15.6701m   154.1858    -22.3528

    total harmonic distortion =    27.3791   percent
```

For further information on Fourier analysis, see "Performing Fourier Analysis" on page 11-35.

.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>
```

Table 11-1: .FFT Statement Parameters

Parameter	Default	Description
output_var	—	Can be any valid output variable, such as voltage, current, or power
START	see Description	Specifies the beginning of the output variable waveform to be analyzed. Defaults to the START value in the .TRAN statement, which defaults to 0.
FROM	see START	In .FFT statements, FROM is an alias for START.

Table 11-1: .FFT Statement Parameters

Parameter	Default	Description
STOP	see Description	Specifies the end of the output variable waveform to be analyzed. Defaults to the TSTOP value in the .TRAN statement.
TO	see STOP	In .FFT statements, TO is an alias for STOP.
NP	1024	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.
FORMAT	NORM	Specifies the output format: NORM = normalized magnitude UNORM = unnormalized magnitude
WINDOW	RECT	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
ALFA	3.0	Specifies the parameter used in GAUSS and KAISER windows to control the highest side-lobe level, bandwidth, and so on. $1.0 <= ALFA <= 20.0$
FREQ	0.0 (Hz)	Specifies a frequency of interest. If FREQ is nonzero, the output listing is limited to the harmonics of this frequency, based on FMIN and FMAX. The THD for these harmonics also is printed.

Table 11-1: .FFT Statement Parameters

Parameter	Default	Description
FMIN	1.0/T (Hz)	Specifies the minimum frequency for which FFT output is printed in the listing file, or which is used in THD calculations. T = (STOP-START)
FMAX	0.5*NP*FMIN (Hz)	Specifies the maximum frequency for which FFT output is printed in the listing file, or which is used in THD calculations.

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, a *.ft0* and a *.ft1* 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

 frequency     frequency      fft_mag       fft_mag      fft_phase
   index         (hz)          (db)                        (deg)
     2          1.0000k        0.            1.0000       -3.8093m
     4          2.0000k      -125.5914     525.3264n      -5.2406
     6          3.0000k      -106.3740       4.8007u     -98.5448
     8          4.0000k      -113.5753       2.0952u      -5.5966
    10          5.0000k      -112.6689       2.3257u    -103.4041
    12          6.0000k      -118.3365       1.2111u     167.2651
```

```
14          7.0000k   -109.8888        3.2030u   -100.7151
16          8.0000k   -117.4413        1.3426u    161.1255
18          9.0000k    -97.5293       13.2903u     70.0515
20         10.0000k   -114.3693        1.9122u    -12.5492
total harmonic distortion =      1.5065m  percent
```

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

Column Heading	Description
frequency index	Runs from 1 to NP/2, or the corresponding index for FMIN and FMAX. The DC component corresponding to index 0 is displayed independently.
frequency	Actual frequency associated with the index
fft_mag (db), fft_mag	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.
fft_phase	Associated phase, in degrees

Notes:

1. The following formula should be used as a guideline when specifying a
   frequency range for FFT output:

   frequency increment = 1.0/(STOP - 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.

Avant!

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:

```
    Y_C = jωC        for capacitors
```

and

```
    Y_L = 1/jωL      for inductors
```

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:

DATA = datanm	Data name referred to in the .AC statement
incr	voltage, current, element or model parameter increment value ***Note:*** *If "type" variation is used, the "np" (number of points) is specified instead of "incr".*
fstart	Starting frequency ***Note:*** *If type variation "POI" (list of points) is used, a list of frequency values is specified instead of "fstart fstop".*
fstop	Final frequency
MONTE = val	Produces a number *val* 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.
np	Number of points per decade or per octave, or just number of points, depending on the preceding keyword
start	Starting voltage, current, any element or model parameter value
stop	Final voltage, current, any element or model parameter value

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.

```
.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 *datanm*.

```
.AC DEC  10  1  10K  SWEEP  DATA = datanm
```

The following example illustrates a frequency sweep along with a Monte Carlo analysis with 30 trials.

```
.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

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.
ACOUT	AC output calculation method for the difference in values of magnitude, phase and decibels for prints and plots. Default = 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.

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.

DIM2	Intermodulation distortion, difference. The relative magnitude and phase of the frequency component (F1 - F2).
DIM3	Intermodulation distortion, second difference. The relative magnitude and phase of the frequency component $(2 \cdot F1 - F2)$.
HD2	Second-order harmonic distortion. The relative magnitude and phase of the frequency component $2 \cdot F1$ (ignoring F2).
HD3	Third-order harmonic distortion. The relative magnitude and phase of the frequency component $3 \cdot F1$ (ignoring F2).
SIM2	Intermodulation distortion, sum. The relative magnitude and phase of the frequency component (F1 + F2).

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:

Rload	The resistor element name of the output load resistor into which the output power is fed
inter	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.
skw2	Ratio of the second frequency F2 to the nominal analysis frequency F1. The acceptable range is $1e\text{-}3 < skw2 \leq 0.999$. If skw2 is omitted, a value of 0.9 is assumed.

| *refpwr* | Reference power level used in computing the distortion products. If omitted, a value of 1mW, measured in decibels magnitude (dbM), is assumed. The value must be ≥ 1e-10. |
| *spwf* | Amplitude of the second frequency F2. The value must be ≥ 1e-3. Default = 1.0. |

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:

ovv	Nodal voltage output variable defining the node at which the noise is summed
srcnam	Name of the independent voltage or current source to be used as the noise input reference
inter	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.

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:

$$onoise = \sum_{n=1}^{n} |Z_n \cdot I_n|^2$$

where:

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 KF and AF 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.

MAXFLD	Maximum allowed number of folding intervals. The highest frequency (in Hertz) considered by the algorithm is given by:

$$\text{FMAX} = \text{MAXFLD} \cdot \text{FS}$$

Default = 10.0.

BETA	Integrator duty cycle; specifies an optional noise integrator at the sampling node

$\text{BETA} = 0$ no integrator

$\text{BETA} = 1$ simple integrator (default)

If the integrator is clocked (that is, it only integrates during a fraction of the sampling interval 1/FS), then BETA should be set to the duty cycle of the integrator.

.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>
```

where:

input	AC input voltage or current source name
output	Output port. It can be an output voltage, V(n1,n2), or an output current, I(source), or I(element).
RIN	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	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.

Example

One-port network:

```
.NET     VINAC       RIN = 50
.NET     IIN         RIN = 50
```

Two-port network:

```
.NET  V(10,30)  VINAC   ROUT = 75   RIN = 50
.NET  I(RX)     VINAC   ROUT = 75   RIN = 50
```

AC Network Analysis - Output Specification

Syntax
```
  Xij(z), ZIN(z), ZOUT(z), YIN(z), YOUT(z)
```

where:

X	Specifies Z for impedance, Y for admittance, H for hybrid, and S for scattering
ij	i and j can be 1 or 2. They identify which matrix parameter is to be printed.

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

```
.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:[1] Star-Hspice Network Analysis Results

```
*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
```

```
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

Figure 12-3: ZIN 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)　.NET Vout Isrc　　　　V　　　=　　[Z]　　[I]

2)　.NET Iout Vsrc　　　　I　　　=　　[Y]　　[V]

3)　.NET Iout Isrc　　　　$[V1\ I2]^T$ =　　[H]　　$[I1\ V2]^T$

4)　.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 I2 must be zero is not satisfied in the circuit above. Instead, V2 is zero, which results in incorrect values for Z11 and Z21.

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:

$$V1 \; = \; H11 \cdot I1 + H12 \cdot V2$$

$$I2 \; = \; H21 \cdot I1 + H22 \cdot V2$$

For Hybrid parameter calculations of H11 and H21, V2 is set to zero (due to the DC voltage source V_{CE}), while for H12 and H22 calculations, I1 is set to zero (due to the DC current source I_B). Setting I1 and V2 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
.option  nopage list
+ newtol reli = 1e-5 absi = 1e-10 relv = 1e-5 relvdc = 1e-7
+ nomod post gmindc = 1e-12
.op
.param vbe = 0 ib = 0 ic = 0 vce = 0

$ H-parameter
.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

.model bjt   npn   subs = 1
```

```
+ bf = 1.292755e+02 br = 8.379600e+00
+ is = 8.753000e-18 nf = 9.710631e-01 nr = 9.643484e-01
+ ise = 3.428000e-16 isc = 1.855000e-17 iss = 0.000000e+00
+ ne = 2.000000e+00 nc = 9.460594e-01 ns = 1.000000e+00
+ vaf = 4.942130e+01 var = 4.589800e+00
+ ikf = 5.763400e-03 ikr = 5.000000e-03 irb = 8.002451e-07
+ rc = 1.216835e+02 rb = 1.786930e+04 rbm = 8.123460e+01
+ re = 2.136400e+00
+ cje = 9.894950e-14 mje = 4.567345e-01 vje = 1.090217e+00
+ cjc = 5.248670e-14 mjc = 1.318637e-01 vjc = 5.184017e-01
+ xcjc = 6.720303e-01
+ cjs = 9.671580e-14 mjs = 2.395731e-01 vjs = 5.000000e-01
+ tf = 3.319200e-11 itf = 1.457110e-02 xtf = 2.778660e+01
+ vtf = 1.157900e+00 ptf = 6.000000e-05
+ xti = 4.460500e+00 xtb = 1.456600e+00 eg = 1.153300e+00
+ tikf1 = -5.397800e-03 tirb1 = -1.071400e-03
+ tre1 = -1.121900e-02 trb1 = 3.039900e-03
+ trc1 = -4.020700e-03 trm1 = 0.000000e+00

.print ac par('ib') par('ic')
+ h11(m) h12(m) h21(m) h22(m)
+ z11(m) z12(m) z21(m) z22(m)
+ s11(m) s21(m) s12(m) s22(m)
+ y11(m) y21(m) y12(m) y22(m)

.ac Dec 10  1e6  5g  sweep  data = bias


.data bias
     vbe          vce            ib             ic
 771.5648m    292.5047m      1.2330u      126.9400u
 797.2571m    323.9037m      2.6525u      265.0100u
 821.3907m    848.7848m      5.0275u      486.9900u
 843.5569m     1.6596        8.4783u      789.9700u
 864.2217m     2.4031       13.0750u        1.1616m
 884.3707m     2.0850       19.0950u        1.5675m
.enddata
.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'
q1   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'
q1   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'
q1   c b e 0  bjt
```

References

1. Goyal, Ravender. "S-Parameter Output From SPICE Program", *MSN & CT,* February 1988, pp. 63 and 66.

Avant!

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:

$$Mean \quad = \frac{x_1 + x_2 + \dots + x_n}{N}$$

$$Variance \quad = \frac{(x_1 - Mean)^2 + \dots (x_n - Mean)^2}{N - 1}$$

$$Sigma \quad = \sqrt{Variance}$$

$$Average\ Deviation \quad = \frac{|x_1 - Mean| + \dots + |x_n - Mean|}{N - 1}$$

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

Ambient Temperature

System Temperature

Part Temperature

source   drain   source   drain
        gate   gate

Model Junction Temperature   Part Junction Temperature

Temperatures in Star-Hspice are calculated as differences from ambient temperature:

$$Tambient + \Delta system + \Delta part + \Delta junction\ =\ Tjunction$$

$$Ids\ =\ f(Tjunction, Tmodel)$$

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:

element temperature = circuit temperature + 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

Type	Param	Slow	Fast
NMOS	XL	+	-
	RSH	+	-
	DELVTO	+	-
	TOX	+	-
	XW	-	+
PMOS	XL	+	-
	RSH	+	-
	DELVTO	-	+
	TOX	+	-
	XW	-	+

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.

Figure 13-3: Historical Records for Skew Parameters in a MOS Process

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.

Figure 13-4: Worst Case Corners Library File for a CMOS Process Model

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
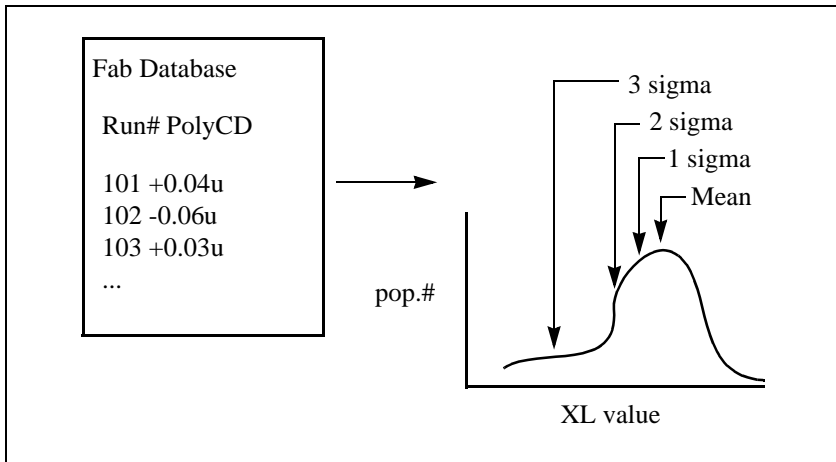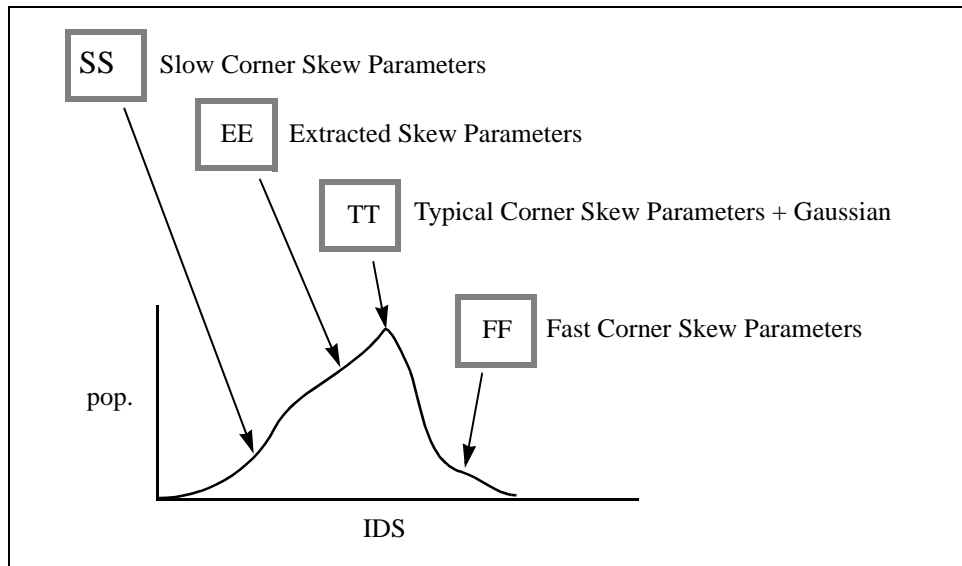
.PARAM
$ polysilicon Critical Dimensions
+ polycd=agauss(0,0.06u,1) xl='polycd-sigma*0.06u'
$ Active layer Critical Dimensions
+ nactcd=agauss(0,0.3u,1) xwn='nactcd+sigma*0.3u'
+ pactcd=agauss(0,0.3u,1) xwp='pactcd+sigma*0.3u'
```

```
$ 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

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 $val^{-1/2}$.

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 · 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

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
```

```
.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 au1 0 -1
rau1 au1 0 rau_1

Iau10 au10 0 -1
rau10 au10 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

Figure 13-8: Uniform Functions

Figure 13-9: Limit Functions

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.

Figure 13-10: Major and Minor Distribution of Manufacturing Variations

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
.DC 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
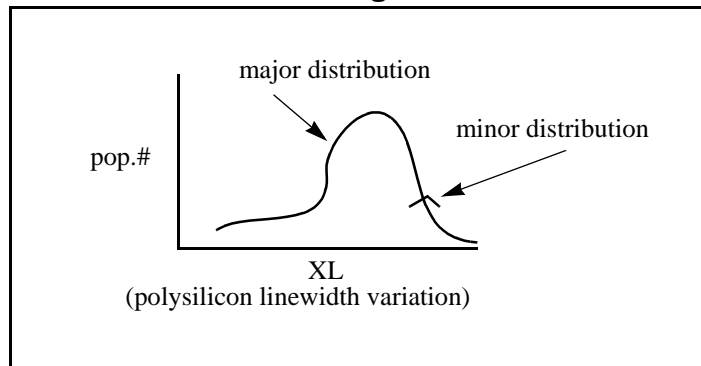
M1  1  2  GND GND  NCH  W=10U L=LEFF
M2  1  2  VDD VDD  PCH  W=20U L=LEFF

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: MON1.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.

Figure 13-12: Monte Carlo 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

Leff = Ldrawn - 2 · DEL

the model poly distribution is half the physical per-side values.

```
Cla 1 0 CMOD   W=ELPOLY   L=ELPOLY
Clb 1 0 CMOD   W=ELPOLY   L=ELPOLY
ClC 1 0 CMOD   W=ELPOLY   L=ELPOLY
ClD 1 0 CMOD   W=ELPOLY   L=ELPOLY

$ 10U POLYWIDTH,0.05U=1SIGMA
$ CAP MODEL USES 2*MODPOLY  .05u= 1 sigma
$ 5angstrom oxide thichness AT 1SIGMA
.PARAM ELPOLY=AGAUSS(10U,0.02U,1)
+      MODPOLY=AGAUSS(0,.05U,1)
+      POLYCAP=AGAUSS(200e-10,5e-10,1)

.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.

```
Cla 1 0 CMOD   SCALE=ELCAP
Clb 1 0 CMOD   SCALE=ELCAP
ClC 1 0 CMOD   SCALE=ELCAP
ClD 1 0 CMOD   SCALE=ELCAP
.PARAM  ELCAP=Gauss(1,.01,2) $ 1% at 2 sigma
+ MODCAP=Gauss(.25p,.1,2) $10% at 2 sigma
.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
```

```
.eom
```

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'
+ rshpcd=agauss(150,20,1) rshp='rshpcd-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 u1=.02 x3u1=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
+ 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 x2u1=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 x3u1=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
.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.

Figure 13-13: Sweep of Skew Parameters from -3 Sigma to +3 Sigma

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

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
```

```
vtopcd: delvtop = 4.1776E-02
rshncd: rshn = 45.16
rshpcd: rshp = 166.2
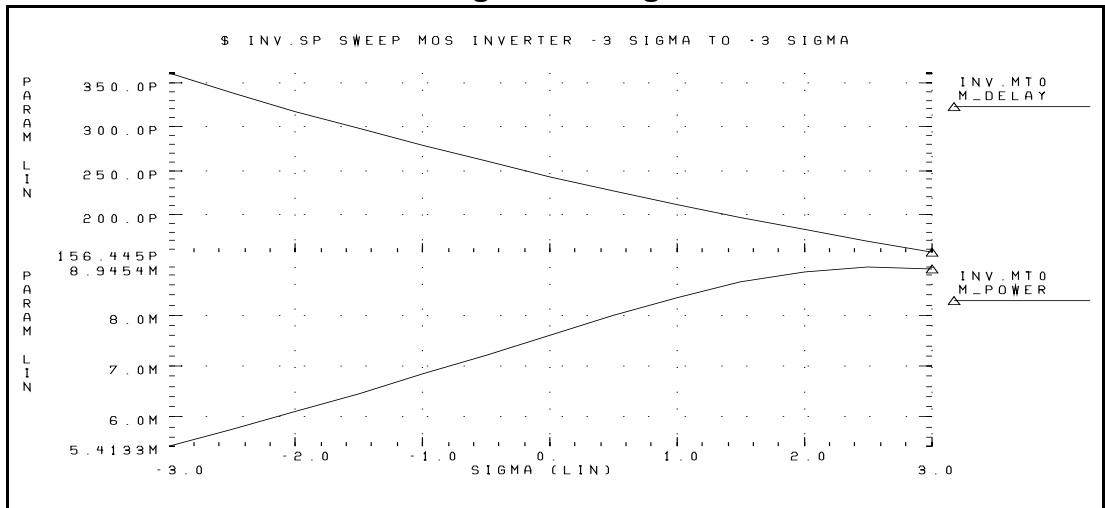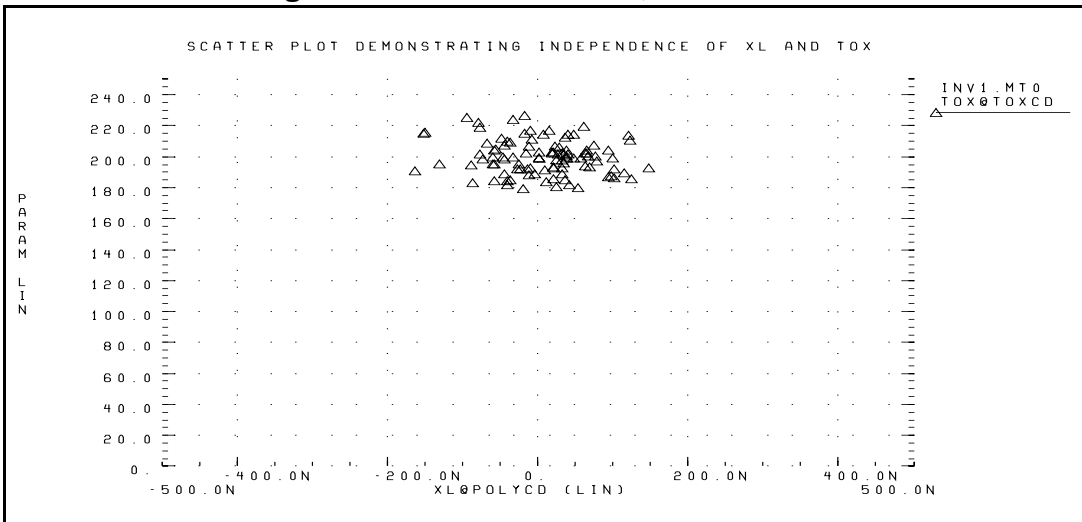
m_delay = 1.7946E-10 targ= 3.4746E-10 trig= 1.6799E-10
m_power = 7.7781E-03 from= 0.0000E+00 to= 1.7946E-10
```

From the preceding listing, you can see that the *m_delay* value of 1.79e-10 seconds is the fastest pair delay. In addition, the Monte Carlo parameters can be examined.

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).

Figure 13-17: Sensitivity of Delay with Poly CD (XL)

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.

Figure 13-18: Sensitivity of Delay with TOX

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 9-48 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.

- .MODEL modname OPT ...
- .PARAM parameter=OPTxxx (init, min, max)
- A .DC, .AC, or .TRAN analysis statement with MODEL=modname, OPTIMIZE=OPTxxx, and RESULTS=measurename
- .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:

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.
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.
delta	The final parameter value is the initial guess $\pm$ ($n \cdot$delta). If delta is not specified, the final parameter value can be anything between low_limit and upper_limit. This is useful for optimizing transistor drawn widths and lengths, which must be quantized.

Example

```
.PARAM vtx=OPT1(.7,.3,1.0) uox=OPT1(650,400,900)
```

In the above example, the parameters *uox* and *vtx* are the variable model parameters to optimize a model for a given set of electrical specifications. The parameter *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).

.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.

CUT	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.
DIFSIZ	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 Δx = delta. Default = 1e-3.
GRAD	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.
ITROPT	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.
LEVEL	Selects the optimizing algorithm to use. Currently, the only option is LEVEL=1, a modified Levenburg-Marquardt algorithm.
MAX	Sets the upper limit on CLOSE. Values greater than 100 are recommended. Default=6000.

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:

$$\Delta par\_val = \text{DIFSIZ} \cdot \text{MAX}(par\_val, \text{PARMIN})$$

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 then 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 (*ids* versus *vgs*) and drain curves (*ids* versus *vds*).
The Level 3 parameters VTO, GAMMA, UO, VMAX, THETA, and KAPPA 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 COMP1 (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=10e-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 be 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.

Figure 13-21: Level 3 MOSFET Optimization

MOS Level 13 Model DC Optimization

This example shows an optimization of I-V data to a Level 13 MOS model. The data consists of gate curves (*ids* versus *vgs*) and drain curves (*ids* versus *vds*). This example demonstrates two-stage optimization. The Level 13 parameters vfb0, k1, muz, x2m, and u00 are optimized to the gate data. Then the Level 13 parameters MUS, X3MS, U1, 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.

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 u1=u1
$...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)
+ u1      = opt2(0.1,0,1)
+ alpha   = opt2(1,1e-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
```
.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 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^{-rc}).
- .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

```
.title RCOPT.sp
.option post
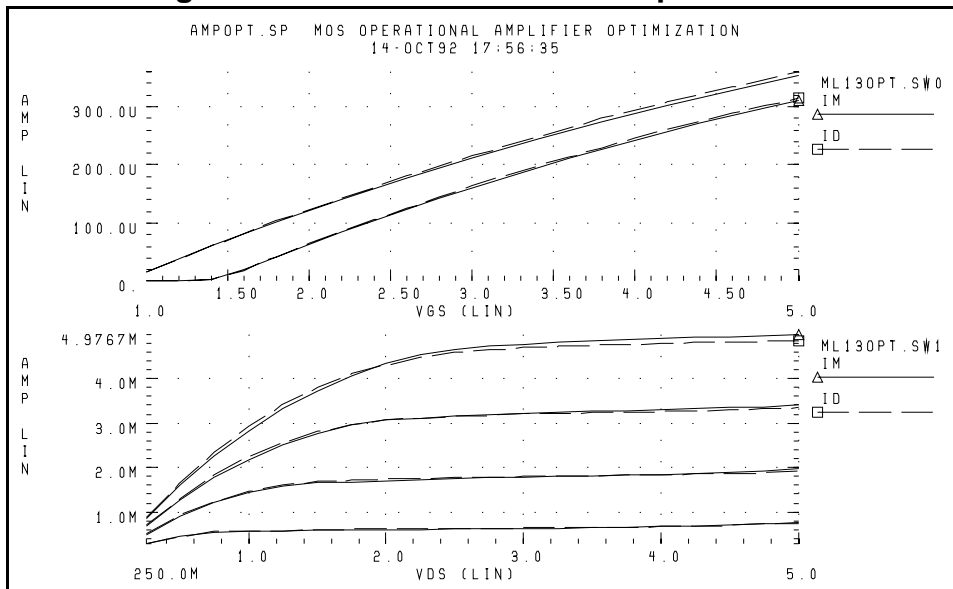
.PARAM RX=OPTRC(.5, 1E-2, 1E+2)
.PARAM CX=OPTRC(.5, 1E-2, 1E+2)

.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

```
*                                      %NORM-SEN     %CHANGE
.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**

Figure 13-24: Power Dissipation and Time Constant (WATT) 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

```
*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)
vinb 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
.mea tran tfopt  trig v(data) val=2.5 rise=1 targ v(out)
+ val=2.5 fall=1 goal 5.0n
.mea 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

```
*                                      %norm-sen    %change

.param wp2           =  84.4981u  $   22.5877  -989.3733u
.param wn2           =  34.1401u  $    7.6568  -659.2874u
.param wp3           = 161.7354u  $ 730.7865m -351.7833u
.param wn3           = 248.6829u  $    8.1362    -2.2416m
.param wp4           = 238.9825u  $    1.2798    -1.5774m
.param wn4           =  61.3509u  $ 315.4656m   43.5213m
.param wp5           =   1.7753m  $    4.1713    2.1652m
.param wn5           =   1.0238m  $    5.8506  413.9667u
.param wp10          = 268.3125u  $    8.1917    -2.0266m
.param wn10          = 115.6907u  $   40.5975  -422.8411u
.param wp11          = 153.1344u  $ 482.0655m  -30.6813m
*** optimize results measure names and values
* tfopt      =    5.2056n
* tropt      =    5.5513n
* rmspowo    = 200.1808m
```

Figure 13-25: Tristate Buffer Optimization Circuit

**Figure 13-26: Tristate Input/Output Optimization
ACIC2B.TR0 = Before Optimization, ACIC2B.TR1 = Optimized Result**

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
```

```
.MEASURE AC COMP3 ERR1 PAR(S12M) S12(M)
.MEASURE AC COMP4 ERR1 PAR(S12P) S12(P) MINVAL=10
.MEASURE AC COMP5 ERR1 PAR(S21M) S21(M)
.MEASURE AC COMP6 ERR1 PAR(S21P) S21(P) MINVAL=10
.MEASURE AC COMP7 ERR1 PAR(S22M) S22(M)
.AC DATA=MEASURED
.PRINT PAR(S11M) S11(M) PAR(S11P) S11(P)
.PRINT PAR(S12M) S12(M) PAR(S12P) S12(P)
.PRINT PAR(S21M) S21(M) PAR(S21P) S21(P)
.PRINT PAR(S22M) S22(M) PAR(S22P) S22(P)
.DATA MEASURED
FREQ    S11M    S11P    S21M    S21P    S12M    S12P S22M    S22P
100ME   .6      -52     19.75   148     .02     65   .87     - 21
200ME   .56     -95     15.30   127     .032    49   .69     - 33
500ME   .56     -149    7.69    97      .044    41   .45     - 41
1000ME .58      -174    4.07    77      .061    42   .39     - 47
2000ME .61      159     2.03    50      .095    40   .39     - 70
.ENDDATA
.PARAM FREQ=100ME S11M=0, S11P=0, S21M=0, S21P=0, S12M=0,
+  S12P=0, S22M=0, S22P=0
.END
```

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
```

```
.PARAM  RB   =  2.0582   $ 12.8257M  2.3084
.PARAM  RE   =869.8714M  $ 66.8123M  4.5597
.PARAM  RC   = 54.2262   $  3.1427M 20.7359
.PARAM  IS   = 99.9900P  $  3.6533X 34.4463M
```

Figure 13-27: BJT-S Parameter Optimization

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 INGOLD=2 NOPAGE VNTOL=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 MJE=0.5 MJC=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=RE
+ 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
```
*                                 %NORM-SEN %CHANGE

.PARAM BF = 1.4603E+02 $ 2.7540E+00 -7.3185E-07
.PARAM IS = 2.8814E-15 $ 3.7307E+00 -5.0101E-07
.PARAM NF = 9.9490E-01 $ 9.1532E+01 -1.0130E-08
.PARAM IKF = 8.4949E-02 $ 1.3782E-02 -8.8082E-08
.PARAM RE = 6.2358E-01 $ 8.6337E-02 -3.7665E-08
.PARAM ISE = 5.0569E-16 $ 1.0221E-01 -3.1041E-05
.PARAM NE = 1.3489E+00 $ 1.7806E+00 2.1942E-07
```

Optimization Results
```
RESIDUAL SUM OF SQUARES = 1.82776
```

Optimized Parameters OPT2
```
*                                 %NORM-SEN %CHANGE
```

```
.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

Figure 13-29: BJT Optimization Reverse 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
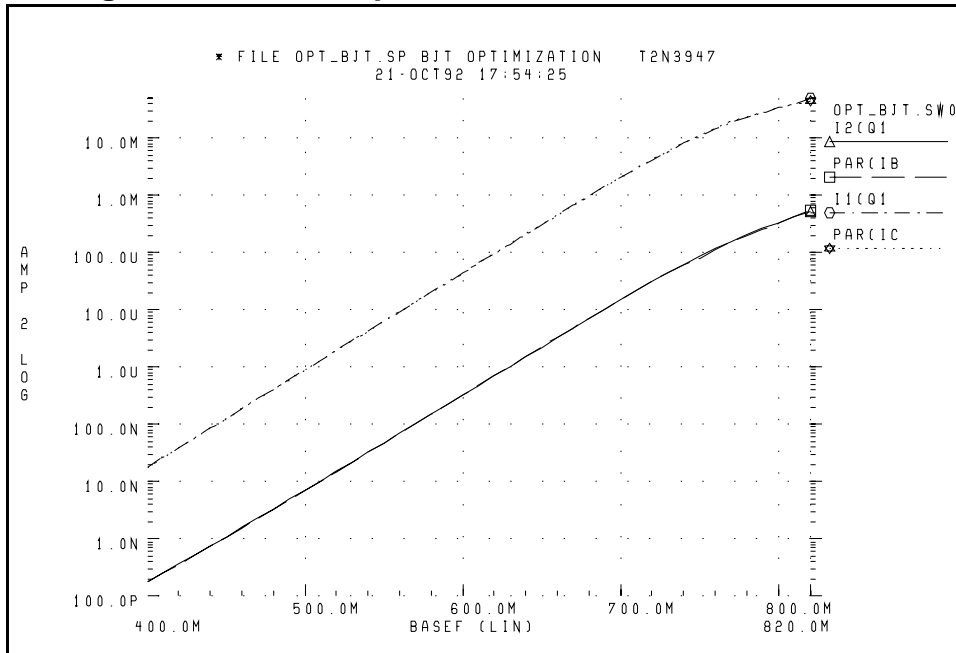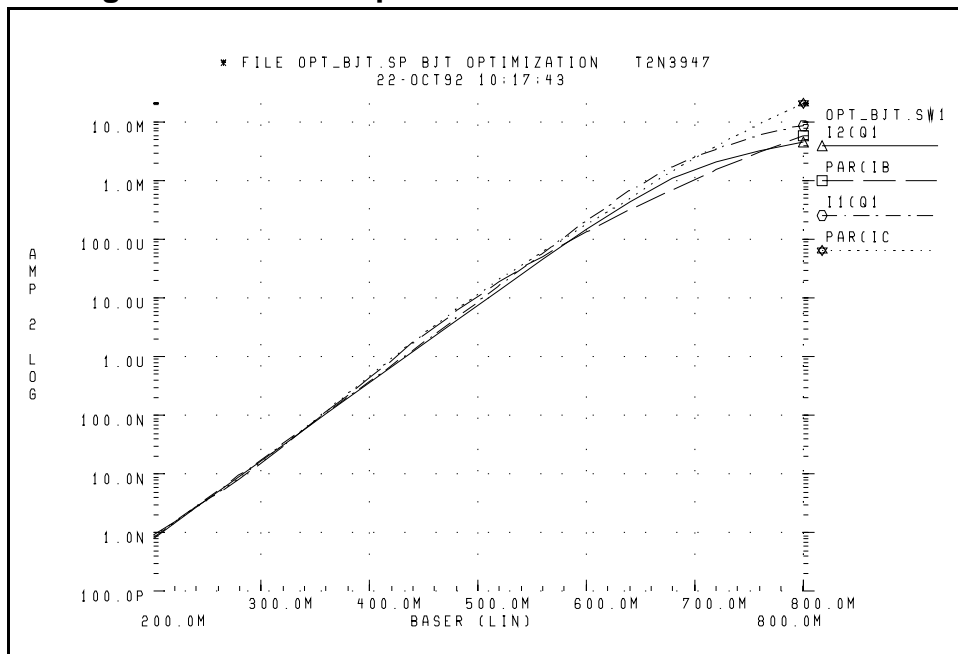
VG GATE 0 XVGS
VD DRAIN 0 XVDS
J1 DRAIN GATE 0 JFETN1

.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

```
*                                %NORM-SEN    %CHANGE
.PARAM VTO          = -1.1067    $   64.6110      43.9224M
.PARAM VGEXP        = 2.9475     $   13.2024     219.4709M
.PARAM GAMDS        = 0.         $    0.           0.
.PARAM BETA         = 11.8701M   $   17.2347     136.8216M
.PARAM LAMBDA       = 138.9821M  $    2.2766      -1.5754
.PARAM RDS          = 928.3216M  $  704.3204M    464.0863M
.PARAM ALPHA        = 2.2914     $  728.7492M    168.4004M
.PARAM UCRIT        = 1.0000M    $   18.2438M   -125.0856
.PARAM SATEXP       = 1.4211     $    1.2241       2.2218
```

Figure 13-30: JFET Optimization

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
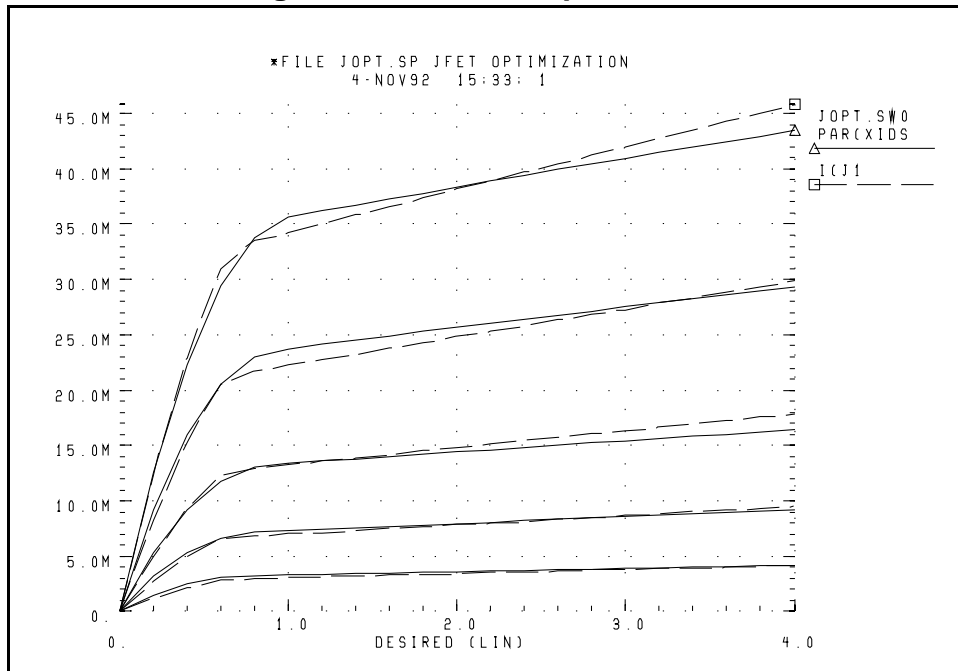
.OPTION RELV=1E-3 RELVAR=.01 NOMOD ACCT POST
.PARAM VDD=5    VREF='VDD/2'
VDD VSUPPLY 0 VDD
VIN+    VIN+     0 PWL(0 ,'VREF-10M'   10NS 'VREF+10M' )
VINBAR+ VINBAR+  0 PWL(0 ,'VREF+10M'   10NS 'VREF-10M' )
VIN- VIN-  0  VREF
VBIAS VBIAS 0 BIAS
.GLOBAL VSUPPLY VBIAS

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=OPT1
+ 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

Avant!

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.

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.

```
.PARAMETER REXX=1
R1 1 2 REXX
.MODEL REXX R RES=1
```

Wire Model Parameters

Name (Alias)	Units	Default	Description
BULK	gnd		Default reference node for capacitance
CAP	F	0	Default capacitance
CAPSW	F/m	0	Sidewall fringing capacitance
COX	F/m^2	0	Bottomwall capacitance
DI		0	Relative dielectric constant
DLR	m	0	Difference between drawn length and actual length (for resistance calculation only). For capacitance calculation, DW is used DLReff=DLR · SCALM
DW	m	0	Difference between drawn width and actual width DWeff=DW · SCALM
L	m	0	Default length of wire Lscaled=L · SHRINK · SCALM
LEVEL			Model selector (not used)
RAC	ohm		Default AC resistance (RACeff default is Reff)
RES	ohm	0	Default resistance
RSH		0	Sheet resistance/square
SHRINK		1	Shrink factor
TC1C	1/deg	0	First-order temperature coefficient for capacitance
TC2C	$1/deg^2$	0	Second-order temperature coefficient for capacitance
TC1R	1/deg	0	First-order temperature coefficient for resistance
TC2R	$1/deg^2$	0	Second-order temperature coefficient for resistance

Name (Alias)	Units	Default	Description
THICK	m	0	Dielectric thickness
TREF	deg C	TNOM	Temperature reference for model parameters
W	m	0	Default width of wire Wscaled=W · SHRINK · SCALM

Resistor Model 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:

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

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

If element resistance is specified:

$$Reff = \frac{R \cdot SCALE(element)}{M}$$

Otherwise, if $(Weff \cdot Leff \cdot RSH)$ is greater than zero, then:

$$Reff = \frac{Leff \cdot RSH \cdot SCALE(element)}{M \cdot Weff}$$

If $(Weff \cdot Leff \cdot RSH)$ is zero, then:

$$Reff = \frac{RES \cdot SCALE(element)}{M}$$

If AC resistance is specified in the element, then:

$$RACeff = \frac{AC \cdot SCALE(element)}{M}$$

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

$$RACeff = \frac{RAC \cdot SCALE(element)}{M}$$

If neither are specified, it defaults to:

$$RACeff = Reff$$

If the resistance is less than option RESMIN, it is reset it to RESMIN and a warning message is displayed.

$$RESMIN = \frac{1}{GMAX \cdot 1000 \cdot M}$$

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.

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

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

If the element capacitance C is specified:

$$CAPeff = C \cdot SCALE(element) \cdot M$$

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

$$CAPeff = M \cdot SCALE(element)$$
$$\cdot [Leff \cdot Weff \cdot COX + 2 \cdot (Leff + Weff) \cdot CAPSW]$$

The computation of the bottom wall capacitance COX is based upon a hierarchy of defaults and specified values involving the dielectric thickness THICK, the relative dielectric constant DI, and two absolute dielectric constants εo and εox, as follows:

1. If COX=value is given, the value is used.

2. If COX is not given specifically but THICK (the dielectric thickness) is given and nonzero:

   a. If DI=value is given and nonzero then:
   $$COX = \frac{DI \cdot \varepsilon o}{THICK}$$
   b. If DI is not given, or is zero, then:
   $$COX = \frac{\varepsilon ox}{THICK}$$
   where

   $$\varepsilon o = 8.8542149e\text{-}12 \, \text{F/meter}$$

   $$\varepsilon ox = 3.453148e\text{-}11 \, \text{F/meter}$$

3. If COX is not given and THICK= 0 is an error.

   If only the model capacitance CAP is specified, then:

   $$CAPeff = CAP \cdot SCALE(element) \cdot M$$

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:

$$inr = \left(NOISE \cdot \frac{4kT}{Rval} \right)^{1/2}$$

where *NOISE* is a model parameter that defaults to 1. To eliminate the contribution of resistor noise, use the NOISE parameter. To specify the NOISE parameter, use a model for the resistor.

Noise Summary Print Out Definitions

RX	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.
TOT, V^2/Hz	Total output noise: $OT = RX^2 \cdot inr^2$

Resistor Temperature Equations

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

$$R(t) = R \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

$$RAC(t) = RAC \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

$$C(t) = C \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

Δt	t - tnom
t	Element temperature in °K: t = circuit temp + DTEMP + 273.15
tnom	Nominal temperature in °K: tnom = 273.15 + TNOM

Capacitor Device Model and Equations

Capacitance Model

Syntax

```
.MODEL mname C parameter=value
```

mname	Model name
C	Specifies a capacitance model
parameter	Any model parameter name

Capacitance Parameters

Name (Alias)	Units	Default	Description
CAP	F	0	Default capacitance value
CAPSW	F/m	0	Sidewall fringing capacitance
COX	F/m^2	0	Bottomwall capacitance
DEL	m	0	Difference between drawn width and actual width or length DELeff = DEL · SCALM
DI		0	Relative dielectric constant
L	m	0	Default length of capacitor Lscaled = L · SHRINK · SCALM
SHRINK		1	Shrink factor
TC1	1/deg	0	First temperature coefficient for capacitance
TC2	$1/deg^2$	0	Second temperature coefficient for capacitance
THICK	m	0	Insulator thickness

Name (Alias)	Units	Default	Description
TREF	deg C	TNOM	Reference temperature
W	m	0	Default width of capacitor Wscaled = W · SHRINK · SCALM

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:

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

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

If the element capacitance C is specified:

$$CAPeff = C \cdot SCALE(element) \cdot M$$

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

$$CAPeff = M \cdot SCALE(element) \cdot$$
$$[Leff \cdot Weff \cdot COX + 2 \cdot (Leff + Weff) \cdot CAPSW]$$

If COX is not specified, but THICK is not zero, then:

$$COX = \frac{DI \cdot \varepsilon o}{THICK} \qquad \text{if DI not zero}$$

or

$$COX = \frac{\varepsilon ox}{THICK} \qquad \text{if DI=0}$$

where

$$\varepsilon o = 8.8542149e\text{-}12 \quad \frac{F}{meter}$$

$$\varepsilon ox = 3.453148e\text{-}11 \quad \frac{F}{meter}$$

If only model capacitance CAP is specified, then:

$$CAPeff = CAP \cdot SCALE(element) \cdot M$$

Capacitance Temperature Equation

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

$$C(t) = C \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

Δt	t - tnom
t	Element temperature in degrees Kelvin
	`t=circuit temp + DTEMP + 273.15`
tnom	Nominal temperature in degrees Kelvin
	`tnom+273.15 + TNOM`

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>…)
```

mname	Model name. Elements refer to the model by this name.
L	Identifies a saturable core model
CORE	Identifies a Jiles-Atherton Ferromagnetic Core model
level=x	Equation selection for Jiles-Atherton model
pname1=val1	Value of the model parameter. Each core model can include several model parameters.

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 l1 l2 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 l bs=6k br=3k hs=1 hcr=.1 hc=.8 ac=1 lc=16
.end
```

Magnetic Core Model Parameters

Name (Alias)	Units	Default	Description
AC	cm · 2	1.0	Core area
BS	Gauss	13000	Magnetic flux density at saturation
BR	Gauss	12000	Residual magnetization
HC	Oersted	0.8	Coercive magnetizing force
HCR	Oersted	0.6	Critical magnetizing force
HS	Oersted	1.5	Magnetizing force at saturation
LC	cm	3.0	Core length
LG	cm	0.0	Gap length
TC	s	0.0	Core growth time constant

Figure 14-1: Magnetic Saturable Core Model

Jiles-Atherton Core Model Parameters

Name (Alias)	Units	Default	Description
LEVEL		2	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
AREA, (AC)	cm^2	1	Mean of magnetic core cross section. AC is an alias of AREA
PATH, (LC)	cm	3	Mean of magnetic core path length. LC is an alias of PATH
MS	amp/ meter	1e6	Magnetization saturation
A	amp/ meter	1e3	Characterizes the shape of the anhysteretic magnetization
ALPHA		1e-3	Represents the coupling between the magnetic domains
C		0.2	Domain flexing parameter
K	amp/ meter	500	Domain an isotropy parameter

Magnetic Core Element Outputs

Output Variable	Description
LX1	magnetic field, *h* (oersted)
LX2	magnetic flux density, *b* (gauss)
LX3	slope of the magnetization curve, $\dfrac{dm}{dh}$

Output Variable	Description
LX4	bulk magnetization, m (amp/meter)
LX5	slope of the anhysteretic magnetization curve, $\dfrac{dm_{an}}{dh}$
LX6	anhysteretic magnetization, m_{an} (amp/meter)
LX7	effective magnetic field, h_e (amp/meter)

Inductor Device Equations

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:

$$L(t) = L \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

Δt	t - tnom
t	Element temperature in degrees Kelvin
	`t=circuit temp + DTEMP + 273.15`
tnom	Nominal temperature in degrees Kelvin
	`tnom=273.15 + TNOM`

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

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

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

Linear branch relation for transient analysis:

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

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

Linear branch relation for AC analysis:

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

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

Note: *You must define an inductor reference by a mutual inductor statement; otherwise 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   is

m_{an} Magnetization level, if domain walls could move freely.

h_e Effective magnetic field.

h Magnetic field.

MS Model parameter that represents the saturation magnetization.

A Model parameter that characterizes the shape of the anhysteretic magnetization.

$ALPHA$ 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:

$$\frac{dm_{an}}{dh} = \frac{1}{3 \cdot \dfrac{A}{MS} - ALPHA}$$

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

$$\frac{dm}{dh} = \frac{(m_{an} - m)}{(1 + C) \cdot K} + \frac{C}{1 + C} \cdot \frac{dm_{an}}{dh}$$

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 μ_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
.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(rl) 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

Figure 14-4: Variation of Hysteretic b-h Curve: as K Increases, the Loop Widens and Rotates Clockwise

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
l1 2 0 nt=50
k1 l1 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
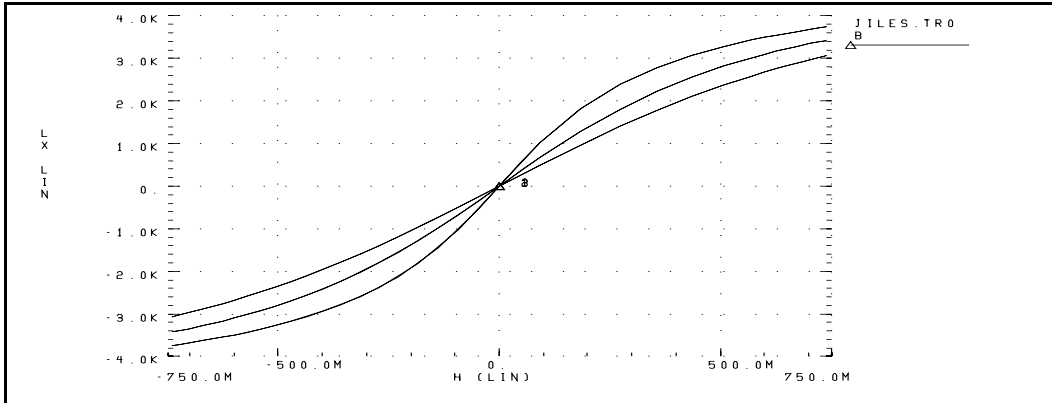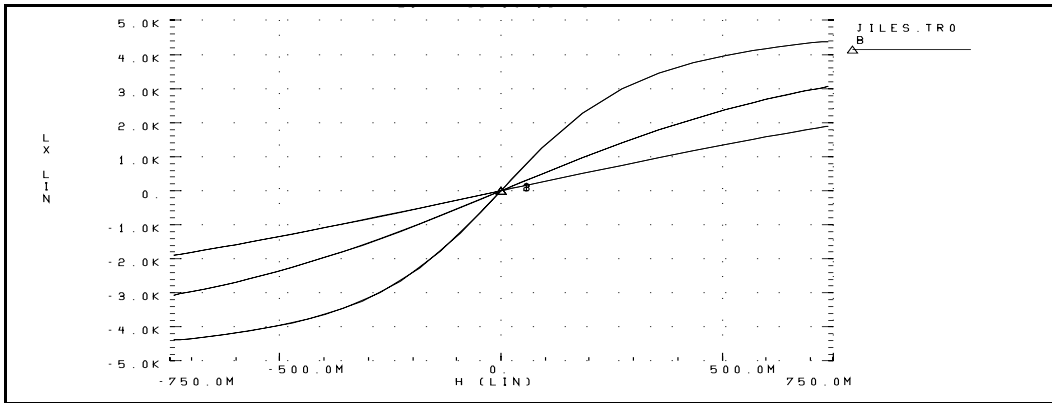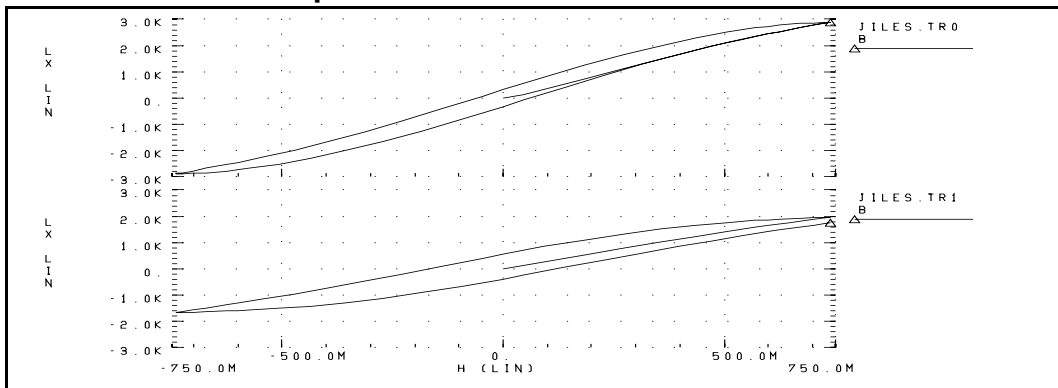
.probe b=lx2(k1) h=lx1(k1) i(rl) v(1)
.probe dmdh=lx3(k1) m=lx4(k1) dmandh=lx5(k1)
+ man=lx6(k1)
.probe l=lv1(l1) heff=lx7(k1)
.end
```

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)
```

```
+ 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
l1 2 0 nt=20
k1 l1 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(l1) 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
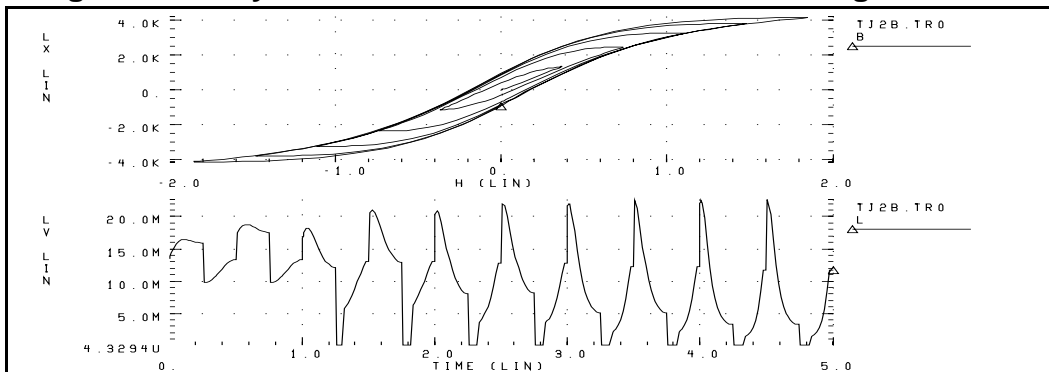
 norm of gradient < grad= 1.0000E-06 on last iterations
**** optimized parameters opt1

.param pms = 267.5975k
.param pa = 27.8196
.param pk = 37.2947
.param pc = 316.4197m
```

```
*** 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)

Avant!

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 Fowler-Nordheim Diode

■ 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:

Function	Control Options
Capacitance	DCAP, DCCAP
Conductance	GMIN, GMINDC
Geometry	SCALM, SCALE

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

Function	Parameters
Netlist	Dxxx, n+, n-, mname
Initialization	IC, OFF
Temperature	DTEMP
Geometric junction	AREA, L, M, PJ, W
Geometric capacitance (LEVEL=3 only)	LM, LP, WM, WP

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)

Function	Parameters
model type	LEVEL
DC parameters	IBV, IK, IKR, IS, ISW, N, RS, VB, RS
geometric junction	AREA, M, PJ
geometric capacitance (LEVEL=3 only)	L, LM, LP, SHRINK, W, WM, WP, XM, XOJ, XOM, XP, XW
capacitance	CJ, CJP, FC, FCS, M, MJSW, PB, PHP, TT
noise	AK, KF

Setting Junction DC Parameters in LEVEL 1 and 3

Name (Alias)	Units	Default	Description

AREA		1.0	Junction area
			For LEVEL=1
			`AREAeff = AREA · M, unitless`
			For LEVEL=3
			`AREAeff=AREA · SCALM`^{2}` · SHRINK`^{2}` · M`
			`unit = meter`^{2}
			If you specify W and L:
			`AREAeff = Weff  · Leff  · M unit =`
			`meter`^{2}

EXPLI	amp/ AREAeff	1e15	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. `EXPLIeff = EXPLI  · AREAeff`
IB	amp	1.0e-3	Current at breakdown voltage For LEVEL=3 `IBVeff = IBV  · AREAeff / SCALM`^{2}
IBV	amp	1.0e-3	Current at breakdown voltage For LEVEL=3 `IBVeff = IBV  · AREAeff / SCALM`^{2}
IK (IKF, JBF)	amp/ AREAeff	0.0	Forward knee current (intersection of the high- and low-current asymptotes) `IKeff = IK · AREAeff.`
IKR (JBR)	amp/ AREAeff	0.0	Reverse knee current (intersection of the high- and low-current asymptotes) `IKReff = IKR · AREAeff.`

Name (Alias)	Units	Default	Description
IS (JS)	amp/ AREAeff	LEVEL 1= 1.0e-14 LEVEL 3= 0.0	If you use an IS value less than EPSMIN, the program resets the value of IS to EPSMIN and displays a warning message. `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/SCALM`^{2}
JSW (ISP)	amp/ PJeff	0.0	Sidewall saturation current per unit junction periphery For LEVEL=1 `JSWeff = PJeff · JSW` For LEVEL=3 `JSWeff = PJeff · JSW/SCALM`
L			Default length of diode `Leff = L Þ SHRINK Þ SCALM+ XWeff`
LEVEL		1	Diode model selector LEVEL=1 or LEVEL=3 selects junction diode model LEVEL=2 selects Fowler-Nordheim model
N		1.0	Emission coefficient
PJ		0.0	Junction periphery For LEVEL=1 `PJeff = PJ · M, unitless` For LEVEL=3 `PJeff = PJ · SCALM Þ M Þ SHRINK, meter` If W and L are specified `PJeff = (2 · Weff + 2  · Leff) · M, meter`

Name (Alias)	Units	Default	Description
RS	ohms or ohms/m^2 (see note below)	0.0	Ohmic series resistance For LEVEL=1 RSeff = RS/AREAeff For LEVEL=3 RSeff= RS·SCALM^2/AREAeff
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 `XWeff = XW · SCALM`

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.*

Setting Junction Capacitance Parameters

Name (Alias)	Units	Default	Description

CJ (CJA, CJO)	F/ AREAeff	0.0	Zero-bias junction capacitance per unit junction bottomwall area
			For LEVEL=1
			`CJOeff = CJO · AREAeff`
			
			For LEVEL=3
			`CJeff = CJ · AREAeff/SCALM²`

CJP (CJSW)	F/PJeff	0.0	Zero-bias junction capacitance per unit junction periphery (PJ)
			For LEVEL=1
			`CJPeff = CJP · PJeff`
			
			For LEVEL=3
			`CJPeff = CJP · PJeff/SCALM`

FC		0.5	Coefficient for forward-bias depletion area capacitance formula
FCS		0.5	Coefficient for the forward-bias depletion periphery capacitance formula
M (EXA, MJ)		0.5	Area junction grading coefficient
MJSW (EXP)		0.33	Periphery junction grading coefficient
PB (PHI, VJ, PHA)	V	0.8	Area junction contact potential
PHP	V	PB	Periphery junction contact potential
TT	s	0.0	Transit time

Setting Metal and Poly Capacitor Parameters for LEVEL=3

Name (Alias)	Units	Default	Description
LM	m	0.0	Use this parameter when LM is not specified in the element statement. `LMeff = LM · SCALM · SHRINK`
LP	m	0.0	Use this parameter if LP is not specified in the element statement. `LPeff = LP · SCALM · SHRINK`
WM	m	0.0	Use this parameter if WM is not specified in the element statement. `WMeff = WM · SCALM · SHRINK`
WP	m	0.0	Use this parameter if WP is not specified in the element statement. `WPeff = WP · SCALM · SHRINK`
XM	m	0.0	XM accounts for masking and etching effects: `XMeff = XM · SCALM.`
XOI		10k	Thickness of the poly to bulk oxide
XOM	Å	10k	Thickness of the metal to bulk oxide
XP	m	0.0	Accounts for masking and etching effects `XPeff = XP · SCALM`

Setting Noise Parameters for LEVEL=1 and 3

Name (Alias)	Units	Default	Description
AF		1.0	Flicker noise exponent
KF		0.0	Flicker noise coefficient

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.

```
PJeff=PJ·M
```

PJeff is then used to scale CJP, the zero-bias junction capacitance, and the sidewall saturation current, JSW.

```
CJPeff=PJeff·CJP
JSWeff=PJeff·JSW
```

AREA and M are used to obtain AREAeff.

```
AREAeff =AREA·M
```

CJO, IK, IKR, IBV, and IS are multiplied by AREAeff to obtain their effective scaled values. RS, however, is divided by AREAeff.

```
IKeff=AREAeff·IK
IKReff=AREAeff·IKR
IBVeff=AREAeff·IBV
ISeff=AREAeff·IS
RSeff=RS/AREAeff
CJOeff= CJO·AREAeff
```

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:

$$\text{AREAeff} = \text{AREA} \cdot \text{M} \cdot \text{SCALE}^2 \cdot \text{SHRINK}^2$$
$$\text{PJeff} = \text{PJ} \cdot \text{SCALE} \cdot \text{M} \cdot \text{SHRINK}$$

or, if W and L are specified:

$$\text{AREAeff} = \text{Weff} \cdot \text{Leff} \cdot \text{M}$$
$$\text{PJeff} = (2 \cdot \text{Weff} + 2 \cdot \text{Leff}) \cdot \text{M}$$

where

$$\text{Weff} = \text{W} \cdot \text{SCALE} \cdot \text{SHRINK} + \text{XWeff}$$
$$\text{Leff} = \text{L} \cdot \text{SCALE} \cdot \text{SHRINK} + \text{XWeff}$$

To find the value of JSWeff and CJPeff use the formula:

$$\text{JSWeff} = \text{PJeff} \cdot (\text{JSW/SCALM})$$
$$\text{CJPeff} = \text{PJeff} \cdot (\text{CJP/SCALM})$$

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:

$$\text{IKeff} = \text{AREAeff} \cdot \text{IK}$$

```
IKReff= AREAeff · IKR
```
$$IBVeff=(AREAeff \cdot IBV)/SCALM^2$$
$$ISeff=IS \cdot (AREAeff/SCALM^2)$$
$$RSeff=RS/(AREAeff \cdot SCALM^2)$$
$$CJOeff=AREAeff \cdot (CJO/SCALM^2)$$

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 *forward biased* or turned on. The anode is the p-doped side of a diode, and the cathode is the n-doped side.

Figure 15-1: Diode Current Convention

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 *(id)*. For noise and AC analyses, the actual *id* current is not used. The partial derivative of *id* with respect to the terminal voltage *vd* is used instead. The name for this partial derivative is:

Conductance

$$gd = \frac{\partial id}{\partial vd}$$

The drain current (id) equation accounts for all basic DC effects of the diodes. Star-Hspice assumes capacitance effects to be separate from the id 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)

Variable	Parameter
Resistance coefficient	TRS
Capacitance coefficient	CTA, CTP
Energy gap	EG, GAP1, GAP2
Transit time coefficient	TTT1, TTT2
Reference temperature	TREF
Temperature selectors	TLEV, TLEVC
Miscellaneous	TM1, TM2, TPB, TPHP
Saturation current	XT1

Setting Temperature Effect Parameters LEVEL=1 and 3

Name (Alias)	Units	Default	Description
CTA (CTC)	1/°	0.0	Temperature coefficient for area junction capacitance (CJ). Set parameter TLEVC to 1 to let CTAI override default temperature coefficient.
CTP	1/°	0.0	Temperature coefficient for periphery junction capacitance (CJP). Set TLEVC to 1 to let CTP override default temperature coefficient.

Name (Alias)	Units	Default	Description
EG	eV		Energy gap for pn junction diode For TLEV=0, 1, default=1.11, for TLEV=2, default=1.16 1.17 - silicon 0.69 - Schottky barrier diode 0.67 - germanium 1.52 - gallium arsenide
GAP1	eV/°	7.02e-4	7.02e-4 - silicon (old value) 4.73e-4 - silicon 4.56e-4 - germanium 5.41e-4 - gallium arsenide
GAP2	°×	1108	1108 - silicon (old value) 636 - silicon 210 - germanium 204 - gallium arsenide
TCV	1/°	0.0	Breakdown voltage temperature coefficient
TLEV		0.0	Temperature equation selector for diode; interacts with TLEVC
TLEVC		0.0	Level selector for diode temperature, junction capacitances and contact potentials; interacts with TLEV
TM1	1/°	0.0	First-order temperature coefficient for MJ
TM2	$1/°^2$	0.0	Second-order temperature coefficient for MJ
TPB (TVJ)	V/°	0.0	Temperature coefficient for PB. Set parameter TLEVC to 1 or 2 to enable TPB to override default temperature compensation.
TPHP	V/°	0.0	Temperature coefficient for PHP. Set parameter TLEVC to 1 or 2 to enable TPHP to override default temperature compensation.

Name (Alias)	Units	Default	Description
TREF	×	25.0	Model reference temperature (LEVEL 1 or 3 only)
TRS	1/°	0.0	Resistance temperature coefficient
TTT1	1/°	0.0	First order temperature coefficient for TT
TTT2	1/°^{2}	0.0	Second order temperature coefficient for TT
XTI		3.0	Saturation current temperature exponent. Set XTI=3.0 for silicon-diffused junction. Set XTI=2.0 for Schottky barrier diode.

Using Junction Diode Equations

Table 15-4 shows the diode equation variable definition.

Table 15-4: Equation Variable Definitions

Variable	Definition
cd	total diode capacitance
f	frequency
gd	diode conductance
id	diode DC current
id1	current without high level injection
ind	diode equivalent noise current
inrs	series resistor equivalent noise current
vd	voltage across the diode

Table 15-5 shows the equation quantity definition:

Table 15-5: Equation Quantity Definition

Quantity	Definition
tox	3.453143e-11 F/m
k	1.38062e-23 (Boltzmann's constant)
q	1.60212e-19 (electron charge)
t	temperature in °Kelvin
Δt	t - tnom
tnom	nominal temperature of parameter measurements in °Kelvin
vt(t)	$k \cdot$ t/q: thermal voltage
vt(tnom)	$k \cdot$ tnom/q: thermal voltage

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: vd > -10 · vt

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

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

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: BVeff < vd < -10 · vt

$$id = -ISeff$$

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

Note: BV is always described as a positive number.

Breakdown: vd < - BVeff

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

The BV parameter is adjusted as follows to obtain BVeff:

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

If IBVeff > ibreak, then,

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

Otherwise,

$$IBVeff \ = \ ibreak$$

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

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

Forward Bias

$$id \ = \ \frac{id1}{1 + \left(\frac{id1}{IKeff}\right)^{1/2}}$$

Reverse Bias

$$id = \frac{id1}{1 + \left(\dfrac{id1}{IKReff}\right)^{1/2}}$$

where id1 is

For vd $\geq$ -BV$_{eff}$:

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

Otherwise:

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

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.

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

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:

vd < FC · PB

$$cdepa = CJeff \cdot \left(1 - \frac{vd}{PB}\right)^{-MJ}$$

vd Š FC · PB

$$cdepa = CJeff \cdot \frac{1 - FC \cdot (1 + MJ) + MJ \cdot \dfrac{vd}{PB}}{(1 - FC)^{(1 + MJ)}}$$

The junction periphery capacitance formula is:

vd < FCS · PHP

$$cdepp = CJPeff \cdot \left(1 - \frac{vd}{PHP}\right)^{-MJSW}$$

vd Š FCS · PHP

$$cdepp = CJPeff \cdot \frac{1 - FCS \cdot (1 + MJSW) + MJSW \cdot \dfrac{vd}{PHP}}{(1 - FCS)^{(1 + MJSW)}}$$

then,

$$cdep = cdepa + cdepp$$

DCAP=2 (default)

The total depletion capacitance formula is:

vd < 0

$$cdep = CJeff \cdot \left(1 - \frac{vd}{PB}\right)^{-MJ} + CJPeff \cdot \left(1 - \frac{vd}{PHP}\right)^{-MJSW}$$

vd Š 0

$$cdep = CJeff \cdot \left(1 + MJ \cdot \frac{vd}{PB}\right) + CJPeff \cdot \left(1 + MJSW \cdot \frac{vd}{PHP}\right)$$

DCAP=3

Limits peak depletion capacitance to FC · CGDeff or FC · CGSeff, with proper fall-off when forward bias exceeds PB (FC ≥ 1).

Metal and Poly Capacitance Equations (LEVEL=3 Only)

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

$$cmetal = \left(\frac{\varepsilon_{ox}}{XOI}\right) \cdot (WPeff + XPeff) \cdot (LPeff + XPeff) \cdot M$$

$$cpoly = \left(\frac{\varepsilon_{ox}}{XOM}\right) \cdot (WMeff + XMeff) \cdot (LMeff + XMeff) \cdot M$$

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:

$$inrs = \left(\frac{4 \cdot k \cdot t}{RSeff} \right)^{1/2}$$

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:

$$ind = \left(2 \cdot q \cdot id + \frac{KF \cdot id^{AF}}{f} \right)^{1/2}$$

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

$$egnom = 1.16 - 7.02e\text{-}4 \cdot \frac{tnom^2}{tnom + 1108.0}$$

$$eg(t) = 1.16 - 7.02e\text{-}4 \cdot \frac{t^2}{t + 1108.0}$$

TLEV=2

$$egnom = EG - GAP1 \cdot \frac{tnom^2}{tnom + GAP2}$$

$$eg(t) = EG - GAP1 \cdot \frac{t^2}{t + GAP2}$$

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

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

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

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

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

TLEVC=1 or 2

$$PB(t) = PB - TPB \cdot \Delta t$$

$$PHP(t) = PHP - TPHP \cdot \Delta t$$

TLEVC=3

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

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

where TLEV=0 or 1

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

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

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

$$CJ(t) \; = \; CJ \cdot \left[1 + MJ \cdot \left(4.0e\text{-}4 \cdot \Delta t - \frac{PB(t)}{PB} + 1 \right) \right]$$

$$CJSW(t) \; = \; CJSW \cdot \left[1 + MJSW \cdot \left(4.0e\text{-}4 \cdot \Delta t - \frac{PHP(t)}{PHP} + 1 \right) \right]$$

TLEVC=1

$$CJ(t) \; = \; CJ \cdot (1 + CTA \cdot \Delta t)$$

$$CJSW(t) \; = \; CJSW \cdot (1 + CTP \cdot \Delta t)$$

TLEVC=2

$$CJ(t) \; = \; CJ \cdot \left(\frac{PB}{PB(t)} \right)^{MJ}$$

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

$$CJSW(t) \; = \; CJSW \cdot \left(\frac{PHP}{PHP(t)} \right)^{MJSW}$$

TLEVC=3

$$CJ(t) \; = \; CJ \cdot \left(1 - 0.5 \cdot dpbdt \cdot \frac{\Delta t}{PB} \right)$$

$$CJSW(t) \; = \; CJSW \cdot \left(1 - 0.5 \cdot dphpdt \cdot \frac{\Delta t}{PHP} \right)$$

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 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 15-6 shows the Fowler-Nordheim diode model parameters for LEVEL 2.

Table 15-6: Fowler-Nordheim Diode Model Parameters

Name (alias)	Units	Default	Description
EF	V/cm	1.0e8	Forward critical electric field
ER	V/cm	EF	Reverse critical electric field
JF	amp/V^2	1.0e-10	Forward Fowler-Nordheim current coefficient
JR	amp/V^2	JF	Reverse Fowler-Nordheim current coefficient
L	m	0.0	Length of diode for calculation of Fowler-Nordheim current `Leff = L · SCALM  · SHRINK + XWeff`
TOX	Å	100.0	Thickness of oxide layer
W	m	0.0	Width of diode for calculation of Fowler-Nordheim current `Weff = W · SCALM · SHRINK + XWeff`
XW	m	0.0	`XWeff = XW · SCALM`

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:

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

Forward Bias: vd Š 0

$$id = AREAeff \cdot JF \cdot \left(\frac{vd}{TOX}\right)^2 \cdot e^{\frac{-EF \cdot TOX}{vd}}$$

Reverse Bias: vd < 0

$$id = -AREAeff \cdot JR \cdot \left(\frac{vd}{TOX}\right)^2 \cdot e^{\frac{ER \cdot TOX}{vd}}$$

Fowler-Nordheim Diode Capacitances

The Fowler-Nordheim diode capacitance is a constant derived from:

$$cd = AREAeff \cdot \frac{\varepsilon_{ox}}{TOX}$$

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
```

Avant!

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 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:

Function	Control Options
Capacitance	DCAP, DCCAP
Conductance	GMIN, GMINDC

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
pname1	Each BJT model can include several model parameters.
PNP	Identifies a PNP transistor model

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

DC	BF, BR, IBC, IBE, IS, ISS, NF, NR, NS, VAF, VAR
beta degradation	ISC, ISE, NC, NE, IKF, IKR
geometric	SUBS, BULK
resistor	RB, RBM, RE, RC, IRB
junction capacitor	CJC, CJE, CJS, FC, MJC, MJE, MJS, VJC, VJE, VJS, XCJC
parasitic capacitance	CBCP, CBEP, CCSP
transit time	ITF, PTF, TF, VT, VTF, XTF
noise	KF, AF

Using BJT Basic DC Model Parameters

Name (Alias)	Unit	Default	Description
BF (BFM)		100.0	Ideal maximum forward Beta
BR (BRM)		1.0	Ideal maximum reverse Beta
BULK (NSUB)		0.0	Sets the bulk node to a global node name. A substrate terminal node name (ns) in the element statement overrides BULK.

Name (Alias)	Unit	Default	Description
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. `IBCeff = IBC · AREAB · M` AREAC replaces AREAB, 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. `EXPLIeff = EXPLI · AREAeff`
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. `IBEeff = IBE · AREA · 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. `ISeff = IS · AREA · M`
ISS	amp	0.0	Reverse saturation current bulk-to-collector or bulk-to-base, depending on vertical or lateral geometry selection `SSeff = ISS · AREA · M`

Name (Alias)	Unit	Default	Description
LEVEL		1.0	Model selector
NF		1.0	Forward current emission coefficient
NR		1.0	Reverse current emission coefficient
NS		1.0	Substrate current emission coefficient
SUBS			Substrate connection selector: +1 for vertical geometry, -1 for lateral geometry default=1 for NPN, default=-1 for PNP
UPDATE		0	UPDATE = 1 selects alternate base charge equation

Using Low-Current Beta Degradation Effect Parameters

Name (Alias)	Unit	Default	Description

ISC (C4, JLC)	amp	0.0	Base-collector leakage saturation current. If ISC is greater than 1e-4, then:
			
			`ISC = IS · ISC`
			
			otherwise:
			
			`ISCeff = ISC · AREAB · 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 · ISE`
			
			otherwise:
			
			`ISEeff = ISE · AREA · M`

NC (NLC)		2.0	Base-collector leakage emission coefficient
NE (NLE)		1.5	Base-emitter leakage emission coefficient

Using Base Width Modulation Parameters

Name (Alias)	Unit	Default	Description
VAF (VA, VBF)	V	0.0	Forward early voltage. Use zero to indicate an infinite value.
VAR (VB, VRB, BV)	V	0.0	Reverse early voltage. Use zero to indicate an infinite value.

Using High-Current Beta Degradation Effect Parameters

Name (Alias)	Unit	Default	Description
IKF (IK, JBF)	amp	0.0	Corner for forward Beta high-current roll-off. Use zero to indicate an infinite value. `IKFeff = IKF · AREA · M`
IKR (JBR)	amp	0.0	Corner for reverse Beta high-current roll-off. Use zero to indicate an infinite value `IKReff = IKR · AREA · M`
NKF		0.5	Exponent for high-current Beta roll-off

Using Parasitic Resistance Parameters

Name (Alias)	Unit	Default	Description
IRB (JRB, IOB)	amp	0.0	Base current, where base resistance falls half-way to RBM. Use zero to indicate an infinite value. `IRBeff = IRB · AREA · M`
RB	ohm	0.0	Base resistance `RBeff = RB / (AREA · M)`
RBM	ohm	RB	Minimum high-current base resistance `RBMeff = RBM / (AREA · M)`
RE	ohm	0.0	Emitter resistance `REeff = RE / (AREA · M)`
RC	ohm	0.0	Collector resistance `RCeff = RC / (AREA · M)`

Using Junction Capacitor Parameters

Name (Alias)	Unit	Default	Description
CJC	F	0.0	Base-collector zero-bias depletion capacitance Vertical: `CJCeff = CJC·AREAB·M` Lateral: `CJCeff = CJC ·AREAC·M`
CJE	F	0.0	Base-emitter zero-bias depletion capacitance (vertical and lateral): `CJEeff = CJE·AREA·M`
CJS (CCS, CSUB)	F	0.0	Zero-bias collector substrate capacitance Vertical: `CJSeff = CJS ·AREAC·M` Lateral: `CJSeff = CJS·AREAB·M`
FC		0.5	Coefficient for forward bias depletion capacitance formula for DCAP=1 DCAP Default=2 and FC is ignored
MJC (MC)		0.33	Base-collector junction exponent (grading factor)
MJE (ME)		0.33	Base-emitter junction exponent (grading factor)
MJS(ESUB)		0.5	Substrate junction exponent (grading factor)
VJC (PC)	V	0.75	Base-collector built-in potential
VJE (PE)	V	0.75	Base-emitter built-in potential
VJS (PSUB)	V	0.75	Substrate junction built in potential
XCJC (CDIS)		1.0	Internal base fraction of base-collector depletion capacitance

Using Parasitic Capacitances Parameters

Name	Unit	Default	Description
CBCP	F	0.0	External base-collector constant capacitance `CBCPeff = CBCP · AREA · M`
CBEP	F	0.0	External base-emitter constant capacitance `CBEPeff = CBEP · AREA · M`
CCSP	F	0.0	External collector substrate constant capacitance (vertical) or base substrate (lateral) `CCSPeff = CCSP · AREA · M`

Using Transit Time Parameters

Name (Alias)	Unit	Default	Description
ITF (JTF)	amp	0.0	TF high-current parameter `ITFeff = ITF · AREA · M`
PTF	x	0.0	Frequency multiplier to determine excess phase
TF	s	0.0	Base forward transit time
TR	s	0.0	Base reverse transit time
VTF	V	0.0	TF base-collector voltage dependence coefficient. Zero indicates an infinite value.
XTF		0.0	TF bias dependence coefficient

Using Noise Parameters

Name	Unit	Default	Description
AF		1.0	Flicker-noise exponent
KF		0.0	Flicker-noise coefficient

Using BJT LEVEL=2 Model Parameters

Name	Unit	Default	Description
BRS		1.0	Reverse beta for substrate BJT.
GAMMA		0.0	Epitaxial doping factor, $GAMMA = (2 \cdot ni / n)^2$ where n is epitaxial impurity concentration
NEPI		1.0	Emission coefficient
QCO	Coul	0.0	Epitaxial charge factor Vertical: `QCOeff=QCO·AREAB·M` Lateral: `QCOeff=QCO·AREAC·M`
RC	ohm	0.0	Resistance of the epitaxial region under equilibrium conditions `RCeff=RC/(AREA·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

Function	Parameter
base modulation	TVAF1, TVAF2, TVAR1, TVAR2
capacitor	CTC, CTE, CTS
capacitor potentials	TVJC, TVJE, TVJS
DC	TBF1, TBF2, TBR1, TBR2, TIKF1, TIKF2, TIKR1, TIKR2, TIRB1, TIRB2, TISC1, TISC2, TIS1, TIS2, TISE1, TISE2, TISS1, TISS2, XTB, XTI
emission coefficients	TNC1, TNC2, TNE1, TNE2, TNF1, TNF2, TNR1, TNR2, TNS1, TNS2
energy gap	EG, GAP1, GAP2
equation selectors	TLEV, TLEVC
grading	MJC, MJE, MJS, TMJC1, TMJC2, TMJE1, TMJE2, TMJS1, TMJS2
resistors	TRB1, TRB2, TRC1, TRC2, TRE1, TRE2, TRM1, TRM2
transit time	TTF1, TTF2, TTR1, TTR2

Using Temperature Effect Parameters

Name (Alias)	Unit	Default	Description
BEX		2.42	VO temperature exponent (LEVEL 2 only)
BEXV		1.90	RC temperature exponent (LEVEL 2 only)

Name (Alias)	Unit	Default	Description
CTC	1/$^\circ$	0.0	Temperature coefficient for zero-bias base collector capacitance. TLEVC=1 enables CTC to override the default Star-Hspice temperature compensation.
CTE	1/$^\circ$	0.0	Temperature coefficient for zero-bias base emitter capacitance. TLEVC=1 enables CTE to override the default Star-Hspice temperature compensation.
CTS	1/$^\circ$	0.0	Temperature coefficient for zero-bias substrate capacitance. TLEVC=1 enables CTS to override the default Star-Hspice temperature compensation.
EG	eV		Energy gap for pn junction for TLEV=0 or 1, default=1.11; for TLEV=2, default=1.16 1.17 - silicon 0.69 - Schottky barrier diode 0.67 - germanium 1.52 - gallium arsenide
GAP1	eV/$^\circ$	7.02e-4	First bandgap correction factor (from Sze, alpha term) 7.02e-4 - silicon 4.73e-4 - silicon 4.56e-4 - germanium 5.41e-4 - gallium arsenide

Name (Alias)	Unit	Default	Description
GAP2	x	1108	Second bandgap correction factor (from Sze, beta term) 1108 - silicon 636 - silicon 210 - germanium 204 - gallium arsenide
MJC (MC)		0.33	Base-collector junction exponent (grading factor)
MJE (ME)		0.33	Base-emitter junction exponent (grading factor)
MJS (ESUB)		0.5	Substrate junction exponent (grading factor)
TBF1	$1/^\circ$	0.0	First-order temperature coefficient for BF
TBF2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for BF
TBR1	$1/^\circ$	0.0	First-order temperature coefficient for BR
TBR2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for BR
TIKF1	$1/^\circ$	0.0	First-order temperature coefficient for IKF
TIKF2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for IKF
TIKR1	$1/^\circ$	0.0	First-order temperature coefficient for IKR
TIKR2	$1/^{\circ 2}$		Second-order temperature coefficient for IKR
TIRB1	$1/^\circ$	0.0	First-order temperature coefficient for IRB
TIRB2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for IRB
TISC1	$1/^\circ$	0.0	First-order temperature coefficient for ISC TLEV=3 enables TISC1.
TISC2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for ISC TLEV=3 enables TISC2.

Name (Alias)	Unit	Default	Description
TIS1	$1/^\circ$	0.0	First-order temperature coefficient for IS or IBE and IBC TLEV=3 enables TIS1.
TIS2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for IS or IBE and IBC TLEV=3 enables TIS2.
TISE1	$1/^\circ$	0.0	First-order temperature coefficient for ISE TLEV=3 enables TISE1.
TISE2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for ISE. TLEV=3 enables TISE2.
TISS1	$1/^\circ$	0.0	First-order temperature coefficient for ISS TLEV=3 enables TISS1.
TISS2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for ISS TLEV=3 enables TISS2.
TITF1			First-order temperature coefficient for ITF
TITF2			Second-order temperature coefficient for ITF
TLEV		1	Temperature equation level selector for BJTs (interacts with TLEVC)
TLEVC		1	Temperature equation level selector for BJTs, junction capacitances and potentials (interacts with TLEV)
TMJC1	$1/^\circ$	0.0	First-order temperature coefficient for MJC
TMJC2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for MJC
TMJE1	$1/^\circ$	0.0	First order temperature coefficient for MJE
TMJE2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for MJE
TMJS1	$1/^\circ$	0.0	First-order temperature coefficient for MJS
TMJS2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for MJS
TNC1	$1/^\circ$	0.0	First-order temperature coefficient for NC
TNC2		0.0	Second-order temperature coefficient for NC

Name (Alias)	Unit	Default	Description
TNE1	$1/^{\circ}$	0.0	First-order temperature coefficient for NE
TNE2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for NE
TNF1	$1/^{\circ}$	0.0	First-order temperature coefficient for NF
TNF2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for NF
TNR1	$1/^{\circ}$	0.0	First-order temperature coefficient for NR
TNR2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for NR
TNS1	$1/^{\circ}$	0.0	First-order temperature coefficient for NS
TNS2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for NS
TRB1 (TRB)	$1/^{\circ}$	0.0	First-order temperature coefficient for RB
TRB2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for RB
TRC1 (TRC)	$1/^{\circ}$	0.0	First-order temperature coefficient for RC
TRC2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for RC
TRE1 (TRE)	$1/^{\circ}$	0.0	First-order temperature coefficient for RE
TRE2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for RE
TRM1	$1/^{\circ}$	TRB1	Firs-order temperature coefficient for RBM
TRM2	$1/^{\circ 2}$	TRB2	Second-order temperature coefficient for RBM
TTF1	$1/^{\circ}$	0.0	First-order temperature coefficient for TF
TTF2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for TF
TTR1	$1/^{\circ}$	0.0	First-order temperature coefficient for TR
TTR2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for TR
TVAF1	$1/^{\circ}$	0.0	First-order temperature coefficient for VAF
TVAF2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for VAF
TVAR1	$1/^{\circ}$	0.0	First-order temperature coefficient for VAR
TVAR2	$1/^{\circ 2}$	0.0	Second-order temperature coefficient for VAR

Name (Alias)	Unit	Default	Description
TVJC	V/°	0.0	Temperature coefficient for VJC. TLEVC=1 or 2 enables TVJC to override the default Star Hspice temperature compensation.
TVJE	V/°	0.0	Temperature coefficient for VJE. TLEVC=1 or 2 enables TVJE to override the default Star Hspice temperature compensation.
TVJS	V/°	0.0	Temperature coefficient for VJS. TLEVC=1 or 2 enables TVJS to override the default Star Hspice temperature compensation.
XTB (TB, TCB)		0.0	Forward and reverse Beta temperature exponent (used with TLEV=0, 1 or 2)
XTI		3.0	Saturation current temperature exponent. Use XTI = 3.0 for silicon diffused junction. Set XTI = 2.0 for Schottky barrier diode.

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:

$$Ieff \ = \ AREA \cdot M \cdot I$$

where I is either IBE, IS, ISE, IKF, IKR, or IRB.

For both the vertical and lateral, the resistor model parameters, RB, RBM, RE, and RC are scaled by the following equation.

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

where R is either RB, RBM, RE, or RC.

Understanding 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.

Figure 16-1: BJT Current Convention

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

$$g\mu = \left.\frac{\partial ib}{\partial vbc}\right|_{vbe = const.}$$

Forward Base Conductance

$$g\pi = \left.\frac{\partial ib}{\partial vbe}\right|_{vbc = const.}$$

Collector Conductance

$$g_o = \left.\frac{\partial ic}{\partial vce}\right|_{vbe\,=\,const.} = \left.-\frac{\partial ic}{\partial vbc}\right|_{vbe\,=\,const.}$$

Transconductance

$$gm = \left.\frac{\partial ic}{\partial vbe}\right|_{vce\,=\,cons}$$

$$= \frac{\partial ic}{\partial vbe} + \frac{\partial ic}{\partial vbc}$$

$$= \frac{\partial ic}{\partial vbe} - g_o$$

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

Variable	Definitions
cbc	Internal base to collector capacitance
cbcx	External base to collector capacitance
cbe	Internal base to emitter capacitance
csc	Substrate to collector capacitance (vertical transistor only)
cbs	Base to substrate capacitance (lateral transistor only)
f	Frequency
gbc	Reverse base conductance
gbe	Forward base conductance
gm	Transconductance

Table 16-3: Equation Variable Names

Variable	Definitions
gsc	Substrate to collector conductance (vertical transistor only)
go	Collector conductance
gbs	Base to substrate conductance (lateral transistor only)
ib	External base terminal current
ibc	DC current base to collector
ibe	DC current base to emitter
ic	External collector terminal current
ice	DC current collector to emitter
inb	Base current equivalent noise
inc	Collector current equivalent noise
inrb	Base resistor current equivalent noise
inrc	Collector resistor equivalent noise
inre	Emitter resistor current equivalent noise
ibs	DC current base to substrate (lateral transistor only)
isc	DC current substrate to collector (vertical transistor only)
qb	Normalized base charge
rb	Base resistance
rbb	Short-circuit base resistance
vbs	Internal base substrate voltage
vsc	Internal substrate collector voltage

Table 16-4: Equation Constants

Quantities	Definitions
k	1.38062e-23 (Boltzmann's constant)
q	1.60212e-19 (electron charge)
t	temperature in $^\circ$Kelvin
Δt	t - tnom
tnom	tnom = 273.15 + TNOM in $^\circ$Kelvin
vt(t)	$k \cdot t/q$
vt(tmon)	$k \cdot tnom/q$

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.

Figure 16-8: Vertical Transistor (SUBS = +1)

Figure 16-9: Lateral Transistor (SUBS = -1)

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:

$$ic = \frac{ISeff}{qb} \cdot \left(e^{\frac{vbe}{NF \cdot vt}} - e^{\frac{vbc}{NR \cdot vt}} \right) - \frac{ISeff}{BR} \cdot \left(e^{\frac{vbc}{NR \cdot vt}} - 1 \right) - ISCeff \cdot \left(e^{\frac{vbc}{NC \cdot vt}} - 1 \right)$$

$$ib = \frac{ISeff}{BF} \cdot \left(e^{\frac{vbe}{NF \cdot vt}} - 1 \right) + \frac{ISeff}{BR} \cdot \left(e^{\frac{vbc}{NR \cdot vt}} - 1 \right) + ISEeff \cdot \left(e^{\frac{vbe}{NE \cdot vt}} - 1 \right)$$

$$+ ISCeff \cdot \left(e^{\frac{vbc}{NC \cdot vt}} - 1 \right)$$

Current Equations - IBE and IBC

If IBE and IBC are specified, instead of IS:

$$ic = \frac{IBEeff}{qb} \cdot \left(e^{\frac{vbe}{NF \cdot vt}} - 1\right) - \frac{IBCeff}{qb} \cdot \left(e^{\frac{vbc}{NR \cdot vt}} - 1\right) - \frac{IBCeff}{BR} \cdot \left(e^{\frac{vbc}{NR \cdot vt}} - 1\right)$$

$$- ISCeff \cdot \left(e^{\frac{vbc}{NC \cdot vt}} - 1\right)$$

$$ib = \frac{IBEeff}{BF} \cdot \left(e^{\frac{vbe}{NF \cdot vt}} - 1\right) + \frac{IBCeff}{BR} \cdot \left(e^{\frac{vbc}{NR \cdot vt}} - 1\right) + ISEeff \cdot \left(e^{\frac{vbe}{NE \cdot vt}} - 1\right)$$

$$+ ISCeff \cdot \left(e^{\frac{vbc}{NC \cdot vt}} - 1\right)$$

$IBCeff = IBC \cdot AREAB \cdot M$ *Vertical*

$IBCeff = IBC \cdot AREAC \cdot M$ *Lateral*

$IBEeff = IBE \cdot AREA \cdot M$ *Vertical or Lateral*

$ISCeff = ISC \cdot AREAB \cdot M$ *Vertical*

$ISCeff = ISC \cdot AREAC \cdot M$ *Lateral*

$ISEeff = ISE \cdot AREA \cdot M$ *Vertical or Lateral*

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

Using Substrate Current Equations

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

Vertical Transistors

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

$$isc = -ISSeff \qquad\qquad\qquad vsc \leq -10 \cdot NS \cdot vt$$

Lateral Transistors

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

$$ibs = -ISSeff \qquad\qquad\qquad vbs \leq -10 \cdot NS \cdot vt$$

If both IBE and IBC are *not* specified:

$$ISSeff = ISS \cdot AREA \cdot M$$

If both IBE and IBC are specified:

$$ISSeff = ISS \cdot AREAC \cdot M \quad \text{vertical}$$

$$ISSeff = ISS \cdot AREAB \cdot M \quad \text{lateral}$$

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 $\dfrac{vbc}{VAF} + \dfrac{vbe}{VAR} < 0$, then

$$q1 = \cfrac{1}{\left(1 - \dfrac{vbc}{VAF} - \dfrac{vbe}{VAR}\right)}$$

Otherwise, if UPDATE=1 and $\dfrac{vbc}{VAF} + \dfrac{vbe}{VAR} \geq 0$, then

$$q1 = 1 + \frac{vbc}{VAF} + \frac{vbe}{VAR}$$

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

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

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:

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

If IRB is specified:

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

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

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 cbe by the following formula:

$$cbe = cbediff + cbedep$$

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

Note: *When you run a DC sweep on a BJT, use .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:

ibe $\leq$ 0

$$cbediff = \frac{\partial}{\partial vbe}\left(TF \cdot \frac{ibe}{qb}\right)$$

ibe > 0

$$cbediff = \frac{\partial}{\partial vbe}\left[TF \cdot (1 + argtf) \cdot \frac{ibe}{qb}\right]$$

where:

$$argtf = XTF \cdot \left(\frac{ibe}{ibe + ITF} \right)^2 \cdot e^{\frac{vbc}{1.44 \cdot VTF}}$$

The forward part of the collector-emitter branch current is determined as follows:

$$ibe = ISeff \cdot \left(e^{\frac{vbe}{NF \cdot vt}} - 1 \right)$$

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

$$cbedep = CJEeff \cdot \left(1 - \frac{vbe}{VJE} \right)^{-MJE}$$

vbe $\geq$ FC $\cdot$ VJE

$$cbedep = CJEeff \cdot \frac{1 - FC \cdot (1 + MJE) + MJE \cdot \frac{vbe}{VJE}}{(1 - FC)^{(1 + MJE)}}$$

DCAP=2

The base-emitter depletion capacitance is determined as follows:

vbe < 0

$$cbedep \ = \ CJEeff \cdot \left(1 - \frac{vbe}{VJE} \right)^{-MJE}$$

vbe ≥ 0

$$cbedep \ = \ CJEeff \cdot \left(1 + MJE \cdot \frac{vbe}{VJE} \right)$$

DCAP=3

Limits peak depletion capacitance to FC · CJCeff or FC · CJEeff, with proper fall-off when forward bias exceeds PB (FC ≥ 1).

Determining Base Collector Capacitance

Determine the base collector capacitance cbc as follows:

$$cbc \ = \ cbcdiff + cbcdep$$

where *cbcdiff* and *cbcdep* are the base-collector diffusion and depletion capacitances, respectively.

Determining Base Collector Diffusion Capacitance

$$cbcdiff \ = \ \frac{\partial}{\partial vbc}(TR \cdot ibc)$$

where the internal base-collector current ibc is:

$$ibc \ = \ ISeff \cdot \left(e^{\frac{vbc}{NR \cdot vt}} - 1 \right)$$

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:

vbc < FC · VJC

$$cbcdep \ = \ XCJC \cdot CJCeff \cdot \left(1 - \frac{vbc}{VJC}\right)^{-MJC}$$

vbc ≥ FC · VJC

$$cbcdep \ = \ XCJC \cdot CJCeff \cdot \frac{1 - FC \cdot (1 + MJC) + MJC \cdot \dfrac{vbc}{VJC}}{(1 - FC)^{(1 + MJC)}}$$

DCAP=2

Specify DCAP=2 to select one of the following equations:

vbc < 0

$$cbcdep \ = \ XCJC \cdot CJCeff \cdot \left(1 - \frac{vbc}{VJC}\right)^{-MJC}$$

vbc ≥ 0

$$cbcdep \ = \ XCJC \cdot CJCeff \cdot \left(1 + MJC \cdot \frac{vbc}{VJC}\right)$$

External Base — Internal Collector Junction Capacitance

The base-collector capacitance is modeled as a distributed capacitance when the model parameter XCJC is set. Since the default setting of XCJC is one, the entire base-collector capacitance is on the internal base node cbc.

DCAP=1

Specify DCAP=1 to select one of the following equations:

vbcx < FC · VJC

$$cbcx = CJCeff \cdot (1 - XCJC) \cdot \left(1 - \frac{vbcx}{VJC}\right)^{-MJC}$$

vbcx ≥ FC · VJC

$$cbcx = CJCeff \cdot (1 - XCJC) \cdot \frac{1 - FC \cdot (1 + MJC) + MJC \cdot \dfrac{vbcx}{VJC}}{(1 - FC)^{(1 + MJC)}}$$

DCAP=2

Specify DCAP=2 to select one of the following equations:

vbcx < 0

$$cbcx = CJCeff \cdot (1 - XCJC) \cdot \left(1 - \frac{vbcx}{VJC}\right)^{-MJC}$$

vbcx ≥ 0

$$cbcx = CJCeff \cdot (1 - XCJC) \cdot \left(1 + MJC \cdot \frac{vbcx}{VJC}\right)$$

where *vbcx* is the voltage between the external base node and the internal collector node.

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 vbs < 0

$$cbs \ = \ CJSeff \cdot \left(1 - \frac{vbs}{VJS} \right)^{-MJS}$$

Forward Bias vbs ≥ 0

$$cbs \ = \ CJSeff \cdot \left(1 + MJS \cdot \frac{vbs}{VJS} \right)$$

Using Substrate Capacitance Equation — Vertical

Substrate to Collector Diode

Reverse Bias vsc < 0

$$csc \ = \ CJSeff \cdot \left(1 - \frac{vsc}{VJS} \right)^{-MJS}$$

Forward Bias vsc ≥ 0

$$csc \ = \ CJSeff \cdot \left(1 + MJS \cdot \frac{vsc}{VJS} \right)$$

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:

$$excess\ phase \ = \ \left(2 \cdot \pi \cdot PTF \cdot \frac{TF}{360} \right) \cdot (2 \cdot \pi \cdot f)$$

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:

$$inrc = \left(\frac{4 \cdot k \cdot t}{RCeff}\right)^{1/2}$$

The mean square short-circuit emitter resistance noise current equation is:

$$inre = \left(\frac{4 \cdot k \cdot t}{REeff}\right)^{1/2}$$

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:

$$2 \cdot q \cdot fknee$$

where fknee is noise knee frequency (typically 100 Hz to 10 MHz) and q is electron charge.

$$inb^2 = (2 \cdot q \cdot ib) + \left(\frac{KF \cdot ib^{AF}}{f}\right)$$

$$inb^2 = shot\ noise^2 + flicker\ noise^2$$

$$shot\ noise = (2 \cdot q \cdot ib)^{1/2}$$

$$flicker\ noise = \left(\frac{KF \cdot ib^{AF}}{f}\right)^{1/2}$$

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:

TLEV = 0, 1 or 3

$$egnom = 1.16 - 7.02e{-}4 \cdot \frac{tnom^2}{tnom + 1108.0}$$

$$eg(t) = 1.16 - 7.02e{-}4 \cdot \frac{t^2}{t + 1108.0}$$

TLEV=2

$$egnom = EG - GAP1 \cdot \frac{tnom^2}{tnom + GAP2}$$

$$eg(t) = EG - GAP1 \cdot \frac{t^2}{t + GAP2}$$

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):

$$BF(t) = BF \cdot \left(\frac{t}{tnom}\right)^{XTB}$$

$$BR(t) = BR \cdot \left(\frac{t}{tnom}\right)^{XTB}$$

$$ISE(t) = \frac{ISE}{\left(\dfrac{t}{tnom}\right)^{XTB}} \cdot e^{\frac{facln}{NE}}$$

$$ISC(t) = \frac{ISC}{\left(\dfrac{t}{tnom}\right)^{XTB}} \cdot e^{\frac{facln}{NC}}$$

$$ISS(t) = \frac{ISS}{\left(\dfrac{t}{tnom}\right)^{XTB}} \cdot e^{\frac{facln}{NS}}$$

The parameter XTB usually should be set to zero for TLEV=2.

$$IS(t) = IS \cdot e^{facln}$$

$$IBE(t) = IBE \cdot e^{\frac{facln}{NF}}$$

$$IBC(t) = IBC \cdot e^{\frac{facln}{NR}}$$

TLEV=0, 1 or 3

$$facln = \frac{EG}{vt(tnom)} - \frac{EG}{vt(t)} + XTI \cdot \ln\left(\frac{t}{tnom}\right)$$

TLEV=2

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

Using Saturation and Temperature Equations, TLEV=1

The basic BJT temperature compensation equations for beta and the saturation currents when TLEV=1:

$$BF(t) = BF \cdot (1 + XTB \cdot \Delta t)$$

$$BR(t) = BR \cdot (1 + XTB \cdot \Delta t)$$

$$ISE(t) = \frac{ISE}{1 + XTB \cdot \Delta t} \cdot e^{\frac{facln}{NE}}$$

$$ISC(t) = \frac{ISC}{1 + XTB \cdot \Delta t} \cdot e^{\frac{facln}{NC}}$$

$$ISS(t) = \frac{ISS}{1 + XTB \cdot \Delta t} \cdot e^{\frac{facln}{NS}}$$

$$IS(t) = IS \cdot e^{facln}$$

$$IBE(t) = IBE \cdot e^{\frac{facln}{NF}}$$

$$IBC(t) = IBC \cdot e^{\frac{facln}{NR}}$$

where:

$$facln = \frac{EG}{vt(tnom)} - \frac{EG}{vt(t)} + XTI \cdot \ln\left(\frac{t}{tnom}\right)$$

TLEV=0, 1, 2

The parameters IKF, IKR, and IRB are also modified as:

$$IKF(t) = IKF \cdot (1 + TIKF1 \cdot \Delta t + TIKF2 \cdot \Delta t^2)$$

$$IKR(t) = IKR \cdot (1 + TIKR1 \cdot \Delta t + TIKR2 \cdot \Delta t^2)$$

$$IRB(t) = IRB \cdot (1 + TIRB1 \cdot \Delta t + TIRB2 \cdot \Delta t^2)$$

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

$$CJE(t) = CJE \cdot \left(1 - 0.5 \cdot dvjedt \cdot \frac{\Delta t}{VJE}\right)$$

$$CJC(t) \ = \ CJC \cdot \left(1 - 0.5 \cdot dvjcdt \cdot \frac{\Delta t}{VJC} \right)$$

$$CJS(t) \ = \ CJS \cdot \left(1 - 0.5 \cdot dvjsdt \cdot \frac{\Delta t}{VJS} \right)$$

$$VJE(t) \ = \ VJE + dvjedt \cdot \Delta t$$

$$VJC(t) \ = \ VJC + dvjcdt \cdot \Delta t$$

$$VJS(t) \ = \ VJS + dvjsdt \cdot \Delta t$$

where TLEV= 0, 1, or 3

$$dvjedt \ = \ -\frac{egnom + 3 \cdot vt(tnom) + (1.16 - egnom) \cdot \left(2 - \frac{tnom}{tnom + 1108} \right) - VJE}{tnom}$$

$$dvjcdt \ = \ -\frac{egnom + 3 \cdot vt(tnom) + (1.16 - egnom) \cdot \left(2 - \frac{tnom}{tnom + 1108} \right) - VJC}{tnom}$$

$$dvjsdt \ = \ -\frac{egnom + 3 \cdot vt(tnom) + (1.16 - egnom) \cdot \left(2 - \frac{tnom}{tnom + 1108} \right) - VJS}{tnom}$$

and TLEV=2

$$dvjedt \ = \ -\frac{egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left(2 - \frac{tnom}{tnom + GAP2} \right) - VJE}{tnom}$$

$$dvjcdt \ = \ -\frac{egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left(2 - \frac{tnom}{tnom + GAP2} \right) - VJC}{tnom}$$

$$dvjsdt \ = \ -\frac{egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left(2 - \frac{tnom}{tnom + GAP2} \right) - VJS}{tnom}$$

Using Parasitic Resistor Temperature Equations

The parasitic resistors, as a function of temperature regardless of TLEV value, are determined as:

$$RE(t) \ = \ RE \cdot (1 + TRE1 \cdot \Delta t + TRE2 \cdot \Delta t^2)$$

$$RB(t) \ = \ RB \cdot (1 + TRB1 \cdot \Delta t + TRB2 \cdot \Delta t^2)$$

$$RBM(t) \ = \ RBM \cdot (1 + TRM1 \cdot \Delta t + TRM2 \cdot \Delta t^2)$$

$$RC(t) \ = \ RC \cdot (1 + TRC1 \cdot \Delta t + TRC2 \cdot \Delta t^2)$$

Using BJT LEVEL=2 Temperature Equations

The model parameters of BJT LEVEL 2 model are modified for temperature compensation as:

$$GAMMA(t) \ = \ GAMMA \cdot e^{(facln)}$$

$$RC(t) \ = \ RC \cdot \left(\frac{t}{tnom}\right)^{BEX}$$

$$VO(t) \ = \ VO \cdot \left(\frac{t}{tnom}\right)^{BEXV}$$

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 fT 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 *Iepi* and charge storage elements *Ci* and *Cx* model the quasi-saturation effects. The parasitic substrate bipolar transistor is also included in the vertical transistor by the diode D and current source *Ibs*.

Figure 16-11: Vertical npn Bipolar Transistor (SUBS=+1)

Figure 16-12: Lateral npn Bipolar Transistor (SUBS=-1)

Using Epitaxial Current Source Iepi

The epitaxial current value, Iepi, is determined by the equation:

$$Iepi = \frac{ki - kx - \ln\left(\frac{1 + ki}{1 + kx}\right) + \frac{vbc - vbcx}{NEPI \cdot vt}}{\left(\frac{RCeff}{NEPI \cdot vt}\right) \cdot \left(1 + \frac{|vbc - vbcx|}{VO}\right)}$$

where:

$$ki = [1 + GAMMA \cdot e^{vbc/(NEPI \cdot vt)}]^{1/2}$$

$$kx = [1 + GAMMA \cdot e^{vbcx/(NEPI \cdot vt)}]^{1/2}$$

In special cases when the model parameter GAMMA is set to zero, ki and kx become one and,

$$Iepi = \frac{vbc - vbcx}{RCeff \cdot \left(1 + \dfrac{|vbc - vbcx|}{VO}\right)}$$

Using Epitaxial Charge Storage Elements Ci and Cx

The epitaxial charges are determined by:

$$qi = QCOeff \cdot \left(ki - 1 - \frac{GAMMA}{2}\right)$$

and:

$$qx = QCOeff \cdot \left(kx - 1 - \frac{GAMMA}{2}\right)$$

The corresponding capacitances are calculated as:

$$Ci = \frac{\partial}{\partial vbc}(qi) = \left(\frac{GAMMA \cdot QCOeff}{2 \cdot NEPI \cdot vt \cdot kx}\right) \cdot e^{vbc/(NEPI \cdot vt)}$$

and:

$$Cx = \frac{\partial}{\partial vbcx}(qx) = \left(\frac{GAMMA \cdot QCOeff}{2 \cdot NEPI \cdot vt \cdot kx}\right) \cdot e^{vbcx/(NEPI \cdot vt)}$$

In the special case where GAMMA=0 the Ci and 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 vce=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  PE=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
```

```
+ 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
*

.MODEL SBJT NPN
+ IBE='IES*SF*SIES' IBC='ICS*SF*SICS'
+ CJE='CJE*SF*SCBE' CJC='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 SIC=1M 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.

---

1. C. McAndrew, J. Seitchik, D. Bowers, M. Dunn, M. Foisy, I. Getreu, M. McSwain, S. Moinian, J. Parker, D. Roulston, M. Schroter, P. van Wijnen, and L. Wagner, *"VBIC95: The vertical bipolar intercompany model," IEEE Journal of Solid State Circuits,* vol.31, p.1476-1483, 1996.

Figure 16-14: Transient Analysis

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 nc nb ne <ns> <nT> mname <regular parameters> <tnodeout>

where nT is the node for temperature. If this node is given, but ns is not given, the flag "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
```

```
    Vsub 104 0 0.
    Vemi 103 0 0.
    R1   101 201 10
    R2   102 202 10
    R4   104 204 10
    R3   103 203 10
    Q1 202 201 203 204 VBIC_EXAMPLE
    .model VBIC_EXAMPLE npn LEVEL=4
    + afn=1 ajc=-0.5 aje=0.5 ajs=0.5
    + avc1=0 avc2=0 bfn=1 cbco=0 cbeo=0 cjc=2e-14
    + cjcp=4e-13 cje=le-13 cjep=le-13 cth=0
    + ea=1.12 eaic=1.12 eaie=1.12 eais=1.12 eanc=1.12
    + eane=1.12 eans=1.12 fc=0.9 gamm=2e-11 hrcf=2
    + ibci=2e-17 ibcip=0 ibcn=5e-15 ibcnp=0
    + ibei=1e-18 ibeip=0 iben=5e-15 ibenp=0
    + ikf=2e-3 ikp=2e-4 ikr=2e-4 is=le-16 isp=le-15 itf=8e-2
    + kfn=0 mc=0.33 me=0.33 ms=0.33
    + nci=1 ncip=1 ncn=2 ncnp=2 nei=1 nen=2
    + nf=1 nfp=1 nr=1 pc=0.75 pe=0.75 ps=0.75 qco=le-12 qtf=0
    + rbi=4 rbp=4 rbx=1 rci=6 rcx=1 re=0.2 rs=2
    + rth=300 tavc=0 td=2e-11 tf=10e-12 tnf=0 tr=100e-12
    + tnom=25 tref=25 vef=10 ver=4 vo=2
    + vtf=0 wbe=1 wsp=1
    + xii=3 xin=3 xis=3 xrb=0 xrc=0 xre=0 xrs=0 xtf=20 xvo=0
    *.END
```

 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=1.000e+00 NFP=1.00000e+00 IBEIP=0
+ IBENP=0 IBCIP=0 NCIP=1.00000e+00
```

```
+ 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

Name (Alias)	Unit	Default	Description
AFN		1	Flicker noise exponent for current
AJC		-0.5	Base-collector capacitance switching parameter
AJE		-0.5	Base-emitter capacitance switching parameter
AJS		-0.5	Substrate-collector capacitance switching parameter
AVC1	V^{-1}	0	Base-collector weak avalanche parameter 1
AVC2	V^{-1}	0	Base-collector weak avalanche parameter 2
BFN		1	Flicker noise exponent for 1/f dependence
CBCO (CBC0)	F	0	Extrinsic base-collector overlap capacitance
CBEO (CBE0)	F	0	Extrinsic base-emitter overlap capacitance
CJC	F	0	Base-collector intrinsic zero bias capacitance
CJCP	F	0	Substrate-collector zero bias capacitance
CJE	F	0	Base-emitter zero bias capacitance
CJEP	F	0	Base-collector extrinsic zero bias capacitance
CTH	J/K	0	Thermal capacitance
EA	eV	1.12	Activation energy for IS
EAIC	eV	1.12	Activation energy for IBCI/IBEIP
EAIE	eV	1.12	Activation energy for IBEI
EAIS	eV	1.12	Activation energy for IBCIP
EANC	eV	1.12	Activation energy for IBCN/IBENP
EANE	eV	1.12	Activation energy for IBEN
EANS	eV	1.12	Activation energy for IBCNP
FC		0.9	Forward bias depletion capacitance limit

Table 16-5: Default Model Parameters for BJT, LEVEL 4

Name (Alias)	Unit	Default	Description
GAMM		0	Epi doping parameter
HRCF		1	High-current RC factor
IBCI	A	1e-16	Ideal base-collector saturation current
IBCIP	A	0	Ideal parasitic base-collector saturation current
IBCN	A	1e-15	Non-ideal base-collector saturation current
IBCNP	A	0	Non-ideal parasitic base-collector saturation current
IBEI	A	1e-18	Ideal base-emitter saturation current
IBEIP	A	0	Ideal parasitic base-emitter saturation current
IBEN	A	1e-15	Non-ideal base-emitter saturation current
IBENP	A	0	Non-ideal parasitic base-emitter saturation current
IKF	A	2e-3	Forward knee current
IKP	A	2e-4	Parasitic knee current
IKR	A	2e-4	Reverse knee current
IS	A	1e-16	Transport saturation current
ISP	A	1e-16	Parasitic transport saturation current
ITF	A	1e-3	Coefficient of TF dependence in Ic
KFN		0	Base-emitter flicker noise constant
MC		0.33	Base-collector grading coefficient
ME		0.33	Base-emitter grading coefficient
MS		0.33	Substrate-collector grading coefficient
NCI		1	Ideal base-collector emission coefficient
NCIP		1	Ideal parasitic base-collector emission coefficient
NCN		2	Non-ideal base-collector emission coefficient

Table 16-5: Default Model Parameters for BJT, LEVEL 4

Name (Alias)	Unit	Default	Description
NCNP		2	Non-ideal parasitic base-collector emission coefficient
NEI		1	Ideal base-emitter emission coefficient
NEN		2	Non-ideal base-emitter emission coefficient
NF		1	Forward emission coefficient
NFP		1	Parasitic forward emission coefficient
NR		1	Reverse emission coefficient
PC	V	0.75	Base-collector built-in potential
PE	V	0.75	Base-emitter built-in potential
PS	V	0.75	Substrate-collector built-in potential
QCO (QC0)	C	0	Epi charge parameter
QTF		0	Variation of TF with base-width modulation
RBI	Ohm	1e-1	Intrinsic base resistance
RBP	Ohm	1e-1	Parasitic base resistance
RBX	Ohm	1e-1	Extrinsic base resistance
RCI	Ohm	1e-1	Intrinsic collector resistance
RCX	Ohm	1e-1	Extrinsic collector resistance
RE	Ohm	1e-1	Emitter resistance
RS	Ohm	1e-1	Substrate resistance
RTH	K/W	0	Thermal resistance
TAVC	1/K	0	Temperature coefficient of AVC2
TD	s	0	Forward excess-phase delay time
TF	s	1e-11	Forward transit time

Table 16-5: Default Model Parameters for BJT, LEVEL 4

Name (Alias)	Unit	Default	Description
TNF	1/K	0	Temperature coefficient of NF
TR	s	1e-11	Reverse transit time
TREF (TNOM)	°C	27	Nominal measurement temperature of parameters (please do not use TNOM alias, though it is allowed)
VEF	V	0	Forward Early voltage
VER	V	0	Reverse Early voltage
VO (V0)	V	0	Epi drift saturation voltage
VTF	V	0	Coefficient of TF dependence on Vbc
WBE		1	Portion of IBEI from Vbei, 1-WBE from Vbex
WSP		1	Portion of ICCP from Vbep, 1-WSP from Vbci
XII		3	Temperature exponent of IBEI/IBCI/IBEIP/IBCIP
XIN		3	Temperature exponent of IBEN/IBCN/IBENP/ IBCNP
XIS		3	Temperature exponent of IS
XRB		1	Temperature exponent of base resistance
XRC		1	Temperature exponent of collector resistance
XRE		1	Temperature exponent of emitter resistance
XRS		1	Temperature exponent of substrate resistance
XTF		0	Coefficient of TF bias dependence
XVO (XV0)		0	Temperature exponent of VO

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 $1.0e^{-3}$. Values smaller than this will be reassigned a value of $1.0e^{-3.}$

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 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

Parameter	Unit	Default	Description
LEVEL	-	-	LEVEL 6 for MEXTRAM
EXAVL	-	0	Flag for extended modeling of avalanche currents
EXMOD	-	0	Flag for extended modeling of the reverse current gain
EXPHI	-	1	Flag for distributed high frequency effects

Basic Parameters

Parameter	Unit	Default	Description
TREF	^{o}C	0.0	Model nominal temperature
IS	A	5.E-17	Collector-emitter saturation current
BF	A	140.0	Ideal forward current gain
XIBI	-	0.0	Fraction of ideal base current that belongs to the sidewall
IBF	A	2.0E-14	Saturation current of the non-ideal forward base current
VLF	V	0.5	Cross-over voltage of the non-ideal forward base current
IK	A	15.E-3	High-injection knee current
BRI	-	16.0	Ideal reverse current gain
IBR	A	8.0e-15	Saturation current of the non-ideal reverse base current
VLR	V	0.5	Cross-over voltage of the non-ideal reverse base current

Parameter	Unit	Default	Description
XEXT	-	0.5	Part of I EX ,Q EX ,Q TEX and I SUB that depends on the base-collector voltage VBC1
QBO	C	1.2e-12	Base charge at zero bias
ETA	-	4.0	Factor of the built-in field of the base
AVL	-	50.	Weak avalanche parameter
EFI	-	0.7	Electric field intercept (with EXAVL=1)
IHC	A	3.e-3	Critical current for hot carriers
RCC	ohm	25.	Constant part of the collector resistance
RCV	ohm	750.	Resistance of the unmodulated epilayer
SCRCV	ohm	1000.0	Space charge resistance of the epilayer
SFH	-	0.6	Current spreading factor epilayer
RBC	ohm	50.	Constant part of the base resistance
RBV	ohm	100.	Variable part of the base resistance at zero bias
RE	ohm	2.0	Emitter series resistance
TAUNE	s	3.e-10	Minimum delay time of neutral and emitter charge
MTAU	-	1.18	Non-ideality factor of the neutral and emitter charge
CJE	F	2.5e-13	Zero bias collector-base depletion capacitance
VDE	V	0.9	Emitter-base diffusion voltage
PE	-	0.33	Emitter-base grading coefficient
XCJE	F	0.5	Fraction of the emitter-base depletion capacitance that belongs to the sidewall

Parameter	Unit	Default	Description
CJC	F	1.3e-13	Zero bias collector-base depletion capacitance
VDC	V	0.6	Collector-base diffusion voltage
PC	-	0.4	Collector-base grading coefficient variable part
XP	F	0.2	Constant part of CJC
MC	-	0.5	Collector current modulation coefficient
XCJC	-	0.1	Fraction of the collector-base depletion capacitance under the emitter area
VGE	V	1.01	Band-gap voltage of the emitter
VGB	V	1.18	Band-gap voltage of the base
VGC	V	1.205	Band-gap voltage of the collector
VGJ	V	1.1	Band-gap voltage recombination emitter-base junction
VI	V	0.040	Ionization voltage base dope
NA	cm^-3	3.0E17	Maximum base dope concentration
ER	-	2.E-3	Temperature coefficient of VLF and VLR
AB	-	1.35	Temperature coefficient resistivity of the base
AEPI	-	2.15	Temperature coefficient resistivity of the epilayer
AEX	-	1.	Temperature coefficient resistivity of the extrinsic base
AC	-	0.4	Temperature coefficient resistivity of the buried layer
KF	-	2.E-16	Flicker noise coefficient ideal base current

Parameter	Unit	Default	Description
KFN	-	2.E-16	Flicker noise coefficient non-ideal base current
AF	-	1.0	Flicker noise exponent
ISS	A	6.E-16	Base-substrate saturation current
IKS	A	5.E-6	Knee current of the substrate
CJS	F	1.e-12	Zero bias collector-substrate depletion capacitance
VDS	V	0.5	Collector-substrate diffusion voltage
PS	-	0.33	Collector-substrate grading coefficient
VGS	V	1.15	Band-gap voltage of the substrate
AS	-	2.15	For a closed buried layer: AS=AC For an open buried layer: AS=AEPI

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 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 16-85.

Note: Self-heating is not supported in the 2000.4 Star-Hspice release.

Model Implementation

Model Parameters

Parameter	Unit	Default	Description
LEVEL		9	HiCUM BJT level in Hspice
TREF	C	26.85	Temperature in simulation

Internal Transistors

Transfer Current Parameters

Parameter	Unit	Default	Factor	Description
C10	A^2s	3.76e-32	M^2	Constant(IS*QP0)
Qp0	As	2.78e-14		Zero-bias hole charge
ICH	A	2.09e-0Z		High-current correction for 2D/3D

Parameter	Unit	Default	Factor	Description
HFC	-	1.0		Weighting factor for Qfc (mainly for HBTs)
HFE	-	1.0		Weighting factor for Qef in HBTs
HJCI	-	1.0		Weighting factor for Qjci in HBTs
HJEI	-	0.0		Weighting factor for Qjei in HBTs
ALIT	-	0.45		Factor for additional delay time of iT

BE Depletion Capacitance Parameters

Parameter	Unit	Default	Factor	Description
VDEI	V	0.95		Built-in voltage
CJEI0	F	8.11e-15		Zero-bias value
ZEI	-	0.5		Exponent coefficient
ALJEI	-	1.8		Ratio of max. to zero-bias value

BC Depletion Capacitance Parameters

Parameter	Unit	Default	Factor	Description
CJCI0	F	1.16e-15	M^2	Zero-bias value
VDCI	V	0.8		Built-in voltage
ZCI	-	0.333		Exponent coefficient
VPTCI	V	416		Punch-through voltage (=q Nci w^2ci /(2epsilion))

Forward Transit Time Parameters

Parameter	Unit	Default	Factor	Description
T0	s	4.75e-12		Low current transit time at $V_{B'C'}=0$
DT0H	s	2.1e-12		Time constant for base and BC SCR width modulation
TBVL	s	40e-12		Voltage for modeling carrier jam at low $V_{C'E'}$
TEF0	s	1.8e-12		Storage time in neutral emitter
GTFE	-	1.4		Exponent factor for current dep. emitter transit time
THCS	s	3.0e-11		Saturation time constant at high current densities
ALHC	-	0.75		Smoothing factor for current dep. C and B transit time
FTHC	-	0.6		Partitioning factor for base and collection portion
ALQF	-	0.225		Factor for additional delay time of Q_f

Critical Current Parameters

Parameter	Unit	Default	Factor	Description
RCI0	Ohm	127.8	1/M	Low-field resistance of internal collector region
VLIM	V	0.7		Voltage separating ohmic and SCR regime
VPT	V	5.0		Epi punch-through vtg. of BC SCR
VCES	V	0.1		Internal CE sat. vtg.

Inverse Transit Time Parameter

Parameter	Unit	Default	Factor	Description
TR	s	1.0e-9		Time constant for inverse operation

Base Current Components Parameters

Parameter	Unit	Default	Factor	Description
IBEIS	A	1.16e-20	M	BE saturation current
MBEI	-	1.015		BE saturation current
IREIS	A	1.16e-6	M	BE recombination saturation current
MREI	-	2.0		BE recombination non-ideality factor
IBCIS	A	1.16e-20	M	BC saturation current
MBCI	-	1.015		BC non-ideality factor

Weak BC Avalanche Breakdown Parameters

Parameter	Unit	Default	Factor	Description
FAVL	1/V	1.186		Prefactor for CB avalanche effect
QAVL	As	1.11e-14	M	Exponent factor for CB avalanche effect

Internal Base Resistance Parameters

Parameter	Unit	Default	Factor	Description
RBI0	Ohm	0	1/M	Value at zero-bias
FDQR0	-	0.0		Correction factor for modulation by BE abd BC SCR
FGEO	-	0.73		Geometry factor (value corresponding to long emitter stripe)

Parameter	Unit	Default	Factor	Description
FQI	-	0.9055		Ratio of internal to total minority charge
FCRBI	-	0.0		Ratio of h.f. shunt to total internal capacitance.

Lateral Scaling

Parameter	Unit	Default	Factor	Description
LATB	-	3.765		Scaling factor for Qfc in 1_E
LATL	-	0.342		Scaling factor for Qfc in l_E direction

Peripheral Elements

BE Depletion Capacitance

Parameter	Unit	Default	Factor	Description
CJEP0	F	2.07e-15	M	Zero-bias value
VDEP	V	1.05		Built-in voltage
ZEP	-	0.4		Depletion coeff
ALJEP	-	2.4		Ratio of max. to zero-bias value

Base Current

Parameter	Unit	Default	Factor	Description
IBEPS	A	3.72e-21	M	Saturation current
MBEP	-	1.015		Non-ideality factor
IREPS	A	1e-30	M	Recombination saturation factor
MREP	-	2.0		Recombination non-ideality factor

BE Tunneling

Parameter	Unit	Default	Factor	Description
IBETS	A	0	M	Saturation current
ABET	-	0.0		Exponent coefficient

External Elements

BC Capacitance

Parameter	Unit	Default	Factor	Description
CJCX0	F	5.393e-15	M	Zero-bias depletion value
VDCX	V	0.7		Built-in voltage
ZCX	-	0.333		Exponent coefficient
VPTCX	V	100		Punch-through voltage
CCOX	F	2.97e-15	M	Collector oxide capacitance
FBC	-	0.1526		Partitioning factor for C_BCX =C'_BCx+C"_BCx

BC Base Current Component

Parameter	Unit	Default	Factor	Description
IBCXS	A	4.39e-20	M	Saturation current
MBCX	-	1.03		Non-ideality factor

Other External Elements

Parameter	Unit	Default	Factor	Description
CEOX	F	1.13e-15	M	Emitter-base isolation overlap cap
RBX	Ohm	0	1/M	External base series resistance
RE	Ohm	0	1/M	Emitter series resistance
RCX	Ohm	0	1/M	External collector series resistance

Substrate Transistor Parameters

Parameter	Unit	Default	Factor	Description
ITSS	A	0.0	M	Transfer saturation current
MSF	-	0.0		Non-ideality factor (forward transfer current)
TSF	-	0.0		Minority charge storage transit time
ISCS	A	0.0	M	Saturation current of CS diode
MSC	-	0.0		Non-ideality factor of CS diode

Collector-Substrate Depletion Capacitance

Parameter	Unit	Default	Factor	Description
CJS0	F	3.64e-14	M	Zero-bias value of CS depletion cap
VDS	V	0.6		Built-in voltage
ZS	-	0.447		Exponent coefficient
VPTS	V	1000		Punch-through voltage

Substrate Coupling Network

Parameter	Unit	Default	Factor	Description
RSU	Ohm	0	1/M	Substrate series resistance
CSU	F	0		Substrate capacitance from permittivity of bulk material

Noise Parameters

Parameter	Unit	Default	Factor	Description
KF	-	1.43e-8		Flicker noise factor (no unit only for AF=2!)
AF	-	2.0		Flicker noise exponent factor
KRBI	-	1.17		Factor for internal base resistance

Temperature Dependence Parameters

Parameter	Unit	Default	Factor	Description
VGB	V	1.17		Bandgap-voltage
ALB	1/K	6.3e-3		Relative temperature coefficient of forward current gain
ALT0	1/K	0		First-order relative temperature coefficient of TEF0
KT0	1/K	0		Second-order relative temperature coefficient of TEF0
ZETACI	-	1.6		Temperature exponent factor RCI0
ALVS	1/K	1e-3		Relative temperature coefficient of saturation drift velocity
ALCES	1/K	0.4e-3		Relative temperature coefficient of VCES
ZETARBI	-	0.588		Temperature exponent factor of RBI0
ZETARBX	-	0.2060		Temperature exponent factor of RBX
ZETARCX	-	0.2230		Temperature exponent factor of RCX
ZETARE	-	0		Temperature exponent factor of RE
ALFAV	1/K	8.25e-5		Relative temperature coefficient for avalanche breakdown
ALQAV	1/K	1.96e-4		Relative temperature coefficient for avalanche breakdown

Self-Heating Parameters

Parameter	Unit	Default	Factor	Description
RTH	K/W	0	1/M	Thermal resistance (not supported)
CTH	Ws/K	0	M	Thermal resistance (not supported)

Other Parameters

Parameter	Unit	Default	Factor	Description
FBCS	-	-1.0		Determine external BC capacitance partitioning
IS	-1.0	A		Ideal saturation current
KRBI	-	1.0		Noise analysis of internal resistance
MCF	-	1.0		Non-ideality factor of reverse current between base and collector. VT=VT*MCF
MSR	-	1.0		Non-ideality factor of reverse current in substrate transistor. VT=VT*MSR
ZETACX	-	1.0		Temperature exponent factor (epi-layer)

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 Celsisu. (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)   **(b)**

(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

```
.DATA test_data vbe vce vsub
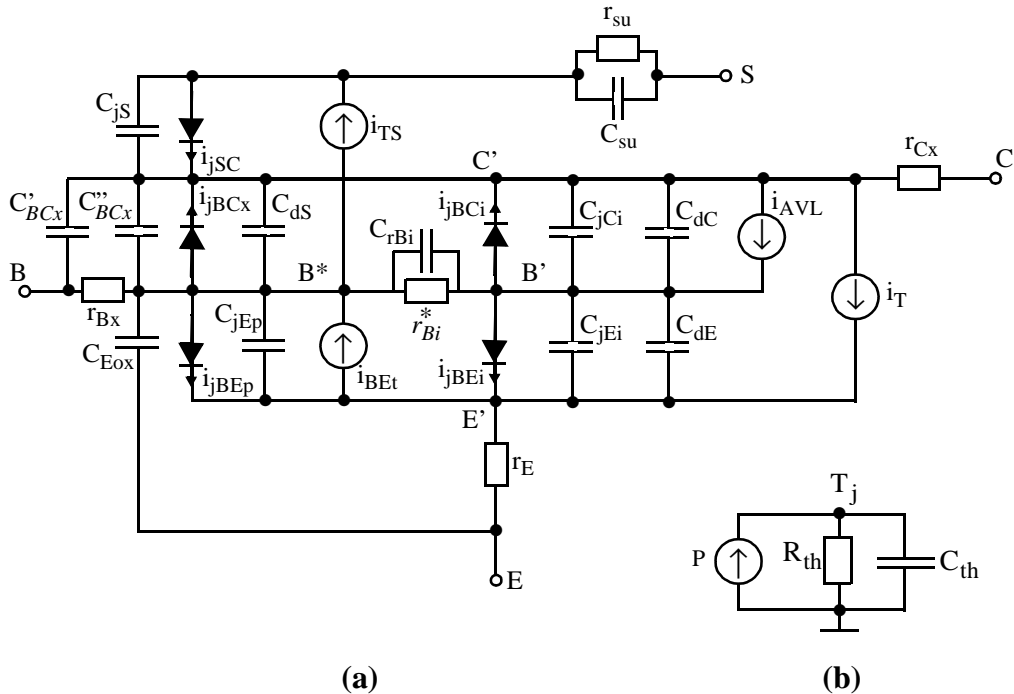
0.0   0.0   0.0
0.1   0.0   0.0
0.2   0.0   0.0
0.3   0.0   0.0
0.4   0.0   0.0
0.5   0.0   0.0
0.6   0.0   0.0
```

```
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=.000000E+00
+          hjci=.100000E+01 tr=1.00000000E-9
+          cjei0=.81100E-14 vdei=.950000E+00 zei=.5000000E+00
+          aljei=.18000E+01
+          cjci0=.11600E-14 vdci=.800000E+00 zci=.3330000E+00
+          vptci=.41600E+03
+          rci0=.127800E+03 vlim=.700000E+00 vpt=.5000000E+01
+          vces=.100000E+00
+          t0=.47500000E-11 dt0h=.210000E-11 tbvl=.400000E-11
+          tef0=.180000E-11 gtfe=.140000E+01 thcs=.300000E-10
+          alhc=.750000E+00
+          fthc=.600000E+00
+          latb=.376500E+01 latl=.342000E+00 fqi=.9055000E+00
+          alit=.450000E+00 alqf=.225000E+00
+          favl=.118600E+01 qavl=.111000E-13 alfav=.82500E-04
```

```
+        alqav=.19600E-03
+        ibeis=.11600E-19 mbei=.101500E+01 ibeps=.10000E-29
+        mbep=.200000E+01
+        ireis=.11600E-15 mrei=.200000E+01 ireps=.10000E-29
+        mrep=.200000E+01
+        rbi0=.000000E+00 fdqr0=.00000E+00 fgeo=.730000E+00
+        fcrbi=.00000E+00
+        cjep0=.00000E+00 vdep=.105000E+01 zep=.4000000E+00
+        aljep=.24000E+01
+        ceox=.000000E+00
+        cjcx0=.00000E+00 vdcx=.700000E+00 zcx=.3330000E+00
+        vptcx=.10000E+03
+        ccox=.000000E+00 fbc=.1526000E+00
+        ibcxs=.10000E-29 mbcx=.200000E+01 ibcis=.11600E-19
+        mbci=.101500E+01
+        cjs0=.000000E+00 vds=.6000000E+00 zs=.44700000E+00
+        vpts=.100000E+04
+        rcx=.0000000E+00 rbx=.0000000E+00 re=.00000000E+00
+        kf=.00000000E+00 af=.00000000E+00
+        vgb=.1170000E+01 alb=.6300000E-02 alt0=.000000E+00
+        kt0=.0000000E+00
+        zetaci=.1600E+01 alvs=.100000E-02 alces=.40000E-03
+        zetarbi=0.5880E+00                 zetarcx=0.2230E+00
+        zetarbx=0.2060E+00                 zetare=0.0000E+00
+        rth=0.0 cth=0.0
+        ibets=.00000E+00 abet=.000000E+00
+        itss=.000000E+00 msf=.0000000E+00 tsf=0.000000E+00
+        iscs=.000000E+00
+        msc=.0000000E+00
+        rsu=.0000000E+00 csu=.0000000E+00


.END
```

Avant!

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.

LEVEL=1	SPICE model
LEVEL=2	Modified SPICE model, gate modulation of LAMBDA
LEVEL=3	Hyperbolic tangent MESFET model (Curtice, Statz, Meta, TriQuint Models)
SAT=0	Curtice model (Default)
SAT=1	Curtice model with user defined VGST exponent
SAT=2	Cubic approximation of Curtice model with gate field degradation (Statz model)
SAT=3	Avant! variable saturation model

The model parameter CAPOP selects the type of capacitor model:

CAPOP=0	SPICE depletion capacitor model
CAPOP=1	Charge conserving, symmetric capacitor model (Statz)
CAPOP=2	Avant! improvements to CAPOP=1

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:

ACM=0	SPICE method (default)
ACM=1	Physically based method

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:

DCAP	Capacitance equation selector
GMIN, GRAMP, GMINDC	Conductance options
SCALM	Model scaling option
DCCAP	Invokes capacitance calculation in DC analysis

Table 17-1: JFET Options

Function	**Control Options**
capacitance	DCAP, DCCAP
conductance	GMIN, GMINDC, GRAMP
scaling	SCALM

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.

Figure 17-1: JFET Current Convention, N-Channel

For a p-channel device, the following must be reversed:

■ Polarities 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

$$gds = \frac{\partial(ids)}{\partial(vds)}\bigg|_{vgs = const.}$$

The ids 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:

$$igd = ISeff \cdot \left(e^{\frac{vgd}{N \cdot vt}} - 1 \right) \qquad vgd > -10 \cdot N \cdot vt$$

$$igd = -ISeff \qquad\qquad vgd \le -10 \cdot N \cdot vt$$

$$igs = ISeff \cdot \left(e^{\frac{vgs}{N \cdot vt}} - 1 \right) \qquad vgs > -10 \cdot N \cdot vt$$

$$igs = -ISeff \qquad\qquad vgs \le -10 \cdot N \cdot vt$$

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

Variable/ Quantity	Definitions
cgd	Gate to drain capacitance
cgs	Gate to source capacitance
ggd	Gate to drain AC conductance
ggs	Gate to source AC conductance
gds	Drain to source AC conductance controlled by vds
gm	Drain to source AC transconductance controlled by vgs
igd	Gate to drain current
igs	Gate to source current
ids	DC drain to source current
ind	Equivalent noise current drain to source
inrd	Equivalent noise current drain resistor
inrs	Equivalent noise current source resistor
rd	Drain resistance
rs	Source resistance
vgd	Internal gate-drain voltage
vgs	Internal gate-source voltage
f	Frequency
ε_o	Vacuum permittivity = 8.854e-12 F/m
k	1.38062e-23 (Boltzmann's constant)
q	1.60212e-19 (electron charge)
t	Temperature in $^\circ$K
Dt	t - tnom

Table 17-2: Equation Variable Names and Constants

Variable/ Quantity	Definitions
tnom	Nominal temperature of parameter measurements in $^\circ$K (user-input in $^\circ$C). Tnom = 273.15 + TNOM
vt(t)	$k \cdot t/q$
vt(tnom)	$k \cdot tnom/q$

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

Model Parameters Common to All Levels	
Geometric	ACM, ALIGN, AREA, HDIF, L, LDEL, LDIF, RD, RG, RS, RSH, RSHG, RSHL, W, WDEL
Capacitance	CAPOP, CGD, CGS, FC, M, PB, TT
Subthreshold	ND, NG
Noise	AF, KF
LEVEL=1 Model Parameters (JFET)	
DC	BETA, IS, LAMBDA, N, VTO
LEVEL=2 Model Parameters (JFET)	
DC	BETA, IS, LAMBDA, LAM1, N, VTO
LEVEL=3 Model Parameters (MESFET)	
DC	ALPHA, BETA, D, GAMDS, IS, N, K1, LAMBDA, NCHAN, SAT, SATEXP, UCRIT, VBI, VGEXP, VP, VTO

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

Name (Alias)	Units	Default	Description
ACM			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
ALIGN	m	0	Misalignment of gate
AREA			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.
HDIF	m	0	Distance of the heavily diffused or low resistance region from source or drain contact to lightly doped region
IS	amp	1.0e-14	Gate junction saturation current ISeff = IS $\cdot$ AREAeff
L	m	0.0	Default length of FET. Override this parameter using the element L. Leff = L $\cdot$ SCALM + LDELeff
LDEL	m	0.0	Difference between drawn and actual or optical device length LDELeff = LDEL $\cdot$ SCALM
LDIF	m	0	Distance of the lightly doped region from heavily doped region to transistor edge

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) RDeff = RD /AREAeff, ACM=0
RG	ohm	0.0	Gate resistance (see the ACM section) RGeff = RG · AREAeff, ACM=0
RS	ohm	0.0	Source ohmic resistance (see the ACM section) RSeff = RS /AREAeff, 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. Weff = W · SCALM + WDELeff
WDEL	m	0.0	Difference between drawn and actual or optical device width WDELeff = WDEL · SCALM

Gate Capacitance LEVEL 1, 2, and 3 Parameters

Name (Alias)	Units	Default	Description
CAPOP		0.0	Capacitor model selector: CAPOP=0 – default capacitance equation based on diode depletion layer CAPOP=1 – symmetric capacitance equations (Statz) CAPOP=2 – Avant! improvement to CAPOP=1
CALPHA	ALPHA		Saturation factor for capacitance model (CAPOP=2 only)
CAPDS	F	0	Drain to source capacitance for TriQuint model CAPDSeff=CAPDS $\cdot \dfrac{\text{Weff}}{\text{Leff}} \cdot \text{M}$
CGAMDS	GAMDS		Threshold lowering factor for capacitance (CAPOP=2 only)
CGD	F	0.0	Zero-bias gate-drain junction capacitance CGDeff = CGD · AREAeff Override this parameter by specifying GCAP.
CGS	F	0.0	Zero-bias gate-source junction capacitance CGSeff = CGS · AREAeff Override this parameter by specifying GCAP
CRAT		0.666	Source fraction of gate capacitance (used with GCAP)
GCAP	F		Zero-bias gate capacitance. If specified, CGSeff = GCAP · CRAT · AREAeff and CGDeff = GCAP · (1-CRAT) · AREAeff

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)		0.50	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

Name (Alias)	Units	Default	Description
LEVEL		1.0	LEVEL=1 invokes SPICE JFET model
BETA	amp/V^2	1.0e-4	Transconductance parameter, gain $$BETAeff = BETA \cdot \frac{Weff \cdot M}{Leff}$$
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

Name (Alias)	Units	Default	Description
LEVEL		1.0	Level of FET DC model. LEVEL=2 is based on modifications to the SPICE model for gate modulation of LAMBDA.
BETA	amp /V^2	1.0e-4	Transconductance parameter, gain $$BETAeff = BETA \cdot \frac{Weff \cdot M}{Leff}$$
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)

Name (Alias)	Units	Default	Description
VTO	V	-2.0	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.

DC Model LEVEL 3 Parameters

Name (Alias)	Units	Default	Description
LEVEL		1.0	Level of FET DC model. LEVEL=3 is the Curtice MESFET model.
A	m	0.5µ	Active layer thickness Aeff = A · SCALM
ALPHA	1/V	2.0	Saturation factor
BETA	amp /V^2	1.0e-4	Transconductance parameter, gain $BETAeff = BETA \cdot \dfrac{Weff \cdot M}{Leff}$
D		11.7	Semiconductor dielectric constant: Si=11.7, GaAs=10.9
DELTA		0	Ids feedback parameter of TriQuint model
GAMDS (*GAMMA*)		0	Drain voltage, induced threshold voltage lowering coefficient
LAMBDA	1/V	0.0	Channel length modulation parameter
K1	V$^{1/2}$	0.0	Threshold voltage sensitivity to bulk node
NCHAN	atom/ cm^3	1.552e16	Effective dopant concentration in the channel
ND	1/V	0.0	Drain subthreshold factor
NG		0.0	Gate subthreshold factor (typical value=1)

Name (Alias)	Units	Default	Description
SAT		0.0	Saturation factor SAT=0 (standard Curtice model) SAT= (Curtice model with hyperbolic tangent coefficient) SAT=2 (cubic approximation of Curtice model (Statz))
SATEXP		3	Drain voltage exponent
UCRIT	V/cm	0	Critical field for mobility degradation
VBI		1.0	Gate diode built-in voltage
VGEXP (Q)		2.0	Gate voltage exponent
VP			Dinch-off voltage (default is calculated)
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.

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

$$AREAeff = \frac{Weff}{Leff} \cdot m$$

$$RDeff = \frac{RD}{AREAeff}$$

$$RSeff = \frac{RS}{AREAeff}$$

$$RGeff = RG \cdot \frac{AREAeff}{m^2}$$

ACM=1

$$AREAeff = Weff \cdot Leff \cdot m$$

$$RDeff = \frac{RD}{m}$$

Or if RD=0,

$$RDeff = RSH \cdot \frac{HDIF}{Weff \cdot m} + RSHL \cdot \frac{LDIF + ALIGN}{Weff \cdot m}$$

$$RGeff = \frac{RG}{m}$$

or if RG=0,

$$RGeff = RSHG \cdot \frac{Weff}{Leff \cdot m}$$

$$RSeff = \frac{RS}{m}$$

or if RS=0,

$$RSeff = RSH \cdot \frac{HDIF}{Weff \cdot m} + RSHL \cdot \frac{LDIF - ALIGN}{Weff \cdot m}$$

Resulting calculations

$$ISeff = IS \cdot AREAeff$$

$$CGSeff = CGS \cdot AREAeff$$

$$CGDeff = CGD \cdot AREAeff$$

$$BETAeff = BETA \cdot \frac{Weff}{Leff} \cdot m$$

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
$$$ 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
$$$ 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:

vgd < FC · PB

$$cgd = CGDeff \cdot \left(1 - \frac{vgd}{PB}\right)^{-M}$$

vgs < FC · PB

$$cgs = CGSeff \cdot \left(1 - \frac{vgs}{PB}\right)^{-M}$$

Forward Bias:

vgd Š FC · PB

$$cgd = TT \cdot \frac{\partial igd}{\partial vgd} + CGDeff \cdot \frac{1 - FC \cdot (1 + M) + M \cdot \dfrac{vgd}{PB}}{(1 - FC)^{M+1}}$$

vgs Š FC · PB

$$cgs = TT \cdot \frac{\partial igs}{\partial vgs} + CGSeff \cdot \frac{1 - FC \cdot (1 + M) + M \cdot \dfrac{vgs}{PB}}{(1 - FC)^{M+1}}$$

DCAP=2 (Star-Hspice Default)

Reverse Bias:

vgd < 0

$$cgd = CGDeff \cdot \left(1 - \frac{vgd}{PB}\right)^{-M}$$

vgs < 0

$$cgs = CGSeff \cdot \left(1 - \frac{vgs}{PB}\right)^{-M}$$

Forward Bias:

vgd Š 0

$$cgd = TT \cdot \frac{\partial igd}{\partial vgd} + CGDeff \cdot \left(1 + M \cdot \frac{vgd}{PB}\right)$$

vgs Š 0

$$cgs = TT \cdot \frac{\partial igs}{\partial vgs} + CGSeff \cdot \left(1 + M \cdot \frac{vgs}{PB}\right)$$

DCAP=3

Limits peak depletion capacitance to FC · CGDeff or FC · 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.

$$
Cgs = \frac{CGS}{4\sqrt{1 - \dfrac{vnew}{PB}}} \cdot \left[1 + \frac{veff - vte}{\sqrt{(veff - vte)^2 + (0.2)^2}} \right] \cdot
$$

$$
\left[1 + \frac{vds}{\sqrt{vds^2 + \left(\dfrac{1}{ALPHA}\right)^2}} \right] + \left[\frac{CGD}{2} \cdot \left(1 - \frac{vds}{\sqrt{vds^2 + \left(\dfrac{1}{ALPHA}\right)^2}} \right) \right]
$$

$$
Cgd = \left(\frac{CGS}{4\sqrt{1 - \dfrac{vnew}{PB}}} \cdot \left[1 + \frac{veff - vte}{\sqrt{(veff - vte)^2 + (0.2)^2}} \right] \cdot \left[1 - \frac{vds}{\sqrt{vds^2 + \left(\dfrac{1}{ALPHA}\right)^2}} \right] \right) +
$$

$$
\left(\frac{CGD}{2} \cdot \left[1 + \frac{vds}{\sqrt{vds^2 + \left(\dfrac{1}{ALPHA}\right)^2}} \right] \right)
$$

where:

$$
vte = VTO + GAMDS \cdot vds + K1(vbs) = \text{effective threshold}
$$

$$
veff = \frac{1}{2}\left[vgs + vgd + \sqrt{vds^2 + \left(\frac{1}{ALPHA}\right)^2} \right]
$$

and:

$$vnew = \frac{1}{2}[veff + vte + \sqrt{(veff - vte)^2 + (0.2)^2}]$$

```
CGD = High -vds Cgd at vgs = 0
CGS = High -vds Cgs at vgs = 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 vgs below the threshold voltage and Vds>0 (normal bias condition), Cgd is greater than Cgs and rises with Vds, while Cgs drops with Vds.

- Although Cgd properly goes to a small constant representing a sidewall capacitance, Cgs drops asymptotically to zero with decreasing Vgs.

- (For the behavior for Vds<0, interchange Cgs and Cgd and replace Vds with -Vds 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 Vgs 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 · 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

Parameter	Default	Description
CALPHA	ALPHA	Saturation factor for capacitance model
CGAMDS	GAMDS	Threshold lowering factor for capacitance
CVTO	VTO	Threshold voltage for capacitance model
FC	0.5	PB multiplier – typical value 0.9 gate diode limiting voltage=FC · PB.
M (MJ)	0.5	Junction grading coefficient
VDEL	0.2	Transition width for Vgs

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 $\rightarrow$ 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

Figure 17-6: CAPOP=1 vs. CAPOP=2. Cgs, Cgd vs. Vds for Vgs=-1.5, -1.0, -0.5, 0

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:

$$vgst = vgs - VTO$$

vgst< 0   Channel pinched off

$$ids = 0$$

0<vgst<vds   Saturated region

$$ids = BETAeff \cdot vgst^2 \cdot (1 + LAMBDA \cdot vds)$$

0<vds<vgst   Linear region

$$ids = BETAeff \cdot vds \cdot (2 \cdot vgst - vds) \cdot (1 + LAMBDA \cdot vds)$$

The drain current at zero vgs bias (ids) is related to VTO and BETA by the equation:

$$idss = BETAeff \cdot VTO^2$$

At a given vgs, LAMBDA can be determined from a pair of drain current and drain voltage points measured in the saturation region where vgst<vds:

$$LAMBDA = \left(\frac{ids2 - ids1}{ids1 \cdot vds2 - ids2 \cdot vds1} \right)$$

DC Model LEVEL 2

The DC characteristics of the JFET LEVEL 2 model are represented by the nonlinear current source (ids). The value of ids is determined by the equations:

$$vgst = vgs - VTO$$

vgst<0   Channel pinched off

$$ids = 0$$

0<vgst $\leq$ vds, vgs Š 0   Saturated region, forward bias

$$ids = BETAeff \cdot vgst^2 \cdot [1 + LAMBDA \cdot (vds - vgst) \cdot (1 + LAM1 \cdot vgs)]$$

0<vgst<vds, vgs<0   Saturated region, reverse bias

$$ids = BETAeff \cdot vgst^2 \cdot \left[1 - LAMBDA \cdot (vds - vgst) \cdot \frac{vgst}{VTO}\right]$$

0<vds<vgst   Linear region

$$ids = BETAeff \cdot vds(2 \cdot vgst - vds)$$

DC Model LEVEL 3

The DC characteristics of the MESFET LEVEL 3 model are represented by the nonlinear hyperbolic tangent current source (ids). The value of ids is determined by the equations:

vds>0   Forward region

If model parameters VP and VTO are not specified they are calculated as:

$$VP = -\frac{q \cdot NCHAN \cdot Aeff^2}{2 \cdot D \cdot \varepsilon_o}$$

$$VTO = VP + VBI$$

then,

$$vgst = vgs - [VTO + GAMDS \cdot vds + K1(vbs)]$$

$$beteff = \frac{BETAeff}{(1 + UCRIT \cdot vgst)}$$

vgst<0   Channel pinched off

$$ids = idsubthreshold(N0, ND, vds, vgs)$$

vgst>0, SAT=0   On region

$$ids = beteff \cdot (vgst^{VGEXP}) \cdot (1 + LAMBDA \cdot vds) \cdot \tanh(ALPHA \cdot vds)$$

$$+ idsubthreshold(N0, ND, vds, vgs)$$

vgst>0, SAT=1   On region

$$ids = beteff \cdot (vgst^{VGEXP}) \cdot (1 + LAMBDA \cdot vds) \cdot \tanh\left(ALPHA \cdot \frac{vds}{vgst}\right)$$

$$+ idsubthreshold(N0, ND, vds, vgs)$$

vgst>0, SAT=2, vds<3/ALPHA   On region

$$ids = beteff \cdot vgst^2 \cdot (1 + LAMBDA \cdot vds) \cdot \left[1 - \left(1 - ALPHA \cdot \frac{vds}{3}\right)^3\right]$$

$$+ idsubthreshold(N0, ND, vds, vgs)$$

vgst>0, SAT=2, vds>3/ALPHA   On region

$$ids = beteff \cdot vgst^2 \cdot (1 + LAMBDA \cdot vds)$$

$$+ idsubthreshold(N0, ND, vds, vgs)$$

If vgst >0, SAT=3 is the same as SAT=2, except exponent 3 and denominator 3 are parameterized as SATEXP, and exponent 2 of vgst is parameterized as VGEXP.

Note: *idsubthreshold is a special function that calculates the subthreshold currents given the model parameters N0 and ND.*

JFET and MESFET Noise Models

Noise Parameters

Name (Alias)	Default	Description
AF	1.0	Flicker noise exponent
KF	0.0	Flicker noise coefficient. Reasonable values for KF are in the range 1e-19 to 1e-25 V^2 F.
NLEV	2.0	Noise equation selector
GDSNOI	1.0	Channel noise coefficient. Use with NLEV=3.

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:

$$inrd = \left(\frac{4 \cdot k \cdot t}{rd}\right)^{1/2}$$

$$inrs = \left(\frac{4 \cdot k \cdot t}{rs}\right)^{1/2}$$

Channel thermal and flicker noise are modeled by the current source *ind* and defined by the equation:

$$ind = channelthermalnoise + flickernoise$$

If the model parameter NLEV is less than 3, then:

$$channelthermalnoise = \left(\frac{8 \cdot k \cdot t \cdot gm}{3}\right)^{1/2}$$

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 *Tsivids, Yanis P., Operation and Modeling of the MOS Transistor, McGraw-Hill, 1987, p. 340*).

For NLEV=3

$$channel\,thermal\,noise = \left(\frac{8kt}{3} \cdot BETAeff \cdot (vgs - VTO) \cdot \frac{1 + a + a^2}{a} \cdot GDSNOI \right)$$

where

$$\alpha = 1 - \frac{vds}{vgs - VTO}, \quad \textit{Linear region}$$

$$\alpha = 0 \qquad\qquad\qquad \textit{Saturation region}$$

The flicker noise is calculated as:

$$flicker\,noise = \left(\frac{KF \cdot ids^{AF}}{f} \right)^{1/2}$$

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
RG, V^2/HZ	output thermal noise due to gate resistor
ID, V^2/HZ	output thermal noise due to channel
FN, V^2/HZ	output flicker noise
TOT, V^2/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)

Function	Parameter
capacitance	CTD, CTS
DC	M, TCV, XTI
energy gap	EG, GAP1, GAP2
equation selections	TLEV, TLEVC
grading	M
mobility	BEX
resistance	TRD, TRS

Temperature Effect Parameters

Name (Alias)	Units	Default	Description
BETATCE	1/°	0.0	Beta temperature coefficient for TriQuint model
BEX		0.0	Mobility temperature exponent, correction for low field mobility
CTD	1/°	0.0	Temperature coefficient for gate-drain junction capacitance. TLEVC=1 enables CTD to override the default temperature compensation.
CTS	1/°	0.0	Temperature coefficient for gate-source junction capacitance. TLEVC=1 enables CTS to override the default temperature compensation.
EG	eV	1.16	Energy gap for the gate to drain and gate to source diodes at 0 °K 1.17 - silicon 0.69 - Schottky barrier diode 0.67 - germanium 1.52 - gallium arsenide
GAP1	eV/°	7.02e-4	First bandgap correction factor, from Sze, alpha term 7.02e-4 - silicon 4.73e-4 - silicon 4.56e-4 - germanium 5.41e-4 - gallium arsenide

Name (Alias)	Units	Default	Description
GAP2	x	1108	Second bandgap correction factor, from Sze, beta term 1108 - silicon 636 - silicon 210 - germanium 204 - gallium arsenide
M (MJ)		0.50	Grading coefficient for gate-drain and gate-source diodes 0.50 - step junction 0.33 - linear graded junction
N		1.0	Emission coefficient for gate-drain and gate-source diodes
TCV (VTOTC)	$1/^\circ$	0.0	Temperature compensation coefficient for VTO (threshold voltage)
TLEV		0.0	Temperature equation selector for junction diodes. Interacts with the TLEVC parameter.
TLEVC		0.0	Temperature equation selector for junction capacitances and potential. Interacts with the TLEV parameter.
TPB	$V/^\circ$	0.0	Temperature coefficient for PB. TLEVC=1 or 2 overrides the default temperature compensation.
TRD (TDR1)	$1/^\circ$	0.0	Temperature coefficient for drain resistance
TRG (TRG1)	$1/^\circ$	0	Temperature coefficient for gate resistance
TRS (TRS1)	$1/^\circ$	0.0	Temperature coefficient for source resistance

Name (Alias)	Units	Default	Description
XTI		0.0	Saturation current temperature exponent XTI=3 for silicon diffused junction or XTI=2 for Schottky barrier diode

Temperature Compensation Equations

Energy Gap Temperature Equations

To determine energy gap for temperature compensation, use the equation:

TLEV = 0 or 1

$$egnom = 1.16 - 7.02e{-}4 \cdot \frac{tnom^2}{tnom + 1108.0}$$

$$eg(t) = 1.16 - 7.02e{-}4 \cdot \frac{t^2}{t + 1108.0}$$

TLEV = 2

$$egnom = EG - GAP1 \cdot \frac{tnom^2}{tnom + GAP2}$$

$$eg(t) = EG - GAP1 \cdot \frac{t^2}{t + GAP2}$$

Saturation Current Temperature Equations

The saturation current of the gate junctions of the JFET varies with temperature according to the equation:

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

TLEV=0 or 1

$$facln = \frac{EG}{vt(tnom)} - \frac{EG}{vt(t)} + XTI \cdot \ln\left(\frac{t}{tnom}\right)$$

TLEV=2

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

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

$$CGS(t) = CGS \cdot \left[1 + M \cdot \left(4.0e\text{-}4 \cdot \Delta t - \frac{PB(t)}{PB} + 1\right)\right]$$

$$CGD(t) = CGD \cdot \left[1 + M \cdot \left(4.0e\text{-}4 \cdot \Delta t - \frac{PB(t)}{PB} + 1\right)\right]$$

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

$$CGS(t) = CGS \cdot \left(1 - 0.5 \cdot dpbdt \cdot \frac{\Delta t}{PB} \right)$$

$$CGD(t) = CGD \cdot \left(1 - 0.5 \cdot dpbdt \cdot \frac{\Delta t}{PB} \right)$$

where:

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

TLEV=0 or 1

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

TLEV=2

$$dpbdt = \frac{-\left[egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left(2 - \frac{tnom}{tnom + GAP2}\right) - PB\right]}{tnom}$$

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:

$$BETA(t) = BETA \cdot \left(\frac{t}{tnom}\right)^{BEX} \qquad \textit{If BETATCE=0}$$

Otherwise (TriQuint model):

$$BETA(T) = BETA \cdot 1.01^{BETATCE(t - tnom)}$$

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 MESFET Model For Use In the Design of GaAs Integrated Circuits, IEEE Tran, Microwave* and *H.Statz, 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.

Alias	HSPICE Printout Name	Note
Q	*VGEXP*	
GAMMA	*GAMDS*	sign opposite of TriQuint's original
VTOTC	*TCV*	sign opposite of TriQuint's original

Alias	HSPICE Printout Name	Note
TRG1	*TRG*	
TRD1	*TRD*	
TRS1	*TRS*	

TOM Model Parameters

Name (Alias)	Description
BETATCE	Temperature coefficient for BETA If betatce is set to a nonzero value: $$BETA(temp) = BETA(tnom) \cdot 1.01^{(BETATCE \cdot (temp - tnom))}$$ The more common HSPICE Beta temperature update is: $$BETA(temp) = BETA(tnom) \cdot \left(\frac{temp}{tnom}\right)^{BEX}$$
DELTA	Ids feedback parameter of the TOM model. This parameter is not used if its value is zero. DELTA can be negative or positive. $$i_{ds} \Rightarrow \frac{i_{ds}}{max[(-1 + v_{ntol}),(DELTA + v_{ds} \cdot i_{ds})]}$$
CAPDS	Drain-to-source capacitance $$CAPDSeff = CAPDS \cdot \frac{Weff}{Leff} \cdot M$$

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.*

Avant!

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[1] 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.

---

1. Dmitri Kuznetsov, "Optimal Transient Simulation of Transmission Lines," *IEEE Trans. Circuits Syst.*, vol. 43, pp. 110-121, Feb. 1996.

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:

$$-\frac{\partial}{\partial z}\mathbf{v}(z, \omega) = [\mathbf{R}(\omega) + j\omega\mathbf{L}(\omega)]\mathbf{i}(z, \omega)$$

$$-\frac{\partial}{\partial z}\mathbf{i}(z, \omega) = [\mathbf{G}(\omega) + j\omega\mathbf{C}(\omega)]\mathbf{v}(z, \omega)$$

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:

$$\mathbf{R}(f) \cong \mathbf{R_o} + \sqrt{f}(1 + j)\mathbf{R_s}$$

where $\mathbf{R_o}$ is the DC resistance matrix and $\mathbf{R_s}$ 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:

$$\mathbf{G}(f) \cong \mathbf{G_o} + \frac{f}{\sqrt{1 + (f/f_{gd})^2}}\mathbf{G_d}$$

where $\mathbf{G_o}$ models the shunt current due to free electrons in imperfect dielectrics and $\mathbf{G_d}$ 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 $\mathbf{R_o}$, $\mathbf{R_s}$, $\mathbf{G_o}$, and $\mathbf{G_d}$ are nonnegative (can be zero). Off-diagonal terms of impedance matrices **L**, $\mathbf{R_o}$, and $\mathbf{R_s}$ are nonnegative. Off-diagonal terms of admittance matrices **C**, $\mathbf{G_o}$, and $\mathbf{G_d}$ are nonpositive. Off-diagonal terms of all matrices can be zero.

---

1. C. A. Balanis, *Advanced Engineering Electromagnetics*, New York: Wiley, 1989.

The elements of admittance matrices are related to the self/mutual admittances (as those inputted by U Element):

$$Y_{ii} = \sum_{j=1}^{N} Y_{ij}^{(self)/(mutual)}$$

$$Y_{ij} = -Y_{ij}^{mutual}, i \neq j$$

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,[1] 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.

---

1. A good source on transmission lines is: H. B. Bakoglu, *Circuits, Interconnections and Packaging for VLSI.* Reading, MA: Addison-Wesley, 1990.

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.

Figure 18-2: System Model for Transmission Lines

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.

$\mathbf{W_{vr}}$ and $\mathbf{W_{vb}}$ are the forward and backward matrix propagation functions for voltage waves; $\mathbf{T_1}$, $\mathbf{T_2}$ and Γ_1, Γ_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

The propagation functions, $\mathbf{W_{vr}}$ and $\mathbf{W_{vb}}$, describe how a wave is affected by its propagation from one termination to another, and are equal for the forward and backward directions: $\mathbf{W_{vr}}$ and $\mathbf{W_{vb}}$. 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)

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 ($\mathbf{R_s} = \mathbf{G_d} = \mathbf{0}$).

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 $\mathbf{G_d}$ term is introduced as follows:

$$\mathbf{G}(f) \cong \mathbf{G_o} + \frac{f}{\sqrt{1 + (f/f_{gd})^2}} \mathbf{G_d}$$

The default value of f_{gd} is 1GHz. 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 `fgd` 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:

$$\mathbf{R}(f) \cong \mathbf{R_o} + \sqrt{f}(1+j)\mathbf{R_s}$$

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>
```

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)
o1...oN	Nodes for the far-end signal-conductor terminal

oR	Nodes for the far-end reference-conductor terminal
L	Length of a transmission line
RLGCMODEL	Name of the RLGC model
FSMODEL	Name of the field-solver model
RLGCFILE	Name of the external file with RLGC parameters (see "Input Model 4: W Element RLGC File" on page 18-24)
UMODEL	Name of U .MODEL (see "Input Model 2: U Model" on page 18-20).
INCLUDERSIMAG	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).

Figure 18-6: Terminal Node Numbering

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
+ Co=matrix_entries [ Ro=matrix_entries Go=matrix_entries
+ Rs=matrix_entries Gd=matrix_entries Rognd=val Rsgnd=val
+ Lgnd=val ]
```

    N   Number of signal conductors (same as that in the element card)

    L

                DC inductance matrix per unit length $\left[\dfrac{H}{m}\right]$

    C

                DC capacitance matrix per unit length $\left[\dfrac{F}{m}\right]$

Optional Parameters (default values are zero)

Ro	DC resistance matrix per unit length	$\left[\dfrac{\Omega}{m}\right]$
Go	DC shunt conductance matrix per unit length	$\left[\dfrac{S}{m}\right]$
Rs	Skin-effect resistance matrix per unit length	$\left[\dfrac{\Omega}{m\sqrt{Hz}}\right]$
Gd	Dielectric-loss conductance matrix per unit length	$\left[\dfrac{S}{m \cdot Hz}\right]$
Lgnd	DC inductance matrix per unit length for grounds (reference line)	$\left[\dfrac{H}{m}\right]$
Rognd	DC resistance matrix per unit length for grounds	$\left[\dfrac{\Omega}{m}\right]$
Rsgnd	Skin-effect resistance matrix per unit length for grounds	$\left[\dfrac{\Omega}{m\sqrt{Hz}}\right]$

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

(c) Transient Waveforms

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 `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

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 $\mathbf{G_d}$, the W Element defaults $\mathbf{G_d}$ to zero when it uses the U-model input. In future Star-Hspice releases, the W Element will have its own .MODEL with $\mathbf{G_d}$ capability. For this release, use RLGC file to specify nonzero $\mathbf{G_d}$.

Handling the Skin-effect Matrix

The skin-effect resistance $\mathbf{R_s}$ is used differently by U and W Elements. In a W Element, the $\mathbf{R_s}$ matrix specifies the square-root dependence of the frequency-dependent resistance,

$$\mathbf{R}(f) \cong \mathbf{R_o} + \sqrt{f}(1+j)\mathbf{R_s}$$

In a U Element, $\mathbf{R}$ is the value of skin resistance at the frequency:

$$\mathbf{R} \cong \mathbf{R}_c + \mathbf{R_s}$$

where the core resistance $\mathbf{R_c}$ is equivalent to the W Element's DC resistance $\mathbf{R_o}$. The frequency at which U Element computes $\mathbf{R_s}$ is:

$$f_{skin} = \frac{1}{15 \cdot RISETIME}$$

Table 18-1: RLGC Matrices in W Element and U Model

W Element Parameters	U Model Parameters

L, C

$$\begin{bmatrix} L_{11} & & \\ L_{12} & L_{22} & \\ L_{13} & L_{23} & L_{33} \end{bmatrix} \qquad \begin{bmatrix} C_{r1}+C_{12}+C_{13} & & \\ -C_{12} & C_{r2}+C_{12}+C_{23} & \\ -C_{13} & -C_{23} & C_{r3}+C_{13}+C_{23} \end{bmatrix}$$

$\mathbf{G_o}$, $\mathbf{G_d}$

$$\begin{bmatrix} G_{r1}+G_{12}+G_{13} & & \\ -G_{12} & G_{r2}+G_{12}+G_{23} & \\ -G_{13} & -G_{23} & G_{r3}+G_{13}+G_{23} \end{bmatrix} \qquad \begin{bmatrix} 0 & & \\ 0 & 0 & \\ 0 & 0 & 0 \end{bmatrix}$$

Nlay=1 (no skin effect)   Nlay=2 (skin effect present)

$\mathbf{R_o}$

$$\begin{bmatrix} R_{11}+R_{rr} & & \\ R_{rr} & R_{22}+R_{rr} & \\ R_{rr} & R_{rr} & R_{33}+R_{rr} \end{bmatrix} \qquad \begin{bmatrix} R_{1c}+R_{rc} & & \\ R_{rc} & R_{2c}+R_{rc} & \\ R_{rc} & R_{rc} & R_{3c}+R_{rc} \end{bmatrix}$$

Nlay=1 (no skin effect)   Nlay=2 (skin effect present)

$\mathbf{R_s}$

$$\begin{bmatrix} 0 & & \\ 0 & 0 & \\ 0 & 0 & 0 \end{bmatrix} \qquad \frac{1}{\sqrt{f_{skin}}}\begin{bmatrix} R_{1s}+R_{rs} & & \\ R_{rs} & R_{2s}+R_{rs} & \\ R_{rs} & R_{rs} & R_{3s}+R_{rs} \end{bmatrix}$$

If you do not specify the RISETIME option, the U Element uses Tstep from the .tran card.

For U models with	W Element
RLGC input; `Elev=2`	Uses $\mathbf{R_s}$ values you specify in the U model.
Geometric input; `Elev=1`	Divides the $\mathbf{R_s}$ computed internally by the U model, and divides it by $\sqrt{f_{skin}}$ to obtain the $\mathbf{R_s}$. For `Elev=1`, the value of $\mathbf{R_s}$ in the U model printout is not be the same as $\mathbf{R_s}$ actually used by the W Element.
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.

```
* 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
+
+ Cr1=17.43pF
+ C12=5.41pF     Cr2=10.1pF
+ C13=1.08pF     C23=5.72pF     Cr3=17.67pF
+
+ R1c=42.5       R2c=41.0       R3c=33.5
+
+ Gr1=0.44387mS
+ 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)
```

```
.AC lin 1000 0Hz 1GHz
.DC v1 0v 5v 0.1v
.TRAN 0.1ns 200ns
.END
```

Figure 18-8: 4-Conductor Line

Four-conductor line

Input Model 3: Field-Solver Model

Instead of RLGC matrices, you can directly use geometric data with the W Element using a new built-in field solver in Star-Hspice. To use the W Element with a field solver, specify `FSmodel=model_name` on the W Element card. The Star-Hspice field solver is described in "Extracting Transmission Line Parameters" on page 18-34.

Input Model 4: 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:

N   Number of signal conductors (same as that in the element card)

L
              DC inductance matrix per unit length $\left[\dfrac{H}{m}\right]$

C
              DC capacitance matrix per unit length $\left[\dfrac{F}{m}\right]$

Optional Parameters

R$_o$
              DC resistance matrix per unit length $\left[\dfrac{\Omega}{m}\right]$

G$_o$
              DC shunt conductance matrix per unit length $\left[\dfrac{S}{m}\right]$

R$_s$
              Skin-effect resistance matrix per unit length $\left[\dfrac{\Omega}{m\sqrt{Hz}}\right]$

G$_d$
              Dielectric-loss conductance matrix per unit length $\left[\dfrac{S}{m \cdot Hz}\right]$

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 sp 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 sp 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 sp 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 sp 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)
```

```
+ (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 sp 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)
```

```
+ (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 1.727973e+00 6.299210e-03 1.726616e+00)
+ (4.449938e+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 3.414679e+00 6.299210e-03 3.411927e+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 4.357583e+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 7.191186e+00)
+ (9.230769e+08 8.167936e+00 6.299210e-03 8.161252e+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 sp 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)
```

```
+ (1.196411e+07 4.634076e-05 -5.815525e-06 4.633777e-05)
+ (1.648352e+07 6.384583e-05 -8.012324e-06 6.384171e-05)
+ (2.131439e+07 8.255729e-05 -1.036052e-05 8.255196e-05)
+ (2.649007e+07 1.026043e-04 -1.287632e-05 1.025977e-04)
+ (3.204884e+07 1.241351e-04 -1.557833e-05 1.241271e-04)
+ (3.803487e+07 1.473209e-04 -1.848803e-05 1.473114e-04)
+ (4.449938e+07 1.723600e-04 -2.163030e-05 1.723488e-04)
+ (5.150215e+07 1.994838e-04 -2.503421e-05 1.994710e-04)
+ (5.911330e+07 2.289642e-04 -2.873385e-05 2.289494e-04)
+ (6.741573e+07 2.611221e-04 -3.276951e-05 2.611052e-04)
+ (7.650809e+07 2.963396e-04 -3.718913e-05 2.963205e-04)
+ (8.650875e+07 3.350753e-04 -4.205026e-05 3.350537e-04)
+ (9.756098e+07 3.778840e-04 -4.742254e-05 3.778596e-04)
+ (1.098398e+08 4.254437e-04 -5.339105e-05 4.254163e-04)
+ (1.235615e+08 4.785920e-04 -6.006089e-05 4.785611e-04)
+ (1.389961e+08 5.383752e-04 -6.756338e-05 5.383404e-04)
+ (1.564859e+08 6.061186e-04 -7.606484e-05 6.060795e-04)
+ (1.764706e+08 6.835254e-04 -8.577900e-05 6.834812e-04)
+ (1.995249e+08 7.728220e-04 -9.698528e-05 7.727721e-04)
+ (2.264151e+08 8.769759e-04 -1.100561e-04 8.769193e-04)
+ (2.581852e+08 1.000031e-03 -1.254989e-04 9.999669e-04)
+ (2.962963e+08 1.147647e-03 -1.440240e-04 1.147573e-03)
+ (3.428571e+08 1.327992e-03 -1.666563e-04 1.327906e-03)
+ (4.010283e+08 1.553307e-03 -1.949322e-04 1.553207e-03)
+ (4.757709e+08 1.842808e-03 -2.312632e-04 1.842689e-03)
+ (5.753425e+08 2.228480e-03 -2.796630e-04 2.228336e-03)
+ (7.145791e+08 2.767786e-03 -3.473433e-04 2.767608e-03)
+ (9.230769e+08 3.575363e-03 -4.486902e-04 3.575132e-03)
+ (1.269625e+09 4.917650e-03 -6.171404e-04 4.917332e-03)
+ (1.959184e+09 7.588526e-03 -9.523220e-04 7.588036e-03)
+ (4.000000e+09 1.549324e-02 -1.944324e-03 1.549224e-02)
+ )
```

.MODEL Syntax

See the .

Example

This is an example of a 4-conductor transmission line system.

Table 18-2: Input File Listing

| Header, options and sources | ```
W Element Tabular Model Example
.OPTIONS POST
V1 7 0 ac=1v dc=0.5v pulse(0.5v 1.5v 0ns 0.1ns)
V2 8 0 dc=1v
``` |
|---|---|
| Analysis | ```
.DC v1 0.5v 5.5v 0.1v SWEEP length POI 2 1.2 2
.AC lin 200 0Hz 1GHz SWEEP Ro POI 3 400 41.6667 400
.TRAN 0.1ns 50ns
``` |
| Termination | ```
R1 7 1 50
R2 4 0 450
R3 5 0 450
R8 6 0 450
R5 4 5 10800
R6 5 6 10800
R7 4 6 1393.5
``` |
| Analytical RLGC model (W Element) | ```
.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 RLGCMODEL=analymod
.MODEL analymod sp W MODELTYPE=RLGC N=3
+ Lo=2.41667e-6
+    0.694444e-6   2.36111e-6
+    0.638889e-6   0.694444e-6   2.41667e-6
+ Co=20.9877e-12
+   -12.3457e-12    29.3210e-12
+   -4.01235e-12   -12.3457e-12   20.9877e-12
+ Ro=41.6667
+    0         41.6667
+    0         0          41.6667
+ Go=0.585937e-3
+    0         0.585937e-3
+    0         0             0.585937e-3
+ Rs=0.785e-5
+    0         0.785e-5
+    0         0             0.785e-5
+ Gd=0.285e-6
+    0         0.285e-6
+    0         0             0.285e-6
.ENDS sub
``` |

| Tabular RLGC model (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

| RLGC table model definition | ```
.MODEL trmod sp W MODELTYPE=TABLE N=3
+ LMODEL=lmod CMODEL=cmod RMODEL=rmod GMODEL=gmod
``` |
| --- | --- |
| C model | ```
.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 )
``` |
| L model | ```
.MODEL lmod sp N=3 VALTYPE=REAL INTERPOLATION=LINEAR
+ INFINITY= 2.41667e-06 6.94444e-07 2.36111e-06
+ 6.38889e-07 6.94444e-07 2.41667e-06 FSTOP=1e+07
+ DATA=( 25
+ 2.41667e-06 6.94444e-07 2.36111e-06 6.38889e-07
+ 6.94444e-07 2.41667e-06 2.41861e-06 6.94444e-07
...
+ 2.41707e-06 6.94444e-07 2.36151e-06 6.38889e-07
+ 6.94444e-07 2.41707e-06 )
``` |
| R model | ```
.MODEL rmod sp N=3 VALTYPE=REAL INTERPOLATION=LINEAR
+ FSTOP=1e+10
+ DATA=( 200
+ 41.6667 0 41.6667 0 0 41.6667
+ 41.7223 0 41.7223 0 0 41.7223
...
+ 42.4497 0 42.4497 0 0 42.4497
+ 42.4517 0 42.4517 0 0 42.4517 )
``` |
| G model | ```
.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 0 1.42377
+ 1.42381 0 1.42381 0 0 1.42381 )
``` |

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[1] and filament method that is also implemented in Raphael[2].

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: $\mathbf{C}$, $\mathbf{L}$, $\mathbf{R_o}$, $\mathbf{R_s}$, $\mathbf{G_o}$, and $\mathbf{G_d}$ (see "Using Transmission Line Equations and Parameters" on page 18-3).

---

1. K. S. Oh, D. B. Kuznetsov, and J. E. Schutt-Aine, "Capacitance computations in a multilayered dielectric medium using closed-form spatial Green's functions," *IEEE Trans. Microwave Theory and Tech.*, vol. 42, pp. 1443-1453, August 1994.
2. *Raphael Reference Manual*, Avant! Corporation, December 1998.

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, $\mathbf{G_o}$ and $\mathbf{G_d}$.

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 W 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: CONDUCTIVITY = -1 (perfect conductor), ER = 1, UR = 1. 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: CONDUCTIVITY = 0 (lossless dielectric), LOSSTANGENT = 0 (lossless dielectric), ER = 1, UR = 1. AIR is a predefined dielectric name with default values and cannot be redefined.

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

```
.LAYERSTACK sname <BACKGROUND=mname>
+ <LAYER=(mname,thickness) ...>
```

sname	Layer stack name
mname	Material name
BACKGROUND	Background dielectric material name. By default, AIR is assumed for the background.
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>`

WIDTH	Width of rectangle (length in x-direction).
HEIGHT	Height of rectangle (length in y-direction).
NW	Number of segments for the width discretization.
NH	Number of segments for the height discretization.

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

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>
```

RADIUS   Radius of the circle.

N   Number of segments for discretization.

Figure 18-10: Coordinates of a Circle

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 18-11: Coordinates of a Strip Polygon

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

```
POLYGON VERTEX=(x1 y1 x2 y2 ...) <N=(n1,n2,...)>
```

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 (x1 y1) to (x2 y2).

Figure 18-12: Coordinates of a Polygon

Example

Rectangular polygon using the default discretization:

```
POLYGON VERTEX=(1 10 1 11 5 11 5 10)
```

Rectangular polygon using five segments for each edge:

```
POLYGON VERTEX=(1 10 1 11 5 11 5 10) N=5
```

Rectangular polygon using the different number of segments for each edge:

```
POLYGON VERTEX=(1 10 1 11 5 11 5 10) 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>
```

Option	Description	Default Value
name	Option name.	
ACCURACY	Sets the solver accuracy to either LOW, MEDIUM, or HIGH.	HIGH
GRIDFACTOR	Multiplication factor (integer) to determine the final number of segments used in discretization.	1
PRINTDATA	Specifies that the solver prints output matrices.	NO
COMPUTEGO	Specifies that the solver computes the static conductance matrix.	YES
COMPUTEGD	Specifies that the solver computes the dielectric loss matrix.	NO
COMPUTERO	Specifies that the solver computes the DC resistance matrix.	YES
COMPUTERS	Specifies that the solver computes the skin-effect resistance matrix.	NO

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 GRIDFACTOR to obtain the final number of segments.

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* HIGH *option, increase the* GRIDFACTOR *value.*

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 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 *i*th signal conductor, counting without reference and floating conductors, is associated with the *i*th input and output terminals specified in the corresponding W Element. Floating and reference conductors can appear in any order.

Syntax

```
.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> ) ...
```

`mname`	Model name
`LAYERSTACK`	Associated layer stack name
`FSOPTIONS`	Associated option name. If this entry is not specified, the default options are used.
`RLGCFILE`	Specifies the output file for RLGC 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.
`OUTPUTFORMAT`	Specifies the W Element model syntax format for RLGC matrices in RLGCFILE. The default format is a RLGC model.
`SHAPE`	Shape name
`x y`	The coordinate of the local origin.
`MATERIAL`	Conductor material name. If this entry is not specified, `PEC` is assumed.
`TYPE`	One of the following conductor types: `SIGNAL`: a signal node in W Element. `REFERENCE`: the reference node in W Element. `FLOATING`: floating conductor, no reference to W Element. The default value of `TYPE` is `SIGNAL`.

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-Hpsice netlist.

In this case, you can derive the exact analytical formulas for all of the transmission line parameters:[1]

Figure 18-13: Cylindrical Conductor Above a Ground Plane

$$C = \frac{2\pi\varepsilon}{\mathrm{acosh}\left(\frac{2H}{d}\right)}$$

$\sigma_c = 5.76e7$

$\varepsilon_r = 4.0$ d = 1 mm   H = 3 mm

$\tan \delta = 1.2e\text{-}3$

Ideal Ground Plane (PEC)

---

1. S. Ramo, J. R. Whinnery, and T. V. Duzer, Fields and Waves in Communication Electronics, 2nd ed. New York: Wiley, 1984.

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
``` |
|---|---|
| W Element | ```
W1 in1 gnd out1 gnd FSmodel=cir_trans N=1 l=0.5
``` |
| 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
``` |

$$L = \frac{1}{\mu\varepsilon}C^{-1}$$

$$G = \frac{\sigma_d}{\varepsilon}C = \omega \cdot \tan(\delta) \cdot C$$

$$R = \frac{1}{\sigma_c \delta \pi d}\left[\frac{2H/d}{\sqrt{(2H/d)^2 - 1}}\right] = \sqrt{f}\sqrt{\frac{\pi\mu}{\sigma_c}}\frac{1}{\pi d}\left[\frac{2H/d}{\sqrt{(2H/d)^2 - 1}}\right]$$

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 (R_0) and conductance (G_0) are not included in the computed values.

Table 18-5: Comparison Result for Example 1

Value	Exact	Computed
C (pF/m)	89.81	89.66
L (nH/m)	494.9	543.9[a]
G (mS/m)	0.1354	0.1352
R (Ω/m)	1.194	1.178

a. Due to the magnetic coupling within the resistive conductor, the real inductance value computed is larger than the "exact" value obtained from the free space relationship between capacitance and inductance. If you set zero to the resistivity of the conductor (by using "PEC" as the material of the conductor), the computed inductance value will be 498.3 [nH/m], which is close to the "exact" value.

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```
	 ```.OPTION PROBE POST```
	 ```+ VIMPULSE in1 gnd PULSE 4.82v 0v 5n 0.5n 0.5n```
	 ```+ 25n```

W Element	```W1 in1 in2 in3 gnd out1 out2 out3 gnd``` ```+ FSmodel=cond3_sys N=3 l=0.5```
Materials	```.MATERIAL diel_1 DIELECTRIC ER=4.3``` ```.MATERIAL diel_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=(diel_1,0.2mm),```
	 ```+ LAYER=(diel_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

Computed	$\begin{bmatrix} 141.1 & -21.36 & -0.90 \\ -21.36 & 92.66 & -17.72 \\ -0.90 & -17.72 & 87.26 \end{bmatrix}$ (pF/m)
Raphael (Finite-Difference Solver)	$\begin{bmatrix} 139.5 & -23.46 & -1.89 \\ -23.69 & 94.60 & -19.89 \\ -1.82 & -19.52 & 85.48 \end{bmatrix}$ (pF/m)
Reference[a]	$\begin{bmatrix} 142.1 & -21.76 & -0.89 \\ -21.73 & 93.53 & -18.10 \\ -0.89 & -18.10 & 87.96 \end{bmatrix}$ (pF/m)

a. W. Delbare and D. D. Zutter, "Space-domain Green's function approach to the capacitance calculation of multiconductor lines in multilayered dielectrics with improved surface charge modeling," *IEEE Trans. Microwave Theory and Tech.*, vol. 37, pp. 1562-1568, October 1989.

Figure 18-15 shows the results of convergence analysis performed based on the total capacitance of the first conductor with respect to the GRIDFACTOR parameter.

Figure 18-15: Convergence of Accuracy Modes

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.

Figure 18-16: Two Traces Between Two Ground Planes

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	$\begin{bmatrix} 214.1 & -105.2 \\ -105.2 & 214.1 \end{bmatrix}$ (pF/m)
FEM Solver	$\begin{bmatrix} 217.7 & -108.2 \\ -108.2 & 217.7 \end{bmatrix}$ (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

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`

Parameter definitions	`.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)`

W Element	`W1 in1 in2 in3 0 out1 out2 out3 0` `+ FSMODEL=cond3_sys N=3 l=0.5`

| Materials | ```
.MATERIAL diel_1 DIELECTRIC ER=4.3
.MATERIAL diel_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=(diel_1,dY1),
+ LAYER= (diel_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
``` |

Avant!

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 19-45.

Two types of analysis are supported for input/output buffers:

- DC analysis
- Transient 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 alphanumeric 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, *value_i,* to the keyword, *keyword_i.* Optional keywords are given in square brackets (see *"Specifying Common Keywords" on page 19-23* for more information).

Figure 19-1: Circuit Diagram for Package

The *gnd* node on the circuit diagram for buffers denotes the ideal SPICE ground node (the notation *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 *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 19-23 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:

```
R_pkg node_out node_pkg R_pkg_value
L_pkg node_pkg node_pin L_pkg_value
C_pkg node_pin gnd      C_pkgvalue
```

where values for *R_pkg*, *L_pkg*, and *C_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

I/O	Input/Output
I/V curve	Current-voltage curve
PU, PD	Pullup, pulldown
PC, GC	Power clamp, ground clamp

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 19-23 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:

```
B_INPUT nd_pc nd_gc nd_in nd_out_of_in
+ file='filename' model='model_name'
+ [typ={typ|min|max|fast|slow}] [power={on|off}]
+ [buffer={1|input}]
+ [interpol={1|2}]
+ [nowarn]
+ [c_com_pc=c_com_pc_value]
+ [c_com_gc=c_com_gc_value]
```

where the total number of external nodes is equal to 4.

If keyword *power=on* (default) is specified, nodes *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, so you can print out the voltage values if required. For example:

```
.PRINT V(nd_pc) V(nd_gc)
```

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_in*. The buffer measures and processes the voltage on this node and sends a response to node *nd_out_of_in*. The node *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

V_out_of_in is a digital signal that assumes values of either 0 or 1 depending on the voltages *V_in, Vinh, Vinl*, 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 < (Vinh+Vinl)/2 and to 1 in the opposite case.
and if V_out_of_in=1	then it goes to 0 only if V_in < Vinl
and if V_out_of_in=0	then it goes to 1 only if V_in>Vinh
Polarity=Inverting	Initially *V_out_of_in* is set to 0 if V_in > (Vinh+Vinl)/2 and to 1 in the opposite case
and if V_out_of_in=1	then it goes to 0 only if V_in > Vinh
and if V_out_of_in=0	then it goes to 1 only if V_in <Vinl

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 Vinl, Vinh, and parameter Polarity). It can be used to drive other circuits.

Output Buffer

The syntax for an output buffer element card is:

```
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] [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]
```

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 *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 19-3: Output Buffer

If keyword *power=on* (default) is specified, nodes *nd_pu*, *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, so you can print out the voltage values if required. For example:

```
.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*. A voltage source must be connected to the node *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. Node *nd_in* can not be connected to the ground.

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.

Tristate Buffer

The syntax for a tristate buffer element card is:

```
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}]
+ [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]
```

Nodes *nd_pc* and *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 *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 19-4: Tristate Buffer

If *keyword power=on* (default) is specified, nodes *nd_pu*, *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:

```
.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 *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:

```
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. Nodes *nd_in*, *nd_en* can not be connected to the ground.

V_in and *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, *V_en*, supersedes the input signal, *V_in.*

If:

ENABLE = Active-High	At t=0 for transient analysis (or for DC analysis), the buffer goes to the ENABLE state if V_en > 0.5 and to DISABLE in the opposite case.
ENABLE = Active-Low	At t=0 for transient analysis (or for DC analysis), the buffer goes to ENABLE state if V_en < 0.5 and to DISABLE in the opposite case.

The buffer is in ENABLE state	Begins 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	Begins 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:

```
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]
```

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:

```
.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:

```
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. Nodes *nd_in, nd_en* can not be connected to the ground.

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 19-10 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_out_of_in*). Because *V_out_of_in* is not always defined (e.g. the buffer is in ENABLE state, or Vinl < V_out < Vinh 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_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_out_of_in* is a voltage source whose value is a function of *V_out* (as well as of thresholds Vinl, Vinh 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_pu* and *nd_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_pu=nd_state_pu* should not be specified.
- Similarly, because open source buffers do not have pulldown circuitry, the option *xv_pd=nd_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_pu* and *nd_pd*, even if the buffer does not have pullup or pulldown circuitry.

All rules given in "Input/Output Buffer" on page 19-14 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:

```
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}]
+ [interpol={1|2}]
+ [nowarn]
+ [c_com_pc=c_com_pc_value]
+ [c_com_gc=c_com_gc_value]
```

The input ECL buffer is similar to the input buffer. The only difference is in default values for *Vinl* and *Vinh*.

Figure 19-6: Input ECL Buffer

Output ECL Buffer

The syntax of the output ECL buffer element card is:

```
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]
```

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:

```
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]
```

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 for more information.

Figure 19-8: Tristate ECL Buffer

Input-Output ECL Buffer

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]
```

Nodes *nd_pc* and *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 *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 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 19-14 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
```
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
```
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= {Buffer_Number | Buffer_Type}

Where *buffer_number* is an integer from the range $1 \leq N \leq 17$. Each buffer has an assigned number as follows:

Buffer Type	Buffer Number	Number of nodes (nominal or min/max)	Notes
INPUT	1	4	
OUTPUT	2	4/6	
INPUT_OUTPUT	3	6/8	
THREE_STATE	4	5/7	
OPEN_DRAIN	5	4/6	
IO_OPEN_DRAIN	6	6/8	
OPEN_SINK	7	4/6	
IO_OPEN_SINK	8	6/8	
OPEN_SOURCE	9	4/6	
IO_OPEN_SOURCE	10	6/8	
INPUT_ECL	11	4	
OUTPUT_ECL	12	3/5	
IO_ECL	13	5/7	
THREE_STATE_ECL	14	4/6	was 17
SERIES	15	not implemented	
SERIES_SWITCH	16	not implemented	
TERMINATOR	17	not implemented	was 14

The value of *buffer_number* and *buffer_type* should match the buffer type specified by keyword *model='...'*. The keyword *buffer= {Buffer_Number | 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).

Parameter/Data	Fast	Slow
C_comp	min	max
Temp_Range	max	min
Voltage_Range	max	min
Pullup_Ref	max	min
Pulldown_Ref	min	max
POWER_Clamp_Ref	max	min
GND_Clamp_Ref	min	max
Rgnd	max	min
Rpower	max	min
Rac	max	min
Cac	min	max
Pulldown	max	min
Pullup	max	min
GND_ Clamp	max	min
POWER_Clamp	max	min

Parameter/Data	Fast	Slow
Ramp	max	min
Rising_waveform	max	min
Falling_waveform	max	min
V_fixture	max	min

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.

interpol={1|2}

Default is *interpol=1* (recommended). The I/V curves and V(t) curves need to be interpolated. Keyword *interpol=1* uses linear interpolation and *interpol=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:

```
.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.*

---

fwf_tune=fwf_tune_value
rwf_tune=rwf_tune_value

Keywords *fwf_tune_value* and *rwf_tune_value* 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_pd=1*, *St_pu=0*. 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 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 *fwf_tune* parameter.

Parameters *fwf_tune* and *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 *fwf_tune* and *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 *fwf_tune* and *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, fwf_tune and/or rwf_tune are used even if ramp_fwf=2, ramp_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, *xv_pu=nd_state_pu* and *xv_pd=nd_state_pd*, can be used to print and/or view the state evolution functions *St_pu(t)* and *St_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_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

By default (default 1) the die capacitance C_comp is connected between *node_out (nd_in* for input buffer) and ideal ground. For simulating power bounce and ground bounce it may be desirable to split C_comp into several parts and connect between *node_out (nd_in* for input buffer) and some (or all) of nodes *node_pu , node_pd , node_pc , node_gc.*

If at least one of the optional parameters c_com_pu , c_com_pd , c_com_pc , c_com_gc is specified then the default 1 is not applicable and unspecified parameters takes value zero (default 2). Values c_com_pu , c_com_pd , c_com_pc , c_com_gc are dimensionless and denote fractions of C_comp connected between *node_out (nd_in* for input buffer) and respective nodes (either *node_pu , node_pd , node_pc , * or *node_gc*). In other words C_comp*c_com_pu is capacitance connected between *node_out* and *node_pu*, etc.

If given, values of c_com_pu , c_com_pd , c_com_pc , and c_com_gc should be nonnegative.

It is expected that c_com_pu + c_com_pd + c_com_pc + c_com_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_comp specified in the IBIS files while during simulation different value of C_comp used, namely C_comp*(c_com_pu+c_com_pd+c_com_pc+c_com_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.

Figure 19-13: Input buffers

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_diff is a differential voltage parameter from the IBIS file (default is 200 mV). Add definition of parameter *V_diff*, voltage controlled voltage source *E_diff_out_of_in*, and a resistor *R_diff_out_of_in.*

Example
```
.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_diff_out_of_in* as the output of the differential input buffer.
```
If  V(nd_in_1) - V(nd_in_2) < V_diff,  V(diff_out_of_in) = 0
if  V(nd_in_1) - V(nd_in_2) > V_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
+ Felement 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_gc0
+ 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, *nd_pu, nd_pd, nd_pc, 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 *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 *nd_out_of_in_1*.
IBIS file at16245a.ibs resides in the directory .ibis, which is located in the
directory from which Star-Hspice is run.

```
***********************************************************
* 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
```

```
*************************************************************

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 nd_in 0 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=  1.5900E+00   V_rwf_max=  3.3000E+00   V_rwf_min=  3.3540E-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/

ftp://ftp.eda.org/pub/ibis/

ftp://ftp.vhdl.org/pub/ibis/

Avant!

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

- 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.

Level	MOSFET Model Description	All Platforms including PC	All Platforms except PC
1	Schichman-Hodges model	X	
2	MOS2 Grove-Frohman model (SPICE 2G)	X	
3	MOS3 empirical model (SPICE 2G)	X	

Level	MOSFET Model Description	All Platforms including PC	All Platforms except PC
4	Grove-Frohman: LEVEL 2 model derived from SPICE 2E.3	X	
5	AMI-ASPEC depletion and enhancement (Taylor-Huang)	X	
6	Lattin-Jenkins-Grove (ASPEC style parasitics)	X	
7	Lattin-Jenkins-Grove (SPICE style parasitics)	X	
8	advanced LEVEL 2 model	X	
9 **	AMD		X
10 **	AMD		X
11	Fluke-Mosaid model		X
12 **	CASMOS model (GTE style)		X
13	BSIM model	X	
14 **	Siemens LEVEL=4		X
15	user-defined model based on LEVEL 3		X
16	not used	–	–
17	Cypress model		X
18 **	Sierra 1		X
19 ***	Dallas Semiconductor model		X
20 **	GE-CRD FRANZ		X
21 **	STC-ITT		X
22 **	CASMOS (GEC style)		X
23	Siliconix		X

Level	MOSFET Model Description	All Platforms including PC	All Platforms except PC
24 **	GE-Intersil advanced		X
25 **	CASMOS (Rutherford)		X
26 **	Sierra 2		X
27	SOSFET		X
28	BSIM derivative; Avant! proprietary model	X	
29 ***	not used	–	–
30 ***	VTI		X
31***	Motorola		X
32 ***	AMD		X
33 ***	National Semiconductor		X
34*	(EPFL) not used		X
35 **	Siemens		X
36 ***	Sharp		X
37 ***	TI		X
38	IDS: Cypress depletion model		X
39	BSIM2	X	
41	TI Analog	X	
46 ***	SGS-Thomson MOS LEVEL 3		X
47	BSIM3 Version 2.0		X
49	BSIM3 Version 3 (Enhanced)	X	
50	Philips MOS9	X	
53	BSIM3 Version 3 (Berkeley)	X	
54	UC Berkeley BSIM4 Model	X	

Level	MOSFET Model Description	All Platforms including PC	All Platforms except PC
55	EPFL-EKV Model Ver 2.6, R 11	X	
57	UC Berkeley BSIM3-SOI MOSFET Model Ver 2.0.1	X	
58	University of Florida SOI Model Ver 4.5 (Beta-98.4)	X	
59	UC Berkeley BSIM3-501 FD Model	X	
61	RPI a-Si TFT Model	X	
62	RPI Poli-Si TFT Model	X	
* not officially released ** equations are proprietary – documentation not provided *** requires a license and equations are proprietary – documentation not provided			

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.

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 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 20-1: CAPOP=4 Selections

MOS Level	Default CAPOP	CAPOP=4 selects:
2	2	11
3	2	12
13, 28, 39	13	13
others	2	11

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.

ACM=0	SPICE model, parameters determined by element areas
ACM=1	ASPEC model, parameters function of element width
ACM=2	Avant! model, combination of ACM=0,1 and provisions for lightly doped drain technology
ACM=3	Extension of ACM=2 model that deals with stacked devices (shared source/drains) and source/drain periphery capacitance along gate edge.

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.
MBYPAS	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

Parameter Units	Parameter Value
meter	multiplied by SCALM
meter^2	multiplied by SCALM^2
meter^{-1}	divided by SCALM
meter^{-2}	divided by SCALM^2

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

```
.MODEL mname [PMOS | NMOS] (<LEVEL=val> <keyname1=val1>
+ <keyname2=val2>…) <VERSION=version_number>
```

or

```
.MODEL 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

```
.MODEL MODP PMOS LEVEL=7 VTO=-3.25 GAMMA=1.0)
.MODEL MODN NMOS LEVEL=2 VTO=1.85 TOX=735e-10)
.MODEL 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

1 - 4	Drawn metal gate channel length
2 - 3	Drawn oxide cut
7 - 8	Effective channel length
6 - 9	Etched channel length
8 - 9	Lateral diffusion
5	Drawn diffusion edge
11	Actual diffusion edge

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

Figure 20-4: Isoplanar MOSFET Construction, Part A

1 - 2	Diffusion drawn dimension for nitride
4 - 7	Nitride layer width after etch
3 - 1	Periphery of the diode

The cut from the source to the drain is represented by C - D (Figure 20-5), which includes the contacts.

Figure 20-5: Isoplanar MOSFET Construction, Part B

7 - 8	Drawn channel length L
2 - 5	Actual poly width after etching L + XL where XL<0
3 - 4	Effective channel length after diffusion L + XL - LD
4 - 5	Lateral diffusion LD
9 - 10	Diffusion periphery for diode calculations
5-6	Gate edge to center contact for ACM=1 and ACM=2 calculations

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

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

Variable/ Quantity	Definition
cbd	Bulk-to-drain capacitance
cbs	Bulk-to-source capacitance
cbg	Gate-to-bulk capacitance
cgd	Gate-to-drain capacitance
cgs	Gate-to-source capacitance
f	Frequency
gbd	Bulk-to-drain dynamic conductance
gbs	Bulk-to source dynamic conductance
gds	Drain-to-source dynamic conductance controlled by vds
gdb	Drain-to bulk impact ionization conductance
gm	Drain-to-source dynamic transconductance controlled by vgs
gmbs	Drain-to-source dynamic bulk transconductance controlled by vsb
ibd	Bulk-to-drain DC current
ibs	Bulk-to-source DC current
ids	Drain-to-source DC current
idb	Drain-to-bulk impact ionization current
ind	Drain-to-source equivalent noise circuit
inrd	Drain resistor equivalent noise circuit

Table 20-3: Equation Variables and Constants

Variable/ Quantity	Definition
inrs	Source resistor equivalent noise circuit
rd	Drain resistance
rs	Source resistance
vsb	Source-to-bulk voltage
vds	Drain-to-source voltage
vgs	Gate-to-source voltage
Δt	t-tnom
εsi	1.0359e-10F/m dielectric constant of silicon
k	1.38062e-23 (Boltzmann's constant)
q	1.60212e-19 (electron charge)
t	New temperature of model or element in $^{\circ}$K
tnom	tnom = TNOM + 273.15. This variable represents the nominal temperature of parameter measurements in $^{\circ}$K (user input in $^{\circ}$C).
vt	$k \cdot t/q$
vt(tnom)	$k \cdot tnom/q$

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.

Figure 20-7: MOSFET Current Convention, N-channel

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

$$gm = \frac{\partial(ids)}{\partial(vgs)}$$

Conductance

$$gds = \frac{\partial(ids)}{\partial(vds)}$$

Bulk Transconductance

$$gmbs = \frac{\partial(ids)}{\partial(vbs)}$$

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

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

Name (Alias)	Units	Default	Description
ACM		0	Area calculation method
JS	amp/m^2	0	Bulk junction saturation current JSscaled = JS/SCALM^2 – for ACM=1, unit is amp/m and JSscaled = JS/SCALM.
JSW	amp/m	0	Sidewall bulk junction saturation current JSWscaled = JSW/SCALM.
IS	amp	1e-14	Bulk junction saturation current. For the option ASPEC=1, default=0.
N		1	Emission coefficient
NDS		1	Reverse bias slope coefficient
VNDS	V	-1	Reverse diode current transition point

Using Capacitance Model Parameters

Name (Alias)	Units	Default	Description
CBD	F	0	Zero bias bulk-drain junction capacitance. Used only when CJ and CJSW are 0.
CBS	F	0	Zero bias bulk-source junction capacitance. Used only when CJ and CJSW are 0.

Name (Alias)	Units	Default	Description
CJ (CDB, CSB, CJA)	F/m^2	579.11 $\mu F/m^2$	Zero-bias bulk junction capacitance: CJscaled = CJ/SCALM^2 For ACM=1 the unit is F/m and CJscaled = CJ/SCALM Default for option ASPEC=0 is: $$CJ = \left(\frac{\varepsilon si \cdot q \cdot NSUB}{2 \cdot PB}\right)^{1/2}$$
CJSW (CJP)	F/m	0	Zero-bias sidewall bulk junction capacitance CJSWscaled = CJSW/SCALM Default = 0
CJGATE	F/m	CSJW	Zero-bias gate-edge sidewall bulk junction capacitance (ACM=3 only) CJGATEscaled=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.
FC		0.5	Forward-bias depletion capacitance coefficient (not used)
MJ (EXA, EXJ, EXS, EXD)		0.5	Bulk junction grading coefficient
MJSW (EXP)		0.33	Bulk sidewall junction grading coefficient
NSUB (DNB, NB)	$1/cm^3$	1.0e15	Substrate doping
PB (PHA, PHS, PHD)	V	0.8	Bulk junction contact potential
PHP	V	PB	Bulk sidewall junction contact potential
TT	s	0	Transit time

Using Drain and Source Resistance Model Parameters

Name (Alias)	Units	Default	Description
RD	ohm/ sq	0.0	Drain ohmic resistance. This parameter is usually lightly doped regions' sheet resistance for ACMš1.
RDC	ohm	0.0	Additional drain resistance due to contact resistance
LRD	ohm/m	0	Drain resistance length sensitivity. Use this parameter with automatic model selection in conjunction with WRD and PRD to factor model for device size.
WRD	ohm/m	0	Drain resistance length sensitivity (used with LRD)
PRD	ohm/ m^2	0	Drain resistance product (area) sensitivity (used with LRD)
RS	ohm/ sq	0.0	Source ohmic resistance. This parameter is usually lightly doped regions' sheet resistance for ACMš1.
LRS	ohm/m	0	Source resistance length sensitivity. Use this parameter with automatic model selection in conjunction with WRS and PRS to factor model for device size.
WRS	ohm/m	0	Source resistance width sensitivity (used with LRS)
PRS	ohm/ m^2	0	Source resistance product (area) sensitivity (used with LRS)
RSC	ohm	0.0	Additional source resistance due to contact resistance
RSH (RL)	ohm/ sq	0.0	Drain and source diffusion sheet resistance

Using MOS Geometry Model Parameters

Name (Alias)	Units	Default	Description
HDIF	m	0	Length of heavily doped diffusion, from contact to lightly doped region (ACM=2, 3 only) HDIFwscaled = HDIF · SCALM
LD (DLAT,LATD)	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. LD default=0.75 · XJ. For LEVEL 4 only, lateral diffusion is derived from LD · XJ. LDscaled = LD · SCALM
LDIF	m	0	Length of lightly doped diffusion adjacent to gate (ACM=1, 2) LDIFscaled = LDIF · SCALM
WMLT		1	Width diffusion layer shrink reduction factor
XJ	m	0	Metallurgical junction depth XJscaled = XJ · SCALM
XW (WDEL, DW)	m	0	Accounts for masking and etching effects XWscaled = XW · SCALM

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=.5μm   W=10μm   L=3μm

AD	area of drain (about 80 pm^2)
AS	area of source (about 80 pm^2)
CJ	4e-4 F/m^2
CJSW	1e-10 F/m
JS	1e-8 A/m^2
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:

$$ADeff \ = \ M \cdot AD \cdot WMLT^2 \cdot SCALE^2$$

$$ASeff \ = \ M \cdot AS \cdot WMLT^2 \cdot SCALE^2$$

$$PDeff \ = \ M \cdot PD \cdot WMLT \cdot SCALE$$

$$PSeff \ = \ M \cdot PS \cdot WMLT \cdot SCALE$$

Calculating Effective Saturation Current

For ACM=0, the MOS diode effective saturation currents are calculated as:

Source Diode Saturation Current

Define:

$$val \ = \ JSscaled \cdot ASeff + JSWscaled \cdot PSeff$$

If val > 0 then,

$$isbs \ = \ val$$

Otherwise,

$$isbd \ = \ M \cdot IS$$

Drain Diode Saturation Current

Define:

$$val \ = \ JSscaled \cdot ADeff + JSWscaled \cdot PDeff$$

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 > 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 µm W=10 µm L=3 µm LDIF=0.5 µm

CJ	1e-10 F/m of gate width
	Note the change from F/m^2 (in ACM=0) to F/m.
CJSW	2e-10 F/m of gate width
JS	1e-14 A/m of gate width
	Note the change from A/m^2 (in ACM=0) to A/m
JSW	1e-13 A/m of gate width
NRD	number of squares for drain resistance
NRS	number of squares for source resistance

Calculating Effective Areas and Peripheries

For ACM=1, the effective areas and peripheries are calculated as follows:

$$ADeff \ = \ Weff \cdot WMLT$$

$$ASeff \ = \ Weff \cdot WMLT$$

$$PDff \ = \ Weff$$

$$PSeff \ = \ Weff$$

where:

$$Weff \ = \ M \cdot (Wscaled \cdot WMLT + XWscaled)$$

Note: *The Weff is not quite the same as the weff given in the models LEVEL 1,*
2, 3, 6, and 13 sections. The term $2 \cdot WDscaled$ *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:

$$val \ = \ JSscaled \cdot ASeff + JSWscaled \cdot PSeff$$

If val > 0 then,

$$isbs \ = \ val$$

Otherwise:

$$isbs \ = \ M \cdot IS$$

Drain Diode Saturation Current

Define:

$$val \ = \ JSscaled \cdot ADeff + JSWscaled \cdot PDeff$$

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,

$$RSeff \ = \ \frac{LDscaled + LDIFscaled}{Weff} \cdot RS + \frac{NRS \cdot RSH + RSC}{M}$$

If UPDATE ≥ 1 and LDIF=0 and the ASPEC option is also specified then:

$$RSeff \ = \ \frac{1}{M} \cdot (RS + NRS \cdot RSH + RSC)$$

Drain Resistance

For UPDATE=0:

$$RDeff = \frac{LDscaled + LDIFscaled}{Weff} \cdot RD + \frac{NRD \cdot RSH + RDC}{M}$$

If UPDATE ≥ 1 and LDIF=0 and the ASPEC option is also specified then:

$$RDeff = \frac{1}{M} \cdot (RD + NRD \cdot RSH + RDC)$$

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=0.07μm W=10 μm L=2 μm LDIF=1 μm HDIF= 4 μm, typical MOSFET diode parameter values are:

AD	Area of drain. Default option value for AD is not applicable.
AS	Area of source. Default option value for AS is not applicable.
CJ	1e-4 F/m^2
CJSW	1e-10 F/m
JS	1e-4 A/m^2
JSW	1e-10 A/m
HDIF	Length of heavy doped diffusion contact to gate (about 2 μm)
	HDIFeff=HDIF · WMLT · SCALM
LDIF+LD	Length of lightly doped diffusion (about 0.4μm)
NRD	Number of squares drain resistance. Default option value for NRD is not applicable.
NRS	Number of squares source resistance. Default option value for NRS is not applicable.
PD	Periphery of drain, including the gate width for ACM=2. No default.
PS	Periphery of source, including the gate width for ACM=2. No default.
RD	Resistance (ohm/square) of lightly doped drain diffusion (about 2000)
RS	Resistance (ohm/square) of lightly doped source diffusion (about 2000)
RSH	Diffusion sheet resistance (about 35)

Calculating Effective Areas and Peripheries

For ACM=2, the effective areas and peripheries are calculated as:

If AD is not specified then,

$$ADeff = 2 \cdot HDIFeff \cdot Weff$$

Otherwise,

$$ADeff = M \cdot AD \cdot WMLT^2 \cdot SCALE^2$$

If AS is not specified then,

$$ASeff = 2 \cdot HDIFscaled \cdot Weff$$

Otherwise,

$$ASeff = M \cdot AS \cdot WMLT^2 \cdot SCALE^2$$

If PD is not specified then,

$$PDeff = 4 \cdot HDIFeff + 2 \cdot Weff$$

Otherwise,

$$PDeff = M \cdot PD \cdot WMLT \cdot SCALE$$

If PS is not specified then,

$$PSeff = 4 \cdot HDIFeff + 2 \cdot Weff$$

Otherwise:

$$PSeff = M \cdot PS \cdot WMLT \cdot SCALE$$

where:

$$Weff = M \cdot (Wscaled \cdot WMLT + XWscaled)$$

$$HDIFeff = HDIFscaled$$

$$HDIFscaled = HDIF \cdot SCALM \cdot WMLT$$

Note: *The Weff is not quite the same as the Weff given in the model LEVEL 1, 2, 3, and 6 sections. The term $2 \cdot WDscaled$ is not subtracted.*

Calculating Effective Saturation Currents

For ACM=2, the MOS diode effective saturation currents are calculated as:

Source Diode Saturation Current

Define:

$$val = JSscaled \cdot ASeff + JSWscaled \cdot PSeff$$

If val > 0 then,

$$isbs = val$$

Otherwise:

$$isbs = M \cdot IS$$

Drain Diode Saturation Current

Define:

$$val = JSscaled \cdot ADeff + JSWscaled \cdot PDeff$$

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,

$$RSeff = \frac{LDscaled + LDIFscaled}{Weff} \cdot RS + \left(\frac{NRS \cdot RSH + RSC}{M} \right)$$

Otherwise:

$$RSeff = \frac{RSC}{M} + \frac{HDIFeff \cdot RSH + (LDscaled + LDIFscaled) \cdot RS}{Weff}$$

Drain Resistance

If NRD is specified then,

$$RDeff = \frac{LDscaled + LDIFscaled}{Weff} \cdot RD + \left(\frac{NRD \cdot RSH + RDC}{M} \right)$$

Otherwise:

$$RDeff = \frac{RDC}{M} + \frac{HDIFeff \cdot RSH + (LDscaled + LDIFscaled) \cdot RD}{Weff}$$

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

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,

$$ADeff = 2 \cdot HDIFeff \cdot Weff$$

For GEO=1 or 3,

$$ADeff = HDIFeff \cdot Weff$$

Otherwise:

$$ADeff = M \cdot AD \cdot WMLT^2 \cdot SCALE^2$$

If AS is not specified, then,

For GEO=0 or 1,

$$ASeff = 2 \cdot HDIFeff \cdot Weff$$

For GEO=2 or 3,

$$ASeff = HDIFeff \cdot Weff$$

Otherwise:

$$ASeff = M \cdot AS \cdot WMLT^2 \cdot SCALE^2$$

If PD is not specified, then,

For GEO=0 or 2,

$$PDeff = 4 \cdot HDIFeff + Weff$$

For GEO=1 or 3,

$$PDeff = 2 \cdot HDIFeff$$

Otherwise:

$$PDeff = M \cdot PD \cdot WMLT \cdot SCALE$$

If PS is not specified, then,

For GEO=0 or 1,

$$PSeff = 4 \cdot HDIFeff + Weff$$

For GEO=2 or 3,

$$PSeff = 2 \cdot HDIFeff$$

Otherwise:

$$PSeff = M \cdot PS \cdot WMLT \cdot SCALE$$

The Weff and HDIFeff is calculated as follows:

$$Weff = M \cdot (Wscaled \cdot WMLT + XWscaled)$$

$$HDIFeff = HDIFscaled \cdot WMLT$$

Note: *The Weff is not quite the same as the Weff given in the model LEVEL 1,*
        2, 3, and 6 sections. The term $2 \cdot WDscaled$ 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

vbs > 0,

$$ibs = isbs \cdot (e^{vbs/(N \cdot vt)} - 1)$$

vbd > 0,

$$ibd = isbd \cdot (e^{vbd/(N \cdot vt)} - 1)$$

Drain and Source Diodes Reverse Biased

For 0>vbs>VNDS,

$$ibs = gsbs \cdot vbs$$

For vbs < VNDS,

$$ibs = gsbs \cdot VNDS + \left(\frac{gsbs}{NDS}\right) \cdot (vbs - VNDS)$$

For 0 > vbd > VNDS,

$$ibd = gsbd \cdot vbd$$

For vbd < VNDS,

$$ibd = gsbd \cdot VNDS + \left(\frac{gsbd}{NDS}\right) \cdot (vbd - VNDS)$$

where

$|gsbs| = |isbs|$, and $|gdbd| = |isbd|$

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 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: C0BS, C0BD, C0BS_SW, and C0BD_SW, which depend on geometric parameters, such as ASeff and PSeff calculated under various ACM specifications.

When ACM=3, the intermediate quantities C0BS_SW, and C0BD_SW include an extra term to account for CJGATE.

For ACM=2, the parameter CJGATE has been added in a backward compatible manner. Therefore, the default behavior of CJGATE makes the intermediate quantities C0BS_SW and C0BD_SW the same as for previous versions. The default patterns are:

- If neither CJSW nor CJGATE is specified, both default to zero.
- If CJGATE is not specified, it defaults to CJSW, which in turn defaults to zero.
- If CJGATE is specified, and CJSW is not specified, then CJSW defaults to zero.

The intermediate quantities C0BS, C0BS_SW, C0BD, and C0BD_SW are calculated as follows.

```
C0BS = CJscaled*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,

$$capbs = TT \cdot \frac{\partial ibs}{\partial vbs} + C0BS \cdot \left(1 - \frac{vbs}{PB} \right)^{-MJ}$$

$$+ \text{C0BS\_SW} \cdot \left(1 - \frac{vbs}{PHP} \right)^{-MJSW}$$

For vbs > 0,

$$capbs = TT \cdot \frac{\partial ibs}{\partial vbs} + C0BS \cdot \left(1 + MJ \cdot \frac{vbs}{PB}\right)$$
$$+ C0BS\_SW \cdot \left(1 + MJSW \cdot \frac{vbs}{PHP}\right)$$

Otherwise, if (C0BS + C0BS_SW) ≤ 0, then:

For vbs < 0,

$$capbs = TT \cdot \frac{\partial ibs}{\partial vbs} + M \cdot CBS \cdot \left(1 - \frac{vbs}{PB}\right)^{-MJ}$$

For vbs > 0,

$$capbs = TT \cdot \frac{\partial ibs}{\partial vbs} + M \cdot CBS \cdot \left(1 + MJ \cdot \frac{vbs}{PB}\right)$$

Drain Diode Capacitance

If (C0BD + C0BD_SW) > 0, then:

For vbd < 0,

$$capbd = TT \cdot \frac{\partial ibd}{\partial vbd} + C0BD \cdot \left(1 - \frac{vbd}{PB}\right)^{-MJ}$$
$$+ PDeff \cdot C0BD\_SW \cdot \left(1 - \frac{vbd}{PHP}\right)^{-MJSW}$$

For vbd > 0,

$$capbd = TT \cdot \frac{\partial ibd}{\partial vbd} + C0BD \cdot \left(1 + MJ \cdot \frac{vbd}{PB}\right)$$
$$+ C0BD\_SW \cdot \left(1 + MJSW \cdot \frac{vbd}{PHP}\right)$$

Otherwise, if (ADeff · CJscaled + PDeff · CJSWscaled) ≤ 0, then:

For vbd < 0,

$$capbd = TT \cdot \frac{\partial ibd}{\partial vbd} + M \cdot CBD \cdot \left(1 - \frac{vbd}{PB}\right)^{-MJ}$$

For vbd > 0,

$$capbd = TT \cdot \frac{\partial ibd}{\partial vbd} + M \cdot CBD \cdot \left(1 + MJ \cdot \frac{vbd}{PB}\right)$$

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.

Name (Alias)	Units	Default	Description
DELVTO	V	0.0	Zero-bias threshold voltage shift
GAMMA	$V^{1/2}$	0.527625	Body effect factor. If GAMMA is not set, it is calculated from NSUB.
NGATE	$1/cm^3$		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.
NSS	$1/cm^2$	1.0	Surface state density
NSUB (DNB, NB)	$1/cm^3$	1e15	Substrate doping
PHI	V	0.576036	Surface potential. NSUB default=1e15.
TPG (TPS)		1.0	Type of gate material, used for analytical model only LEVEL 4 TPG default=0 where TPG = 0 al-gate TPG = 1 gate type same as source-drain diffusion TPG = -1 fate type opposite to source-drain diffusion
VTO (VT)	V		Zero-bias threshold voltage

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,

$$GAMMA = \frac{(2 \cdot q \cdot \varepsilon si \cdot NSUB)^{1/2}}{COX}$$

The energy gap, eg, and intrinsic carrier concentration for the above equations are determined by:

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

$$ni = 1.45e\text{+}10 \cdot \left(\frac{tnom}{300}\right)^{3/2} \cdot e^{\left[\frac{q \cdot eg}{2 \cdot k} \cdot \left(\frac{1}{300} - \frac{1}{tnom}\right)\right]} (1/\text{cm}^3)$$

where,

$$tnom = TNOM + 273.15$$

If VTO is not specified, then for Al-Gate (TPG=0), the work function Φms 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:

$$VTO \;=\; vfb + type \cdot (GAMMA \cdot PHI^{1/2} + PHI)$$

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

Name (Alias)	Units	Default	Description
ALPHA	1/V	0.0	Impact ionization current coefficient
LALPHA	µm/V	0.0	ALPHA length sensitivity
WALPHA	µm/V	0.0	ALPHA width sensitivity
VCR	V	0.0	Critical voltage
LVCR	µm · V	0.0	VCR length sensitivity
WVCR	µm · V	0.0	VCR width sensitivity
IIRAT		0.0	Portion of impact ionization current that goes to source

Calculating the Impact Ionization Equations

The current I_impact due to impact ionization effect is calculated as follows:

$$\text{I\_impact} = Ids \cdot ALPHAeff \cdot (vds - vdsat) \cdot e^{\frac{-VCReff}{vds - vdsat}}$$

where:

$$ALPHAeff = ALPHA + LALPHA \cdot 1e\text{-}6 \cdot \left(\frac{1}{Leff} - \frac{1}{LREFeff} \right)$$

$$+ WALPHA \cdot 1e\text{-}6 \cdot \left(\frac{1}{Weff} - \frac{1}{WREFeff} \right)$$

$$VCReff = VCR + LVCR \cdot 1e\text{-}6 \cdot \left(\frac{1}{Leff} - \frac{1}{LREFeff} \right)$$

$$+ WVCR \cdot 1e\text{-}6 \cdot \left(\frac{1}{Weff} - \frac{1}{WREFeff} \right)$$

where:

$$LREFeff = LREF + XLREF - 2 \cdot LD \text{ and}$$

$$WREFeff = WREF + XWREF - 2 \cdot WD$$

Calculating Effective Output Conductance

The element template output allows *gds* to be output directly, for example:
```
.PRINT I(M1) gds=LX8(M1)
```

When using impact ionization current, it is important to note that gds 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))')
```

Figure 20-15: Drain, Source, and Bulk Currents for vgs=3, with IIRAT=0.5

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

```
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
```

Input File

```
$ cascode test
.param pvds=5.0 pvref=1.4 pvin=3.0

vdd dd 0 pvds ac 1
$ 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
```

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

MOS Level	Default CAPOP	CAPOP=4 selects
2	2	11
3	2	12
13, 28, 39	13	13
Other levels	2	11

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), ... 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:

```
dQi/dVj need not equal dQj/dVi
```

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.

$$\frac{dQ1}{dV1} + \frac{dQ2}{dV1} = 0 \qquad\qquad \frac{dQ1}{dV1} + \frac{dQ1}{dV2} = 0$$

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

```
Cii = dQi/dVi      i=1,.4
```

and the transcapacitances are the negative of off-diagonal entries

```
Cij = -dQi/dVj     i not equal to j
```

All of the Cs are normally positive.

Figure 20-17: MOS Capacitances

In Figure 20-17, Cij 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:

$$CGG = \frac{dQg}{dVG}$$

$$CGD = -\frac{dQg}{dVD}$$

$$CDG = -\frac{dQD}{dVG}$$

CGG represents input capacitance: a change in gate voltage requires a current equal to CGGxdVG/dt into the gate terminal.

CGD represents Miller feedback: a change in drain voltage gives a current equal to CGGxdVG/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:

$$QG = Q(vgs)$$

$$QS = -0.5 \cdot Q(vgs)$$

$$QD = -0.5 \cdot Q(vgs)$$

$$QB = 0$$

Consequently:

$$CGD = -\frac{dQG}{dVD} = 0$$

$$CGD = -\frac{dQD}{dVG} = 0.5 \cdot \frac{dQ}{dvgs}$$

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

cbtot   dQB/dVB

cgs   -dQG/dVS

cgd   -dQG/dVD

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:

cdtot   cgd+cdb

cgtot   cgs+cgd+cgb

cstot   cgs+csb

cbtot   cgb+csb+cdb

cgs   cgs

cgd   cgd

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:

```
LX18(m) = dQG/dVGB = CGGBO
LX19(m) = dQG/dVDB = CGDBO
LX20(m) = dQG/dVSB = CGSBO
LX21(m) = dQB/dVGB = CBGBO
LX22(m) = dQB/dVDB = CBDBO
LX23(m) = dQB/dVSB = CBSBO
LX32(m) = dQD/dVG  = CDGBO
LX33(m) = dQD/dVD  = CDDBO
```

```
   LX34(m) = dQD/dVS = CDSBO
```

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 = dQS/dVS = d(-QG-QB-QD)/dVS = -
LX20-LX23-LX34
CGD(reverse) = CGS = -LX20
CDG(reverse) = CSG = -dQS/dVG = d(QG+QB+QD)/dVG =
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:

$$\frac{eox}{tox} = 1e - 3 F/m^2$$

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

```
$
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

$$Leff = 0.6u$$

$$\frac{eox}{tox} = 1e - 3 F/m^2$$

$$Cap = \frac{Leff \cdot Weff \cdot eox}{tox} = 60e - 15F$$

BSIM equations for internal capacitance in saturation with xqc=0.4:

$$body = 1 + 0.5 \cdot \left(1 - \frac{1}{(1.744 + 0.8364 \cdot (PHI0 + vsb))}\right) \cdot \frac{K1}{\sqrt{(PHI0 + vsb)}}$$

$$1 + 0.5 \cdot \left(1 - \frac{1}{(1.744 + 0.8364)}\right) = 1.3062$$

$$cgg = Cap \cdot \left(1 - \frac{1}{(3 \cdot body)}\right) = Cap \cdot 0.7448 = 44.69F$$

$$cgd = 0$$

$$cdg = \left(\frac{4}{15}\right) \cdot Cap = 16F$$

$$cdd = 0$$

Gate-drain overlap $= (ld + meto) \cdot Weff \cdot \dfrac{eox}{tox} = 20e - 15F$

Adding the overlaps,

$$cgg = 44.69F + 2 \cdot 20F = 84.69F$$

$$cgd = 20F$$

$$cdg = 36F$$

$$cdd = 20F$$

Drain-bulk diode cap $cj \cdot ad = (0.5e - 4) \cdot (200e - 12) = 1$

Adding the diodes,

$$cgg = 84.69F$$

$$cgd = 20F$$

$$cdg = 36F$$

$$cdd = 30F$$

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
cgd        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

```
$ 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
```

Figure 20-18: Gate Capacitance

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

Name (Alias)	Units	Default	Description
CAPOP		2.0	Capacitance model selector
COX (CO)	F/m^2	3.453e-4	Oxide capacitance. If COX is not input, it is calculated from TOX. The default value corresponds to the TOX default of 1e-7: COXscaled = $COX/SCALM^2$
TOX	m	1e-7	Represents the oxide thickness, calculated from COX when COX is input. The program uses default if COX is not specified. For TOX>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

Using Gate Overlap Capacitance Model Parameters

Name (Alias)	Units	Default	Description
CGBO (CGB)	F/m	0.0	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
CGDO (CGD, C2)	F/m	0.0	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
CGSO *(CGS, C1)*	F/m	0.0	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
LD (LATD, DLAT)	m		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 · XJ for all levels except LEVEL 4, for which LD default=0.75. LDscaled = LD · SCALM LEVEL 4: LDscaled = LD · XJ · SCALM
METO	m	0.0	Fringing field factor for gate-to-source and gate-to-drain overlap capacitance calculation METOscaled = METO · SCALM
WD	m	0.0	Lateral diffusion into channel from bulk along width WDscaled = WD · SCALM

Using Meyer Capacitance Parameters CAPOP=0, 1, 2

Name (Alias)	Units	Default	Description
CF1	V	0.0	Modified MEYER control for transition of cgs from depletion to weak inversion for CGSO (only for CAPOP=2)
CF2	V	0.1	Modified MEYER control for transition of cgs from weak to strong inversion region (only for CAPOP=2)
CF3		1.0	Modified MEYER control for transition of cgs and cgd from saturation to linear region as a function of vds (only for CAPOP=2)
CF4		50.0	Modified MEYER control for contour of cgb and cgs smoothing factors
CF5		0.667	Modified MEYER control for capacitance multiplier for cgs in saturation region
CF6		500.0	Modified MEYER control for contour of cgd smoothing factor
CGBEX		0.5	cgb exponent (only for CAPOP=1)

Using Charge Conservation Parameters (CAPOP=4)

Name (Alias)	Units	Default	Description
XQC		0.5	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.

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 $\rightarrow$ 40/60
    - XPART=0.4 $\rightarrow$ 40/60
    - XPART=0.5 $\rightarrow$ 50/50
    - XPART=1 $\rightarrow$ 0/100
    - XPART = any other value less than 1 $\rightarrow$ 40/60
    - XPART >1 $\rightarrow$ 0/100If XQC is specified:
- XQC=0 $\rightarrow$ 0/100
    - XQC=0.4 $\rightarrow$ 40/60
    - XQC=0.5 $\rightarrow$ 50/50
    - XQC=1 $\rightarrow$ 0/100
    - XQC = any other value less than 1 $\rightarrow$ 40/60
    - XQC>1 $\rightarrow$ 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

$$CGBOeff = M \cdot Leff \cdot CGBOscaled$$

Otherwise,

$$CGBOeff = 2 \cdot WDscaled \cdot Leff \cdot COXscaled \cdot M$$

Gate-to-Source Overlap Capacitance

If CGSO is specified, then

$$CGSOeff = Weff \cdot CGSOscaled$$

Otherwise,

$$CGSOeff = Weff \cdot (LDscaled + METOscaled) \cdot COXscaled$$

Gate-to-Drain Overlap Capacitance

If CGDO is specified, then

$$CGDOeff = Weff \cdot CGDOscaled$$

Otherwise,

$$CGDOeff = Weff \cdot (LDscaled + METOscaled) \cdot COXscaled$$

The Leff is calculated for each model differently, and it is given in the corresponding model section. The Weff calculation is not quite the same as weff given in the model LEVEL 1, 2, 3, 6, 7 and 13 sections.

$$Weff \ = \ M \cdot (Wscaled \cdot WMLT + XWscaled)$$

The 2·WDscaled factor is not subtracted.

Defining CAPOP=0 — SPICE Meyer Gate Capacitances

Definition:

$$cap \ = \ COXscaled \cdot Weff \cdot Leff$$

Gate-Bulk Capacitance (cgb)

Accumulation, vgs ≤ vth-PH1

$$cgb \ = \ cap$$

Depletion, vgs < vth

$$cgb \ = \ cap \cdot \frac{vth - vgs}{PH1}$$

Strong Inversion, vgs ≥ vth

$$cgb \ = \ 0$$

Gate-Source Capacitance (cgs)

Accumulation, $vgs \le vth - \dfrac{PHI}{2}$

$$cgs \ = \ 0$$

Depletion, vgs ≤ vth

$$cgs \ = \ CF5 \cdot cap + \frac{cap \cdot (vgs - cth)}{0.75 \cdot PHI}$$

Strong Inversion Saturation Region, vgs > vth and vds ≥ vdsat

$$cgs \, = \, CF5 \cdot cap$$

Strong Inversion Linear Region, vgs > vth and vds < vdsat

$$cgs \, = \, CF5 \cdot cap \cdot \left\{ 1 - \left[\frac{vdsat - vds}{2 \cdot (vdsat + vsb) - vds - vsb} \right]^2 \right\}$$

Gate-Drain Capacitance (cgd)

The gate-drain capacitance has value only in the linear region.

Strong Inversion Linear Region, vgs > vth and vds < vdsat.

$$cgd \, = \, CF5 \cdot cap \cdot \left\{ 1 - \left[\frac{vdsat + vsb}{2 \cdot (vdsat + vsb) - vds - vsb} \right]^2 \right\}$$

Example

```
*file capop0.sp---capop=0 capacitances
*
*this file is used to create spice meyer gate c-v plots
**
*(capop=0) for low vds and high vds
*
.options acct=2 post=2 dccap=1 nomod
.dc vg1 -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)')
*****************************************
m1 d1 g1 0 0 mn l=5e-6 w=20e-6 $ create capacitances for
+ vds=0.05
m2 d2 g1 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

$$cap = COXscaled \cdot Weff \cdot Leff$$

In the following equations, G^-, G^+, D^-, and D^+ are smooth factors. They are not user-defined parameters.

Gate-Bulk Capacitance (cgb)

Accumulation, vgs ≤ vfb – vsb

$$cgb = cap$$

Depletion, vgs ≤ vth

$$cgb = \frac{cap}{\left[1 + 4 \cdot \dfrac{vgs + vsb - vfb}{GAMMA^2}\right]^{CGBEX}}$$

Strong Inversion, vgs > vth

$$cgb = \frac{G^+ \cdot cap}{\left[1 + 4 \cdot \dfrac{GAMMA \cdot (vsb + PHI)^2 + vsb + PHI}{GAMMA^2}\right]^{CGBEX}}$$

Note: *In the above equations, GAMMA is replaced by effective* γ *for model level higher than 4.*

Gate-Source Capacitance (cgs)

Low vds (vds < 0.1)

Accumulation, vgs ≤ vth

$$cgs = CF5 \cdot cap \cdot G^- \cdot D^-$$

Weak Inversion, vgs < vth + 0.1

$$cgs = CF5 \cdot cap \cdot \left\{ \frac{vgs - vth}{0.1} \cdot \left[1 - \left(\frac{0.1 - vds}{0.2 - vds} \right)^2 - D^- \right] + D^- \right\}$$

Strong Inversion, vgs ≥ vth + 0.1

$$cgs = CF5 \cdot cap \cdot \left\{ 1 - \left[\frac{vgs - vth - vds}{2 \cdot (vgs - vth) - vds} \right]^2 \right\}$$

High vds (vds ≥ *0.1*)

Accumulation, vgs ≤vth

$$cgs = CF5 \cdot cap \cdot G^-$$

Saturation Region, vgs < vth + vds

$$cgs = CF5 \cdot cap$$

Linear Region, vgs ≥ vth + vds

$$cgs = CF5 \cdot cap \cdot \left\{ 1 - \left[\frac{vgs - vth - vds}{2 \cdot (vgs - vth) - vds} \right]^2 \right\}$$

Gate-Drain Capacitance (cgd)

Low vds (vds < 0.1)

Accumulation, vgs ≤ vth

$$cgd = CF5 \cdot cap \cdot G^- \cdot D^+$$

Weak Inversion, vgs < vth + 0.1

$$cgd = CF5 \cdot cap \cdot \left\{ D^+ + \frac{vgs - vgh}{0.1} \cdot \max\left[0, 1 - \left(\frac{0.1}{0.2 - vds}\right)^2 - D^+\right]\right\}$$

Strong Inversion, vgs ≥ vth + 0.1

$$cgd = CF5 \cdot cap \cdot \max\left\{ D^+, 1 - \left[\frac{vgs - vth}{2 \cdot (vgs - vth) - vds}\right]^2\right\}$$

High vds(vds ≥ *0.1*)

Accumulation, vgs ≤ vth

$$cgd = CF5 \cdot cap \cdot G^- \cdot D^+$$

Saturation Region, vgs < vth + vds

$$cgd = CF5 \cdot cap \cdot D^+$$

Strong Inversion, vgs ≥ vth + vds

$$cgd = CF5 \cdot cap \cdot \max\left\{ D^+, 1 - \left[\frac{vgs - vth}{2 \cdot (vgs - vth) - vds}\right]^2\right\}$$

Example
```
*file capop1.sp---capop1 capacitances
*
*this file creates the modified meyer gate c-v plots
*(capop=1) for low vds and high vds.
*
.options acct=2 post=2 dccap=1 nomod
.dc vg1 -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)')
*****************************************
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
*****************************************
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
+ 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
```

Figure 20-20: CAPOP=1 Capacitances

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

$$cap = COXscaled \cdot Weff \cdot Leff$$

Gate-Bulk Capacitance (cgb)

Accumulation, vgs ≤ vfb - vsb

$$cgb = cap$$

Depletion, vgs ≤ vth

$$cgs = \frac{cap}{\left(1 + 4 \cdot \dfrac{vgs + vsb - vfb}{GAMMA^2}\right)^{1/2}}$$

Inversion, vgs > vth

$$cgb = \frac{G^+ \cdot cap}{\left[1 + 4 \cdot \dfrac{GAMMA \cdot (PHI + vsb)^{1/2} + PHI + vsb}{GAMMA^2}\right]^{1/2}}$$

Note: *In the above equations, GAMMA is replaced by effective γ for model level higher than 4.*

Gate-Source Capacitance (cgs)

Low vds (vds < 0.1)

Accumulation, vgs < vth – CF1

$$cgs = CF5 \cdot cap \cdot G^- \cdot D^-$$

Depletion, vgs ≤ vth + CF2 – CF1

$$cgs = CF5 \cdot cap \cdot \left\{ \frac{vgs - vth + CF1}{CF2} \cdot \left[1 - \left(CF2 - \frac{vds}{2 \cdot CF2 - vds}\right)^2 - D^-\right] + D^- \right\}$$

Strong Inversion, vgs > vth + max (CF2 – CF1, CF3 · vds) UPDATE=0

Strong Inversion, vgs > vth + CF2 – CF1,UPDATE=1

$$cgs = CF5 \cdot cap \cdot \left\{ 1 - \left[\frac{vgs - vth + CF1 - vds}{2 \cdot (vgs - vth + CF1) - vds}\right]^2 \right\}$$

High vds (vds ≥ *0.1)*

Accumulation, vgs < vth – CF1

$$cgs \ = \ CF5 \cdot cap \cdot G^- \cdot D^+, \quad CF1 \neq 0$$

$$cgs \ = \ CF5 \cdot cap \cdot G^-, \qquad CF1 \ = \ 0$$

Weak Inversion, vgs < vth + CF2 – CF1, CF1 ≠ 0

$$cgs \ = \ CF5 \cdot cap \cdot \max\left(\frac{vgs - vth + CF1}{CF2}, D^+\right)$$

Saturation Region, vgs < vth + CF3 · vds

$$cgs \ = \ CF5 \cdot cap$$

Linear Region, vgs > vth + CF3 · vds

$$cgs \ = \ CF5 \cdot cap \cdot \left\{1 - \left[\frac{vgs - vth - vds}{2 \cdot (vgs - vth) - vds}\right]^2\right\}, \quad UPDATE=0, \quad CF1=0$$

$$cgs \ = \ CG5 \cdot cap \cdot \left\{1 - \left[\frac{vgs - vth - CF3 \cdot vds}{2 \cdot (vgs - vth) - CF3 \cdot vds}\right]^2\right\}, \quad UPDATE=1$$

Gate-Drain Capacitance (cgd)

Low vds, (vds < 0.1)

Accumulation, vgs ≤ vth – CF1

$$cgd \ = \ CF5 \cdot cap \cdot G^- \cdot D^-$$

Weak Inversion, vgs < vth + CF2 – CF1

$$cgd \ = \ CF5 \cdot cap \cdot \left\{D^- + \frac{vgs - vth + CF1}{CF2} \cdot \max\left[0, 1 - \left(\frac{CF2}{2 \cdot CF2 - vds}\right)^2 - D^-\right]\right\}$$

Strong Inversion, vgs ≥ vth + CF2 – CF1

$$cgd = CF5 \cdot cap \cdot \max\left\{ D^-,\, 1 - \left[\frac{vgs - vth + CF1}{2 \cdot (vgs - vth + CF1) - vds}\right]^2 \right\}$$

High vds (vds > 0.1)

Accumulation, vgs ≤ vth – CF1

$$cgd = CF5 \cdot cap \cdot G^- \cdot DD^+$$

Saturation Region, vgs ≤ vth + CF3 · vds

$$cgd = CF5 \cdot cap \cdot DD^+$$

Note: In the above equation, DD^+ is a function of CF3, if UPDATE=1.

Linear Region, vgs > vth + CF3 · vds

$$cgd = CF5 \cdot cap \cdot \max\left\{ DD^+,\, 1 - \left[\frac{vgs - vth}{2 \cdot (vgs - vth) - CF3 \cdot vds}\right]^2 \right\}$$

Example

```
*file capop2.sp capop=2 capacitances
*
*this file creates parameterized modified gate capacitances
*(capop=2) for low and high vds.
*
.options acct=2 post=2 dccap=1 nomod
.dc vg1 -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)')
*****************************************
```

```
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
*****************************************
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
+ tox = 9.77e-8 uo = 817 ucrit = 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.
* CDGB = LX32(M) DERIVATIVE OF QD WITH RESPECT TO VGB.
* CDDB = LX33(M) DERIVATIVE OF QD WITH RESPECT TO VDB.
* CDSB = LX34(M) DERIVATIVE OF QD WITH RESPECT TO VSB.
* THE SIX NONRECIPROCAL 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
+ CGSO=0 CGDO=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

```
*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 5NS 50NS 100NS
VBB 0 B PULSE 0 5 0NS 5NS 5NS 50NS 100NS
VDD D D- PULSE 0 5 25NS 5NS 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
*
.MODEL MM 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
```

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
.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
```

Figure 20-26: Charge Conservation Test Circuit

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:

$$vgst = vgs - \frac{(vth + vfb)}{2}$$

$$cox = \frac{\varepsilon ox}{TOX \cdot 1e\text{-}10} \cdot Weff \cdot Leff$$

The gate capacitance *cgs* is calculated according to the equations below in the different regions.

$0.5 \cdot (vth + vfb) > vgs$

$$cgs = 0$$

$0.5 \cdot (vth + vfb) < vgs < vth$

For vgst < vds,

$$cgs = \frac{4}{3} \cdot \frac{cox \cdot vgst}{vth - vfb}$$

For vgst > vds,

$$cgs = arg \cdot \frac{4}{3} \cdot \frac{cox \cdot vgst}{vth - vfb}$$

vgs > vth

For vgst < vds,

$$cgs = \frac{2}{3} \cdot cox$$

For vgst > vds,

$$cgs = arg \cdot \frac{2}{3} \cdot cox$$

$$arg = vgst \cdot \frac{(3 \cdot vgst - 2 \cdot vds)}{(2 \cdot vgst - vds)^2}$$

The gate capacitance *cgd* is calculated according to the equations below in the different regions.

vgs < vth

$$cgd = 0$$

vgs > vth and vgst < vds

$$cgd = 0$$

vgs > vth and vgst > vds

$$cgd = arg \cdot \frac{2}{3} \cdot cox$$

$$arg = (3 \cdot vgst - vds) \cdot \frac{(vgst - vds)}{(2 \cdot vgst - vds)^2}$$

The gate capacitance cgb is combined with the calculation of both oxide capacitance and depletion capacitance as shown below.

$$cgb = \frac{cgbx \cdot cd}{cgbx + cd}$$

Oxide capacitance cgbx, is calculated as:

$$cgbx = cox - cgs - cgd$$

Depletion capacitance cd is voltage-dependent.

$$cd = \frac{\varepsilon si}{wd} \cdot Weff \cdot Leff$$

$$wd = \left(\frac{2 \cdot \varepsilon si \cdot vc}{q \cdot NSUB} \right)^{1/2}$$

$$vc = \textit{The effective voltage from channel to substrate (bulk)}$$

The following shows the equations for vc under various conditions:

vgs + vsb < vfb

$$vc = 0$$

vgs + vsb > vfb

$$vc = vgs + vsb - vfb$$

vgst > 0, vgs < vth, vgst < vds

$$vc = \frac{1}{2} \cdot (vth - vfb) + \frac{3}{2} \cdot vgst + vsb$$

vgst > 0, vgs < vth, vgst > vds

$$vc \ = \ \frac{1}{2} \cdot (vth - vfb) + vgst + \frac{1}{2} \cdot vds + vsb$$

vgs > vth, vgst < vds

$$vc \ = \ vth - vfb + \frac{1}{2} \cdot vgst + vsb$$

vgs > vth, vgst > vds

$$vc \ = \ vth - vfb + \frac{1}{2} \cdot vds + vsb$$

Defining CAPOP=13 — BSIM1-based Charge-Conserving Gate Capacitance Model

See "LEVEL 13 BSIM Model" on page 21-101.

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.

$$Leff \ = \ L + XL - 2 \cdot LD$$

$$Weff \ = \ W + XW - 2 \cdot WD$$

and uses the following equations for AC gate capacitance calculations

$$Leff \ = \ L + XL - 2 \cdot LDAC$$

$$Weff \ = \ W + XW - 2 \cdot WDAC$$

The noise calculations use the DC Weff and Leff 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

Name (Alias)	Units	Default	Description
AF		1.0	Flicker noise exponent
KF		0.0	Flicker noise coefficient. Reasonable values for KF are in the range 1e-19 to 1e-25 V^2F.
NLEV		2.0	Noise equation selector
GDSNOI		1.0	Channel thermal noise coefficient (use with NLEV=3)

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 . 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:

$$ind^2 = (channel\ thermal\ noise)^2 + (flicker\ noise)^2$$

If the model parameter NLEV is less than 3, then

$$channel\ thermal\ noise = \left(\frac{8kT \cdot gm}{3}\right)^{1/2}$$

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 *Tsivids, Yanis P., Operation and Modeling of the MOS Transistor, McGraw-Hill, 1987, p. 340.*

For NLEV=3,

$$channel\ thermal\ noise = \left(\frac{8kt}{3} \cdot \beta \cdot (vgs - vth) \cdot \frac{1 + a + a^2}{1 + a} \cdot GDSNOI\right)^{1/2}$$

where

$$a = 1 - \frac{vds}{vdsat} \qquad Linear\ region$$

$$a = 0 \qquad\qquad Saturation\ region$$

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):

$$flicker\ noise = \left(\frac{KF \cdot Ids^{AF}}{COX \cdot Leff^2 \cdot f}\right)^{1/2}$$

For NLEV=1 the Leff^2 in the above equation is replaced by Weff · Leff.

NLEV=2, 3:

$$flicker\ noise\ =\ \left(\frac{KF \cdot gm^2}{COX \cdot Weff \cdot Leff \cdot f^{AF}}\right)^{1/2}$$

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: $ID = RX^2 \cdot (channel\ thermal\ noise)^2$
FN, V^2/Hz	Output flicker noise: $FN = RX^2 \cdot (flicker\ noise)^2$
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

Name (Alias)	Units	Default	Description
BEX		-1.5	Low field mobility, UO, temperature exponent
CTA	1/°K	0.0	Junction capacitance CJ temperature coefficient. Set TLEVC to 1 to enable CTA to override default Star-Hspice temperature compensation.
CTP	1/°K	0.0	Junction sidewall capacitance CJSW temperature coefficient. Set TLEVC to 1 to enable CTP to override default Star-Hspice temperature compensation.
EG	eV		Energy gap for pn junction diode. Set default=1.11, for TLEV=0 or 1 and default=1.16, for TLEV=2. 1.17 – silicon 0.69 – Schottky barrier diode 0.67 – germanium 1.52 – gallium arsenide
F1EX		0	Bulk junction bottom grading coefficient

Name (Alias)	Units	Default	Description
GAP1	eV/°K	7.02e-4	First bandgap correction factor (from Sze, alpha term) 7.02e-4 – silicon 4.73e-4 – silicon 4.56e-4 – germanium 5.41e-4 – gallium arsenide
GAP2	°K	1108	Second bandgap correction factor (from Sze, beta term) 1108 – silicon 636 – silicon 210 – germanium 204 – gallium arsenide
LAMEX	1/°K	0	LAMBDA temperature coefficient
N		1.0	Emission coefficient
MJ		0.5	Bulk junction bottom grading coefficient
MJSW		0.33	Bulk junction sidewall grading coefficient
PTA	V/°K	0.0	Junction potential PB temperature coefficient. Set TLEVC to 1 or 2 to enable PTA to override default Star-Hspice temperature compensation.
PTC	V/°K	0.0	Fermi potential PHI temperature coefficient. Set TLEVC to 1 or 2 to enable PTC to override default Star-Hspice temperature compensation.
PTP	V/°K	0.0	Junction potential PHP temperature coefficient. Set TLEVC to 1 or 2 to enable PTP to override default Star-Hspice temperature compensation.
TCV	V/°K	0.0	Threshold voltage temperature coefficient. Typical values are +1mV for n-channel and -1mV for p-channel.

Name (Alias)	Units	Default	Description
TLEV		0.0	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.
TLEVC		0.0	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.
TRD	1/$^{\circ}$K	0.0	Temperature coefficient for drain resistor
TRS	1/$^{\circ}$K	0.0	Temperature coefficient for source resistor
XTI		0.0	Saturation current temperature exponent. Use XTI=3 for silicon diffused junction. Set XTI=2 for Schottky barrier diode.

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.

Parameter	Description	Sensitivity Parameters		
		Length	Width	Product
BEX	Low field mobility, UO, temperature exponent	LBEX	WBEX	PBEX
FEX	Velocity saturation temperature exponent	LFEX	WFEX	PFEX
TCV	Threshold voltage temperature coefficient	LTCV	WTCV	PTCV
TRS	Temperature coefficient for source resistor	LTRS	WTRS	PTRS
TRD	Temperature coefficient for drain resistor	LTRD	WTRD	PTRD

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:

TLEV = 0 or 1:

$$egnom \ = \ 1.16 - 7.02e{-}4 \cdot \frac{tnom^2}{tnom + 1108.0}$$

$$eg(t) \ = \ 1.16 - 7.02e{-}4 \cdot \frac{t^2}{t + 1108.0}$$

TLEV = 2:

$$egnom = EG - GAP1 \cdot \frac{tnom^2}{tnom + GAP2}$$

$$eg(t) = EG - GAP1 \cdot \frac{t^2}{t + GAP2}$$

Calculating Saturation Current Temperature Equations

$$isbd(t) = isbd(tnom) \cdot e^{facln/N}$$

$$isbs(t) = isbs(tnom) \cdot e^{facln/N}$$

where

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

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:

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

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

$$CBD(t) = CBD \cdot \left[1 + MJ \cdot \left(400u \cdot \Delta t - \frac{PB(t)}{PB} + 1\right)\right]$$

$$CBS(t) \ = \ CBS \cdot \left[1 + MJ \cdot \left(400u \cdot \Delta t - \frac{PB(t)}{PB} + 1 \right) \right]$$

$$CJ(t) \ = \ CJ \cdot \left[1 + MJ \cdot \left(400u \cdot \Delta t - \frac{PB(t)}{PB} + 1 \right) \right]$$

$$CJSW(t) \ = \ CJSW \cdot \left[1 + MJSW \cdot \left(400u \cdot \Delta t - \frac{PHP(t)}{PHP} + 1 \right) \right]$$

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:

$$PB(t) \ = \ PB - PTA \cdot \Delta t$$

$$PHP(t) \ = \ PHP - PTP \cdot \Delta t$$

$$CBD(t) \ = \ CBD \cdot \left(\frac{PB}{PB(t)} \right)^{MJ}$$

$$CBS(t) \ = \ CBS \cdot \left(\frac{PB}{PB(t)} \right)^{MJ}$$

$$CJ(t) = CJ \cdot \left(\frac{PB}{PB(t)}\right)^{MJ}$$

$$CJSW(t) = CJSW \cdot \left(\frac{PHP}{PHP(t)}\right)^{MJSW}$$

TLEVC=3:

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

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

$$CBD(t) = CBD \cdot \left(1 - 0.5 \cdot dpbdt \cdot \frac{\Delta t}{PB}\right)$$

$$CBS(t) = CBS \cdot \left(1 - 0.5 \cdot dpbdt \cdot \frac{\Delta t}{PB}\right)$$

$$CJ(t) = CJ \cdot \left(1 - 0.5 \cdot dpbdt \cdot \frac{\Delta t}{PB}\right)$$

$$CJSW(t) = CJSW \cdot \left(1 - 0.5 \cdot dphpdt \cdot \frac{\Delta t}{PHP}\right)$$

where for TLEV=0 or 1:

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

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

TLEV=2:

$$dpbdt = -\frac{\left[egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left(2 - \frac{tnom}{tnom + GAP2} \right) - PB \right]}{tnom}$$

$$dphpdt = -\frac{\left[egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left(2 - \frac{tnom}{tnom + GAP2} \right) - PHP \right]}{tnom}$$

Calculating Surface Potential Temperature Equations

TLEVC=0:

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

TLEVC=1:

$$PHI(t) = PHI - PTC \cdot \Delta t$$

If the PHI parameter is not specified, it is calculated as:

$$PHI(t) = 2 \cdot vt(t) \cdot \ln\left(\frac{NSUB}{ni} \right)$$

The intrinsic carrier concentration, ni, must be temperature updated, and it is calculated from the silicon bandgap at room temperature.

$$ni = 145e16 \cdot \left(\frac{t}{tnom} \right)^{3/2} \cdot \exp\left[EG \cdot \left(\frac{t}{tnom} - 1 \right) \cdot \left(\frac{1}{2 \cdot vt(t)} \right) \right]$$

TLEVC=2:

$$PHI(t) = PHI - PTC \cdot \Delta t$$

TLEVC=3:

$$PHI(t) = PHI + dphidt \cdot \Delta t$$

where TLEV=0 or 1:

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

TLEV=2:

$$dphidt = -\frac{\left[egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left(2 - \frac{tnom}{tnom + GAP2}\right) - PHI\right]}{tnom}$$

Calculating Threshold Voltage Temperature Equations

The threshold temperature equations are:

TLEV=0:

$$vbi(t) = vbi(tnom) + \frac{PHI(t) - PHI}{2} + \frac{egnom - eg(t)}{2}$$

$$VTO(t) = vbi(t) + GAMMA \cdot (PHI(t))^{1/2}$$

TLEV=1:

$$VTO(t) = VTO - TCV \cdot \Delta t$$

$$vbi(t) = VTO(t) - GAMMA \cdot (PHI(t))^{1/2}$$

TLEV=2:

$$VTO(t) = VTO + \left(1 + \frac{GAMMA}{2 \cdot PHI^{1/2}}\right) \cdot dphidt \cdot \Delta t$$

$$vbi(t) = VTO(t) - GAMMA \cdot (PHI(t))^{1/2}$$

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:

$$RD(t) \ = \ RS \cdot (1 + TRD \cdot \Delta t)$$

$$RS(t) \ = \ RS \cdot (1 + TRS \cdot \Delta t)$$

Avant!

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

Name (Alias)	Units	Default	Description
LEVEL		1.0	DC model selector. LEVEL 1 is the Schichman-Hodges model.
COX	F/m^2	3.453e-4	Oxide capacitance per unit gate area. If COX is not specified, it is calculated from TOX.
KP (BET, BETA)	A/V^2		Intrinsic transconductance parameter. If KP is not specified and UO and TOX are entered, the parameter is computed from: KP = UO · COX The default=2.0718e-5 (NMOS), 8.632e-6 (PMOS).
LAMBDA (LAM, LA)	V^{-1}	0.0	Channel-length modulation
TOX	m	1e-7	Gate oxide thickness
UO	$cm^2/$ $(V·s)$		Carrier mobility

Effective Width and Length Parameters

Name (Alias)	Units	Default	Description
DEL	m	0.0	Channel length reduction on each side. $DEL_{scaled} = DEL \cdot SCALM$
LD (DLAT, LATD)	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 as: LD Default=0.75 · XJ $LD_{scaled} = LD \cdot SCALM$
LDAC	m		This parameter is the same as LD, but if LDAC is in the .MODEL statement, it replaces LD in the Leff calculation for AC gate capacitance.
LMLT		1.0	Length shrink factor
WD	m	0.0	Lateral diffusion into channel from bulk along width $WD_{scaled} = WD \cdot SCALM$
WDAC	m		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.
WMLT		1.0	Diffusion layer and width shrink factor
XJ	m	0.0	Metallurgical junction depth: $XJ_{scaled} = XJ \cdot SCALM$
XL (DL, (LDEL)	m	0.0	Accounts for masking and etching effects: $XL_{scaled} = XL \cdot SCALM$

Name (Alias)	Units	Default	Description
XW (DW, WDEL)	m	0.0	Accounts for masking and etching effects: $XW_{scaled} = XW \cdot SCALM$

Threshold Voltage Parameters

Name (Alias)	Units	Default	Description
GAMMA	$V^{1/2}$	0.5276	Body effect factor. If GAMMA is not specified, it is calculated from NSUB (See "Common Threshold Voltage Parameters" on page 20-50).
NFS (DFS, NF, DNF)	$cm^{-2} \cdot V^{-1}$	0.0	Fast surface state density
NSUB (DNB, NB)	cm^{-3}	1e15	Bulk surface doping. NSUB is calculated from GAMMA if not specified.
PHI	V	0.576	Surface inversion potential –PH is calculated from NSUB if not specified (See "Common Threshold Voltage Parameters" on page 20-50).
VTO (VT)	V		Zero-bias threshold voltage. If not specified, it is calculated. (See "Common Threshold Voltage Parameters" on page 20-50).

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 run-time 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:

$$L_{eff} = L_{scaled} \cdot LMLT + XL_{scaled} - 2 \cdot (LD_{scaled} + DEL_{scaled})$$

$$W_{eff} = M \cdot (W_{scaled} WMLT + XW_{scaled} - 2 \cdot WD_{scaled})$$

Threshold Voltage, v_{th}

$v_{sb} \geq 0$

$$v_{th} = v_{bi} + GAMMA \cdot (PHI + v_{sb})^{1/2}$$

$v_{sb} < 0$

$$v_{th} = v_{bi} + GAMMA \cdot \left(PHI^{1/2} + 0.5 \frac{v_{sb}}{PHI^{1/2}} \right)$$

Where the built-in voltage v_{bi} is defined as:

$$v_{bi} = v_{fb} + PHI$$

or

$$v_{bi} = VTO - GAMMA \cdot PHI^{1/2}$$

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

Name (Alias)	Units	Default	Description
LEVEL		1.0	DC model selector. LEVEL 2 is the Grove-Frohman model.
COX	F/m^2	3.453e-4	Oxide capacitance per unit gate area. This parameter is calculated from TOX if not specified.
ECRIT (ESAT)	V/cm	0.0	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 · (VMAX / UO)
KP (BET, BETA)	A/V^2	2.0e-5	Intrinsic transconductance. If KP is not specified and UO and TOX are entered, KP is calculated from KP = UO · COX
LAMBDA (LAM, LA)	V^{-1}	0.0	Channel length modulation

Name (Alias)	Units	Default	Description
NEFF		1.0	Total channel charge (fixed and mobile) coefficient
TOX	m	1e-7	Gate oxide thickness
VMAX (VMX, VSAT)	m/s	0.0	Maximum drift velocity of carriers. Use zero to indicate an infinite value.

Effective Width and Length Parameters

Name (Alias)	Units	Default	Description
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 · 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 $WD_{scaled} = WD \cdot SCALM$

Name (Alias)	Units	Default	Description
WDAC	m		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.
WMLT		1.0	Diffusion layer and width shrink factor
WREF	m	0.0	Channel width reference $WREF_{scaled} = WREF \cdot SCALM$
XJ	m	0.0	Metallurgical junction depth $XJ_{scaled} = XJ \cdot SCALM$
XL (DL, LDEL)	m	0.0	Length bias accounts for masking and etching effects $XL_{scaled} = XL \cdot SCALM$
XW (DW, WDEL)	m	0.0	Width bias accounts for masking and etching effects $XW_{scaled} = XW \cdot SCALM$

Threshold Voltage Parameters

Name (Alias)	Units	Default	Description
DELTA		0.0	Narrow width factor for adjusting threshold
GAMMA	$V^{1/2}$	0.5276	Body effect factor. This parameter is calculated from NSUB if not specified (see "Common Threshold Voltage Parameters" on page 20-50).
LND	µm/V	0.0	ND length sensitivity
LN0	µm	0.0	N0 length sensitivity
ND	V^{-1}	0.0	Drain subthreshold factor
N0		0.0	Gate subthreshold factor. Typical value=1.

Name (Alias)	Units	Default	Description
NFS (DFS, NF, DNF)	$cm^{-2} \cdot V^{-1}$	0.0	Fast surface state density
NSUB (DNB, NB)	cm^{-3}	1e15	Bulk surface doping. If NSUB is not specified, it is calculated from GAMMA.
PHI	V	0.576	Surface inversion potential. If PHI is not specified, it is calculated from NSUB (see "Common Threshold Voltage Parameters" on page 20-50).
VTO(VT)	V		Zero-bias threshold voltage. If it is not specified, it is calculated (see "Common Threshold Voltage Parameters" on page 20-50).
WIC		0.0	Subthreshold model selector
WND	µm/V	0.0	ND width sensitivity.
WN0	µm	0.0	N0 width sensitivity

Mobility Parameters

Name (Alias)	Units	Default	Description
MOB		0.0	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. *Note:* *MOB=7 operates as a flag. It invokes the channel length modulation and mobility equations of MOSFET LEVEL 3.*

Name (Alias)	Units	Default	Description
THETA	V^{-1}	0.0	Mobility modulation. THETA is used only when MOB=7. A typical value in this application is THETA=5e-2.
UCRIT	V/cm	1.0e4	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.
UEXP (F2)		0.0	Critical field exponent in the empirical formula which characterizes surface mobility degradation
UO (UB, UBO)	$cm^2/$ $(V \cdot s)$	600 (N) 250 (P)	Low-field bulk mobility. This parameter is calculated from KP if KP is input.
UTRA		0.0	Transverse field coefficient
			Note: SPICE does not use UTRA. Star-Hspice uses it if supplied, but issues a warning because UTRA can hinder convergence.

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.

Cutoff Region, $v_{gs} \leq v_{th}$

$$I_{ds} = 0 \qquad \textit{(see subthreshold current)}$$

On Region, $v_{gs} > v_{th}$

$$I_{ds} = \beta \cdot \left\{ \left(v_{gs} - v_{bi} - \frac{\eta \cdot v_{de}}{2} \right) \cdot v_{de} - \frac{2}{3} \cdot \gamma \cdot [(PHI + v_{de} + v_{sb})^{3/2} - (PHI + v_{sb})^{3/2}] \right\}$$

where:

$$v_{de} = min(v_{ds}, v_{dsat})$$

$$\eta = 1 + DELTA \cdot \frac{\pi \cdot \varepsilon si}{4 \cdot COX \cdot W_{eff}}$$

$$\beta = KP \cdot \frac{W_{eff}}{L_{eff}}$$

Effective Channel Length and Width

The model calculates effective channel length and width from the drawn length and width as follows:

$$L_{eff} = L_{scaled} \cdot LMLT + XL_{scaled} - 2 \cdot (LD_{scaled} + DEL_{scaled})$$

$$W_{eff} = M \cdot (W_{scaled} \cdot WMLT + XW_{scaled} - 2 \cdot WD_{scaled})$$

$$LREF_{eff} = LREF_{scaled} \cdot LMLT + XL_{scaled} - 2 \cdot (LD_{scaled} + DEL_{scaled})$$

$$WREF_{eff} = M \cdot (WREF_{scaled} \cdot WMLT + XW_{scaled} - 2 \cdot WD_{scaled})$$

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 η. To include the narrow width effect, specify the model parameter DELTA. The short-channel effect is included through the effective γ. 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[1] 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):

$$u_{eff} = UO \cdot \left[\frac{UCRIT \cdot E_{si}}{COX \cdot (v_{gs} - v_{th} - UTRA \cdot v_{ds})} \right]^{UEXP}$$

Since u_{eff} is less than UO, the program uses the above equation if the bracket term is less than one; otherwise the program uses u_{eff}=UO.

If MOB=7, THETA≠0

$$u_{eff} = \frac{UO}{1 + THETA \cdot (v_{gs} - v_{th})}$$

v_{gs}<v_{th}, u_{eff} = UO

If MOB=7, THETA=0

$$u_{eff} = UO \cdot \left[\frac{UCRIT \cdot E_{si}}{COX \cdot (v_{gs} - v_{th})} \right]^{UEXP}$$

If MOB=7, VMAX>0

$$u_{eff} = \frac{u_{eff}}{1 + u_{eff} \cdot \dfrac{vde}{VMAX \cdot L_{eff}}}$$

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 λ if you do not specify the model parameter LAMBDA.

LAMBDA>0

$$\lambda = LAMBDA$$

VMAX>0, NSUB >0, and LAMBDA≤0

$$\lambda = \frac{X_d}{NEFF^{1/2} \cdot L_{eff} \cdot v_{ds}} \cdot \left\{ \left[\left(\frac{VMAX \cdot X_d}{2 \cdot NEFF^{1/2} \cdot u_{eff}} \right)^2 + v_{ds} - v_{dsat} \right]^{1/2} - \frac{VMAX \cdot X_d}{2 \cdot NEFF^{1/2} \cdot u_{eff}} \right\}$$

VMAX=0, NSUB>0, and LAMBDA ≤0

If MOB=0

$$\lambda = \frac{X_d}{L_{eff} \cdot v_{ds}} \cdot \left\{ \frac{v_{ds} - v_{dsat}}{4} + \left[1 + \left(\frac{v_{ds} - v_{dsat}}{4} \right)^2 \right]^{1/2} \right\}^{1/2}$$

If MOB=7

$$\lambda = \frac{X_d}{L_{eff} \cdot v_{ds}} \cdot \left\{ \left[\frac{v_{ds} - v_{dsat}}{4} + \left(1 + \left(\frac{v_{ds} - v_{dsat}}{4} \right)^2 \right)^{1/2} \right]^{1/2} - 1 \right\}$$

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 Leff 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:

$$\frac{xwb}{1 + \dfrac{xwb - Le}{xwb}}$$

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:

$$fast = vt \cdot \left[\eta + (PHI + v_{sb})^{1/2} \cdot \frac{\partial \gamma}{\partial v_{sb}} + \frac{\gamma}{2 \cdot (PHI + v_{sb})^{1/2}} + \frac{q \cdot NFS}{COX} \right]$$

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:

$$I_{ds} = I_{ds}(v_{gs}, vde, v_{sb}) + isub(N0_{eff}, ND_{eff}, v_{gs}, v_{ds})$$

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

Name (Alias)	Units	Default	Description
LEVEL		1.0	DC model selector. LEVEL=3 is an empirical model.
COX	F/m^2	3.453e-4	Oxide capacitance per unit gate area. If this parameter is not specified, it is calculated from TOX.
DERIV		1	Derivative method selector DERIV=0: analytic DERIV=1: finite difference
KAPPA	V^{-1}	0.2	Saturation field factor. This parameter is used in the channel length modulation equation.
KP (BET, BETA)	A/V^2	2.0e-5	Intrinsic transconductance parameter. If this parameter is not specified and UO and TOX are entered, KP is calculated from $KP = UO \cdot COX$
TOX	m	1e-7	Gate oxide thickness
VMAX (VMX)	m/s	0.0	Maximum drift velocity of carriers. Use zero to indicate an infinite value.

Effective Width and Length Parameters

Name (Alias)	Units	Default	Description
DEL	m	0.0	Channel length reduction on each side $DEL_{scaled} = DEL \cdot SCALM$
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 · XJ.
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.
LREF	m	0.0	Channel length reference $LREF_{scaled} = LREF \cdot SCALM$
LMLT		1.0	Length shrink factor
WD	m	0.0	Lateral diffusion into channel width from bulk $WD_{scaled} = WD \cdot SCALM$
WDAC	m		This parameter is the same as WD, but if WDAC is included in the .MODEL statement, it replaces WD in the Weff calculation for AC gate capacitance.
WMLT		1.0	Diffusion layer and width shrink factor
WREF	m	0.0	Channel width reference $WREF_{scaled} = WREF \cdot SCALM$

Name (Alias)	Units	Default	Description
XJ	m	0.0	Metallurgical junction depth $XJ_{scaled} = XJ \cdot SCALM$
XL (DL, LDEL)	m	0.0	Length bias accounts for masking and etching effects $XL_{scaled} = XL \cdot SCALM$
XW (DW, WDEL)	m	0.0	Width bias accounts for masking and etching effects $XW_{scaled} = XW \cdot SCALM$

Threshold Voltage Parameters

Name (Alias)	Units	Default	Description
DELTA		0.0	Narrow width factor for adjusting threshold
ETA		0.0	Static feedback factor for adjusting threshold
GAMMA	$V^{1/2}$	0.5276	Body effect factor. This parameter is calculated from NSUB if not specified (See "Common Threshold Voltage Parameters" on page 20-50).
LND	µm/V	0.0	ND length sensitivity
LN0	µm	0.0	N0 length sensitivity
ND	V^{-1}	0.0	Drain subthreshold factor
N0		0.0	Gate subthreshold factor (typical value=1)
NFS (DFS,NF, DNF)	$cm^{-2} \cdot V^{-1}$	0.0	Fast surface state density

Name (Alias)	Units	Default	Description
NSUB (DNB, NB)	cm^{-3}	1e15	Bulk surface doping. This parameter is calculated from GAMMA if not specified.
PHI	V	0.576	Surface inversion potential. This parameter is calculated from NSUB if not specified (see "Common Threshold Voltage Parameters" on page 20-50).
VTO (VT)	V		Zero-bias threshold voltage. This parameter is calculated if not specified (see "Common Threshold Voltage Parameters" on page 20-50).
WIC		0.0	Sub-threshold model selector
WND	μm/V	0.0	ND width sensitivity
WN0	μm	0.0	N0 width sensitivity

Mobility Parameters

Name (Alias)	Units	Default	Description
THETA	V^{-1}	0.0	Mobility degradation factor
UO (UB,UBO)	cm^2/(V·s)	600(N) 250(P)	Low field bulk mobility. This parameter is calculated from KP if KP is specified.

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}.

Cutoff Region, $v_{gs} \leq v_{th}$

$$I_{ds} = 0 \qquad \textit{(See subthreshold current)}$$

On Region, $v_{gs} > v_{th}$

$$I_{ds} = \beta \cdot \left(v_{gs} - v_{th} - \frac{1 + fb}{2} \cdot vde \right) \cdot vde$$

where:

$$\beta = KP \cdot \frac{W_{eff}}{L_{eff}}$$

$$= u_{eff} \cdot COX \cdot \frac{W_{eff}}{L_{eff}}$$

$$vde = \min(v_{ds}, v_{dsat})$$

and:

$$fb = f_n + \frac{GAMMA \cdot f_s}{4 \cdot (PHI + v_{sb})^{1/2}}$$

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:

$$f_n = \frac{DELTA}{W_{eff}} \cdot \frac{1}{4} \cdot \frac{2\pi \cdot E_{si}}{COX}$$

The term f_s expresses the effect of the short channel and is determined as:

$$f_s = 1 - \frac{XJ_{scaled}}{L_{eff}} \cdot \left\{ \frac{LD_{scaled} + W_c}{XJ_{scaled}} \cdot \left[1 - \left(\frac{W_p}{XJ_{scaled} + W_p} \right)^2 \right]^{1/2} - \frac{LD_{scaled}}{XJ_{scaled}} \right\}$$

$$W_p = X_d \cdot (PHI + v_{sb})^{1/2}$$

$$X_d = \left(\frac{2 \cdot E_{si}}{q \cdot NSUB} \right)^{1/2}$$

$$W_c = XJ_{scaled} \cdot \left[0.0631353 + 0.8013292 \cdot \left(\frac{W_p}{XJ_{scaled}} \right) - 0.01110777 \cdot \left(\frac{W_p}{XJ_{scaled}} \right)^2 \right]$$

Effective Channel Length and Width

The model determines effective channel length and width in the LEVEL 3 model as follows:

$$L_{eff} = L_{scaled} \cdot LMLT + XL_{scaled} - 2 \cdot (LD_{scaled} + DEL_{scaled})$$

$$W_{eff} = M \cdot (W_{scaled} \cdot WMLT + XW_{scaled} - 2 \cdot WD_{scaled})$$

$$LREFF = LREF_{scaled} \cdot LMLT + XL_{scaled} - 2 \cdot (LD_{scaled} + DEL_{scaled})$$

$$WREF_{eff} = M \cdot (WREF_{scaled} \cdot WMLT + XW_{scaled} - 2 \cdot WD_{scaled})$$

Threshold Voltage, v_{th}

The effective threshold voltage, including the device size and terminal voltage effects, is calculated by:

$$v_{th} = v_{bi} - \frac{8.14e\text{-}22 \cdot ETA}{COX \cdot L_{eff}^3} \cdot v_{ds} + GAMMA \cdot f_s \cdot (PHI + v_{sb})^{1/2} + f_n \cdot (PHI + v_{sb})$$

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).

$v_{gs} > v_{th}$

$$us = \frac{UO}{1 + THETA \cdot (v_{gs} - v_{th})}$$

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

$$u_{eff} = \frac{us}{1 + \dfrac{vde}{v_c}}$$

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 (ΔL) differently, depending on the VMAX model parameter value.

VMAX = 0

$$\Delta L = X_d \cdot [KAPPA \cdot (v_{ds} - v_{dsat})]^{1/2}$$

VMAX>0

$$\Delta L = -\frac{E_p \cdot X_d^2}{2} + \left[\left(\frac{E_p \cdot X_d^2}{2} \right)^2 + KAPPA \cdot X_d^2 \cdot (v_{ds} - v_{dsat}) \right]^{1/2}$$

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:

$$I_{ds} = \frac{I_{ds}}{1 - \dfrac{\Delta L}{L_{eff}}}$$

In order to prevent the denominator from going to zero, Star-Hspice limits the ΔL value as follows:

If

$$\Delta L > \frac{L_{eff}}{2}$$

then

$$\Delta L = L_{eff} - \frac{\left(\dfrac{L_{eff}}{2} \right)^2}{\Delta L}$$

Subthreshold Current, I_{ds}

This region of operation is characterized by the model parameter for fast surface state (NFS). The modified threshold voltage (von) is determined as follows:

NFS>0

$$von = v_{th} + fast$$

where:

$$fast = v_{tm} \cdot \left[1 + \frac{q \cdot NFS}{COX} + \frac{GAMMA \cdot f_s \cdot (PHI + v_{sb})^{1/2} + f_n \cdot (PHI + v_{sb})}{2 \cdot (PHI + v_{sb})} \right]$$

The current I_{ds} is given by:

v_{gs}<von

$$I_{ds} = I_{ds}(von, vde, v_{sb}) \cdot e^{\frac{v_{gs} - von}{fast}}$$

v_{gs}Švon

$$I_{ds} = I_{ds}(v_{gs}, vde, v_{sb})$$

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_{ds} current is:

$$I_{ds} = I_{ds}(v_{gs}, vde, v_{sb}) + isub(N0_{eff}, ND_{eff}, v_{gs}, v_{ds})$$

Subthreshold current isub for LEVEL=3 is the same as for LEVEL=13 (see page 21-115).

$N0_{eff}$ and ND_{eff} 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,

$$GAMMA \rightarrow f_s \cdot GAMMA$$

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 $0.75 \cdot XJ$. 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	$= 1.3806226e\text{-}23J \cdot K^{-1}$
Electron charge	e	$= 1.6021918e\text{-}19C$
Permittivity of silicon dioxide	ε_{ox}	$= 3.45314379969e\text{-}11F/m$
Permittivity of silicon	ε_{si}	$= 1.035943139907e\text{-}10F/m$

Temperature Compensation

The example below verifies temperature dependence for LEVEL 3.

Input file

```
$ 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:

$$I_{ds} = \frac{beta \cdot 0.5 \cdot (v_{gs} - v_{tm})^2}{1 + fb}$$

$$beta = COX \cdot \left(\frac{W}{L}\right) \cdot UO(t) \quad fb = \frac{GAMMA}{(4 \cdot \sqrt{phi(t)})}$$

Using the model parameters in the input file and the equations from the previous page:

$$beta = (1.2e - 3) \cdot \left(\frac{t}{tref}\right)^{BEX}$$

$$v_{tm} = 0.8 - TCV \cdot (t - tref)$$

$$phi(t) = 0.64 \cdot \left(\frac{t}{tref}\right) - vtherm \cdot \left(egarg + 3 \cdot \log\left(\frac{t}{tref}\right)\right)$$

At room temperature:

$$beta = (1.2e - 3)$$

$$v_{tm} = 0.8$$

$$phi(t) = 0.64$$

$$I_{ds} = (1.2e - 3) \cdot 0.5 \cdot \frac{(2 - 0.8)^2}{1 + \frac{0.2}{\sqrt{0.64}}} = 6.912e - 4$$

At T=100:

$$beta = 1.2e - 3 \cdot (1.251551)^{-1.5} = 0.570545e - 4$$

$$v_{tm} = 0.8 - (1.5e - 3) \cdot 75 = 0.6875$$

$$egarg = 9.399920 \quad vtherm = 3.215466e - 2$$

$$phi(t) = 0.64 \cdot 1.251551 - 0.3238962 = 0.4770964$$

$$I_{ds} = beta \cdot 0.5 \cdot \frac{(2 - vt)^2}{1 + \dfrac{0.2}{\sqrt{phi(t)}}} = 5.724507e - 4$$

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: $\gamma = GAMMA$
- For lateral diffusion, $LD_{scaled} = LD \cdot XJ \cdot SCALM$. 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

Name (Alias)	Units	Default	Description
LEVEL		1.0	Model level selector
DNB (NSUB)	cm^{-3}	0.0	Surface doping
DP	μm	1.0	Implant depth (depletion model only)
ECV	V/μm	1000	Critical field
NI	cm^{-2}	2e11	Implant doping (depletion model only)
PHI	V	0.8	Built-in potential
TOX	Å	0.0	Oxide thickness
TUH		1.5	Implant channel mobility temperature exponent (depletion model only)
ZENH		1.0	Mode flag (enhancement). Set ZENH=0.0 for depletion mode.

Effective Width and Length Parameters

Name (Alias)	Units	Default	Description
DEL (WDEL)	μm	0.0	Channel length reduction on each side
LATD (LD)	μm	1.7 · XJ	Lateral diffusion on each side
LMLT		1.0	Length shrink factor
OXETCH	μm	0.0	Oxide etch
WMLT		1.0	Diffusion layer and width shrink factor

Threshold Voltage Parameters

Name (Alias)	Units	Default	Description
FSS (NFS)	$cm^{-2} \cdot V^{-1}$	0.0	Number of fast surface states
NWM		0.0	Narrow width modifier
SCM		0.0	Short-channel drain source voltage multiplier
VT (VTO)	V	0.0	Extrapolated threshold voltage
XJ	μm	1.5	Junction depth

Mobility Parameters

Name (Alias)	Units	Default	Description
FRC	$Å \cdot s / cm^2$	0.0	Field reduction coefficient
FSB	$V^{1/2} \cdot s / cm^2$	0.0	Lateral mobility coefficient
UB (UO)	$cm^2 / (V \cdot s)$	0.0	Low field bulk mobility

Name (Alias)	Units	Default	Description
UH	cm^2/ (V·s)	900 (N) 300 (P)*	Implant - channel mobility * (For depletion model only)
VST	cm/s	0.0	Saturation velocity

Capacitance Parameters

Name (Alias)	Units	Default	Description
AFC		1.0	Area factor for MOSFET capacitance
CAPOP		6	Gate capacitance selector
METO	μm	0.0	Metal overlap on gate

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.

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, $v_{gs} \leq v_{th}$

$$I_{ds} = 0$$　　　　*(See "Subthreshold Current, I_{ds}" on page 21-41)*

On Region, $v_{gs} \check{S} v_{th}$

$$I_{ds} = \beta \cdot \left\{ \left(v_{gs} - v_{bi} - \frac{vde}{2} \right) \cdot vde - \frac{2}{3} \cdot \gamma \cdot [(\Phi_f + vde + v_{sb})^{3/2} - (\Phi_f + v_{sb})^{3/2}] \right\}$$

where:

$$vde = min(v_{ds}, v_{dsat})$$

$$\beta = UB_{eff} \cdot cox \cdot \frac{W_{eff}}{L_{eff}}$$

$$\Phi_f = 2 \cdot v_{tm} \cdot \ln\left(\frac{DNB}{ni} \right)$$

and gate oxide capacitances per unit area are calculated by:

$$cox = \frac{E_{ox}}{TOX \cdot 1E\text{-}10} \quad \text{F/m}$$

Effective Channel Length and Width

The effective channel length and width in the LEVEL 5 model is determined as follows.

$$W_{eff} = W_{scaled} \cdot WMLT + OXETCH$$

$$L_{eff} = L_{scaled} \cdot LMLT - 2 \cdot (LATD + DEL)$$

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:

$$v_{bi} = v_{fb} + \Phi_f = VTO - \gamma_0 \cdot \Phi_f^{1/2}$$

$$\gamma_0 = \frac{(2 \cdot E_{si} \cdot q \cdot DNB)^{1/2}}{cox}$$

Note: *For LEVEL 5 model, you must specify DNB and VTO parameters. The Star-Hspice program computes γ_0 using DNB and ignores the GAMMA model parameter.*

The effective body effect (γ), including the device size effects, is computed as follows.

$$\gamma = \gamma_0 \cdot (1 - scf) \cdot (1 + ncf)$$

If SCM ≤ 0,

$$scf = 0$$

otherwise,

$$scf = \frac{XJ}{L_{eff}} \cdot \left\{ \left[1 + \frac{2xd}{XJ} \cdot (SCM \cdot v_{ds} + v_{sb} + \Phi_f)^{1/2} \right]^{1/2} - 1 \right\}$$

If NWM ≤ 0,

$$ncf = 0$$

otherwise,

$$ncf = \frac{NWM \cdot X_d \cdot (\Phi_f)^{1/2}}{W_{eff}}$$

where:

$$X_d = \left(\frac{2 \cdot E_{si}}{q \cdot DNB} \right)^{1/2}$$

Saturation Voltage, v_{dsat}

The saturation voltage due to channel pinch-off at the drain side is computed by:

$$v_{sat} = v_{gs} - v_{bi} + \frac{\gamma^2}{2} \cdot \left\{ 1 - \left[1 + \frac{4}{\gamma^2} \cdot (v_{gs} - v_{bi} + \Phi_f + v_{sb}) \right]^{1/2} \right\}$$

$$v_{dsat} = v_{sat}$$

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:

$$UB_{eff} = \cfrac{1}{\cfrac{1}{UB} + \cfrac{FRC \cdot (v_{gs} - v_{th})}{TOX} + \cfrac{vde}{VST \cdot L_e} + FSB \cdot v_{sb}^{1/2}}$$

where:

$$L_e = L_{eff} \qquad \textit{linear region}$$

$$L_e = L_{eff} - \Delta L \; \textit{saturation region}$$

The channel length modulation effect (ΔL) 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:

$$I_{ds} = \cfrac{I_{ds}}{1 - \cfrac{\Delta L}{L_{eff}}}$$

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 ΔL is in microns, assuming XJ is in microns and DNB is in 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:

$$von \ = \ v_{th} + fast$$

where:

$$fast \ = \ v_{tm} \cdot \left[1 + \frac{q \cdot FSS}{cox} + \frac{\gamma}{2 \cdot (\Phi_f + v_{sb})^{1/2}} \right]$$

and vt is the thermal voltage.

The I_{ds} is given by:

Weak Inversion Region, v$_{gs}$ <v$_{th}$

$$I_{ds} \ = \ (von, vde, v_{sb}) \cdot e^{\frac{v_{gs} - von}{fast}}$$

Strong Inversion Region, v$_{gs}$ ≥ v$_{th}$

$$I_{ds} \ = \ I_{ds}(v_{gs}, vde, v_{sb})$$

Note: *The modified threshold voltage (von) produced by FSS is also used in strong inversion; that is, in the mobility equations, von 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[2]. 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, QB. The additional charge from the implant, QBI, 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 (UH)
- 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

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

$$I_{ds} = \beta 1 \cdot \left\{ q \cdot NI \cdot vde + cav \cdot \left[(v_{gs} - v_{fb}) \cdot vde - \frac{vde^2}{2} \right] \right.$$

$$\frac{2}{3} \cdot cav \cdot \gamma \cdot [(vde + v_{sb} + \Phi_d)^{3/2} - (v_{sb} + \Phi_d)^{3/2}]$$

Enhancement, v_{gs}-v_{fb} Švde >0

$$I_{ds} = \beta 1 \cdot \left\{ q \cdot NI \cdot vde - \frac{2}{3} \cdot cav \cdot \gamma \cdot [(vde + v_{sb} + \Phi_d)^{3/2} - (v_{sb} + \Phi_d)^{3/2}] \right\}$$

$$+ \beta \cdot \left[(v_{gs} - v_{fb}) \cdot vde - \frac{vde^2}{2} \right]$$

Partial Enhancement, v_{gs}-v_{fb}<vde

$$I_{ds} = \beta 1 \cdot \left\{ q \cdot NI \cdot vde + cav \cdot \left[(v_{gs} - v_{fb}) \cdot vde - \frac{vde^2}{2} \right] \right.$$

$$\left. - \frac{2}{3} \cdot cav \cdot \gamma \cdot [(vde + v_{sb} + \Phi_d)^{3/2} - (v_{sb} + \Phi_d)^{3/2}] \right\}$$

$$+ \left(\frac{1}{2}\beta - \beta 1 \cdot cav \right) \cdot (v_{gs} - v_{fb})^2$$

where:

$$\beta 1 = UH \cdot \frac{W_{eff}}{L_{eff}}$$

$$\beta = UBeff \cdot cox \cdot \frac{W_{eff}}{L_{eff}}$$

$$cav = \frac{cox \cdot cs}{cox + cs}$$

$$cs = \frac{2.77 E_{si}}{DP \cdot 1e\text{-}4}$$

$$\Phi_d = v_{tm} \cdot \ln\left(\frac{DNB \cdot nd}{ni^2}\right)$$

$$nd = \frac{NI \cdot 1e4}{DP}$$

and:

$$vde = \min(v_{ds}, v_{dsat})$$

The saturation voltage, threshold voltage, and effective γ are described in the following sections.

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:

$$v_{th} = v_{fb} - \beta d \cdot [vch - \gamma \cdot (\Phi_d + v_{sb})^{1/2}]$$

where:

$$v_{fb} = VTO + \beta d \cdot (vch - \gamma_0 \cdot \Phi_d^{1/2})$$

$$\beta d = \frac{UH \cdot cav}{UB \cdot cox}$$

$$vch = \frac{q \cdot NI}{cav}$$

$$\gamma_0 = \frac{(2 \cdot E_{si} \cdot q \cdot na1)^{1/2}}{cav}$$

$$na1 = \frac{nd \cdot DNB}{nd + DNB}$$

$$nd = \frac{NI}{DP \cdot 1e\text{-}4}$$

The effective γ, including small device size effects, is computed as follows:

$$\gamma = \gamma_0 \cdot (1 - scf) \cdot (1 + ncf)$$

where:

If SCM ≤ 0,

$$scf = 0$$

otherwise,

$$scf = \frac{XJ}{L_{eff}} \cdot \left\{ \left[1 + \frac{2xd}{XJ} \cdot (SCM \cdot v_{ds} + v_b + \Phi_d)^{1/2} \right]^{1/2} - 1 \right\}$$

If NWM ≤ 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 $v_{gs} \le v_{th}$, the surface is inverted and a residual DC current exists. When v_{sb} is large enough to make $v_{th} > vin_{th}$, then v_{th} is used as the inversion threshold voltage. In order to determine the residual current, vin_{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 vin_{th}, which is computed as:*
>
> $$vinth = v_{fb} - \frac{q \cdot NI}{cox} - v_{sb}$$

Saturation Voltage, v_{dsat}

The saturation voltage (v_{sat}) is determined as:

$$v_{sat} = v_{gs} - v_{fb} + vch + \frac{\gamma 2}{2} \cdot \left\{ 1 - \left[1 + \frac{4}{\gamma 2} \cdot (v_{gs} - v_{fb} + vch + v_{sb} + \Phi_d) \right]^{1/2} \right\}$$

$$v_{dsat} = v_{sat}$$

IF ECV is not equal to 1000 (V/μm), Star-Hspice modifies v_{sat} to include the carrier velocity saturation effect.

$$v_{dsat} = v_{sat} + v_c - (v_{sat} + v_c^2)^{1/2}$$

where:

$$v_c = ECV \cdot L_{eff}$$

Mobility Reduction, UB_{eff}

The surface mobility (UB) is dependent upon terminal voltages as follows:

$$UB_{eff} = \cfrac{1}{\cfrac{1}{UB} + \cfrac{FRC \cdot (v_{gs} - v_{th})}{TOX} + \cfrac{vde}{VST \cdot le} + FSB \cdot v_{sb}^{1/2}}$$

where:

$$L_e = L_{eff} \qquad\qquad \textit{Linear region}$$

$$L_e = L_{eff} - \Delta L \qquad\qquad \textit{Saturation region}$$

The channel length modulation effect (ΔL) is defined next.

Channel Length Modulation

The channel length modulation effect is included by modifying the I_{ds} current as:

$$I_{ds} = \frac{I_{ds}}{1 - \dfrac{\Delta L}{L_{eff}}}$$

where:

$$\Delta L = 1e4 \cdot \left[\frac{2.73e3 \cdot XJ}{na1 \cdot \ln\left(\dfrac{1e20}{na1}\right)} \right]^{1/3} \cdot [(v_{ds} - v_{dsat} + PHI)^{1/3} - PHI^{1/3}]$$

The ΔL 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:

$$von = v_{th} + fast$$

where:

$$
fast = v_{tm} \cdot \left[1 + \frac{q \cdot FSS}{cox} + \frac{\gamma}{2 \cdot (\Phi_d + v_{sb})^{1/2}} \right]
$$

If von < vin$_{th}$, then vin$_{th}$ is substituted for von.

> ***Note:*** *The Star-Hspice LEVEL 5 model uses the following subthreshold model only if v_{gs} < von and the device is either in partial or full enhancement mode. Otherwise, it use the model in enhancement mode (ZENH=1). The subthreshold current calculated below includes the residual DC current.*

If v$_{gs}$ <von then,

Partial Enhancement, v$_{gs}$-v$_{fb}$ < vde

$$
I_{ds} = \beta 1 \cdot \left\{ q \cdot NI \cdot vde + cav \cdot \left[(von - v_{fb}) \cdot vde - \frac{vde^2}{2} \right] \right.
$$

$$
\left. - \frac{2}{3} \cdot cav \cdot \gamma \cdot [(vde + v_{sb} + \Phi_d)^{3/2} - (v_{sb} + \Phi_d)^{3/2}] \right\}
$$

$$
+ \frac{1}{2} \cdot \left(\beta \cdot e^{\frac{v_{gs} - von}{fast}} - \beta 1 \cdot cav \right) \cdot (von - v_{fb})^2
$$

Full Enhancement, v_{gs}-v_{fb} Š vde > 0

$$I_{ds} = \beta 1 \cdot \left\{ q \cdot NI \cdot vde - \frac{2}{3} \cdot cav \cdot \gamma [(vde + v_{sb} + \Phi_d)^{3/2} - (v_{sb} + \Phi_d)^{3/2}] \right\}$$

$$+ \beta \cdot \left[(von - v_{fb}) \cdot vde - \frac{vde^2}{2} \right] \cdot e^{\frac{v_{gs} - von}{fast}}$$

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
```

Depletion Mode

```
.MODEL 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

Name (Alias)	Units	Default	Description
LEVEL		1.0	IDS equation selector LEVEL=6 Lattin-Jenkins-Grove model, using ASPEC-style parasitics **Note:** 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.) LEVEL=7 Lattin-Jenkins-Grove model, using SPICE-style parasitics
CLM (GDS)		0.0	Channel length modulation equation selector
DNB (NSUB)	$1/cm^3$	1.0e15	Substrate doping
DNS (NI)	$1/cm^3$	0.0	Surface substrate doping
ECRIT (ESAT)	V/cm	0.0	Drain-source critical field. Use zero to indicate an infinite value, typically 40,000 V/cm.

Name (Alias)	Units	Default	Description
GAMMA	$V^{1/2}$		Body effect factor. If this parameter is not input, GAMMA is calculated from DNB. GAMMA is the body effect when vsb < VB0. If vsb > VB0, LGAMMA is used. Using GAMMA, LGAMMA, and VB0 allows a two-step approximation of a non-homogeneous substrate.
LGAMMA	$V^{1/2}$	0.0	This parameter is the body effect factor when vsb > VB0. When the Poon-Yau GAMMA expression is used, LGAMMA is junction depth, in microns. In this case LGAMMA is multiplied by SCALM.
MOB		0.0	Mobility equation selector
NWM		0.0	Narrow width modulation of GAMMA
SCM		0.0	Short-channel modulation of GAMMA
UO (UB, UBO)	$cm^2/$ $(V \cdot s)$	600 (N) 250 (P)	This parameter is the low field bulk mobility. It is calculated from KP if KP is supplied.
UPDATE		0.0	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.

Name (Alias)	Units	Default	Description
VB0 (VB)	V	0.0	Reference voltage for GAMMA switch. If vsb < VB0, GAMMA is used. If vsb > VB0, LGAMMA is used in the ids equation.
VMAX (VMX)	cm/s	0.0	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. Typical values: electrons 8.4e6 cm/s holes 4.3e6 cm/s

Effective Length and Width Parameters

Name (Alias)	Units	Default	Description
DEL	m	0.0	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
LD (DLAT, LATD)	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. LD Default=0.75 · XJ. LDscaled = LD · SCALM

Name (Alias)	Units	Default	Description
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.
LREF	m	0.0	Channel length reference LREFscaled = LREF · SCALM
LMLT		1.0	Length shrink factor
WD	m	0.0	Lateral diffusion into channel from bulk along width WDscaled = WD · SCALM
WDAC	m		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.
WMLT		1.0	Diffusion layer and width shrink factor
WREF	m	0.0	Channel width reference WREFscaled = WREF · SCALM
XJ	m	0.0	Metallurgical junction depth XJscaled = XJ · SCALM
XL (DL, LDEL)	m	0.0	Accounts for masking and etching effects XLscaled = XL · SCALM
XW (DW, WDEL)	m	0.0	Accounts for masking and etching effects XWscaled = XW · SCALM

Threshold Voltage Parameters

Name (Alias)	Units	Default	Description
FDS		0.0	Field, drain to source, controls reduction of threshold due to source-drain electric field
LND	µm/V	0.0	ND length sensitivity
LN0	µm	0.0	N0 length sensitivity
ND	1/V	0.0	Drain subthreshold factor. Typical value=1.
N0		0.0	Gate subthreshold factor. Typical value=1.
NFS (DFS, NF)	$cm^{-2} \cdot V^{-1}$	0.0	Fast surface state density
NWE	m	0.0	Narrow width effect, direct compensation of VTO NWEscaled = NWE · SCALM
UFDS		0.0	High field FDS
VFDS	V	0.0	Reference voltage for selection of FDS OR UFDS: FDS used if vds ≤ VFDS UFDS used if vds > VFDS
VSH	V	0.0	Threshold voltage shifter for zero-bias threshold voltage (VTO) reduction as a function of the ratio of LD to Leff
VTO (VT)	V		Zero-bias threshold voltage. This parameter is calculated if not specified (see "Common Threshold Voltage Parameters" on page 20-50).
WEX			Weak inversion exponent

Name (Alias)	Units	Default	Description
WIC		0.0	Subthreshold model selector
WND	μm/V	0.0	ND width sensitivity
WN0	μm	0.0	N0 width sensitivity

Alternate Saturation Model Parameters

Name (Alias)	Units	Default	Description
KA		1.0	Alternate saturation model: short-channel vds scaling factor coefficient
KU		0.0	Lateral field mobility parameter
MAL		0.5	Alternate saturation model: short-channel vds scaling factor exponent
MBL		1.0	Exponent for mobility reduction due to source-drain electric field
NU		1.0	Mobility reduction due to source-drain electric field

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 ISPICE 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,

TOX $= 690$

UO (UB) $= 750$ cm^2/(V · s) (N-ch)

UTRA (F3) $= 0.0$

UPDATE=0 then,

TOX $= 1000$

UO (UB) $= 750$ cm^2/(V · s) (N-ch)

UTRA (F3) $= 0.0$

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,

$$RD = \frac{LATDscaled + LDIFscaled}{Weff} \cdot RD + NRD \cdot \frac{RL}{M}$$

$$RS = \frac{LATDscaled + LDIFscaled}{Weff} \cdot RS + NRS \cdot \frac{RL}{M}$$

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

$$ids = \beta \cdot \left\{ \left(vgs - vbi - \frac{\eta \cdot vde}{2} \right) \cdot vde - \frac{2}{3} \cdot \gamma \cdot [(PHI + vde + vsb)^{3/2} - (PHI + vsb)^{3/2}] \right\}$$

where:

$$vde = \min(vds, vdsat)$$

$$\eta = 1 + \frac{NWEscaled}{weff}$$

$$\beta = ueff \cdot COX \cdot \frac{weff}{Leff}$$

Include the narrow-width effect through η, vbi, and γ values. For the narrow-width effect, specify model parameters NWE and/or NWM. Include the short-channel effect through parameters vbi and γ.

Effective Channel Length and Width

The model calculates effective channel length and width from the drawn length and width as follows:

$$leff = Lscaled \cdot LMLT + XLscaled - 2 \cdot (LDscaled + DELscaled)$$

$$weff = M \cdot (Wscaled \cdot WMLT + XWscaled - 2 \cdot WDscaled)$$

$$LREFeff = LREFscaled \cdot LMLT + XLscaled - 2 \cdot (LDscaled + DELscaled)$$

$$WREFeff = M \cdot (WREFscaled \cdot WMLT + XWscaled - 2 \cdot WDscaled)$$

Threshold Voltage, vth

The model determines effective threshold voltage as follows:

$$vth = vbi + \gamma \cdot (PHI + vsb)^{1/2}$$

The built-in voltage vbi and γ 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.

$$scf = 1 - \frac{LGAMMA}{leff} \cdot \left\{ \left[1 + \frac{2 \cdot LAMBDA}{LGAMMA} \cdot (PHI + vsb)^{1/2} \right]^{1/2} - 1 \right\}$$

Specify the model parameter XJ to modify the model parameter GAMMA by the short-channel factor (gl).

$$gl = 1 - \frac{XJscaled}{leff} \cdot \left\{ \left[1 + \frac{2 \cdot LAMBDA}{XJscaled} \cdot (PHI + vsb + SCM \cdot vds)^{1/2} \right]^{1/2} - 1 \right\}$$

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:

$$xd = \left(\frac{2 \cdot \varepsilon si}{q \cdot DNB}\right)^{1/2}$$

Finally, the effective γ, including short-channel and narrow width effects, is:

$$\gamma = GAMMA \cdot gw \cdot gl \cdot scf$$

Effective Built-in Voltage, vbi

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 vbi 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 vbi modification. To include this effect, you must specify the model parameter FDS and/or UFDS and VFDS.

The expressions for vbi, which sum up the above features, are:

vds≤VFDS , or VFDS=0

$$vbi = VTO - \gamma \cdot PHI^{1/2} + (\eta - 1) \cdot (PHI + vsb) - \frac{LDscaled}{Leff} \cdot VSH - \frac{\varepsilon si}{COX \cdot Leff} \cdot FDS \cdot vds$$

vds>VFDS

$$vbi = VTO - \gamma \cdot PHI^{1/2} + (\eta - 1) \cdot (PHI + vsb) - \frac{LDscaled}{leff} \cdot VSH - \frac{esi}{COX \cdot Leff}$$

$$\cdot [(FDS - UFDS) \cdot VFDS + UFDS \cdot vds]$$

The above equations describe piecewise linear variations of vbi as a function of vds. If you do not specify VFDS, the first equation for vbi is used.

Note: *Star-Hspice calculates model parameters such as VTO, PHI, and GAMMA, if they are not user-specified (see "Common Threshold Voltage Parameters" on page 20-50).*

Multi-Level Gamma, VBO>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 LGAMMA, respectively, represent the corresponding body effects coefficients for the implant layer and the substrate. Figure 21-1 shows the variation of vth as a function of vsb for Multi-Level Gamma.

Figure 21-1: Threshold Voltage Variation

The threshold voltage equations for different regions are as follows:

Channel Depletion Region is in the Implant Layer, vsb ≤ VBO

$$\gamma = \gamma i$$

$$vth = vbi + \gamma i \cdot (vsb + PHI)^{1/2}$$

$$vbi = VTO - \gamma i \cdot (PHI)^{1/2}$$

Channel Depletion Region is Expanded into the Bulk, vsb> VBO

$$\gamma = \gamma b$$

$$vth = vbi + \gamma b \cdot (vsb + PHI)^{1/2}$$

$$vbi = vtb - \gamma b \cdot (PHI)^{1/2}$$

In order for the threshold voltage to be continuous at vsb=VBO, vtb must be:

$$vtb = VTO + (\gamma i - \gamma b) \cdot [(VBO + PHI)^{1/2} - (PHI)^{1/2}]$$

The γi and γb are effective values of GAMMA and LGAMMA, respectively. The model computes them as γ in single-gamma models, except the scf factor is 1.0.

$$\gamma i = GAMMA \cdot gw \cdot gl$$

$$\gamma b = LGAMMA \cdot gw \cdot gl$$

Effective Built-in Voltage, vbi for VBO>0

 For vds ≤VFDS,

if vsb≤VBO,

$$vbi = VTO - \gamma i \cdot (PHI)^{1/2} + (\eta - 1) \cdot (PHI + vsb) - \frac{LDscaled}{Leff} \cdot VSH - \frac{\varepsilon si}{COX \cdot Leff} \cdot FDS \cdot vds$$

if vsb>VBO,

$$vbi = VTO - \gamma b \cdot (PHI)^{1/2} + (\gamma i - \gamma b) \cdot [(VBO + PHI)^{1/2} - (PHI)^{1/2}] + (\eta - 1)$$

$$+ (PHI + vsb) - \frac{LDscaled}{Leff} \cdot VSH - \frac{\varepsilon si}{COX \cdot Leff} \cdot FDS \cdot vds$$

For vds>VFDS,

if vsb≤VBO,

$$vbi = VTO - \gamma i \cdot (PHI)^{1/2} + (\eta - 1) \cdot (PHI + vsb) - \frac{LDscaled}{Leff} \cdot VSH - \frac{\varepsilon si}{COX \cdot Leff}$$

$$\cdot [(FDS - UFDS) \cdot VFDS + UFDS \cdot vds]$$

if vsb>VBO,

$$vbi = VTO - \gamma b \cdot (PHI)^{1/2} + (\gamma i - \gamma b) \cdot [(VBO + PHI)^{1/2} - (PHI)^{1/2}] + (\eta - 1)$$

$$\cdot PHI + vsb - \frac{LDscaled}{Leff} \cdot VSH - \frac{\varepsilon si}{COX \cdot Leff} \cdot [(FDS - UFDS) \cdot VFDS + UFDS \cdot vds]$$

Saturation Voltage, vdsat (UPDATE=0,2)

The saturation voltage due to channel pinch-off at the drain side is determined by:

$$vsat = \frac{vgs - vbi}{\eta} + \frac{1}{2}\left(\frac{\gamma}{\eta}\right)^2 \cdot \left\{ 1 - \left[1 + \left(\frac{2 \cdot \eta}{\gamma}\right)^2 \cdot \left(\frac{vgs - vbi}{\eta} + PHI + vsb\right) \right]^{1/2} \right\}$$

The reduction of saturation voltage due to the carrier velocity saturation effect is included as follows:

$$vdsat = vsat + vc - (vsat^2 + vc^2)^{1/2}$$

where vc 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.

$$vc = ECRIT \cdot Leff$$

or

$$vc = \frac{VMAX \cdot Leff}{ueff}$$

Because vsb>VBO, γ is switched from γi to γb, the ids, vsat, and conductances are not continuous. This problem is demonstrated in the following example. To correct the discontinuity problem, specify model parameter 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 IDS, VTH, VDSAT, GM, GDS and GMBS as a function of vsb 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 VBO, γ is set to γi and used in i_{ds} and vsat calculation. This is not correct; if (vds + vsb) exceeds VBO, the depletion regions at drain side expands into the substrate region, which means γb must be used instead of γi in v_{sat} computation. Since $v_{sat} = v_{gs} - v_{th}$ (drain), the threshold voltage at drain is computed using γi for vsb<VBO. 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 vsb<VBO to the region vsb>VBO.

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
+ 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 21-4: Variation of IDS, VTH and VDSAT for UPDATE=2

Figure 21-5: Variation of GM, GDS and GMBS 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 γi and γb are calculated:

$$vsat1 \; = \; \frac{vgs - vbi1}{\eta} + \frac{1}{2}\left(\frac{\gamma i}{\eta}\right)^2 \cdot \left\{1 - \left[1 + \left(\frac{2 \cdot \eta}{\gamma i}\right)^2 \cdot \left(\frac{vgs - vbi1}{\eta} + PHI + vsb\right)\right]^{1/2}\right\}$$

$$vsat2 \; = \; \frac{vgs - vbi2}{\eta} + \frac{1}{2}\left(\frac{\gamma b}{\eta}\right)^2 \cdot \left\{1 - \left[1 + \left(\frac{2 \cdot \eta}{\gamma b}\right)^2 \cdot \left(\frac{vgs - vbi2}{\eta} + PHI + vsb\right)\right]^{1/2}\right\}$$

vbi1 and vbi2 are built in potentials corresponding to γi and γb, respectively.

If (vdsat1 + vsb) ≤ VBO, then vdsat = vdsat1
If (vdsat2 + vsb) > VBO, then vdsat = vdsat2

> ***Note:*** *The vsat is modified by vc for carrier velocity saturation effects to obtain vdsat.*

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 vsb<VBO-vde, the model forms an entire gate depletion region in the implant layer.

$$ids = \beta \cdot \left\{ \left(vgs - vbi1 - \frac{\eta \cdot vde}{2} \right) \cdot vde - \frac{2}{3} \cdot \gamma i \cdot {}_{[}(PHI + vde + vsb)^{3/2} - (PHI + vsb)^{3/2}] \right\}$$

where vbi1 is the same as vbi for vsb≤VBO.

For vsb ≥ VBO, the entire gate depletion region expands into the bulk area.

$$ids = \beta \cdot \left\{ \left(vgs - vbi2 - \frac{\eta \cdot vde}{2} \right) \cdot vde - \frac{2}{3} \cdot \gamma b \cdot {}_{[}(PHI + vde + vsb)^{3/2} - (PHI + vsb)^{3/2}] \right\}$$

where vbi2 is the same as vbi for vsb>VBO.

$$ids = \beta \cdot \left\{ (vgs - vbi2 - \frac{\eta \cdot vde}{2}) \cdot vde - \frac{2}{3} \cdot yi \cdot [(VBO + PHI)^{3/2} - (vsb + PHI)^{3/2}] \right. $$
$$\left. + (\gamma i - \gamma b) \cdot (VBO + PHI)^{1/2} \cdot \left(VBO - vsb \right) \right\}$$

For VBO-vde<vsb<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:

$$e \ = \ \min(vds, \, vfa \cdot vsat) \quad , \, UPDATE\text{=}0$$

$$s \ = \ \min(vds/vfa, \, vsat) \quad , \, UPDATE\text{=}1,2$$

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 $\check{S}$ 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 ≤ vth - PHI

$$ids = 0$$

Weak Inversion, vth - PHI < vgs ≤ von

$$ids = ids(von, vde, vsb) \cdot \left(1 - \frac{von - vgs}{fast + PHI} \right)^{WEX}$$

Strong Inversion, vgs > von

$$ids = ids(vgs, vde, vsb)$$

Note: *The modified threshold voltage (von) is not used in strong inversion conditions.*

If WIC=3, the subthreshold current is calculated differently. In this case, the ids current is:

$$ids = ids(vgs, vde, vsb) + isub(N0eff, NDeff, vgs, vds)$$

The N0eff and NDeff are functions of effective device width and length.

Effective Mobility, ueff

All mobility equations have the general form:

$$ueff = UO \cdot factor$$

ueff	Effective mobility at analysis temperature.
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:

MOB 0	No mobility reduction (default)
MOB 1	Gm equation
MOB 2	Frohman-Bentchkowski equation
MOB 3	Normal field equation
MOB 4	Universal field mobility reduction
MOB 5	Universal field mobility reduction with independent drain field

MOB 0	No mobility reduction (default)
MOB 6	Modified MOB 3 equations (lateral field effect included)
MOB 7	Modified MOB 3 equations (lateral field effect not included)

These equations are described in the following sections.

MOB=0 Default, No Mobility

factor = 1.0No mobility reduction

MOB=1 Gm Equation

Name (Alias)	Units	Default	Description
F1	1/V	0.0	Gate field mobility reduction
UTRA (F3)	factor	0.0	Source-drain mobility reduction factor

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.

$$factor = \frac{1}{1 + F1 \cdot (vgs - vb1 - F3 \cdot vde)}$$

$$vde = \min(vds, vdsat)$$

Note: *If the alternate saturation model is used, vde is different for UPDATE=0 and UPDATE=1. See "Alternate DC Model, (ISPICE 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

Name (Alias)	Units	Default	Description
F1	V/cm	0.0	Critical gate-bulk electric field at which mobility reduction becomes significant
UEXP (F2)		0.0	Mobility exponent. Use 0.36 factor for n-channel and 0.15 for p¦-channel.
UTRA (F3)	factor	0.0	Source-drain mobility reduction factor
VMAX (VMX)	cm/s	0.0	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.

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.

$$factor = \left[\frac{F1 \cdot \varepsilon si}{COX \cdot (vgs - vbi - F3 \cdot vde)} \right]^{F2}$$

where vde is defined the same as for MOB=1 equation.

MOB=3 Normal Field Equation

Name (Alias)	Units	Default	Description
F1	1/V	0.0	Low-field mobility multiplier
F4		1.0	Mobility summing constant
UEXP (F2)		0.0	Mobility exponent
UTRA (F3)	1/V	0.0	High-field mobility multiplier
VF1	V	0.0	Low to high field mobility (voltage switch)

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

Name (Alias)	Units	Default	Description
ECRIT	V/cm	0.0	Critical electric drain field for mobility reduction. Use zero to indicate an infinite value.
F1	V/cm	0.0	Source-drain mobility reduction field (typical value 1e4 to 5e8)
MOB		0.0	Mobility equation selector. Set MOB=4 for critical field equation, or set MOB=5 for critical field equation with independent drain field.
UEXP (F2)	$1/V^{1/2}$	0.0	Bulk mobility reduction factor (typical value 0 to 0.5)
UTRA (F3)	V/cm	0.0	Critical electric drain field for mobility reduction

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.

$$factor = \frac{1}{1 + \dfrac{COX}{F1 \cdot \varepsilon ox} \cdot (vgs - cth) + \dfrac{vde}{vc} + F2 \cdot (vsb + PHI)^{1/2}}$$

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:

$$ids = \frac{ids}{1 + F1 \cdot \left(vgs - vth - \dfrac{vde}{2}\right) + \dfrac{UTRA}{Leff} \cdot vde}$$

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:

$$ids = \frac{ids}{1 - \dfrac{\Delta L}{Leff}}$$

ΔL 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

CLM = 2	Frohman's electrostatic fringing field equation
CLM = 3	One-sided step depletion layer drain field equation, with carrier velocity saturation
CLM = 4	Wang's equation: linearly graded depletion layer
CLM = 5	Avant!'s channel length modulation
CLM = 6	Avant!'s ΔL equations

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

Name (Alias)	Units	Default	Description
KL		0.0	Empirical constant (saturation voltage)
LAMBDA (LAM, LA)	cm/V$^{1/2}$	1.137e-4	Channel length modulation (–s calculated from NSUB unless specified) Default LAMBDA corresponds to default NSUB value

$$\Delta L = LAMBDA \cdot (vds - vdsat)^{1/2} \cdot \left(\frac{vdsat}{vsat}\right)^{KL}$$

If not user-specified, LAMBDA is calculated as:

$$LAMBDA = \left(\frac{2 \cdot \varepsilon si}{q \cdot DNB} \right)^{1/2}$$

This is a one-sided step depletion region formulation by Grove: ΔL varies with the depletion layer width, which is a function of the difference between the effective saturation voltage (vdsat) and the drain-to-source channel voltage (vds). This equation is typically used for long channels and high dopant concentrations. This corresponds to GDS=1 in MSINC.

CLM=2 Electrostatic Fringing Field

Name (Alias)	Units	Default	Description
A1		0.2	First fringing field factor gate-drain
A2		0.6	Second fringing field factor gate-vdsat

$$\Delta L = \frac{\varepsilon si}{COX} \cdot \frac{vds - vdsat}{A1 \cdot (vds - vgs + vbi) + A2 \cdot (vgs - vbi - vdsat)}$$

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

Name (Alias)	Units	Default	Description
KA		1.0	vds scaling factor for velocity saturation
KCL		1.0	Exponent for vsb scaling factor
KU		0.0	Velocity saturation switch. If KU ≤ 1, the standard velocity saturation equation is used.
LAMBDA (LAM, LA)	cm/V$^{1/2}$	1.137e-4	Channel length modulation. This parameter is calculated from NSUB if not specified. The default LAMBDA corresponds to the default NSUB value.
MAL		0.5	vds exponent for velocity saturation
MCL		1.0	Short channel exponent

$$\Delta L = vfu^{(2 \cdot MCL)} \cdot LAMBDA \cdot$$

$$[(vds - vfa \cdot vsat + KCL \cdot vsb + PHI)^{1/2} - (KCL \cdot vsb + PHI)^{1/2}]$$

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

Name (Alias)	Units	Default	Description
A1	m	0.2	Junction depth A1scaled = A1 Þ SCALM
DND	cm^{-3}	1e20	Drain diffusion concentration

Linearly Graded Depletion Layer

$$\Delta L = \left[\frac{2.73e5 \cdot A1scaled}{DNB \cdot \ln\left(\frac{DND}{DNB}\right)} \right]^{1/3} \cdot [(vds - vdsat + PHI)^{1/3} - PHI^{1/3}]$$

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

Name (Alias)	Units	Default	Description
LAMBDA	amp/V^2	0	Constant coefficient
VGLAM	1/V	0	Constant coefficient

When CLM=5, the current ids is increased by idssat, given as:

$$idssat = \frac{weff}{Leff} \cdot LAMBDA \cdot vds \cdot (vgs - vth) \cdot [1 + VGLAM \cdot (vgs - vth)]$$

$$ids = ids + idssat$$

> ***Note:*** *The equation adds the idssat term to ids in all regions of operation. Also, LAMBDA is a function of temperature.*

CLM=6, Star-Hspice ΔL Equation

Name (Alias)	Units	Default	Description
LAMBDA	1/V^{KL}	0	vds coefficient
LAM1	1/m	0	Channel length coefficient
KL		0	vds exponent
VGLAM	1/V	0	Gate drive coefficient

Unlike the other CLMs, this equation calculates the channel length modulation (ΔL) in all regions of operations and uses it to modify current ids.

$$\Delta L = \frac{Leff \cdot LAMBDA \cdot vds^{KL} \cdot [1 + VGLAM \cdot (vgs - vth)]}{1 + LAM1 \cdot Leff}$$

and:

$$ids = \frac{ids}{1 - \dfrac{\Delta L}{Leff}}$$

> ***Note:*** *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.

PHI	=	$1 \cdot \Phi_f$ (the Fermi potential)
TLEV	=	1
TLEVC	=	1

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,

$$\Phi_s = \frac{PHI}{2}, \text{ where } \Phi_s \text{ is the surface potential.}$$

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:

$$PHI(Level\ 7) = PHI(Level\ 6)/2$$

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

Name (Alias)	Units	Default	Description
LEVEL		1.0	IDS equation selector. Use LEVEL 8 for the advanced model using finite differences.
COX	F/m^2	3.45314e-4	Oxide capacitance per unit gate area. This parameter is calculated from TOX if not specified.
ECRIT (ESAT)	V/cm	0.0	Critical electric field for carrier velocity saturation, from Grove: electrons 6e4 holes 2.4e4 Use zero to indicate an infinite value.
SNVB	$1/(V \cdot cm^3)$	0.0	Slope of doping concentration versus vsb (element parameter). (Multiplied by 1e6)
TOX	m	1e-7	Oxide thickness
VMAX (VMX, VSAT)	m/s	0.0	Maximum drift velocity of carriers. Use zero to indicate an infinite value.

Effective Channel Width and Length Parameters

Name (Alias)	Units	Default	Description
DEL	m	0.0	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
LD (DLAT, LATD)	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 default=0.75 · XJ. LDscaled = LD · 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.
WD	m	0.0	Lateral diffusion into channel from bulk along width WDscaled = WD · SCALM
WDAC	m		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.
LMLT		1.0	Length shrink factor
LREF	m	0.0	Channel length reference LREFscaled = LREF · SCALM
WMLT		1.0	Diffusion layer and width shrink factor

Name (Alias)	Units	Default	Description
WREF	m	0.0	Channel width reference WREFscaled = WREF · SCALM
XJ	m	0.0	Metallurgical junction depth XJscaled = XJ · SCALM
XL (DL, LDEL)	m	0.0	Accounts for masking and etching effects XLscaled = XL · SCALM
XW (WDEL, DW)	m	0.0	Accounts for masking and etching effects XWscaled = XW · SCALM

Threshold Voltage Parameters

Name (Alias)	Units	Default	Description
CAV		0.0	Thermal voltage multiplier for the weak inversion equation
DELTA		0.0	Narrow width factor for adjusting threshold
ETA		0.0	Drain-induced barrier lowering (DIBL) effect coefficient for threshold voltage
GAMMA	$V^{1/2}$		Body effect factor. This parameter is calculated from NSUB if not specified (see "Common Threshold Voltage Parameters" on page 20-50).
LND	μm/V	0.0	ND length sensitivity
LN0	μm	0.0	N0 length sensitivity
ND	1/V	0.0	Drain subthreshold factor (typical value=1)
N0		0.0	Gate subthreshold factor (typical value=1)

Name (Alias)	Units	Default	Description
WIC		0.0	Sub-threshold model selector
WND	$\mu m/V$	0.0	ND width sensitivity
WN0	μm	0.0	N0 width sensitivity
NFS (DFS, NF, DNF)	$cm^{-2} \cdot V^{-1}$	0.0	Fast surface state density
NSUB (DNB, NB)	cm^{-3}	1e15	Bulk surface doping. This parameter is calculated from GAMMA if not specified.
PHI	V	0.576	Surface inversion potential. This parameter is calculated from NSUB if not specified (see "Common Threshold Voltage Parameters" on page 20-50).
VTO(VT)	V		Zero-bias threshold voltage. This parameter is calculated if not specified (see "Common Threshold Voltage Parameters" on page 20-50).

Mobility Parameters

Name (Alias)	Units	Default	Description
MOB		6.0	Mobility equation selector (can be set to 2, 3, 6, or 7 in LEVEL 8)
UCRIT	V/cm	1e4	MOB=6, UEXP>0 Critical field for mobility degradation, UEXP operates as a switch. MOB=6, UEXP≤0 Critical field for mobility degradation. Typical value is 0.01 V^{-1}.
UEXP (F2)		0.0	Critical field exponent in mobility degradation
UTRA	m/V	0.0	Transverse field coefficient (mobility)

Name (Alias)	Units	Default	Description
UO (UB, UBO)	$cm^2/$ $(V \cdot s)$	600 (N) 250 (P)	Low field bulk mobility. This parameter is calculated from KP (BETA) if KP (BETA) is input.

Channel Length Modulation Parameters

Name (Alias)	Units	Default	Description
A1		0.2	Channel length modulation exponent (CLM=8)
CLM		7	Channel length modulation equation selector
LAM1	1/m	0.0	Channel length modulation length correction
LAMBDA (LAM, LA)		0.0	Channel length modulation coefficient

LEVEL 8 Model Equations

This section lists the LEVEL 8 model equations.

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

$$ids = 0 \qquad \textit{(See subthreshold current)}$$

On Region, vgs>vth

$$ids = \beta \cdot \left\{ \left(vgs - vb1 - \frac{\eta \cdot vde}{2} \right) \cdot vde - \frac{2}{3} \cdot \gamma \cdot [(\Phi + vde + vsb)^{3/2} - (\Phi + vsb)^{3/2}] \right\}$$

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:

$$L_{eff} = Lscaled \cdot LMLT + XLscaled - 2 \cdot (LDscaled + DELscaled)$$

$$W_{eff} = M \cdot (Wscaled \cdot WMLT + XWscaled - 2 \cdot WDscaled)$$

$$LREF_{eff} = LREFscaled \cdot LMLT + XLscaled - 2 \cdot (LDscaled + DELscaled)$$

$$WREF_{eff} = M \cdot (WREFscaled \cdot WMLT + XWscaled - 2 \cdot WDscaled)$$

Effective Substrate Doping, nsub

Specify the model parameter SNVB to vary substrate doping concentration linearly as a function of vsb.

$$nsub = NSUB + SNVB \cdot vsb$$

The γ, Φ, and xd parameters are computed using the above equation for nsub.

$$\gamma = \frac{\sqrt{2 \cdot \varepsilon_{si} \cdot q \cdot nsub}}{COX}$$

$$\Phi = 2 \cdot vt \cdot \ln\left(\frac{nsub}{ni}\right)$$

$$xd = \sqrt{\frac{2 \cdot \varepsilon_{si}}{q \cdot nsub}}$$

If SNVB is zero, then $\gamma = $ GAMMA. The γ value is adjusted for short-channel effect the same way as the LEVEL 2 model. Also, Φ is calculated using NSUB.

Threshold Voltage, vth

Specify ETA to include the threshold voltage reduction due to potential barrier lowering effect.

$$vbi = VTO - \gamma \cdot \sqrt{\Phi} - \frac{8.14e\text{-}22 \cdot ETA}{COX \cdot L_{eff}^3} \cdot vds + (\eta - 1) \cdot (vsb + \Phi)$$

$$vth = vbi + \gamma \cdot \sqrt{vsb + \Phi}$$

The γ is modified for short-channel effect, the same as in the LEVEL 2 model, to get effective γ.

Saturation Voltage vdsat

The saturation voltage vsat 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,

$$vdsat = vsat + vc - \sqrt{vsat^2 + vc^2}$$

where:

$$vc = ECRIT \cdot L_{eff}$$

ECRIT ≤ 0 or MOB=7,

$$vdsat = vsat$$

vsat is computed as in the LEVEL=2 model (See).

Effective Mobility, ueff

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

$$u_{eff} = UO \cdot \left[\frac{\varepsilon_{se} \cdot UCRIT}{COX \cdot (vgs - vth - UTRA \cdot vde)} \right]^{UEXP}$$

MOB=3 Mobility Reduction

$$u_{eff} = \frac{UO}{1 + \dfrac{2.1e\text{-}8 \cdot (vgs + vth + egfet - \Phi)}{6 \cdot TOX}}$$

where *egfet* is the silicon energy gap at the analysis temperature.

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

where t is the temperature in degrees Kelvin.

If VMAX>1,

$$u_{eff} = \frac{u_{eff}}{1 + \dfrac{u_{eff}}{VMAX \cdot L_{eff}} \cdot vde}$$

MOB=6 Mobility Reduction

For UEXP>0,

If $\qquad (vgs - vth) > \dfrac{\varepsilon_{si} \cdot \text{UCRIT}}{\text{COX}}$

then $\qquad u_{eff} = \dfrac{UO \cdot \left[\dfrac{\varepsilon_{si} \cdot UCRIT}{COX \cdot (vgs - vth)}\right]^{UEXP}}{1 + \dfrac{UTRA}{L_{eff}} \cdot vde}$

otherwise, $u_{eff} = \dfrac{UO}{1 + \dfrac{UTRA}{L_{eff}} \cdot vde}$

For UEXP=0

$$u_{eff} = \frac{UO}{[1 + UCRIT \cdot (vgs - vth)] \cdot \left(1 + \dfrac{UTRA}{L_{eff}} \cdot vde\right)}$$

UCRIT for UEXP=0 has a dimension of (1/V).

MOB=7 Mobility Reduction

$$u_{eff} = \frac{UO}{1 + UTRA \cdot \left(vgs - vbi - \eta \cdot \dfrac{vde}{2} + \dfrac{body}{vde}\right)}$$

where:

$$body = \frac{2}{3} \cdot \gamma \cdot [(vde + vsb + \Phi)^{3/2} - (vsb + \Phi)^{3/2}]$$

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,

$$\lambda = \frac{xd}{leff \cdot vds} \cdot \sqrt{\frac{vds - vdsat}{4} + \sqrt{1 + \left(\frac{vds - vdsat}{4}\right)^2}}$$

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:

$$ids = \frac{ids}{1 - \dfrac{\Delta L}{L_{eff}}}$$

CLM=7 Intersil Channel Length Modulation

The ΔL is only computed for the saturation region.

vds > vdsat

$$\Delta L = \frac{LAMBDA \cdot L_{eff}}{1 + LAM1 \cdot L_{eff}} \cdot (vds - vdsat)$$

and:

$$ids = \frac{ids}{L - \dfrac{\Delta L}{L_{eff}}}$$

CLM=8

The ΔL is only computed for the saturation region.

vds > vdsat

$$\Delta L = \frac{L_{eff}}{1 + \dfrac{(1 + LAM1 \cdot L_{eff}) \cdot (1 + vde)^{A1}}{LAMBDA \cdot (vds - vde)}}$$

and:

$$ids = \frac{ids}{1 - \dfrac{\Delta L}{L_{eff}}}$$

Subthreshold Current Ids

The LEVEL 8 model has different subthreshold current equations, depending on the value of model parameter CAV.

Define:

$$fast = vt \cdot \left[\eta + \frac{q \cdot NFS}{COX} + \frac{\gamma}{2 \cdot (vsb + \Phi)^{1/2}} + \frac{\varepsilon_{si} \cdot q \cdot SNVB \cdot \sqrt{vsb + \Phi}}{\gamma \cdot COX^2} \right]$$

CAV≠0

$$von = vth + CAV \cdot fast$$

Subthreshold Region, vgs < von

If vgs>vth

$$ids = ids(von, vde, vsb) \cdot e^{\left(-1 - \frac{CAV}{2}\right)} \cdot e^{\left\{\left[\frac{1}{fast} - \frac{(CAV - 2) \cdot (vgs - vth)}{2 \cdot CAV^2 \cdot fast^2}\right](vgs - vth)\right\}}$$

If vgs≤vth

$$ids = ids(von, vde, vsb) \cdot e^{\left(-1 - \frac{CAV}{2}\right)} \cdot e^{\left(\frac{vgs - vth}{fast}\right)}$$

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

Name (Alias)	Units	Default	Description
LEVEL		1	MOSFET model level selector, set to 13 for the Star-Hspice BSIM model
CGBOM, (CGBO)	F/m	2.0e-10	Gate-to-bulk parasitic capacitance (F/m of length)
CGDOM, (CGDO)	F/m	1.5e-9	Gate-to-drain parasitic capacitance (F/m of width)
CGSOM, (CGSO)	F/m	1.5e-9	Gate-to-source parasitic capacitance (F/m of width)
DL0	μm	0.0	Difference between drawn poly and electrical
DW0	μm	0.0	Difference between drawn diffusion and electrical
DUM1		0.0	Dummy (not used)
DUM2		0.0	Dummy (not used)
ETA0		0.0	Linear vds threshold coefficient
LETA	mm	0.0	Length sensitivity

Name (Alias)	Units	Default	Description
WETA	μm	0.0	Width sensitivity
K1	$V^{1/2}$	0.5	Root-vsb threshold coefficient
LK1	$V^{1/2} \cdot$μm	0.0	Length sensitivity
WK1	$V^{1/2} \cdot$μm	0.0	Width sensitivity
K2		0.0	Linear vsb threshold coefficient
LK2	μm	0.0	Length sensitivity
WK2	μm	0.0	Width sensitivity
MUS	$cm^2/$ (V·s)	600	High drain field mobility
LMS (LMUS)	μm·cm^2 /(V·s)	0.0	Length sensitivity
WMS (WMUS)	μm·cm^2 /(V·s)	0.0	Width sensitivity
MUZ	$cm^2/$ (V·s)	600	Low drain field first order mobility
LMUZ	μm·cm^2 /(V·s)	0.0	Length sensitivity
WMUZ	μm·cm^2 /(V·s)	0.0	Width sensitivity
N0		0.5	Low field weak inversion gate drive coefficient (a value of 200 for N0 disables weak inversion calculation)
LN0		0.0	Length sensitivity
WN0		0.0	Width sensitivity
NB0		0.0	Vsb reduction to low field weak inversion gate drive coefficient
LNB		0.0	Length sensitivity

Name (Alias)	Units	Default	Description
WNB		0.0	Width sensitivity
ND0		0.0	Vds reduction to low field weak inversion gate drive coefficient
LND		0.0	Length sensitivity
WND		0.0	Width sensitivity
PHI0	V	0.7	Two times the Fermi potential
LPHI	V·μm	0.0	Length sensitivity
WPHI	V·μm	0.0	Width sensitivity
TREF	°C	25.0	Reference temperature of model (local override of TNOM)
TOXM, (TOX)	μm, (m)	0.02	Gate oxide thickness (TOXM or TOX > 1 is interpreted as Angstroms)
U00	1/V	0.0	Gate field mobility reduction factor
LU0	μm/V	0.0	Length sensitivity
WU0	μm/V	0.0	Width sensitivity
U1	μm/V	0.0	Drain field mobility reduction factor
LU1	$\mu m^2/V$	0.0	Length sensitivity
WU1	$\mu m^2/V$	0.0	Width sensitivity
VDDM	V	50	Critical voltage for high drain field mobility reduction
VFB0 (VFB)	V	-0.3	Flatband voltage
LVFB	V·μm	0.0	Length sensitivity
WVFB	V·μm	0.0	Width sensitivity
X2E	1/V	0.0	Vsb correction to linear vds threshold coefficient

Name (Alias)	Units	Default	Description
LX2E	μm/V	0.0	Length sensitivity
WX2E	μm/V	0.0	Width sensitivity
X2M (X2MZ)	$cm^2/$ $(V^2 \cdot s)$	0.0	Vsb correction to low field first order mobility
LX2M (LX2MZ)	μm$\cdot cm^2$ $/(V^2 \cdot s)$	0.0	Length sensitivity
WX2M (WX2MZ)	μm$\cdot cm^2$ $/(V^2 \cdot s)$	0.0	Width sensitivity
X2MS	$cm^2/$ $(V^2 \cdot s)$	0.0	Vbs reduction to high drain field mobility
LX2MS	μm$\cdot cm^2$ $/(V^2 \cdot s)$	0.0	Length sensitivity
WX2MS	μm$\cdot cm^2$ $/(V^2 \cdot s)$	0.0	Width sensitivity
X2U0	$1/V^2$	0.0	Vsb reduction to GATE field mobility reduction factor
LX2U0	μm/V^2	0.0	Length sensitivity
WX2U0	μm/V^2	0.0	Width sensitivity
X2U1	μm/V^2	0.0	Vsb reduction to DRAIN field mobility reduction factor
LX2U1	$\mu m^2/V^2$	0.0	Length sensitivity
WX2U1	$\mu m^2 /$ V^2	0.0	Width sensitivity
X3E	1/V	0.0	Vds correction to linear vds threshold coefficient
LX3E	μm/V	0.0	Length sensitivity
WX3E	μm/V	0.0	Width sensitivity

Name (Alias)	Units	Default	Description
X3MS	$cm^2/$ $(V^2 \cdot s)$	5.0	Vds reduction to high drain field mobility
LX3MS	$\mu m \cdot cm^2$ $/(V^2 \cdot s)$	0.0	Length sensitivity
WX3MS	$\mu m \cdot cm^2$ $/(V^2 \cdot s)$	0.0	Width sensitivity
X3U1	$\mu m/V^2$	0.0	Vds reduction to drain field mobility reduction factor
LX3U1	$\mu m^2/V^2$	0.0	Length sensitivity
WX3U1	$\mu m^2/V^2$	0.0	Width sensitivity
XPART		1.0	Selector for gate capacitance charge-sharing coefficient

Diffusion Layer Process Parameters

Name (Alias)	Units	Default	Description
CJW, (CJSW)	F/m	0.0	Zero-bias bulk junction sidewall capacitance
CJM, (CJ)	F/m^2	4.5e-5	Zero-bias bulk junction bottom capacitance
DS	m	0.0	Average variation of size due to side etching or mask compensation (not used)
IJS, (JS)	A/m^2	0	Bulk junction saturation current
JSW	A/m	0.0	Sidewall bulk junction saturation current
MJ0, (MJ)		0.5	Bulk junction bottom grading coefficient
MJW, (MJSW)		0.33	Bulk junction sidewall grading coefficient
PJ, (PB)	V	0.8	Bulk junction bottom potential

Name (Alias)	Units	Default	Description
PJW, (PHP)	V	0.8	Bulk junction sidewall potential
RSHM, (RSH)	ohm/sq	0.0	Sheet resistance/square
WDF	m	0.0	Default width of the layer (not used)

Note: *The wire model includes poly and metal layer process parameters.*

Basic Model Parameters

Name (Alias)	Units	Default	Description
LD (DLAT, LATD)	m		Lateral diffusion into channel from source and drain diffusion. If LD and XJ are unspecified, then 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
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 LREFscaled = LREF · SCALM

Name (Alias)	Units	Default	Description
WD	m	0.0	Lateral diffusion into channel from bulk along width WDscaled = WD · SCALM
WDAC	m		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.
WMLT		1.0	Diffusion layer and width shrink factor
XL (DL, LDEL)	m	0.0	Accounts for masking and etching effects XLscaled = XL · SCALM
XW (DW, WDEL)	m	0.0	Accounts for masking and etching effects XWscaled = XW · SCALM
WREF	m	0.0 *	Reference channel width WREFscaled = WREF · SCALM

Note: **If LREF 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

Name (Alias)	Units	Default	Description
BEX		-1.5	Temperature exponent for MUZ and MUS mobility parameters
FEX		0.0	Temperature exponent for mobility reduction factor U1
TCV	V/$^\circ$K	0.0	Flat-band voltage temperature coefficient
TREF	$^\circ$C	25	Temperature at which parameters are extracted. This parameter defaults to the option *TNOM*, which defaults to 25 $^\circ$C.

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.

$$A \; = \; A0 + LA \cdot \left(\frac{1}{Leff} - \frac{1}{LREFeff} \right) + WA \cdot \left(\frac{1}{Weff} - \frac{1}{WREFeff} \right)$$

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

$$VFB0 \; = \; -0.350v$$

$$LVFB \; = \; -0.1v\mu$$

$$WVFB \; = \; 0.08v \cdot \mu$$

$$Leff \; = \; 1 \cdot 10^{-6}m \; = \; 1\mu$$

$$Weff \; = \; 2 \cdot 10^{-6}m \; = \; 2\mu$$

$$LREFeff \; = \; 2 \cdot 10^{-6}m \; = \; 2\mu$$

$$WREFeff \; = \; 1 \cdot 10^{-5}m \; = \; 10\mu$$

$$zvfb \; = \; VFB0 + LVFB \cdot \left(\frac{1}{Leff} - \frac{1}{LREFeff} \right) + WVFB \cdot \left(\frac{1}{Weff} - \frac{1}{WREFeff} \right)$$

$$zvfb \; = \; -0.35v + -0.1v \cdot \mu \cdot \left(\frac{1}{1\mu} - \frac{1}{2\mu} \right) + 0.08v \cdot \mu \cdot \left(\frac{1}{2\mu} - \frac{1}{10\mu} \right)$$

$$zvfb \; = \; -0.35v - 0.05v + 0.032v$$

$$zvfb \; = \; -0.368v$$

.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:

Model Version	Effect of VERSION on BSIM model
9007B	LEVEL 13 BSIM model introduced: no changes
9007D	Removes the K2 limit
92A	Changes the TOX parameter default from 1000 A to 200 A
92B	Adds the K2LIM parameter, which specifies the K2 limit
93A	Introduces gds constraints
93A.02	VERSION parameter introduced
95.1	Fixes nonprinting TREF and incorrect GMBS problems
96.1	Flatband voltage temperature adjustment has been changed

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,

$$Leff = Lscaled \cdot LMLT - DL0 \cdot 1e\text{-}6$$

$$LREFeff = LREFscaled \cdot LMLT - DL0 \cdot 1e\text{-}6$$

Otherwise, if XL or LD is specified,

$$Leff = Lscaled \cdot LMLT + XLscaled - 2 \cdot LDscaled$$

$$LREFeff = LREFscaled \cdot LMLT + XLscaled - 2 \cdot LDscaled$$

If DW0 is specified, then

$$Weff = Wscaled \cdot WMLT - DW0 \cdot 1e\text{-}6$$

$$WREFeff = WREFscaled \cdot WMLT - DW0 \cdot 1e\text{-}6$$

Otherwise, if XW or WD is specified, then

$$Weff = Wscaled \cdot WMLT + XWscaled - 2 \cdot WDscaled$$

$$WREFeff = WREFscaled \cdot WMLT + XWscaled - 2 \cdot WDscaled$$

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, vgs ≤ vth

$ids = 0$ *(see subthreshold current)*

On Region, vgs > vth

For vds < vdsat, triode region:

$$ids = \frac{\beta}{1 + xu1 \cdot vds} \cdot \left[(vgs - vth) \cdot vds - \frac{body}{2} \cdot vds^2 \right]$$

For vds **Š**≥ vdsat, saturation region:

$$ids = \frac{\beta}{2 \cdot body \cdot arg} \cdot (vgs - vth)^2$$

where:

$$\beta = ueff \cdot COX \cdot \frac{Weff}{Leff}$$

$$ueff = \frac{uo}{1 + xu0 \cdot (vgs - vth)}$$

$$xu0 = zu0 - zx2u0 \cdot vsb$$

The carrier mobility, uo, is calculated by quadratic interpolation through three data points.

$$uo|_{vds = 0} = MUZ - zx2mz \cdot vsb$$

$$uo|_{vds = VDDM} = zmus - zx2ms \cdot vsb$$

and the sensitivity of uo to vds at vds=VDDM, 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:

$$arg = \frac{1}{2} \cdot [1 + vc + (1 + 2 \cdot vc)^{1/2}]$$

where:

$$vc = \frac{xu1 \cdot (vgs - vth)}{body}$$

and:

$$xu1 = zu1 - zx2u1 \cdot vsb + zx3u1 \cdot (vds - VDDM), \quad \textit{UPDATE=2}$$

$$xu1 = \frac{zu1 - zx2u1 \cdot vsb + zx3u1 \cdot (vds - VDDM)}{Leff}, \quad \textit{UPDATE=0, 1}$$

Threshold Voltage

The threshold voltage can be expressed as:

$$vth = zvfb + zphi + gamma \cdot (zphi + vsb)^{1/2} - xeta \cdot vds$$

where

$$gamma = zk1 - zk2 \cdot (zphi + vsb)^{1/2}$$

and:

$$xeta = zeta - zx2e \cdot vsb + zx3e \cdot (vds - VDDM), \quad UPDATE=0, 2$$

$$xeta = zeta + zx2e \cdot (zphi + vsb) + zx3e \cdot (vds - VDDM), \quad UPDATE=1$$

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:

$$Iexp = \beta_o \cdot vt^2 \cdot e^{1.8} \cdot e^{\frac{vgs - vth}{xn \cdot vt}} \cdot \left(1 - e^{-\frac{vds}{vt}}\right)$$

$$Ilim = 4.5 \cdot \beta_o \cdot vt^2$$

$$\beta_o = uo \cdot COX \cdot \frac{Weff}{Leff}$$

and:

$$xn = zn0 - znb \cdot vsb + znd \cdot vds$$

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:

$$r = RSH \cdot \frac{Leff}{Weff}$$

Capacitances:

$$c = COX \cdot Leff \cdot Weff + 2 \cdot CAPSW \cdot (Leff + Weff)$$

Temperature Effect

$$MUZ(t) = MUZ \cdot \left(\frac{t}{tnom}\right)^{BEX} \qquad UPDATE=0, 1$$

$$zmus(t) = zmus \cdot \left(\frac{t}{tnom}\right)^{BEX} \qquad UPDATE=0, 1$$

$$uo(t) = uo\left(\frac{t}{tnom}\right)^{BEX} \qquad UPDATE=2$$

$$xu1(t) = xu1 \cdot \left(\frac{t}{tnom}\right)^{FEX}$$

$$zvfb(t) = zvfb - \Delta t \cdot TCV$$

where:

$$\Delta t = t - tnom$$

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:

$$vtho = zvfb + zphi + zk1 \cdot (zphi + vsb)^{1/2}$$

$$cap = COX \cdot Leff \cdot Weff$$

$$vpof = \frac{vgs - vtho}{body}$$

$$argx = \frac{body \cdot vds}{12 \cdot (vgs - vtho - 0.5 \cdot body \cdot vds)}$$

If $(vgs - vtho - 0.5 \cdot body \cdot vds) \leq 1e\text{-}8$ then,

$$argx = \frac{1}{6}$$

$$argy = \frac{(vgs - vtho)^2 - 0.75 \cdot body \cdot (vgs - vtho) \cdot vds + 0.15 \cdot body^2 \cdot vds^2}{6 \cdot (vgs - vtho - 0.5 \cdot body \cdot vds)^3}$$

If $(vgs - vtho - 0.5 \cdot body \cdot vds) \leq 1e\text{-}8$ then,

$$argy = \frac{4}{15}$$

Regions Charge Expressions

Accumulation Region, vgs ≤ vtho, vgs ≤ zvfb - vsb

$$Qg = cap \cdot (vgs - zvfb + vsb)$$

$$Qb = -qg$$

$$Qs = 0$$

$$Qd = 0$$

Subthreshold Region, vgs ≤ vtho, vgs > zvfb - vsb

$$Qg = \frac{cap \cdot zk1}{2} \cdot \left\{ [(zk1)^2 + 4(vgs - zvfb + vsb)]^{1/2} - zk1 \right\}$$

$$Qb = -qg$$

$$Qs = 0$$

50/50 Channel-Charge Partitioning for Drain and Source, XPART=.5

Triode Region, vgs > vtho, vds ≤ vpof

$$Qg = cap \cdot (vgs - zvfb - zphi - 0.5 \cdot vds + vds \cdot argx)$$

$$Qb = cap \cdot [-vtho + zvfb + zphi + (1 - body) \cdot (0.5 - argx) \cdot vds]$$

$$Qd = -0.5 \cdot (qg + qb)$$

$$Qs = Qd$$

Saturation Region, vgs > vtho, vds > vpof

$$Qg \ = \ cap \cdot \left(vgs - zvfb - zphi - \frac{vgs - vtho}{3 \cdot body} \right)$$

$$Qb \ = \ cap \cdot \left[zvfb + zphi - vtho + (1 - body) \cdot \frac{(vgs - vtho)}{3 \cdot body} \right]$$

$$Qd \ = \ -\frac{cap}{3} \cdot (vgs - vtho)$$

$$Qs \ = \ Qd$$

40/60 Channel-Charge Partitioning for Drain and Source, XPART=0

Triode Region, vgs > vtho, vds ≤ vpof

$$Qg \ = \ cap \cdot (vgs - xvfb - zphi - 0.5 \cdot vds + argx \cdot vds)$$

$$Qb \ = \ cap \cdot [- vtho + zvfb + zphi + (1 - body) \cdot (0.5 - argx) \cdot vds]$$

$$Qd \ = \ -(cap \cdot [0.5 \cdot (vgs - vtho - body \cdot vds) + body \cdot argy \cdot vds])$$

$$Qs \ = \ -(Qg + Qb + Qd)$$

Saturation Region, vgs > vtho, vds > vpof

$$Qg \ = \ cap \cdot \left(vgs - zvfb - zphi - \frac{vgs - vtho}{3 \cdot body} \right)$$

$$Qb \ = \ cap \cdot \left[zvfb + zphi - vtho + (1 - body) \cdot \frac{(vgs - vtho)}{3 \cdot body} \right]$$

$$Qd \ = \ -\frac{4 \cdot cap}{15} \cdot (vgs - vtho)$$

$$Qs \ = \ \frac{3}{2} \cdot Qd$$

0/100 Channel-Charge Partitioning for Drain and Source, XPART=1

Triode Region, vgs > vtho, vds ≤ vpof

$$Qg = cap \cdot (vgs - zvfb - zphi - 0.5 \cdot vds + vds \cdot argx)$$

$$Qb = cap \cdot [-vtho + zvfb + zphi + (1 - body) \cdot (0.5 - argx) \cdot vds]$$

$$Qd = -(cap \cdot [0.5 \cdot (vgs - vtho) - body \cdot vds \cdot (0.75 - 1.5 \cdot argx)])$$

$$Qs = -(Qg + Qb + Qd)$$

Saturation Region, vgs > vtho, vds > vpof

$$Qg = cap \cdot \left(vgs - zvfb - zphi - \frac{vgs - vtho}{3 \cdot body}\right)$$

$$Qb = cap \cdot \left[zvfb + zphi - vtho + (1 - body) \cdot \frac{(vgs - vtho)}{3 \cdot body}\right]$$

$$Qd = 0$$

$$Qs = -Qg - Qb$$

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 *VBS* dependence. This feature is gained at the expense of some accuracy in the saturation region, particularly at high Vgs. 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, vdsat, and ids for a very simple model, with many parameters set to zero. There is no series resistance, RSH=0. Diode current has been turned off, JS=JSW=IS=0. The LEVEL 13 subthreshold current has been turned off by n0=200. The geometry parameters are set to zero, so Leff=L=1u, Weff=W=1u.

A value of TOX has been chosen to give:

$$cox = \frac{2.00000e-3F}{m^2}$$

The test is at vbs=-0.35, so that phi-vbs=1.0:

```
$ t1
.option ingold=2 numdgt=6
vd d 0 5
vg g 0 5
vb b 0 -0.35
m1 d g 0 b nch w=10u L=1u
.dc vd 4 5 1
.print ids=lx4(m1) vth=lv9(m1) vdsat=lv10(m1)
.model nch nmos LEVEL=13
+ vfb0=-0.4 lvfb=0 wvfb=0
+ phi0=0.65 lphi=0 wphi=0
+ k1=0.5 lk1=0 wk1=0
+ k2=0 lk2=0 wk2=0
+ eta0=1e-3 leta=0 weta=0
+ muz=600 mus=700 x3ms=10
+ xl=0 ld=0 xw=0 wd=0
+ u00=0 lu0=0 wu0=0
+ u1=0 lu1=0 wu1=0
+ tox=172.657
+ acm=2 rsh=0 js=0 jsw=0 is=0 n0=200
.end
```

Results from Star-Hspice

```
ids             vth             vdsat
1.09907e-02 7.45000e-01 3.69000e+00
```

Calculations at vgs=vds=5, vbs=-0.35

$$phi - vbs = 1$$

$$vth = -0.4 + 0.65 + (0.5 \cdot 1) - (ETA \cdot vds) = 0.75 - (0.001 \cdot vds) = 0.745$$

$$g = 1 - \frac{1}{(1.744 + 0.8364 \cdot 1)} = 0.612463$$

$$body = 1 + \frac{g \cdot 0.5}{(2 \cdot 1)} = 1 + 0.25 \cdot g = 1.153116$$

$$vc = 0 \; arg = 1$$

$$vdsat = \frac{(vgs - vth)}{body \cdot \sqrt{(arg)}} = \frac{(5 - 0.745)}{body} = 3.69000$$

At vds=VDDM (default VDDM=5), mobility=mus=700

$$ids = cox \cdot \left(\frac{Weff}{Leff}\right) \cdot 700 \cdot \frac{(vgs - vth)^2}{(2 \cdot body \cdot arg)}$$

$$ids = \left(\frac{10 \cdot 700 \cdot 4.255^2}{2 \cdot 1.15311 \cdot 1}\right) \cdot cox = 54953.36 \cdot cox$$

$$ids = 1.09907e - 2$$

These calculations agree with the Star-Hspice results given above.

Compatibility Notes

Model Parameter Naming

The following names are HSPICE-specific: U00, DL0, DW0, PHI0, 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>=0, where zx3ms=X3MS, with L, W adjustment

3. Check that

    zmuz+VDDM · 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

UC Berkeley SPICE 2, 3	Avant!'s Star-Hspice
VFB [V]	VFB0 (VFB)
PHI [V]	PHI0
K1 [$V^{1/2}$]	same
K2	same

Table 21-1: Comparison of Star-Hspice Parameters with UCB SPICE 2 and 3

UC Berkeley SPICE 2, 3	Avant!'s Star-Hspice
* ETA	ETA0
MUZ [cm^2/V·s]	same
* DL [μm]	DL0
* DW [μm]	DW0
* U0 [1/V]	U00
* U1 [μ/V]	same
X2MZ [cm^2/V^2·s]	X2M (X2MZ)
LX2MZ [μm·cm^2/V^2·s]	X2M (LX2MZ)
WX2MZ [μm·cm^2/V^2·s]	WX2M (WX2MZ)
X2E [1/V]	same
X3E [1/V]	same
X2U0 [1/V^2]	same
X2U1 [μm/V^2]	same
MUS [cm^2/V·s]	same
LMUS [μm·cm^2/V·s]	LMS (LMUS)
WMUS [μm·cm^2/V·s]	WMS (WMUS)
X2MS [cm^2/V^2·s]	same
X3MS [cm^2/V^2·s]	same
X3U1 [μm/V^2]	same
* TOX [μm]	TOXM[μ] (TOX[m])
* TEMP [$^\circ$C]	TREF
* VDD [V]	VDDM

**Table 21-1: Comparison of Star-Hspice Parameters with
UCB SPICE 2 and 3**

UC Berkeley SPICE 2, 3	Avant!'s Star-Hspice
CGDO [F/m]	CGDOM (CGDO)
CGSO [F/m]	CGSOM (CGSO)
CGBO [F/m]	CGBOM (CGBO)
XPART	same
N0	same
* NB	NB0
* ND	ND0
RSH [ohm/sq]	RSHM (RSH)
JS [A/m^2]	IJS (JS)
PB [V]	PJ (PB)
MJ	MJ0 (MJ)
* PBSW [V]	PJW (PHP)
MJSW	MJW (MJSW)
CJ [F/m^2]	CJM (CJ)
CJSW [F/m]	CCJW (CJSW)
* WDF [m]	–
* DELL [m]	–

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

$$xeta = zeta - (zx2e \cdot vsb) + zx3e \cdot (vds - VDDM)$$

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->-5,    VS=-5VB=0,3)
* P-CHANNEL LX8=GD (VD=0 to -5,VG=-5,        VS=-5,VB=0,3)
* P-CHANNEL LX9=GB (VD=0,      VG=0,         VS=-5,VB=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
```

```
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

```
VN6   5 36 0
VP6   0 46 0
M21   36 99   0 90 PC_NM1   8U 8U
M22   46 98 15 91 PC_PM1   8U 8U
```

GM B CVN7 5 37 0

```
VP7   0 47 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
```

First Model Parameter Format

```
*nmos 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-004,4.20E-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
```

```
+ 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, 1.91000E-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
*p+ diffusion layer
+140.0,4.0E-004,2.4E-010,1.00E-008,0.700E000
+0.8000e000,0.5,0.33,0,0
```

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
```

```
+ 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=-2.21153E-02
+ ETA0=2.11768E-03  LETA=3.04656E-04   WETA=-1.14155E-03
+ MUZ=4.93528E+02   DL0=5.39503E-02    DW0=4.54432E-01
+ U00=5.81155E-02   LU0=4.95498E-02    WU0=-1.96838E-02
+ U1=-5.88405E-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=4.84793E+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=-1.99620E+00
+ X3U1=-1.44106E-02 LX3U1=8.14508E-02  WX3U1=7.56591E-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=4.20E-010
+ IJS=1.00E-008     PJ=0.700E000
+ PJW=0.8000E000    MJ0=0.5            MJW=0.33
+ WDF=0             DS=0
```

PMOS Model

```
.MODEL PC_PM1 PMOS LEVEL=13
+ VFB0=-5.63441E-01 LVFB=-1.06809E-01  WVFB=1.32967E-01
+ PHI0=7.46390E-01  LPHI=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=-8.78447E-02
+ ETA0=-5.59772E-03 LETA=2.50199E-02   WETA=-5.66587E-04
+ MUZ=1.73854E+02   DL0=2.72457E-01    DW0=6.57818E-01
+ U00=1.26943E-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
```

```
+ X2E=1.55874E-03    LX2E=4.80364E-03    WX2E=-1.45355E-03
+ X3E=4.20214E-04    LX3E=-2.05447E-03   WX3E=-7.44369E-04
+ X2U0=1.00044E-02   LX2U0=-4.43607E-03  WX2U0=1.05796E-03
+ X2U1=-5.64102E-04  LX2U1=1.97407E-03   WX2U1=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-01
+ X3MS=1.11851E+00   LX3MS=9.75265E+00   WX3MS=-1.88238E+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
+ 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
*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 µm or 3 µm processing model.

Setting SOSLEV=1 selects the 5 µm model; otherwise the 3 µm model is automatically set, including the second-order effects (default=3 µm).

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=5μm. The automatic default=3μm.
pname	Parameter model

LEVEL 27 Model Parameters

5-μm Model Parameters

Name (Alias)	Units	Default	Description
CGDO	F/m		Gate-drain overlap capacitance per unit channel width. The default=3.1e-10 (n-type), 2.2e-10 (p-type).
CGSO	F/m		Gate-source overlap capacitance per unit channel width. The default=3.1e-10 (n-type), 2.2e-10 (p-type).
LD	m		Lateral diffusion. The default=0.6μ (n- type), 0.3μ (p-type).
RSH	ohm/ sq		Drain and source diffusion sheet resistance. The default=25 (n¦-type), 100 (p-type).
SOSLEV		1	Model index
TOX	m	7.0e-8	Oxide thickness
UO	cm^2/ (V·s)		Surface mobility. The default=350 (n-type), 220 (p-type).
VTO	V		Threshold voltage. The default=1.25 (n-type), -1.25 (p-type).

3-μm Model Parameters

Name (Alias)	Units	Default	Description
A	m/V	0.1μm	Channel length shortening coefficient (2nd effect)
ALPHA	V/m		Threshold voltage length dependence. The default=0.15μ (n-type), 0.18μ (p-type).
CAPOP		7	Capacitance model selector

Name (Alias)	Units	Default	Description
CGDO	F/m		Gate-drain overlap capacitance per unit channel width. The default=4.6e-10 (n-type), 3.6e-10 (p-type).
CGSO	F/m		Gate-source overlap capacitance per unit channel width. The default=4.6e-10 (n-type), 3.6e-10 (p-type).
EC	V/m		Critical electric field for velocity saturation (2nd effect). The default=3.0e6 (n-type), 7.5e6 (p-type).
FB			Body effect coefficient (2nd effect). The default=0.15 (n-type), 0 (p-type).
LD	m		Lateral diffusion. The default=0.3μ (n-type), 0.2μ (p-type).
LEVEL		27	Model level selector
RSH	ohm/ sq		Drain and source diffusion sheet resistance. The default=25 (n-type), 80 (p-type).
SOSLEV		2	Model index
THETA	1/V		Mobility degradation coefficient (2nd effect). The default=0.055 (n-type), 0.075 (p-type).
TOX	m	3.4e-8	Oxide thickness
UO	cm^2/ (V·s)		Surface mobility. The default=370 (n-type), 215 (p-type).
VTO	V		Threshold voltage. The default=0.83 (n-type), -0.74 (p-type).

Example

```
*FILE ML27IV.SP: IDS AND VGS CURVES FOR NMOS AND PMOS SOSFETS.
.OPTIONS ACCT LIST NOPAGE NOMOD
.OP
```

```
.DC VDDN 0 5.0 .1

* 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
```

```
 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
```

```
*
.MODEL P1 PMOS LEVEL=27 SOSLEV=2
+ VTO=-0.7212 TOX=0.34E-7 THETA=0.75E-1
+ FB=0.0 EC=0.75E7 A=0.1E-6
+ UO=215 CGSO=0.36E-9 CGDO=0.36E-9
+ RSH=80 LD=0.2E-6
*
.END
```

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 CJGATE=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.

Figure 21-6: Non-fully Depleted SOI Model

Example

```
ssoi.sp LEVEL=3 floating bulk model
** non-fully depleted
* test 1st order soi model with floating substrate
.option nomod post
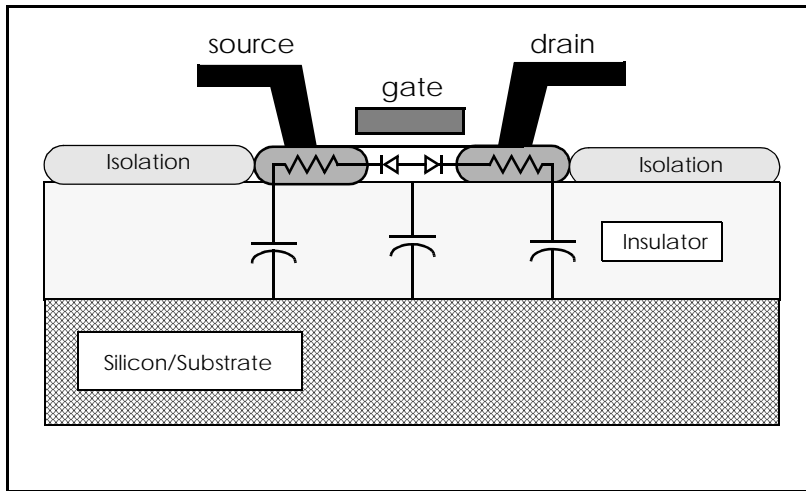
* substrate capacitance 3.45e-11 is for SiO2
.param t_sub_ox=.5u    subcap='3.45e-11/t_sub_ox' hdif=1.5u

.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
```

```
    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
 .end
```

Figure 21-7: LEVEL 3 Floating Bulk Model

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

Name (Alias)	Units	Default	Description
LEVEL		1	MOSFET model level selector. Set this parameter to 28 for this Star-Hspice model.
B1		0.0	Lower vdsat transition point
LB1	μm	0.0	Length sensitivity
WB1	μm	0.0	Width sensitivity
B2		1	Upper vdsat transition point
LB2	μm	0.0	Length sensitivity

Name (Alias)	Units	Default	Description
WB2	µm	0.0	Width sensitivity
CGBO	F/m	2.0e-10	Gate-to-bulk parasitic capacitance (F/m of length)
CGDO	F/m	1.5e-9	Gate-to-drain parasitic capacitance (F/m of width)
CGSO	F/m	1.5e-9	Gate-to-source parasitic capacitance (F/m of width)
ETA0		0.0	Linear vds threshold coefficient
LETA	µm	0.0	Length sensitivity
WETA	µm	0.0	Width sensitivity
ETAMN		0.0	Minimum linear vds threshold coefficient
LETAMN	µm	0.0	Length sensitivity
WETAMN	µm	0.0	Width sensitivity
GAMMN	$V^{1/2}$	0.0	Minimum root-vsb threshold coefficient
LGAMN	$V^{1/2} \cdot \mu m$	0.0	Length sensitivity
WGAMN	$V^{1/2} \cdot \mu m$	0.0	Width sensitivity
K1	$V^{1/2}$	0.5	Root-vsb threshold coefficient
LK1	$V^{1/2} \cdot \mu m$	0.0	Length sensitivity
WK1	$V^{1/2} \cdot \mu m$	0.0	Width sensitivity
K2		0.0	Linear vsb threshold coefficient
LK2	µm	0.0	Length sensitivity
WK2	µm	0.0	Width sensitivity
MUZ	$cm^2/V \cdot s$	600	Low drain field first order mobility
LMUZ	$\mu m \cdot cm^2/ V \cdot s$	0.0	Length sensitivity

Name (Alias)	Units	Default	Description
WMUZ	μm·cm$^{2/}$V·s	0.0	Width sensitivity
N0		200	Low field weak inversion gate drive coefficient (value of 200 for N0 disables weak inversion calculation)
LN0	μm	0.0	Length sensitivity
WN0	μm	0.0	Width sensitivity
NB0		0.0	Vsb reduction to low field weak inversion gate drive coefficient
LNB	μm	0.0	Length sensitivity
WNB	μm	0.0	Width sensitivity
ND0		0.0	Vds reduction to low field weak inversion gate drive coefficient
LND	μm	0.0	Length sensitivity
WND	μm	0.0	Width sensitivity
PHI0	V	0.7	Two times the Fermi potential
LPHI	V·μm	0.0	Length sensitivity
WPHI	V·μm	0.0	Width sensitivity
TOXM (TOX)	μm (m)	0.02	Gate oxide thickness (if TOXM or TOX > 1, Angstroms is assumed)
U00	1/V	0.0	Gate field mobility reduction factor
LU0	μm/V	0.0	Length sensitivity
WU0	μm/V	0.0	Width sensitivity
U1	1/V	0.0	Drain field mobility reduction factor
LU1	μm/V	0.0	Length sensitivity
WU1	μm/V	0.0	Width sensitivity

Name (Alias)	Units	Default	Description
VDDM	V	5.0	Critical voltage for high drain field mobility reduction
VFB0 (VFB)	V	-0.3	Flatband voltage
LVFB	V·μm	0.0	Length sensitivity
WVFB	V·μm	0.0	Width sensitivity
WFAC		4	Weak inversion factor
LWFAC	μm	0.0	Length sensitivity
WWFAC	μm	0.0	Width sensitivity
WFACU		0.0	Second weak inversion factor
LWFACU	μm	0.0	Length sensitivity
WWFACU	μm	0.0	Width sensitivity
X2E	1/V	0.0	Vsb correction to linear vds threshold coefficient
LX2E	μm/V	0.0	Length sensitivity
WX2E	μm/V	0.0	Width sensitivity
X2M (X2MZ)	$cm^2/V^2 \cdot s$	0.0	Vsb correction to low field first order mobility
LX2M (LX2MZ)	$\mu m \cdot cm^2/V^2 \cdot s$	0.0	Length sensitivity
WX2M (WX2MZ)	$\mu m \cdot cm^2/V^2 \cdot s$	0.0	Width sensitivity
X2U0	$1/V^2$	0.0	Vsb reduction to GATE field mobility reduction factor
LX2U0	$\mu m/V^2$	0.0	Length sensitivity
WX2U0	$\mu m/V^2$	0.0	Width sensitivity

Name (Alias)	Units	Default	Description
X2U1	$\mu m/V^2$	0.0	Vsb reduction to DRAIN field mobility reduction factor
LX2U1	$\mu m^2/V^2$	0.0	Length sensitivity
WX2U1	$\mu m^2/V^2$	0.0	Width sensitivity
X33M	$cm^2/V^2 \cdot s$	0.0	Gate field reduction of X3MS
LX33M	$\mu m \cdot cm^2/V^2 \cdot s$	0.0	Length sensitivity
WX33M	$\mu m \cdot cm^2/V^2 \cdot s$	0.0	Width sensitivity
X3E	$1/V$	0.0	Vds correction to linear vds threshold coefficient
LX3E	$\mu m/V$	0.0	Length sensitivity
WX3E	$\mu m/V$	0.0	Width sensitivity
X3MS	$cm^2/V^2 \cdot s$	5.0	Vds correction for high drain field mobility
LX3MS	$\mu m \cdot cm^2/V^2 \cdot s$	0.0	Length sensitivity
WX3MS	$\mu m \cdot cm^2/V^2 \cdot s$	0.0	Width sensitivity
X3U1	$1/V^2$	0.0	Vds reduction to drain field mobility reduction factor
LX3U1	$\mu m/V^2$	0.0	Length sensitivity
WX3U1	$\mu m/V^2$	0.0	Width sensitivity
XPART		1.0	Selector for gate capacitance charge sharing coefficient

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

Name (Alias)	Units	Default	Description
LD (DLAT, LATD)	m		Lateral diffusion into channel from source and drain diffusion. If LD and XJ are unspecified, the LD default=0.0. When LD is unspecified but XJ is specified, LD is calculated from XJ. The LD default=0.75 XJ. LDscaled = LD · 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	Reference channel length LREFscaled = LREF · SCALM

Name (Alias)	Units	Default	Description
XLREF	m	0.0	Difference between physical (on wafer) and drawn reference channel length XLREFscaled = XLREF · SCALM
WD	m	0.0	Lateral diffusion into channel from bulk along width WDscaled = WD · SCALM
WDAC	m		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.
WMLT		1.0	Diffusion layer and width shrink factor
XL (DL, LDEL)	m	0.0	Accounts for masking and etching effects XLscaled = XL · SCALM
XW (DW, WDEL)	m	0.0	Accounts for masking and etching effects XWscaled = XW · SCALM
WREF	m	0.0	Reference channel width WREFscaled = WREF · SCALM
XWREF	m	0.0	Difference between physical (on wafer) and drawn reference channel width XWREFscaled = XWREF · SCALM

Temperature Parameters

Name (Alias)	Units	Default	Description
BEX		-1.5	Temperature exponent for MUZ, X2M, X3MS, X33M mobility parameters
FEX		0.0	Temperature exponent for mobility reduction factor U1
TCV	V/°K	0.0	Flat-band voltage temperature coefficient

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

$$VFB0 = -0.350v$$

$$LVFB = -0.1v\mu$$

$$WVFB = 0.08v \cdot \mu$$

$$Leff = 1 \cdot 10^{-6}m = 1\mu$$

$$Weff = 2 \cdot 10^{-6}m = 2\mu$$

$$LREFeff = 2 \cdot 10^{-6}m = 2\mu$$

$$WREFeff = 1 \cdot 10^{-5}m = 10\mu$$

$$zvfb = VFB0 + LVFB \cdot \left(\frac{1}{Leff} - \frac{1}{LREFeff}\right) + WVFB \cdot \left(\frac{1}{Weff} - \frac{1}{WREFeff}\right)$$

$$zvfb = -0.35v + -0.1v \cdot \mu \cdot \left(\frac{1}{1\mu} - \frac{1}{2\mu}\right) + 0.08v \cdot \mu \cdot \left(\frac{1}{2\mu} - \frac{1}{10\mu}\right)$$

$$zvfb = -0.35v - 0.05v + 0.032v$$

$$zvfb = -0.368v$$

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,

$Leff = L+XL-2 \cdot LD \quad Weff = W+XW-2 \cdot WD$

Note: *If LDAC and WDAC are included in the .MODEL statement,*
Leff=L+XL-2·LDAC Weff=W+XW-2·WDAC

Syntax

```
Lscaled = L·SCALE
Wscaled = W·SCALE
XLscaled = XL·SCALM
LDscaled = LD·SCALM
XWscaled = XW·SCALM
WDscaled = WD·SCALM
Leff = Lscaled·LMLT+XLscaled−2·LDscaled
LREFeff = LREFscaled·LMLT+XLREFscaled−2·LDscaled
Weff = M·(Wscaled·WMLT+XWREFscaled−2·WDscaled)
WREFeff = M·(WREFscaled·WMLT+XWscaled−2·WDscaled)
```

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.

$xbs=(zphi-vbs)^{1/2}$

$xeta=zeta+zx2e \cdot vbs+zx3e \cdot vds$

vth=zvfb + zphi + zk1·xbs - zk2·xbs^2-xeta·vds

This equation is quadratic in xbs and vds. It is joined to linear equations at d(vth)/d(xbs) = zgammn and at d(vth)/d(vds) = -zetamn, which prevents the quadratics from going in the wrong direction.

Both gammn 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.

$$v_{gst} = v_{gs} - v_{th}$$

$$cx3ms = \frac{zx3ms}{(muz + zx33m \cdot v_{gst})}$$

$$m_{eff} = (zmuz + zx2m \cdot v_{bs})$$

$$\cdot (1 + cx3ms \cdot (VDDM + v_{ds} - (VDDM \cdot VDDM + v_{ds} \cdot v_{ds})^{1/2}))$$

$$xu0 = zu0 + zx2u0 \cdot v_{bs}$$

$$u_{eff} = \frac{m_{eff}}{(1 + xu0 \cdot v_{gst})}$$

$$beta = u_{eff} \cdot COX \cdot \frac{W_{eff}}{L_{eff}}$$

Saturation Voltage (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.

$$xbs = (zphi - v_{bs})^{1/2}$$

$$g = 1 - \frac{1}{(1.744 + 0.8364 \cdot xbs^2)}$$

$$body = \frac{1 + g \cdot zk1}{(2 \cdot xbs)}$$

$$xu1 = zu1 + vbs \cdot zx2u1$$

$$rx = (body^2 + zu1 \cdot 2 \cdot body \cdot v_{gst} + zx3u1 \cdot 4 \cdot v_{gst}^2)^{1/2}$$

$$v_{dsat} = \frac{2 \cdot v_{gst}}{(body + rx)}$$

This is the value of vds that makes the partial derivative of

$$f(v_{ds}, v_{gst}, v_{bs}) = (v_{gst} - body/2 \cdot v_{ds}) \cdot \frac{v_{ds}}{(1 + (xu1 + zx3u1 \cdot v_{ds}) \cdot v_{ds})}$$

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.

$$v1 = v_{dsat} - zb1 \cdot \frac{v_{dsat}}{1 + v_{dsat}}$$

$$v2 = v_{dsat} + zb2 \cdot v_{gst}$$

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, znb, 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,

$$Itotal = Istrong + Iweak \cdot \left(1 - exp\left(\frac{-v_{ds}}{vtherm}\right)\right)$$

In deep subthreshold,

$$xn = zn0 + znb \cdot v_{bs} + znd \cdot v_{ds}$$

$$vtherm = \frac{KT}{Q}$$

$$xweak = \frac{(v_{gs} - vt)}{(xn \cdot vtherm)}$$

$$Iweak = const \cdot exp(xweak)$$

The modification of this formula near threshold is controlled by zwfac and zwfacu. Just above threshold, the device is in saturation:

$$Istrong = const \cdot xweak^2$$

so Iweak needs an xweak^2 term to cancel the kink in gm at threshold. Then Iweak goes to zero for xweak>A0, which is at a small voltage above threshold. Iweak has four regions:

(1) xweak < -zwfac+A0

$$Iweak = const \cdot exp(xweak)$$

(2) -zwfac+A0 < xweak < 0

$$Iweak = const \cdot exp(xweak - const \cdot wf)$$

where wf is the integral with respect to xweak of

$$dwf = \frac{(xweak + zwfac - A0)^2}{[(1 + xweak + zwfac - A0)(1 + zwfacu \cdot (xweak + zwfac - A0))]}$$

(3) 0 < xweak < A0

$$Iweak = (same\ formula\ as\ in\ region\ 2) - const \cdot xweak^2$$

(4) A0 < xweak

$$Iweak = 0$$

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-vsbc < 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 γ 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

Name (Alias)	Units	Default	Description
LEVEL		1.0	Model level selector. This parameter is set to 38 for this model.
DNB (NSUB)	cm^{-3}	0.0	Surface doping density.

Name (Alias)	Units	Default	Description
DP	μm	1.0	Implant depth
ECV	V/μm	1000	Critical field
KCS		2.77	Implant capacitance integration constant
NI	cm^{-2}	2e11	Implant doping
PHI	V	0.8	Built-in potential
TOX	Å	0.0	Oxide thickness

Effective Width and Length Parameters

Name (Alias)	Units	Default	Description
DEL (WDEL)	m	0.0	Channel length reduction on each side
LATD (LD)	m	1.7 · XJ	Lateral diffusion on each side
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
OXETCH	μm	0.0	Oxide etch
WMLT		1.0	Diffusion layer and width shrink factor

Threshold Voltage Parameters

Name (Alias)	Units	Default	Description
FSS (NFS)	cm^{-2}·V^{-1}	0.0	Number of fast surface states
NWM		0.0	Narrow width modifier

Name (Alias)	Units	Default	Description
SCM		0.0	Short-channel drain source voltage multiplier
BetaGam		1.0	Body effect transition ratio
LBetaGam	µm	0.0	BetaGam dependence on channel length
WBetaGam	µm	0.0	BetaGam dependence on channel width
DVSBC	V	0.0	Empirical body effect transition voltage adjustment
LDVSBC	V·µm	0.0	L-dependent body effect transition voltage adjustment
WDVSBC	V·µm	0.0	W-dependent body effect transition voltage adjustment
TDVSBC	V/K	0.0	Body effect transition voltage shift due to temperature
VT (VTO)	V	0.0	Extrapolated threshold voltage
LVT (LVTO)	V·µm	0.0	VT dependence on channel length
WVT (WVTO)	V·µm	0.0	VT dependence on channel width
ETA		0.0	Channel-length independent drain-induced barrier lowering
LETA(DIBL)	µm	0.0	Channel-length dependent drain-induced barrier lowering
WETA	µm	0.0	Channel-width dependent drain-induced barrier lowering
DVIN	V	0.0	Empirical surface inversion voltage adjustment
XJ	µm	1.5	Junction depth

Mobility Parameters

Name (Alias)	Units	Default	Description
FRC	Å·s/cm^2	0.0	Field reduction coefficient
LFRC	10^{-4}Å·s/cm	0.0	FRC sensitivity to effective channel length
WFRC	10^{-4}Å·s/cm	0.0	FRC sensitivity to effective channel width
VFRC	Å·s/(cm^2·V)	0.0	Field reduction coefficient variation due to drain bias
LVFRC	10^{-4}Å·s/(cm·V)	0.0	VFRC sensitivity to effective channel length
WVFRC	10^{-4}Å·s/(cm·V)	0.0	VFRC sensitivity to effective channel width
BFRC	Å·s/(cm^2·V)	0.0	Field reduction coefficient variation due to substrate bias.
LBFRC	10^{-4}Å·s/(cm·V)	0.0	BFRC sensitivity to effective channel length
WBFRC	10^{-4}Å·s/(cm·V)	0.0	BFRC sensitivity to effective channel width
FSB	V$^{1/2}$· s/cm^2	0.0	Substrate bias-induced mobility degradation coefficient
LFSB	10^{-4}V$^{1/2}$· s/cm	0.0	FSB sensitivity to effective channel length
WFSB	10^{-4}V$^{1/2}$· s/cm	0.0	FSB sensitivity to effective channel width
UO (UB)	cm^2/(V·s)	600	Low field bulk mobility
LUO(LUB)	cm^2·µm/(V·s)	0.0	UO sensitivity to effective channel length

Name (Alias)	Units	Default	Description
WUO(WUB)	cm^2· μm/ (V·s)	0.0	UO sensitivity to effective channel width
FRCEX(F1EX)		0.0	Temperature coefficient for *FRC*
UH	cm^2/ (V·s)	900	Implant-channel mobility
KBeta1		1.0	Effective implant-channel mobility modifier
LKBeta1	μm	0.0	Length-dependent implant-channel mobility modifier
WKBeta1	μm	0.0	Width-dependent implant-channel mobility modifier
KI0(KIO)		1.0	Residue current coefficient
LKI0(LKIO)	μm	0.0	Length-dependent residue current coefficient
WKI0(WKIO)	μm	0.0	Width-dependent residue current coefficient
HEX(TUH)		-1.5	Implant channel mobility temperature exponent
BEX		-1.5	Surface channel mobility temperature exponent
VST	cm/s	0.0	Saturation velocity
UHSAT	μm/V	0.0	Implant-channel mobility saturation factor

Capacitance Parameters

Name (Alias)	Units	Default	Description
AFC		1.0	Area factor for MOSFET capacitance
CAPOP		6	Gate capacitance selector
METO	μm	0.0	Metal overlap on gate

LEVEL 38 Model Equations

The LEVEL 38 model equations follow.

IDS Equations

Depletion, vgs-vfb <0

$$ids = \beta 1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde + cav \cdot \left[(vgs - vfb) \cdot vde - \frac{vde^2}{2} \right] \right.$$

$$\left. -\frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Icrit \right\}$$

Enhancement, vgs-vfb Š vde >0

$$ids = \beta 1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde - \frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Icrit \right\}$$

$$+ \beta \cdot \left[(vgs - vfb) \cdot vde - \frac{vde^2}{2} \right]$$

Partial Enhancement, vgs-vfb<vde

$$ids = \beta 1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde + cav \cdot \left[(vgs - vfb) \cdot vde - \frac{vde^2}{2} \right] \right.$$

$$\left. -\frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Icrit \right\}$$

$$+ \left(\frac{1}{2}\beta - \frac{1}{2}\beta 1 \cdot cav \right) \cdot (vgs - vfb)^2$$

where:

$$\beta 1 = \frac{zKBeta1}{1 + UHSAT \cdot \dfrac{vde}{Leff}} \cdot UH \cdot \frac{Weff}{Leff}$$

$$\beta = UBeff \cdot cox \cdot \frac{Weff}{Leff}$$

$$cav = \frac{cox \cdot cs}{cox + cs}$$

$$cs = \frac{KCS \cdot \varepsilon_{Si}}{DP \cdot 1e\text{-}4}$$

$$Phid = vt \cdot \ln\left(\frac{DNB \cdot nd}{ni^2} \right)$$

$$nd = \frac{NI \cdot 1e4}{DP}$$

and:

$$vde = \min(vds, vdsat)$$

The temperature dependence of the mobility terms assume the ordinary exponential form:

$$UH(t) = UH(tnom) \cdot \left(\frac{t}{tnom}\right)^{TUH}$$

$$zUO(t) = zUO(tnom) \cdot \left(\frac{t}{tnom}\right)^{TUH}$$

The continuity term at the body effect transition point is given by

$$Icrit = -\frac{2}{3} \cdot cav \cdot [(vde + vsbc + Phid)^{3/2} - (vsbc + Phid)^{3/2}] \cdot \gamma \cdot \left(\frac{1}{zBetaGam} - 1\right)$$

for vsb>vsbc; $Icrit = 0$ otherwise.

The saturation voltage, threshold voltage, body effect transition voltage, and body effect coefficient $\overline{\gamma}$ 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:

$$vth = vfb - \beta d \cdot [vch - \overline{\gamma} \cdot (Phid + vsb)^{1/2} + vcrit]$$

where:

$$vfb = zVTO - zETA \cdot vds + \beta d \cdot (vch - \gamma_0 \cdot Phid^{1/2})$$

$$vcrit = \left(\gamma - \frac{\gamma}{zBetaGam}\right) \cdot (Phid + vsbc)^{1/2} \quad \text{for vsb > vsbc; 0 otherwise.}$$

$$\beta d = \frac{UH \cdot cav}{zUO \cdot cox}$$

$$vch = \frac{q \cdot NI}{cav}$$

$$\gamma_0 = \frac{(2 \cdot \varepsilon si \cdot q \cdot na1)^{1/2}}{cav}$$

$$na1 = \frac{nd \cdot DNB}{nd + DNB}$$

$$nd = \frac{NI}{DP \cdot 1e\text{-}4}$$

The effective $\bar{\gamma}$, including small device size effects, is computed as follows:

$$\bar{\gamma} = \frac{\gamma}{zBetaGam} \text{ for vsb>vsbc, and } = g \text{ otherwise.}$$

$$\gamma = \gamma_0 \cdot (1 - scf) \cdot (1 + ncf)$$

where:

If SCM ≤0,

$$scf = 0$$

otherwise,

$$scf = \frac{XJ}{Leff} \cdot \left\{ \left[1 + \frac{2xd}{XJ} \cdot (SCM \cdot vds + vsb + Phid)^{1/2} \right]^{1/2} - 1 \right\}$$

If NWM ≤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:

$$Vsbc = \frac{qDP^2}{2\varepsilon si}\left(\frac{NI}{DP \cdot 1e-4} - DNB\right) + zDVSBC + TDVSBC \cdot (t-tnom) - Phid$$

When vgs ≤ 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:

$$vinth = vfb - \frac{q \cdot NI}{cox} - vsb + DVIN - zETA \cdot vds$$

Saturation Voltage, vdsat

The saturation voltage vsat is determined by:

$$vsat = vgs - vfb + vch + \frac{\gamma^2}{2} \cdot \left\{ 1 - \left[1 + \frac{4}{\gamma^2} \cdot (vgs - vfb + vch + vsb + Phid) \right]^{1/2} \right\}$$

$$vdsat = vsat$$

Star-Hspice modifies vsat to include carrier velocity saturation effect:

$$vdsat \ = \ vsat + vc - (vsat^2 + vc^2)^{1/2}$$

where:

$$vc \ = \ ECV \cdot Leff$$

Mobility Reduction, UBeff

The surface mobility UB is dependent upon terminal voltages as follows:

$$UBeff \ = \ \cfrac{1}{\cfrac{1}{zUO} + \cfrac{(zFRC + zVFRC \cdot vde + zBFRC \cdot vsb) \cdot (vgs - vfb)}{TOX} + \cfrac{vde}{VST \cdot Le} + zFSB \cdot vsb^{1/2}}}$$

where:

$$Le \ = \ Leff \qquad \textit{Linear region}$$

$$Le \ = \ Leff - \Delta L \quad \textit{Saturation region}$$

and at elevated temperatures

$$zFRC(t) \ = \ zFRC(tnom) \cdot \left(\frac{t}{tnom}\right)^{FRCEX}$$

The ΔL 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:

$$ids = \frac{ids}{1 - \dfrac{\Delta L}{Leff}}$$

where:

$$\Delta L = 1e4 \cdot \left[\frac{2.73e3 \cdot XJ}{na1 \cdot \ln\left(\dfrac{1e20}{na1}\right)} \right]^{1/3} \cdot [(vds - vdsat + PHI)^{1/3} - PHI^{1/3}]$$

The ΔL is in microns, assuming XJ is in microns and na1 is in 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:

$$von = max(vth, vinth) + fast$$

where:

$$fast = vt \cdot \left[1 + \frac{q \cdot FSS}{cox} + \frac{\gamma}{2 \cdot (Phid + vsb)^{1/2}} \right]$$

If vgs <von, then

Partial Enhancement, 0< vgs-vfb < vde

$$ids = \beta 1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde + cav \cdot \left[(von - vfb) \cdot vde - \frac{vde^2}{2} \right] \right.$$

$$\left. - \frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Icrit \right\}$$

$$+ \frac{1}{2} \cdot \left(\beta \cdot e^{\frac{vgs - von}{fast}} - \beta 1 \cdot cav \right) \cdot (von - vfb)^2$$

Full Enhancement, vgs-vfb Š-vde > 0

$$ids = \beta 1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde - \frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Icrit \right\}$$

$$+ \beta \cdot \left[(von - vfb) \cdot vde - \frac{vde^2}{2} \right] \cdot e^{\frac{vgs - von}{fast}}$$

Depletion, vgs-vfb Š< 0

$$ids = \beta 1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde + cav \cdot \left[(vgs - vfb) \cdot vde - \frac{vde^2}{2} \right] \right.$$

$$\left. - \frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Icrit \right\} \cdot e^{\frac{vgs - von}{fast}}$$

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=-0.01 WFSB=5e-5    WVFRC=-0.00
+ WBFRC=-0.4e-3
+ KI0=.9KBETA1=.5
+ LKI0=0.16LKBETA1=-0.15
+ WKI0=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=8000UHSAT=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)[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

Name (Alias)	Units	Default	Description
TOX	m	0.02	Gate oxide thickness. (TOX > 1 is assumed to be in Angstroms)
TEMP	C	-	NOT USED IN Star-Hspice (see the following compatibility notes)
VDD	V	5	Drain supply voltage (NMOS convention)
VGG	V	5	Gate supply voltage (NMOS convention)
VBB	V	-5	Body supply voltage (NMOS convention)

Name (Alias)	Units	Default	Description
DL	μ	0	Channel length reduction
DW	μ	0	Channel width reduction
VGHIGH	V	0	Upper bound of the weak-strong inversion transition region
VGLOW	V	0	Lower bound of same
VFB	V	-0.3	Flat band voltage
PHI	V	0.8	Surface potential
K1	V^{-1}	0.5	Body effect coefficient
K2	-	0	Second order body effect coefficient (for nonuniform channel doping)
ETA0	-	0	Drain-induced barrier lowering coefficient.
ETAB	V^{-1}	0	Sensitivity of drain-induced barrier lowering coefficient to V_{bs}
MU0	$cm^2/V \cdot s$	400	Low-field mobility
MU0B	$cm^2/V^2 \cdot s$	0	Sensitivity of low-field mobility to V_{bs}
MUS0	$cm^2/V \cdot s$	600	High drain field mobility
MUSB	$cm^2/V^2 \cdot s$	0	Sensitivity of high drain field mobility to V_{bs}
MU20	-	0	Empirical parameter for output resistance
MU2B	V^{-1}	0	Sensitivity of empirical parameter to V_{bs}
MU2G	V^{-1}	0	Sensitivity of empirical parameter to V_{gs}
MU30	$cm^2/V^2 \cdot s$	0	Empirical parameter for output resistance
MU3B	$cm^2/V^3 \cdot s$	0	Sensitivity of empirical parameter to V_{bs}

Name (Alias)	Units	Default	Description
MU3G	$cm^2/$ $V^3 \cdot s$	0	Sensitivity of empirical parameter to V_{gs}
MU40	$cm^2/$ $V^3 \cdot s$	0	Empirical parameter for output resistance
MU4B	$cm^2/$ $V^4 \cdot s$	0	Sensitivity of empirical parameter to V_{bs}
MU4G	$cm^2/$ $V^4 \cdot s$	0	Sensitivity of empirical parameter to V_{gs}
UA0	V^{-1}	0	First-order vertical-field mobility reduction factor
UAB	V^{-2}	0	Sensitivity of first-order factor to V_{bs}
UB0	V^{-2}	0	Second-order vertical-field mobility reduction factor
UBB	V^{-3}	0	Sensitivity of second-order factor to V_{bs}
U10	V^{-1}	0	High drain field (velocity saturation) mobility reduction factor
U1B	V^{-2}	0	Sensitivity of mobility reduction factor to V_{bs}
U1D	V^{-2}	0	Sensitivity of mobility reduction factor to V_{ds}
N0	-	0.5	Subthreshold swing coefficient
NB	$V^{1/2}$	0	Sensitivity of subthreshold swing to V_{bs}
ND	V^{-1}	0	Sensitivity of subthreshold swing to V_{ds}
VOF0	-	0	Threshold offset (normalized to NKT/q) for subthreshold.
VOFB	V^{-1}	0	Sensitivity of offset to V_{bs}.
VOFD	V^{-1}	0	Sensitivity of offset to V_{ds}.
AI0	-	0	Impact ionization coefficient.

Name (Alias)	Units	Default	Description
AIB	V^{-1}	0	Sensitivity of impact ionization coefficient to V_{bs}.
BI0	V	0	Impact ionization exponent.
BIB	-	0	Sensitivity of impact ionization exponent to V_{bs}.
DELL	m	-	Length reduction of source drain diffusion. NOT USED IN Star-Hspice!
WDF	m	-	Default width. NOT USED IN Star-Hspice. Use ".OPTION DEFW=#" in the netlist instead.

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

Name (Alias)	Units	Default	Description
CGDO	F/m	-	Gate-drain overlap capacitance. Calculated if not specified and if LD or METO, and TOX are.
CGSO	F/m	-	Gate-source overlap capacitance. This parameter is calculated if not specified and if LD or METO, and TOX are.

Name (Alias)	Units	Default	Description
CGBO	F/m	-	Gate-bulk overlap capacitance. This parameter is calculated if not specified and if WD and TOX are.
RSH	ohm/sq	0	Source/drain sheet resistance.
JS	A/m^2	0	Source/drain bulk diode reverse saturation current density.
PB	V	0.8	Source/drain bulk junction potential.
PBSW	V	PB	Sidewall junction potential
CJ	F/m^2	0	Source/drain bulk zero-bias junction capacitance
CJSW	F/m	0	Sidewall junction capacitance
MJ	-	0.5	Source/drain bulk junction grading coefficient
MJSW		0.33	Sidewall junction grading coefficient

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

Name (Alias)	Units	Default	Description
ACM	-	0	MOS S/D parasitics selector. ACM=0 is SPICE style. ACM=2 or 3 is recommended for LDD.
SPICE3	-	0	SPICE3 model compatibility selector. For accurate SPICE3 BSIM2, set SPICE3=1.
DERIV	-	0	Derivative selector: DERIV=0 $\Rightarrow$ analytic. DERIV=1 $\Rightarrow$ finite difference
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 Leff calculation for AC gate capacitance.

Name (Alias)	Units	Default	Description
XW	m	0	Difference between physical (on wafer) and drawn S/D active width. This parameter is used for W_{eff} calculation only if DW=0. $XW_{scaled} = XW \cdot SCALM$
WMLT	-	1.0	Diffusion and gate width shrink factor
WD	m	0	Channel stop lateral diffusion under gate (per side). This parameter is used for W_{eff} calculation only if DW=0. WD scaled=WD $\cdot$ SCALM
WDAC	m		This parameter is the same as WD, but if WDAC is included in the .MODEL statement, it replaces WD in the Weff calculation for AC gate capacitance.
LREF	m	0 (∞)	Reference channel length for length adjustment of BSIM model parameters. For Berkeley compatibility (LREF-> ∞), use LREF=0. $LREF_{scaled} = LREF \cdot SCALM$
XLREF	m	0.0	Difference between physical and drawn reference channel length
WREF	m	0 (∞)	Reference device width for width adjustment of BSIM model parameters. For Berkeley compatibility (WREF-> ∞), use WREF=0. $WREF_{scaled} = WREF \cdot SCALM$
XWREF	m	0.0	Difference between physical and drawn reference channel width
DELVTO	V	0	Threshold voltage shift. This parameter is "type" sensitive. For example, DELVTO>0 increases the magnitude of n-channel threshold and decreases the magnitude of p-channel threshold. It adds to the element-line DELVTO parameter.

Name (Alias)	Units	Default	Description
ALPHA	V^{-1}	0	Impact ionization coefficient. This parameter has associated geometry sensitivity parameters. Choose between BSIM2 (A10>0 and HSPICE (ALPHA>0) impact ionization modeling. D*o not use both.*
VCR	V	0	Impact ionization critical voltage. This parameter has associated geometry sensitivity parameters.
IIRAT	-	0	Impact ionization source bulk current partitioning factor. One corresponds to 100% source. Zero corresponds to 100% bulk.
TCV	V/C	0	Zero-bias threshold voltage temperature coefficient. The sign of TCV is adjusted automatically for NMOS and PMOS to make threshold decrease in magnitude with rising temperature.
BEX	-	-1.5	Temperature exponent for mobility
FEX	-	0	Temperature exponent for velocity saturation
Px	*[x]*·μ μ^2	0	P*x* is Avant!'s proprietary WL-product sensitivity parameter for *x*, where *x* is a model parameter with length and width sensitivity.

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:

$$V_{dsat} = \frac{V_{gs} - V_{th}}{a\sqrt{K}},$$

$$K = \frac{1 + V_c + \sqrt{1 + 2V_c}}{2},$$

$$V_c = \frac{U_{1S}(V_{gs} - V_{th})}{a[1 + UA(V_{gs} - V_{th}) + UB(V_{gs} - V_{th})^2]},$$

$$U_{1S} = U10 + U1B \cdot V_{bs},$$

$$U1 = U_{1S}\left[1 - \Theta(V_{dsat} - V_{ds})\frac{U1D(V_{ds} - V_{dsat})^2}{V_{dsat}^2}\right]$$

where $\Theta(x)$ is the usual unit step function,

$$\beta' = \beta_0 + \beta_1 \tanh\left(MU2\frac{V_{ds}}{V_{dsat}}\right) + \beta_3 V_{ds} - \beta_4 V_{ds}^2$$

$$\beta_0 = \frac{W_{eff}}{L_{eff}} MU \cdot C_{ox},$$

$$\beta_1 = \beta_S - (\beta_0 + \beta_3 VDD - \beta_4 VDD^2),$$

$$\beta_i = \frac{W_{eff}}{L_{eff}} MUi \cdot C_{ox}, \quad i = S, 3, 4,$$

$$a = 1 + \frac{gK1}{2\sqrt{PHI - V_{bs}}},$$

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

$$f = AI \cdot e^{\frac{-BI}{V_{ds} - V_{dsat}}}.$$

Weak Inversion ($V_{gs} < V_{th} + VGLOW$; [VGLOW<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 $V_{tm} = \dfrac{kT}{q}$ and $N = N0 + \dfrac{NB}{\sqrt{PHI - V_{bs}}} + ND \cdot V_{ds}$

Strong inversion-to-weak inversion transition region (V_{th} +VGLOW$\leq V_{gs} \leq \varsigma_{th}$ +VGHIGH):

$$V_{geff}(V_{gst}) = \sum_{j=0}^{3} C_j V_{gst}^j$$

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}=VGLOW, the saturation equation is assumed to be valid for all V_{ds} (that is,
$V_{dsat}(V_{geff}(\text{VGLOW})) \approx 0$), to allow a match to the subthreshold equation given above. The coefficients Cj 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 $V_{gs} = V_{th} + \text{VGLOW}$ and $V_{gs} = V_{th} + \text{VGHIGH}$.

Effective Length and Width

If DL is nonzero:

$$L_{eff} = L_{scaled} \cdot LMLT - DL$$

$$LREF_{eff} = LREF_{scaled} \cdot LMLT - DL$$

Otherwise,

$$L_{eff} = L_{scaled} \cdot LMLT + XL_{scaled} - 2 \cdot LD_{scaled}$$

$$LREF_{eff} = LREF_{scaled} \cdot LMLT + XLREF_{scaled} - 2 \cdot LD_{scaled}$$

If DW is nonzero:

$$W_{eff} = (W_{scaled} \cdot WMLT - DW) \cdot M$$

$$WREF_{eff} = (WREF_{scaled} \cdot WMLT - DW) \cdot M$$

Otherwise,

$$W_{eff} = (W_{scaled} \cdot WMLT + XW - 2 \cdot WD_{scaled}) \cdot M$$

$$WREF_{eff} = (WREF_{scaled} \cdot WMLT + XWREF_{scaled} - 2 \cdot WD_{scaled}) \cdot M$$

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 WP, LP, and PP, respectively. The value of the parameter P' adjusted for width, length, and WL-product is:

$$P' = P + WP \cdot \left(\frac{1}{W_{eff}} - \frac{1}{WREF_{eff}} \right) + LP \cdot \left(\frac{1}{L_{eff}} - \frac{1}{LREF_{eff}} \right)$$
$$+ PP \cdot \left(\frac{1}{W_{eff}} - \frac{1}{WREF_{eff}} \right) \cdot \left(\frac{1}{L_{eff}} - \frac{1}{LREF_{eff}} \right)$$

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:

Model Version	Effect of VERSION on BSIM2 Model
92A	LEVEL 39 BSIM2 model introduced: no changes
92B	No changes
93A	Introduces gds constraints, fixes WMU3B parameter defect, and introduces MU4 parameter defect
93A.02	VERSION parameter introduced, fixes MU4 parameter defect
95.1	Fixes defects that cause PMUSB, LDAC, WDAC parameter problems, fixes GMBS defect when gds constraints are used
96.1	Limited ETA + ETAB · vb5≥0

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:

$MU2 \geq 0$ $ND \geq 0$ $AI \geq 0$

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 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.

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. $Q_s = -(Q_g+Q_d+Q_b)$ in all regions.

Accumulation region ($V_{gs}<V_{bs}+VFB$):

$$Q_g = C_{ox}W_{eff} \cdot L_{eff}(V_{gs} - V_{bs} - VFB)$$

$$Q_b = -Q_g$$

$$Q_d = 0$$

Subthreshold region ($V_{bs} + VFB<V_{gs}<V_{th} + VGLOW$):

$$Q_g = C_{ox}W_{eff} \cdot L_{eff}(V_{gs} - V_{bs} - VFB)$$

$$\cdot \left[1 - \frac{V_{gs} - V_{bs} - VFB}{V_{gs} - V_{bs} - VFB - V_{gst}} + \frac{1}{3}\left\{ \frac{V_{gs} - V_{bs} - VFB}{V_{gs} - V_{bs} - VFB - V_{gst}} \right\}^2 \right]$$

$$Q_b = -Q_g$$

$$Q_d = 0$$

Saturation region ($V_{ds} > V_{dsat}$):

$$Q_g = \frac{2}{3} C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst} + Q_{bulk}$$

where:

$$Q_{bulk} = \frac{1}{3} C_{ox} W_{eff} \cdot L_{eff} [V_{th} - V_{bs} - VFB]$$

$$Q_b = -Q_{bulk}$$

$$Q_d = -\frac{4}{10} \cdot \frac{2}{3} C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst} = \left(-\frac{4}{15}\right) C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst}$$

Linear region ($V_{ds} < V_{dsat}$):

$$Q_g = \frac{2}{3} C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst} \cdot \left[\frac{3\left(1 - \frac{V_{ds}}{V_{dsat}}\right) + \left(\frac{V_{ds}}{V_{dsat}}\right)^2}{2 - \frac{V_{ds}}{V_{dsat}}} \right] + Q_{bulk}$$

$$Q_b = -Q_{bulk}$$

$$Q_d = -\frac{1}{3} C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst}$$

$$\cdot \left[\frac{3\left(1 - \frac{V_{ds}}{V_{dsat}}\right) + \left(\frac{V_{ds}}{V_{dsat}}\right)^2}{2 - \frac{V_{ds}}{V_{dsat}}} + \frac{\frac{V_{ds}}{V_{dsat}}\left(1 - \frac{V_{ds}}{V_{dsat}}\right) + 0.2\left(\frac{V_{ds}}{V_{dsat}}\right)^2}{\left(2 - \frac{V_{ds}}{V_{dsat}}\right)^2} \right] + Q_{bulk}$$

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:

$$V_{th}(T) = V_{bi}(T) + K1 \cdot \sqrt{\phi(T) - V_{bs}} - K2 \cdot (\phi(T) - V_{bs}) - ETA \cdot V_{ds}$$

where:

$$V_{bi}(T) = V_{to}(T) - K1 \cdot \sqrt{\phi(T)} + K2 \cdot \phi(T) \ ,$$

$$V_{to}(T) = V_{to} - TCV \cdot (T - T_{nom}) \ ,$$

and the nominal-temperature, zero-bias threshold voltage is given by

$$\begin{aligned} V_{to} &= V_{bi} + K1 \cdot \sqrt{PHI} - K2 \cdot PHI \\ &= VFB + PHI + K1 \cdot \sqrt{PHI} - K2 \cdot PHI, \end{aligned}$$

and $\phi(T)$ is calculated according to the value of TLEVC as specified.

Mobility is adjusted according to

$$\mu(T) \ = \ \mu(T_{nom}) \cdot \left(\frac{T}{T_{nom}}\right)^{BEX} \text{where} \ \ \mu \ = \ \frac{\beta'}{C_{ox}(W_{eff}/L_{eff})}.$$

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 13-8and "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 $V_{ds} = 2 \cdot VDD$, $V_{gs} = 2 \cdot VGG$, and $V_{bs} = 2 \cdot VBB$).

To avoid drain current discontinuity at $V_{ds} = V_{dsat}$, be sure that BI $\neq$ if AI0 $\neq$ 0.

To prevent negative g_{ds}, be sure that ETA > 0 and that MU3 > 0 and MU4 < MU3 / (4 * VDD). 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, VGLOW $\leq$ 0, and VGHIGH $\geq$ 0.

The BSIM2 gate capacitance model of SPICE 3E tends to display negative C_{gs} in subthreshold. This appears to be due to $C_{gg} \rightarrow 0$ as $V_{gs} \rightarrow V_{th}$ by construction of the gate charge equation, so that $C_{gs} = C_{gg} - C_{gd} - C_{gb} \rightarrow - C_{gd} - C_{gb} \approx - C_{gb}$. 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

Figure 21-11: $_{gm}$/I_{DS} vs. V_{gs} for V_{ds} = 0.1V, V_{bs} = 0, -2V; BSIM2 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
+ WVGHIGH= 0.000000E+00
+ VGLOW =-7.820000E-02  LVGLOW = 0.000000E+00
+ WVGLOW = 0.000000E+00
+ VFB =-5.760000E-01    LVFB = 0.000000E+00
+ WVFB = 0.000000E+00
+ PHI = 6.500000E-01    LPHI = 0.000000E+00
```

```
+ 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
+ MUS0 = 7.050000E+02   LMUS0 = 0.000000E+00
+ WMUS0 = 0.000000E+00
+ MUSB = 0.000000E+00   LMUSB = 0.000000E+00
+ WMUSB = 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
```

```
+ U10 = 0.000000E+00    LU10 = 7.900000E-01
+ WU10 = 0.000000E+00
+ U1B = 0.000000E+00    LU1B = 0.000000E+00
+ WU1B = 0.000000E+00
+ U1D = 0.000000E+00    LU1D = 0.000000E+00
+ WU1D = 0.000000E+00
+ N0 = 8.370000E-01     LN0 = 0.000000E+00    WN0 = 0.000000E+00
+ NB = 6.660000E-01     LNB = 0.000000E+00    WNB = 0.000000E+00
+ ND = 0.000000E+00     LND = 0.000000E+00    WND = 0.000000E+00
+ VOF0 = 4.770000E-01   LVOF0 = 0.000000E+00
+ WVOF0 = 0.000000E+00
+ VOFB =-3.400000E-02   LVOFB = 0.000000E+00
+ WVOFB = 0.000000E+00
+ VOFD =-6.900000E-02   LVOFD = 0.000000E+00
+ WVOFD = 0.000000E+00
+ AI0 = 1.840000E+00    LAI0 = 0.000000E+00
+ WAI0 = 0.000000E+00
+ AIB = 0.000000E+00    LAIB = 0.000000E+00
+ WAIB = 0.000000E+00
+ BI0 = 2.000000E+01    LBI0 = 0.000000E+00
+ WBI0 = 0.000000E+00
+ BIB = 0.000000E+00    LBIB = 0.000000E+00
+ WBIB = 0.000000E+00
+ DELL = 0.000000E+00   WDF = 0.000000E+00
```

Common SPICE Parameters

```
+ CGDO = 1.000000E-09   CGSO = 1.000000E-09
+ CGBO = 2.500000E-11
+ RSH = 3.640000E+01    JS = 1.380000E-06
+ PB = 8.000000E-01     PBSW = 8.000000E-01
+ CJ = 4.310000E-04     CJSW = 3.960000E-10
+ MJ = 4.560000E-01     MJSW = 3.020000E-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
+ CJGATE = 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

Name	Units	Default	Description
UO	cm^2/V/s	1.0	Mobility
VTO	V	0.0	Zero voltage threshold voltage
PHI	V	0.0	Surface potential
NFS	cm^2	0.0	Fast surface state density
NSS	cm^2	0.0	Surface state density
T1	m	280n	First thin film thickness
T2	m	0.0	Second thin film thickness
E1		3.9	Dielectric constant of 1st film
E2		0.0	Dielectric constant of 2nd film
THETA	V^{-1}	0.0	Mobility modulation
ETA	V^{-1}	0.0	Static feedback on threshold voltage (difficulty of band bending)
VMAX	m/s	1e6	Maximum drift velocity of carriers
GO	ohm^{-1}	10e-15	Conductance of TFT leakage current
DEFF		2.0	Drain voltage effect for TFT leakage current
NU		0.0	First order temperature gradient
CHI		0.5	Temperature exponential part
PSI		1e-20	Temperature exponential part

Name	Units	Default	Description
K2		2.0	Temperature exponential part
VTIME	s	10m	Voltage stress
TREF		1.5	Temperature gradient of UO
RD	ohm	1.0K	(External) drain resistance
RS	ohm	1.0K	(External) source resistance
CGSO	F	1.0p	TFT gate-to-source overlap capacitance
CGDO	F	1.0p	TFT gate-to-drain overlap capacitance
CSC	F/m^2	10μ	Space charge capacitance
FREQ	Hz	400	Frequency of device
FEFF		0.5	Frequency effect constant
TAU	s	10n	Relaxation time constant

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 ^{o}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 ^{o}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=1μ 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 vgs and vds are inputs. Ids, gm, and gds are the DC outputs, and the gate-to-source capacitance Cgs and the gate-to-drain capacitance Cgd are the AC outputs. Electron charge is q, Boltzmann's constant is k, and the permittivity of a vacuum is $\varepsilon0$.

Scaling by SCALE has been done prior to evaluation of the equations. Scaling by M is done after evaluation.

The variables gm_{tft} and gds_{tft} 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, $Cgdi = 0$, $Cgsi = 0$, $phi = PHI$, $vto = VTO$, $uo = UO$

If $uo = 0$ then $uo = 1$

Cfm, the dielectric capacitance per unit area, is computed as follows:

If $T1 \neq 0$ and $T2 \neq 0$, then $Cfm = \dfrac{(\varepsilon0 \cdot E1 \cdot E2)}{((T2 \cdot E1) + (T1 \cdot E2))}$

If $T1 = 0$ and $T2 \neq 0$, then $Cfm = \dfrac{(\varepsilon0 \cdot E2)}{T2}$

If $T2 = 0$ and $T1 \neq 0$, then $Cfm = \dfrac{(\varepsilon0 \cdot E1)}{T1}$

$kp = uo \cdot Cfm \cdot 10^{-4}$

TEMP is the Star-Hspice device simulation temperature, specified in $^{\circ}$C, but converted to $^{\circ}$K internally for the evaluation of these equations.

$vt = \dfrac{(k \cdot TEMP)}{q}$

$eg = (2 \cdot 10^{4} \cdot (TEMP - 312)) + 1.4$

$$vto = vto + (DELVTOmodel \cdot type) + (DELVTOelement \cdot type)$$

$$vbi = vto$$

$$ratio = \frac{TEMP}{312}$$

If $VTIME \le 1$, 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.*

$$vfb = vto - (0.5 \cdot PHI) + (0.5 \cdot (1.4 - eg))$$

$$vbi = vfb + (0.5 + PHI \cdot ratio)$$

$$vto = vbi \text{ (printback definition)}$$

$$phi = phi \cdot ratio \text{ (printback definition)}$$

$$vfb = vbi - phi \text{ (printback definition)}$$

$$vdsat = 0$$

$$beta = kp \cdot W \cdot L$$

$$vth = vbi + (ETA \cdot vds)$$

If $NU \ne 0$ and $K2 \ne 0$ and $PSI \ne 0$ and $VTIME > 1$, then:

$$vth = vth + f(vgs, vds, NU, K2, PSI, CHI, VTIME, TEMP)$$

$$von = vth$$

If $NFS \ne 0$, then:

$$xn = 1 + \left(\frac{(q \cdot NFS \cdot 10^4 \cdot W \cdot L)}{Cfm} \right)$$

$$von = f(vth, (vt \cdot xn))$$

Cutoff Region (NFS = 0, vgs $\leq$ von)

If $NFS = 0$ and $vgs \leq von$, then:

$$Cgdi = 0$$

$$Cgsi = 0$$

$$Ids = GO \cdot f(vgs, (DEFF \cdot vds))$$

$$gm = GO$$

$$gds = GO \cdot DEFF$$

Noncutoff Region (NFS $\neq$ 0)

If $vgs > von$, then:

$$vgsx = vgs$$

If $vgs \leq von$, then:

$$vgsx = von$$

Mobility modulation by vgs:

$$ueff = f(uo, \eta, vgs, THETA)$$

If $VMAX > 0$, then:

$$vdsc = \frac{L \cdot VMAX}{ueff}$$

$$vdsat = (vgsx - vth) + vdsc - \sqrt{((vgsx - vth)^2 + vdsc^2)}$$

$$Cfmlw = \frac{(Cfm \cdot CSC)}{(Cfm + CSC)} \cdot L \cdot W$$

$Cfmlw$ is the series combination of the dielectric and space charge capacitance of the MIS structure.

If vds < vdsat, then:

$$vdsx = vds$$

$$epsfm = Cfm \cdot \frac{(T2 + T1)}{\varepsilon 0}$$

$epsfm$ is the effective equivalent dielectric constant of the insulator layers.

$$fval = 0.8 + \left(\frac{epsfm - 0.8}{1 + (2 \cdot \pi \cdot FREQ \cdot TAU)^2} \right)$$

$$Cgdi = f(Cfmlw \cdot f(efm, 0.8) \cdot (\exp(fval, FEFF, vgs - vth - vds)))$$

$$Cgsi = f(Cfmlw \cdot f(efm, 0.8) \cdot (\exp(fval, FEFF, (vgs - vth), vds)))$$

Otherwise, vds Š ≥ vdsat :

$$vdsx = vdsat$$

$$Cgdi = Cfmlw$$

$$Cgsi = \frac{Cfmlw}{2}$$

If $vdsx \neq 0$, then:

$$cdnorm = vdsx \cdot \left(vgsx - vth - \frac{vdsx}{2} \right)$$

Normalized drain current:

$$gm_{tft} = vdsx$$

$$gds_{tft} = vgsx - vth - vdsx$$

$$cd1 = beta \cdot cdnorm$$

Drain current without velocity saturation effect:

$$beta = beta \cdot fgate$$

$$idrain = beta \cdot cdnorm$$

$$gm_{tft} = (beta \cdot gm_{tft}) + (dfgdvg \cdot cd1)$$

Velocity saturation factor—if $VMAX \neq 0$, then:

$$fdrain = \frac{1}{\left(1 + \left(\frac{vdsx}{vdsc}\right)\right)}$$

$$dfddvg = -dfgdvg \cdot \frac{((fdrain^2) \cdot vdsx)}{(vdsc \cdot fgate)}$$

$$dfddvd = \frac{-(fdrain^2)}{vdsc}$$

Strong inversion current:

$$gm_{tft} = (fdrain \cdot gm_{tft}) + (dfddvg \cdot idrain)$$

$$gds_{tft} = (fdrain \cdot gds_{tft}) + (dfddvd \cdot idrain)$$

$$idrain = fdrain \cdot idrain$$

$$beta = beta \cdot fdrain$$

$$Ids = idrain \cdot f(GO, vgs, DEFF, vds)$$

$$gm = f(gm_{tft}, GO)$$

$$gds = f(gds_{tft}, GO, DEFF)$$

Weak inversion current—if $vgs < von$, then:

$$idrain = idrain \cdot \exp\left(\frac{(vgs - von)}{(vt \cdot xn)}\right)$$

$$Ids = idrain + f(GO, vgs, DEFF, vds)$$

$$gm_{tft} = \frac{idrain}{(vt \cdot xn)}$$

$$gm = f(gm_{tft}, GO)$$

$$gds_{tft} = gds_{tft} \cdot \exp\left(\frac{(vgs - von)}{(vt \cdot xn)}\right)$$

$$gds = gds_{tft} + f(GO, DEFF)$$

vdsx = 0:

$$Ids = f(GO \cdot vgs, DEFF, vds)$$

$$gm = GO$$

$$gds_{tft} = beta \cdot (vgsx - vth)$$

If $NFS \neq 0$ and $vgs < von$, then

$$gds_{tft} = gds_{tft} \cdot \exp\left(\frac{(vgs - von)}{(vt \cdot xn)}\right)$$

$$gds = f(gds_{tft}, GO, DEFF)$$

Cgd, Cgs

$$Cgd = Cgdi + CGDO$$

$$Cgs = Cgsi + CGSO$$

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:

$$Leff \ = \ L + XL - 2 \cdot LD$$

The Berkeley BSIM1 and BSIM2 models use Leff = 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

LEVEL	2	3	13	28	39
Number of parameters	13	12	60	63	99
Minimal number of parameters	12	11	17	18	28
Physical parameters	50%	55%	12%	11%	7%
Continuous derivatives	no	no	yes	yes	yes
Positive GDS	yes	yes	no	yes	no
Monotonic GM/IDS	no	no	no	yes	yes

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

Figure 21-18: LEVEL 39 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

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 = vs. Vds at Vgs=2, 3, 4, 5, Vbs=0

■ These models still have a small change in slope of Gds at Vdsat, 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

Figure 21-29: LEVEL 28 Ids vs. Vgs Curves

Figure 21-30: LEVEL 39 Ids vs. Vgs Curves

LEVEL 2, 3, 28 - Gm/Ids Model vs. Data

■ gm/Ids -vs.- Vgs at Vds=0.1, Vbs =0, -2

■ The LEVEL 2 and 3 models have spikes at Vgs=Vth. The data, and the LEVEL 28 model, is monotonic decreasing.

Figure 21-31: LEVEL 2 gm/Ids vs. Vgs 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

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

Figure 21-38: LEVELs 13, 28, 39 gds vs. Vds Curves

Gm/Ids vs. Vgs at Vds=0.1, Vbs=0, 2

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

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

Figure 21-42: LEVELs 13, 28, 39 gm/Ids vs. Vgs Curves

References

1. Vladimirescu, Andrei and Liu, Sally. "Simulation of MOS Integrated Circuits Using SPICE2." University of California at Berkeley: Memorandum No. UCB/ERL M80/7, February 1980.

2. Huang, J.S., and Taylor, G.W. "Modeling of an Ion-Implanted Silicon Gate Depletion-Mode IGFET." *IEEE Trans. Elec. Dev.,* Vol. ED-22, pp. 995-1000, Nov. 1975.

3. Frohman-Bentchkowski, D. and Grove, A. S. "On the Effect of Mobility Variation on MOS Device Characteristics," *Proc. IEEE,* 56, 1968.

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.

5. Marciniak, W. et. al., "Comments on the Huang and Taylor Model of Ion-Implanted Silicon-gate Depletion-Mode IGFET,*" Solid State Electron.,* Vol. 28, No.3, pp. 313-315, 1985.

6. Ballay, N. et. al., "Analytic Modeling of Depletion-Mode MOSFET with Short- and Narrow-Channel Effects,*" IEEE PROC,* Vol. 128, Pt.I, No.6 (1981).

7. Tsividis, Y. Operations and Modeling of the MOS Transistor, McGraw-Hill, New York, 1987 p. 145; p. 241f. BFRC's counterpart in BSIM is x2u0.

8. Jeng, M. C. *Design and Modeling of Deep Submicrometer MOSFETs*, Ph.D. Dissertation, University of California, Berkeley, 1989.

9. Duster, J.S., Jeng,M.C., Ko, P. K. and Hu, C. *User's Guide for the BSIM2 Parameter Extraction Program and the SPICE3 with BSIM Implementation*. Industrial Liaison Program, Software Distribution Office, University of California, Berkeley, May 1990.

10. Duster, J.S., Jeng, M.C., Ko, P. K., and Hu, C. *User's Guide for the BSIM2 Parameter Extraction Program and the SPICE3 with BSIM Implementation*. Industrial Liaison Program, Software Distribution Office, University of California, Berkeley, May 1990.

Avant!

Chapter 22

Selecting MOSFET Models: LEVEL 47-62

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 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

Name	Unit	Default	Description
VTH0	V	0.7	Threshold voltage of long channel at $V_{bs} = 0$ and small V_{ds} (0.7 for n-channel, - 0.7 for p-channel)
K1	$V^{1/2}$	0.53	First-order body effect coefficient
K2		-0.0186	Second-order body effect coefficient
K3		80.0	Narrow width effect coefficient
K3B	1/V	0	Body width coefficient of narrow width effect
KT1	V	-0.11	Temperature coefficient for threshold voltage
KT2		0.022	Body bias coefficient of threshold temperature effect
GAMMA1	$V^{1/2}$	See "LEVEL 47 Model Equations"	Body effect coefficient, near interface
GAMMA2	$V^{1/2}$	See "LEVEL 47 Model Equations"	Body effect coefficient in the bulk
W0	m	2.5e-6	Narrow width effect coefficient
NLX	m	1.74e-7	Lateral nonuniform doping along channel
TOX	m	150e-10	Gate oxide thickness

Name	Unit	Default	Description
XJ	m	0.15e-6	Junction depth
DL	m	0.0	Channel length reduction on one side (multiplied by SCALM)
DW	m	0.0	Channel width reduction on one side (multiplied by SCALM)
NPEAK	cm^{-3} (see Note 8)	1.7e17	Peak doping concentration near interface
NSUB	cm^{-3}	6.0e16	Substrate doping concentration
PHI	V	See "LEVEL 47 Model Equations"	Surface potential under strong inversion
XT	m	1.55e-7	Doping depth
VBM	V	-5.0	Maximum substrate bias
VBX	V	See "LEVEL 47 Model Equations"	V_{bs} at which the depletion width equals XT
DVT0		2.2	Short-channel effect coefficient 0
DVT1		0.53	Short-channel effect coefficient 1
DVT2	1/V	-0.032	Short-channel effect coefficient 2
U0	m^2/ Vsec (see Note 8)	0.067	Low field mobility at T = *TREF* (0.067 for n-channel, 0.025 for p-channel)
UA	m/V	2.25e-9	First-order mobility degradation coefficient
UA1	m/V	4.31e-9	Temperature coefficient of *UA*

Name	Unit	Default	Description
UB	m^2/V^2	5.87e-19	Second-order mobility degradation coefficient
UB1	m^2/V^2	-7.61e-18	Temperature coefficient of *UB*
UC	1/V	0.0465	Body bias sensitivity coefficient of mobility
UC1	1/V	-0.056	Temperature coefficient of *UC*
VSAT	cm/sec	8e6	Saturation velocity of carrier at T = *TREF*
AT	m/sec	3.3e4	Temperature coefficient of *VSAT*
RDSW	ohm · μm	0.0	Source drain resistance per unit width
RDS0	ohm	0.0	Source drain contact resistance
LDD	m	0.0	Total length of LDD region
ETA		0.3	Coefficient of drain voltage reduction
ETA0		0.08	Subthreshold region DIBL (Drain Induced Barrier Lowering) coefficient
ETAB	1/V	-0.07	Subthreshold region DIBL coefficient
EM	V/m	4.1e7	Electrical field in channel above which hot carrier effect dominates
NFACTOR		1.0	Subthreshold region swing
VOFF	V	-0.11	Offset voltage in subthreshold region
LITL	m		Characteristic length. The default is $$LITL = \left(\frac{\varepsilon_{si} T_{ox} X_j}{\varepsilon_{ox}} \right)^{1/2}$$
VGLOW	V	-0.12	Lower bound of the weak-strong inversion transition region
VGHIGH	V	0.12	Upper bound of the weak-strong inversion transition region

Name	Unit	Default	Description
CDSC	F/m^2	2.4e-4	Drain/source and channel coupling capacitance
CDSCB	F/Vm^2	0	Body coefficient for CDSC
CIT	F/m^2	0.0	Interface state capacitance
PCLM		1.3	Coefficient of channel length modulation
PDIBL1		0.39	DIBL (Drain Induced Barrier Lowering) effect coefficient 1
PDIBL2		0.0086	DIBL effect coefficient 2
DROUT		0.56	DIBL effect coefficient 3
DSUB		DROUT	DIBL coefficient in subthreshold region
PSCBE1	V/m	4.24e8	Substrate current induced body effect exponent 1
PSCBE2	m/V	1.0e-5	Substrate current induced body effect coefficient 2
A0		1	Bulk charge effect. The default is 4.4 for PMOS.
TNOM (TREF)	°C	25	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.
SUBTHMOD		2	Subthreshold model selector
SATMOD		2	Saturation model selector
KETA	1/V	-0.047	Body bias coefficient of the bulk charge effect
A1	1/V	0	First nonsaturation factor (0 for NMOS, 0.23 for PMOS)

Name	Unit	Default	Description
A2		1.0	Second nonsaturation factor (1.0 for NMOS, 0.08 for PMOS)
UTE		-1.5	Mobility temperature exponent
KT1L	Vm	0	Channel length sensitivity of temperature coefficient for threshold voltage
UC0*	$(V/m)^2$		Temperature coefficient
BULKMOD		1	Bulk charge model selector
XPART		1	Charge partitioning flag
VFB	V		Flat-band voltage
PVAG		0	Gate dependence of output resistance
* UC0 has no effect on the model			

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 BSIM3 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 cm^{-3} as follows: if *NPEAK* is greater than 1e20, it is multiplied by 1e-6. *U0* is converted to $m^2/Vsec$ as follows: if *U0* is greater than 1, it is multiplied by 1e-4. You must enter the parameter *NSUB* in cm^{-3} units.

9. The specified value of *VTH0* for p-channel in the .MODEL statement should be negative.

10. The default value of *KT1* is -0.11. The negative sign ensures that the absolute value of threshold decreases with increasing temperature for NMOS and PMOS.

11. Model parameter *LITL* is not allowed to go below a minimum value of 1.0e-9 m, to avoid a possible divide by zero error.

12. *VSAT*, 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 *KT1* and *KT2* for *VTH*, *UTE* for *U0*, *AT* for *VSAT*, *UA1* for *UA*, *UB1* for *UB*, and *UC1* for *UC*.

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 *XL*, *LD*, *XW*, *WD*, *CJ*, *CJSW*, *JS*, and *JSW*.

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:

$$L_{\text{eff}} = L + XL - (2 \cdot LD)$$

$$W_{\text{eff}} = W + XW - (2 \cdot WD)$$

The UCB SPICE3 equations used for BSIM3 are:

$$L_{\text{eff}} = L - (2 \cdot DL)$$

$$W_{\text{eff}} = W - (2 \cdot DW)$$

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,

$$Leff = L+XL-2 \cdot LDAC, \quad Weff = W+XW-2 \cdot WDAC$$

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:
.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:

.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}, $K1$, $K2$, ϕ_s, N_{lx}, $K3$, W_0, T_{ox}, V_{bi}, D_{vt0}, D_{vt1}, D_{vt2}, N_{peak}, N_{sub}, Υ_1, Υ_2, V_{bx}, V_{bm}, V_{bi}, X_t, $TREF$

$$V_{th} = V_{th0} + K1(\sqrt{\phi_s - V_{bs}} - \sqrt{\phi_s}) - K2V_{bs} + K1\left(\sqrt{1 + \frac{N_{lx}}{L_{eff}}\sqrt{\frac{\phi_s}{\phi_s - V_{bs}}}} - 1\right)\sqrt{\phi_s}$$

$$+ \quad (K3 + K3B \cdot V_{bs}) \cdot \left(\frac{T_{ox}}{W_{eff} + W_0}\right)\phi_s - \Delta V_{th}$$

$$T_{ratio} = \frac{(TEMP + DTEMP + 273.15)}{(TREF + 273.15)}$$

$$\Delta V_{th} = \theta_{th}(L_{eff}) \cdot (V_{bi} - \phi_s)$$

$$\theta_{th}(L_{eff}) = D_{vt0} \cdot \left[exp\left(\frac{-D_{vt1} \cdot L_{eff}}{2l_t}\right) + 2\,exp\left(\frac{-D_{vt1} \cdot L_{eff}}{l_t}\right)\right]$$

$$l_t = \sqrt{3 \cdot T_{ox} \cdot X_{dep}} \cdot (1 + D_{vt2} \cdot V_{bs})$$

$$X_{dep} = \sqrt{\frac{2 \cdot \varepsilon_{si} \cdot (\phi_s - V_{bs})}{q \cdot N_{peak}}}$$

If Phi_s is not specified as a model parameter, then

$$\phi_s = 2 \cdot V_{tm} \cdot ln\left(\frac{N_{peak}}{n_i}\right) \qquad (N_{peak}\ \text{and}\ n_i\ \text{in cm}^{-3})$$

$$V_{tm} = K \cdot T / q$$

$$n_i = 1.45e10 \cdot \left(\frac{T}{300.15}\right)^{1.5} \cdot exp(21.5565981 - Eg/(2 \cdot V_{tm}))$$

$$Eg = 1.16 - (7.02e-4) \cdot T^2/(T + 1108.0)$$

If *K1*, *K2* are not specified as model parameters, then they are calculated as follows:

$$K_1 = \Upsilon_2 - 2 \cdot K_2 \cdot \sqrt{\phi_s - V_{bm}}$$

$$K_2 = (\Upsilon_1 - \Upsilon_2) \cdot \frac{\sqrt{\phi_s - V_{bx}} - \sqrt{\phi_s}}{2 \cdot \sqrt{\phi_s} \cdot (\sqrt{\phi_s - V_{bm}} - \sqrt{\phi_s}) + V_{bm}}$$

$$\Upsilon_1 = \frac{\sqrt{2 \cdot q \cdot \varepsilon_{si} \cdot N_{peak}}}{C_{ox}}$$

$$\Upsilon_2 = \frac{\sqrt{2 \cdot q \cdot \varepsilon_{si} \cdot N_{sub}}}{C_{ox}}$$

$$V_{bx} = \phi_s - \left(\frac{q \cdot N_{peak} \cdot X_t^2}{2 \cdot \varepsilon_{si}}\right)$$

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

$\mu_0, \ U_a, \ U_b, \ U_c$

$$\mu_{\text{eff}} = \frac{\mu_0}{1 + U_a \cdot \left(\frac{V_{gs} + V_{\text{th}}}{T_{\text{ox}}}\right) + U_b \cdot \left(\frac{V_{gs} + V_{\text{th}}}{T_{\text{ox}}}\right)^2 + U_c \cdot V_{bs}}$$

Drain Saturation Voltage

Model Parameters

$A_0, \ v_{\text{sat}}, \ X_j, \ A_1, \ A_2, \ R_{ds0}, \ R_{dsw}$

Rds and Pfactor:

$$R_{ds} = R_{ds0} + R_{dsw} / (1e6 \cdot W_{\text{eff}})$$

$$Pfactor = A_1 \cdot V_{gst} + A_2 \qquad \text{(if } Pfactor > 1 \text{, it is set to } Pfactor = 1)$$

$$V_{gst} = V_{gs} - V_{\text{th}}$$

Vdsat for the case Rds = 0 and Pfactor = 1:

$$V_{\text{dsat}} = \frac{E_{\text{sat}} \cdot L_{\text{eff}} \cdot V_{gst}}{A_{\text{bulk}} \cdot E_{\text{sat}} \cdot L_{\text{eff}} + V_{gst}}$$

For BULKMOD = 1,

$$A_{\text{bulk}} = \left(1 + \frac{K1 \cdot A_0 \cdot L_{\text{eff}}}{(L_{\text{eff}} + T1) \cdot T1s \cdot 2}\right) / (1 + KETA \cdot V_{bs})$$

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 $V_{\text{bs}} \leq 0$,

$$T1s = \sqrt{\phi_s - V_{bs}}$$

For $V_{\text{bs}} \geq 0$,

$$T1s = \frac{\phi_s \cdot \sqrt{\phi_s}}{\phi_s + \dfrac{V_{bs}}{2}}$$

$$E_{\text{sat}} = 2 \cdot \frac{v_{\text{sat}}}{\mu_{\text{eff}}}$$

Vdsat for the general case:

V_{dsat} is the solution of *Tmpa* * V_{dsat} * V_{dsat} - *Tmpb* * V_{dsat} + *Tmpc* = 0

$$V_{\text{dsat}} = \left(Tmpb - \sqrt{Tmpb^2 - 4 \cdot Tmpa \cdot Tmpc} \right) / (2 \cdot Tmpa)$$

$$Tmpa = A_{\text{bulk}} \cdot (A_{\text{bulk}} \cdot W_{\text{eff}} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot R_{ds} - 1 + 1/Pfactor)$$

$$Tmpb = V_{gst} \cdot (2/Pfactor - 1) + (A_{\text{bulk}} \cdot E_{\text{sat}} \cdot L_{\text{eff}}) + (3 \cdot A_{\text{bulk}} \cdot V_{gst} \cdot W_{\text{eff}} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot R_{ds})$$

$$Tmpc = (V_{gst} \cdot E_{\text{sat}} \cdot L_{\text{eff}}) + (V_{gst}^2 \cdot 2 \cdot W_{\text{eff}} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot R_{ds})$$

Linear Region

$$I_{\text{dslin0}} = \mu_{\text{eff}} \cdot C_{\text{ox}} \cdot \frac{W_{\text{eff}}}{L_{\text{eff}}} \cdot \frac{1}{1 + V_{ds}/(E_{\text{sat}} \cdot L)} \cdot \left(V_{gs} - V_{\text{th}} - A_{\text{bulk}} \cdot \frac{V_{ds}}{2} \right) \cdot V_{ds}$$

$$I_{ds} = \frac{I_{\text{dslin0}}}{1 + \dfrac{R_{ds} \cdot I_{\text{dslin0}}}{V_{ds}}}$$

Saturation Region

Model Parameters

$litl,\ eta,\ L_{dd},\ E_m,\ D_{\text{rout}},\ P_{\text{clm}},\ P_{\text{dibl1}},\ P_{\text{dibl2}},\ P_{\text{scbe1}},\ P_{\text{scbe2}}$

Vasat and Fvag:

$$V_{\text{asat}} = \frac{E_{\text{sat}} \cdot L_{\text{eff}} + V_{\text{dsat}} + 2R_{ds} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot W_{\text{eff}} \cdot \left(V_{gst} - \dfrac{A_{\text{bulk}} \cdot V_{\text{dsat}}}{2} \right)}{2/Pfactor - 1 + R_{ds} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot W_{\text{eff}} \cdot A_{\text{bulk}}}$$

$$F_{vag} = 1 + \frac{P_{vag} \cdot V_{gst}}{E_{\text{sat}} \cdot L_{\text{eff}}}$$

Early Voltage, satMod = 1:

$$V_A = V_{\text{asat}} + F_{vag} \cdot \left(\frac{1 + eta \cdot \dfrac{L_{dd}}{litl}}{P_{\text{clm}} \cdot A_{\text{bulk}}} \right) \cdot \left(\frac{(A_{\text{bulk}} \cdot E_{\text{sat}} \cdot L_{\text{eff}} + V_{gst} - \lambda \cdot (V_{ds} - V_{\text{dsat}})) \cdot (V_{ds} - V_{\text{dsat}})}{E_{\text{sat}} \cdot litl} \right)$$

$$\lambda = \frac{A_{\text{bulk}} \cdot E_{\text{sat}} \cdot L_{\text{eff}} + (V_{gst})}{2 \cdot litl \cdot E_m}$$

Early Voltage, satMod = 2:

$$V_A = V_{\text{asat}} + F_{vag} \cdot U_{vds} \cdot \left(\frac{1}{V_{\text{aclm}}} + \frac{1}{V_{\text{adibl}}} \right)^{-1}$$

$$U_{vds} = 1 + eta \cdot \frac{L_{dd}}{litl}$$

$$V_{\text{aclm}} = \frac{1}{P_{\text{clm}}} \cdot \frac{A_{\text{bulk}} \cdot E_{\text{sat}} \cdot L_{\text{eff}} + V_{gst}}{A_{\text{bulk}} \cdot E_{\text{sat}} \cdot litl} \cdot (V_{ds} - V_{\text{dsat}})$$

$$V_{\text{adibl}} = \frac{1}{\theta_{\text{rout}}} \cdot \left[(V_{gs} - V_{\text{th}}) - \left(\frac{1}{A_{\text{bulk}} \cdot V_{\text{dsat}}} + \frac{1}{V_{gst}} \right)^{-1} \right]$$

$$\theta_{\text{rout}} = P_{\text{dibl1}} \cdot \left[exp \left(\frac{-D_{\text{rout}} \cdot L_{\text{eff}}}{2 \cdot l_t} \right) + 2 \, exp \left(\frac{-D_{\text{rout}} \cdot L_{\text{eff}}}{l_t} \right) \right] + P_{\text{dibl2}}$$

$$V_{\text{ahce}} = \left[\frac{P_{\text{scbe2}}}{L_{\text{eff}}} \cdot exp \left(\frac{-P_{\text{scbe1}} \cdot litl}{V_{ds} - V_{\text{dsat}}} \right) \right]^{-1}$$

Drain Current

$$I_{\text{dsat}} = W_{\text{eff}} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot (V_{gs} - V_{\text{th}} - A_{\text{bulk}} \cdot V_{\text{dsat}}) \cdot Pfactor$$

$$Pfactor = A_1 \cdot V_{gst} + A_2$$

$$I_{ds} = I_{\text{dsat}} \cdot \left(1 + \frac{V_{ds} - V_{\text{dsat}}}{V_A} \right) \cdot \left(1 + \frac{V_{ds} - V_{\text{dsat}}}{V_{\text{ahce}}} \right)$$

Subthreshold Region

Model Parameters

$Nfactor, \ C_{\text{dsc}}, \ C_{\text{dscb}}, V_{\text{off}}, \ C_{\text{it}}, D_{\text{sub}}, eta_0, \ eta_b$

n and DIBL:

$$n = 1 + \frac{Nfactor \cdot 1.034e{-}10}{X_{\text{dep}} \cdot C_{\text{ox}}} + \frac{(C_{\text{dsc}} + C_{\text{dscb}} \cdot V_{bs}) \cdot \left[exp\left(\frac{-L_{\text{eff}}}{2 \cdot l_t}\right) + 2\,exp\left(\frac{-L_{\text{eff}}}{l_t}\right) \right] + C_{\text{it}}}{C_{\text{ox}}}$$

$$DIBL = (eta_0 + eta_b \cdot V_{bs}) \cdot \left[exp\left(\frac{-D_{\text{sub}} \cdot L_{\text{eff}}}{2 \cdot l_{t0}}\right) + 2\,exp\left(\frac{-D_{\text{sub}} \cdot L_{\text{eff}}}{l_{t0}}\right) \right]$$

$$l_{t0} = \sqrt{3 \cdot T_{\text{ox}} \cdot X_{\text{dep0}}}$$

$$X_{\text{dep0}} = \sqrt{\frac{2 \cdot \varepsilon_{\text{si}} \cdot \phi_s}{q \cdot N_{\text{peak}}}}$$

If subthMod = 0,

$$I_{ds} = g_m = g_{ds} = g_{mb} = 0$$

If subthMod = 1,

$$I_{ds} = \frac{I_{\text{limit}} \cdot I_{\text{exp}}}{I_{\text{limit}} + I_{\text{exp}}} \cdot \left[1 - exp\left(\frac{-V_{ds}}{V_{tm}}\right) \right]$$

$$I_{\text{limit}} = \frac{9}{2} \cdot u0 \cdot \sqrt{\frac{q\varepsilon_{\text{si}} \cdot N_{\text{peak}}}{2 \cdot \phi_s}} \cdot \frac{W_{\text{eff}}}{L_{\text{eff}}} \cdot V_{tm}^2$$

$$I_{\text{exp}} = u0 \cdot \sqrt{\frac{q \cdot \varepsilon_{\text{si}} \cdot N_{\text{peak}}}{2 \cdot \phi_s}} \cdot \frac{W_{\text{eff}}}{L_{\text{eff}}} \cdot V_{tm}^2 \cdot exp\left(\frac{V_{gs} - V_{\text{th}} - V_{\text{off}} + DIBL \cdot V_{ds}}{n \cdot V_{tm}}\right)$$

If subthMod = 2,

$$I_{ds} = u0 \cdot \sqrt{\frac{q \cdot \varepsilon_{\text{si}} \cdot N_{\text{peak}}}{2 \cdot \phi_s}} \cdot \frac{W_{\text{eff}}}{L_{\text{eff}}} \cdot V_{tm}^2 \cdot \left[1 - exp\left(\frac{-V_{ds}}{V_{tm}}\right) \right] \cdot exp\left(\frac{V_{gs} - V_{\text{th}} - V_{\text{off}} + DIBL \cdot V_{ds}}{n \cdot V_{tm}}\right)$$

Transition Region (for subthMod = 2 only)

Model Parameters

$V_{\text{gshigh}}, \ V_{\text{gslow}}$

$$I_{ds} = (1-t)^2 \cdot I_{\text{dslow}} + 2 \cdot (1-t) \cdot t \cdot I_p + t^2 \cdot I_{\text{dshigh}}$$

$$t = \left(\frac{V_p - V_{\text{gslow}}}{V_{\text{gslow}} - 2 \cdot V_p + V_{\text{gshigh}}} \right) \cdot \left(\sqrt{1 + \frac{(V_{\text{gslow}} - 2 \cdot V_p + V_{\text{gshigh}})(V_{gs} - V_{\text{th}} - V_{\text{gslow}})}{(V_p - V_{\text{gslow}})^2}} - 1 \right)$$

$$V_p = \frac{(g_{\text{mhigh}} \cdot V_{\text{gshigh}} - g_{\text{mlow}} \cdot V_{\text{gslow}}) - (I_{\text{dshigh}} - I_{\text{dslow}})}{g_{\text{mhigh}} - g_{\text{mlow}}}$$

$$I_p = I_{\text{dslow}} + g_{\text{mlow}} \cdot (V_p - V_{\text{gslow}})$$

Temperature Compensation

Model Parameters

$A_t, \ U_{a1}, \ U_{b1}, \ U_{c1}, \ KT1, \ KT2, UTE$

$$V_{\text{th}}(temp) = V_{\text{th}}(tref) + (KT1 + KT2 \cdot V_{bs}) \cdot (T_{\text{ratio}} - 1)$$

$$u0(temp) = u0(tref) \cdot (T_{\text{ratio}})^{UTE}$$

$$V_{\text{sat}}(temp) = V_{\text{sat}}(tref) - A_t \cdot (T_{\text{ratio}} - 1)$$

$$U_a(temp) = U_a(tref) + U_{a1} \cdot (T_{\text{ratio}} - 1)$$

$$U_b(temp) = U_b(tref) + U_{b1} \cdot (T_{\text{ratio}} - 1)$$

$$U_c(temp) = U_c(tref) + U_{c1} \cdot (T_{\text{ratio}} - 1)$$

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 Versions3.0, 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:

Berkeley Release	VERSION	HSPVER
Version 3.0 (October 1995)	3.0	98.0
Version 3.0 with June 1998 bug fixes	3.0	98.2
Version 3.1 (December 1996)	3.1	98.0

Berkeley Release	VERSION	HSPVER
Version 3.1 with June 1998 bug fixes	3.1	98.2
Version 3.2 (June 16,1998)	3.2	98.2
Version 3.2.1 (April 20, 1999)	3.21	99.2
Version 3.2.2 (April 20, 1999)	3.22	99.2

Version 3.2 Features

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

The Berkeley NonQuasi-Static (NQS) model is now available (as of the 98.2 release) in Star-Hspice for LEVELs 49 and 53. This model provides a first-order correction to the quasi-static charge models. See *M.Chan, K.- Y. Hui, C. Hu, and P.-K. Ko, IEEE Trans. Electron Devices, vol. ED-45, pp.834-841, 1998*. The Star-Hspice 98.2 release supports only the model parameter implementation.

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:

```
   (Bulk-source capacitance)
if (Ps > Weff)
   Cbs = AS * Cjbs + (PS - Weff) * Cjbssw + Weff *
   Cjbsswg
else
   Cbs = AS * Cjbs + PS * Cjbsswg
```

Area and perimeter factors AS, PS default to 0 if not specified on the element line.

```
if (Vbs < 0)
   Cjbs = Cj * (1 - (Vbs/Pb))^-Mj
   Cjbssw = Cjsw * (1 - (Vbs/Pbsw))^-Mjsw
   Cjbsswg = Cjswg * (1 - (Vbs/Pbswg))^-Mjswg
else
   Cjbs = Cj * (1 + Mj * (Vbs/Pb))
   Cjbssw = Cjsw * (1 + Mjsw * (Vbs/Pbsw))
   Cjbsswg = Cjswg * (1 + Mjswg * (Vbs/Pbswg))
```

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 * HDIFeff * Weff
```

else:

```
AD = AD * WMLT^2
```

if AS is not specified on the element line:

```
AS = 2 * HDIFeff * Weff
```

else:

```
AS = AS * WMLT^2
```

if PS is not specified on the element line:

```
PS = 4 * HDIFeff + 2 * Weff
```

else:

```
PS = PS * WMLT
```

if PD is not specified on the element line:
```
PD = 4 * HDIFeff + 2 * Weff
```

else:
```
PD = PD * 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.

```
Weff = W * WMLT + XW
```

where:

```
HDIFeff = HDIF * WMLT
```

W	is the width specified on the element line
HDIF	is a heavy diffusion length specified in the model card
WMLT	is a shrink factor specified in the model card
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:

- NJ (ignored, instead use N)
- CJSWG (ignored, instead use CJGATE)
- MJSWG (ignored, there is no equivalent HSPICE parameter, the gate sidewall grading coefficient will be set = MJSW)
- PBSW (ignored, instead use PHP)
- PBSWG (ignored, there is no equivalent HSPICE parameter, the gate sidewall contact potential will be set = 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,ALPHA0, CIT is only recommended but not required in the BSIM3-lite model card.

Parameters Excluded from BSIM3-Lite Model

Parameter	Comments
mobmod	Recommended default or set = 1
nqsmod	Recommended default or set = 0
toxm	default = tox
ll	default = 0
lln	default = 1
lw	default = 0
lwn	default = 1
lwl	default = 0
wl	default = 0
wln	default = 1
ww	default = 0

Parameter	Comments
wwn	default = 1
wwl	default = 0
dwg	default = 0
dwb	default = 0
llc	default = 0
lwc	default = 0
lwlc	default = 0
wlc	default = 0
wwc	default = 0
wwlc	default = 0
b0	default = 0
b1	default = 0
vbx	do not define
vbm	do not define
xt	do not define
nsub	do not define
nlx	default = 0, std default=1.74e-7
gamma1	do not define
gamma2	do not define
ngate	Recommended default or set = 0
k3	default = 0, std default=80
k3b	default = 0
w0	no effect
dvt0	default = 0, std default=2.2

Parameter	Comments
dvt1	default = 0, std default=0.53
dvt2	default = 0, std default=-0.032
dvt0w	default = 0
dvt1w	default = 0, std default=5.3e6
dvt2w	default = 0, std default=-0.032
dsub	default = 0
prwg	default = 0
prwb	default = 0
wr	Recommended default or set = 1
drout	default = 0, std default=0.56
pdiblc1	default = 0, std default=0.39
cit	Recommended default or set = 0
alpha0	Recommended default or set = 0 for Version 3.2
kt1l	default = 0

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 $1/\text{Leff}$ and $1/\text{Weff}$ by specifying four terms: the parameter Xo, a length term Xl, a width term Xw, and a product term Xp. The parameter value at a given L,W is then interpolated as:

```
X = Xo + Xl/Leff + Xw/Weff + Xp/Leff/Weff
```

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

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 = Xo + Xl*(1/Leff - 1/LREF) + Xw*(1/Weff - 1/WREF) + Xp/(1/Leff -
1/LREF)/(1/Weff - 1/WREF)
```

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.

VERSION	BSIM3v3	LEVEL 49	LEVEL 53
3.0	1	1	1
3.1	2	0	2
3.2	3	3	3

Hspice VFBFLAG

The capacitance model CAPMOD=0 normally calculates the threshold voltage as Vth = vfbc + phi + k1 * sqrt(phi - vbs), where vfbc 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 VFBFLAG=1. The capacitance model CAPMOD=0 will calculate the threshold voltage as Vth = vth0 + k1 * sqrt(phi - vbs) - k1 * sqrt(phi). The VFBFLAG 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$^{\mathrm{o}}$C in Star-Hspice, but is 27$^{\mathrm{o}}$C in SPICE3. If the BSIM3 model parameters are specified at 27$^{\mathrm{o}}$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

Name	Unit	Default	Bin	Description
VERSION	-	3.2	No	Selects from BSIM3 Versions 3.0, 3.1, 3.2. Warning is issued if not explicitly set.
HSPVER	-	98.2	No	Selects from Hspice Versions: 98.2, 97.4, 97.2, 96.4, 96.3, 96.1
PARAMCHK	-	0	No	PARAMCHK=1 will check model parameters for range compliance
APWARN	-	0	No	When > 0 turns off warning message for PS,PD < Weff (Hspice specific)
BINFLAG	-	0	No	Uses wref, lref when set > 0.9 (Hspice specific)
MOBMOD	-	1	No	Mobility model selector
CAPMOD	-	3	No	Selects from charge models 0,1,2,3 LEVEL 49 CAPMOD defaults to 0.
CAPOP	-	-	No	Obsolete for LEVELs 49, 53. Ignored by Hspice (Hspice specific) in all versions.
NOIMOD	-	1	No	Berkeley noise model flag

Name	Unit	Default	Bin	Description
NLEV	-	-(off)	No	Star-Hspice noise model flag (non-zero overrides NOIMOD) (Hspice specific). See "Using Noise Models" on page 20-99 for more information.
NQSMOD	-	0 (off)	No	NQS Model flag
SFVTFLAG	-	0 (off)	No	Spline function for Vth (Hspice specific)
VFBFLAG	-	0 (off)	No	VFB selector for CAPMOD=0 (Hspice specific)

Basic Model Parameters

Name	Unit	Default	Bin	Description
VGSLIM	V	0	No	Asymptotic Vgs value, Min value is 5V. 0-value indicates an asymptote of infinity. (Hspice and LEVEL 49 specific)
TOX	m	150e-10	No	Gate oxide thickness
XJ	m	0.15e-6	Yes	Junction depth
NGATE	cm^{-3}	0	Yes	Poly gate doping concentration
VTH0 (VTHO)	V	0.7 NMOS -0.7 PMOS	Yes	Threshold voltage of long channel device at V_{bs} = 0 and small V_{ds}
NSUB	cm^{-3}	6.0e16	Yes	Substrate doping concentration
NCH	cm^{-3} See Note6	1.7e17	Yes	Peak doping concentration near interface
NLX	m	1.74e-7	Yes	Lateral nonuniform doping along channel

Name	Unit	Default	Bin	Description
K1	$V^{1/2}$	0.50	Yes	First-order body effect coefficient
K2	-	-0.0186	Yes	Second-order body effect coefficient
K3	-	80.0	Yes	Narrow width effect coefficient
K3B	1/V	0	Yes	Body width coefficient of narrow width effect
W0	m	2.5e-6	Yes	Narrow width effect coefficient
DVT0W	1/m	0	Yes	Narrow width coefficient 0, for Vth, at small L
DVT1W	1/m	5.3e6	Yes	Narrow width coefficient 1, for Vth, at small L
DVT2W	1/V	-0.032	Yes	Narrow width coefficient 2, for Vth, at small L
DVT0	-	2.2	Yes	Short channel effect coefficient 0, for V_{th}
DVT1	-	0.53	Yes	Short channel effect coefficient 1, for V_{th}
DVT2	1/V	-0.032	Yes	Short channel effect coefficient 2, for V_{th}
ETA0	-	0.08	Yes	Subthreshold region DIBL (drain induced barrier lowering) coefficient
ETAB	1/V	-0.07	Yes	Subthreshold region DIBL coefficient
DSUB	-	DROUT	Yes	DIBL coefficient exponent in subthreshold region
VBM	V	-3.0	Yes	Maximum substrate bias, for V_{th} calculation

Name	Unit	Default	Bin	Description
U0	cm^2/V/sec	670 nmos 250 pmos	Yes	Low field mobility at T = *TREF* = *TNOM*
UA	m/V	2.25e-9	Yes	First-order mobility degradation coefficient
UB	m^2/V^2	5.87e-19	Yes	Second-order mobility degradation coefficient
UC	1/V	-4.65e-11 or -0.0465	Yes	Body bias sensitivity coefficient of mobility -4.65e-11 for MOBMOD=1,2 or, -0.0465 for MOBMOD = 3
A0	-	1.0	Yes	Bulk charge effect coefficient for channel length
AGS	1/V	0.0	Yes	Gate bias coefficient of Abulk
B0	m	0.0	Yes	Bulk charge effect coefficient for channel width
B1	m	0.0	Yes	Bulk charge effect width offset
KETA	1/V	-0.047	Yes	Body-bias coefficient of bulk charge effect
VOFF	V	-0.08	Yes	Offset voltage in subthreshold region
VSAT	m/sec	8e4	Yes	Saturation velocity of carrier at T = *TREF* = *TNOM*
A1	1/V	0	Yes	First nonsaturation factor
A2	-	1.0	Yes	Second nonsaturation factor
RDSW	ohm · μm	0.0	Yes	Parasitic source drain resistance per unit width
PRWG	1/V	0	Yes	Gate bias effect coefficient of RDSW

Name	Unit	Default	Bin	Description
PRWB	$1/V^{1/2}$	0	Yes	Body effect coefficient of RDSW
WR	-	1.0	Yes	Width offset from Weff for Rds calculation
NFACTOR	-	1.0	Yes	Subthreshold region swing
CIT	F/m^2	0.0	Yes	Interface state capacitance
CDSC	F/m^2	2.4e-4	Yes	Drain/source and channel coupling capacitance
CDSCD	F/Vm^2	0	Yes	Drain bias sensitivity of CDSC
CDSCB	F/Vm^2	0	Yes	Body coefficient for CDSC
PCLM	-	1.3	Yes	Coefficient of channel length modulation values $\leq$ 0 will result in an error message and program exit.
PDIBLC1	-	0.39	Yes	DIBL (drain induced barrier lowering) effect coefficient 1
PDIBLC2	-	0.0086	Yes	DIBL effect coefficient 2
PDIBLCB	1/V	0	Yes	Body effect coefficient of DIBL effect coefficients
DROUT	-	0.56	Yes	Length dependence coefficient of the DIBL correction parameter in R_{out}
PSCBE1	V/m	4.24e8	Yes	substrate current induced body effect exponent 1
PSCBE2	V/m	1.0e-5	Yes	Substrate current induced body effect coefficient 2
PVAG	-	0	Yes	Gate dependence of Early voltage
DELTA	V	0.01	Yes	Effective Vds parameter
ALPHA0	m/V	0	Yes	First parameter of impact ionization current

Name	Unit	Default	Bin	Description
BETA0	V	30	Yes	Second parameter of impact ionization current
RSH	0.0	ohm/ square	No	Source/drain sheet resistance in ohm per square

AC and Capacitance Parameters

Name	Unit	Default	Bin	Description
XPART	-	0	No	Charge partitioning rate flag (default deviates from BSIM3V3=0) LEVEL 49 XPART defaults to 1
CGSO	F/m	p1 (see Note1)	No	Non-LDD region source-gate overlap capacitance per unit channel length
CGDO	F/m	p2 (see Note2)	No	Non-LDD region source-gate overlap capacitance per unit channel length
CGBO	F/m	0	No	Gate-bulk overlap capacitance per unit channel length
CGS1	F/m	0.0	Yes	Lightly doped source-gate overlap region capacitance
CGD1	F/m	0.0	Yes	Lightly doped drain-gate overlap region capacitance
CKAPPA	F/m	0.6	Yes	Coefficient for lightly doped region overlap capacitance fringing field capacitance
CF	F/m	(see Note3)	Yes	Fringing field capacitance
CLC	m	0.1e-6	Yes	Constant term for the short channel model

Name	Unit	Default	Bin	Description
CLE	-	0.6	Yes	Exponential term for the short channel model
VFBCV	V	-1.0	Yes	Flat band voltage used only in CAPMOD=0 C-V calculations

Length and Width Parameters

Name	Unit	Default	Bin	Description
WINT	m	0.0	No	Width offset fitting parameter from I-V without bias
WLN	-	1.0	No	Power of length dependence of width offset
WW	m^{WWN}	0.0	No	Coefficient of width dependence for width offset
WWN	-	1.0	No	Power of width dependence of width offset.
WWL	m^{WWN} $*m^{WLN}$	0.0	No	Coefficient of length and width cross term for width offset
DWG	m/V	0.0	Yes	Coefficient of Weff's gate dependence
DWB	$m/V^{1/2}$	0.0	Yes	Coefficient of Weff's substrate body bias dependence
LINT	m	0.0	No	Length offset fitting parameter from I-V without bias
LL	m^{LLN}	0.0	No	Coefficient of length dependence for length offset
LLN	-	1.0	No	Power of length dependence of length offset
LW	m^{LWN}	0.0	No	Coefficient of width dependence for length offset

Name	Unit	Default	Bin	Description
LWN	-	1.0	No	Power of width dependence of length offset
LWL	m^{LWN} $*m^{LLN}$	0.0	No	Coefficient of length and width cross term for length offset
DLC	m	LINT	No	Length offset fitting parameter from CV
DWC	m	WINT	No	Width offset fitting parameter from CV

Temperature Parameters

Name	Unit	Default	Bin	Description
KT1	V	0.0	Yes	Temperature coefficient for Vth
KT1L	m-V	0.0	Yes	Temperature coefficient for channel length dependence of Vth
KT2	-	0.022	Yes	Body bias coefficient of Vth temperature effect
UTE	-	-1.5	Yes	Mobility temperature exponent
UA1	m/V	4.31e-9	Yes	Temperature coefficient for UA
UB1	$(m/V)^2$	-7.61e-18	Yes	Temperature coefficient for UB
UC1	m/V^2	-5.69e-11	Yes	Temperature coefficient for UC
AT	m/sec	3.3e4	Yes	Temperature coefficient for saturation velocity
PRT	ohm-um	0	Yes	Temperature coefficient for RDSW
XTI	-	3.0	No	Junction current temperature exponent

Bin Description Parameters

Name	Unit	Default	Bin	Description
LMIN	m	0.0	No	Maximum channel length
LMAX	m	1.0	No	Maximum channel length
WMIN	m	0.0	No	Minimum channel width
WMAX	m	1.0	No	Maximum channel width
BINUNIT				Assumes weff, leff, wref, lref units are in microns when BINUNIT=1 or meters otherwise

Process Parameters

Name	Unit	Default	Bin	Description
GAMMA1	$V^{1/2}$	see Note 8	Yes	Body effect coefficient near the surface
GAMMA2	$V^{1/2}$	see Note 9	Yes	Body effect coefficient in the bulk
VBX	V	see Note 10	Yes	VBX at which the depletion region width equals XT
XT	m	1.55e-7	Yes	Doping depth

Noise Parameters

Name	Unit	Default	Bin	Description
NIOA	-	1.0e20 nmos 9.9e18 pmos	No	Body effect coefficient near the surface
NOIB	-	5.0e4 nmos 2.4e3 pmos	No	Body effect coefficient in the bulk
NOIC	-	-1.4e-12 nmos 1.4e-12 pmos	No	VBX at which the depletion region width equals XT
EM	V/m	$4.1e^7$	No	Flicker noise parameter
AF	-	1.0	No	Flicker noise exponent
KF	-	0.0	No	Flicker noise coefficient
EF	-	1.0	No	Flicker noise frequency exponent

> *Note:* *See also "Using Noise Models" on page 20-99, for Hspice noise model usage (Hspice parameter NLEV overrides Berkeley NOIMOD).*

Junction Parameters

Name	Unit	Default	Bin	Description
ACM	-	10	No	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
JS	A/m^2	0.0	No	Bulk junction saturation current (Default deviates from BSIM3v3 = $1.0e^{-4}$)
JSW	A/m	0.0	No	Sidewall bulk junction saturation current
NJ	-	1	No	Emission coefficient (used only with Berkeley junction model, i.e.,ACM=10-13)
N	-	1	No	Emission coefficient (Hspice-specific), (used only with Hspice junction model, i.e., ACM=0-3)
CJ	F/m^2	$5.79e^{-4}$	No	Zero-bias bulk junction capacitance (Default deviates from BSIM3v3 = $5.0e^{-4}$)
CJSW	F/m	0.0	No	Zero-bias sidewall bulk junction capacitance (Default deviates from BSIM3v3 = $5.0e^{-10}$)
CJSWG	F/m	CJSW	No	Zero-bias gate-edge sidewall bulk junction capacitance (only used with Berkeley junction model, i.e., ACM=10-13)
CJGATE	F/m	CJSW	No	Zero-bias gate-edge sidewall bulk junction capacitance (Hspice-specific) (only used with ACM=3!)
PB, PHIB	V	1.0	No	Bulk junction contact potential
PBSW	V	1.0	No	Sidewall bulk junction contact potential

Name	Unit	Default	Bin	Description
PHP	V	1.0	No	Sidewall bulk junction contact potential (Hspice-specific) (only used with Hspice junction model, i.e., ACM=0-3)
PBSWG	V	PBSW	No	Gate-edge sidewall bulk junction contact potential (only used with Berkeley junction model, i.e., ACM=10-13). Note: there is no equivalent Hspice parameter. Gate-edge contact potential is always set to PHP for Hspice junction model.
MJ	-	0.5	No	Bulk junction grading coefficient
MJSW	-	0.33	No	Sidewall bulk junction grading coefficient
MJSWG	-	MJSW	No	Gate-edge sidewall bulk junction grading coefficient (only used with Berkeley junction model, i.e., ACM=10-13) **Note:** there is no equivalent Hspice parameter. Gate-edge grading coefficient is always set to MJSW for Hspice junction model.

Note: *See for Hspice junction diode model usage.*

NonQuasi-Static (NQS) Parameters

Name	Unit	Default	Bin	Description
ELM	-	5.0	Yes	Elmore constant

Version 3.2 Parameters

Name	Unit	Default	Bin	Description
TOXM	m	TOX	No	Reference gate oxide thickness
VFB	V	See Note 11	Yes	DC flatband voltage
NOFF	-	1.0	Yes	I-V parameter for weak to strong inversion transition
VOFFCV	-	0.0	Yes	C-V parameter for weak to strong inversion transition
JTH	A	0.1	No	Diode limiting current
ALPHA1	V^{-1}	0.0	Yes	Substrate current parameter
ACDE	m/V	1.0	Yes	Exponential coefficient for charge thickness in the accumulation and depletion regions
MOIN	m/V	15.0	Yes	Coefficient for gate-bias dependent surface potential
TPB	V/K	0.0	No	Temperature coefficient of PB
TPBSW	V/K	0.0	No	Temperature coefficient of PBSW
TPBSWG	V/K	0.0	No	Temperature coefficient of PBSWG
TCJ	V/K	0.0	No	Temperature coefficient of CJ
TCJSW	V/K	0.0	No	Temperature coefficient of CJSW
TCJSWG	V/K	0.0	No	Temperature coefficient of CJSWG
LLC	m^{lln}	LL	No	Coefficient of length dependence for C-V channel length offset
LWC	m^{lwn}	LW	No	Coefficient of width dependence for C-V channel length offset
LWLC	$m^{lln+lwn}$	LWL	No	Coefficient of length and width for C-V channel length offset

Name	Unit	Default	Bin	Description
WLC	m^{wln}	WL	No	Coefficient of length dependence for C-V channel width offset
WWC	m^{wwn}	WW	No	Coefficient of width dependence for C-V channel width offset
WWLC	m^{wln+} wwn	WWL	No	Coefficient of length and width cross terms for C-V channel width offset

Notes:

1. If C_{gso} is not given, it is calculated as follows:
   If (*dlc* is given and is greater than *0.0*), then,

   cgso = p1 = max(0,dlc*cox - cgs1)

   Otherwise, *cgso = 0.6*xj*cox*

2. If C_{gdo} is not given, it is calculated as follows:

   if (*dlc* is given and is greater than *0.0*), then,

       cgdo = p2 = max(0,dlc*cox - cgd1)

   Otherwise *cgdo = 0.6*xj*cox*

3. If C_f is not given, it is calculated using:

$$C_f = \frac{2\varepsilon_{ox}}{\pi}\log\left(1 + \frac{4\times10^{-7}}{T_{ox}}\right)$$

4. If V_{th0} is not specified in the .MODEL statement, it is calculated with $V_{fb} = -1$, using:

$$V_{th0} = V_{fb} + \phi_s + K_1\sqrt{\phi_s}$$

5. If K_1 and K_2 are not given, they are calculated using:

$$K_1 = GAMMA_2 + 2K_2\sqrt{\phi_s - V_{bs}}$$

$$K_2 = \frac{(GAMMA_2 - GAMMA_1)(\sqrt{\phi_s - V_{bs}} - \sqrt{\phi_s})}{2\sqrt{\phi_s}(\sqrt{\phi_s - V_{bm}} - \sqrt{\phi_s}) + V_{bm}}$$

6. If n_{ch} is not given, and $GAMMA_1$ is given, n_{ch} is calculated from:

$$n_{ch} = \frac{GAMMA_1^2 C_{OX}^2}{2q\varepsilon_{si}}$$

If both n_{ch} and $GAMMA_1$ are not given, n_{ch} defaults to 1.7e17 per cubic meter and $GAMMA_1$ is calculated from n_{ch}.

7. If *PHI* is not given, it is calculated using:

$$\phi_s = 2\frac{k_B T}{q}\log\left(\frac{n_{ch}}{n_i}\right)$$

$$n_i = 1.45 \times 10^{10}\left(\frac{T}{300.15}\right)^{1.5}\exp\left(21.5565981 - \frac{qE_g(T)}{2k_B T}\right)$$

$$E_g(T) = 1.16 - \frac{7.02 \times 10^{-4}T^2}{T + 1108}$$

8. If $GAMMA_1$ is not given, it is calculated using:

$$GAMMA_1 = \frac{\sqrt{2q\varepsilon_{si}n_{ch}}}{C_{ox}}$$

9. If $GAMMA_2$ is not given, it is calculated using:

$$GAMMA_2 = \frac{\sqrt{2q\varepsilon_{si}n_{sub}}}{C_{ox}}$$

10. If V_{bx} is not given, it is calculated using:

$$V_{bx} = \phi_s - \frac{qn_{ch}X_t^2}{2\varepsilon_{si}}$$

11. 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

The model parameter U0 can be entered in meters or centimeters. U0 is converted to m^2/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

Name	Limits	Comments
TOX	<= 0 Fatal < 10-9 Warn if parmchk=1	
TOXM	<= 0 Fatal < 10-9 Warn if parmchk=1	
XJ	<= 0 Fatal	

Name	Limits	Comments
NGATE	< 0 Fatal > 10^{25} Fatal <= 10^{18} Fatal if parmchk=1	if > 10^{23}　NGATE is multiplied by 10-6. This is done prior to the other limit checks. LEVEL 49 gives < 0 Fatal > 10^{25} Warn <= 10^{18} Warn if paramchk==1
NSUB	<= 0 Fatal <= 10^{14} Warn if parmchk=1 >= 10^{21} Warn if parmchk=1	NSUB is ignored if k1,k2 are defined
NCH	<= 0 Fatal <= 10^{15} Warn if parmchk=1 >= 10^{21} Warn if parmchk=1	if > 10^{20} NCH is multiplied by 10-6. This is done prior to the other limit checks.
NLX	< -Leff Fatal < 0 Warn if parmchk=1	
W0	= -Weff Fatal w0 + Weff < 10^{-7} Warn if paramchk==1	
DVT1W	< 0 Fatal	< 0 LEVEL 49 gives Warn
DVT0	< 0 Warn if paramchk=1	
DVT1	< 0 Fatal	< 0 LEVEL 49 gives Warn
ETA0	<= 0 Warn if paramchk=1	
DSUB	< 0 Fatal	< 0 LEVEL 49 gives Warn
VBM		Ignored if K1,K2 are defined
U0	<= 0 Fatal	
B1	= -Weff Fatal B1 + Weff < 10^{-7} Warn if paramchk=1	

Name	Limits	Comments
VSAT	<= 0 Fatal < 10^3 Warn if paramchk==1	
A1	-	See a2 conditions
A2	< 0.01 Warn and reset a2=0.01 if paramchk=1 > 1 Warn and reset a2=1,a1=0 if paramchk=1	
DELTA	< 0 Fatal	
RDSW	< 0.001 Warn if paramchk=1 and reset rdsw=0	
NFACTOR	< 0 Warn if paramchk=1	
CDSC	< 0 Warn if paramchk=1	
CDSCD	< 0 Warn if paramchk=1	
PCLM	<= 0 Fatal	
PDIBLC1	< 0 Warn if paramchk=1	
PDIBLC2	< 0 Warn if paramchk=1	
PS	< Weff Warn	
DROUT	< 0 Fatal if paramchk=1	LEVEL 49 gives Warn
PSCBE2	<= 0 warn if paramchk=1	
CGS0	< 0 Warn and reset to 0 if paramchk=1	
CGD0	< 0 Warn and reset to 0 if paramchk=1	
CGB0	< 0 Warn and reset to 0 if paramchk=1	
ACDE	< 0.4, > 1.6 Warn	

Name	Limits	Comments
MOIN	< 5.0, > 25 Warn	
IJTH	< 0 Fatal	
NOFF	< 0.1, > 4.0 Warn	

Element Parameters Range Limits

Name	Limits	Comments
PD	< Weff, Warn	
PS	< Weff, Warn	
Leff	< 5.0 x 10^{-8} Fatal	
Weff	< 1.0 x 10^{-7} Fatal	
LeffCV	< 5.0 x 10^{-8} Fatal	
WeffCV	< 1.0 x 10^{-7} Fatal	

LEVEL 49, 53 Equations

The effective channel length and width used in all model equations are:

$$L_{eff} = L_{drawn} - 2\,dL$$

$$W_{eff} = W_{drawn} - 2\,dW$$

$$W'_{eff} = W_{drawn} - 2\,dW'$$

$$Wdrawn = W * WMULT + XW$$

$$Ldrawn = L * LMULT + XL$$

where the unprimed W_{eff} is bias-dependent, and the primed quantity is bias-independent.

$$dW = dW' + dW_g V_{gsteff} + dW_b(\sqrt{\phi_s - V_{bseff}} - \sqrt{\phi_s})$$

$$dW' = W_{int} + \frac{W_L}{L^{WLN}} + \frac{W_W}{W^{WWN}} + \frac{W_{WL}}{L^{WLN}W^{WWN}}$$

$$dL = L_{int} + \frac{L_L}{L^{LLN}} + \frac{L_W}{W^{LWN}} + \frac{L_{WL}}{L^{LLN}W^{LWN}}$$

For C-V calculations dW' is replaced with

$$dW' = DWC + \frac{W_{LC}}{L^{WLN}} + \frac{W_{WC}}{W^{WWN}} + \frac{W_{WLC}}{L^{WLN}W^{WWN}}$$

and dL' is replaced with

$$dL = DLC + \frac{L_{LC}}{L^{LLN}} + \frac{L_{WC}}{W^{LWN}} + \frac{L_{WLC}}{L^{LLN}W^{LWN}}$$

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

```
.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=.5341038 k2=1.703463E-03 k3=-17.24589
+ dvt0=.1767506 dvt1=.5109418 dvt2=-0.05
+ nlx=9.979638E-08 w0=1e-6
+ k3b=4.139039
+ vsat=97662.05 ua=-1.748481E-09 ub=3.178541E-18 uc=1.3623e-10
+ rdsw=298.873 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 dvt1w=5.3e6 dvt2w=-0.0032
+ kt1=-.3 kt2=-.03
+ at=33000
+ ute=-1.5
+ ua1=4.31E-09 ub1=7.61E-18 uc1=-2.378e-10
+ kt1l=1e-8
+ wr=1 b0=1e-7 b1=1e-7 dwg=5e-8 dwb=2e-8 delta=0.015
+ cgdl=1e-10 cgsl=1e-10 cgbo=1e-10 xpart=0.0
+ cgdo=0.4e-9 cgso=0.4e-9
+ clc=0.1e-6
+ cle=0.6
+ ckappa=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
+ nlx=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
+ pclm=2.6813153 pdiblc1=4.003703E-02 pdiblc2=.00329051
+ pdiblcb=-2.e-4
+ drout=.1380235 pscbe1=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
+ ua1=4.31E-09 ub1=7.61E-18 uc1=-2.378e-10
+ kt1l=0
+ wr=1 b0=1e-7 b1=1e-7 dwg=5e-8 dwb=2e-8 delta=0.015
+ cgdl=1e-10 cgsl=1e-10 cgbo=1e-10 xpart=0.0
+ cgdo=0.4e-9 cgso=0.4e-9
+ clc=0.1e-6
+ cle=0.6
+ ckappa=0.6
```

LEVEL 50 Philips MOS9 Model

The Philips MOS Model 9, LEVEL 902, is available as LEVEL 50 in Star-Hspice (based on the "Unclassified Report NL-UR 003/94" by R.M.D.A. Velghe, D.B.M. Klaassen, and F.M. Klaassen).

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, MJSW, 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

Name	Unit	Default (N)	Default (P)	Description
LER	m	1.1e-6	1.25e-6	Reference Leff
WER	m	20.0e-6	20.0e-6	Reference Weff
LVAR	m	-220.0e-9	-460.0e-9	Variation in gate length
LAP	m	100.0e-9	25.0e-9	Lateral diffusion per side
WVAR	m	-25.0e-9	-130.0e-9	Variation in active width
WOT	m	0.0	0.0	Channel-stop diffusion per side
TR	°C	21.0	21.0	Reference temperature for model
VTOR	V	730.0e-3	1.1	Threshold voltage at zero bias
STVTO	V/K	-1.2e-3	-1.7e-3	Temperature dependence of VTO
SLVTO	Vm	-135.0e-9	35.0e-9	Length dependence of VTO

Name	Unit	Default (N)	Default (P)	Description
SL2VTO	Vm^2	0.0	0.0	Second length dependence of VTO
SWVTO	Vm	130.0e-9	50.0e-9	Width dependence of VTO
KOR	$V^{-1/2}$	650.0e-3	470.0e-3	Low-back-bias body factor
SLKO	$V^{-1/2}m$	-130.0e-9	-200.0e-9	Length dependence of KO
SWKO	$V^{-1/2}m$	2.0e-9	115.0e-9	Width dependence of KO
KR	$V^{-1/2}$	110.0e-3	470.0e-3	High-back-bias body factor
SLK	$V^{-1/2}m$	-280.0e-9	-200.0e-9	Length dependence of K
SWK	$V^{-1/2}m$	275.0e-9	115.0e-9	Width dependence of K
PHIBR	V	650.0e-3	650.0e-3	Strong inversion surface potential
VSBXR	V	660.0e-3	0.0	Transition voltage for dual-k-factor model
SLVSBX	Vm	0.0	0.0	Length dependence of *VSBX*
SWVSBX	Vm	-675.0e-9	0.0	Width dependence of *VSBX*
BETSQ	AV^{-2}	83.0e-6	26.1e-6	Gain factor of infinite square transistor
ETABET	-	1.6	1.6	Exponent of temperature dependence of gain factor
THE1R	V^{-1}	190.0e-3	190.0e-3	Gate-induced mobility reduction coefficient
STTHE1R	V^{-1}/K	0.0	0.0	Temperature dependence coefficient of *THE1R*
SLTHE1R	$V^{-1}m$	140.0e-9	70.0e-9	Length dependence coefficient of *THE1R*
STLTHE1	$V^{-1}m/K$	0.0	0.0	Temperature dependence of, length dependence of *THE1R*

Name	Unit	Default (N)	Default (P)	Description
SWTHE1	$V^{-1}m$	-58.0e-9	-80.0e-9	Width dependence coefficient of *THE1R*
THE2R	$V^{-1/2}$	12.0e-3	165.0e-3	Back-bias induced mobility reduction coefficient
STTHE2R	$V^{-1/2}/K$	0.0	0.0	Temperature dependence coefficient of *THE2R*
SLTHE2R	$V^{-1/2}m$	-33.0e-9	-75.0e-9	Length dependence coefficient of *THE2R*
STLTHE2	$V^{-1/2}m/K$	0.0	0.0	Temperature dependence of, length dependence of *THE2R*
SWTHE2	$V^{-1/2}m$	30.0e-9	20.0e-9	Width dependence coefficient of *THE2R*
THE3R	V^{-1}	145.0e-3	27.0e-3	Lateral field induced mobility reduction coefficient
STTHE3R	V^{-1}/K	-660.0e-6	0.0	Temperature dependence coefficient of *THE3R*
SLTHE3R	$V^{-1}m$	185.0e-9	27.0e-9	Length dependence coefficient of *THE3R*
STLTHE3	$V^{-1}m/K$	-620.0e-12	0.0	Temperature dependence of, length dependence of *THE3R*
SWTHE3	$V^{-1}m$	20.0e-9	11.0e-9	Width dependence coefficient of *THE3R*
GAM1R	-	145.0e-3	77.0e-3	Drain-induced threshold shift coefficient, for high gate drive
SLGAM1	-	160.0e-9	105.0e-9	Length dependence of *GAM1R*
SWGAM1	-	-10.0e-9	-11.0e-9	Width dependence of *GAM1R*
ETADSR	-	600.0e-3	600.0e-3	Exponent of drain dependence of *GAM1R*

Name	Unit	Default (N)	Default (P)	Description
ALPR	-	3.0e-3	44.0e-3	Channel length modulation factor
ETAALP	-	150.0e-3	170.0e-3	Exponent of length dependence of *ALPR*
SLALP	-	-5.65e-3	9.0e-3	Coefficient of length dependence of *ALPR*
SWALP	m	1.67e-9	180.0e-12	Coefficient of width dependence of *ALPR*
VPR	V	340.0e-3	235.0e-3	Characteristic voltage for channel length modulation
GAMOOR	-	18.0e-3	7.0e-3	Drain-induced threshold shift coefficient, at zero gate drive, and zero back-bias
SLGAMOO	m^2	20.0e-15	11.0e-15	Length dependence of *GAMOOR*
ETAGAMR	-	2.0	1.0	Exponent of back-bias dependence of zero gate-drive, drain-induced threshold shift
MOR	-	500.0e-3	375.0e-3	Subthreshold slope factor
STMO	K^{-1}	0.0	0.0	Temperature dependence coefficient of *MOR*
SLMO	$m^{1/2}$	280.0e-6	47.0e-6	Length dependence coefficient of *MOR*
ETAMR	-	2.0	1.0	Exponent of back-bias dependence of subthreshold slope
ZET1R	-	420.0e-3	1.3	Weak-inversion correction factor
ETAZET	-	170.0e-3	30.0e-3	Exponent of length dependence of *ZET1R*
SLZET1	-	-390.0e-3	-2.8	Length dependence coefficient of *ZET1R*

Name	Unit	Default (N)	Default (P)	Description
VSBTR	V	2.1	100.0	Limiting voltage for back-bias dependence
SLVSBT	Vm	-4.4e-6	0.0	Length dependence of *VSBTR*
A1R	-	6.0	10.0	Weak-avalanche current factor
STA1	K^{-1}	0.0	0.0	Temperature coefficient of *A1R*
SLA1	m	1.3e-6	-15.0e-6	Length dependence of *A1R*
SWA1	m	3.0e-6	30.0e-6	Width dependence of *A1R*
A2R	V	38.0	59.0	Exponent of weak-avalanche current
SLA2	Vm	1.0e-6	-8.0e-6	Length dependence of *A2R*
SWA2	Vm	2.0e-6	15.0e-6	Width dependence of *A2R*
A3R	-	650.0e-3	520.0e-3	Factor of minimum drain bias above which avalanche sets in
SLA3	m	-550.0e-9	-450.0e-9	Length dependence of *A3R*
SWA3	m	0.0	-140.0e-9	Width dependence of *A3R*
TOX	m	25.0e-9	25.0e-9	Oxide thickness
COL	F/m	320.0e-12	320.0e-12	Gate overlap capacitance per unit width
WDOG	m	0	0	Characteristic drawn gate width below which dogboning appears
FTHE1	-	0	0	Coefficient describing the width dependence of THE1 for W < WDOG
NFMOD		0		Flicker noise selector 0 selects old flicker noise model added in release 98.4
NTR	J	24.4e-21	21.1e-21	Thermal noise coefficient

Name	Unit	Default (N)	Default (P)	Description
NFR	V^2	70.0e-12	21.4e-12	Flicker noise coefficient
NFAR	$V^{-1}m^{-2}$	7.15e+22	1.53xe+22	1st flicker noise coefficient added in release 98.4
NFBR	$V^{-1}m^{-2}$	2.16e+06	4.06e+06	2nd flicker noise coefficient added in release 98.4
NFCR	V^{-1}	0.0	2.92e-10	3rd flicker noise coefficient added in release 98.4

JUNCAP Model Parameters

Name	Unit	Default	Description
JUNCAP	-	0	JUNCAP flag: 0-off, 1-on
DTA	^oc	0.0	Temperature offset of the JUNCAP element with respect to T_A
VR	V	0.0	Voltage at which the parameters have been determined
JSGBR	$A*m^{-2}$	1.00e-3	Bottom saturation-current density due to electron-hole generation at $V=V_R$
JSDBR	$A*m^{-2}$	1.00e-3	Bottom saturation-current density due to diffusion from back contact
JSGSR	$A*m^{-1}$	1.00e-3	Sidewall saturation-current density due to electron-hole generation at $V=V_R$
JSDSR	$A*m^{-1}$	1.00e-3	Sidewall saturation-current density due to diffusion from back contact
JSGGR	$A*m^{-1}$	1.00e-3	Gate edge saturation-current density due to electron-hole generation at $V=V_R$

Name	Unit	Default	Description
JSDGR	$A*m^{-1}$	1.00e-3	Gate edge saturation-current density due to diffusion from back contact
NB	-	1.00	Emission coefficient of the bottom forward current
NS	-	1.00	Emission coefficient of the sidewall forward current
NG	-	1.00	Emission coefficient of the gate edge forward current
CJBR	$F*m^{-2}$	1.00e-12	Bottom junction capacitance at $V=V_R$
CJSR	$F*m^{-1}$	1.00e-12	Sidewall junction capacitance at $V=V_R$
CJGR	$F*m^{-1}$	1.00e-12	Gate edge junction capacitance at $V=V_R$
VDBR	v	1.00	Diffusion voltage of the bottom junction at $T=T_R$
VDSR	v	1.00	Diffusion voltage of the sidewall junction at $T=T_R$
VDGR	v	1.00	Diffusion voltage of the gate edge junction at $T=T_R$
PB	-	0.40	Bottom-junction grading coefficient
PS	-	0.40	Sidewall-junction grading coefficient
PG	-	0.40	Gate edge-junction grading coefficient

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 ^{o}C in Star-Hspice, but is 27 ^{o}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.

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.

```
.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   sta1 = 0   sla1 = 0   swa1 = 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 (Rii) 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 Rii-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 Vth, 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

Parameter	Default	Binnable	Description
LEVEL (SPICE3 parameter)	14	NA	SPICE3 model selector
VERSION	4.0.0	NA	Model version number
BINUNIT	1	NA	Binning unit selector
PARAMCHK	1	NA	Switch for parameter value check
MOBMOD	1	NA	Mobility model selector
RDSMOD	0	NA	Bias-dependent source/drain resistance model selector
IGCMOD	0	NA	Gate-to-channel tunneling current model selector
IGBMOD	0	NA	Gate-to-substrate tunneling current model selector
CAPMOD	2	NA	Capacitance model selector
RGATEMOD	0 (no gate resistance)		Gate resistance model selector
RBODYMOD	0 (network off)	NA	Substrate resistance network model selector
TRNQSMOD	0	NA	Transient NQS model selector
ACNQSMOD	0	NA	AC small-signal NQS model selector
FNOIMOD	1	NA	Flicker noise model selector

Parameter	Default	Binnable	Description
TNOIMOD	0	NA	Thermal noise model selector
DIOMOD	1	NA	Source/drain junction diode IV model selector
PERMOD	1	NA	Whether PS/PD includes the gate-edge perimeter
GEOMOD	0 (isolated)	NA	Geometry-dependent parasitics model selector
RGEOMOD	0	NA	Source/drain diffusion resistance and contact model selector

Process Parameters

Parameter	Default	Binnable	Description
EPSROX	3.9 (SiO_2)	No	Gate dielectric constant relative to vacuum
TOXE	3.0e-9m	No	Electrical gate equivalent oxide thickness
TOXP	TOXE	No	Physical gate equivalent oxide thickness
TOXM	TOXE	No	Tox at which parameters are extracted
DTOX	0.0m	No	Defined as (TOXE-TOXP)
XJ	1.5e-7m	Yes	S/D junction depth
GAMMA1 ($\gamma 1$ in equation)	calculated ($V^{1/2}$)	Yes	Body-effect coefficient near the surface
GAMMA2 ($\gamma 2$ in equation)	calculated ($V^{1/2}$)	Yes	Body-effect coefficient in the bulk
NDEP	1.7e17cm^{-3}	Yes	Channel doping concentration at depletion edge for zero body bias
NSUB	6.0e16cm^{-3}	Yes	Substrate doping concentration
NGATE	0.0cm^{-3}	Yes	Poly Si gate doping concentration

Parameter	Default	Binnable	Description
NSD	$1.0e20cm^{-3}$	Yes	Source/drain doping concentration
VBX	calculated (v)	No	V_{bs} at which the depletion region width equals XT
XT	1.55e-7m	Yes	Doping depth
RSH	0.0ohm/ square	No	Source/drain sheet resistance
RSHG	0.1ohm/ square	No	Gate electrode sheet resistance

Basic Model Parameters

Parameter	Default	Binnable	Description
VTH0 or VTHO	0.7V (NMOS) -0.7V (PMOS)	Yes	Long-channel threshold voltage at $V_{bs}=0$
VFB	-1.0V	Yes	Flat-band voltage PHIN
PHIN	0.0V	Yes	Non-uniform vertical doping effect on surface potential
K1	$0.5V^{1/2}$	Yes	First-order body bias coefficient
K2	0.0	Yes	Second-order body bias coefficient
K3	80.0	Yes	Narrow width coefficient
K3B	$0.0V^{-1}$	Yes	Body effect coefficient of K3
W0	2.5e-6m	Yes	Narrow width parameter
LPE0	1.74e-7m	Yes	Lateral non-uniform doping parameter
LPEB	0.0m	Yes	Lateral non-uniform doping effect on K1

Parameter	Default	Binnable	Description
VBM	-3.0V	Yes	Maximum applied body bias in VTH0 calculation
DVT0	2.2	Yes	First coefficient of short-channel effect on V_{th}
DVT1	0.53	Yes	Second coefficient of short-channel effect on V_{th}
DVT2	-0.032V^{-1}	Yes	Body-bias coefficient of short-channel effect on V_{th}
DVTP0	0.0m	Yes	First coefficient of drain-induced V_{th} shift due to for long-channel pocket devices
DVTP1	0.0V^{-1}	Yes	First coefficient of drain-induced V_{th} shift due to for long-channel pocket devices
DVT0W	0.0	Yes	First coefficient of narrow width effect on V_{th} for small channel length
DVT1W	5.3e6m^{-1}	Yes	Second coefficient of narrow width effect on V_{th} for small channel length
DVT2W	-0.032V^{-1}	Yes	Body-bias coefficient of narrow width effect for small channel length
U0	0.067m^2/(Vs) (NMOS); 0.025 m^2/(Vs) (PMOS)	Yes	Low-field mobility
UA	1.0e-9m/V for MOBMOD=0 and 1; 1.0e-15m/V for MOBMOD=2	Yes	Coefficient of first-order mobility degradation due to vertical field
UB	1.0e-19m^2/V^2	Yes	Coefficient of second-order mobility degradation due to vertical field

Parameter	Default	Binnable	Description
UC	-0.0465V^{-1} for MOB-MOD=1; -0.0465e-9 m/V^2 for MOBMOD=0 and 2	Yes	Coefficient of mobility degradation due to body-bias effect
EU	1.67 (NMOS); 1.0 (PMOS)	No	Exponent for mobility degradation of MOBMOD=2
VSAT	8.0e4m/s	Yes	Saturation velocity
A0	1.0	Yes	Coefficient of channel-length dependence of bulk charge effect
AGS	0.0V^{-1}	Yes	Coefficient of V_{gs} dependence of bulk charge effect
B0	0.0m	Yes	Bulk charge effect coefficient for channel width
B1	0.0m	Yes	Bulk charge effect width offset
KETA	-0.047V^{-1}	Yes	Body-bias coefficient of bulk charge effect
A1	0.0V^{-1}	Yes	First non-saturation effect parameter
A2	1.0	Yes	Second non-saturation factor
WINT	0.0m	No	Channel-width offset parameter
LINT	0.0m	No	Channel-length offset parameter
DWG	0.0m/V	Yes	Coefficient of gate bias dependence of W_{eff}
DWB	0.0m/V$^{1/2}$	Yes	Coefficient of body bias dependence of W_{eff} bias dependence
VOFF	-0.08V	Yes	Offset voltage in subthreshold region for large W and L

Parameter	Default	Binnable	Description
VOFFL	0.0mV	No	Channel-length dependence of VOFF
MINV	0.0	Yes	V_{gsteff} fitting parameter for moderate inversion condition
NFACTOR	1.0	Yes	Subthreshold swing factor
ETA0	0.08	Yes	DIBL coefficient in subthreshold region
ETAB	$-0.07V^{-1}$	Yes	Body-bias coefficient for the subthreshold DIBL effect
DSUB	DROUT	Yes	DIBL coefficient exponent in subthreshold region
CIT	$0.0F/m^2$	Yes	Interface trap capacitance
CDSC	$2.4e-4F/m^2$	Yes	Coupling capacitance between source/drain and channel
CDSCB	$0.0F/(Vm^2)$	Yes	Body-bias sensitivity of CDSC
CDSCD	$0.0(F/Vm^2)$	Yes	Drain-bias sensitivity of DCSC
PCLM	1.3	Yes	Channel-length modulation parameter
PDIBLC1	0.39	Yes	Parameter for DIBL effect on Rout
PDIBLC2	0.0086	Yes	Parameter for DIBL effect on Rout
PDIBLCB	$0.0V^{-1}$	Yes	Body bias coefficient of DIBL effect on Rout
DROUT	0.56	Yes	Channel-length dependence of DIBL effect on Rout
PSCBE1	4.24e8V/m	Yes	First substrate current induced body-effect parameter
PSCBE2	1.0e-5m/V	Yes	Second substrate current induced body-effect parameter

Parameter	Default	Binnable	Description
PVAG	0.0	Yes	Gate-bias dependence of Early voltage
DELTA (δ in equation)	0.01V	Yes	Parameter for DC V_{dseff}
FPROUT	$0.0V/m^{0.5}$	Yes	Effect of pocket implant on Rout degradation
PDITS	$0.0V^{-1}$	Yes	Impact of drain-induced V_{th} shift on Rout.
PDITSL	$0.0m^{-1}$	No	Channel-length dependence of drain-induced V_{th} shift for Rout.
PDITSD	$0.0V^{-1}$	Yes	Vds dependence of drain-induced V_{th} shift for Rout

Parameters for Asymmetric and Bias-Dependent R_{ds} Model

Parameter	Default	Binnable	Description
RDSW	200.0 ohm(μm)^{WR}	Yes	Zero bias LLD resistance per unit width for RDSMOD=0
RDSWMIN	0.0 ohm(μm)^{WR}	No	LDD resistance per unit width at high V_{gs} and zero V_{bs} for RDSMOD=0
RDW	100.0 ohm(μm)^{WR}	Yes	Zero bias lightly-doped drain resistance $R_d(v)$ per unit width for RDSMOD=1
RDWMIN	0.0 ohm(μm)^{WR}	No	Lightly-doped drain resistance per unit width at high V_{gs} and zero V_{bs} for RDSMOD=1
RSW	100.0 ohm(μm)^{WR}	Yes	Zero bias lightly-doped source resistance $R_s(V)$ per unit width for RDSMOD=1

Parameter	Default	Binnable	Description
RSWMIN	0.0 ohm$(\mu m)^{WR}$	No	Lightly-doped source resistance per unit width at high V_{gs} and zero V_{bs} for RDSMOD=1
PRWG	1.0V^{-1}	Yes	Gate-bias dependence of LDD resistance
PRWB	0.0V$^{-0.5}$	Yes	Body-bias dependence of LDD resistance
WR	1.0	Yes	Channel-width dependence parameter of LDD resistance
NRS	1.0	No	Number of source diffusion squares
NRD	1.0	No	Number of drain diffusion squares

Impact Ionization Current Model Parameters

Parameter	Default	Binnable	Description
ALPHA0	0.0Am/V	Yes	First parameter of impact ionization current
ALPHA1	0.0A/V	Yes	Isub parameter for length scaling
BETA0	30.0V	Yes	The second parameter for impact ionization current

Gate-Induced Drain Leakage Model Parameters

Parameter	Default	Binnable	Description
AGIDL	0.0ohm	Yes	Pre-exponential coefficient for GIDL
BGIDL	2.3e9V/m	Yes	Exponential coefficient for GIDL
CGIDL	0.5V^3	Yes	Parameter for body-bias effect on GIDL
DGIDL	0.8V	Yes	Fitting parameter for band bending for GIDL

Gate Dielectric Tunneling Current Model Parameters

Parameter	Default	Binnable	Description
AIGBACC	0.43 $(F_s{}^2/_g)^{0.5}m^{-1}$	Yes	Parameter for I_{gb} in accumulation
BIGBACC	0.054 $(F_s{}^2/_g)^{0.5}$ $m^{-1}V^{-1}$	Yes	Parameter for I_{gb} in accumulation
CIGBACC	0.075V^{-1}	Yes	Parameter for I_{gb} in accumulation
NIGBACC	1.0	Yes	Parameter for I_{gb} in accumulation
AIGBINV	0.35 $(F_s{}^2/_g)^{0.5}m^{-1}$	Yes	Parameter for I_{gb} in inversion
BIGBINV	0.03 $(F_s{}^2/_g)^{0.5}m^{-1}V^{-1}$	Yes	Parameter for I_{gb} in inversion
CIGBINV	0.0006V^{-1}	Yes	Parameter for I_{gb} in inversion
EIGBINV	1.1V	Yes	Parameter for I_{gb} in inversion
NIGBINV	3.0	Yes	Parameter for I_{gb} in inversion
AIGC	0.054 (NMOS) and 0.31 (PMOS) $(F_s{}^2/_g)^{0.5}m^{-1}$	Yes	Parameter for I_{gcs} and I_{gcd}
BIGC	0.054 (NMOS) and 0.024 (PMOS) $(F_s{}^2/_g)^{0.5}$ $m^{-1}V^{-1}$	Yes	Parameter for I_{gcs} and I_{gcd}
CIGC	0.075 (NMOS) and 0.03(PMOS) V^{-1}	Yes	Parameter for I_{gcs} and I_{gcd}
AIGSD	0.43 (NMOS) and 0.31 (PMOS) $(F_s{}^2/_g)^{0.5}$ m^{-1}	Yes	Parameter for I_{gs} and I_{gd}
BIGSD	0.054 (NMOS) 0.024 (PMOS) $(F_s{}^2/_g)^{0.5}$ $m^{-1}V^{-1}$	Yes	Parameter for I_{gs} and I_{gd}
CIGSD	0.075 (NMOS) and 0.03 (PMOS) V^{-1}	Yes	Parameter for I_{gs} and I_{gd}
DLCIG	LINT	Yes	Source/drain overlap length for Igs and Igd
NIGC	1.0	Yes	Parameter for I_{gcs}, I_{gcd}, I_{gs} and I_{gd}

Parameter	Default	Binnable	Description
POXEDGE	1.0	Yes	Factor for the gate oxide thickness in source/drain overlap regions
PIGCD	1.0	Yes	V_{ds} dependence of I_{gcs} and I_{gcd}
NTOX	1.0	Yes	Exponent for the gate oxide ratio
TOXREF	3.0e-9m	No	Nominal gate oxide thickness for gate dielectric tunneling current model only

Charge and Capacitance Model Parameters

Parameter	Default	Binnable	Description
XPART	0.0	No	Charge partition parameter
CGSO	calculated (F/m)	No	Non LDD region source-gate overlap capacitance per unit channel width
CGDO	calculated (F/m)	No	Non LDD region drain-gate overlap capacitance per unit channel width
CGBO	0.0 (F/m)	No	Gate-bulk overlap capacitance per unit channel length
CGSL	0.0F/m	Yes	Overlap capacitance between gate and lightly-doped source region
CGDL	0.0F/m	Yes	Overlap capacitance between gate and lightly-doped source region
CKAPPAS	0.6V	Yes	Coefficient of bias-dependent overlap capacitance for the source side
CKAPPAD	CKAPPAS	Yes	Coefficient of bias-dependent overlap capacitance for the drain side
CF	calculated (F/m)	Yes	Fringing field capacitance
CLC	1.0e-7m	Yes	Constant term for the short channel model

Parameter	Default	Binnable	Description
CLE	0.6	Yes	Exponential term for the short channel model
DLC	LINT (m)	No	Channel-length offset parameter for CV model
DWC	WINT (m)	No	Channel-width offset parameter for CV model
VFBCV	-1.0V	Yes	Flat-band voltage parameter (for CAPMOD=0 only)
NOFF	1.0	Yes	CV parameter in $V_{gsteff,CV}$ for week to strong inversion
VOFFCV	0.0V	Yes	CV parameter in $V_{gsteff,CV}$ for week to strong inversion
ACDE	1.0m/V	Yes	Exponential coefficient for charge thickness in CAPMOD=2 for accumulation and depletion regions
MOIN	15.0	Yes	Coefficient for the gate-bias dependent surface potential

High-Speed/RF Model Parameters

Parameter	Default	Binnable	Description
XRCRG1	12.0	Yes	Parameter for distributed channel-resistance effect for both intrinsic-input resistance and charge-deficit NQS models
XRCRG2	1.0	Yes	Parameter to account for the excess channel diffusion resistance for both intrinsic input resistance and charge-deficit NQS models
RBPB	50.0ohm	No	Resistance connected between bNodePrime and bNode

Parameter	Default	Binnable	Description
RBPD	50.0ohm	No	Resistance connected between bNodePrime and dbNode
RBPS	50.0ohm	No	Resistance connected between bNodePrime and sbNode
RBDB	50.0ohm	No	Resistance connected between dbNode and dbNode
RBSB	50.0ohm	No	Resistance connected between sbNode and bNode
GBMIN	1.0e-12mho	No	Conductance in parallel with each of the five substrate resistances to avoid potential numerical instability due to unreasonably too large a substrate resistance

Flicker and Thermal Noise Model Parameters

Parameter	Default	Binnable	Description
NOIA	6.25e41 $(eV)^{-1}s^{1-EF}m^{-3}$ for NMOS; 6.188e40 $(eV)^{-1}s^{1-EF}m^{-3}$ for PMOS	No	Flicker noise parameter A
NOIB	3.125e26 $(eV)^{-1}s^{1-EF}m^{-1}$ for NMOS; 1.5e25 $(eV)^{-1}s^{1-EF}m^{-1}$ for PMOS	No	Flicker noise parameter B
NOIC	8.75 $(eV)^{-1}S^{1-EF}m$	No	Flicker noise parameter C
EM	4.1e7V/m	No	Saturation field
AF	1.0	No	Flicker noise exponent
EF	1.0	No	Flicker noise frequency exponent
KF	0.0 $A^{2-EF}s^{1-EF}F$	No	Flicker noise coefficient

Parameter	Default	Binnable	Description
NTNOI	1.0	No	Noise factor for short-channel devices for TNOIMOD=0 only
TNOIA	1.5	No	Coefficient of channel-length dependence of total channel thermal noise
TNOIB	3.5	No	Channel-length dependence parameter for channel thermal noise partitioning

Layout-Dependent Parasitics Model Parameters

Parameter	Default	Binnable	Description
DMCG	0.0m	No	Distance from S/D contact center to the gate edge
DMCI	DMCG	No	Distance from S/D contact center to the isolation edge in the channel-length direction
DMDG	0.0m	No	Same as DMCG but for merged device only
DMCGT	0.0m	No	DMCG of test structures
NF	1	No	Number of device figures
DWJ	DWC (in CVmodel)	No	Offset of the S/D junction width
MIN	0	No	Whether to minimize the number of drain or source diffusions for even-number fingered device
XGW	0.0m	No	Distance from the gate contact to the channel edge
XGL	0.0m	No	Offset of the gate length due to variations in patterning
NGCON	1	No	Number of gate contacts

Asymmetric Source/Drain Junction Diode Model Parameters

Parameter	Default	Binnable	Description
IJTHSREV	IJTHSREV=0.1A	No	Limiting current in reverse bias region
IJTHDREV	IJTHDREV= IJTHSREV	No	Limiting current in reverse bias region
IJTHSFWD	IJTHSFWD=0.1A	No	Limiting current in forward bias region
IJTHDFWD	IJTHDFWD= IJTHSFWD	No	Limiting current in forward bias region
XJBVS	XJBVS=1.0	No	Fitting parameter for diode breakdown
XJBVD	XJBVD=XJBVS	No	Fitting parameter for diode breakdown
BVS	BVS=10.0V	No	Breakdown voltage
BVD	BVD=BVS	No	Breakdown voltage
JSS	JSS=1.0e-4A/m^2	No	Bottom junction reverse saturation current density
JSD	JSD=JSS	No	Bottom junction reverse saturation current density
JSWS	JSWS=0.0A/m	No	Isolation-edge sidewall reverse saturation current density
JSWD	JSWD=JSWS	No	Isolation-edge sidewall reverse saturation current density
JSWGS	JSWGS=0.0A/m	No	Gate-edge sidewall reverse saturation current density
JSWGD	JSWGD=JSWGS	No	Gate-edge sidewall reverse saturation current density
CJS	CJS=5.0e-4 F/m^2	No	Bottom junction capacitance per unit area at zero bias
CJD	CJD=CJS	No	Bottom junction capacitance per unit area at zero bias

Parameter	Default	Binnable	Description
MJS	MJS=0.5	No	Bottom junction capacitance grading coefficient
MJD	MJD=MJS	No	Bottom junction capacitance grading coefficient
MJSWS	MJSWS=0.33	No	Isolation-edge sidewall junction capacitance grading coefficient
MJSWD	MJSWD=MJSWS	No	Isolation-edge sidewall junction capacitance grading coefficient
CJSWS	CJSWS=5.0e-10 F/m	No	Isolation-edge sidewall junction capacitance per unit area
CJSWD	CJSWD=CJSWS	No	Isolation-edge sidewall junction capacitance per unit area
CJSWGS	CJSWGS=CJSWS	No	Gate-edge sidewall junction capacitance per unit length
CJSWGD	CJSWGD=CJSWS	No	Gate-edge sidewall junction capacitance per unit length
MJSWGS	MJSWGS=MJSWS	No	Gate-edge sidewall junction capacitance grading coefficient
MJSWGD	MJSWGD=MJSWS	No	Gate-edge sidewall junction capacitance grading coefficient
PBS	PBS=1.0V	No	Bottom junction built-in potential
PBD	PBD=PBS	No	Bottom junction built-in potential
PBSWS	PBSWS=1.0V	No	Isolation-edge sidewall junction built-in potential

Parameter	Default	Binnable	Description
PBSWD	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

Temperature Dependence Parameters

Parameter	Default	Binnable	Description
TNOM	27°C	No	Temperature at which parameters are extracted
UTE	-1.5	Yes	Mobility temperature exponent
KT1	-0.11V	Yes	Temperature coefficient for threshold voltage
KT1L	0.0Vm	Yes	Channel length dependence of the temperature coefficient for threshold voltage
KT2	0.022	Yes	Body-bias coefficient of *Vth* temperature effect
UA1	1.0e-9m/V	Yes	Temperature coefficient for UA
UB1	-1.0e-18 (m/V^2)	Yes	Temperature coefficient for UB
UC1	0.067V^{-1} for MOBMOD=1; 0.025m/V^2 for MOBMOD=0 and 2	Yes	Temperature coefficient for UC
AT	3.3e4m/s	Yes	Temperature coefficient for saturation velocity

Parameter	Default	Binnable	Description
PRT	0.0ohm-m	Yes	Temperature coefficient for Rdsw
NJS, NJD	NJS=1.0; NJD=NJS	No	Emission coefficients of junction for source and drain junctions, respectively
XTIS, XTID	XTIS=3.0; XTID=XTIS	No	Junction current temperature exponents for source and drain junction, respectively
TPB	0.0V/K	No	Temperature coefficient of PB
TPBSW	0.0V/K	No	Temperature coefficient of PBSW
TPBSWG	0.0V/K	No	Temperature coefficient of PBSWG
TCJ	$0.0K^{-1}$	No	Temperature coefficient of CJ
TCJSW	$0.0K^{-1}$	No	Temperature coefficient of CJSW
TCJSWG	$0.0K^{-1}$	No	Temperature coefficient of CJSWG

dW and *dL* Parameters

Parameter	Default	Binnable	Description
WL	$0.0m^{WLN}$	No	Coefficient of length dependence for width offset
WLN	1.0	No	Power of length dependence of width offset
WW	$0.0m^{WWN}$	No	Coefficient of width dependence for width offset
WWN	1.0	No	Power of width dependence of width offset
WWL	$0.0\ m^{WWN+WLN}$	No	Coefficient of length and width cross term dependence for width offset
LL	$0.0m^{LLN}$	No	Coefficient of length dependence for length offset
LLN	1.0	No	Power of length dependence for length offset

Parameter	Default	Binnable	Description
LW	$0.0m^{LWN}$	No	Coefficient of width dependence for length offset
LWN	1.0	No	Power of width dependence for length offset
LWL	$0.0\ m^{LWN+LLN}$	No	Coefficient of length and width cross term dependence for length offset
LLC	LL	No	Coefficient of length dependence for CV channel length offset
LWC	LW	No	Coefficient of width dependence for CV channel length offset
LWLC	LWL	No	Coefficient of length and width cross-term dependence for CV channel length offset
WLC	WL	No	Coefficient of length dependence for CV channel width offset
WWC	WW	No	Coefficient of width dependence for CV channel width offset
WWLC	WWL	No	Coefficient of length and width cross-term dependence for CV channel width offset

Range Parameters for Model Application

Parameter	Default	Binnable	Description
LMIN	0.0m	No	Minimum channel length
LMAX	1.0m	No	Maximum channel length
WMIN	0.0m	No	Minimum channel width
WMAX	1.0m	No	Maximum channel width

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:

$V_G = V_{GB}$ Intrinsic gate-to-bulk voltage

$V_S = V_{SB}$ Intrinsic source-to-bulk voltage

$V_D = V_{DB}$ 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 VFB, VTO and TCV) 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 η factor for effective mobility calculation.

Equivalent Circuit

Gate   *Intrinsic EKV model elements*

Bulk

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

Name	Unit	Default	Description
L	m	-	Channel length
W	m	-	Channel width
M or NP	-	1.0	Parallel multiple device number
N or NS	-	1.0	Series multiple device number

EKV Intrinsic Model Parameters
Process Related Parameters

Name	Unit	Default	Range	Description
COX[a]	F/m^2	0.7E-3	-	Gate oxide capacitance per unit area
XJ	m	0.1E-6	$\geq$ 1.0E-9	Junction depth
DW[b]	m	0	-	Channel width correction
DL	m	0	-	Channel length correction

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

Name	Unit	Default[a]	Range	Description
VTO[b]	V	0.5	-	Long-channel threshold voltage
GAMMA	$\sqrt{V}$	1.0	$\geq$ 0	Body effect parameter
PHI	V	0.7	$\geq$ 0.1	Bulk Fermi potential (*2)
KP	A/V^2	50.0E-6	-	Transconductance parameter
E0 (EO)	V/m	1.0E12	$\geq$ 1E5	Mobility reduction coefficient

Name	Unit	Default[a]	Range	Description
UCRIT	V/m	2.0E6	$\geq$ 1E5	Longitudinal critical field

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.

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.

Name	Unit[a]	Default	Range	Description
TOX[b]	m	-	$\geq$ 0	Oxide thickness
NSUB[c]	cm^{-3}	-	$\geq$ 0	Channel doping
VFB[d]	V	-	-	Flat-band voltage
UO[e]	$cm^2/(Vs)$	-	$\geq$ 0	Low-field mobility
VMAX[f]	m/s	-	$\geq$ 0	Saturation velocity
THETA[g]	$1/V$	0	$\geq$ 0	Mobility reduction coefficient

a. Note the choice of cm as basic unit for NSUB and UO, while TOX and VMAX are in m

b. Optional parameter used to calculate COX.

c. Optional parameter accounting for the dependence of GAMMA on COX, as well as for calculation of PHI.

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

Name	Unit	Default	Range	Description
LAMBDA	-	0.5	≥ 0	Depletion length coefficient (channel length modulation)
WETA	-	0.25	-	Narrow-channel effect coefficient
LETA	-	0.1	-	Short-channel effect coefficient

Reverse Short-channel Effect Parameters

Name	Unit	Default	Range	Description
Q0 (QO)	$A \cdot s / m^2$	0	-	Reverse short channel effect peak charge density
LK	m	0.29E-6	$\geq 1.0E-8$	Reverse short channel effect characteristic length

Impact Ionization Related Parameters

Name	Unit	Default	Range	Description
IBA	$1 / m$	0	-	First impact ionization coefficient
IBB	V / m	3.0E8	$\geq 1.0E8$	Second impact ionization coefficient
IBN	-	1.0	≥ 0.1	Saturation voltage factor for impact ionization

Intrinsic Model Temperature Parameters

Name	Unit	Default	Description
TCV	V / K	1.0E-3	Threshold voltage temperature coefficient
BEX	-	-1.5	Mobility temperature exponent
UCEX	-	0.8	Longitudinal critical field temperature exponent
IBBT	$1 / K$	9.0E-4	Temperature coefficient for IBB

Matching Parameters

Name	Unit	Default	Description
AVTO	Vm	0^a	Area related threshold voltage mismatch parameter
AKP	m	0	Area related gain mismatch parameter
AGAMMA	$\sqrt{V}m$	0	Area related body effect mismatch parameter

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

Name	Unit	Default	Description
KF	-^{a}	0	Flicker noise coefficient
AF	-	1	Flicker noise exponent

a. Unit of KF may depend on flicker noise model chosen if options are available.

Setup Parameters

Name	Unit	Default	Description
NQS^a	-	0	Non-Quasi-Static (NQS) operation switch
SATLIM^b	-	exp(4)	Ratio defining the saturation limit i_f/i_r
XQC^c	-	0.4	Charge/capacitance model selector

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

$$\varepsilon_{si} = SCALE \cdot 104.5 \times 10^{-12} [F/m] \qquad \text{Permittivity of silicon}$$

$$\varepsilon_{ox} = SCALE \cdot 34.5 \times 10^{-12} [F/m] \qquad \text{Permittivity of silicon dioxide}$$

$$q = 1.602 \times 10^{-19} [C] \qquad \text{Magnitude of electron charge}$$

$$k = 1.3807 \times 10^{-23} [JK^{-1}] \qquad \text{Boltzmann constant}$$

$$T_{ref} = 300.15 [K] \qquad \text{Reference temperature}$$

$$T_{nom} \ [K] \qquad \textit{Nominal} \text{ temperature of model parameters}$$

$$T \ [K] \qquad \text{Model simulation temperature}$$

$$V_t(T) = \frac{k \cdot T}{q} \qquad \text{Thermal voltage}$$

$$E_g(T) = \left(1.16 - 0.000702 \cdot \frac{T^2}{T + 1108}\right)[eV] \qquad \text{Energy gap}$$

$$n_i(T) = 1.45 \times 10^{16} \cdot \left(\frac{T}{T_{ref}}\right) \cdot \exp\left(\frac{E_g(T_{ref})}{2 \cdot V_t(T_{ref})} - \frac{E_g(T)}{2 \cdot V_t(T)}\right)[m^{-3}] \qquad \textit{Intrinsic carrier concentration}$$

Parameter Preprocessing

Handling of Model Parameters for P-channel MOSFETs

For P-channel devices, the sign of VFB, VTO and TCV is inversed before processing. Therefore, VTO and TCV are usually positive and VFB 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:

$$\text{COX} = \begin{cases} \varepsilon_{ox} / \text{TOX} & \text{for: } \text{TOX} > 0 \\ default & otherwise \end{cases}$$

If GAMMA is not specified, then it is initialized as:

$$\text{GAMMA} = \begin{cases} \dfrac{\sqrt{2q\varepsilon_{si} \cdot (\text{NSUB} \cdot 10^6)}}{\text{COX}} & \text{for: } \text{NSUB} > 0 \\ default & otherwise \end{cases}$$

If PHI is not specified, then it is initialized as:

$$\text{PHI} = \begin{cases} 2V_t(T_{nom}) \cdot \ln\left(\dfrac{\text{NSUB} \cdot 10^6}{n_i(T_{nom})}\right) & \text{for: } \text{NSUB} > 0 \\ default & otherwise \end{cases}$$

If VTO is not specified, then it is initialized as:

$$\text{VTO} = \begin{cases} \text{VFB} + \text{PHI} + \text{GAMMA} \cdot \sqrt{\text{PHI}} & if \text{ VFB } specified \\ default & otherwise \end{cases}$$

If DP is not specified, then it is initialized as:

$$\text{KP} = \begin{cases} (\text{UO} \cdot 10^{-4}) \cdot \text{COX} & \text{for: } \text{UO} > 0 \\ default & otherwise \end{cases}$$

If UCRIT is not specified, then it is initialized as:

$$
\texttt{UCRIT} = \begin{cases} \texttt{VMAX}/(\texttt{UO} \cdot 10^{-4}) & \text{for: } \texttt{VMAX} > 0, \texttt{UO} > 0 \\ default & otherwise \end{cases}
$$

If E0 is not specified, then a simplified mobility model is used with the parameter THETA:

$$
\texttt{E0} = \begin{cases} 0 & if\ \texttt{THETA}\ specified \\ default & otherwise \end{cases}
$$

Note: *The value zero is given to* `E0` *here, indicating that the simplified mobility model is used in conjunction with* `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

$$
\texttt{VTO}(T) = \texttt{VTO} - \texttt{TCV} \cdot (T - T_{nom})
$$

$$
\texttt{KP}(T) = \texttt{KP} \cdot \left(\frac{T}{T_{nom}}\right)^{\texttt{BEX}}
$$

$$
\texttt{UCRIT}(T) = \texttt{UCRIT} \cdot \left(\frac{T}{T_{nom}}\right)^{\texttt{UCEX}}
$$

$$\text{PHI}(T) = \text{PHI} \cdot \frac{T}{T_{nom}} - 3 \cdot V_t \cdot \ln\left(\frac{T}{T_{nom}}\right) - E_g(T_{nom}) \cdot \frac{T}{T_{nom}} + E_g(T)$$

$$\text{IBB}(T) = \text{IBB} \cdot [1.0 + \text{IBBT} \cdot (T - T_{nom})]$$

Bulk Referenced Intrinsic Voltages

Voltages are all referred to the local substrate ("Bulk Reference and Symmetry" on page 22-85):

$$V_G = V_{GB} = V_{GS} - V_{BS} \qquad \text{Intrinsic gate-to-bulk voltage}$$

$$V_S = V_{SB} = -V_{BS} \qquad \text{Intrinsic source-to-bulk voltage}$$

$$V_D = V_{DB} = V_{DS} - V_{BS} \qquad \text{Intrinsic drain-to-bulk voltage}$$

For P-channel devices, all signs of the above voltages are inverted prior to processing.

Effective Channel Length and Width

$$W_{eff} = W + \text{DW}$$

$$L_{eff} = L + \text{DL}$$

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:

$$\text{VTO}_a = \text{VTO} + \frac{\text{AVTO}}{\sqrt{\text{NP} \cdot W_{eff} \cdot \text{NS} \cdot L_{eff}}}$$

$$KP_a = KP \cdot \left(1 + \frac{AKP}{\sqrt{NP \cdot W_{eff} \cdot NS \cdot L_{eff}}}\right)$$

$$GAMMA_a = GAMMA + \frac{AGAMMA}{\sqrt{NP \cdot W_{eff} \cdot NS \cdot L_{eff}}}$$

These model equations are only applicable in Monte-Carlo and sensitivity simulations.

Note that since negative values for both KP_a and $GAMMA_a$ are not physically meaningful, these are clipped at zero.

Reverse Short-channel Effect (RSCE)

$$C_\varepsilon = 4 \cdot (22 \times 10^{-3})^2 \qquad C_A = 0.028$$

$$\xi = C_A \cdot \left(10 \cdot \frac{L_{eff}}{LK} - 1\right)$$

$$\Delta V_{RSCE} = \frac{2 \cdot Q0}{COX} \cdot \frac{1}{\left[1 + \frac{1}{2} \cdot (\xi + \sqrt{\xi^2 + C_\varepsilon})\right]^2}$$

Effective Gate Voltage Including RSCE

$$V_G' = V_G - VTO_a - \Delta V_{RSCE} + PHI + GAMMA_a\sqrt{PHI}$$

Effective substrate factor including charge-sharing for short and narrow channels

Pinch-off voltage for narrow-channel effect:

$$V_{P0} = \begin{cases} V_G' - PHI - GAMMA_a\left(\sqrt{V_G' + \left(\frac{GAMMA_a}{2}\right)^2} - \frac{GAMMA_a}{2}\right) & \text{for: } V_G' > 0 \\ -PHI & \text{for: } V_G' \leq 0 \end{cases}$$

Effective substrate factor accounting for charge-sharing:

$$V'_{S(D)} = \frac{1}{2} \cdot [V_{S(D)} + \mathrm{PHI} + \sqrt{(V_{S(D)} + \mathrm{PHI})^2 + (4V_t)^2}]$$

Note: Equation prevents the argument of the square roots in the subsequent code from becoming negative.

$$\gamma^\circ = \mathrm{GAMMA}_a - \frac{\varepsilon_{si}}{\mathrm{COX}} \cdot \left[\frac{\mathrm{LETA}}{L_{eff}} \cdot (\sqrt{V'_S} + \sqrt{V'_D}) - \frac{3 \cdot \mathrm{WETA}}{W_{eff}} \cdot \sqrt{V_{P0} + \mathrm{PHI}} \right]$$

$$\gamma' = \frac{1}{2} \cdot (\gamma^\circ + \sqrt{\gamma^{\circ 2} + 0.1 \cdot V_t})$$

Note: The purpose of Equation is to prevent the effective substrate factor from becoming negative.

Pinch-off Voltage Including Short- and Narrow-channel Effects

$$V_P = \begin{cases} V_G' - \mathrm{PHI} - \gamma' \cdot \left(\sqrt{V_G' + \left(\frac{\gamma'}{2}\right)^2} - \frac{\gamma'}{2} \right) & \text{for:}\quad V_G' > 0 \\ -\mathrm{PHI} & \text{for:}\quad V_G' \le 0 \end{cases}$$

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

$$n = 1 + \frac{\mathrm{GAMMA}_a}{2 \cdot \sqrt{V_P + \mathrm{PHI} + 4V_t}}$$

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:

$$y = \sqrt{0.25 + i} - 0.5$$

$$v = 2y + \ln(y)$$

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

$$V_C = \text{UCRIT} \cdot \text{NS} \cdot L_{eff}$$

Note: *Equation accounts for multiple series device number* NS.

$$V_{DSS} = V_C \cdot \left[\sqrt{\frac{1}{4} + \frac{V_t}{V_C} \cdot \sqrt{i_f}} - \frac{1}{2} \right]$$

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

$$V_{DSS}' = V_C \cdot \left[\sqrt{\frac{1}{4} + \frac{V_t}{V_C} \cdot \left(\sqrt{i_f} - \frac{3}{4} \cdot \ln(i_f) \right)} - \frac{1}{2} \right] + V_t \cdot \left[\ln\left(\frac{V_C}{2V_t} \right) - 0.6 \right]$$

Channel-length Modulation

$$\Delta V = 4 \cdot V_t \cdot \sqrt{\text{LAMBDA} \cdot \left(\sqrt{i_f} - \frac{V_{DSS}}{V_t} \right) + \frac{1}{64}}$$

$$V_{ds} = \frac{V_D - V_S}{2}$$

$$V_{ip} = \sqrt{V_{DSS}^2 + \Delta V^2} - \sqrt{(V_{ds} - V_{DSS})^2 + \Delta V^2}$$

$$L_C = \sqrt{\frac{\varepsilon_{si}}{\text{COX}} \cdot \text{XJ}}$$

$$\Delta L = \text{LAMBDA} \cdot L_C \cdot \ln\left(1 + \frac{V_{ds} - V_{ip}}{L_C \cdot \text{UCRIT}} \right)$$

Equivalent Channel Length Including Channel-length Modulation and Velocity Saturation

$$L' \;=\; \text{NS} \cdot L_{eff} - \Delta L + \frac{V_{ds} + V_{ip}}{\text{UCRIT}}$$

$$L_{min} \;=\; \text{NS} \cdot L_{eff} / 10$$

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:

$$i_r' = F\left[\frac{V_P - V_{ds} - V_S - \sqrt{V_{D\acute{S}S}^2 + \Delta V^2} + \sqrt{(V_{ds} - V_{D\acute{S}S})^2 + \Delta V^2}}{V_t}\right]$$

Reverse normalized current for mobility model, intrinsic charges/capacitances, thermal noise model and NQS time-constant:

$$i_r \;=\; F\left[\frac{V_P - V_D}{V_t}\right]$$

Transconductance Factor and Mobility Reduction Due to Vertical Field

$$\beta_0 = \text{KP}_a \cdot \frac{\text{NP} \cdot W_{eff}}{L_{eq}}$$

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.

$$\eta = \begin{cases} 1/2 & for & NMOS \\ 1/3 & for & PMOS \end{cases}$$

$$q_{B0} = \text{GAMMA}_a \cdot \sqrt{\text{PHI}}$$

$$\beta_0' = \beta_0 \cdot \left(1 + \frac{\text{COX}}{\text{E0} \cdot \varepsilon_{si}} \cdot q_{B0}\right)$$

$$\beta = \frac{\beta_0'}{1 + \frac{\text{COX}}{\text{E0} \cdot \varepsilon_{si}} \cdot V_t \cdot |q_B + \eta \cdot q_I|}$$

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 β_0' ensures that $\beta \approx \beta_0$ when $q_I \ll q_B$. The formulation of β 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 E0 is *not* specified (see parameter preprocessing), the simpler mobility model is taken into account according to:

$$V_P' = \frac{1}{2} \cdot (V_P + \sqrt{V_P^2 + 2V_t^2})$$

$$\beta = \frac{\beta_0}{1 + \text{THETA} \cdot V_P'}$$

Specific Current

$$I_S = 2 \cdot n \cdot \beta \cdot V_t^2$$

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:

$$g_{mg} \equiv \frac{\partial I_{DS}}{\partial V_G} \qquad g_{ms} \equiv -\frac{\partial I_{DS}}{\partial V_S} \qquad g_{md} \equiv \frac{\partial I_{DS}}{\partial V_{DS}}$$

Note the following relationships with the derivatives where the source is taken as reference:

$$g_m \equiv \frac{\partial I_{DS}}{\partial V_{GS}} = g_{mg} \qquad g_{mbs} \equiv \frac{\partial I_{DS}}{\partial V_{BS}} = g_{ms} - g_{mg} - g_{md} \qquad g_{ds} \equiv \frac{\partial I_{DS}}{\partial V_{DS}} = g_{md}$$

The analytic derivatives are available with the model code.

Impact Ionization Current

$$V_{ib} = V_D - V_S - \text{IBN} \cdot 2 \cdot V_{DSS}$$

$$I_{DB} = \begin{cases} I_{DS} \cdot \dfrac{\text{IBA}}{\text{IBB}} \cdot V_{ib} \cdot \exp\left(\dfrac{-\text{IBB} \cdot L_C}{V_{ib}}\right) & \text{for: } V_{ib} > 0 \\ 0 & \text{for: } V_{ib} \leq 0 \end{cases}$$

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

$$n_q = 1 + \frac{\text{GAMMA}_a}{2 \cdot \sqrt{V_P + \text{PHI} + 10^{-6}}}$$

Normalized Intrinsic Node Charges:

$$x_f = \sqrt{\frac{1}{4} + i_f}$$

$$x_r = \sqrt{\frac{1}{4} + i_r}$$

$$q_D = -n_q \cdot \left(\frac{4}{15} \cdot \frac{3x_r^3 + 6x_r^2 x_f + 4x_r x_f^2 + 2x_f^3}{(x_f + x_r)^2} - \frac{1}{2} \right)$$

$$q_S = -n_q \cdot \left(\frac{4}{15} \cdot \frac{3x_f^3 + 6x_f^2 x_r + 4x_f x_r^2 + 2x_r^3}{(x_f + x_r)^2} - \frac{1}{2} \right)$$

$$q_I = q_S + q_D = -n_q \cdot \left(\frac{4}{3} \cdot \frac{x_f^2 + x_f x_r + x_r^2}{x_f + x_r} - 1 \right)$$

$$q_B = \begin{cases} (-\text{GAMMA}_a \cdot \sqrt{V_P + \text{PHI} + 10^{-6}}) \cdot \frac{1}{V_t} - \left(\frac{n_q - 1}{n_q} \right) \cdot q_I & \text{for:} V'_G > 0 \\ \\ -V'_G \cdot \frac{1}{V_t} & \text{for:} V'_G \leq 0 \end{cases}$$

$$q_G = -q_I - q_{OX} - q_B$$

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:

$$C_{ox} = \text{COX} \cdot \text{NP} \cdot W_{eff} \cdot \text{NS} \cdot L_{eff}$$

$$Q_{(I, B, D, S, G)} = C_{ox} \cdot V_t \cdot q_{(I, B, D, S, G)}$$

Intrinsic Capacitances

Transcapacitances

The intrinsic capacitances are obtained through derivation of the node charges with respect to the terminal voltages:

$$C_{xy} = \pm \frac{\partial}{\partial V_y}(Q_x) \qquad x, y = G, D, S, B$$

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 22-86, can be obtained when neglecting the slight bias dependence of the slope factor n, resulting in the following simple functions:

$$c_{gs} = \frac{2}{3} \cdot \left(1 - \frac{x_r^2 + x_r + \frac{1}{2}x_f}{(x_f + x_r)^2} \right)$$

$$c_{gd} = \frac{2}{3} \cdot \left(1 - \frac{x_f^2 + x_f + \frac{1}{2}x_r}{(x_f + x_r)^2} \right)$$

$$c_{gb} = \left(\frac{n_q - 1}{n_q}\right) \cdot (1 - c_{gs} - c_{gd})$$

$$c_{sb} = (n_q - 1) \cdot c_{gs}$$

$$c_{db} = (n_q - 1) \cdot c_{gd}$$

Note that this simplified capacitances model can be chosen by setting XQC=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 τ is the bias dependent characteristic time constant.

τ_0 is the intrinsic time constant defined as:

$$\tau_0 = \frac{(\text{NS} \cdot L_{eff})^2}{2 \cdot \mu_{eff} \cdot V_t} = \frac{C_{ox}}{2 \cdot V_t \cdot \beta}$$

$$\tau = \tau_0 \cdot \frac{4}{15} \cdot \frac{(x_f^2 + 3x_f x_r + x_r^2)}{(x_f + x_r)^3}$$

$$I_{DS}(s) = \frac{I_{DS}}{1 + \text{NQS} \cdot s \cdot \tau}$$

The corresponding small-signal (.AC) transadmittances are then given by:

$$Y_{mg}(s) = \frac{g_{mg}}{1 + \text{NQS} \cdot s \cdot \tau}$$

$$Y_{ms}(s) = \frac{g_{ms}}{1 + \mathrm{NQS} \cdot s \cdot \tau}$$

$$Y_{md}(s) = \frac{g_{md}}{1 + \mathrm{NQS} \cdot s \cdot \tau}$$

$$Y_{mbs}(s) = Y_{ms}(s) - Y_{mg}(s) - Y_{md}(s)$$

Note that the availability of the NQS model is simulator-dependent.

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:

$$S_{thermal} = 4kT \cdot \frac{\mu_{eff}}{(\mathrm{NS} \cdot L_{eff})^2} \cdot |Q_I| = 4kT \cdot \beta \cdot |q_I|$$

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:

$$S_{flicker} = \frac{\mathrm{KF} \cdot g_{mg}^2}{\mathrm{NP} \cdot W_{eff} \cdot \mathrm{NS} \cdot L_{eff} \cdot \mathrm{COX} \cdot f^{\mathrm{AF}}}$$

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_{mbs}, g_{md}, V_P, n, β, I_S, i_f, i_r, i_r', τ, τ_0

intrinsic charges/capacitances

Transconductance efficiency factor:

$tef = g_{ms} \cdot V_t / I_{DS}$

'Early voltage':

$VM = I_{DS} / g_{md}$

'Overdrive voltage':

$n \cdot (V_P - V_S) \approx V_G - \text{VTO}_a - n \cdot V_S$

For P-channel devices, $n \cdot (V_P - V_S)$ is given a negative sign.

'SPICE-like' threshold voltage:

$VTH = \text{VTO}_a + \Delta V_{RSCE} + \gamma' \cdot \sqrt{V'_S} - \text{GAMMA}_a \cdot \sqrt{\text{PHI}}$

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:

$VDSAT = 2V_{DSS} + 4V_t$

For P-channel devices, *VDSAT* is given a negative sign.

Saturation / non-saturation flag:

$$'SAT' \quad or \quad '1' \qquad for \quad \frac{i_f}{i_r} > \texttt{SATLIM}$$

$$'LIN' \quad or \quad '0' \qquad for \quad \frac{i_f}{i_r} \leq \texttt{SATLIM}$$

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.

Name	Unit	Default	Example	'Lower'	'Upper'	Estimation[a]
COX	F/m^2	0.7E-3	3.45E-3	-	-	$\varepsilon_{ox}/\texttt{TOX}$
XJ	m	0.1E-6	0.15E-6	0.01E-6	1E-6	XJ
VTO	V	0.5	0.7	0	2	VTO
GAMMA	$\sqrt{V}$	1.0	0.7	0	2	$\sqrt{2q\varepsilon_{si} \cdot \texttt{NSUB}}/\texttt{COX}$

Name	Unit	Default	Example	'Lower'	'Upper'	Estimation[a]
PHI[b]	V	0.7	0.5	0.3	2	$2V_t \cdot \ln(\text{NSUB}/n_i)$
KP	A/V^2	50E-6	150E-6	10E-6	-	$\text{UO} \cdot \text{COX}$
E0	V/m	1.0E12	200E6	$0.1/(0.4 \cdot \text{TOX}$	-	$0.2/(\text{THETA} \cdot \text{TOX})$
UCRIT	V/m	2.0E6	2.3E6	1.0E6	25E6	VMAX/UO
DL	m	0	-0.15*L$_{min}$	-0.5*L$_{min}$	0.5*L$_{min}$	$\text{XL} - 2 \cdot \text{LD}$
DW	m	0	-0.1*W$_{min}$	-0.5*W$_{min}$	0.5*W$_{min}$	$\text{XW} - 2 \cdot \text{WD}$
LAMBDA	-	0.5	0.8	0	3	-
LETA	-	0.1	0.3	0	2	-
WETA	-	0.25	0.2	0	2	-
Q0	As/m^2	0.0	230E-6	0	-	-
LK	m	0.29E-6	0.4E-6	0.05E-6	2E-6	-
IBA	1/m	0.0	2.0E8	0.0	5.0E8	$\text{ALPHA} \cdot \text{VCR}/L_C$
IBB	V/m	3.0E8	2.0E8	1.8E8	4.0E8	VCR/L_C
IBN	-	1.0	0.6	0.4	1.0	-

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.

$\varepsilon_{ox} = 0.0345\text{E-9}$ F/m $q = 1.609\text{E-19}$ C $k = 1.381\text{E-23}$ J/K $L_c = \sqrt{\varepsilon_{si} \cdot \text{XJ}/\text{COX}}$

$\varepsilon_{si} = 0.104\text{E-9}$ F/m $n_i = 1.45\text{E16}$ m^{-3} $V_t = kT/q = 0.0259$ V (at room temperature)

Note: *Parameters in this table suppose m (meter) has been chosen as length unit. L$_{min}$ and W$_{min}$ 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 τ_0 is calculated as a function of the effective β 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 β 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 COX, GAMMA and/or PHI, VTO, KP and UCRIT as a function of the optional parameters TOX, NSUB, VFB, UO, and VMAX, respectively. NSUB and UO have *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 THETA, is reinstated as an option.

Consequence: Simplifies adaptation from earlier model versions to the current version.

Parameter synonyms

Description: The parameters E0 and Q0 can be called by their synonyms EO and 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 VTH in the operating point information is modified to include charge-sharing and reverse short-channel effects. The analytical expression for the saturation voltage $VDSAT$ 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 NS.

Corrections from EPFL R12, July 30, 1999

The following corrections were released by EPFL.

Correction 1- 99/07/30 mb (r12) corrected dGAMMAprime_dVG (narrow channel). An error has been detected in the analytical model derivatives. It is in the derivatives of GAMMAprime 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

> Mxxx nd ng ns ne <np> <nb> <nT> 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>
> + <NBC=val> <NSEG=val> <PDBCP=val> <PSBCP=val>
> + <AGBCP=val> <AEBCP=val> <VBSUSR=val> <TNODEOUT>
> + <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
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.
RDC	Additional drain resistance due to contact resistance with units of ohms. This value overrides the RDC setting in the model specification. Default=0.0.
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.

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

Parameter	Unit	Default	Description
LEVEL	-	-	LEVEL 57 for BSIM3SOI
SHMOD	-	0	Flag for self-heating: 0 - no self-heating 1 - self-heating
MOBMOD	-	1	Mobility model selector
CAPMOD	-	2	Flag for the short channel capacitance model
NOIMOD	-	1	Flag for noise model

Process Parameters

Parameter	Unit	Default	Description
TSI	m	1.0e-7	Silicon film thickness
TBOX	m	3.0e-7	Buried oxide thickness
TOX	m	1.0e-8	Gate oxide thickness
XJ	m	-	S/D junction depth
NCH	1/cm^3	1.7e17	Channel doping concentration
NSUB	1/cm^3	6.0e16	Substrate doping concentration
NGATE	1/cm^3	0	Poly gate doping concentration

DC Parameters

Parameter	Unit	Default	Description
VTH0	v	NMOS 0.7 PMOS -0.7	Threshold voltage @ Vbs=0 for long wide device
K1	V$^{1/2}$	0.6	First-order body effect coefficient
K1W1	m	0	First-order effect width dependent parameter
K1W2	m	0	Second-order effect width dependent parameter
K2	-	0	Second-order body effect coefficient
K3	-	0	Narrow coefficient
K3B	1/V	0	Body effect coefficient of k3
KB1	-	1	Backgate body charge coefficient
W0	m	0	Narrow width parameter
NLX	m	1.74e-7	Lateral non-uniform doping parameter

Parameter	Unit	Default	Description
DVT0	-	2.2	First coefficient of short-channel effect on Vth
DVT1	-	0.53	Second coefficient of short-channel effect on Vth
DVT2	1/V	-0.032	Body-bias coefficient of short-channel effect on Vth
DVT0W	-	0	First coefficient of narrow width effect on Vth for small channel length
DVT1W	-	5.3e6	Second coefficient of narrow width effect on Vth for small channel length
DVT2W	1/V	-0.032	Body-bias coefficient of narrow width effect on Vth for small channel length
U0	cm^2/(V-sec)	NMOS-670 PMOS-250	Mobility at Temp=Tnom
UA	m/V	2.25e-9	First-order mobility degradation coefficient
UB	$(m/V)^2$	5.87e-19	Second-order mobility degradation coefficient
UC	1/V	-0.0465	Body-effect of mobility degradation coefficient
VSAT	m/sec	8e4	Saturation velocity at Temp=Tnom
A0	-	1.0	Bulk charge effect coefficient for channel length
AGS	1/V	0.0	Gate bias coefficient of A_{bulk}
B0	m	0.0	Bulk charge effect coefficient for channel width
B1	m	0.0	Bulk charge effect width offset

Parameter	Unit	Default	Description
KETA	1/V	-0.6	Body-bias coefficient of bulk charge effect
KETAS	V	0.0	Surface potential adjustment for bulk charge effect
A1	1/V	0.0	First non-saturation effect parameter
A2	-	1.0	Second non-saturation effect parameter
RDSW	$\Omega * \mu m^{wr}$	100	Parasitic resistance per unit width
PRWB	$1/V^1$	0	Body effect coefficient of Rdsw
PRWG	$1/V^{1/2}$	0	Gate bias effect coefficient of Rdsw
WR	-	1	Width offset from Weff for Rds calculation
NFACTOR	-	1	Subthreshold swing factor
WINT	m	0.0	Width offset fitting parameter from I-V without bias
LINT	m	0.0	Length offset fitting parameter from I-V without bias
DWG	m/V	0.0	Coefficient of Weff's gate dependence
DWB	$m/V^{1/2}$	0.0	Coefficient of Weff's substrate body bias dependence
DWBC	m	0.0	Width offset for body contact isolation edge
VOFF	v	-0.08	Offset voltage in the subthreshold region for large W and L
ETA0	-	0.08	DIBL coefficient in the subthreshold region

Parameter	Unit	Default	Description
ETAB	1/V	-0.07	Body-bias coefficient for the subthreshold DIBL effect
DSUB	-	0.56	DIBL coefficient exponent
CIT	F/m^2	0.0	Interface trap capacitance
CDSC	F/m^2	2.4e-4	Drain/source to channel coupling capacitance
CDSCB	F/m^2	0	Body-bias sensitivity of cdsc
CDSCD	F/m^2	0	Drain-bias sensitivity of cdsc
PCLM	-	1.3	Channel length modulation parameter
PDIBLC1	-	0.39	First output resistance DIBL effect correction parameter
PDIBLC2	-	0.0086	Second output resistance DIBL effect correction parameter
DROUT	-	0.56	L dependence coefficient of the DIBL correction parameter in Rout
PVAG	-	0.0	Gate dependence of Early voltage
DELTA	-	0.01	Effective V_{ds} parameter
ALPHA0	m/V	0.0	The first parameter of impact ionization current
FBJTII	-	0.0	Fraction of bipolar current affecting the impact ionization
BETA0	1/V	0.0	First V_{ds} dependence parameter of impact ionization current
BETA1	-	0.0	Second V_{ds} dependence parameter of impact ionization current
BETA2	V	0.1	Third V_{ds} dependence parameter of impact ionization current

Parameter	Unit	Default	Description
VDSATII0	V	0.9	Nominal drain saturation voltage at threshold for impact ionization current
TII	-	0	Temperature dependence parameter for impact ionization current
LII	-	0	Channel length dependence parameter for impact ionization current
ESATI	V/m	1.e7	Saturation channel electric field for impact ionization current
SII0	1/V	0.5	First V_{gs} dependence parameter for impact ionization current
SII1	1/V	0.1	Second V_{gs} dependence parameter for impact ionization current
SII2	1/V	0	Third V_{gs} dependence parameter for impact ionization current
SIID	1/V	0	V_{ds} dependence parameter of drain saturation voltage for impact ionization current
AGIDL	1/W	0.0	GIDL constant
BGIDL	V/m	0.0	GIDL exponential coefficient
NGIDL	V	1.2	GIDL V_{ds} enhancement coefficient
NTUN	-	10.0	Reverse tunneling non-ideality factor
NDIODE	-	1.0	Diode non-ideality factor
NRECF0	-	2.0	Recombination non-ideality factor at forward bias
NRECR0	-	10	Recombination non-ideality factor at reversed bias
ISBJT	A/m^2	1.0e-6	BJT injection saturation current

Parameter	Unit	Default	Description
ISDIF	A/m^2	0	Body to source/drain injection saturation current
ISREC	A/m^2	1.0e-5	Recombination in depletion saturation current
ISTUN	A/m^2	0.0	Reverse tunneling saturation current
LN	m	2.0e-6	Electron/hole diffusion length
VREC0	V	0.0	Voltage dependent parameter for recombination current
VTUN0	V	0.0	Voltage dependent parameter for tunneling current
NBJT	-	1	Power coefficient of channel length dependency for bipolar current
LBJT0	m	0.2e-6	Reference channel length for bipolar current
VABJT	V	10	Early voltage for bipolar current
AELY	V/m	0	Channel length dependency of early voltage for bipolar current
AHLI	-	0	High-level injection parameter for bipolar current
RBODY	ohm/m^2	0.0	Intrinsic body contact sheet resistance
RBSH	ohm/m^2	0.0	Extrinsic body contact sheet resistance
RSH	ohm/square	0.0	Source/drain sheet resistance in ohm per square

AC and Capacitance Parameters

Parameter	Unit	Default	Description
XPART	-	0	Charge partitioning rate flag
CGSO	F/m	calculated	Non LDD region source-gate overlap capacitance per channel length
CGDO	F/m	0	Non LDD region drain-gate overlap capacitance per channel length
CGEO	F/m	0	Gate substrate overlap capacitance per unit channel length
CJSWG	F/m^2	1.e-10	Source/drain (gate side) sidewall junction capacitance per unit width (normalized to 100nm Tsi)
PBSWG	V	0.7	Source/drain (gate side) sidewall junction capacitance built in potential
MJSWG	V	0.5	Source/drain (gate side) sidewall junction capacitance grading coefficient
TT	second	1ps	Diffusion capacitance transit time coefficient
NDIF	-	-1	Power coefficient of channel length dependency for diffusion capacitance
LDIF0	-	1	Channel length dependency coefficient of diffusion cap.
VSDFB	V	cal.	Source/drain bottom diffusion capacitance flatband voltage
VSDTH	V	cal.	Source/drain bottom diffusion capacitance threshold voltage
CSDMIN	V	cal.	Source/drain bottom diffusion minimum capacitance

Parameter	Unit	Default	Description
ASD	V	0.3	Source/drain bottom diffusion smoothing parameter
CSDESW	F/m	0.0	Source/drain sidewall fringing capacitance per unit length
CGSL	F/m	0.0	Lightly doped source-gate region overlap capacitance
CGDL	F/m	0.0	Lightly doped drain-gate region overlap capacitance
CKAPPA	F/m	0.6	Coefficient for lightly doped region overlap capacitance fringing field capacitance
CF	F/m	cal.	Gate to source/drain fringing field capacitance
CLC	m	0.1e-7	Constant term for the short channel model
CLE	-	0.0	Exponential term for the short channel model
DLC	m	lint	Length offset fitting parameter for gate charge
DLCB	m	lint	Length offset fitting parameter for body charge
DLBG	m	0	Length offset fitting parameter for backgate charge
DWC	m	wint	Width offset fitting parameter from C-V
DELVT	V	0.0	Threshold voltage adjust for C-V
FBODY	-	1.0	Scaling factor for body charge

Parameter	Unit	Default	Description
ACDE	m/V	1.0	Exponential coefficient for charge thickness in CapMod=3 for accumulation and depletion regions
MOIN	$V^{1/2}$	15.0	Coefficient for the gate-bias dependent surface potential

Temperature Parameters

Parameter	Unit	Default	Description
TNOM	^{o}C	25	Temperature at which parameters are expected
UTE	-	-1.5	Mobility temperature exponent
KT1	V	-0.11	Temperature coefficient for the threshold voltage
KTIL	V*m	0	Channel length dependence of the temperature coefficient for the threshold voltage
KT2	-	0.022	Body-bias coefficient of the threshold voltage temperature effect
UA1	m/V	4.31e-9	Temperature coefficient for U_a
UB1	$(m/V)^2$	-7.61e-18	Temperature coefficient for U_b
UC1	1/V	-0.056	Temperature coefficient for U_c
AT	m/sec	3.3e4	Temperature coefficient for U_a
TCJSWG	1/K	0	Temperature coefficient of C_{jswg}
TPBSWG	V/K	0	Temperature coefficient of P_{bswg}
CTH0	m^{o}C/(W*s)	0	Normalized thermal capacity
PRT	Ω-um	0	Temperature coefficient for Rdsw

Parameter	Unit	Default	Description
RTH0	m^{o}C/W	0	Normalized thermal resistance
NTRECF	-	0	Temperature coefficient for N_{recf}
NTRECR	-	0	Temperature coefficient for N_{recr}
XBJT	-	1	Power dependence of j_{bjt} on temperature
XDIF	-	XBJT	Power dependence of j_{dif} on temperature
XREC	-	1	Power dependence of j_{rec} on temperature
XTUN	-	0	Power dependence of j_{tun} on temperature

Notes:

- BSIMPD2.01 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.

- BSIMPD refers substrate to the silicon below buried oxide, not the well region in BSIM3. It is used to calculated backgate flatband voltage (Vfbb) and parameters related to source/drain diffusion bottom capacitance (Vsdth, Vsdfb, Csdmin). Positive nsub means the same type of doping as the body and negative nsub means opposite type of doping.

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)

Name	Alias	Description
L	LV1	Channel length (L)
W	LV2	Channel width (W)
AD	LV3	Area of the drain diode (AD)
AS	LV4	Area of the source diode (AS)
ICVDS	LV5	Initial condition for drain-source voltage (VDS)
ICVGS	LV6	Initial condition for gate-source voltage (VGS)
ICVES	LV7	Initial condition for Substrate-source voltage (VES)
VTH	LV9	Threshold voltage (bias dependent)
VDSAT	LV10	Saturation voltage (VDSAT)
PD	LV11	Drain diode periphery (PD)
PS	LV12	Source diode periphery (PS)
RDS	LV13	Drain resistance (squares) (RDS)
RSS	LV14	Source resistance (squares) (RSS)
GDEFF	LV16	Effective drain conductance (1/RDeff)
GSEFF	LV17	Effective source conductance (1/RSeff)
COVLGS	LV36	Gate-source overlap capacitance
COVLGD	LV37	Gate-drain overlap capacitance
COVLGE	LV38	Gate-substrate overlap capacitance
VES	LX1	Substrate-source voltage (VES)
VGS	LX2	Gate-source voltage (VGS)
VDS	LX3	Drain-source voltage (VDS)
CDO	LX4	DC drain current (CDO)

Name	Alias	Description
CBSO	LX5	DC source-body diode current (CBSO)
CBDO	LX6	DC drain-body diode current (CBDO)
GMO	LX7	DC gate transconductance (GMO)
GDSO	LX8	DC drain-source conductance (GDSO)
GMESO	LX9	DC substrate transconductance (GMBSO)
GBDO	LX10	Conductance of the drain diode (GBDO)
GBSO	LX11	Conductance of the source diode (GBSO)

Meyer and Charge Conservation Model Parameters

Name	Alias	Description
QB	LX12	Body charge (QB)
CQB	LX13	Body charge current (CQB)
QG	LX14	Gate charge (QG)
CQG	LX15	Gate charge current (CQG)
QD	LX16	Channel charge (QD)
CQD	LX17	Channel charge current (CQD)
CGGBO	LX18	$GGBO = \partial Qg/\partial Vgb$ = CGS + CGD + CGB
CGDBO	LX19	$CGDBO = \partial Qg/\partial Vdb$, (for Meyer CGD=-CGDBO)
CGSBO	LX20	$CGSBO = \partial Qg/\partial Vsb$, (for Meyer CGS=-CGSBO)
CBGBO	LX21	$CBGBO = \partial Qb/\partial Vgb$, (for Meyer CGB=-CBGBO)
CBDBO	LX22	$CBDBO = \partial Qb/\partial Vdb$

Name	Alias	Description
CBSBO	LX23	$CBSBO = \partial Qb/\partial Vsb$
CDGBO	LX32	$CDGBO = \partial Qd/\partial Vgb$
CDDBO	LX33	$CDDBO = \partial Qd/\partial Vdb$
CDSBO	LX34	$CDSBO = \partial Qd/\partial Vsb$
QE	LX35	Substrate charge (QE)
CQE	LX36	Substrate charge current (CQE)
CDEBO	LX37	$CDEBO = \partial Qd/\partial Veb$
CBEBO	LX38	$BEBO = \partial Qb/\partial Veb$
CEEBO	LX39	$EEBO = \partial Qe/\partial Veb$
CEGBO	LX40	$CEGBO = \partial Qe/\partial Vgb$
CEDBO	LX41	$EDBO = \partial Qe/\partial Vdb$
CESBO	LX42	$ESBO = \partial Qe/\partial Vsb$
VBS	LX43	Body-source voltage (VBS)
ICH	LX44	Channel current
IBJT	LX45	Parasitic BJT collector current
III	LX46	Impact ionization current
IGIDL	LX47	GIDL current
ITUN	LX48	Tunneling current

LEVEL 57 Updates to BSIM3-SOI PD versions 2.2 and 2.21

- 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.

Using BSIM3-SOI PD in Star-HSPICE

You can use BSIM3-SOI PD versions 2.0, 2.2, and 2.21 in Star-HSPICE by applying the model parameter VERSION. To use the PD model version equal to or greater than 2.2, the model parameter should be set to VERSION=2.2. For example:

- PD2.0 is invoked when VERSION=2.0
- PD2.2 and PD2.21 are invoked when VERSION=2.2

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.

```
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         k1=.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=1.1
+ 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         xtun=0.01        ahli=1e-9
+ lbjt0=0.2e-6    ln=2e-6         nbjt=.8          ndif=-1
+ aely=1e8        vabjt=0         u0=352           ua=1.3e-11
+ ub=1.7e-18      uc=-4e-10       w0=1.16e-06      ags=.25
+ A1=0            A2=1            b0=.01           b1=10
+ rdsw=0          prwg=0          prwb=-.2         wr=1
+ rbody=1E0       rbsh=0.0        a0=1.4           keta=0.1
+ ketas=0.2       vsat=135000     dwg=0            dwb=0
+ alpha0=1e-8     beta0=0         beta1=0.05       beta2=0.07
+ vdsatii0=.8     esatii=1e7      voff=-.14        nfactor=.7
+ cdsc=.00002     cdscb=0         cdscd=0          cit=0
+ pclm=2.9        pvag=12         pdiblc1=.18      pdiblc2=.004
+ pdiblcb=-.234   drout=.2        delta=.01        eta0=.05
+ etab=0          dsub=.2         rth0=.005        clc=.0000001
+ cle=.6          cf=1e-20        ckappa=.6        cgdl=1e-20
+ cgsl=1e-20      kt1=-.3         kt1l=0           kt2=.022
+ ute=-1.5        ua1=4.31e-09    ub1=-7.61e-18    uc1=-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 $K \cdot W^{-1}$, defaulted to 0.0
CTH	Thermal capacitance, unit in $W \cdot s \cdot K^{-1}$, 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

Parameter	Unit	Default	Typical Value	Description
LEVEL	-	-	-	LEVEL 57 for UFSOI
NFDMOD	-	0	0	Model selector (0: FD)
BJT	-	1	1	Parasitic bipolar flag (0: off; 1: on)
SELFT	-	0	0	Self-heating flag (0: no self-heating; 1: approximate model; 2: full self-heating)
TPG	-	1	-	Type of gate polysilicon (+1: opposite to body; -1: same as body)
TPS	-	-1	-	Type of substrate (+1: opposite to body; -1: same as body)

Structural Parameters

Parameter	Unit	Default	Typical Value	Description
TOXF	m	1.0e-8	$(3-8)\times10^{-9}$	Front-gate oxide thickness
TOXB	m	0.5e-6	$(80-400)$ $\times10^{-9}$	Back-gate oxide thickness
NSUB	cm^{-3}	1.0e15	$10^{15}-10^{17}$	Substrate doping density
NGATE	cm^{-3}	0.0	$10^{19}-10^{20}$	Poly-gate doping density (0 for no poly-gate depletion)
NDS	cm^{-3}	5.0e19	$10^{19}-10^{20}$	Source/drain doping density
TB	m	0.1e-6	$(30-100)$ $\times10^{-9}$	Film (body) thickness
NBODY	cm^{-3}	5.0e16	$10^{17}-10^{18}$	Film (body) doping density
LLDD	m	0.0	$(0.05-0.2)$ $\times10^{-6}$	LDD/LDS region length (0 for no LDD)
NLDD	cm^{-3}	5.0e19	1×10^{19}	LDD/LDS doping density (>1e19: LDD/LDS treated as D/S extensions)
DL	m	0.0	$(0.05-0.15)\times10^{-6}$	Channel-length reduction
DW	m	0.0	$(0.1-0.5)$ $\times10^{-6}$	Channel-width reduction

Electrical Parameters

Parameter	Unit	Default	Typical Value	Description
NQFF	cm^{-2}	0.0	$\sim 10^{10}$	Front oxide fixed charge (normalized)
NQFB	cm^{-2}	0.0	$\sim 10^{11}$	Back oxide fixed charge (normalized)
NQFSW	cm^{-2}	0.0	$\sim \pm 10^{12}$	Effective sidewall fixed charge (0 for no narrow-width effect)
NSF	$cm^{-2} \cdot eV^{-1}$	0.0	$\sim 10^{10}$	Front surface state density
NSB	$cm^{-2} \cdot eV^{-1}$	0.0	$\sim 10^{11}$	Back surface state density
QM	-	0.0 -0.5		Energy quantization parameter (0 for no quantization)
UO	$cm^2 \cdot V^{-1} \cdot s^{-1}$	7.0e2	200-700 (nMOS) 70-400 (pMOS)	Low-field mobility
THETA	$cm \cdot V^{-1}$	7.0e2	$(0.1\text{-}3) \times 10^{-6}$	Mobility degradation coefficient
VSAT	$cm \cdot s^{-1}$	1.0e-6	$(0.5\text{-}1) \times 10$	Carrier saturated drift velocity
ALPHA	cm^{-1}	0.0	2.45×10^6	Impact-ionization coefficient (0 for no impact ionization)
BETA	$V \cdot cm^{-1}$	0.0	1.92×10^6	Impact-ionization exponential factor (0 for no impact ionization)
BGIDL	$V \cdot cm^{-1}$	0.0	$(4\text{-}8) \times 10^9$	Exponential factor for gate-induced drain leakage (0 for no GIDL)
GAMMA	-	0.3	0.3-1.0	BOX fringing field weighting factor

Parameter	Unit	Default	Typical Value	Description
KAPPA	-	0.5	0.5-1.0	BOX fringing field weighting factor
JRO	$A \cdot m^{-1}$	1.0e-10	$10^{-11}-10^{-9}$	Body-source/drain junction recombination current coefficient
M	-	2.0	1.0-2.0	Body-source/drain junction recombination ideality factor
LDIFF	m	1.0e-7	$(0.1-0.5) \times 10^{-6}$	Effective diffusion length in source/drain
SEFF	$cm \cdot s^{-1}$	1.0e5	$(0.5-5) \times 10^5$	Effective recombination velocity in source/drain
CGFDO	$F \cdot m^{-1}$	0.0	1×10^{-10}	Gate-drain overlap capacitance
CGFSO	$F \cdot m^{-1}$	0.0	1×10^{-10}	Gate-source overlap capacitance
CGFBO	$F \cdot m^{-1}$	0.0	0.0	Gate-body overlap capacitance
RD	ohm·m	0.0	200-1000	Specific drain parasitic resistance
RS	ohm·m	0.0	200-1000	Specific source parasitic resistance
RHOB	ohm/sq.	0.0	30×10^3	Body sheet resistance
FNK	F·A	0.0	$0-10^{-2}$	Flicker noise coefficient
FNA	-	1.0	0.5-2	Flicker noise exponent

Optional Parameters

Parameter	Unit	Default	Typical Value	Description
VFBF	V	calc.	-1 (nMOS) 1 (pMOS)	Front-gate flatband voltage
VFBB	V	calc.	-	Back-gate flatband voltage
WKF	V	calc.	~ VFBF	Front-gate work function difference
WKB	V	calc.	-	Back-gate work function difference
TAUO	s	calc.	10^{-7}-10^{-5}	Carrier lifetime in lightly doped regions
BFACT	-	0.3	0.1-0.5	V_{DS}-averaging factor for mobility degradation
FVBJT	-	0.0	0-1	BJT current directional partitioning factor (0 for lateral 1D flow)
RHOSD	ohm/ sq.	0.0	50	Source/drain sheet resistance

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

Parameter	Unit	Default	Typical Value	Description
LEVEL	-	-	-	LEVEL 57 for UFSOI
NFDMOD	-	0	-	Model selector(1: NFD)
BJT	-	1	1	Parasitic bipolar flag (0: off; 1: on)
SELFT	-	0	0	Self-heating flag (0: no self-heating; 1: approximate model; 2: full self-heating)
TPG	-	1	1	Type of gate polysilicon (+1: opposite to body; -1: same as body)
TPS	-	-1	-1	Type of substrate (+1: opposite to body; -1: same as body)

Structural Parameters

Parameter	Unit	Default	Typical Value	Description
TOXF	m	1.0e-8	$(3-8) \times 10^{-9}$	Front-gate oxide thickness
TOXB	m	0.5e-6	$(80-400) \times 10^{-9}$	Back-gate oxide thickness
NSUB	cm^{-3}	1.0e15	10^{15}-10^{17}	Substrate doping density
NGATE	cm^{-3}	0.0	10^{19}-10^{20}	Poly-gate doping density (0 for no poly-gate depletion)
NDS	cm^{-3}	5.0e19	10^{19}-10^{20}	Source/drain doping density
TF	m	0.2e-6	$(3-8) \times 10^{-9}$	Silicon film thickness
TB	m	0.1e-6	$(30-100) \times 10^{-9}$	Film (body) thickness

Parameter	Unit	Default	Typical Value	Description
THALO	m	0.0	-	Halo thickness (0 for no halo)
NBL	cm^{-3}	5.0e16	10^{17}-10^{18}	Low body doping density
NBH	cm^{-3}	5.0e17	10^{19}-10^{20}	Halo doping density
NHALO	cm^{-3}	-	~10^{18}	Halo doping density
LRSCE	m	0.0	~0.1×10^{-6}	Characteristic length for reverse short-channel effect (0 for no RSCE)
LLDD	m	0.0	$(0.05\text{-}0.2) \times 10^{-6}$	LDD/LDS region length (0 for no LDD)
NLDD	cm^{-3}	5.0e19	1×10^{19}	LDD/LDS doping density (>1e19: LDD/LDS treated as D/S extensions)
DL	m	0.0	$(0.05\text{-}0.15) \times 10^{-6}$	Channel-length reduction
DW	m	0.0	$(0.1\text{-}0.5) \times 10^{-6}$	Channel-width reduction

Electrical Parameters

Parameter	Unit	Default	Typical Value	Description
NQFF	cm^{-2}	0.0	~ 10^{10}	Front oxide fixed charge (normalized)
NQFB	cm^{-2}	0.0	~ 10^{11}	Back oxide fixed charge (normalized)
NQFSW	cm^{-2}	0.0	~ $\pm 10^{12}$	Effective sidewall fixed charge (0 for no narrow-width effect)

Parameter	Unit	Default	Typical Value	Description
QM	-	0.0 -0.5		Energy quantization parameter (0 for no quantization)
UO	$cm^2 \cdot V^{-1} \cdot s^{-1}$	7.0e2	200-700 (nMOS) 70-400 (pMOS)	Low-field mobility
THETA	$cm \cdot V^{-1}$	7.0e2	$(0.1-3) \times 10^{-6}$	Mobility degradation coefficient
VSAT	$cm \cdot s^{-1}$	1.0e-6	$(0.5-1) \times 10$	Carrier saturated drift velocity
ALPHA	cm^{-1}	0.0	2.45×10^6	Impact-ionization coefficient (0 for no impact ionization)
BETA	$V \cdot cm^{-1}$	0.0	1.92×10^6	Impact-ionization exponential factor (0 for no impact ionization)
BGIDL	$V \cdot cm^{-1}$	0.0	$(4-8) \times 10^9$	Exponential factor for gate-induced drain leakage (0 for no GIDL)
NTR	cm^{-3}	0.0	10^{14}-10^{15}	Effective trap density for trap-assisted junction tunneling (0 for no tunneling)
JRO	$A \cdot m^{-1}$	1.0e-10	10^{-11}-10^{-9}	Body-source/drain junction recombination current coefficient
M	-	2.0	1.0-2.0	Body-source/drain junction recombination ideality factor
LDIFF	m	1.0e-7	$(0.1-0.5) \times 10^{-6}$	Effective diffusion length in source/drain
SEFF	$cm \cdot s^{-1}$	1.0e5	$(0.5-5) \times 10^5$	Effective recombination velocity in source/drain

Parameter	Unit	Default	Typical Value	Description
CGFDO	$F \cdot m^{-1}$	0.0	1×10^{-10}	Gate-drain overlap capacitance
CGFSO	$F \cdot m^{-1}$	0.0	1×10^{-10}	Gate-source overlap capacitance
CGFBO	$F \cdot m^{-1}$	0.0	0.0	Gate-body overlap capacitance
RD	ohm·m	0.0	200-1000	Specific drain parasitic resistance
RS	ohm·m	0.0	200-1000	Specific source parasitic resistance
RHOB	ohm/sq.	0.0	30×10^3	Body sheet resistance
FNK	$F \cdot A$	0.0	$0\text{-}10^{-2}$	Flicker noise coefficient
FNA	-	1.0	0.5-2	Flicker noise exponent

Optional Parameters

Parameter	Unit	Default	Typical Value	Description
VFBF	V	calc.	-1 (nMOS) 1 (pMOS)	Front-gate flatband voltage
VFBB	V	calc.	-	Back-gate flatband voltage
WKF	V	calc.	~ VFBF	Front-gate work function difference
WKB	V	calc.	-	Back-gate work function difference
TAUO	s	calc.	$10^{-7}\text{-}10^{-5}$	Carrier lifetime in lightly doped regions
BFACT	-	0.3	0.1-0.5	V_{DS}-averaging factor for mobility degradation
FVBJT	-	0.0	0-1	BJT current directional partitioning factor (0 for lateral 1D flow)

Parameter	Unit	Default	Typical Value	Description
RHOSD	ohm/ sq.	0.0	50	Source/drain sheet resistance

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.

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)

Name	Alias	Description
L	LV1	Channel length (L)
W	LV2	Channel width (W)
AD	LV3	Area of the drain diode (AD)
AS	LV4	Area of the source diode (AS)
ICVDS	LV5	Initial condition for drain-source voltage (VDS)
ICVGS	LV6	Initial condition for gate-source voltage (VGS)
ICVES	LV7	Initial condition for Substrate-source voltage (VES)
VTH	LV9	Threshold voltage (bias dependent)
VDSAT	LV10	Saturation voltage (VDSAT)
PD	LV11	Drain diode periphery (PD)
PS	LV12	Source diode periphery (PS)
RDS	LV13	Drain resistance (squares) (RDS)
RSS	LV14	Source resistance (squares) (RSS)
GDEFF	LV16	Effective drain conductance (1/RDeff)
GSEFF	LV17	Effective source conductance (1/RSeff)
VES	LX1	Substrate-source voltage (VES)
VGS	LX2	Gate-source voltage (VGS)
VDS	LX3	Drain-source voltage (VDS)
CDO	LX4	DC drain current (CDO)

Name	Alias	Description
GMO	LX7	DC gate transconductance (GMO)
GDSO	LX8	DC drain-source conductance (GDSO)
GMESO	LX9	DC substrate transconductance (GMBSO)

Meyer and Charge Conservation Model Parameters

Name	Alias	Description
QB	LX12	Body charge (QB)
CQB	LX13	Body charge current (CQB)
QG	LX14	Gate charge (QG)
CQG	LX15	Gate charge current (CQG)
QD	LX16	Channel charge (QD)
CQD	LX17	Channel charge current (CQD)
CGGBO	LX18	$CGGBO = \partial Qg/\partial Vgb$ = CGS + CGD + CGB
CGDBO	LX19	$CGDBO = \partial Qg/\partial Vdb$, (for Meyer CGD=-CGDBO)
CGSBO	LX20	$CGSBO = \partial Qg/\partial Vsb$, (for Meyer CGS=-CGSBO)
CBGBO	LX21	$CBGBO = \partial Qb/\partial Vgb$, (for Meyer CGB=-CBGBO)
CBDBO	LX22	$CBDBO = \partial Qb/\partial Vdb$
CBSBO	LX23	$CBSBO = \partial Qb/\partial Vsb$
CDGBO	LX32	$CDGBO = \partial Qd/\partial Vgb$
CDDBO	LX33	$CDDBO = \partial Qd/\partial Vdb$
CDSBO	LX34	$DSBO = \partial Qd/\partial Vsb$
QE	LX35	Substrate charge (QE)
CQE	LX36	Substrate charge current (CQE)

Name	Alias	Description
VBS	LX43	Body-source voltage (VBS)
ICH	LX44	Channel current
IBJT	LX45	Parasitic BJT collector current
III	LX46	Impact ionization current
IGIDL	LX47	GIDL current
ITUN	LX48	Tunneling current

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

Parameter	Unit	Default	Description
CAPMOD	-	2	Flag for the short channel capacitance model
LEVEL	-	-	LEVEL 59 for BSIM3SOI
MOBMOD	-	1	Mobility model selector
NOIMOD	-	1	Flag for Noise model
SHMOD	-	0	Flag for self-heating: 0 - no self-heating 1 - self-heating

Process Parameters

Parameter	Unit	Default	Description
NCH	$1/cm^3$	1.7e17	Channel doping concentration
NGATE	$1/cm^3$	0	Poly gate doping concentration
NSUB	$1/cm^3$	6.0e16	Substrate doping concentration
TBOX	m	3.0e-7	Buried oxide thickness
TOX	m	1.0e-8	Gate oxide thickness
TSI	m	1.0e-17	Silicon film thickness

DC Parameters

Parameter	Unit	Default	Description
A0	-	1.0	Bulk charge effect coefficient for channel length
A1	1/V	0.0	First non-saturation effect parameter
A2	-	1.0	Second non-saturation effect parameter
ABP	-	1.0	Coefficient of A_{beff} dependency on V_{gst}
ADICE0	-	1	DICE bulk charge factor
AGIDL	1/W	0.0	GIDL constant
AGS	1/V	0.0	Gate bias coefficient of A_{bulk}
AII	1/V	0.0	First Leff dependence Vdsatii parameter
ALPHA0	m/V	0.0	The first parameter of impact ionization current
ALPHA1	1/V	1.0	The second parameter of impact ionization current
B0	m	0.0	Bulk charge effect coefficient for channel width
B1	m	0.0	Bulk charge effect width offset
BGIDL	V/m	0.0	GIDL exponential coefficient
BII	m/V	0.0	Second Leff dependence Vdsatii parameter
CDSC	F/m^2	2.4e-4	Drain/source to channel coupling capacitance
CDSCB	F/m^2	0	Body-bias sensitivity of cdsc
CDSCD	F/m^2	0	Drain-bias sensitivity of cdsc

Parameter	Unit	Default	Description
CII	-	0.0	First V_{ds} dependence Vdsatii parameter
CIT	F/m^2	0.0	Interface trap capacitance
DELP	V	0.02	Constant for limiting V_{bseff} to surface potential
DELTA	-	0.01	Effective V_{ds} parameter
DII	V	-1.0	Second V_{ds} dependence Vdsatii parameter
DROUT	-	0.56	L dependence coefficient of the DIBL correction parameter in R_{out}
DSUB	-	0.56	DIBL coefficient exponent
DVBD0	V	0	First coefficient of $V_{bs}0$ dependency on Leff
DVBD1	V	0	Second coefficient of $V_{bs}0$ dependency on Leff
DVT0	-	2.2	First coefficient of short-channel effect on V_{th}
DVT0W	-	0	First coefficient of narrow width effect on V_{th} for small channel length
DVT1	-	0.53	Second coefficient of short-channel effect on V_{th}
DVT1W	-	5.3e6	Second coefficient of narrow width effect on V_{th} for small channel length
DVT2	1/V	-0.032	Body-bias coefficient of short-channel effect on V_{th}
DVT2W	1/V	-0.032	Body-bias coefficient of narrow width effect on V_{th} for small channel length

Parameter	Unit	Default	Description
DWB	m/V$^{1/2}$	0.0	Coefficient of Weff's substrate body bias dependence
DWG	m/V	0.0	Coefficient of Weff's gate dependence
EDL	m	2e-6	Electron diffusion length
ETA0	-	0.08	DIBL coefficient in the subthreshold region
ETAB	1/V	-0.07	Body-bias coefficient for the subthreshold DIBL effect
ISBJT	A/m^2	1.0e-6	BJT injection saturation current
ISDIF	A/m^2	0	Body to source/drain injection saturation current
ISREC	A/m^2	1.0e-5	Recombination in depletion saturation current
ISTUN	A/m^2	0.0	Reverse tunneling saturation current
K1	V$^{1/2}$	0.6	First-order body effect coefficient
K2	-	0	Second-order body effect coefficient
K3	-	0	Narrow coefficient
K3B	1/V	0	Body effect coefficient of k3
KB1	-	1	Coefficient of $V_{bs}0$ dependency on V_{gbs}
KB3	-	1	Coefficient of $V_{bs}0$ dependency on V_{gs} at subthreshold region
KBJT1	m/V	0	Parasitic bipolar early effect coefficient
KETA	m	-0.6	Body-bias coefficient of bulk charge effect

Parameter	Unit	Default	Description
LINT	m	0.0	Length offset fitting parameter from I-V without bias
MXC	-	-0.9	Fitting parameter for A_{beff} calculation
NDIODE	-	1.0	Diode non-ideality factor
NFACTOR	-	1	Subthreshold swing factor
NGIDL	V	1.2	GIDL V_{ds} enhancement coefficient
NLX	m	1.74e-7	Lateral non-uniform doping parameter
NTUN	-	10.0	Reverse tunneling non-ideality factor
PCLM	-	1.3	Channel length modulation parameter
PDIBL1	-	0.39	First output resistance DIBL effect correction parameter
PDIBL2	-	0.0086	Second output resistance DIBL effect correction parameter
PRWB	$1/V^1$	0	Body effect coefficient of Rdsw
PRWG	$1/V^{1/2}$	0	Gate bias effect coefficient of Rdsw
PVAG		0.0	Gate dependence of Early voltage
RBODY	ohm/m^2	0.0	Intrinsic body contact sheet resistance
RBSH	ohm/m^2	0.0	Extrinsic body contact sheet resistance
RDSW	$\Omega * \mu m^{wr}$	100	Parasitic resistance per unit width
RSH	ohm/square	0.0	Source/drain sheet resistance in ohm per square
U0	$cm^2/(V-sec)$	NMOS-670 PMOS-250	Mobility at Temp=Tnom

Parameter	Unit	Default	Description
UA	m/V	2.25e-9	First-order mobility degradation coefficient
UB	$(m/V)^2$	5.87e-19	Second-order mobility degradation coefficient
UC	1/V	-0.0465	Body-effect of mobility degradation coefficient
VBSA	V	0	Transition body voltage offset
VOFF	v	-0.08	Offset voltage in the subthreshold region for large W and L
VSAT	m/sec	8e4	Saturation velocity at Temp=Tnom
VTH0	v	NMOS 0.7 PMOS -0.7	Threshold voltage @ Vbs=0 for long wide device
W0	m	0	Narrow width parameter
WINT	m	0.0	Width offset fitting parameter from I-V without bias
WR	-	1	Width offset from Weff for R_{ds} calculation

AC and Capacitance Parameters

Parameter	Unit	Default	Description
ASD	V	0.3	Source/drain bottom diffusion smoothing parameter
CF	F/m	cal.	Gate to source/drain fringing field capacitance
CGDL	F/m	0.0	Lightly doped drain-gate region overlap capacitance

Parameter	Unit	Default	Description
CGDO	F/m	calculated	Non LDD region drain-gate overlap capacitance per channel length
CGEO	F/m	0.0	Gate-substrate overlap capacitance per channel length
CGSL	F/m	0.0	Lightly doped source-gate region overlap capacitance
CGSO	F/m	calculated	Non LDD region source-gate overlap capacitance per channel length
CJSWG	F/m^2	1.e-10	Source/drain (gate side) sidewall junction capacitance per unit width (normalized to 100nm Tsi)
CKAPPA	F/m	0.6	Coefficient for lightly doped region overlap capacitance fringing field capacitance
CLC	m	0.1e-7	Constant term for the short channel model
CLE	-	0.0	Exponential term for the short channel model
CSDESW	F/m	0.0	Source/drain sidewall fringing capacitance per unit length
CSDMIN	V	cal.	Source/drain bottom diffusion minimum capacitance
DLC	m	lint	Length offset fitting parameter for gate charge
DWC	m	wint	Width offset fitting parameter from C-V
MJSWG	V	0.5	Source/drain (gate side) sidewall junction capacitance grading coefficient

Parameter	Unit	Default	Description
PBSWG	V	0.7	Source/drain (gate side) sidewall junction capacitance built in potential
TT	second	1ps	Diffusion capacitance transit time coefficient
VSDFB	V	cal.	Source/drain bottom diffusion capacitance flatband voltage
VSDTH	V	cal.	Source/drain bottom diffusion capacitance threshold voltage
XPART	-	0	Charge partitioning rate flag

Temperature Parameters

Parameter	Unit	Default	Description
AT	m/sec	3.3e4	Temperature coefficient for U_a
CTH0	m^{o}C/(W*s)	0	Normalized thermal capacity
KT1	V	-0.11	Temperature coefficient for the threshold voltage
KT2	-	0.022	Body-bias coefficient of the threshold voltage temperature effect
KTIL	V*m	0	Channel length dependence of the temperature coefficient for the threshold voltage
PRT	Ω-um	0	Temperature coefficient for R_{dsw}
RTH0	m^{o}C/W	0	Normalized thermal resistance
TNOM	^{o}C	25	Temperature at which parameters are expected
UA1	m/V	4.31e-9	Temperature coefficient for U_a
UB1	(m/V)^2	-7.61e-18	Temperature coefficient for U_b

Parameter	Unit	Default	Description
UC1	1/V	-0.056	Temperature coefficient for U_c
UTE	-	-1.5	Mobility temperature exponent
XBJT	-	1	Power dependence of j_{bjt} on temperature
XDIF	-	XBJT	Power dependence of j_{dif} on temperature
XREC	-	1	Power dependence of j_{rec} on temperature
XTUN	-	0	Power dependence of j_{tun} on temperature

Note: *BSIMFD refers substrate to the silicon below buried oxide, not the well region in BSIM3. It is used to calculated backgate flatband voltage (V_{fbb}) and parameters related to source/drain diffusion bottom capacitance (V_{sdth}, V_{sdfb}, C_{sdmin}). Positive nsub means the same type of doping as the body and negative nsub means opposite type of doping.*

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

Name	Alias	Description
L	LV1	Channel length (L)
W	LV2	Channel width (W)
AD	LV3	Area of the drain diode (AD)

Name	Alias	Description
AS	LV4	Area of the source diode (AS)
ICVDS	LV5	Initial condition for drain-source voltage (VDS)
ICVGS	LV6	Initial condition for gate-source voltage (VGS)
ICVES	LV7	Initial condition for Substrate-source voltage (VES)
VTH	LV9	Threshold voltage (bias dependent)
VDSAT	LV10	Saturation voltage (VDSAT)
PD	LV11	Drain diode periphery (PD)
PS	LV12	Source diode periphery (PS)
RDS	LV13	Drain resistance (squares) (RDS)
RSS	LV14	Source resistance (squares) (RSS)
GDEFF	LV16	Effective drain conductance (1/RDeff)
GSEFF	LV17	Effective source conductance (1/RSeff)
COVLGS	LV36	Gate-source overlap capacitance
COVLGD	LV37	Gate-drain overlap capacitance
COVLGE	LV38	Gate-substrate overlap capacitance
VES	LX1	Substrate-source voltage (VES)
VGS	LX2	Gate-source voltage (VGS)
VDS	LX3	Drain-source voltage (VDS)
CDO	LX4	DC drain current (CDO)
CBSO	LX5	DC source-body diode current (CBSO)
CBDO	LX6	DC drain-body diode current (CBDO)
GMO	LX7	DC gate transconductance (GMO)
GDSO	LX8	DC drain-source conductance (GDSO)
GMESO	LX9	DC substrate transconductance (GMBSO)

Name	Alias	Description
GBDO	LX10	Conductance of the drain diode (GBDO)
GBSO	LX11	Conductance of the source diode (GBSO)

Meyer and Charge Conservation Model Parameters

Name	Alias	Description
QB	LX12	Body charge (QB)
CQB	LX13	Body charge current (CQB)
QG	LX14	Gate charge (QG)
CQG	LX15	Gate charge current (CQG)
QD	LX16	Channel charge (QD)
CQD	LX17	Channel charge current (CQD)
CGGBO	LX18	$GGBO = \partial Qg/\partial Vgb$ = CGS + CGD + CGB
CGDBO	LX19	$CGDBO = \partial Qg/\partial Vdb$, (for Meyer CGD=-CGDBO)
CGSBO	LX20	$CGSBO = \partial Qg/\partial Vsb$, (for Meyer CGS=-CGSBO)
CBGBO	LX21	$BGBO = \partial Qb/\partial Vgb$, (for Meyer CGB=-CBGBO)
CBDBO	LX22	$CBDBO = \partial Qb/\partial Vdb$
CBSBO	LX23	$CBSBO = \partial Qb/\partial Vsb$
CDGBO	LX32	$CDGBO = \partial Qd/\partial Vgb$
CDDBO	LX33	$CDDBO = \partial Qd/\partial Vdb$
CDSBO	LX34	$DSBO = \partial Qd/\partial Vsb$
QE	LX35	Substrate charge (QE)
CQE	LX36	Substrate charge current (CQE)
CDEBO	LX37	$CDEBO = \partial Qd/\partial Veb$

Name	Alias	Description
CBEBO	LX38	$BEBO = \partial Qb/\partial Veb$
CEEBO	LX39	$EEBO = \partial Qe/\partial Veb$
CEGBO	LX40	$CEGBO = \partial Qe/\partial Vgb$
CEDBO	LX41	$EDBO = \partial Qe/\partial Vdb$
CESBO	LX42	$CESBO = \partial Qe/\partial Vsb$
VBS	LX43	Body-source voltage (VBS)
ICH	LX44	Channel current
IBJT	LX45	Parasitic BJT collector current
III	LX46	Impact Ionization current
IGIDL	LX47	GIDL current
ITUN	LX48	Tunneling current

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-SPCIE MOS15 a-Si TFT model.

2. The default value for L is 100µm, and the default value for W is 100µm.

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 25C in Star-Hspice, but is 27C in some other simulators. The user may choose whether or not to set the nominal simulation temperature to 27C, 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 vgsl = 7 vmin = 0.3 vto = 0.0
```

LEVEL 61 Model Parameters

Name	Unit	Default	Description
ALPHASAT	-	0.6	Saturation modulation parameter
CGDO	F/m	0.0	Gate-drain overlap capacitance per meter channel width
CGSO	F/m	0.0	Gate-source overlap capacitance per meter channel width
DEF0	eV	0.6	Dark Fermi level position
DELTA	-	5	Transition width parameter
EL	eV	0.35	Activation energy of the hole leakage current
EMU	eV	0.06	Field effect mobility activation energy
EPS	-	11	Relative dielectric constant of substrate

Name	Unit	Default	Description
EPSI	-	7.4	Relative dielectric constant of gate insulator
GAMMA	-	0.4	Power law mobility parameter
GMIN	$m^{-3}eV^{-1}$	1E23	Minimum density of deep states
IOL	A	3E-14	Zero bias leakage current parameter
KASAT	$1/^{\circ}C$	0.006	Temperature coefficient of ALPHASAT
KVT	$V/^{\circ}C$	-0.036	Threshold voltage temperature coefficient
LAMBDA	1/V	0.0008	Output conductance parameter
M	-	2.5	Knee shape parameter
MUBAND	m^2/Vs	0.001	Conduction band mobility
RD	μ	0.0	Drain resistance
RS	μ	0.0	Source resistance
SIGMA0	A	1E-14	Minimum leakage current parameter
TNOM	$^{\circ}C$	25	Parameter measurement temperature
TOX	m	1E-7	Thin-oxide thickness
V0	V	0.12	Characteristic voltage for deep states
VAA	V	7.5E3	Characteristic voltage for field effect mobility
VDSL	V	7	Hole leakage current drain voltage parameter
VFB	V	-3	Flat band voltage
VGSL	V	7	Hole leakage current gate voltage parameter
VMIN	V	0.3	Convergence parameter
VTO	V	0.0	Zero-bias threshold voltage

Equivalent Circuit

Model Equations

Drain Current

$$I_{ds} = I_{leakage} + I_{ab}$$

$$I_{ab} = g_{ch}V_{dse}(1 + LAMBDA \cdot V_{ds})$$

$$V_{dse} = \frac{V_{ds}}{[1 + (V_{ds}/V_{sate})^M]^{1/M}}$$

$$V_{sate} = \alpha_{sat}V_{gte}$$

$$g_{ch} = \frac{g_{chi}}{1 + g_{chi}(RS + RD)}$$

$$g_{chi} = qn_sW \cdot MUBAND/L$$

$$n_s = \frac{n_{sa} n_{sb}}{n_{sa} + n_{sb}}$$

$$n_{sa} = \frac{\text{EPSI} \cdot V_{gte}}{q \cdot \text{TOX}} \left(\frac{V_{gte}}{V_{aat}}\right)^{\text{GAMMA}}$$

$$n_{sb} = n_{so} \left(\frac{t_m}{\text{TOX}} \frac{V_{gfbe}}{\text{V0}} \frac{\text{EPSI}}{\text{EPS}}\right)^{\frac{2 \cdot \text{V0}}{V_e}}$$

$$n_{so} = N_c t_m \frac{V_e}{\text{V0}} \exp\left(-\frac{\text{DEF0}}{V_{th}}\right)$$

$$N_c = 3.0 \cdot 10^{25} \text{m}^{-3}$$

$$V_e = \frac{2 \cdot \text{V0} \cdot V_{tho}}{2 \cdot \text{V0} - V_{th}}$$

$$t_m = \sqrt{\frac{\text{EPS}}{2q \cdot \text{GMIN}}}$$

$$V_{gte} = \frac{\text{VMIN}}{2}\left[1 + \frac{V_{gt}}{\text{VMIN}} + \sqrt{\text{DELTA}^2 + \left(\frac{V_{gt}}{\text{VMIN}} - 1\right)^2}\right]$$

$$V_{gt} = V_{gs} - V_T$$

$$V_{gfbe} = \frac{\text{VMIN}}{2}\left[1 + \frac{V_{gfb}}{\text{VMIN}} + \sqrt{\text{DELTA}^2 + \left(\frac{V_{gfb}}{\text{VMIN}} - 1\right)^2}\right]$$

$$V_{gfb} = V_{gs} - \text{VFB}$$

$$I_{leakage} = I_{hl} + I_{\min}$$

$$I_{hl} = \text{IOL}\left[\exp\left(\frac{V_{ds}}{\text{VDSL}}\right) - 1\right]\exp\left(-\frac{V_{gs}}{\text{VGSL}}\right)\exp\left[\frac{\text{EL}}{q}\left(\frac{1}{V_{tho}} - \frac{1}{V_{th}}\right)\right]$$

$$I_{\min} = \text{SIGMA0} \cdot V_{ds}$$

Temperature Dependence

$$V_{tho} = k_B \cdot \text{TNOM}/q$$

$$V_{th} = k_B \cdot (\text{TEMP})/q$$

$$V_{aat} = VAA\exp\left[\frac{\text{EMU}}{q \cdot \text{GAMMA}}\left(\frac{1}{V_{th}} - \frac{1}{V_{tho}}\right)\right]$$

$$V_T = \text{VTO} + \text{KVT}(\text{TEMP} - \text{TNOM})$$

$$\alpha_{sat} = \text{ALPHASAT} + \text{KASAT}(\text{TEMP} - \text{TNOM})$$

Capacitance

$$C_{gs} = C_f + \frac{2}{3}C_{gc}\left[1 - \left(\frac{V_{sate} - V_{dse}}{2V_{sate} - V_{dse}}\right)^2\right]$$

$$C_{gd} = C_f + \frac{2}{3}C_{gc}\left[1 - \left(\frac{V_{sate}}{2V_{sate} - V_{dse}}\right)^2\right]$$

$$C_f = 0.5 \cdot \text{EPS} \cdot W$$

$$C_{gs} = q\frac{dn_{sc}}{dV_{gs}}$$

$$n_{sc} = \frac{n_{sac}n_{sbc}}{n_{sac} + n_{sbc}}$$

$$n_{sac} = \frac{\text{EPSI} \cdot V_{gte}}{q \cdot \text{TOX}}$$

$$n_{sbc} = n_{sb}$$

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 μm:
    - 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_{gc} 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 100µm, and the default value for W is 100µm.

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 mu1 = 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

Name	Unit	Default	Description
ASAT	-	1	Proportionality constant of Vsat
AT	m/V	3E-8	DIBL parameter 1
BLK	-	0.001	Leakage barrier lowering constant
BT	m·V	1.9E-6	DIBL parameter 2
CGDO	F/m	0	Gate-drain overlap capacitance per meter channel width
CGSO	F/m	0	Gate-source overlap capacitance per meter channel width
DASAT	1/$^\circ$C	0	Temperature coefficient of ASAT
DD	m	1400 Å	Vds field constant
DELTA	-	4.0	Transition width parameter
DG	m	2000 Å	Vgs field constant
DMU1	cm^2/Vs$^\circ$C	0	Temperature coefficient of MU1
DVT	V	0	The difference between VON and the threshold voltage
DVTO	V/$^\circ$C	0	Temperature coefficient of VTO
EB	EV	0.68	Barrier height of diode
ETA	-	7	Subthreshold ideality factor
ETAC0	-	ETA	Capacitance subthreshold ideality factor at zero drain bias
ETAC00	1/V	0	Capacitance subthreshold coefficient of drain bias
I0	A/m	6.0	Leakage scaling constant
I00	A/m	150	Reverse diode saturation current

Name	Unit	Default	Description
LASAT	M	0	Coefficient for length dependence of ASAT
LKINK	M	19E-6	Kink effect constant
MC	-	3.0	Capacitance knee shape parameter
MK	-	1.3	Kink effect exponent
MMU	-	3.0	Low field mobility exponent
MU0	cm^2/Vs	100	High field mobility
MU1	cm^2/Vs	0.0022	Low field mobility parameter
MUS	cm^2/Vs	1.0	Subthreshold mobility
RD	μ	0	Drain resistance
RDX	Ω	0	Resistance in series with C$_{gd}$
RS	μ	0	Source resistance
RSX	Ω	0	Resistance in series with C$_{gs}$
TNOM	°C	25	Parameter measurement temperature
TOX	m	1e-7	Thin-oxide thickness
V0	V	0.12	Characteristic voltage for deep states
VFB	V	-0.1	Flat band voltage
VKINK	V	9.1	Kink effect voltage
VON	V	0	On-voltage
VTO	V	0	Zero-bias threshold voltage

Equivalent Circuit

Model Equations

Drain Current

The expression for the subthreshold current is given by:

$$I_{sub} = \text{MUS} \cdot C_{ox}\frac{\text{W}}{\text{L}}V_{sth}^{2}\exp\left(\frac{V_{GT}}{V_{sth}}\right)\left[1 - \exp\left(-\frac{V_{DS}}{V_{sth}}\right)\right]$$

$$V_{sth} = ETA \cdot V_{th}, V_{th} = k_B \cdot TEMP/q$$

$$C_{ox} = \varepsilon_i \cdot \text{L} \cdot \text{W}/\text{TOX}$$

$$V_{GT} = V_{GS} - V_T$$

$$V_T = V_{TX} - \frac{\text{AT} \cdot V_{DS}^{2} + \text{BT}}{\text{L}}$$

where $\varepsilon\iota$ is the dielectric constant of the oxide and k_B is the Boltzmanns constant.

Above threshold ($VGT > 0$), the conduction current is given by:

$$
I_a = \begin{cases}
\mu_{FET} C_{ox} \dfrac{W}{L} \left(V_{GT} V_{DS} - \dfrac{V_{DS}^2}{2\alpha_{sat}} \right) & \text{for } V_{DS} < \alpha_{sat} V_{GT} \\[4ex]
\mu_{FET} C_{ox} \dfrac{W}{L} \dfrac{V_{GT}^2 \alpha_{sat}}{2} & \text{for } V_{DS} \geq \alpha_{sat} V_{GT}
\end{cases}
$$

$$
\frac{1}{\mu_{FET}} = \frac{1}{\text{MUO}} + \frac{1}{\mu_1 (2 V_{GT} / V_{sth})^{\text{MMU}}}
$$

Subthreshold leakage current is the result of thermionic field emission of carriers through the grain boundary trap states and is described by:

$$
I_{leak} = \text{I0} \cdot \text{W} \left[\exp\left(\frac{q \cdot \text{BLK} \cdot V_{DS}}{kT} \right) - 1 \right] [X_{TFE} + X_{TE}] + I_{diode}
$$

$$
X_{TFE} = \frac{X_{TFE,lo} X_{TFE,hi}}{X_{TFE,lo} + X_{TFE,hi}}
$$

$$
X_{TE} = \exp(-W_C)
$$

$$
W_C = (E_c - E_t) / kT = 0.55 \text{eV} / kT
$$

$$X_{TEF,lo} = \begin{cases} \dfrac{4\sqrt{\pi}}{3}f\exp\left(\dfrac{4}{27}f^2 - W_C\right) & \text{for } f \leq f_{lo} \\[4mm] X_{TFE,lo}(f_{lo})\exp\left[\left(\dfrac{1}{f_{lo}} + \dfrac{8}{27}f_{lo}\right)(f - f_{lo})\right] & \text{for } f > f_{lo} \end{cases}$$

$$f = F/F_0$$

$$F = \left[\dfrac{V_{DS}}{DD} - \dfrac{V_{GS} - VFB}{DG}\right]$$

$$F_0 = (kT)^{3/2}\dfrac{4}{3}\dfrac{2\pi\sqrt{2m^*}}{qh}$$

$$m^* = 0.27m_0$$

$$f_{lo} = \dfrac{3}{2}(\sqrt{W_C + 1} - 1)$$

$$X_{TEF,hi} = \begin{cases} \dfrac{2W_C}{3}\exp\left(1 - \dfrac{2W_C}{3}\right) & \text{for } f < f_{hi} \\[4mm] \left(1 - \dfrac{\sqrt[3]{W_C}}{2f}\right)^{-1}\exp\left(\dfrac{-W_C^{3/2}}{f}\right) & \text{for } f \geq f_{hi} \end{cases}$$

$$f_{hi} = 3\left(\dfrac{W_C^{3/2}}{2W_C - 3}\right)$$

$$I_{diode} = 100 \cdot W\exp\left(-\dfrac{EB}{k_B T}\right)\left[1 - \exp\left(-\dfrac{qV_{DS}}{k_B T}\right)\right]$$

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:

$$I_{kink} = A_{kink}I_a(V_{DS} - V_{DSAT})\exp\left(-\frac{\text{VKINK}}{V_{DS} - V_{DSAT}}\right)$$

$$A_{kink} = \frac{1}{\text{VKINK}}\left(\frac{\text{LKINK}}{\text{L}}\right)^{\text{MK}}, V_{DSAT} = \alpha_{sat}V_{GT}$$

The impact ionization current, *Ikink*, is added to the drain current.

Threshold Voltage

If VTO is not specified:

$$V_T = \text{VON} - \text{DVT}$$

else:

$$V_T = \text{VTO}$$

Temperature Dependence

$$V_{TX} = V_T - \text{DVTO}(\text{TEMP} - \text{TNOM})$$

$$\mu_1 = \text{MU1} + \text{DMU1}(\text{TEMP} - \text{TNOM})$$

$$\alpha_{sat} = \text{ASAT} - \frac{\text{LASAT}}{\text{L}} - \text{DASAT}(\text{TEMP} - \text{TNOM})$$

Capacitance

$$C_{gs} = C_f + \frac{2}{3}C_{gcd}\left[1 - \left(\frac{V_{DSAT} - V_{DSE}}{2V_{DSAT} - V_{DSE}}\right)^2\right]$$

$$C_{gd} = C_f + \frac{2}{3}C_{gcs}\left[1 - \left(\frac{V_{DSAT}}{2V_{DSAT} - V_{DSE}}\right)^2\right]$$

$$C_f = 0.5 \cdot \text{EPS} \cdot \text{W}$$

$$C_{gcd} = \frac{C_{ox}}{1 + \eta_{cd}\exp\left(-\frac{V_{GT}}{\eta_{cd}V_{th}}\right)}$$

$$C_{gcs} = \frac{C_{ox}}{1 + \text{ETAC0} \cdot \exp\left(-\frac{V_{GT}}{\text{ETAC0} \cdot V_{th}}\right)}$$

$$C_{ox} = \text{W} \cdot \text{L} \cdot \varepsilon_i / \text{TOX}, \eta_{cd} = \text{ETAC0} + \text{ETAC00} \cdot V_{DSE}$$

$$V_{DSE} = \frac{V_{DS}}{[1 + (V_{DS}/V_{DSAT})^{\text{MC}}]^{1/\text{MC}}}$$

Avant!

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:

```
%cp -r mos3 mos222
```

3. Add the following line in the configuration file *config*:

```
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:

```
% 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 <u>not</u> 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
```

```
[1790]  |              0|            0|NOTY  |GLOB  |0      |UNDEF  |sqrt
[1758]  |              0|            0|NOTY  |GLOB  |0      |UNDEF  |strcasecmp
[1848]  |              0|            0|NOTY  |GLOB  |0      |UNDEF  |strcpy
[1858]  |              0|            0|NOTY  |GLOB  |0      |UNDEF  |strlen
[1800]  |              0|            0|NOTY  |GLOB  |0      |UNDEF  |strncpy
```

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 (*gs, gd*)
- ■ Substrate diode current (*ibd, ibs*)
- ■ Substrate diode conductance (*gbd, gbs)*
- ■ Substrate diode charge (*qbd, qbs)*
- ■ Substrate diode junction capacitance (*capbd, capbs*)
- ■ Substrate impact ionization current (*isub*)
- ■ Substrate impact ionization transconductances *(gbgs=dIsub/dVgs, gbds=dIsub/dVds*, *gbbs=dIsub/dVbs*)
- ■ 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 *vds, vgs and vbs* are used to provide bias conditions, while the rest of the entries carry the results from evaluating the model equations.

```
/* must be consistent with its counterpart in HSPICE */
typedef struct CMI_var {

  /* device input formation */
  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 *CMIenv*. The CMIenv structure can be accessed via the global variable *pCMIenv* (pointer to the global CMIenv struct). The structure for *CMIenv* is defined in type *CMI_ENV,* which can be found in the *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.

```
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);
/* extended model topology,0 is normal mos, 1 is berkeley SOI,
etc.  */
int     topoid;
} CMI_MOSDEF;
```

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

```
/* 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)
```

pmodel	Pointer to the model instance
pmos	1 if PMOS, 0 if NMOS
level	model level value passed from parser

Example

```
int
#ifdef __STDC__
CMImos3ResetModel(
char *pmodel,
int   pmos)
#else
CMImos3ResetModel(pmodel,pmos)
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".

```
int CMI_ResetInstance(char* pinst)
```

  pinst   Pointer to the instance

Example

```
int
#ifdef __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.

```
int CMI_AssignModelParm(char* pmodel, char* pname,double value)
```

      pmodel   Pointer to the model instance

      pname   *S*tring of parameter name

      value   Parameter value

Example

```
int
#ifdef __STDC__
CMImos3AssignMP(
char   *pmodel,
char   *pname,
double  value)
#else
CMImos3AssignMP(pmodel,pname,value)
char   *pmodel;
char   *pname;
double  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.

```
int CMI_AssignInstanceParm(char *pinst,char* pname,double value)
```

pinst	Pointer to the instance
pname	String of parameter name
value	Parameter value

Example

```
int
#ifdef __STDC__
CMImos3AssignIP(
char    *ptran,
char    *pname,
double  value)
#else
CMImos3AssignIP(ptran,pname,value)
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.

```
int CMI_SetupModel(char* pmodel)
```

   pmodel   Pointer to the model

Example

```
int
#ifdef __STDC__
CMImos3SetupModel(
```

```
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.

```
int CMI_SetupInstance(char* pinst)
```

   pinst   Pointer to the instance

Example

```
int
#ifdef __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.

```
   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

```
   int
   #ifdef __STDC__
   CMImos3Evaluate(
   CMI_VAR *pslot,
   char    *pmodel,
   char    *ptr)
   #else
   CMImos3Evaluate(pslot,pmodel,ptr)
   CMI_VAR *pslot;
   char    *pmodel;
   char    *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)
```

pvar   Pointer to *CMI_VAR* variable

pmodel   Pointer to the model

pinst   Pointer to the instance

Example

```
int
#ifdef __STDC__
CMImos3DiodeEval(
CMI_VAR *pslot,
char    *pmodel,
char    *ptr)
#else
CMImos3Diode(pslot,pmodel,ptr)
CMI_VAR *pslot;
char    *pmodel;
char    *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;
```

```
   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.*

```
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

```
int
#ifdef __STDC__
CMImos3Noise(
CMI_VAR *pslot,
char    *pmodel,
char    *ptr)
#else
CMImos3Noise(pslot,pmodel,ptr)
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)
```

   pmodel   Pointer to the model

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.

```
int CMI_FreeModel(char *pmodel)
```

   pmodel   Pointer to the model

Example

```
   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.

```
int CMI_FreeInstance(char *pmodel. char *pinst)
```

   pmodel   Pointer to the model

   pinst   Pointer to the instance

Example

```
   int
   #ifdef __STDC__
   CMImos3FreeInstance(
   char    *pmodel,
   char    *ptr)
   #else
   CMImos3FreeInstance(pmodel,ptr)
   char    *pmodel;
   char    *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)
   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);
```

```
        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 post-
simulation time.

```
int CMI_Conclude(void)
```

Example

```
int
#ifdef __STDC__
CMImos3Conclude(void)
#else
```

```
    CMImos3Conclude(void)
    #endif
    {
(void)CMImos3conclude();

    return 0;
    } /* int CMImos3Conclude() */Internal Routines */
```

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

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	*CMIxxxGetIpar.c*
CMIxxxSetIpar.c	*CMIxxxGetMpar.c*
CMIxxxSetMpar.c	*CMIxxxeval.c*

where *xxx* is the model name.

Figure 23-2: Hierarchy of Interface and Internal Routines

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".

```
struct TOPO1 {
   double vps;
   double ves;
   double delTemp;       /* T node */
   double selfheat;
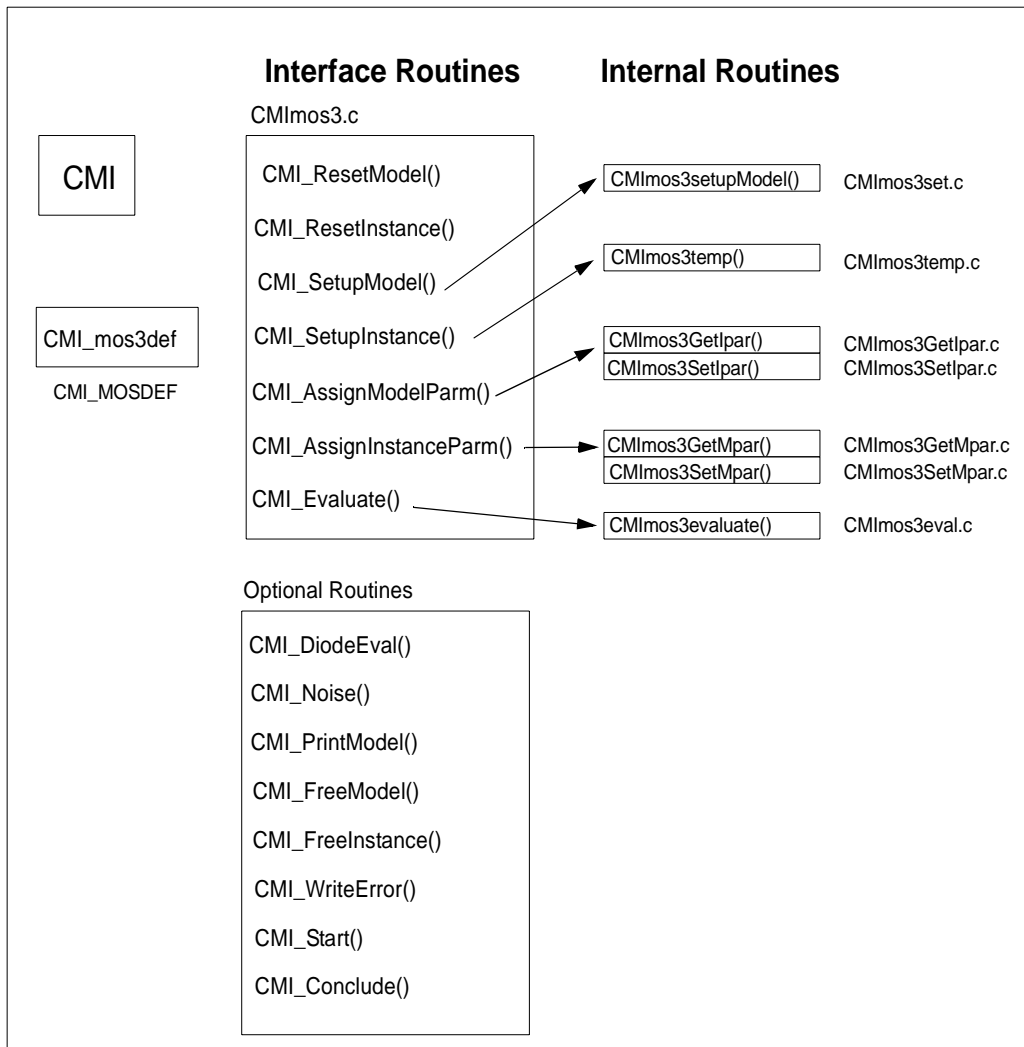
   double qsub;
   double qth;
   double cbodcon;

   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:

$$vds = vd - vs$$

$$vgs = vg - vs$$

$$vbs = vb - vs$$

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:

```
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.

Figure 23-3: MOSFET (node1, node2, node3, node4) - N-channel

node1 (drain node)

I1 (M1)

node4 (substrate node)

node2 (gate node)

I2 (M1)

I4 (M1)

node3 (source node)

I3 (M1)

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.

```
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().

Avant!

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 *vmax*, using the MAX function over the time region of interest. Then it calculates *vmin* 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 *TDval.* Finally, the delay *Tdelay* is calculated using a fixed value for the measure threshold.

Example

```
.MEAS TRAN vmax MAX V(out) FROM=TDval TO=Tstop
.MEAS TRAN vmin MIN V(out) FROM=TDval TO=Tstop
.MEAS TRAN Trise TRIG V(out) val='vmin+0.1*vmax'  TD=TDval
+ RISE=1 TARG  V(out) val='0.9*vmax' RISE=1
.MEAS TRAN  Tfall TRIG V(out) val='0.9*vmax' TD=TDval
+ FALL=2 TARG  V(out) val='vmin+0.1*vmax' FALL=2
.MEAS TRAN Tdelay TRIG V(in) val=2.5 TD=TDval 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 *Tmid*
- Defines a bounded region by finding the pedestal voltage (*Vmid*) and then finding the first time that the signal crossed this value, *Tfrom*
- Measures the ripple in the defined region using the peak-to-peak (PP) measure function from *Tfrom* to *Tmid*

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 (*m_delay* and *rms_power*). This example is based on the demonstration file *$installdir/ demo/hspice/cchar/load1.sp*.

Input File Example

```
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
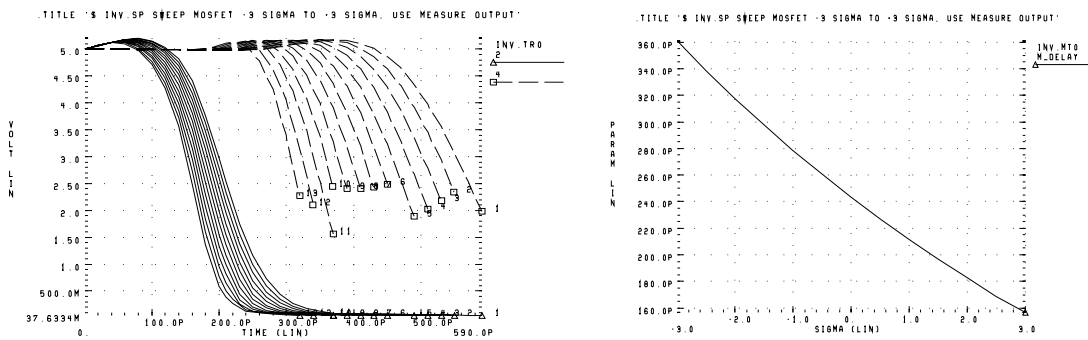
x1 in 2 inv
x2 2 3 inv
x3 3 4 inv m=fanout
```

Output Statistical Results

```
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
```

Figure 24-4: Inverter Delay and Power versus Fanout

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

```
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
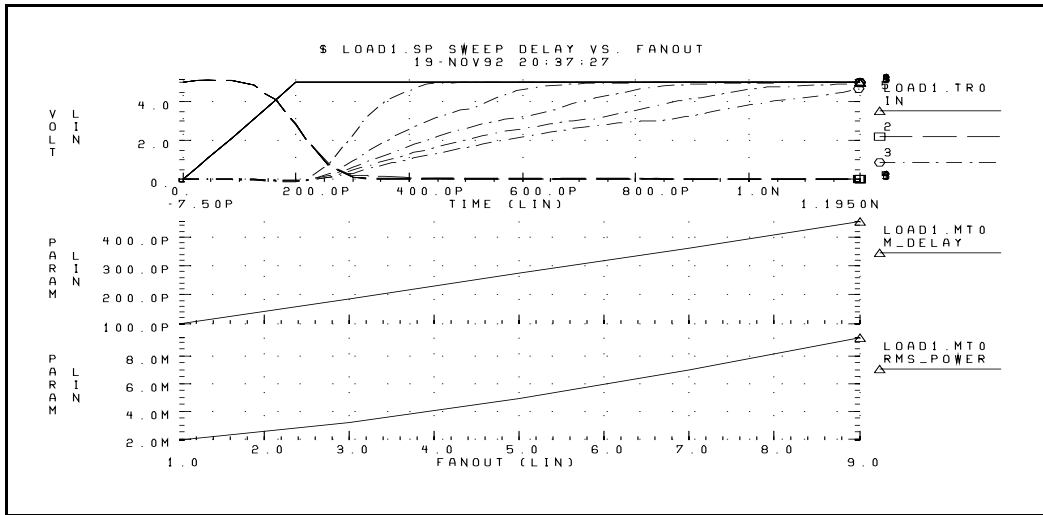
.ac lin 2 10k 100k sweep pdcin 0 5 .1
.measure ac incap find par( '-1 * ii(vin)/
+ (hertz*twopi)' ) AT=10000hertz
```

Figure 24-5: Graph of Pin Capacitance versus Inverter Input Voltage

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

Figure 24-8: Cell Characterization Circuit 2

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 $0.1 \cdot \text{VDD}$ (that is,

V(3) = 0.1VDD) and ends when the voltage at node 3 is equal to $0.9 \cdot$ 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 40INV 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
.ENDDATA
```

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
```

```
+ 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

Avant!

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
- Analyzing Board Signal Integrity

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 25-1: High-Speed Design Problems and Solutions

Signal Integrity Problem	Causes	Solution
Noise: delta I (current)	Multiple simultaneously switching drivers; high-speed devices create larger delta I	Adjust or evaluate location, size, and value of decoupling capacitors.
Noise: coupled (crosstalk)	Closely spaced parallel traces	Establish parallel line length design rules.
Noise: reflective	Impedance mismatch	Reduce the number of connectors and select proper impedance connectors.
Delay: path length	Poor placement and routing; too many or too few layers; chip pitch	Choose MCM or other high-density packaging technology.
Propagation speed	Dielectric medium	Choose dielectric with lowest dielectric constant.
Delay: rise time degradation	Resistive loss and impedance mismatch	Adjust width, thickness and length of line.

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.

Figure 25-1: Simulation of Output Buffer with 2 ns Delay and 1.8 ns Rise and Fall Times

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:

```
critical length=(rise time)*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

Figure 25-4: General Method for TDR Optimization

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

Figure 25-5: SPICE Model for Package-Plus-Test Fixture Optimized Parameters: XTD, CSMA, LPIN, and LPK

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

```
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

```
$ SPICE MODEL FOR PACKAGE-PLUS-TEST FIXTURE
$ AUTHOR: DAVID H. SMITH & RAJ M. SAVARA
.OPTION POST RELV=1E-4 RELVAR=1E-2
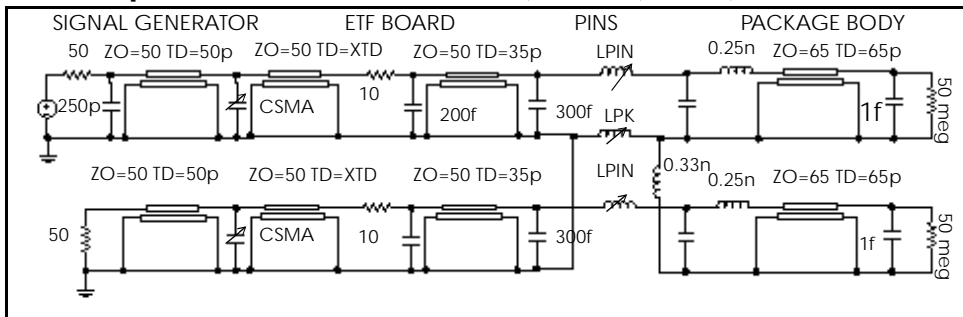
$ DEFINE PARAMETERS
.PARAM LV=-0.05 HV=0.01 TD=1P TR=25P TF=50P TPW=10N TPER=15N

$ PARAMETERS TO BE OPTIMIZED
.PARAM CSMA=OPT1(500f,90f,900f,5f)
+      XTD=OPT1(150p,100p,200p)
```

```
+       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
RLOS1 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-
     + 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

$ COMPARISON: MEASURED/CALCULATED TRANSMITTED SIGNAL
31 JUL91 08:26:44

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.

Figure 25-11: I/O Drivers/Receivers with Package Lead Inductance, Parallel 4" Lossy Microstrip Connectors

Example

This is an example of connecting I/O chips with transmission lines.

```
.OPTIONS SEARCH='$installdir/parts/signet'
.OPTIONS POST=2 TNOM=27 NOMOD LIST METHOD=GEAR
.TEMP 27
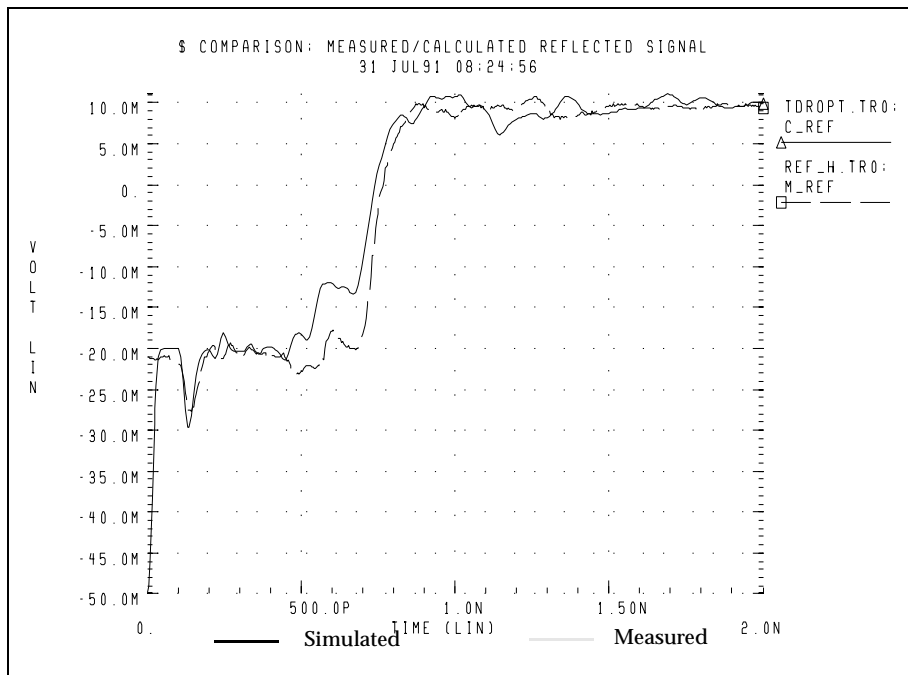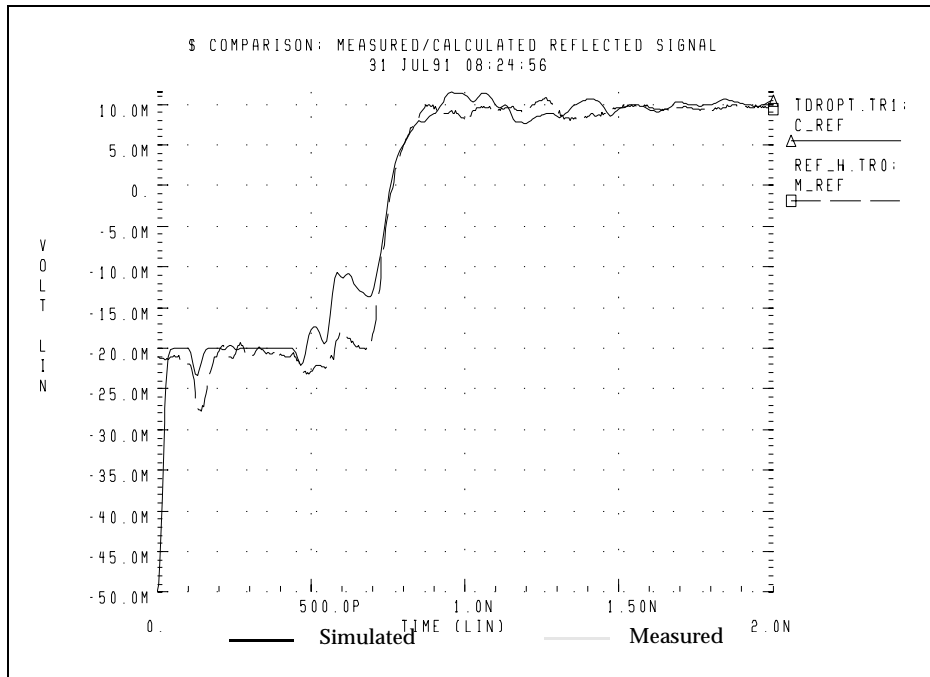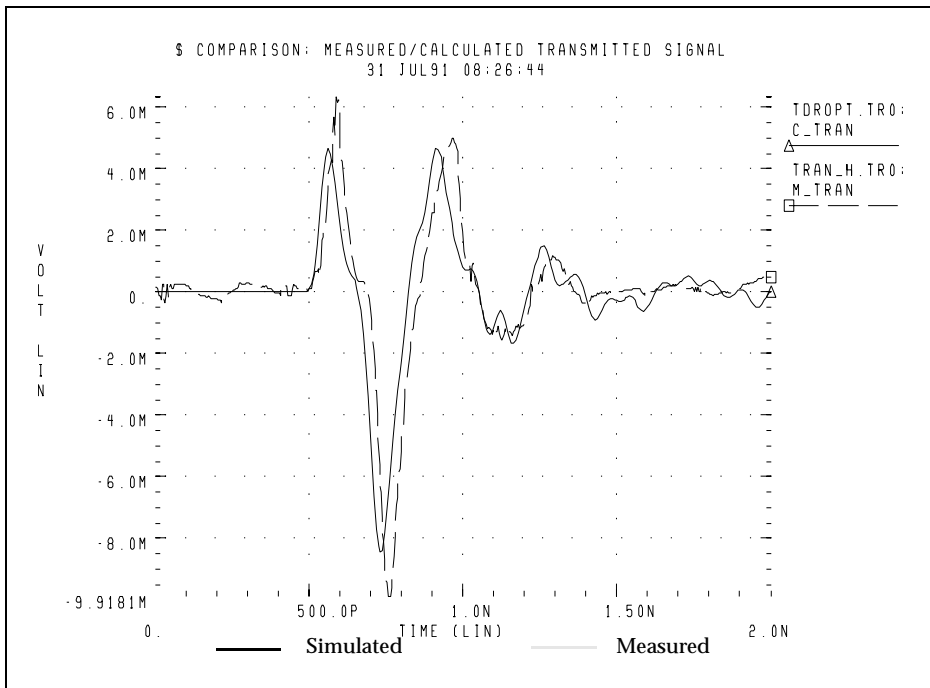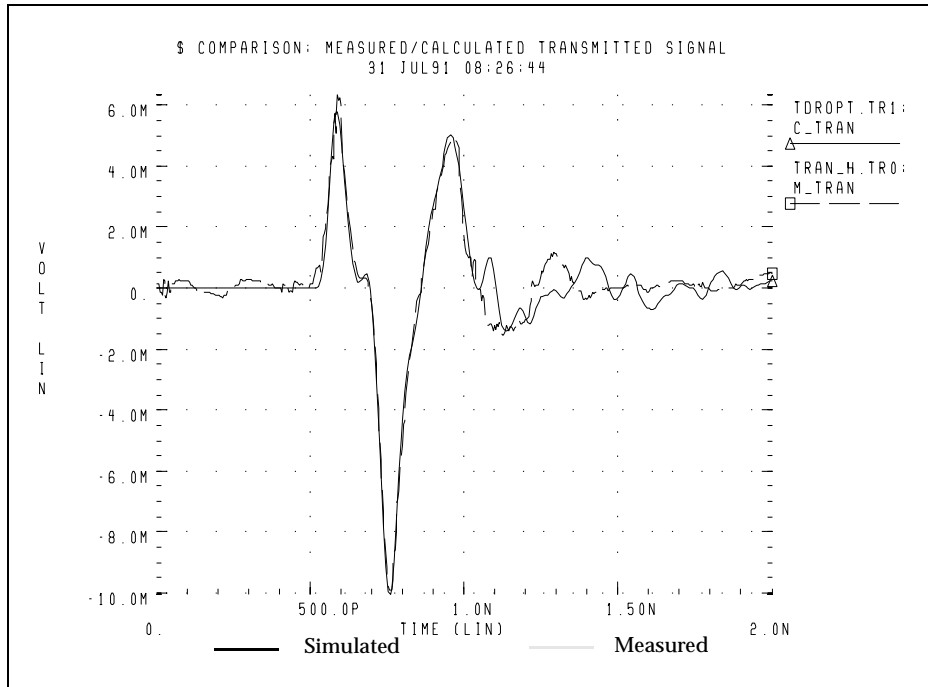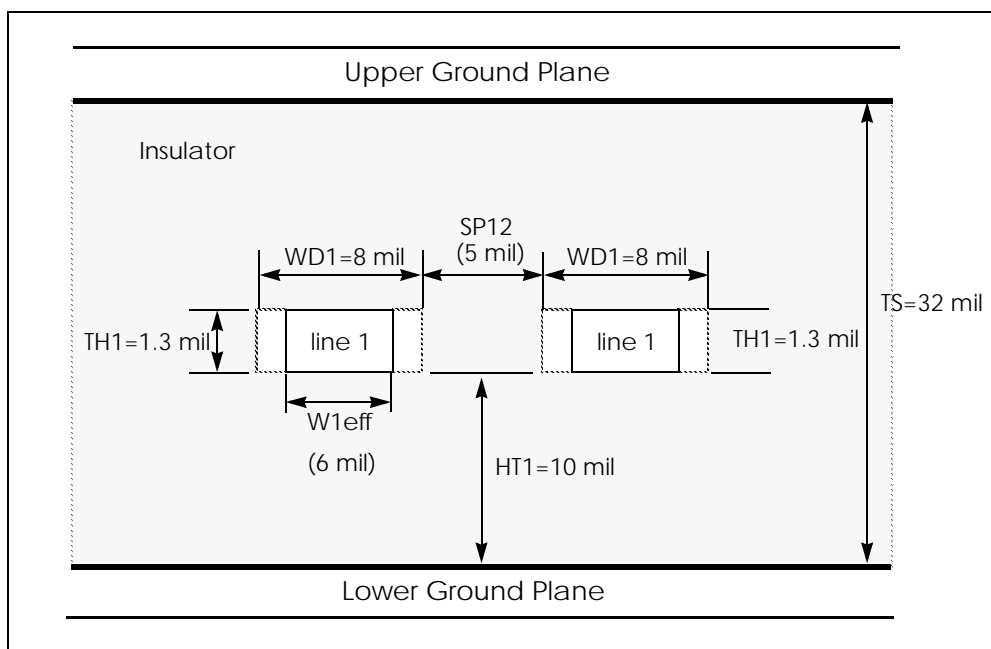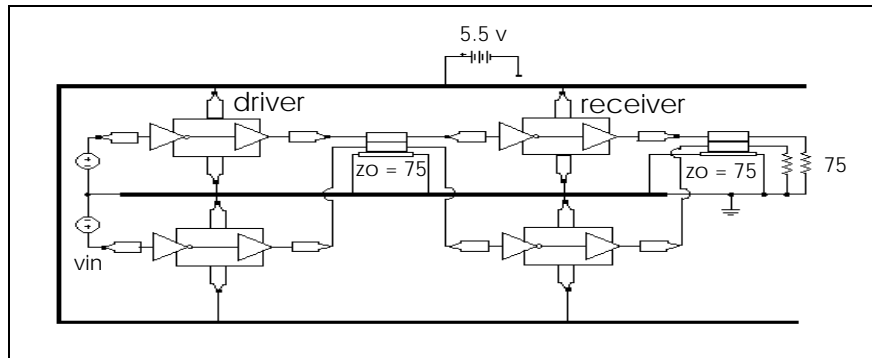
$ DEFINE PARAMETER VALUES
.PARAM LV=0 HV=3 TD1=10n TR1=3n TF1=3n TPW=20n TPER=100n
+ TD2=20n TR2=2n TF2=2n LNGTH=101.6m

$ 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(TLOUT2)
.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
```

Nodes	Description
I (IOB only)	output of the TTL/CMOS receiver
O (IOB only)	input pad driver stage
I1 (IOB4 only)	input data 1
I2 (IOB4 only)	input data 2
DRIV_IN (IOB4 only)	
PAD	bonding pad connection
TS	three-state control input (5 V disables)
FAST	slew rate control (5 V fast)
PPUB (IOB only)	pad pull-up enable (0 V enables)
PUP (IOB4 only)	pad pull-up enable (0 V enables)
PDOWN (IOB4 only)	pad pull-up enable (5 V enables)
TTL (IOB only)	CMOS/TTL input threshold select (5 V selects TTL)
VDD	5 volt supply
GND	ground
Parameters	**Description**
XIL_SIG	model distribution: (default 0) -3==> slow 0==> typical +3==> fast

XIL_DTEMP	Buffer temperature difference from ambient The default = 0 degrees if ambient is 25 degrees and the buffer is 10 degrees hotter than XIL_DTEMP=10.
XIL_SHRINK	Old or new part; (default is new) 0==>old 1==>new

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.

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
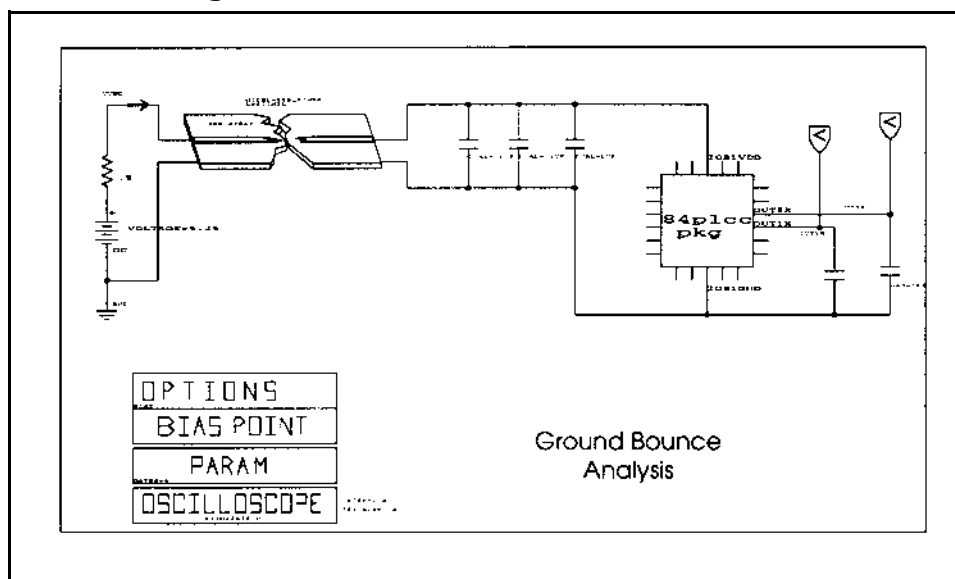
* The following is the netlist for the above schematic(fig. 10-13)
.op
.option  post list
.tran 1ns 50ns  sweep gates 8 32 4
.measure bounce  max v(out1x)
*.tran .1ns 7ns
.param gates=8
.print v(out1x) v(out8x)  i(vdd) power
$.param xil_dtemp=-65  $ -40 degrees c  (65 degrees from +25 degrees)
vdd   vdd gnd 5.25
vgnd return gnd 0
upower1 vdd return   iob1vdd iob1gnd   pcb_power  L=600mil
* local power supply capacitors
xc1a iob1vdd iob1gnd cap_mod cval=.1u
xc1b iob1vdd iob1gnd cap_mod cval=.1u
xc1c iob1vdd iob1gnd cap_mod cval=1u
xgnd_b   iob1vdd iob1gnd 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

.macro 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
```

```
x1   I1 gnd PAD1x TS FAST PPUB TTL chipvdd chipgnd  xil_iob
+ xil_sig=0 xil_dtemp=0  xil_shrink=1  m=1
```

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
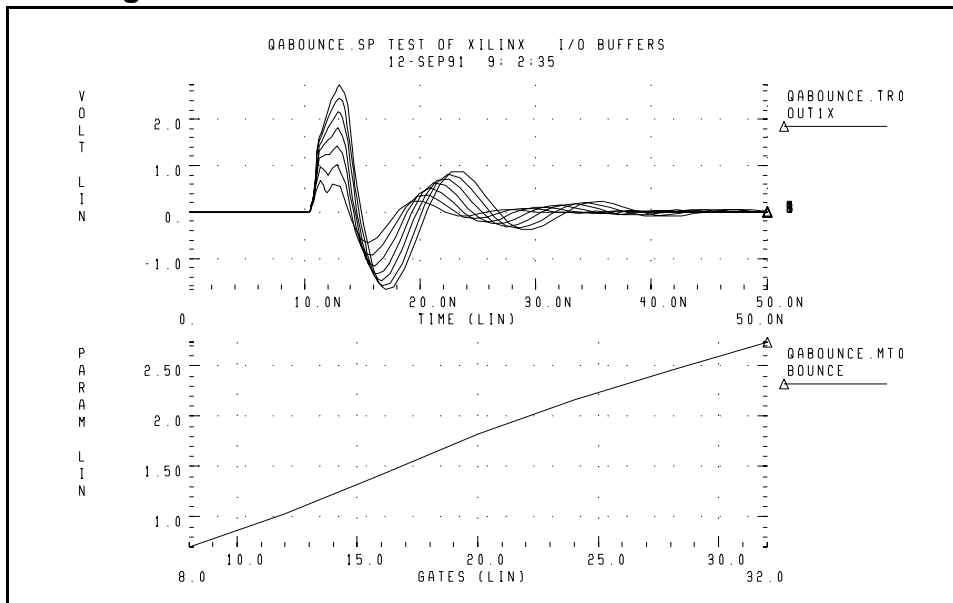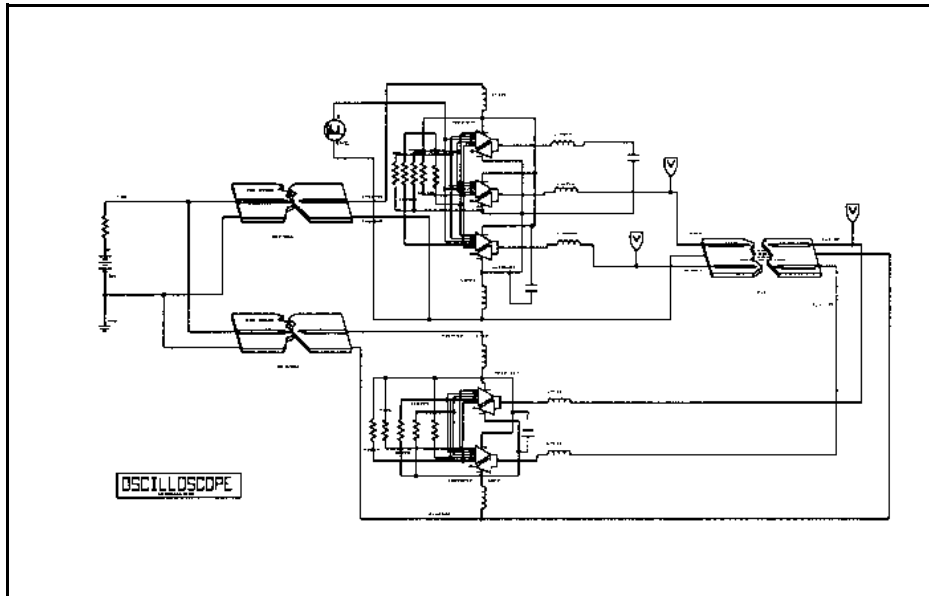
* The following is the netlist for the above schematic ( fig 10-15)
.op
.option  nomod post=2
*.tran .1ns 5ns sweep xil_sig -3 3 3
.tran .1ns 15ns
.print v(out1x) v(out3x)  i(vdd) v(irec)

vdd  vdd gnd 5
vgnd return gnd 0
```

```
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 iobuffers in tristate

xtri1  Irec O pad_tri1 TSrec FAST PPUB TTL    chipvdd chipgnd xil_iob
+ xil_sig=0 xil_dtemp=0 xil_shrink=1 m=1
xtri2  Irec O pad_tri2 TSrec FAST PPUB TTL    chipvdd chipgnd xil_iob
+ xil_sig=0 xil_dtemp=0 xil_shrink=1 m=1
```

Control Setting

```
rin_output    O    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  tri1 pad_tri1 L=3nh r=0.01
ltri2  tri2 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 iobuffers simultaneously switching through approx.
*  3nh lead inductance
* 1 iob is a receiver (tristated)

vout O chipgnd pwl 0ns 0v, 1ns 0v, 1.25ns 4v, 7ns 4v, 7.25ns 0v, 12ns
0v R
x3  I3 O PAD3x TS FAST PPUB TTL    chipvdd chipgnd  xil_iob
+ xil_sig=0 xil_dtemp=0 xil_shrink=1 m=3
```

```
x1  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

Far End Driven line

Near End Driven line

Near and far end quite line

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:
```

```
*        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 PULLL-UP ENABLE. ACTIVE LOW.
*        TTL    CMOS/TTL INPUT THRESHOLD SELECT. HIGH
*               SELECTS TTL.
*        VDD    POSITIVE SUPPLY CONNECTION FOR INTERNAL
*               CIRCUITRY.
*   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.
*        GND    CIRCUIT GROUND
```

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%S+%X;:0[(3'1:67*8-:1:\[
    kp39H2J9#Yo%XpVY#O!rDI$UqhmE%:\7%(3e%:\7\5O)1-5i# ;

    .ENDS XIL_IOB
```

PCI Modeling Using Star-Hspice

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

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: PCI Speedway

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

File Name	Description
Circuit files	
pci_wc.sp	worst-case PCI Speedway circuit file, ten devices
pci_mont.sp	example Monte Carlo file
pci_lab.sp	file prepared for the lab outlined by this document
Subcircuit "include" files	
pci_in_w.inc	worst-case PCI input load, called by *.sp* file
pci_ii_win.inc	worst-case PCI output driver, called by *.sp* file
pci_ii_t.inc	typical PCI output driver
pci_ii_b.inc	best-case PCI output driver
trace.inc	printed circuit trace subcircuit, called by *.sp* file

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
```

```
.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 td=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 Ci_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
```

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 *.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 *.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_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 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
Vsupply   VCC      0 DC VCCDC
```

PCI Speedway Subsections

To understand the reference driver and the Speedway subsections, refer to
.

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
```

```
********************************************************************
*               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
```

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

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:        *
```

```
*                                                                     *
*       .param vccdc=??.?V        $ set system voltage here           *
*                                                                     *
*  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 Ci_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
***********************************************************************
```

```
* Cin + Ci_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  ouput vcc gnd.          *
*                                                               *
*                                                               *
*  change                                                       *
*  position                                                     *
*  here                                                         *
*************************************************************************
Vdrvout  load1  drvout                      $ 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 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 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
```

```
+      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=LINETH 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=LINETH 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=LINETH 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=LINETH 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=LINETH 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=LINETH DLEVOUT=LINELYR CORRECT=1
Xstub10      stub10   load10    TRACE      LENGTH=STUB
+     W=STUBWD H=STUBHT T=STUBTH 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
***********************************************************************
.END
```

Analyzing Board Signal Integrity

This section describes the features of a tool station that analyzes printed circuit boards using Star-Hspice and PADS-LPE, a layout extraction module. A number of board layouts from different routers, such as PADS, Allegro, Mentor, PDIF and Racal, can be read in. Specify critical net names to generate Star-Hspice equivalent statements. These are transmission line (U-line) element statements for board traces, and buffer calls for the input/output and glue buffers. The model name is automatically generated for each U-line, based on the layer number, width of the line, and whether it a stripline or microstrip. You must then create the .MODEL statements. Obtain the needed inputs from the board stack-up and the model name itself. A simple mechanism allows you to choose net names classified under different signal classes. The signal class is a group of signals with common delay specification.

To maintain accuracy, local ground is represented. Handle direct terminations to power supplies carefully. Comment lines about using local ground/power are generated routinely in each partial input file. Use a field solver to get the board power/ground characteristics, then use it to degrade the supplies for direct terminations. Specify the ground/power node for each net before you run the extraction.

Special template input files for delay and overshoots/undershoots are provided. Use these files to create input files for the Star-Hspice runs. Complete the partial Star-Hspice decks and include .MEASURE, .PRINT, .OPTION, and .SOURCE statements as demonstrated in the templates.

Input Files for the Extraction Process

This section describes the input files needed for the Star-Hspice extraction process: DEMO.DAT, DEMO.SRC, DEMO.MDD, DEMO.MDL, DEMO.SNF, DEMO.BST

DEMO.DAT

This file contains the name of the routed database file with the *.SRC* extension from the router, and net names for the creation of partial Star-Hspice input netlist

files. You must specify the ground node for each net. After the extraction, the ground/power nodes appear as comments. Instead of using values of 5 V or 0 V, the node names are used in the resulting netlist. To get a more accurate representation of the supplies, use at least a resistor to get realistic levels.

```
DA[00]          GND=node0 PWR=node12 (Net names list)
DA[06]          GND=node1 PWR=node14
TRLW.EXE /PADS (call to the PADS Translator)
x%n i%n o1 %node tsi1 fast1 ppub1 ttl1 vdd1 gnd1 %mdl_iob
+ %mdl_sig=0 %mdl_dtemp=0 %mdl_shrink=0
```

(This is a prototype statement used to specify the pin order)

DEMO.SRC

This routed database file from a router such as PADS is generated using the ASCII OUT option in the PADS environment.

DEMO.MDD

This file contains path names to model files(*.MDL) used to instantiate buffer calls.

```
.BEGIN
C:\DEMOBRD\MDB\XIL3.MDL
C:\DEMOBRD\MDB\EB3.MDL
C:\DEMOBRD\MDB\DBX1.MDL
.END
```

DEMO.MDL

These model files contain pin types and part names.

```
.BEGIN
.NAME
part name (or package name)
.TYPE
part type (can be Internal for all ACTIVE devices except
diodes, resistor, capacitor, inductor, connector, diode,
package)
.INPUT PINS
pin_name list
```

```
.OUTPUT PINS
pin_name_list
.BI PINS
bidirectional_pins_list
.TRI PINS
tristate_pins_list
.PWR PINS
power_pins
.GND PINS
gnd pins
.NC PINS
unused_pin _list
.DCTERM PINS
pin_name resistance voltage
.ANODE PIN
pin_name
.CATHODE PIN
pin_name
   .VALUE
pin1 pin2 value (for resistors, inductors, capacitors)
```

DEMO.SNF

This special nets file should contain names for global power/ground.

```
.POWERBUS
VCC 5.0
GND 0.0
```

DEMO.BST

The board stack-up file describes the board as a series of structures such as
microstrip or stripline.

```
.UNITS
mils (or cm)
.STRUCTURE
microstrip (or stripline)
.DIELECTRIC
air
T=55    ER=4.5
.ENDS
```

```
.SIGNAL
L=1 H=15 T=1.4
.ENDS
.ENDS
.THRU_VIA (or .BURIED_VIA, if applicable)
1pf
.STRUCTURE
microstrip
.DIELECTRIC
T=55 ER=4.5
air
.SIGNAL
L=2 H=15 T=1.4
.ENDS
.ENDS
.RESISTIVITY value (This allows the resistivity to be specified
in ohm-cm.Default is copper)
.WID_ADJ value (Allows different built width vs. drawn width)
```

Output Files from the Extraction Process

This section describes the output files you get from the Star-Hspice extraction
process.

DEMO.NET

This file is a partial Star-Hspice-compatible netlist. One file is created for each
net specified in the *.DAT* file. The netlist describes a Xilinx driver, driving a
transmission line into a through via (represented by a capacitor) to the bottom
side of the board into a Xilinx receiver. The board is made of four layers. Using
the templates provided, complete the input files to run Star-Hspice.

```
x1 i1 o1 n1 tsi1 fast1 ppub1 ttl1 vdd gnd xil_iob
+ xil_sig=0 xil_dtemp=0 xil_shrink=1
x2 i2 n3 pad1 tsi2 fast1 ppub1 ttl1 vdd1 gnd1
+ xil_iob  xil_sig=0 xil_dtemp=0 xil_shrink=1
u1 n1 gnd n2 out_ref_1 umod1_4 l=1200mil
u2 n2 out_ref_1 n3 out_ref_2 umod4_4 l=300mil
c1 n2 out_ref_1 0.01pf
```

DEMO.CRF

Generate this cross-reference file containing names of buffer instances found in the database for which there are no models pointed to by the *.MDL* file. Use this list as a reference to create the missing *.MDL* files.

Running PADS-LPE (The Extraction Program)

The program can be invoked from within Microsoft$^®$ Windows$^™$ as "PADS-LPE".

You must add the path name to the file *autoexec.bat* so it can be invoked without the complete path name.

Delay Analysis Template

Use this Star-Hspice input file as a template for DELAY analysis of a net from a PADS file.

```
*   Database

*supplies
v1 vdd gnd 5.0v

v3 a gnd 0v
v4 b gnd 0v
v5 c gnd 5v
v6 d gnd 0v
v9 e gnd 5v


* Piece-wise linear statement - a 0.5ns rise time is assumed

vdat o1 gnd pwl 0ns 0v 9.5ns 0v 10ns 5v 19.5ns 5v 20ns 0v
+ 30ns 0v
```

Control Settings

```
r1 a tsi1 1
r2 b tsi2 1
r3 c fast1 1
r4 d ttl1 1
```

```
r5 e ppub1 1
c6 i2 gnd 0.5pf
v7 vdd1 gnd 4.5v
v8 gnd1 gnd 0.5v
```

Default Netlist Created Automatically Per Net

```
x1 i1 o1 n1 tsi1 fast1 ppub1 ttl1 vdd gnd xil_iob
+ xil_sig=0 xil_dtemp=0 xil_shrink=1
x2 i2 n3 pad1 tsi2 fast1 ppub1 ttl1 vdd1 gnd1
+ xil_iob  xil_sig=0 xil_dtemp=0 xil_shrink=1
u1 n1 gnd n2 out_ref_1 umod1_4 l=1200mil
u2 n2 out_ref_1 n3 out_ref_2 umod4_4 l=300mil
c1 n2 out_ref_1 0.01pf
```

.Measure Statements For Rise And Fall Delay Measurement

```
.measure tran rise1 trig v(o1) val=2.5ns rise=1
+ targ v(n3) val=2.5ns rise=1
.measure tran fall1 trig v(o1) val=2.5ns fall=1
+ targ v(n3) val=2.5ns fall=1
.model umod1_4 u LEVEL=3 elev=1 plev=1 llev=1 th=1.4mil
+ ht=10mil kd=5 dlev=1 wd=4mil
.model umod4_4 u LEVEL=3 elev=1 plev=1 llev=1 th=1.4mil
+ ht=10mil kd=5 dlev=1 wd=4mil
.option post list
.tran 0.1ns 30ns
.end
```

Figure 25-21: Delay Template Output

Overshoot/Undershoot Analysis Template

```
* This HSPICE input file can be used as a template for OVER/
UNDER SHOOTS
* analysis of a net from a PADS Database
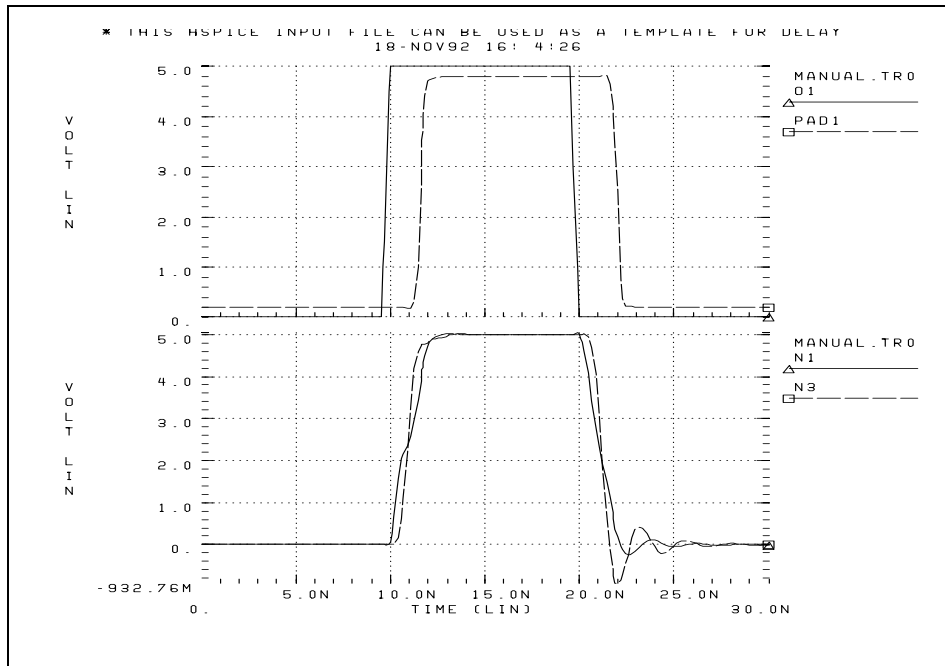
*supplies
v1 vdd gnd 5.0v

v3 a gnd 0v
v4 b gnd 0v
v5 c gnd 5v
v6 d gnd 0v
v9 e gnd 5v

* Piece-wise linear statement - a 0.5ns rise time is assumed

vdat o1 gnd pwl 0ns 0v 9.5ns 0v 10ns 5v 19.5ns 5v 20ns 0v
```

```
+ 39.5ns 0v 40ns 5v 50ns 5v
```

Control Settings

```
r1 a tsi1 1
r2 b tsi2 1
r3 c fast1 1
r4 d ttl1 1
r5 e ppub1 1
*c6 i2 gnd 0.5pf
v7 vdd1 gnd 4.8v
v8 gnd1 gnd 0.2v
```

Default Netlist Created Automatically Per Net

```
* The netlist describes a Xilinx driver, driving a transmission
line
* into a through via(represented by a capacitor) to the bottom
side
* of the board into a Xilinx receiver. The board is made of 4
layers.

x1 i1 o1 n1 tsi1 fast1 ppub1 ttl1 vdd gnd xil_iob
+ xil_sig=0 xil_dtemp=0 xil_shrink=1
x2 i2 n3 pad1 tsi2 fast1 ppub1 ttl1 vdd1 gnd1
+ xil_iob  xil_sig=0 xil_dtemp=0 xil_shrink=1
u1 n1 gnd n2 out_ref_1 umod1_4 l=1200mil
u2 n2 out_ref_1 n3 out_ref_2 umod4_4 l=1200mil
c1 n2 out_ref_1 0.01pf


* measure statements for overshoots and undershoots
* Node n3 was chosen since it is the input directly to the
receiver,
* and may cause * the receiver to switch falsely. Overshoot is
* more of a concern after the first fall,
* Undershoot should be examined after the first rise.


* Over/Under shoots after the first rise of node n3

.measure tran osh1 when v(n3)='0.5*5' cross=1
.measure tran osh2 when v(n3)='0.5*5' cross=2
.measure tran tmid_o param='(osh1+osh2)/2'
.measure tran vmid_o find v(n3) AT='tmid_o'
```

```
.measure tran o_from when v(n3)='vmid_o' rise=1
.measure tran overshoot1 MAX v(n3) from='o_from' to='tmid_o'
.measure tran undershoot1 MIN v(n3) from='o_from' to='tmid_o'

* Over/Under shoots after the first fall of node n3

.measure tran ush1 when v(n3)='0.5*5' cross=2
.measure tran ush2 when v(n3)='0.5*5' cross=3
.measure tran tmid_u param='(ush1+ush2)/2'
.measure tran vmid_u find v(n3) AT='tmid_u'
.measure tran u_from when v(n3)='vmid_u' fall=1
.measure tran overshoot2 MAX v(n3) from='u_from' to='tmid_u'
.measure tran undershoot2 MIN v(n3) from='u_from' to='tmid_u'

.model umod1_4 u LEVEL=3 elev=1 plev=1 llev=1 th=1.4mil
+ ht=10mil kd=5 dlev=1 wd=40mil

.model umod4_4 u LEVEL=3 elev=1 plev=1 llev=1 th=1.4mil
+ ht=10mil kd=5 dlev=1 wd=40mil

.option post
.tran 0.1ns 50ns
.end
```

Figure 25-22: Overshoot Template Example

Chapter 26

Performing Behavioral Modeling

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:

```
 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 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.

Figure 26-1: Netlisting by Signal Mode

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:

```
E1 5 0 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 FA=V(3,2). The dependent source statement then specifies that P0=1 and P1=2.5. From the one-dimensional polynomial equation above, the defining equation for E1 is:

$$E1 = 1 + 2.5 \cdot V(3,2)$$

Two-Dimensional Function

Where the function is two-dimensional, that is, a function of two node voltages or two branch currents, FV is determined by the following expression:

$$FV = P0 + (P1 \cdot FA) + (P2 \cdot FB) + (P3 \cdot FA^2) + (P4 \cdot FA \cdot FB) + (P5 \cdot FB^2)$$
$$+ (P6 \cdot FA^3) + (P7 \cdot FA^2 \cdot FB) + (P8 \cdot FA \cdot FB^2) + (P9 \cdot FB^3) + \dots$$

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, E1, as:

$$E1 = 3 \cdot V(3,2) + 4 \cdot V(7,6)^2$$

To implement this function, use the following controlled source element statement:

```
E1 1 0 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:

$$
\begin{aligned}
FV = \ &P0 + (P1 \cdot FA) + (P2 \cdot FB) + (P3 \cdot FC) + (P4 \cdot FA^2) \\
&+ (P5 \cdot FA \cdot FB) + (P6 \cdot FA \cdot FC) + (P7 \cdot FB^2) + (P8 \cdot FB \cdot FC) \\
&+ (P9 \cdot FC^2) + (P10 \cdot FA^3) + (P11 \cdot FA^2 \cdot FB) + (P12 \cdot FA^2 \cdot FC) \\
&+ (P13 \cdot FA \cdot FB^2) + (P14 \cdot FA \cdot FB \cdot FC) + (P15 \cdot FA \cdot FC^2) \\
&+ (P16 \cdot FB^3) + (P17 \cdot FB^2 \cdot FC) + (P18 \cdot FB \cdot FC^2) \\
&+ (P19 \cdot FC^3) + (P20 \cdot FA^4) + \ldots
\end{aligned}
$$

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 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>

Behavioral 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: $V(in+,in-) = j \cdot V(n+,n-)$

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,

$$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$

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.
TC*1, TC2*	First and second order temperature coefficients. The SCALE is updated by temperature:

$$SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

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 *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

A simple inverter with no delay can be built with a piecewise linear transfer function.

```
Einv out 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)

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.

```
Evco out 0 VOL='v0+gain*SIN(6.28*freq*v(control)*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:

Linear

Gxxx n+ n- VCR in+ in- transfactor <MAX=val> <MIN=val> <SCALE=val>
+ <M=val><TC1=val> <TC2=val> <IC=val>

Polynomial

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

Gxxx n+ n- VCR PWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val>
+ <TC1=val> <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Gxxx n+ n- VCR NPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val>
+ <TC1=val> <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Gxxx n+ n- VCR PPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val>
+ <TC1=val> <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Multi-Input Gates

Gxxx n+ n- VCR gatetype(k) in1+ in1- ... ink+ ink- <DELTA=val>
+ <TC1=val> <TC2=val> <SCALE=val> <M=val> x1,y1 ... x100,y100 <IC=val>

Voltage Controlled Capacitor (VCCAP)

The syntax is:

    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 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 '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+)

G Element Parameters

ABS	Output is absolute value if 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, TIME, temperature (TEMPER), and frequency (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. 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,

$$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$

	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:
	$SCALEeff \ = \ SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$
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.

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).

```
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 0 .02 gt i_anode cathode cur='20m*v(anode,cathode)
*v(grid,cathode)'
```

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,

$$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$

The values of tstep and tstop are specified in the .TRAN statement.

p0, p1 . . .	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:

$$SCALEeff \ = \ SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

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 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.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>
+ <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

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.

MIN	Minimum output current value. The default is undefined, and 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,

$$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$

	The values of tstep and tstop are specified in the .TRAN statement.
p0, p1 …	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:

$SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$

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

```
$ 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:

$$I(F2)=1e\text{-}3 + 1.3e\text{-}3 \cdot I(VCC)$$

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 example 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

```
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
```

Figure 26-2: NAND/AND Gates

Behavioral D-Latch

In this example, a D flip-flop is modeled by one input NAND gates and NPWL/ PPWL functions.

Figure 26-3: D-Latch Circuit

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
```

```
+ -4.6000 2.5719K
+ -4.4000 2.6813K
+ -4.2000 2.8035K
+ -4.0000 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
+ -2.2000 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
xmn 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
```

```
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS inv

.subckt dlatch data clck clckn q qb cinv=1p
xtg1 data a clck clckn tgate ctg='cinv/2'
xtg2 q ax clckn clck tgate ctg='cinv/2'
rx ax a 5
xinv1 a qb inv capout=cinv
xinv2 qb q inv capout=cinv
.ends dlatch

.end
```

Figure 26-4: D-Latch Response

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(clck) data=v(d) q=v(q)
```

Waveforms

```
vdata d 0 pulse(0,5 2n,1n,1n 28n,50n)
vclk 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 clckn clck 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

Subcircuit Definition

```
.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

```
.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
```

```
+ 5.0            2.3k
.ends nmos
```

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 Vout/I(mn) where mn is the pulldown transistor. RP is calculated as (VCC-Vout)/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

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.

Figure 26-10: 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
```

The triple range DC sweep allows coarse grid before and after:

```
* 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

x1 in out inv
```

```
.macro inv in out
mn 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
******
volt           rn

0.             3.312e+14
285.71429m     317.3503g
571.42857m     304.0682x
857.14286m     1.1222x
1.14286        107.6844k
1.42857        32.1373k
1.71429        14.6984k
2.00000        7.7108k
2.10000        5.8210k
2.12105        5.1059k
2.14211        3.2036k
2.16316        1.6906k
2.18421        1.4421k
2.20526        1.3255k
2.22632        1.2470k
2.24737        1.1860k
2.26842        1.1360k
2.28947        1.0935k
2.31053        1.0565k
2.33158        1.0238k
2.35263        994.3804
2.37368        967.7559
2.39474        943.4266
2.41579        921.0413
2.43684        900.3251
2.45789        881.0585
2.47895        863.0632
2.50000        846.1922
2.75000        701.5119
3.20000        560.6908
3.65000        479.8893
4.10000        426.4486
```

```
4.55000          387.7524
5.00000          357.4228
```

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 120N
.PROBE V(out) V(outm)
VC VCC 0 VCC
VCCM VCCM 0 VCC
X1  IN  OUT  INV
X1M 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
.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

```
*                                  %NORM-SEN    %CHANGE
.PARAM TDELAY        =   1.3251N $   37.6164   -48.6429U
.PARAM CAPL          = 390.2613F $   37.2396    60.2596U
.PARAM CAPH          = 364.2716F $   25.1440    62.1922U
```

Optimize Results Measure Names and Values

```
* RISEX           =   2.7018N
* FALLX           =   2.5388N
* PROPFX          =   2.0738N
* PROPRX          =   2.1107N
```

Figure 26-12: CMOS Inverter Response

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 DIFSIZ=.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
```

```
+                      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
OTIMIZATION 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:

$$Vout = -\frac{gain}{Ri \cdot Ci} \cdot \int_0^t Vin \cdot dt + Vout(0)$$

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
```

Figure 26-15: Response of Integrator to a Triangle Waveform

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:

$$Vout = -gain \cdot Rd \cdot Cd \cdot \frac{d}{dt}Vin$$

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 1e12
*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, $Rin = n^2 \cdot RL$.

```
MATCHING IMPEDANCE BY USING IDEAL TRANSFORMER
E   OUT 0 TRANSFORMER  IN 0   10
RL OUT 0 8
VIN IN 0 1
.OP
.END
```

Figure 26-18: Ideal Transformer Example

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=lx0(g) gd=lx2(g)
g 1 0 pwl(1) 1 0 delta=delta
+ -50mv,-5ma 50mv,5ma 200mv,1ma 400mv,.05ma
+ 500mv,2ma 600mv,15ma
.end
```

Figure 26-19: Tunnel Diode Characteristic

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 modelled 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
vdum 2 1 0
```

```
.dc iscr 0 1u 1n
.probe vscr=lx0(h) transr=lx3(h)
h 1 0 pwl(1) vdum -5na,-2v 5na,2v 15na,.1v 1ua,.3v 10ua,.5
+ delta=5na
.end
```

Figure 26-20: Silicon Controlled Rectifier

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
```

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 in- in+ out vcc vee modelname AV=val
```

COMP=1 (external compensation)

The syntax is:
```
xa1 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 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.

Names (Alias)	Units	Default	Description
AV (AVD)	volt/ volt		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.
AV1K	volt/ volt		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.

Names (Alias)	Units	Default	Description
C2	farad		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.
CMRR	volt/ volt		Common mode rejection ratio. This is usually between 80 and 110 dB. This can be entered as 100 dB or as 100000.
COMP			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
DEF			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)

Names (Alias)	Units	Default	Description
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.

Names (Alias)	Units	Default	Description
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.

Names (Alias)	Units	Default	Description
RAC (r0ac, roac)	ohm		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.
ROUT	ohm		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.
SRNEG *(SRN)*	volt		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.
SRPOS (SRP)	volt		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.
TEMP	°C		Temperature in degrees Celsius. This usually is set to the temperature at which the model parameters were measured, which typically is 25 °C.
VCC	volt		Positive power supply reference voltage for VOPOS. The amplifier VOPOS was measured with respect to VCC.

Names (Alias)	Units	Default	Description
VEE	volt		Negative power supply voltage. The amplifier VONEG was measured with respect to VCC.
VONEG (*VON)*	volt		Maximum negative output voltage. This is less than VEE (the negative power-supply voltage) by the internal voltage drop.
VOPOS (VOP)	volt		Maximum positive output voltage. This is less than VCC, the positive power supply voltage, by the internal voltage drop.
VOS	volt		Input offset voltage. This is the voltage required between the input differential transistors to zero the output voltage. This is generally a fundamental electrical characteristic. Typical values for bipolar amplifiers are in the range 0.1 mV to 10 mV. VOS is measured in volts. VOS can cause a failure to converge for some amplifiers. If this occurs, try setting VOS to 0 or use the initial conditions described above for convergence.

Op-Amp Model Parameter Defaults

Parameter	Description	Defaults		
		DEF=0	DEF=1	DEF=2
AV	Amplifier voltage gain	160k	417k	200k
AV1K	Amplifier voltage gain at 1 kHz	-	1.2 k	3 k
C2	Feedback capacitance	30 p	30 p	10 p
CMRR	Common mode rejection ratio	96 db 63.1k	106 db 199.5k	90 db 31.63k
COMP	Compensation level selector	0	0	0

Parameter	Description	Defaults		
		DEF=0	DEF=1	DEF=2
DEF	Default level selector	0	1	2
DELPHS	Delta phase at unity gain	25°	17°	52°
DIS	Diode saturation current	8e-16	8e-16	8e-16
FREQ	Unity gain frequency	600 k	-	-
IB	Input bias current	30 n	250 n	40 n
IBOS	Input bias offset current	1.5 n	0.7 n	5 n
ISC	Output short circuit current	25 mA	25 mA	25 mA
LEVIN	Input circuit level selector	1	1	1
LEVOUT	Output circuit level selector	1	1	1
MANU	Manufacturer's name	-	-	-
PWR	Power dissipation	72 mW	60 mW	50 mW
RAC	AC output resistance	0	75	70
ROUT	DC output resistance	200	550	100
SRPOS	Positive output slew rate	450 k	1 meg	1 meg
SRNEG	Negative output slew rate	450 k	800 k	800 k
TEMP	Temperature of model	25 deg	25 deg	25 deg
VCC	Positive supply voltage for VOPOS	20	15	15
VEE	Negative supply voltage for VONEG	-20	-15	-15
VONEG	Maximum negative output	-14	-14	-14
VOPOS	Maximum positive output	14	14	14
VOS	Input offset voltage	0	0.3 m	0.5 m

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

```
$$ FILE  ALM124.SP
.OPTION NOMOD AUTOSTOP  SEARCH=' '
.OP VOL
.AC DEC 10 1HZ  10MEGHZ
.MODEL PLOTDB   PLOT XSCAL=2 YSCAL=3
.MODEL PLOTLOGX PLOT XSCAL=2
```

```
.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=     .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
*
```

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.
```

```
   NODE     =VOLTAGE        NODE    =VOLTAGE        NODE   =VOLTAGE
 + 0:IN      =    0.      0:IN+  =-433.4007U 0:IN-  =   3.3021M
 + 0:OUT0    =   7.0678M 0:VCC  =  15.0000  0:VEE  = -15.0000

 unitfreq    = 907.855K  TARG   = 907.856K  TRIG   =   1.000
 PHASEMARGIN =  66.403
 gain(db)    =  99.663  AT      =   1.000
 FROM        =   1.000  TO      =  10.000X
 gain(mag)   =  96.192K AT      =   1.000
 FROM        =   1.000  TO      =  10.000X
 srpos       =   2.030U TARG    =  35.471U  TRIG   =  33.442U
 srneg       =   1.990U TARG    =   7.064U  TRIG   =   5.074U
```

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

$I(g) = F(Vin+ - Vin-)$
$e = V(out1)$
$eo = V(out2)$
$V(out) = F(I(h))$

Example

```
0p_amp.sp --- operational amplifier
*
.options post=2
.tran .001ms 2ms
```

```
.ac dec 10 .1hz 10me'
*.graph tran vout=v(output)
*.graph tran vin=v(input)
*.graph ac model=grap voutdb=vdb(output)
*.graph ac model=grap vphase=vp(output)
.probe tran vout=v(output) vin=v(input)
.probe ac voutdb=vdb(output) vphase=vp(output)
.model grap plot xscal=2
```

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 pwl(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
```

```
ro out1 out 75
vdum out dum 0
h dum 0 pwl(1) vdum 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
.end
```

Figure 26-25: AC Analysis Response

Figure 26-26: Transient Analysis Response[1]·

Inverting Comparator with Hysteresis

An inverting comparator is modelled by a piecewise linear VCVS.

Figure 26-27: Inverting Comparator with Hysteresis

Two reference voltages corresponding to volow and vohigh of Ecomp characteristic are:

$$Vreflow \ = \ \frac{Volow \cdot Rb}{Rb + Rf}$$

$$Vrefhigh \ = \ \frac{Vohigh \cdot Rb}{Rb + Rf}$$

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 hysterisis
.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
```

Figure 26-28: Response of Comparator

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

```
vcob.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
```

Figure 26-29: Voltage Controlled Oscillator Response

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 t2. The frequency of this LC circuit is given by:

$$freq = \frac{1}{6.28 \cdot \sqrt{L \cdot C}}$$

At time t2, 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:

$$E = (0.5 \cdot C \cdot V^2) + (0.5 \cdot L \cdot I^2)$$

$$E = 0.5 \cdot C \cdot Vm^2, I = 0$$

$$E = 0.5 \cdot L \cdot Im^2, V = 0$$

Assuming at time t2, when V=0, C changes to $A \cdot C$, then:

$$0.5 \cdot L \cdot Im^2 = 0.5 \cdot Vm^2 = 0.5 \cdot (A \cdot C) \cdot Vm'^2$$

and from the above equation:

$$Vm' = \frac{Vm}{\sqrt{A}}$$

$$Qm' = \sqrt{A} \cdot Vm$$

The second condition to consider is when V=Vin, C changes to $A \cdot C$. In this case:

$$Qm = Qm'$$

$$C \cdot Vm = A \cdot C \cdot Vm'$$

$$Vm' = \frac{Vm}{A}$$

Therefore, the voltage amplitude is modified between Vm/sqrt(A) and Vm/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
```

```
*
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

Example

```
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
```

```
    xq3 q4 n4 q3 nand2
    xq4 q3 nv q4 nand2
    xnv v d1 nv nand2
    xu1 nr q1 n4 u1 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)
$ = 0.63, v_capture ~= 0.100
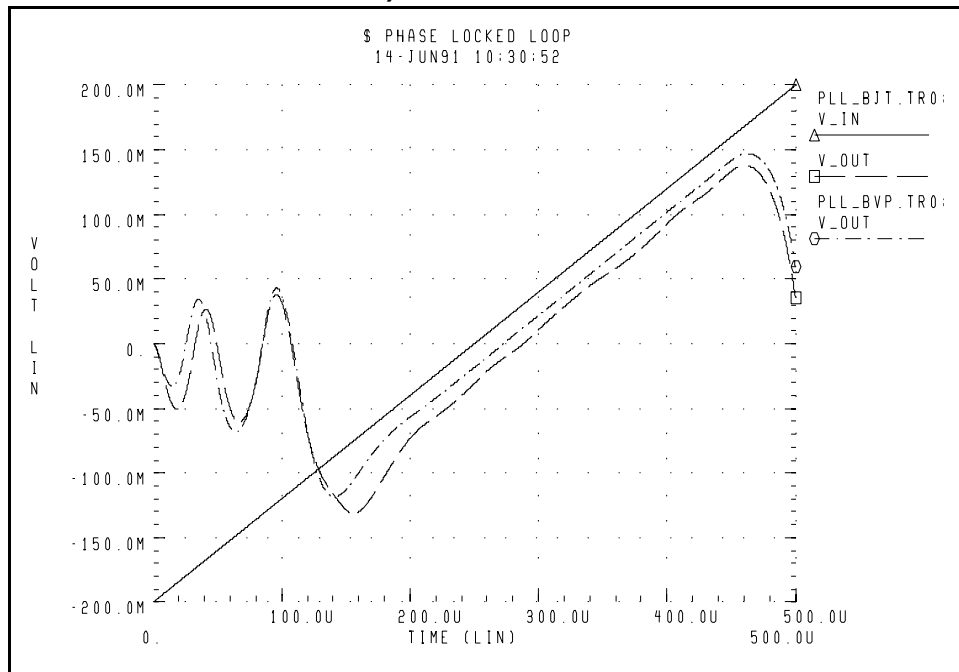
*.ic v(out)=0 v(fin)=0
.tran .2u 500u
.option delmax=0.01u interp
.probe v_in=v(inc,0) v_out=v(out,outb)
.probe v(in) v(osc) v(mout) v(out)
```

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
```

```
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
```

```
.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 pwl 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
xvco1 e eb osc oscb 0 vee vco1
.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 moutb eb 10k
cfb eb 0 1000p
```

Final Output
```
rout out e 100k
cout out 0 100p
routb outb eb 100k
coutb outb 0 100p
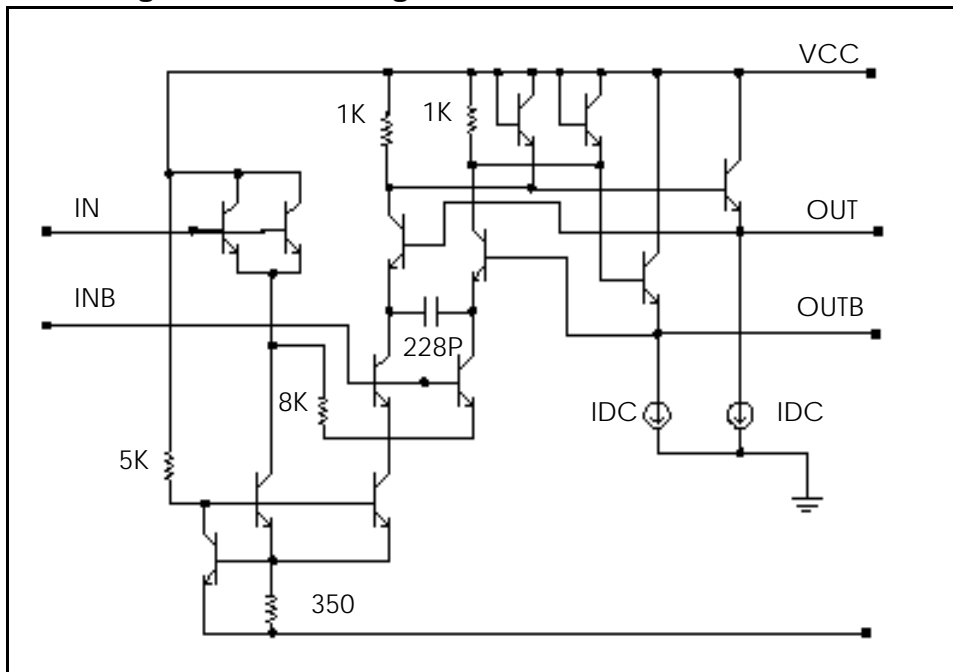
.macro vco in inb out outb f0=100k kf=50k phi=0.0 out_off=0.0
out_amp=1.0
gs 0 s poly(2) c 0 in inb 0 '6.2832e-9*f0' 0 0 '6.2832e-9*kf'
gc c 0 poly(2) s 0 in inb 0 '6.2832e-9*f0' 0 0 '6.2832e-9*kf'
cs s 0 1e-9
cc c 0 1e-9
e1 s_clip 0 pwl(1) s 0 -0.1,-0.1 0.1,0.1
e out 0 s_clip 0 out_off '10*out_amp'
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 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

.macro vco1 in inb e7 e8 vcc vee vco_cap=228.5p
qout vcc vcc b7 npn1
```

```
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 2 c0 b0 npn1 $ ic=2*i0
q16 3 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
+ tbf1=8e-03

.macro pd1 in inb in2 in2b out outb vcc vee
rl 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
```

```
q1 c1 in2 e npn1
q2 c2 in2b e npn1
ie e 0 0.5m
c1 n1 0 1p
c1b n1b 0 1p
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

References

1. Chua & Lin. *Computer Aided Analysis of Electronic Circuit*s. Englewood Cliffs: Prentice-Hall, 1975, page 117. See also "SPICE2 Application Notes for Dependent Sources," by Bert Epler, *IEEE Circuits & Devices Magazine*, September 1987.

Avant!

<div align="right">

Chapter 27

Performing Pole/Zero Analysis

</div>

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:

$$H(s) = \frac{N(s)}{D(s)} = \frac{a_0 s^m + a_1 \cdot s^{(m-1)} + \ldots + a_m}{b_0 s^n + b_1 \cdot s^{(n-1)} + \ldots + b_n}$$

In the factorized form, the general function is:

$$H(s) = \frac{a_0}{b_0} \cdot \frac{(s + z_1)(s + z_2) \ldots (s + z_i) \ldots (s + z_m)}{(s + p_1)(s + p_2) \ldots (s + p_j) \ldots (s + p_m)}$$

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[7] 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 (X0R, X0I) 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 (X1R, X1I) and (X2R, X2I) may be orders of magnitude larger than (X0R, X0I). If there are poles or zeros at high frequencies, X1I and X2I 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
```

```
.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.

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. Default1.0e-6. RITOL is used as follows:

If $\left| X_{imag} \right| \leq RITOL \cdot \left| X_{real} \right|$, then $X_{imag} = 0$

If $\left| X_{real} \right| \leq RITOL \cdot \left| X_{imag} \right|$, then $X_{real} = 0$

(X0R,X0I) *(x1R,X1I)* *(X2R,X21)*	the three complex starting trial points in the Muller algorithm for pole/zero analysis. Defaults:

X0R=-1.23456e6   X0I=0.0
X1R=1.23456e5   X1I=0.0
X2R=+1.23456e6   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[8]. This file can be found in *$installdir/demo/hspice/filters/ flp5th.sp*.

Fifth-Order Low-Pass Filter HSPICE File

```
*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
```

```
.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

Poles (rad/sec)		Poles (hertz)	
Real	**Imag**	**Real**	**Imag**
-6.948473e-02	-4.671778e-01	-1.105884e-02	-7.435365e-02
-6.948473e-02	4.671778e-01	-1.105884e-02	7.435365e-02
-1.182742e-01	-8.914907e-01	-1.882392e-02	-1.418852e-01
-1.182742e-01	8.914907e-01	-1.882392e-02	1.418852e-01
-5.450890e-01	0.000000e+00	-8.675361e-02	0.000000e+00

Table 27-1: Pole/Zero Analysis Results for Low-Pass Filter

Zeros (rad/sec)		Zeros (hertz)	
Real	**Imag**	**Real**	**Imag**
0.000000e+00	-1.286180e+00	0.000000e+00	-2.047019e-01
0.000000e+00	-5.129892e-01	0.000000e+00	-8.164476e-02
0.000000e+00	5.129892e-01	0.000000e+00	8.164476e-02
0.000000e+00	1.286180e+00	0.000000e+00	2.047019e-01
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[9]. 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

Poles (rad/sec)		Poles (hertz)	
Real	**Imag**	**Real**	**Imag**
-5.003939e-02	9.987214e-01	-7.964016e-03	1.589515e-01
-5.003939e-02	-9.987214e-01	-7.964016e-03	-1.589515e-01
-1.414227e+00	0.000000e+00	-2.250812e-01	0.000000e+00
Zeros (rad/sec)		**Zeros (hertz)**	
Real	**Imag**	**Real**	**Imag**
0.000000e+00	-1.414227e+00	0.000000e+00	-2.250812e-01
0.000000e+00	1.414227e+00	0.000000e+00	2.250812e-01
-1.414227e+00	0.000000e+00	-2.250812e-01	0.000000e+00
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.[10] 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 = V(10) / VIN
*  = (S**4) / ((S**2 + 0.7653*S + 1) * (S**2 + 1.8477*S + 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

Table 27-3: Pole/Zero Analysis Results for High-Pass Butterworth Filter

Poles (rad/sec)		Poles (hertz)	
Real	**Imag**	**Real**	**Imag**
-3.827019e-01	-9.240160e-01	-6.090889e-02	1.470617e-01
-3.827019e-01	9.240160e-01	-6.090890e-02	-1.470617e-01
-9.237875e-01	3.828878e-01	-1.470254e-01	6.093849e-02
-9.237875e-01	-3.828878e-01	-1.470254e-01	-6.093849e-02

Zeros (rad/sec)		Zeros (hertz)	
Real	**Imag**	**Real**	**Imag**
0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
Constant Factor = 1.000000e+00			

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)
```

```
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

Table 27-4: Pole/Zero Analysis Results for CMOS Differential Amplifier

Poles (rad/sec)		Poles (hertz)	
Real	**Imag**	**Real**	**Imag**
-1.798766e+06	0.000000e+00	-2.862825e+05	0.000000e+00
-1.126313e+08	-6.822910e+07	-1.792583e+07	-1.085900e+07
-1.126313e+08	6.822910e+07	-1.792583e+07	1.085900e+07

Zeros (rad/sec		Zeros (hertz)	
Real	**Imag**	**Real**	**Imag**
-1.315386e+08	7.679633e+07	-2.093502e+07	1.222251e+07
-1.315386e+08	-7.679633e+07	-2.093502e+07	-1.222251e+07
7.999613e+08	0.000000e+00	1.273178e+08	0.000000e+00
Constant Factor = 3.103553e-01			

Example 5 – Simple Amplifier

The following is an HSPICE input file for pole/zero analysis of an equivalent
circuit of a simple amplifier with RS¼=RPI=RL=1000 ohms, gm=0.04 mho,
CMU=1.0e-11 farad, and CPI¼=1.0e-9 farad[11]. 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)
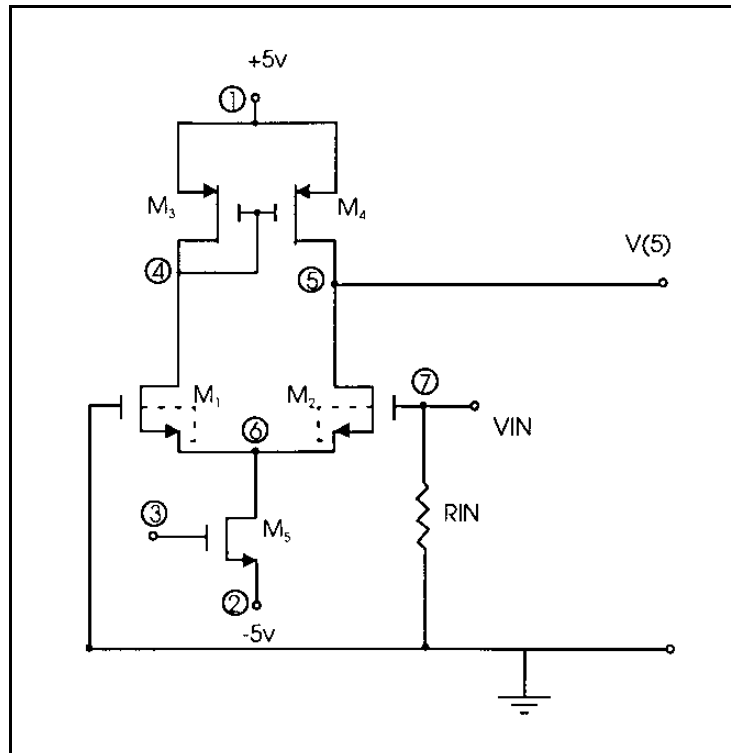
.PZ   V(3)   VIN
RS   1  2  1K
RPI  2  0  1K
```

```
RL    3   0   1K
GMU   3   0   2   0   0.04
CPI   2   0   1NF
CMU   2   3   10PF
VIN   1   0   1
.END
```

Figure 27-6: Simple Amplifier

Table 27-5: Pole/Zero Analysis Results for Amplifier

Poles (rad/sec)		Poles (hertz)	
Real	Imag	Real	Imag
-1.412555+06	0.000000e+00	-2.248151e+05	0.000000e+00
-1.415874+08	0.000000e+00	-2.253434e+07	0.000000e+00
Zeros (rad/sec)		Zeros (hertz)	
Real	Imag	Real	Imag
4.000000e+09	0.000000e+00	6.366198e+08	0.000000e+00
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[12] 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
.PZ  V(OUT) VIN
.AC  DEC  50  .1K  100K
.OPTIONS  POST  DCSTEP=1E3  X0R=-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

Poles (rad/sec)		Poles (hertz)	
Real	**Imag**	**Real**	**Imag**
-4.505616e+02	-2.210451e+04	-7.170911e+01	-3.518042e+03
-4.505616e+02	2.210451e+04	-7.170911e+01	3.518042e+03
-1.835284e+03	2.148369e+04	-2.920944e+02	3.419236e+03
-1.835284e+03	-2.148369e+04	-2.920944e+02	-3.419236e+03
-4.580172e+03	1.944579e+04	-7.289571e+02	3.094894e+03
-4.580172e+03	-1.944579e+04	-7.289571e+02	-3.094894e+03
-9.701962e+03	1.304893e+04	-1.544115e+03	2.076802e+03
-9.701962e+03	-1.304893e+04	-1.544115e+03	-2.076802e+03
-1.353908e+04	0.000000e+00	-2.154811e+03	0.000000e+00
-3.668995e+06	-3.669793e+06	-5.839386e+05	-5.840657e+05
-3.668995e+06	3.669793e+06	-5.839386e+05	5.840657e+05
-3.676439e+06	-3.676184e+06	-5.851234e+05	-5.850828e+05
-3.676439e+06	3.676184e+06	-5.851234e+05	5.850828e+05
-3.687870e+06	3.687391e+06	-5.869428e+05	5.868665e+05
-3.687870e+06	-3.687391e+06	-5.869428e+05	-5.868665e+05
-3.695817e+06	-3.695434e+06	-5.882075e+05	-5.881466e+05
-3.695817e+06	+3.695434e+06	-5.882075e+05	5.881466e+05

Zeroes (rad/sec)		Zeroes (hertz)	
Real	**Imag**	**Real**	**Imag**
-3.220467e-02	-2.516970e+04	-5.125532e-03	-4.005882e+03
-3.220467e-02	2.516970e+04	-5.125533e-03	4.005882e+03
2.524420e-01	-2.383956e+04	4.017739e-02	-3.794184e+03
2.524420e-01	2.383956e+04	4.017739e-02	3.794184e+03

Table 27-6: Pole/Zero Analysis Results for the Active Low-Pass Filter

1.637164e+00	2.981593e+04	2.605627e-01	4.745353e+03
1.637164e+00	-2.981593e+04	2.605627e-01	-4.745353e+03
4.888484e+00	4.852376e+04	7.780265e-01	7.722796e+03
4.888484e+00	-4.852376e+04	7.780265e-01	-7.722796e+03
-3.641366e+06	-3.642634e+06	-5.795413e+05	-5.797432e+05
-3.641366e+06	3.642634e+06	-5.795413e+05	5.797432e+05
-3.649508e+06	-3.649610e+06	-5.808372e+05	-5.808535e+05
-3.649508e+06	3.649610e+06	-5.808372e+05	5.808535e+05
-3.683700e+06	3.683412e+06	-5.862790e+05	5.862333e+05
-3.683700e+06	-3.683412e+06	-5.862790e+05	-5.862333e+05
-3.693882e+06	3.693739e+06	5.878995e+05	5.878768e+05
-3.693882e+06	-3.693739e+06	-5.878995e+05	-5.878768e+05
Constant Factor = 4.451586e+02			

Figure 27-10: 9th Order Low-Pass Filter Response

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.

1. Desoer, Charles A. and Kuh, Ernest S. *Basic Circuit Theory*. New York: McGraw-Hill.1969. Chapter 15.

2. Van Valkenburg, M. E. *Network Analysis.* Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1974, chapters 10 & 13.

3. R.H. Canon, Jr. *Dynamics of Physical Systems*. New York: McGraw-Hill, 1967. This text describes electrical, mechanical, pneumatic, hydraulic, and mixed systems.

4. B.C. Kuo. *Automatic Control Systems*. Englewood Cliffs, New Jersey: Prentice-Hall, 1975. This source discusses control system design, and provides background material on physical modeling.

5. L.T. Pillage, and R.A. Rohrer. *Asymptotic Waveform Evaluation for Timing Analysis*, IEEE Trans CAD. Apr. 1990, pp. 352 - 366. This paper is a good references on interconnect transfer function modeling which deals with transfer function extraction for timing analysis.

6. S. Lin, and E.S. Kuh. *Transient Simulation of Lossy Interconnects Based on the Recursive Convolution Formulation*, *IEEE Trans CAS*. Nov. 1992, pp. 879 - 892. This paper provides another source of interconnect transfer function modeling.

7. Muller, D. E., *A Method for Solving Algebraic Equations Using a Computer, Mathematical Tables and Other Aids to Computation (MTAC).* 1956, Vol. 10,. pp. 208-215.

8. Temes, Gabor C. and Mitra, Sanjit K. *Modern Filter Theory And Design.* J. Wiley, 1973, page 74.

9. Temes, Gabor C. and Lapatra, Jack W. *Circuit Synthesis And Design*, McGraw-Hill. 1977, page 301, example 7-6.

10. Temes, Gabor C. and Mitra, Sanjit K., *Modern Filter Theory And Design.* J. Wiley, 1973, page 348, example 8-3.

11. Desoer, Charles A. and Kuh, Ernest S. *Basic Circuit Theory.* McGraw-Hill, 1969, page 613, example 3.

12. Vlach, Jiri and Singhal, Kishore. *Computer Methods For Circuit Analysis and Design.* Van Nostrand Reinhold Co., 1983, pages 142, 494-496.

Avant!

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

- 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

Window	Equation	Highest Side-Lobe (dB)	Side-Lobe Roll-Off (dB/octave)	3.0-dB Bandwidth (1.0/T)	Worst Case Process Loss (dB)
Rectangular	$W(n)=1$, $0 \le n < \text{NP}^\dagger$	-13	-6	0.89	3.92
Bartlett	$W(n)=2n/(\text{NP-1})$, $0 \le n \le (\text{NP}/2)\text{-}1$ $W(n)=2\text{-}2n/(\text{NP-1})$, $\text{NP}/2 \le n < \text{NP}$	-27	-12	1.28	3.07
Hanning	$W(n)=0.5\text{-}0.5[\cos(2\pi n/(\text{NP-1}))]$, $0 \le n < \text{NP}$	-32	-18	1.44	3.18
Hamming	$W(n)=0.54\text{-}0.46[\cos(2\pi n/(\text{NP-1}))]$, $0 \le n < \text{NP}$	-43	-6	1.30	3.10
Blackman	$W(n)=0.42323$ $-0.49755[\cos(2\pi n/(\text{NP-1}))]$ $+0.07922\cos[\cos(4\pi n/(\text{NP-1}))]$, $0 \le n < \text{NP}$	-58	-18	1.68	3.47
Blackman-Harris	$W(n)=0.35875$ $-0.48829[\cos(2\pi n/(\text{NP-1}))]$ $+0.14128[\cos(4\pi n/(\text{NP-1}))]$ $-0.01168[\cos(6\pi n/(\text{NP-1}))]$, $0 \le n < \text{NP}$	-92	-6	1.90	3.85
Gaussian a=2.5 a=3.0 a=3.5	$W(n)=\exp[-0.5a2(\text{NP}/2\text{-}1\text{-}n)2/(\text{NP})2]$, $0 \le n \le (\text{NP}/2)\text{-}1$ $W(n)=\exp[-0.5a2(n\text{-NP}/2)2/(\text{NP})2]$, $\text{NP}/2 \le n < \text{NP}$	-42 -55 -69	-6 -6 -6	1.33 1.55 1.79	3.14 3.40 3.73
Kaiser-Bessel a=2.0 a=2.5 a=3.0 a=3.5	$W(n)=I0(x2)/I0(x1)$ $x1=pa$ $x2=x1*\text{sqrt}[1\text{-}(2(\text{NP}/2\text{-}1\text{-}n)/\text{NP})2]$, $0 \le n \le (\text{NP}/2)\text{-}1$ $x2=x1*\text{sqrt}[1\text{-}(2(n\text{-NP}/2)/\text{NP})2]$, $\text{NP}/2 \le n < \text{NP}$ I0 is the zero-order modified Bessel function	-46 -57 -69 -82	-6 -6 -6 -6	1.43 1.57 1.71 0.89	3.20 3.38 3.56 3.74

[†]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

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 28-2: .FFT Statement Parameters

Parameter	Default	Description
output_var		Can be any valid output variable, such as voltage, current, or power
START	see Description	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.
FROM	see START	An alias for START in .FFT statements
STOP	see Description	Specifies the end of the output variable waveform to be analyzed. Defaults to the TSTOP value in the .TRAN statement.
TO	see STOP	An alias for STOP in .FFT statements
NP	1024	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.
FORMAT	NORM	Specifies the output format: NORM= normalized magnitude UNORM=unnormalized magnitude

Table 28-2: .FFT Statement Parameters

Parameter	Default	Description
WINDOW	RECT	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
ALFA	3.0	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
FREQ	0.0 (Hz)	Specifies a frequency of interest. If FREQ is nonzero, the output listing is limited to the harmonics of this frequency, based on FMIN and FMAX. The THD for these harmonics also is printed.
FMIN	1.0/T (Hz)	Specifies the minimum frequency for which FFT output is printed in the listing file or which is used in THD calculations. T = (STOP-START)
FMAX	0.5*NP*FMIN (Hz)	Specifies the maximum frequency for which FFT output is printed in the listing file or which is used in THD calculations.

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

frequency      frequency      fft_mag       fft_mag      fft_phase
   index         (hz)          (db)                        (deg)
     2         1.0000k        0.           1.0000        -3.8093m
     4         2.0000k       -125.5914     525.3264n     -5.2406
     6         3.0000k       -106.3740     4.8007u       -98.5448
     8         4.0000k       -113.5753     2.0952u       -5.5966
    10         5.0000k       -112.6689     2.3257u      -103.4041
    12         6.0000k       -118.3365     1.2111u      167.2651
    14         7.0000k       -109.8888     3.2030u      -100.7151
    16         8.0000k       -117.4413     1.3426u      161.1255
```

```
18          9.0000k    -97.5293     13.2903u     70.0515
20         10.0000k   -114.3693      1.9122u    -12.5492
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

Column Heading	Description
Frequency Index	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.
Frequency	The actual frequency associated with the index
fft_mag (dB), fft_mag	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.
fft_phase	The associated phase, in degrees

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:

    frequency increment = 1.0/(STOP - 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 *.ft0*, *.ft1*, ..., *.ftn* 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

$$1) = sa \cdot (offset + \sin(\omega_m(Time - td))) \cdot \sin(\omega_c(Time - td)$$

The preceding equation can be expanded as follows.

$$(1) = (sa \cdot offset \cdot \sin(\omega_c(Time - td)) + 0.5 \cdot sa \cdot \cos((\omega_c - \omega_m)(Time - td)))$$

$$- 0.5 \cdot sa \cdot \cos((\omega_c + \omega_m)(Time - td))$$

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 1

.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                           tnom=  25.000 temp=
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

frequency     frequency     fft_mag      fft_mag     fft_phase
  index         (hz)          (db)                     (deg)
   25         833.3333     -129.4536    336.7584n    -113.0047
   26         866.6667     -143.7912     64.6308n      45.6195
   27         900.0000       -6.0206    500.0008m      35.9963
   28         933.3333     -125.4909    531.4428n     112.6012
   29         966.6667     -142.7650     72.7360n     -32.3152
   30           1.0000k      0.           1.0000      -90.0050
   31           1.0333k   -132.4062    239.7125n      -9.0718
   32           1.0667k   -152.0156     25.0738n       3.4251
   33           1.1000k     -6.0206    499.9989m     143.9933
   34           1.1333k   -147.0134     44.5997n      -3.0046
   35           1.1667k   -147.7864     40.8021n      -4.7543
          ***** 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.

Figure 28-4: AM Modulation

Figure 28-5: AM Modulation Spectrum

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
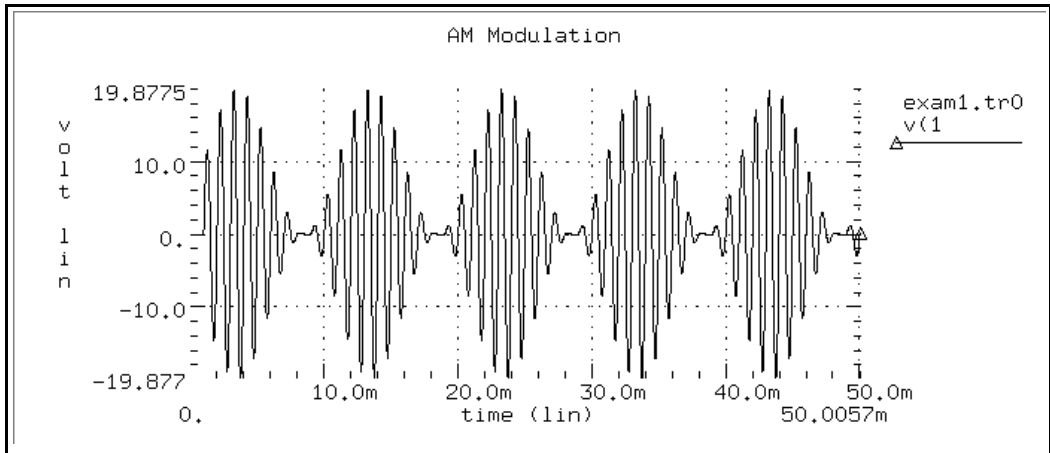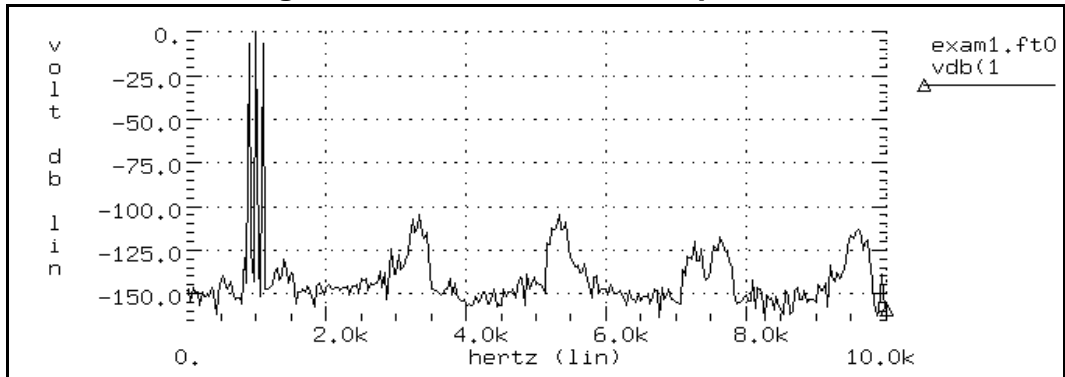
V1 mod1 GND sin(0 5 1K 0 0 0) $ modulating signal
r1 mod1 2 10k
r2 2 3 10k
r3 2 GND 10K
E1 3 GND OPAMP 2 GND $ buffered output of modulating signal
V2 mod2 GND sin(0 5 10K 0 0 0) $ modulated signal
E2 modout GND  vol='(v(3)*v(mod2))/10.0' $ multiply to modulate
V3 8 GND sin(0 5 10K 0 0 0)
E3 demod GND  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)
.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
frequency      frequency     fft_mag      fft_mag      fft_phase
  index          (hz)         (db)                      (deg)
    4           1.0000k       0.           1.0000       35.6762
    8           2.0000k      -26.6737      46.3781m     122.8647
   12           3.0000k      -31.4745      26.6856m     108.1100
   16           4.0000k      -34.4833      18.8728m     103.6867
   20           5.0000k      -36.6608      14.6880m     101.8227
   24           6.0000k      -38.3737      12.0591m     100.9676
   28           7.0000k      -39.7894      10.2455m     100.6167
   32           8.0000k      -40.9976       8.9150m     100.5559
   36           9.0000k      -42.0524       7.8955m     100.6783
   40          10.0000k      -42.9888       7.0886m     100.9240
    total harmonic distortion =     6.2269    percent
******

balanced modulator & demodulator circuit

 ******  fft analysis                          tnom=  25.000 temp=
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
frequency     frequency      fft_mag       fft_mag      fft_phase

  index          (hz)          (db)                       (deg)
    4          1.0000k        0.           1.0000        34.3715
    8          2.0000k       -66.5109     472.5569u     -78.8512
   12          3.0000k       -97.5914      13.1956u     -55.7167
   16          4.0000k      -107.8004       4.0736u     -41.6389
   20          5.0000k      -117.9984       1.2592u     -23.9325
   24          6.0000k      -125.0965     556.1309n      33.3195
   28          7.0000k      -123.6795     654.6722n      74.0461
   32          8.0000k      -122.4362     755.4258n      86.5049
   36          9.0000k      -122.0336     791.2570n      91.6976
   40         10.0000k      -122.0388     790.7840n      94.5380

    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

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
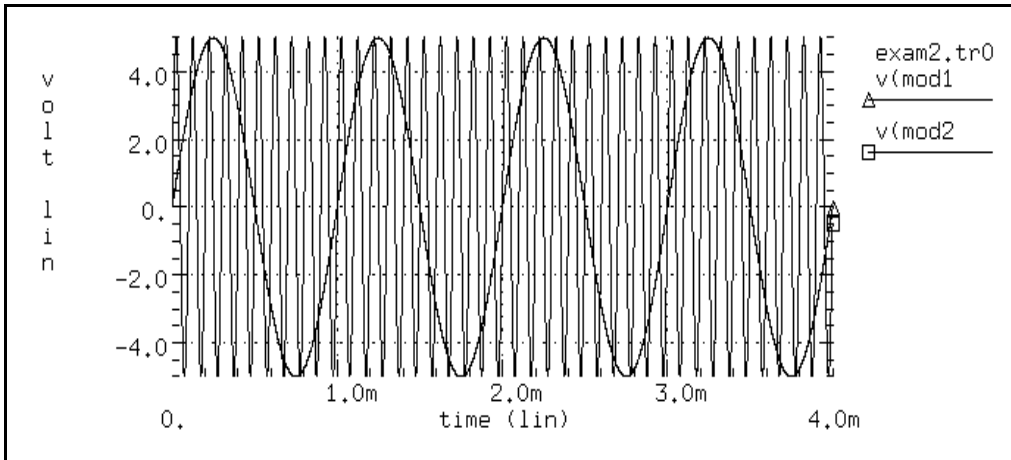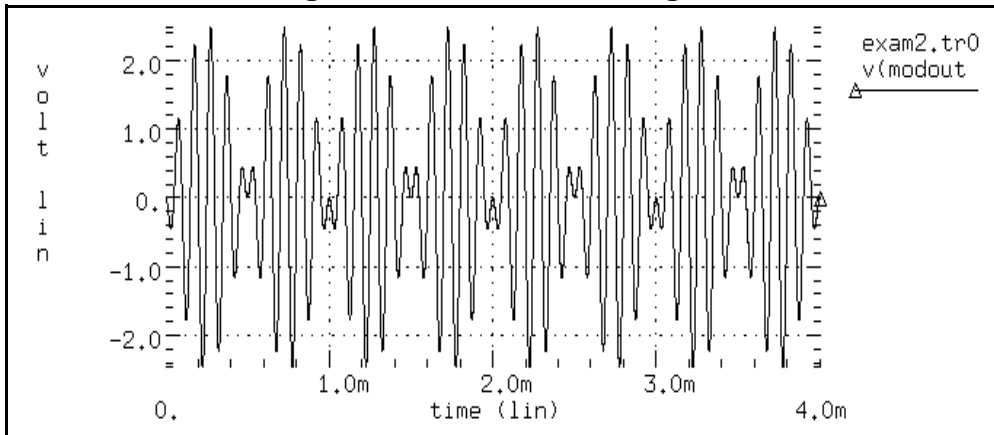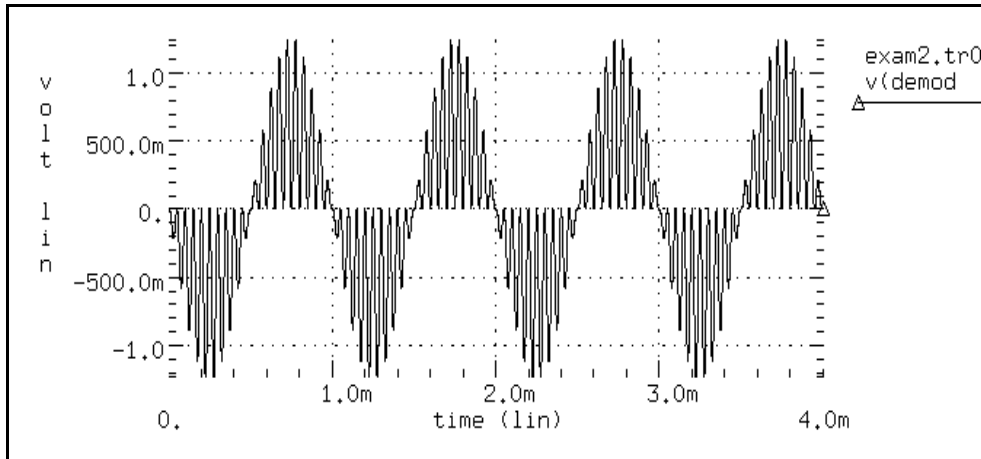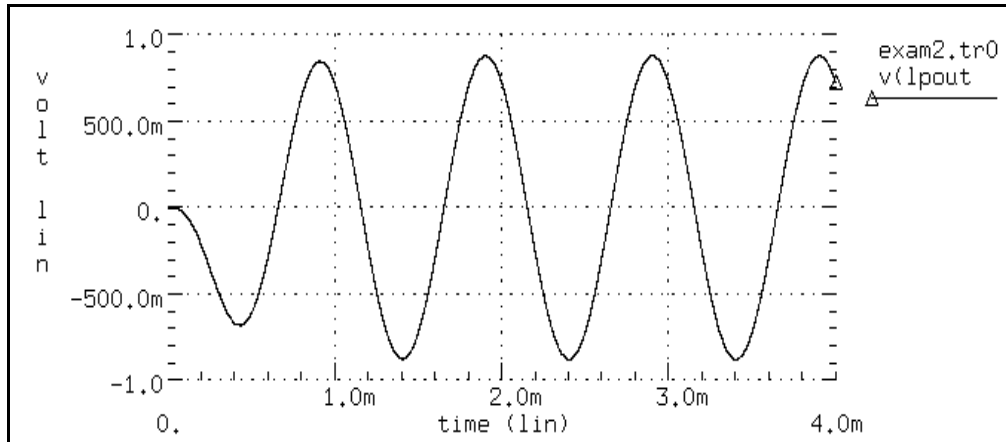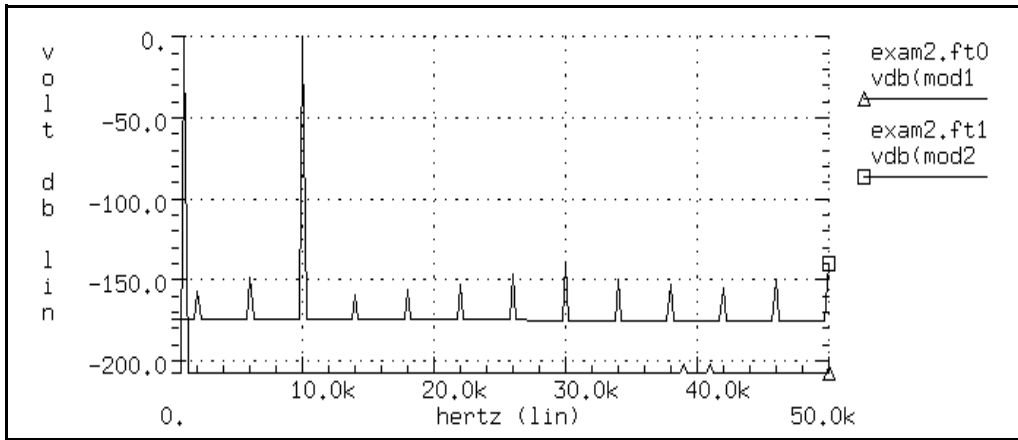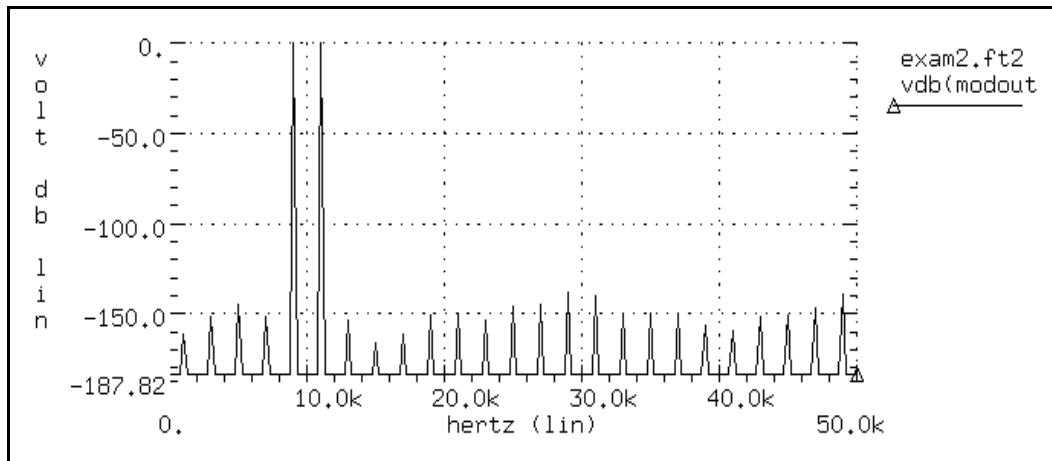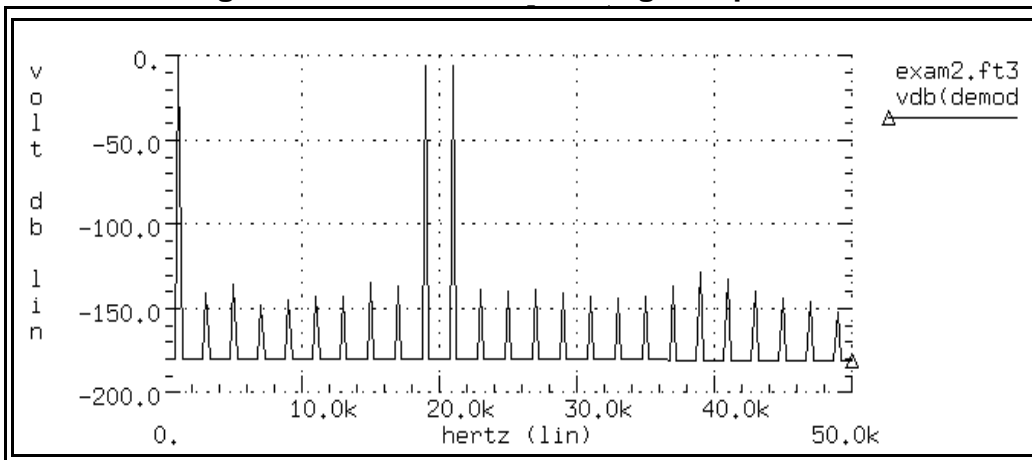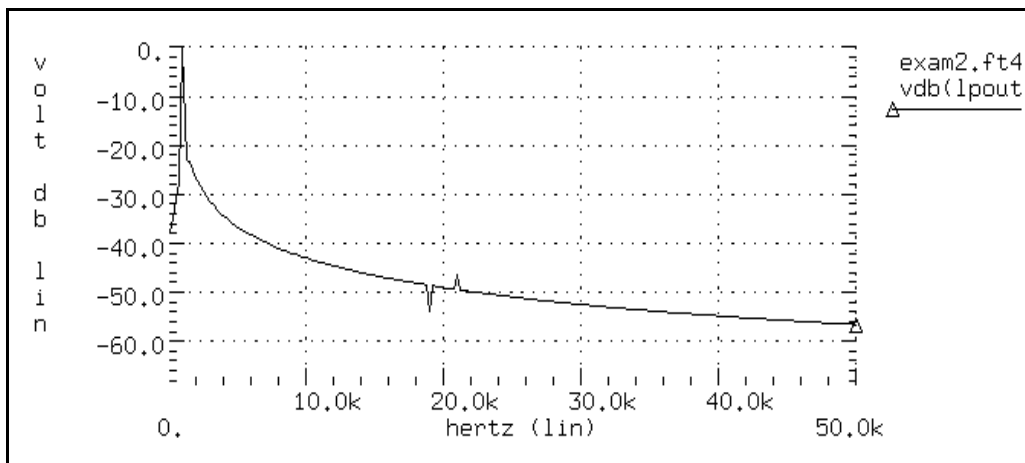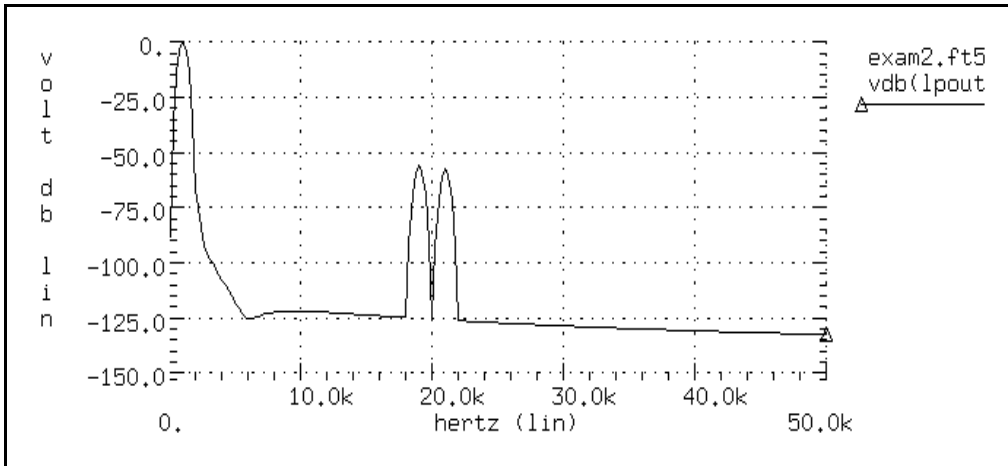
.tran 0.1n 102.4n
.option post probe
.fft v(3)
.fft v(3) window=Bartlett fmin=1.2g fmax=2.2g
.fft v(3) window=hanning fmin=1.2g fmax=2.2g
.fft v(3) window=hamminn fmin=1.2g fmax=2.2g
.fft v(3) window=blackman fmin=1.2g fmax=2.2g
.fft v(3) window=harris fmin=1.2g fmax=2.2g
.fft v(3) window=gaussian fmin=1.2g 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.

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

Figure 28-22: Mixer Output Spectrum, Kaiser-Bessel Window

References

1. For an excellent discussion of DFT windows, see Fredric J. Harris, "On the Use of Windows for Harmonic Analysis with Discrete Fourier Transform", *Proceedings of the IEEE*, Vol. 66, No. 1, Jan. 1978.

Avant!

Chapter 29

Modeling Filters and Networks

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

$$y(k\Delta) \ = \ \Delta \sum_{m=0}^{k} x(m\Delta) \cdot h[(k-m) \cdot \Delta] \ , \ \ k = 0, 1, 2, ...$$

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

$$h(m\Delta) \ = \ \frac{1}{N \cdot \Delta} \sum_{n=0}^{N-1} H(f_n) \cdot e^{\frac{j2\pi nm}{N}} \ , \ \ m = 0, 1, 2, ..., N\text{-}1$$

where N is the number of equally spaced time points and Δ 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

$$f_n \ = \ \frac{n}{N \cdot \Delta} \ , \ \ \ n = 0, 1, 2, ..., N\text{-}1$$

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

Δ 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).

$$\left|H(f_c)\right| = \frac{\left|H_{max}\right|}{1000}$$

Once f_c is selected or calculated, then Δ 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·MAXF/DELF. Then, N is modified to be a power of 2. The effective DELF is determined by 2·MAXF/N to reflect the changes in N.

Laplace Transform

The syntax of the LAPLACE function is:

Transconductance H(s):

```
Gxxx n₊ n₋  LAPLACE in₊ in₋   k₀, k₁, ..., kₙ / d₀,
+ d₁, ..., dₘ
```
<SCALE=val> <TC1=val> <TC2=val> <M=val>

Voltage Gain H(s):

```
Exxx n₊ n₋  LAPLACE in₊ in₋   k₀, k₁, ..., kₙ / d₀,
```

```
    + d₁, ..., dₘ
    <SCALE=val> <TC1=val> <TC2=val>
```

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, ..., d_0, d_1, ..., can be parameterized.

Example

```
Glowpass  0  out  LAPLACE  in  0   1.0 / 1.0  2.0  2.0  1.0
Ehipass   out  0     LAPLACE  in  0   0.0,0.0,0.0,1.0 /
1.0,2.0,2.0,1.0
```

The Glowpass element statement describes a third-order low-pass filter with the transfer function

$$H(s) = \frac{1}{1 + 2s + 2s^2 + s^3}$$

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 α_{z1}, f_{z1}, ..., α_{zn}, f_{zn} / b,
+ α_{p1}, f_{p1}, ..., α_{pm}, f_{pm} <SCALE=val> <TC1=val>
+ <TC2=val> <M=val>

Voltage Gain H(s):

Exxx n_+ n_- POLE in_+ in_- a α_{z1}, f_{z1}, ..., α_{zn}, f_{zn} / b,
+ α_{p1}, f_{p1}, ..., α_{pm}, f_{pm} <SCALE=val> <TC1=val>
+ <TC2=val>

H(s) in terms of poles and zeros is defined by:

$$H(s) = \frac{a \cdot (s + \alpha_{z1} - j2\pi f_{z1})...(s + \alpha_{zn} - j2\pi f_{zn})(s + \alpha_{zn} + j2\pi f_{zn})}{b \cdot (s + \alpha_{p1} - j2\pi f_{p1})...(s + \alpha_{pm} - j2\pi f_{pm})(s + \alpha_{pm} + j2\pi f_{pm})}$$

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, α, 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, 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):

Gxxx $n+$ $n-$ FREQ $in+$ $in-$ f_1, a_1, ϕ_1, ..., f_i, a_i, ϕ_1
+ <DELF=val> <MAXF=val> <SCALE=val> <TC1=val>
+ <TC2=val> <M=val> <LEVEL=val>

Voltage Gain H(s):

Exxx n+ n- FREQ in+ in- f_1, a_1, ϕ_1, ..., f_i, a_i, ϕ_1
+ <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 ϕ_1 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.

DELF, DELTA	Frequency resolution Δf. 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·MAXF/DELF. Since N must be a power of 2, the DELF is adjusted. The default is 1/TSTOP.
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.
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 *M* 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 $1024 \cdot DELF$. Thee 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:

$$SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta$$

> ***Note:*** *Pole/zero analysis is not allowed when the data file contains elements with frequency response specifications. If you include a MAXF=<par> specification in a G or Element Statement, Star-Hspice issues a warning that MAXF is ignored. This is normal.*

Laplace Band-Reject Filter

This example models an active band-reject filter[1] 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:

```
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

```
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

Figure 29-6: Transient Response of Circuit and Functional Element to a Pulse

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, ..., k_n and d_0, ..., 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 *j*2π*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

f(t), t>0	Source Type	$\mathcal{L}\{\,f(t)\,\} = F(s)$
$\delta(t)$	impulse	1
$u(t)$	step	$\dfrac{1}{s}$
t	ramp	$\dfrac{1}{s^2}$
e^{-at}	exponential	$\dfrac{1}{s+a}$
$\sin \omega t$	sine	$\dfrac{\omega}{s^2 + \omega^2}$
$\cos \omega t$	cosine	$\dfrac{s}{s^2 + \omega^2}$

Table 29-1: Laplace Transforms for Common Source Functions

f(t), t>0	Source Type	L { f(t) }= F(s)
$\sin(\omega t + \theta)$	sine	$\dfrac{s\sin(\theta) + \omega\cos(\theta)}{s^2 + \omega^2}$
$\cos(\omega t + \theta)$	cosine	$\dfrac{s\cos(\theta) - \omega\sin(\theta)}{s^2 + \omega^2}$
$\sinh \omega t$	hyperbolic sine	$\dfrac{\omega}{s^2 - \omega^2}$
$\cosh \omega t$	hyperbolic cosine	$\dfrac{s}{s^2 - \omega^2}$
te^{-at}	damped ramp	$\dfrac{1}{(s+a)^2}$
$e^{-at}\sin \omega t$	damped sine	$\dfrac{\omega}{(s+a)^2 + \omega^2}$
$e^{-at}\cos \omega t$	damped cosine	$\dfrac{s+a}{(s+a)^2 + \omega^2}$

Table 29-2: Laplace Transforms for Common Operations

f(t)	L { f(t) } = F(s)
$Kf(t)$	$KF(s)$
$f_1(t) + f_2(t) - f_3(t) + \ldots$	$F_1(s) + F_2(s) - F_3(s) + \ldots$
$\dfrac{d}{dt}f(t)$	$sF(s) - f(0-\ )$

Table 29-2: Laplace Transforms for Common Operations

f(t)	$L\{f(t)\} = F(s)$
$\dfrac{d^2}{dt^2}f(t)$	$s^2 F(s) - sf(0) - \dfrac{d}{dt}f(0-\ )$
$\dfrac{d^n}{dt^n}f(t)$	$s^n F(s) - s^{n-1}f(0) - s^{n-2}\dfrac{d}{dt}f(0)$
$\displaystyle\int_{-\infty}^{t} f(t)\,dt$	$\dfrac{F(s)}{s} + \dfrac{f^{-1}(0)}{s}$
$f(t-a)u(t-a),\ a>0$ (*u* is the step function)	$e^{-as}F(s)$
$e^{-at}f(t)$	$F(s+a)$
$f(at),\ a>0$	$\dfrac{1}{a}F\left(\dfrac{s}{a}\right)$
$tf(t)$	$-\dfrac{d}{ds}(F(s))$
$t^n f(t)$	$-(1)^n \dfrac{d^n}{ds^n}F(s)$
$\dfrac{f(t)}{t}$	$\displaystyle\int_{s}^{\infty} F(u)\,du$　　(*u* is the step function)
$f(t-t_1)$	$e^{-t_1 s}F(s)$

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 29-7, convert the circuit to its equivalent *s*-domain circuit and solve for v_o / v_g.

Figure 29-7: LAPLACE Example 1 Circuit

Use transforms from Table 29-2 to convert the inductor, capacitor, and resistors. $L\{f(t)\}$ represents the Laplace transform of $f(t)$:

$$L\left\{L\frac{d}{dt}f(t)\right\} = L \cdot (sF(s) - f(0)) = 50 \times 10^{-3} \cdot (s - 0) = 0.05s$$

$$L\left\{\frac{1}{C}\int_0^t f(t)d\tau\right\} = \frac{1}{C} \cdot \left(\frac{F(s)}{s} + \frac{f^{-1}(0)}{s}\right) = \frac{1}{10^{-6}} \cdot \left(\frac{1}{s} + 0\right) = \frac{10^6}{s}$$

$$L\{R1 \cdot f(t)\} = R1 \cdot F(s) = R1 = 1000 \ \Omega$$

$$L\{R2 \cdot f(t)\} = R2 \cdot F(s) = R2 = 250 \ \Omega$$

To convert the voltage source to the *s*-domain, use the sin ωt transform from Table 29-1:

$$L\{2\sin 3t\} = 2 \cdot \frac{3}{s^2 + 3^2} = \frac{6}{s^2 + 9}$$

Figure 29-8 displays the *s*-domain equivalent circuit.

Figure 29-8: 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:

$$H(s) = \frac{k_0 + k_1 s + \ldots + k_n s^n}{d_0 + d_1 s + \ldots + d_m s^m}$$

The coefficients from the voltage gain transfer function above are:

$k_0 = 5x10^6$ $k_1 = 1000$

$d_0 = 25x10^6$ $d_1 = 6000$ $d_2 = 1$

Using these coefficients, a Star-Hspice Laplace modeling function call for the voltage gain transfer function of the circuit in Figure 29-7 is:

```
Eexample1 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: $V_{\text{out}} = ks V_{\text{in}}$

G Element: $I_{\text{out}} = ks V_{\text{in}}$

In the time domain:

E Element: $v_{\text{out}} = k\dfrac{dV_{\text{in}}}{dt}$

G Element: $i_{\text{out}} = k\dfrac{dV_{\text{in}}}{dt}$

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: $\quad V_{out} = \frac{k}{s} V_{in}$

G Element: $\quad I_{out} = \frac{k}{s} V_{in}$

In the time domain:

E Element: $v_{\text{out}} = k \int V_{\text{in}} dt$

G Element: $i_{\text{out}} = k \int V_{\text{in}} dt$

For an integrator, the voltage gain transfer function is:

$$H(s) = \frac{V_{\text{out}}}{V_{\text{in}}} = \frac{k}{s}$$

In the general form of the transfer function:

$$H(s) = \frac{k_0 + k_1 s + \ldots + k_n s^n}{d_0 + d_1 s + \ldots + d_m s^m}$$

Like the previous example, if you make $k_0 = k$ and $d_1 = 1$, then the equation becomes:

$$H(s) = \frac{k + 0 + \ldots + 0}{0 + s + \ldots + 0} = \frac{k}{s}$$

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 .

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 α 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:

$$H(s) = \frac{a(s + \alpha_{z1} + j2\pi f_{z1})(s + \alpha_{z1} - j2\pi f_{z1})\ldots(s + \alpha_{zn} + j2\pi f_{zn})(s + \alpha_{zn} - j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})(s + \alpha_{p1} - j2\pi f_{p1})\ldots(s + \alpha_{pm} + j2\pi f_{pm})(s + \alpha_{pm} - j2\pi f_{pm})} \qquad (1)$$

The a, b, α, 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,   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 α 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):

$$(s + \alpha_{pm} + j2\pi f_{pm})(s + \alpha_{pm} - j2\pi f_{pm}) \quad , \text{ for poles}$$

and

$$(s + \alpha_{zn} + j2\pi f_{zn})(s + \alpha_{zn} - j2\pi f_{zn}) \quad , \text{ for zeros}$$

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, *Reduced*$\{H(s)\}$. Find the reduced form by collecting all the general form terms with negative complex roots:

$$H(s) = \frac{a(s + \alpha_{z1} + j2\pi f_{z1})\dots(s + \alpha_{zn} + j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})\dots(s + \alpha_{pm} + j2\pi f_{pm})} \cdot \frac{a(s + \alpha_{z1} - j2\pi f_{z1})\dots(s + \alpha_{zn} - j2\pi f_{zn})}{b(s + \alpha_{p1} - j2\pi f_{p1})\dots(s + \alpha_{pm} - j2\pi f_{pm})}$$

(1)

Then discard the right-hand term, which contains all the terms with negative roots. What remains is the reduced form:

$$Reduced\{H(s)\} = \frac{a(s + \alpha_{z1} + j2\pi f_{z1})\dots(s + \alpha_{zn} + j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})\dots(s + \alpha_{pm} + j2\pi f_{pm})} \quad (2)$$

For this function find the a, b, f, and α 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, α, 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 α 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 α_{z1} = 0.0 f_{z1} = 0.0

b = 1.0 α_{p1} = 0.001 f_{p1} = 0.0

Using these coefficients in the reduced form provides the desired transfer function, $\dfrac{s}{(s + 0.001)}$.

So the general transconductance transfer function POLE function call,

```
Gxxx n+ n- POLE in+ in- a αz1,fz1...αzn,fzn / b
αp1,fp1...αpm,fpm
```

for an element named *Ghigh_pass* becomes:

```
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, α, 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:

$$\frac{a(s + \alpha_{z1} + j2\pi f_{z1})...(s + \alpha_{zn} + j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})...(s + \alpha_{pm} + j2\pi f_{pm})} \ = \ \frac{1.0}{1.0(s + 1.0 + j2\pi \cdot 0.0)(s + 0.5 + j2\pi \cdot 0.15)}$$

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   b = 1.0 α_{p1} = 1.0 f_{p1} = 0 α_{p2} = 0.5 f_{p2} = 0.15

These coefficients can be substituted in the POLE function call for a voltage gain transfer function:

```
Exxx n+ n- POLE in+ in- a αz1,fz1...αzn,fzn / b
αp1,fp1...αpm,fpm
```

for an element named *Elow_pass* to obtain the Star-Hspice statement:

```
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 29-9: Circuits for an RC Line

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 29-9 can be closely approximated by a single pole response, as shown in Figure 29-10.

Figure 29-10: Transient Response of the RC Line and Single-Pole Approximation

Notice in Figure 29-10 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 29-11 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 evaluation[6].

Figure 29-11: 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
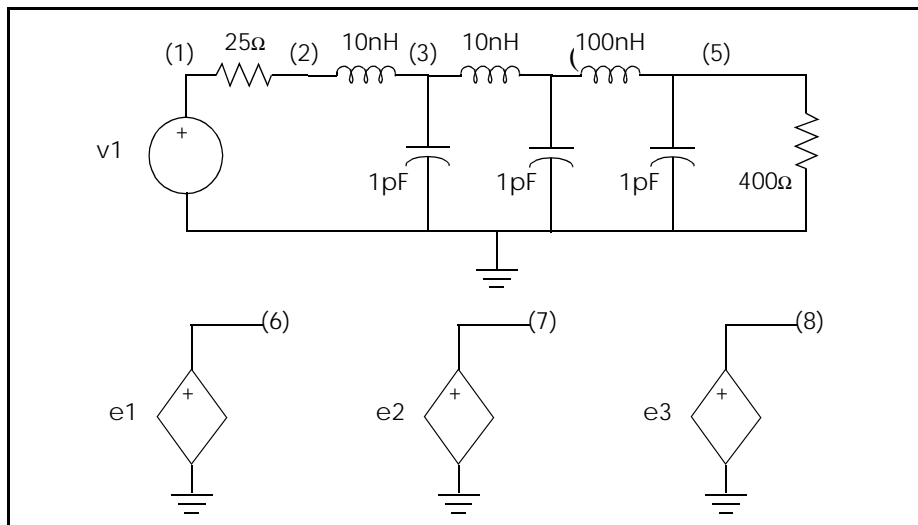
e3 8 0 LAPLACE 1 0 0.94 / 1.0 0.6n
```

```
e2 7 0 LAPLACE 1 0 0.94e20 / 1.0e20 0.348e11 14.8 1.06e-9
2.53e-19
+ SCALE=1.0e-20

e1 6 0 LAPLACE 1 0 0.94 / 1 0.2717e-9 0.12486e-18
.Probe v(1) v(5) v(6) v(7) v(8)
.Print v(1) v(5) v(6) v(7) v(8)
.End
```

Figure 29-12 shows the transient response of the low-loss line, along with
E Element Laplace models using one, two, and four poles[6]. 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.

Figure 29-12: Transient Response of the Low-Loss Line

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 29-13, compare the response of the line using a Y parameter model and a single-pole model.

Figure 29-13: Line and Y Parameter Modeling

The voltage and current definitions for a Y parameter model are shown in Figure 29-14.

Figure 29-14: Y Matrix for the Two-Port Network

The general network in Figure 29-14 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 29-15. Note that the circuit is essentially composed of G Elements.

Figure 29-15: Schematic for the Y Parameter Network

$I_1 = y_{11}V_{in} + y_{12}V_{out}$
$I_2 = y_{21}V_{in} + y_{22}V_{out}$

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.

$$Y = \frac{1}{Z_o} \cdot \begin{bmatrix} \coth(p) & -\text{csch}(p) \\ -\text{csch}(p) & \coth(p) \end{bmatrix},$$

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 *coth* and *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 *coth(p)* and *csch(p)*, with second order numerator and third order denominator, is given below:

$$coth(p) \rightarrow \quad \frac{\left(1 + \frac{2}{5} \cdot p^2\right)}{\left(p + \frac{1}{15} \cdot p^3\right)}$$

$$csch(p) \rightarrow \quad \frac{\left(1 - \frac{1}{20} \cdot p^2\right)}{\left(p + \frac{7}{60} \cdot p^3\right)}$$

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_0 ($Z_o = \sqrt{L/C}$) 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 *csch* and *coth*, but the circuit file contains identical denominators. Although the actual denominators for *csch* and *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 *coth* and *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 29-16 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.

Figure 29-16: Transient Response of the Y Parameter Line Model

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 29-17 and 29-18 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 dcstep=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 dcstep=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:

```
analysis            time        # points    . iter
conv.iter

op point            0.23        1           3
ac analysis         0.47        151         151
transient           0.75        201         226         113
rev= 0
readin              0.22
errchk              0.13
setup               0.10
output              0.00
            total cpu time  1.98 seconds
```

Pole/zero model simulation times:

```
analysis              time        # points    tot. iter
conv.iter

op point              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 29-17: Transient Responses of the Circuit and Pole/Zero Models

Figure 29-18: 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:

$$Res = \frac{Tswitch}{C3}$$

where Tswitch is the period of the pulses PHI1 and PHI2.

Figure 29-19: VCR1.SP Switched Capacitor RC Circuit

Example

```
*FILE:VCR1.SP    A SWITCHED CAPACITOR RC CIRCUIT
.OPTIONS acct NOMOD POST
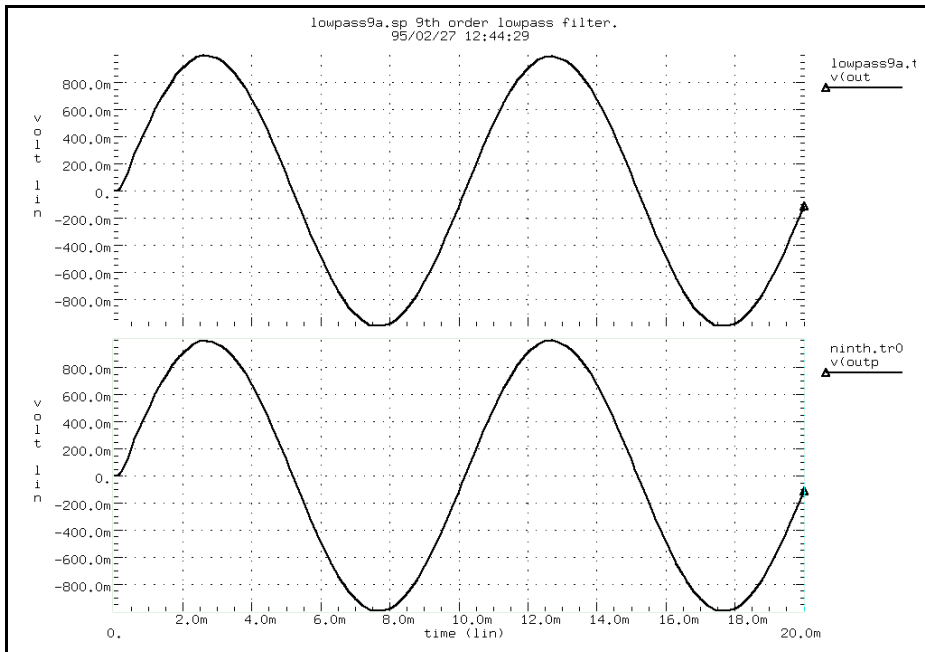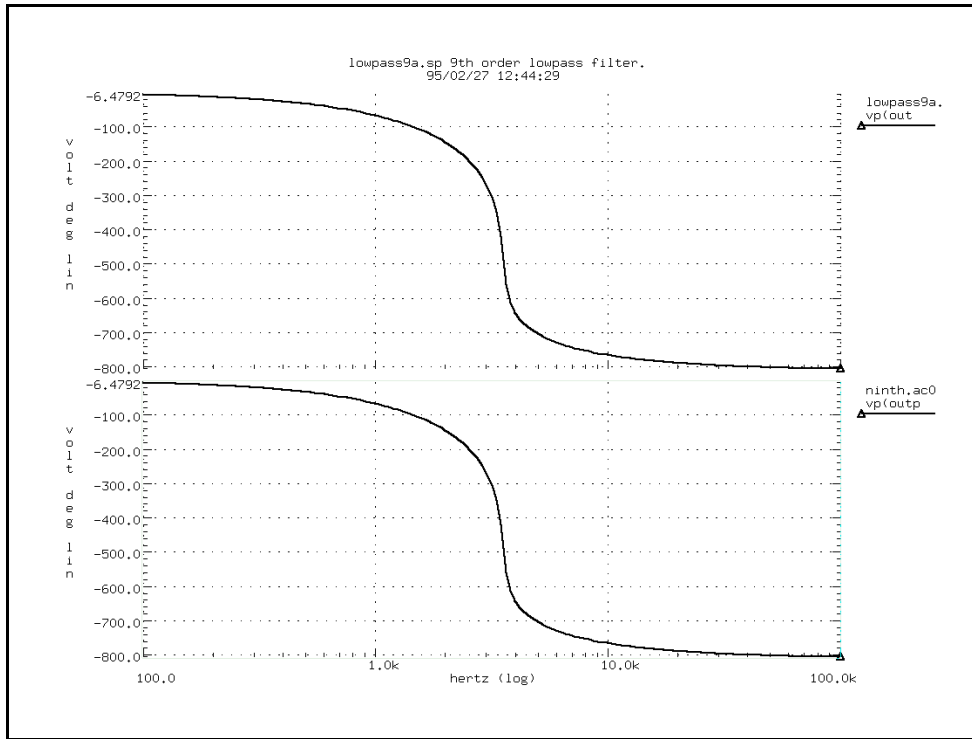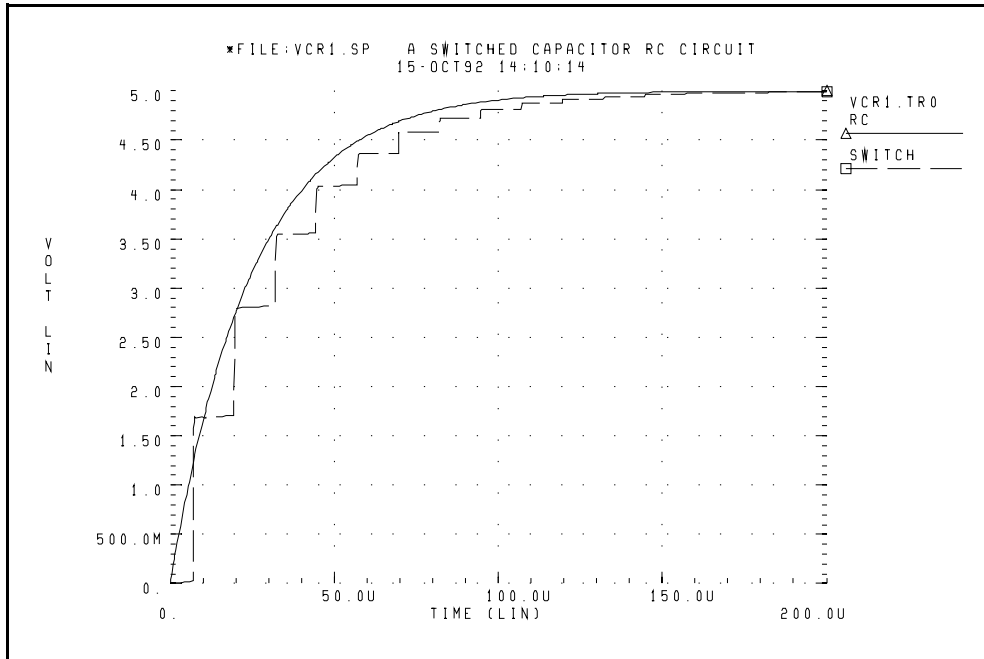
.IC V(SW1)=0 V(RCOUT)=0 V(VCROUT)=0
.TRAN 5U 200U
.GRAPH RC=V(RCOUT)  SWITCH=V(VCROUT)  (0,5)

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
```

```
GVCR1 VCC SW1 PHI1 GND LEVEL=1 MIN=100 MAX=1MEG 1.MEG -.5MEG
GVCR2 SW1 VCROUT PHI2 GND LEVEL=1 MIN=100 MAX=1MEG 1.MEG .5MEG
C3    SW1    GND   .5NF

.END
```

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[8].

Figure 29-20: Linear Section

Figure 29-21: High_Q Biquad Section

Figure 29-22: 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   out 3  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  in  3 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
```

```
Gs6   5   0 VCR PWL(1) phi2 0  0.5v,100meg 1.0v,1.0
C3    5   6  0.59075pf
Gs7   6   7 VCR PWL(1) phi2 0  0.5v,100meg 1.0v,1.0
Gs8   6   0 VCR PWL(1) phi1 0  0.5v,100meg 1.0v,1.0
Eop2 out 0 OPAMP 0 7
Cb    7  out  4.3733pf
Gs9  out 8 VCR PWL(1) phi2 0 0.5v,100meg 1.0v,1.0
Gs10 8    0 VCR PWL(1) phi1 0 0.5v,100meg 1.0v,1.0
C4   out 3 1.6518pf
C2   8    2 0.9963pf
C11  7   in 0.5pf
.ENDS
```

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
C11  8  1  3.15425pf
Gs12 9 out VCR PWL(1) phi2 0  0.5v,100meg 1.0v,1.0
.ENDS
.END
```

Figure 29-23: Response to 1-kHz Sinusoidal Input

Figure 29-24: Response to 2-kHz Sinusoidal Input

References

References for this chapter are listed below.

1. Williams, Arthur B., and Taylor, Fred J. *Electronic Filter Design Handbook*. New York: McGraw-Hill, 1988, pp. 6-20 to 6-23.

2. Nillson, James W. *Electric Circuits*, 4th Edition. Reading, Massachusetts: Addison-Wesley, 1993.

3. Edminister, Joseph A. *Electric Circuits*. New York: McGraw-Hill, 1965.

4. Ghausi, Kelly, and M.S. *On the Effective Dominant Pole of the Distributed RC Networks*. *Jour. Franklin Inst*., June 1965, pp. 417- 429.

5. Elmore, W.C. and Sands, M. *Electronics, National Nuclear Energy Series*, New York: McGraw-Hill, 1949.

6. Pillage, L.T. and Rohrer, R.A. *Asymptotic Waveform Evaluation for Timing Analysis*, *IEEE Trans. CAD*, Apr. 1990, pp. 352 - 366.

7. Kuo, F. F. *Network Analysis and Synthesis*. John Wiley and Sons, 1966.

8. Gregorian, Roubik & Temes, Gabor C. *Analog MOS Integrated Circuits*. J. Wiley, 1986, page 354.

Avant!

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(out) goes from low to high. The following relationship between times T_1(data) and T_2 (clock) must be true in order for the V(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 T_1 to T_2 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.

Figure 30-1: Determining Setup Time with Bisection Violation Analysis

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 (T_1 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 Š |goal value| and the other in a result < |goal value|

- Specified goal value

- Error tolerance value. The bisection process stops when the difference between successive test values ≤ error tolerance. If the other criteria are met, see below.

- Related variables. Variables must be related by a monotonic transfer function, where a steadily progressing time (increase or decrease) results in a single occurrence of the "goal" value at the "target" input variable value

The error tolerance is included in a relation used as a process-termination criterion.

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 TS that produces a VOUT output "minimum high" logic output level of Š 2.7 V – the "goal" value. The "target" value is that setup time that *just* produces the V_{OUT} value of 2.7 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:

```
bisec-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:

```
bisec-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
*
```

```
* 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=l w=wn
mb1 out in vdd vdd pmos l=l w=wp
.ends

.SUBCKT DFF  c d nc nq
Xi64  nc net46 c net36 XGATE
Xi66  nc net38 c net39 XGATE
Xi65  c nq nc net36 XGATE
Xi62  c d nc net39 XGATE
Xi60  net722 nq INV
Xi61  net46 net38 INV
Xi59  net36 net722 INV
Xi58  net39 net46 INV
```

```
    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

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     =   .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
setuptime       = 2.0000E-09  targ= 3.5000E-09  trig= 1.5000E-09
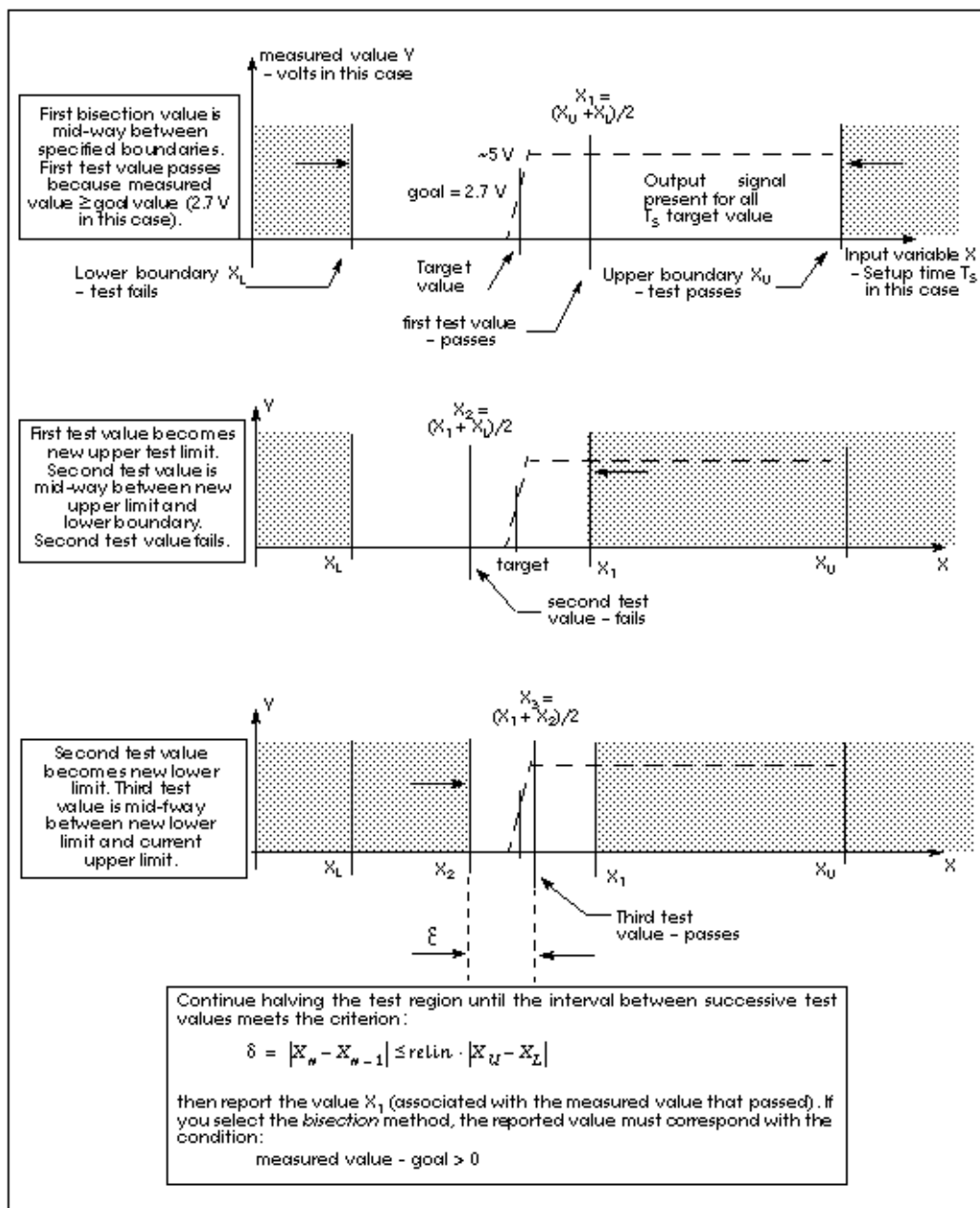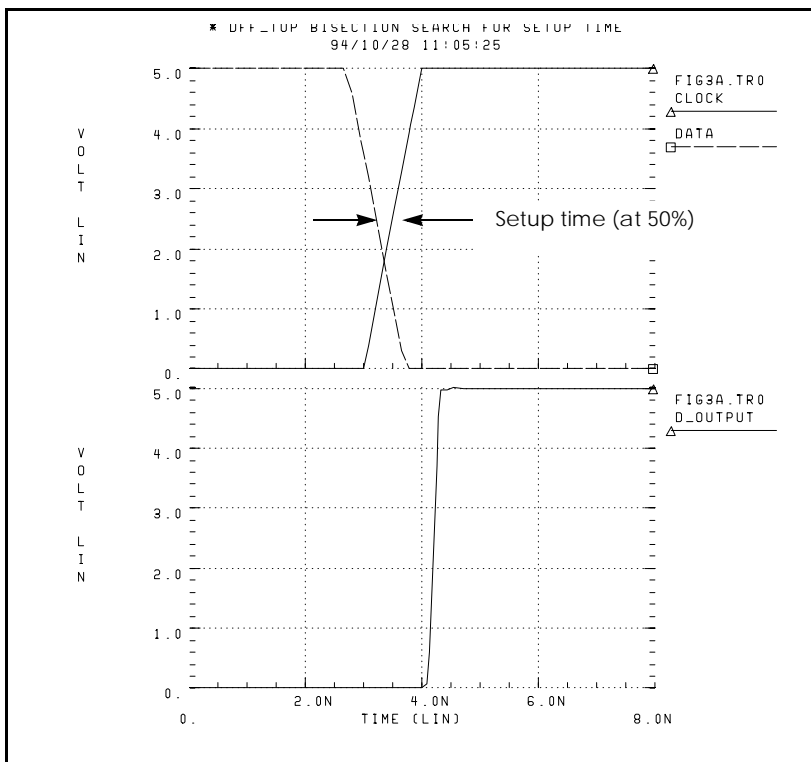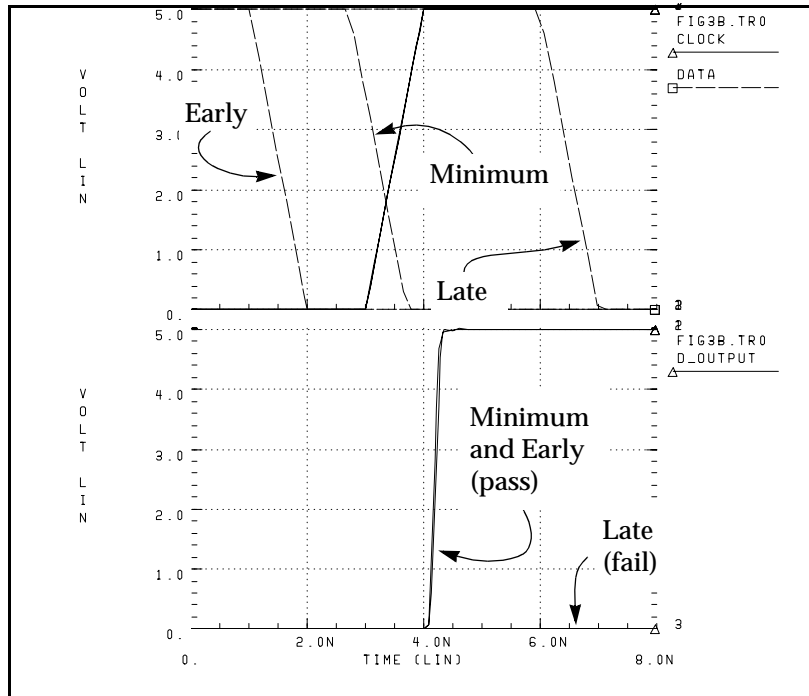
*** parameter DelayTime =    1.719E-09 *** $ Optimal
setuptime       = 2.8120E-10  targ= 3.5000E-09  trig= 3.2188E-09

*** parameter DelayTime =    5.000E-09 *** $ Late
setuptime       = -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 */bisect/inv_a.sp.* in the *$installdir/demo/hspice* directory. The results of this demo are shown in Figure 30-5.

Input listing

```
File: $installdir/demo/bisect/inv_a.sp
$ Inv_a.sp testing bisectional search, cload=10p & 20p
*
* Parameters
.PARAM Cload   =10p                      $ See end of deck for
Alter
.PARAM Tpw     =opt1(0,0,15n)            $ 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 xl=0.06u xwn=0.3u xwp=0.3u
+ tox=200 delvton=0 delvtop=0 rshn=50
+ rshp=150
.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 u1=.02 x3u1=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
+ 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 x2u1=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 x3u1=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

Avant!

Chapter 31

Running Demonstration Files

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
	/cchar	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
X2   A[1]  B[1]  carry-out_1 C[1]  carry-out_2  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
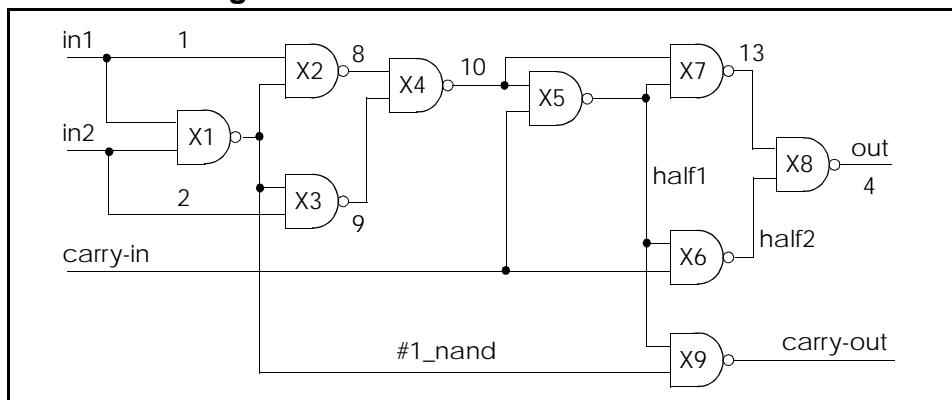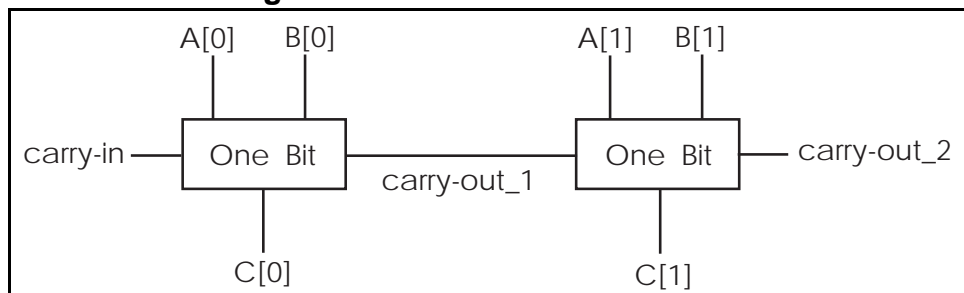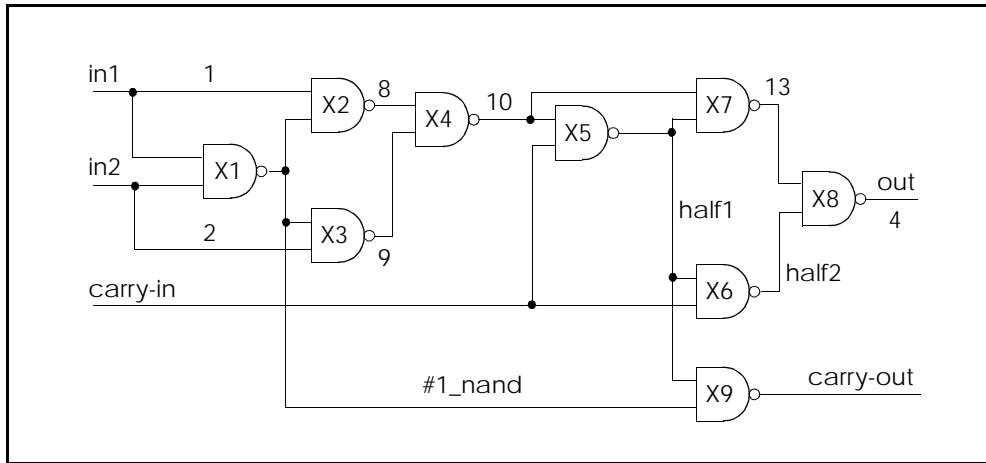
* switch model equivalent of the NAND. Gives a 10 times
* speedup over the MOS version.

.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 out 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
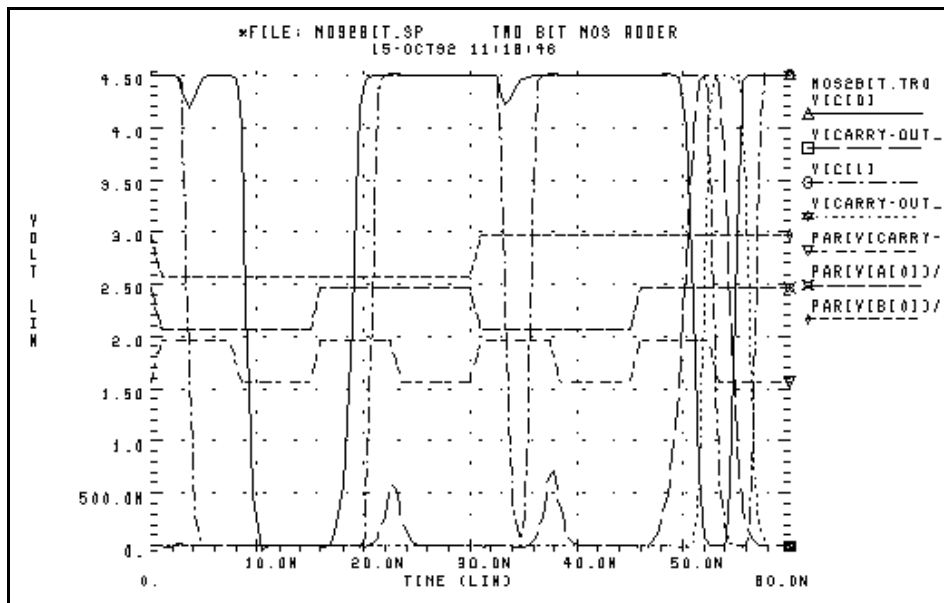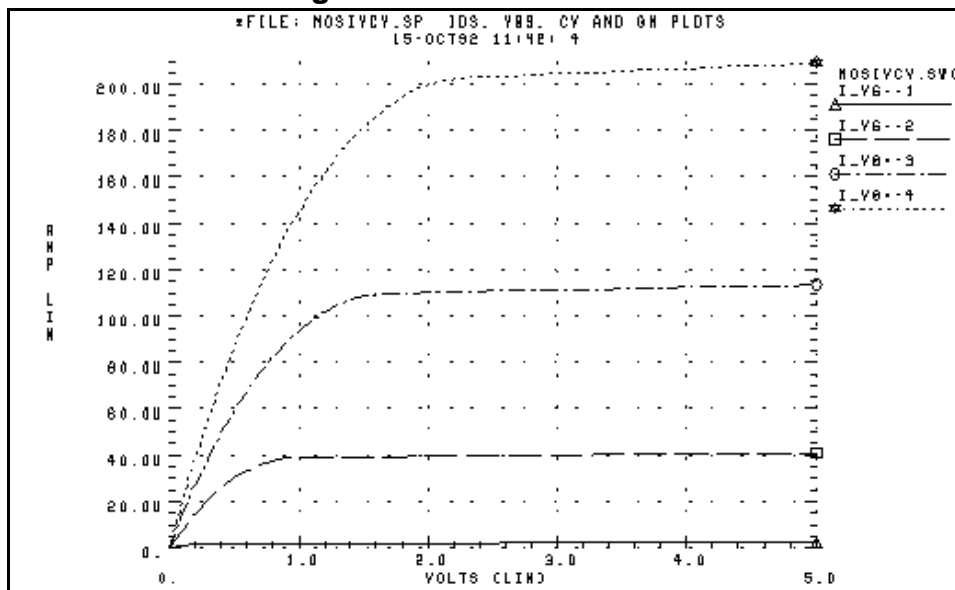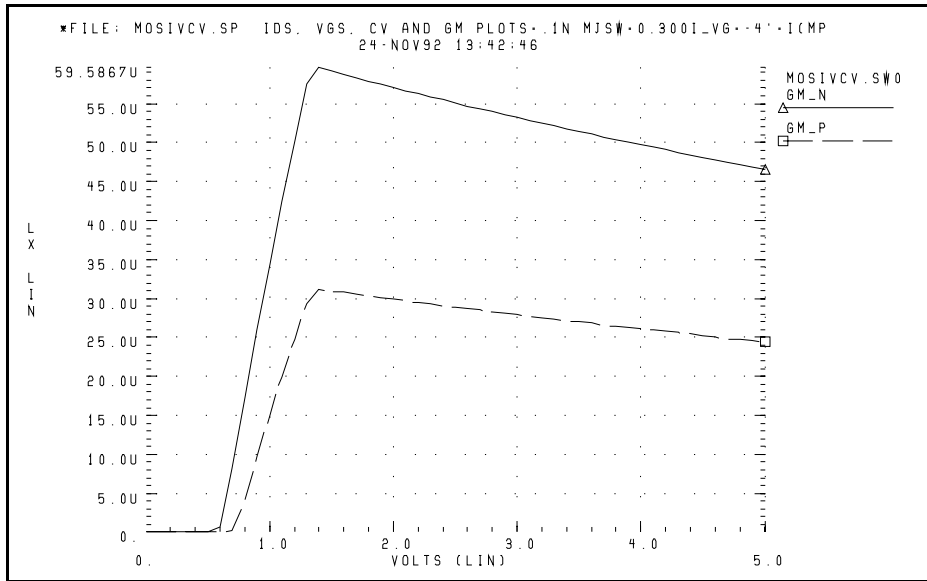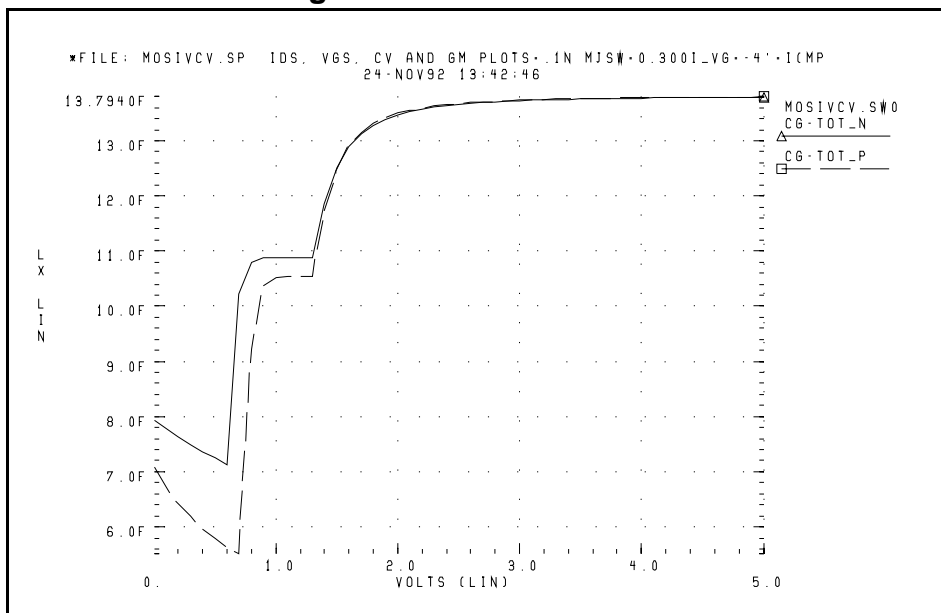
.OPTIONS SCALE=1U DCCAP
.DC VDDN 0 5.0 .1 $VBBN 0 -3 -3      sweep supplies
.PARAM ww=8 LL=2

$ 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
```

```
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   vddlow_n vdd_n  gnd vbb_n N  W=ww L=LL
MP6   gnd vdd_p vddlow_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

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
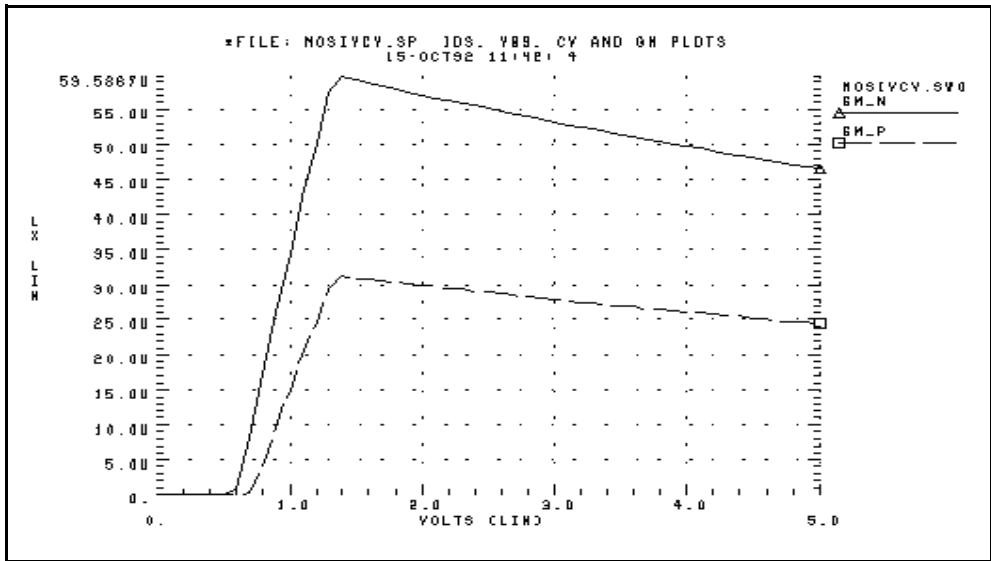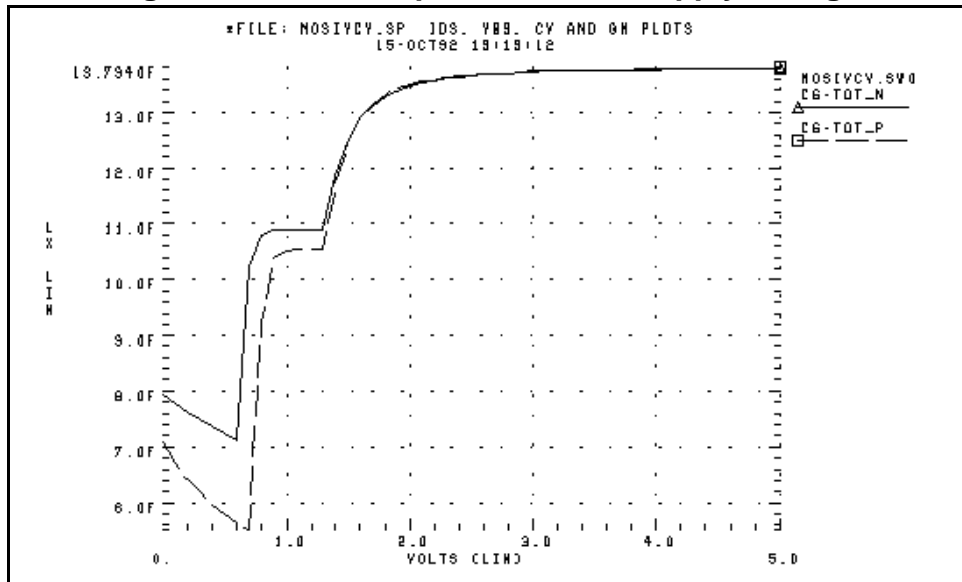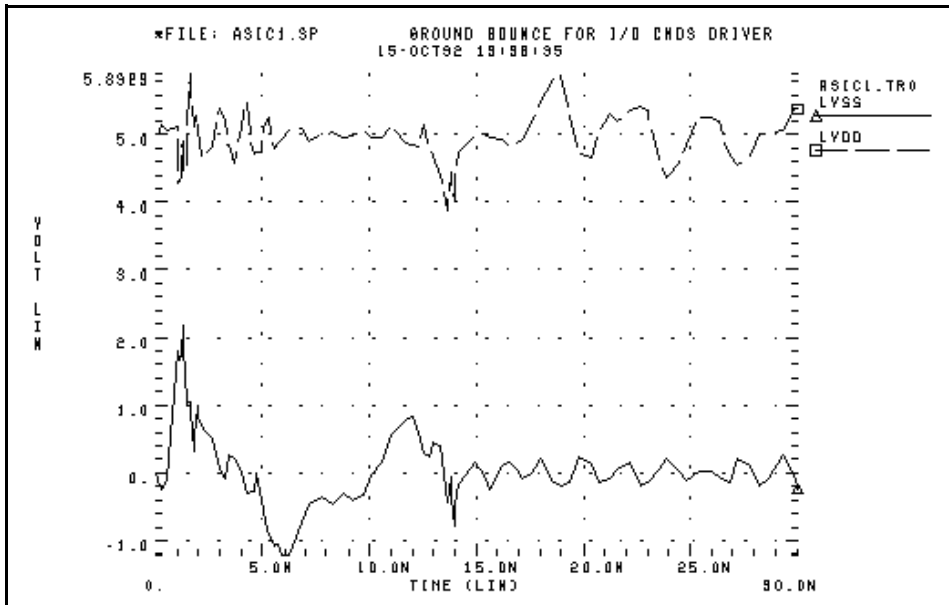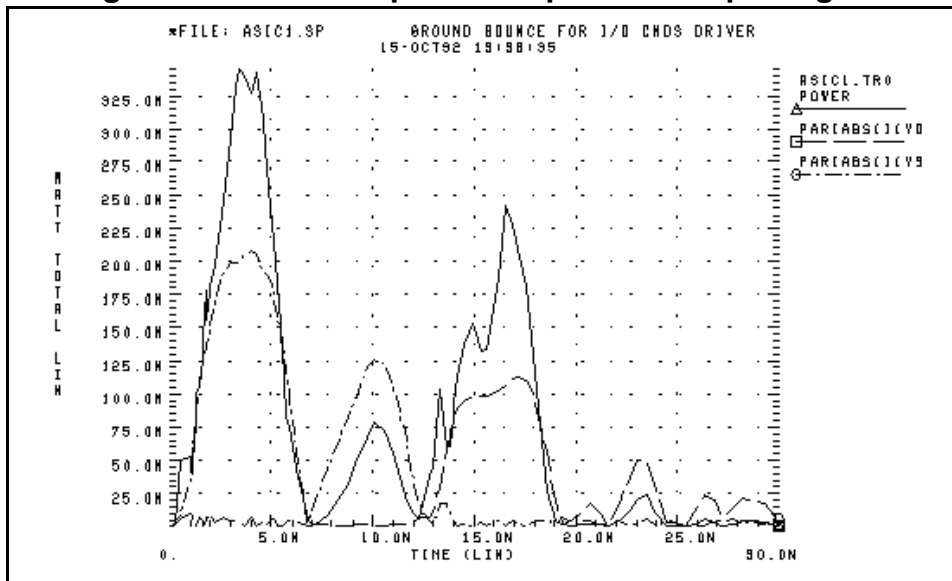
GROUND BOUNCE FOR I/O CMOS DRIVER 1200/1.2 & 800/1.2 MICRONS

.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
```

```
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

*Tranmission 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 tc1r=tc1r_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 tc1r_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

$$Rpoly \ = \ \frac{W}{L} \cdot 40$$

$$poly \ = \ \frac{Esio \cdot nsi}{tox} \cdot L \cdot W$$

$$Rpoly \ = \ \frac{100}{5} \cdot 40 \ = \ 800 \, , \quad Co \ = \ \frac{3.9 \cdot 8.86}{800} \cdot 100 \cdot 5 \ = \ 215fF$$

RC = 138 ps

For a transistor with

channel width = 100 microns

channel length = 1.2 microns

gate oxide thickness = 250 Angstroms

$$Rpoly \ = \ \frac{channel \ \ width}{channel \ \ length} \cdot 40$$

$$Co = \frac{3.9 \cdot 8.86}{\text{Tox}} \cdot \text{channel width} \cdot \text{channel length}$$

RC = 546 ps

You can model the RC delay introduced in modern CMOS technologies by using a nine-stage ladder model.

Figure 31-13: Nine-stage Ladder Model

In this example, the nine-stage ladder model was entered into a Star-Hspice data file, *$installdir/demo/hspice/apps /asic3.sp*, and then optimized by Star-Hspice (with actual measured data of a wide channel transistor as the target data). The optimization produced a nine-stage ladder model that matched the timing characteristics of the physical data. The simulation results for the nine-stage ladder model and the one-stage model were then compared using the nine-stage ladder model as the reference. The one-stage model produces results that are about 10% faster than the actual physical data indicates.

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 g1 source bulk p w='wt/9' l=lt
m3 drain g2 source bulk p w='wt/9' l=lt
m4 drain g3 source bulk p w='wt/9' l=lt
m5 drain g4 source bulk p w='wt/9' l=lt
m6 drain g5 source bulk p w='wt/9' l=lt
m7 drain g6 source bulk p w='wt/9' l=lt
m8 drain g7 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

File Name	Description
Algebraic Output Variable Examples	*$installdir/demo/hspice/alge*
alg.sp	demonstration of algebraic parameters
alg_fil.sp	magnitude response of behavioral filter model
alg_vco.sp	voltage controlled oscillator
alg_vf.sp	voltage-to-frequency converter behavioral model
xalg1.sp	QA of parameters
xalg2.sp	QA of parameters
Applications of General Interest	*$installdir/demo/hspice/apps*
alm124.sp	AC, noise, transient op-amp analysis
alter2.sp	.ALTER examples
ampg.sp	pole/zero analysis of a G source amplifier
asic1.sp	ground bounce for I/O CMOS driver
asic3.sp	ten-stage lumped MOS model
bjt2bit.sp	BJT two-bit adder
bjt4bit.sp	four-bit, all NAND gate binary adder
bjtdiff.sp	BJT diff amp with every analysis type
bjtschmt.sp	bipolar Schmidt trigger
bjtsense.sp	bipolar sense amplifier
cellchar.sp	ASIC inverter cell characterization
crystal.sp	crystal oscillator circuit
gaasamp.sp	simple GaAsFET amplifier
grouptim.sp	group time delay example
inv.sp	sweep MOSFET -3 sigma to +3 sigma, use .MEASURE output
mcdiff.sp	CMOS differential amplifier

File Name	Description
mondc_a.sp	Monte Carlo of MOS diffusion and photolithographic effects
mondc_b.sp	Monte Carlo DC analysis
mont1.sp	Monte Carlo Gaussian, uniform, and limit function
mos2bit.sp	two-bit MOS adder
pll.sp	phase locked loop
sclopass.sp	switched capacitor low-pass filter
worst.sp	worst case skew models using .ALTER
xbjt2bit.sp	BJT NAND gate two-bit binary adder
Behavioral Applications *$installdir/demo/hspice/behave*	
acl.sp	acl gate
amp_mod.sp	amplitude modulator with pulse waveform carrier
behave.sp	AND/NAND gates using G, E Elements
calg2.sp	voltage variable capacitance
det_dff.sp	double edge triggered flip-flop
diff.sp	differentiator circuit
diode.sp	behavioral diode using a PWL VCCS
dlatch.sp	CMOS D-latch using behaviorals
galg1.sp	sampling a sine wave
idealop.sp	ninth-order low-pass filter
integ.sp	integrator circuit
invb_op.sp	optimization of CMOS macromodel inverter
ivx.sp	characterization of PMOS and NMOS as a switch
op_amp.sp	op-amp from Chua and Lin
pd.sp	phase detector modeled by switches
pdb.sp	phase detector using behavioral NAND gates
pwl10.sp	operational amplifier used as voltage follower

File Name	Description
pwl2.sp	PPW-VCCS with gain of 1 amp/volt
pwl4.sp	eight-input NAND gate
pwl7.sp	modeling inverter by a PWL VCVS
pwl8.sp	smoothing the triangle waveform by PWL CCCS
ring5bm.sp	five-stage ring oscillator – macromodel CMOS inverter
ringb.sp	ring oscillator using behavioral model
sampling.sp	sampling a sine wave
scr.sp	silicon controlled rectifier modelled with PWL CCVS
swcap5.sp	fifth-order elliptic switched capacitor filter
switch.sp	test for PWL switch element
swrc.sp	switched capacitor RC circuit
triode.sp	triode model family of curves using behavioral elements
triodex.sp	triode model family of curves using behavioral elements
tunnel.sp	modeling tunnel diode characteristic by PWL VCCS
vcob.sp	voltage controlled oscillator using PWL functions
Benchmarks	*$installdir/demo/hspice/bench*
bigmos1.sp	large MOS simulation
demo.sp	quick demo file to test installation
m2bit.sp	72-transistor two-bit adder – typical cell simulation
m2bitf.sp	fast simulation example
m2bitsw.sp	fast simulation example – same as m2bitf.sp but using behavioral elements
senseamp.sp	bipolar analog test case
Timing Analysis	*$installdir/demo/hspice/bisect*
fig3a.sp	DFF bisection search for setup time
fig3b.sp	DFF early, optimum, and late setup times
inv_a.sp	inverter bisection pass-fail

File Name	Description
BJT and Diode Devices	*$installdir/demo/hspice/bjt*
bjtbeta.sp	plot BJT beta
bjtft.sp	plot BJT FT using s-parameters
bjtgm.sp	plot BJT Gm, Gpi
dpntun.sp	junction tunnel diode
snaphsp.sp	convert SNAP to Star-Hspice
tun.sp	tunnel oxide diode
Cell Characterization	*$installdir/demo/hspice/cchar*
dff.sp	DFF bisection search for setup time
inv3.sp	inverter characterization
inva.sp	inverter characterization
invb.sp	inverter characterization
load1.sp	inverter sweep, delay versus fanout
setupbsc.sp	setup characterization
setupold.sp	setup characterization
setuppas.sp	setup characterization
sigma.sp	sweep MOSFET -3 sigma to +3 sigma, use measure output
tdgtl.a2d	Viewsim A2D Star-Hspice input file
tdgtl.d2a	Viewsim D2A Star-Hspice input file
tdgtl.sp	two-bit adder using D2A Elements
Circuit Optimization	*$installdir/demo/hspice/ciropt*
ampgain.sp	set unity gain frequency of BJT diff pair
ampopt.sp	optimize area, power, speed of MOS amp
asic2.sp	optimize speed, power of CMOS output buffer
asic6.sp	find best width of CMOS input buffer
delayopt.sp	optimize group delay of LCR circuit

File Name	Description
lpopt.sp	match lossy filter to ideal filter
opttemp.sp	find first and second temperature coefficients of resistor
rcopt.sp	optimize speed, power for RC circuit
DDL	*$installdir/demo/hspice/ddl*
ad8bit.sp	eight-bit A/D flash converter
alf155.sp	National JFET op-amp characterization
alf156.sp	National JFET op-amp characterization
alf157.sp	National JFET op-amp characterization
alf255.sp	National JFET op-amp characterization
alf347.sp	National JFET op-amp characterization
alf351.sp	National wide bandwidth JFET input op-amp characterization
alf353.sp	National wide bandwidth dual JFET input op-amp char.
alf355.sp	Motorola JFET op-amp characterization
alf356.sp	Motorola JFET op-amp characterization
alf357.sp	Motorola JFET op-amp characterization
alf3741.sp	
alm101a.sp	
alm107.sp	National op-amp characterization
alm108.sp	National op-amp characterization
alm108a.sp	National op-amp characterization
alm118.sp	National op-amp characterization
alm124.sp	National low power quad op-amp characterization
alm124a.sp	National low power quad op-amp characterization
alm158.sp	National op-amp characterization
alm158a.sp	National op-amp characterization
alm201.sp	LM201 op-amp characterization

File Name	Description
alm201a.sp	LM201 op-amp characterization
alm207.sp	National op-amp characterization
alm208.sp	National op-amp characterization
alm208a.sp	National op-amp characterization
alm224.sp	National op-amp characterization
alm258.sp	National op-amp characterization
alm258a.sp	National op-amp characterization
alm301a.sp	National op-amp characterization
alm307.sp	National op-amp characterization
alm308.sp	National op-amp characterization
alm308a.sp	National op-amp characterization
alm318.sp	National op-amp characterization
alm324.sp	National op-amp characterization
alm358.sp	National op-amp characterization
alm358a.sp	National op-amp characterization
alm725.sp	National op-amp characterization
alm741.sp	National op-amp characterization
alm747.sp	National op-amp characterization
alm747c.sp	National op-amp characterization
alm1458.sp	National dual op-amp characterization
alm1558.sp	National dual op-amp characterization
alm2902.sp	National op-amp characterization
alm2904.sp	National op-amp characterization
amc1458.sp	Motorola internally compensated high performance op-amp characterization
amc1536.sp	Motorola internally compensated high voltage op-amp characterization
amc1741.sp	Motorola internally compensated high performance op-amp characterization

File Name	Description
amc1747.sp	Motorola internally compensated high performance op-amp characterization
ane5534.sp	TI low noise, high speed op-amp characterization
anjm4558.sp	TI dual op-amp characterization
anjm4559.sp	TI dual op-amp characterization
anjm4560.sp	TI dual op-amp characterization
aop04.sp	PMI op-amp characterization
aop07.sp	PMI ultra low offset voltage op-amp characterization
aop14.sp	PMI op-amp characterization
aop15b.sp	PMI precision JFET input op-amp characterization
aop16b.sp	PMI precision JFET input op-amp characterization
at094cns.sp	TI op-amp characterization
atl071c.sp	TI low noise op-amp characterization
atl072c.sp	TI low noise op-amp characterization
atl074c.sp	TI low noise op-amp characterization
atl081c.sp	TI JFET op-amp characterization
atl082c.sp	TI JFET op-amp characterization
atl084c.sp	TI JFET op-amp characterization
atl092cp.sp	TI op-amp characterization
atl094cn.sp	TI op-amp characterization
aupc358.sp	NEC general dual op-amp characterization
aupc1251.sp	NEC general dual op-amp characterization
j2n3330.sp	JFET 2n3330 I-V characteristics
mirf340.sp	IRF340 I-V characteristics
t2n2222.sp	BJT 2n2222 characterization
Device Optimization	*$installdir/demo/hspice/devopt*
beta.sp	LEVEL=2 beta optimization

File Name	Description
bjtopt.sp	s-parameter optimization of 2n6604 BJT
bjtopt1.sp	2n2222 DC optimization
bjtopt2.sp	2n2222 Hfe optimization
d.sp	diode, multiple temperatures
dcopt1.sp	1n3019 diode I-V and C-V optimization
gaas.sp	JFET optimization
jopt.sp	300u/1u GaAs FET DC optimization
jopt2.sp	JFET optimization
joptac.sp	300u/1u GaAs FET 40 MHz–20 GHz s-parameter optimization
l3.sp	MOS LEVEL 3 optimization
l3a.sp	MOS LEVEL 3 optimization
l28.sp	LEVEL=28 optimization
ml2opt.sp	MOS LEVEL=2 I-V optimization
ml3opt.sp	MOS LEVEL=3 I-V optimization
ml6opt.sp	MOS LEVEL=6 I-V optimization
ml13opt.sp	MOS LEVEL=13 I-V optimization
opt_bjt.sp	2n3947 forward and reverse Gummel optimization
Fourier Analysis	*$installdir/demo/hspice/fft*
am.sp	FFT analysis, AM source
bart.sp	FFT analysis, Bartlett window
black.sp	FFT analysis, Blackman window
dist.sp	FFT analysis, second harmonic distortion
exam1.sp	FFT analysis, AM source
exam3.sp	FFT analysis, high frequency signal detection test
exam4.sp	FFT analysis, small-signal harmonic distortion test
exp.sp	FFT analysis, exponential source

File Name	Description
fft.sp	FFT analysis, transient, sweeping a resistor
fft1.sp	FFT analysis, transient
fft2.sp	FFT analysis on the product of three waveforms
fft3.sp	FFT analysis, transient, sweeping frequency
fft4.sp	FFT analysis, transient, Monte Carlo Gaussian distribution
fft5.sp	FFT analysis, data-driven transient analysis
fft6.sp	FFT analysis, sinusoidal source
gauss.sp	FFT analysis, Gaussian window
hamm.sp	FFT analysis, Hamming window
hann.sp	FFT analysis, Hanning window
harris.sp	FFT analysis, Blackman-Harris window
intermod.sp	FFT analysis, intermodulation distortion
kaiser.sp	FFT analysis, Kaiser window
mod.sp	FFT analysis, modulated pulse
pulse.sp	FFT analysis, pulse source
pwl.sp	FFT analysis, piecewise linear source
rect.sp	FFT analysis, rectangular window
rectan.sp	FFT analysis, rectangular window
sffm.sp	FFT analysis, single-frequency FM source
sine.sp	FFT analysis, sinusoidal source
swcap5.sp	FFT analysis, fifth-order elliptic switched capacitor filter
tri.sp	FFT analysis, rectangular window
win.sp	FFT analysis, window test
window.sp	FFT analysis, window test
winreal.sp	FFT analysis, window test
Filters	*$installdir/demo/hspice/filters*

File Name	Description
fbp_1.sp	bandpass LCR filter measurement
fbp_2.sp	bandpass LCR filter pole/zero
fbpnet.sp	bandpass LCR filter s-parameters
fbprlc.sp	LCR AC analysis for resonance
fhp4th.sp	high-pass LCR fourth-order Butterworth filter
fkerwin.sp	pole/zero analysis of Kerwin's circuit
flp5th.sp	low-pass fifth-order filter
flp9th.sp	low-pass ninth-order FNDR with ideal op-amps
micro1.sp	test of microstrip
micro2.sp	test of microstrip
tcoax.sp	test of RG58/AU coax
trans1m.sp	FR-4 printed circuit lumped transmission line
Magnetics	*$installdir/demo/hspice/mag*
aircore.sp	air core transformer circuit
bhloop.sp	b-h loop nonlinear magnetic core transformer
mag2.sp	three primary, two secondary magnetic core transformer
magcore.sp	magnetic core transformer circuit
royerosc.sp	Royer magnetic core oscillator
MOSFET Devices	*$installdir/demo/hspice/mos*
bsim3.sp	LEVEL=47 BSIM3 model
cap13.sp	plot MOS capacitances LEVEL=13 model
cap_b.sp	capacitances for LEVEL=13 model
cap_m.sp	capacitance for LEVEL=13 model
capop0.sp	plot MOS capacitances LEVEL=2
capop1.sp	plot MOS capacitances LEVEL=2
capop2.sp	plot MOS capacitances LEVEL=2

File Name	Description
capop4.sp	plot MOS capacitances LEVEL=6
chrgpump.sp	charge conservation test LEVEL=3
iiplot.sp	impact ionization current plot
ml6fex.sp	plot temperature effects LEVEL=6
ml13fex.sp	plot temperature effects LEVEL=13
ml13ft.sp	s-parameters for LEVEL=13
ml13iv.sp	plot I-V for LEVEL=13
ml27iv.sp	plot I-V for LEVEL=27 SOSFET
mosiv.sp	plot I-V for user include file
mosivcv.sp	plot I-V and C-V for LEVEL=3
qpulse.sp	charge conservation test LEVEL=6
qswitch.sp	charge conservation test LEVEL=6
selector.sp	automatic width and length model selector
tgam2.sp	LEVEL=6 gamma model
tmos34.sp	MOS LEVEL=34 EPFL, test DC
Peripheral Component Interconnect *$installdir/demo/hspice/pci*	
pci_lab.sp	Intel Peripheral Component Interconnect demonstration
pci_mont.sp	PCI Monte Carlo example
pci_wc.sp	PCI worst-case modeling
Radiation Effects *$installdir/demo/hspice/rad*	
brad1.sp	bipolar radiation effects example
brad2.sp	bipolar radiation effects example
brad3.sp	bipolar radiation effects example
brad4.sp	bipolar radiation effects example
brad5.sp	bipolar radiation effects example
brad6.sp	bipolar radiation effects example

File Name	Description
drad1.sp	diode radiation effects example
drad2.sp	diode radiation effects example
drad4.sp	diode radiation effects example
drad5.sp	diode radiation effects example
drad6.sp	diode radiation effects example
dradarb2.sp	diode radiation effects example
jex1.sp	JFET radiation effects example
jex2.sp	JFET radiation effects example
jprad1.sp	JFET radiation effects example
jprad2.sp	JFET radiation effects example
jprad4.sp	JFET radiation effects example
jrad1.sp	JFET radiation effects example
jrad2.sp	JFET radiation effects example
jrad3.sp	JFET radiation effects example
jrad4.sp	JFET radiation effects example
jrad5.sp	JFET radiation effects example
jrad6.sp	JFET radiation effects example
mrad1.sp	MOSFET radiation effects example
mrad2.sp	MOSFET radiation effects example
mrad3.sp	MOSFET radiation effects example
mrad3p.sp	MOSFET radiation effects example
mrad3px.sp	MOSFET radiation effects example
rad1.sp	total MOSFET dose example
rad2.sp	diode photocurrent test circuit
rad3.sp	diode photocurrent test circuit RLEV=3
rad4.sp	diode photocurrent test circuit

File Name	Description
rad5.sp	BJT photocurrent test circuit with an NPN transistor
rad6.sp	BJT secondary photocurrent effect which varies with R1
rad7.sp	BJT RLEV=6 example (semi-empirical model)
rad8.sp	JFET RLEV=1 example with Wirth-Rogers square pulse
rad9.sp	JFET stepwise increasing radiation source
rad10.sp	GaAs RLEV=5 example (semi-empirical model)
rad11.sp	NMOS E-mode LEVEL=8 with Wirth-Rogers square pulse
rad12.sp	NMOS 0.5x resistive voltage divider
rad13.sp	three-input NMOS NAND gate with non-EPI, EPI, and SOS examples
rad14.sp	GaAs differential amplifier circuit
rad14dc.sp	n-channel JFET DC I-V curves
Sources	*$installdir/demo/hspice/sources*
amsrc.sp	amplitude modulation
exp.sp	exponential independent source
pulse.sp	test of pulse
pwl.sp	repeated piecewise linear source
pwl10.sp	op-amp voltage follower
rtest.sp	voltage controlled resistor inverter chain
sffm.sp	single frequency FM modulation source
sin.sp	sinusoidal source waveform
vcr1.sp	switched capacitor network using G-switch
Transmission Lines	*$installdir/demo/hspice/tline*
fr4.sp	microstrip test FR-4 PC board material
fr4o.sp	optimizing model for microstrip FR-4 PC board material
fr4x.sp	FR4 microstrip test
hd.sp	ground bounce for I/O CMOS driver

File Name	Description
rcsnubts.sp	ground bounce for I/O CMOS driver at snubber output
rcsnubtt.sp	ground bounce for I/O CMOS driver
strip1.sp	two series microstrips (8 mil and 16 mil wide)
strip2.sp	two microstrips coupled together
t14p.sp	1400 mil by 140 mil, 50 ohm tline on FR-4 50 MHz - 10.05 GHz
t14xx.sp	1400 mil by 140 mil, 50 ohm tline on FR-4 optimization
t1400.sp	1400 mil by 140 mil, 50 ohm tline on FR-4 optimization
tcoax.sp	RG58/AU coax with 50 ohm termination
tfr4.sp	microstrip test
tfr4o.sp	microstrip test
tl.sp	series source coupled and shunt terminated transmission lines
transmis.sp	algebraics and lumped transmission lines
twin2.sp	twinlead model
xfr4.sp	microstrip test subcircuit expanded
xfr4a.sp	microstrip test subcircuit expanded, larger ground resistance
xfr4b.sp	microstrip test
xulump.sp	test 5-, 20-, and 100-lump U models

Avant!

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 RISETIME. For example:

```
.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, initilizations, 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

* *

Documentation

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.

- 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

* *

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

* *

Input

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., ..installdir../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 "...installdir\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 A-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 A-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:

Recommended	**Not Recommended**
.SUBCKT	.SUBCKT
.PROTECT	.PROTECT
~~	~~~
.UNPROTECT	.ENDS
.ENDS	.UNPROTECT

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

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	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

Sharp	SHARP	LEVEL 36
TI	TI	LEVEL 37
		LEVEL 41
Vitesse	VITESSE	LEVEL 6

As of July 12, 2000, these are the contact people:

AMD	Barbara Vervenne	(512) 602-5783
Dallas Semi	Roy Hensley	(214) 450-3858
Hitachi	Len Mizrah	(408) 456-2081
IBM	Cal Bittner	(902) 769-6528
Lucent	Roberta Kulp	(610) 712-2614
Motorola	Ralph Sokel	(512) 933-5264
National	Chen Young Tao	(408) 721-5665
SGS Thomson	Claude Frank	33 476 92 63 47 (Tel) 33 476 08 96 52 (Fax)
Sharp	Sigenobu Kiyoi	81 7 4365 4273
TI	Jeff Brunson	(972) 480-2481
Vitesse	Aubrey Grey	(805) 388-7517

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=abs(gm-go)/(2*pi*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.

For another discussion on this, see "Semiconductor Device Modeling with SPICE" by Massobrio and Antognetti, McGraw Hill, 1993, page 99.

Solution: 560

* *

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: 562

* *

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 .

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 A-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 8-33 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 A-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.

Window	Coherent Power Gain	Coherent Power Gain (dB)
Rectangular	1.0	0.0
Bartlett	0.5	-6.021
Hanning	0.5	-6.021
Hamming	0.54	-5.352
Blackman	0.42323	-7.468
Blackman - Harris	0.35875	-8.904
Gaussian (a=3)	0.72	-2.853
Kaiser - Bessel (a=3)	0.4	-7.959

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

D	MOSFET or JFET drain
E	BJT emitter
G	MOSFET or JFET gate
S	BJT substrate or MOSFET, JFET source

Solution: 360

* *

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 A-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 A-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 .

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

* *

Avant!

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**

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

```
┌──────────────────────────────────────────────┐
│                                                │
│   ┌────────────────────────────────────────┐  │
│   │         Design Framework II            │  │
│   ├────────────────────┬───────────────────┤  │
│   │   Analog Artist    │     Composer      │  │
│   └────────────────────┴───────────────────┘  │
│          ┌───────────────────────┐             │
│          │       AvanLink        │             │
│      ┌───┴───────────────────────┴───┐         │
│      │         Star-Hspice           │         │
│      └───────────────────────────────┘         │
│                                                │
└──────────────────────────────────────────────┘
```

AvanLink Design Flow

Figure B-3 is an overview of the process involved in creating a design with AvanLink.

Figure B-3: AvanLink Design Flow

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

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.

Figure B-6: AvanLink for Falcon Framework with Design Architect

Environment

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.

Figure B-7: AvanLink-DA in a Falcon Framework Environment

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

Figure B-10: AvanLink-DA Library Migration Function

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.

Avant!

Appendix C

Ideal and Lumped 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) 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_{rise} = source risetime

R_{source} = source output impedance

Interconnect properties

Z0 = characteristic impedance

TD = 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.

Figure C-1: Wire Model Selection Chart

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.

Figure C-2: U Model, T Element and U Element Relationship

The T and U Elements do not support the <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:

```
**warning**: the smallest T-line delay (TD) = 0.245E-14 is too
small
Please check TD, L and SCALE specification
```

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

Ideal Transmission Line	Lossy Transmission Line
lossless	includes loss effects
used with voltage sources	used with buffer drivers
no limit on input risetime	prefiltering necessary for fast rise
less CPU time for long delays	less CPU time for short delays
differential mode only	supports common mode simulation
no ground bounce	includes reference plane reactance
single conductor	up to five signal conductors allowed
AC and transient analysis	AC and transient analysis

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

Ideal Element Circuit   Lossy Element Circuit

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.

Figure C-4: U Model Geometric Structures

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

The input and output of the ideal transmission line have the following relationships:

$$Vin|_t = V(out - refout)|_{t-TD} + (iout \times Z_0)|_{t-TD}$$

$$Vout|_t = V(in - refin)|_{t-TD} + (iin \times Z_0)|_{t-TD}$$

The signal delays for ideal transmission lines are given by:

> TDeff=TD·L

If TD is given, or

> TDeff=NL/F

if NL and F are given, or

> TDeff=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

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.

Figure C-8: Dielectric and Reference Plane Configurations

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

Parameter	Units	Default	Description
LEVEL		required	(=3) required for lossy transmission lines model
ELEV		required	Electrical model (=1 for geometry)
DLEV			Dielectric model (=0 for sea, =1 for microstrip, =2 stripline, =3 overlay; default is 1)
PLEV		required	Transmission line physical model (=1 for planar)
LLEV			Omit or include the reference plane inductance (=0 to omit, =1 to include; default is 0)
NL			Number of conductors (from 1 to 5)
WD	m		Width of each conductor
HT	m		Height of all conductors

Table C-2: U Element Physical Parameters

Parameter	Units	Default	Description
TH	m		Thickness of all conductors
THB	m		Reference plane thickness
TS	m		Distance between reference planes for stripline (default for DLEV=2 is 2 HT + TH. TS is not used when DLEV=0 or 1)
SP	m		Spacing between conductors (required if NL > 1)
KD			Dielectric constant
XW	m		Perturbation of conductor width added (default is 0)
CEXT	F/m		External capacitance between reference plane and ground. Only used when LLEV=1, this overrides the computed characteristic.
CMULT		1	Dielectric constant of material between reference plane and ground (default is 1 – only used when LLEV=1)
HGP	m		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).
CORKD			Perturbation multiplier for dielectric (default is 1)
WLUMP		20	Number of lumps per wavelength for error control
MAXL		20	Maximum number of lumps per element

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: U Element Conductor Dimensions

Loss Parameters

Loss parameters for the U model are shown in Table C-3.

Table C-3: U Element Loss Parameters

Parameter	Units	Description
RHO	ohm·m	Conductor resistivity (default is rho of copper, 17E-9 ohm·m)
RHOB	ohm·m	Reference plane resistivity (default value is for copper)
NLAY		Number of layers for conductor resistance computation (=1 for DC resistance or core resistance, =2 for core and skin resistance at skin effect frequency)
SIG	mho/m	Dielectric conductivity

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 C-4: Recommended Ranges

Parameter	Min	Max
NL	1	5
KD	1	24
WD/HT	0.08	5
TH/HT	0	1
TH/WD	0	1
SP/HT	0.15	7.5
SP/WD	1	5

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

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*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*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 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

Name (Alias)	Units	Default	Description
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 $0 \leq THK2 < 3 \cdot HT$ (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 $TS = TH + 2 \cdot HT$ (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

Name (Alias)	Units	Default	Description
RHO	ohm·m	17E-9	Resistivity of conductor material – defaults to value for copper
RHOB	ohm·m	rho	Resistivity of reference plane material
SIG1(SIG)	mho/m	0.0	Conductivity of dielectric
KD1(KD)	—	4.0	Relative dielectric constant of dielectric
KD2		KD	Relative dielectric constant of overlay dielectric for DLEV=3 $$1 < KD1 < 4 \cdot KD$$
CORKD	—	1.0	Correction multiplier for KD

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

Geometric Coax Parameters

Name (Alias)	Units	Default	Description
RA	m	req	Outer radius of inner conductor
RB	m	req	Inner radius of outer conductor (shield)
RD	m	ra+rb	Outer radius of outer conductor (shield)
HGP	m	RD	Distance from shield to SPICE ground
RHO	ohm·m	17E-9	Resistivity of conductor material – defaults to value for copper
RHOB	ohm·m	rho	Resistivity of shield material
SIG	mho/m	0.0	Conductivity of dielectric
KD	—	4.0	Relative dielectric constant of dielectric
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 shield to SPICE ground – used only to override HGP and CMULT computation
SHTHK	m	2.54E-4	Coaxial shield conductor thickness

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

Figure C-18: Geometric Twinlead, DLEV=2, Shielded

Geometric Twinlead Parameters (ELEV=1)

Name (Alias)	Units	Default	Description
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

Name (Alias)	Units	Default	Description
TS2	m	TS1	Insulation thickness on second conductor
TS3	m	TS1	Insulation thickness of spacer between conductor
The following parameters apply to shielded twinlead:			
RHOB	ohm· m	rho	Resistivity of shield material (if present)
OD1	m	req.	Maximum outer dimension of shield
SHTHK	m	2.54E-4	Twinlead shield conductor thickness

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

Figure C-20: ELEV=2 Model Keywords for Conductor PLEV=1

	Ref. plane	line 1	line 2	line 3	line 4	line 5	
HSPICE ground	CPR GPR	CP1 GP1	CP2 GP2	CP3 GP3	CP4 GP4	CP5 GP5	
Ref. plane	RRR LRR	CR1 GR1 LR1	CR2 GR2 LR2	CR3 GR3 LR3	CR4 GR4 LR4	CR5 GR5 LR5	
line 1		R11 L11	C12 G12 L12	C13 G13 L13	C14 G14 L14	C15 G15 L15	LLEV=1 parameter only
line 2			R22 L22	C23 G23 L23	C24 G24 L24	C25 G25 L25	
line 3				R33 L33	C34 G34 L34	C35 G35 L35	
line 4					R44 L44	C45 G45 L45	
line 5						R55 L55	

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:

$$CX = \sum CXY$$

$$X \neq Y$$

$$Cjk \quad = CX \text{ on diagonal}$$
$$= -CXY \text{ off diagonal}$$

The matrices for the example given above provide the following "O.C.G." capacitances:

$$C1 = CR1 + C12 + C13$$
$$C2 = CR2 + C12 + C23$$
$$C3 = CR3 + C13 + C23$$
$$CR = CR1 + CR2 + CR3 + CPR$$

Also, the Maxwell matrix is given as:

$$jk = \begin{bmatrix} CR & -CR1 & -CR2 & -CR3 \\ -CR1 & C1 & -C12 & -C13 \\ -CR2 & -C12 & C2 & -C23 \\ -CR3 & -C13 & -C23 & C3 \end{bmatrix}$$

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:

$$CPR = sum\ (Cjk),\ j=R,\ k=R:3$$

$$CP1 = sum\ (C1k),\ j=1,\ k=R:3$$

$$CP2 = sum\ (C2k),\ j=2,\ k=R:3$$

$$CP3 = sum\ (C3k),\ j=3,\ k=R:3$$

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

Name (Alias)	Units	Default	Description
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 $\sqrt{\text{(frequency)}}$ 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

Input Parameters	Basis of Computation
ZK, DELAY, DELEN, CAPL, CLEN	redundant. Discard CAPL and CLEN.
ZK, VREL, CAPL, CLEN	redundant. Discard CAPL and CLEN.
ZK, DELAY, DELEN	VREL=DELEN/(DELAY · CLIGHT)
ZK and *VREL*	C=1/(ZK · VREL · CLIGHT) L=ZK/(VREL · CLIGHT)
ZK, CAPL, CLEN	C=CAPL/CLEN L=C · ZK^2
CAPL, CLEN, DELAY, DELEN	VREL=DELEN/(DELAY · CLIGHT)
CAPL, CLEN, VREL	LC=CAPL/CLEN LL=1/(C · $VREL^2$ · $CLIGHT^2$)

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.

Table C-6: Input Attenuation Variations

AT1 Input	Computation of attenuation factor and linear resistance
AT1 = −2.4dB	v(out)/v(in) = 0.758 = $10^{(+AT1/20)}$ (for dB < 0)
AT1 = +2.4dB	v(out)/v(in) = 0.758 = $10^{(-AT1/20)}$ (for dB > 0)
AT1 = 1.318	v(out)/v(in) = 0.758 = 1/AT1 (for ATl < 1)
AT1 = 0.758	v(out)/v(in) = 0.758 = AT1 (for ATl > 1)

The attenuation factor is used to compute the exponential loss parameter and linear resistance.

$$\alpha = \frac{\ln((v(in))/(v(out)))}{ATlin}$$

$$R = 2 \cdot \alpha \cdot \sqrt{(LL)/(LC)}$$

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.

Figure C-22: Three Coupled Striplines (PCB Scale)

Figure C-23: Schematic Using the Three Coupled Striplines U Model

The Star-Hspice input file for the simulation is shown below.

```
* Stripline circuit
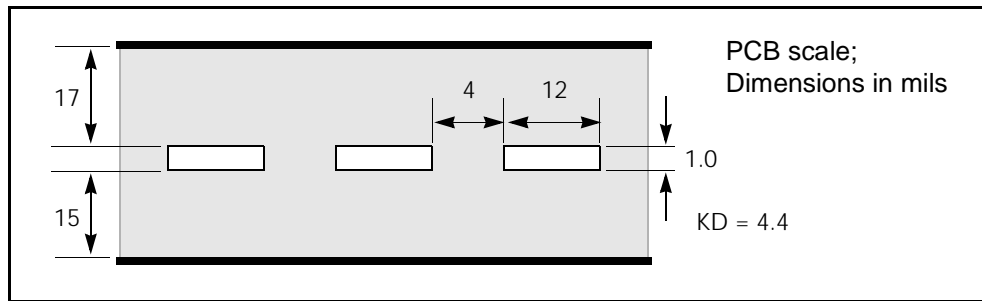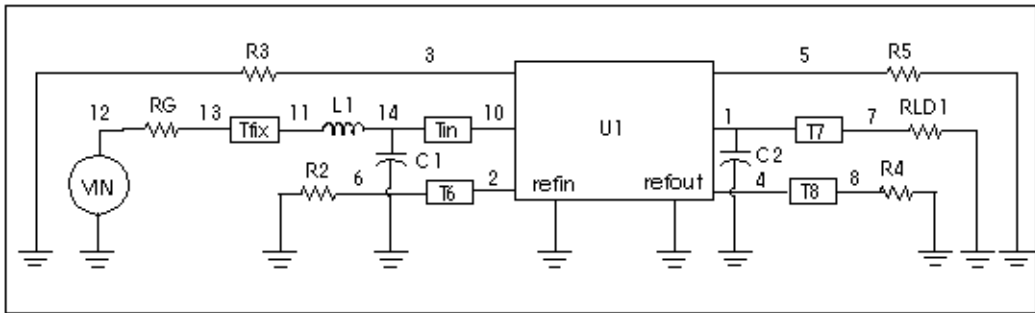
.Tran 50ps 7.5ns
.Options Post NoMod Accurate Probe Method=Gear
VIN 12 0 PWL 0 0v 250ps 0v 350ps 2v
L1 14 11 2.5n
C1 14 0 2p
Tin 14 0 10 0 ZO=50 TD=0.17ns
Tfix 13 0 11 0 ZO=45 TD=500ps
RG 12 13 50
RLD1 7 0 50
C2 1 0 2p
U1 3 10 2 0 5 1 4 0 USTRIP L=0.178
T6 2 0 6 0 ZO=50 TD=0.17ns
T7 1 0 7 0 ZO=50 TD=0.17ns
```

```
T8 4 0 8 0 ZO=50 TD=0.17ns
R2 6 0 50
R3 3 0 50
R4 8 0 50
R5 5 0 50
.Model USTRIP U LEVEL=3 PLev=1 Elev=1 Dlev=2 Nl=3 Ht=381u
Wd=305u
+ Th=25u Sp=102u Ts=838u Kd=4.7
.Probe v(13) v(7) v(8)
.End
```

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

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: Measured Versus Computed Forward 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)

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: Schematic Using the Three Coupled Lines U Model

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.

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.

Figure C-33: Computed Furthest Coupled-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          ****
  names  values   units     names  values   units     names  values    units
  -----  ------   -----     -----  ------   -----     -----  ------    -----
  --- u-model control parameters ---
 maxl=  20.00    #lumps   wlump= 20.00                 elev=  1.00
 plev=   1.00                 llev= 0.                 nlay=  2.00     #layrs
   nl=   3.00    #lines       nb= 1.00     #refpl
  --- begin type specific parameters ---
 dlev=   0.    #            kd=   3.90             corkd=   1.00
  sig=   0.    mho/m       rho=  28.00n ohm*m      rhob=  28.00n ohm*m
   xw=   0.    meter        wd=   5.00u meter        ht=   2.00u meter
   th=   1.00u meter       thb=   1.42u meter      skin=  10.31u meter
skinb=  10.31u meter       wd2=   5.00u meter       ht2=   2.00u meter
  th2=   1.00u meter       sp1=  15.00u meter       wd3=   5.00u meter
  ht3=   2.00u meter       th3=   1.00u meter       sp2=  15.00u meter
  cr1= 170.28p f/m         gr1=   0.    mho/m       l11= 246.48n h/m
  cr2= 168.25p f/m         gr2=   0.    mho/m       c12=   5.02p f/m
  g12=   0.    mho/m       l12=   6.97n h/m         l22= 243.44n h/m
  cr3= 170.28p f/m         gr3=   0.    mho/m       c13= 649.53f f/m
  g13=   0.    mho/m       l13=   1.11n h/m         c23=   5.02p f/m
  g23=   0.    mho/m       l23=   6.97n h/m         l33= 246.48n h/m
  --- two layer (skin and core) parameters ---
  rrs=   1.12k ohm/m       rrc=   0.    ohm/m       r1s=   5.60k ohm/m
  r1c=   0.    ohm/m       r2s=   5.60k ohm/m       r2c=   0.    ohm/m
  r3s=   5.60k ohm/m       r3c=   0.    ohm/m
```

U Element Parameters

cij	coupling capacitance from conductor i to conductor j (positive)
cr$j$	self capacitance/m of conductor j to the reference plane
cpr	capacitance of the reference plane to the HSPICE ground
gij	conductance/m from conductor i to conductor j (zero if sig=0)
gr$j$	conductance/m from conductor j to the reference plane (0 if sig=0)
gpr	conductance from the reference plane to the HSPICE ground, always=0
ht$i$	height of conductor i above the reference plane (only ht is input, all heights are the same)

lr$i$	inductance/m from conductor i to the reference plane
lrr	inductance/m of the reference plane
lij	inductance/m from conductor i to conductor j
ljj	self inductance/m of conductor j
rrc	core resistance/m of the reference plane (if NLAY = 2, zero of skin depth > 90% of thb)
rrr	resistance/m of the reference plane (if NLAY = 1)
rrs	skin resistance/m of the reference plane (if NLAY = 2)
ris	skin resistance/m of conductor i (if NLAY = 2)
ric	core resistance/m of conductor i (if NLAY = 2, zero if skin depth > 50% of th)
rjj	resistance/m of conductor j (if NLAY = 1)
skin	skin depth
skinb	skin depth of the reference plane
sp$i$	spacing between conductor i and conductor i+1 (only sp is input, all spacings are the same)
th$i$	thickness of conductor i (only th is input, all thicknesses are the same)
wd$i$	width of conductor i (only wd is input – all widths are the same)

The total conductor resistance is indicated by rjj when NLAY = 1, or by ris + ric 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

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:

$$
\begin{bmatrix}
C_{ii} & \dots & C_{ji} \\
\vdots & & \\
C_{ij} & & C_{jj}
\end{bmatrix}
$$

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

Capacitance (pF/m)	176　-5.02　-0.65 -5.02　178　-5.02 -0.65　-5.02　176
Inductance (nH/m)	246　6.97　1.11 6.97　243　6.97 1.11　6.97　246

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,

$$
\begin{aligned}
C_{11} &= C_{R1} + C_{12} + C_{13} \\
&= 170.28 + 5.02 + 0.65 \\
&= 175.95, \text{ or } 176 \text{ pF/m}
\end{aligned}
$$

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

Table C-8: Capacitance and Inductance for the Single Line MCM-Scale Stripline

Capacitance (pF/m)	164.4
Inductance (nH/m)	236.5

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

Capacitance (pF/m)	59 −19 −2.5 −0.8 −0.4
	−19 69 −18 −2.2 −0.8
	−2.5 −18 69 −18 −2.5
	−0.8 −2.2 −18 69 −19
	−0.4 −0.8 −2.5 −19 57
Inductance (nH/m)	676 309 179 116 81
	309 641 297 174 116
	179 297 637 297 179
	116 174 297 641 309
	81 116 179 309 676

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
```

```
+ 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=30.8p clen=1ft vrel=0.66
+ fr1=100meg at1=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:

```
.MODEL 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

top signal layer

ground plane

middle signal layer

middle signal layer

ground/supply plane

bottom signal layer

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
```

```
*
.model rg54a/u  u         LEVEL=3     plev=2        elev=3
+              Zk=58      vrel=.66
+             fr1=100meg  at1=3.1db   atlen=100ft
*
.model rg15/u   u         LEVEL=3     plev=2        elev=3
+              Zk=53.5    vrel=.66
+             fr1=100meg  at1=4.1db   atlen=100ft
*
.model rg53/u   u         LEVEL=3     plev=2        elev=3
+              Zk=53.5    vrel=.66
+             fr1=100meg  at1=4.1db   atlen=100ft
*
.model rg58a/u  u         LEVEL=3     plev=2        elev=3
+              Zk=50      vrel=.66
+             fr1=100meg  at1=5.3db   atlen=100ft
*
.model rg58c/u  u         LEVEL=3     plev=2        elev=3
+              Zk=50      vrel=.66
+             fr1=100meg  at1=5.3db   atlen=100ft
*
.model rg59b/u  u         LEVEL=3     plev=2        elev=3
+              Zk=75      vrel=.66
+             fr1=100meg  at1=3.75db    atlen=100ft
*
.model rg62/u   u         LEVEL=3     plev=2        elev=3
+              Zk=93      vrel=.84
+             fr1=100meg  at1=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
+             fr1=57meg   at1=1.7db   atlen=100ft
*
.model tw/un    u         LEVEL=3     plev=3        elev=3
*             Unshielded TV type twinlead
+              Zk=300     vrel=.733
+             fr1=100meg  at1=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:

TDeff = RISETIME/20

where: TDeff = 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 = value

At the threshold, two lumps are used. Above the threshold, the number of lumps is determined by:

number of lumps = minimum of 20 or [1+(TDeff/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{MAXL, [1+(TDeff⁄RISETIME)*WLUMP]}
Threshold: TDeff = RISETIME⁄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

Figure C-39: Circuit Diagram of an RC Filter

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.
*
*  cuttoff 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'
```

```
*
.UNPROT
.EOM
```

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.

$$RFLT|_{DC} = 0.99999 \cdot RFLT$$

$$RFLT|_{FC} = 0.97 \cdot RFLT$$

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.

Figure C-40: Multistage RC Filter Input and Output

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

Figure C-43: Impedance from TDR at Input

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
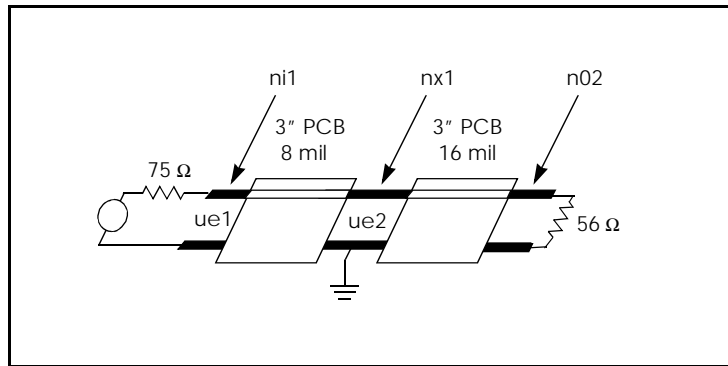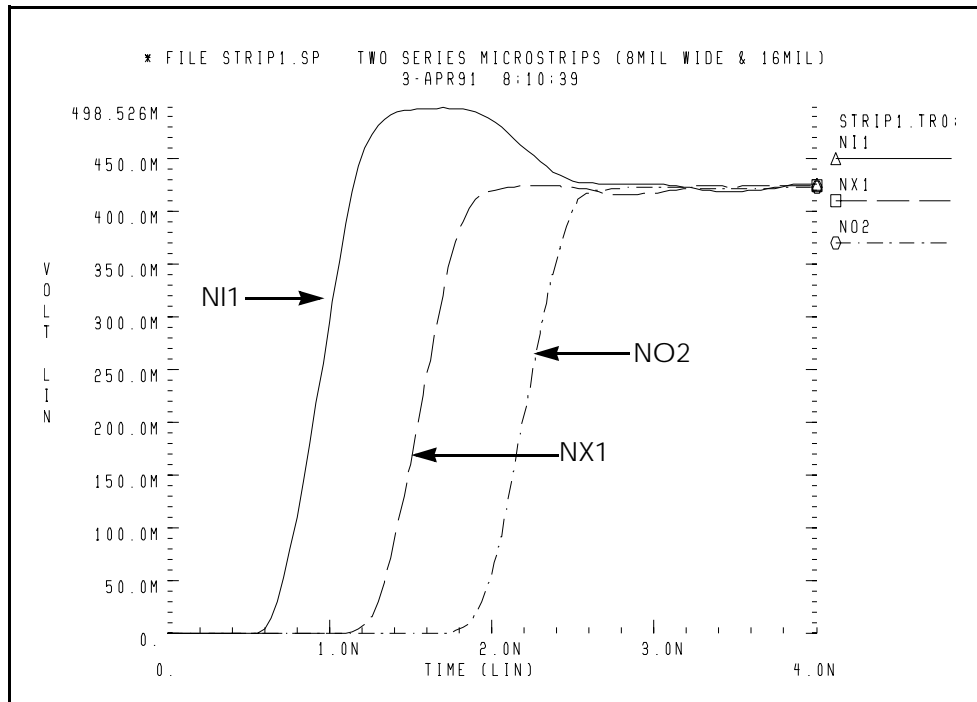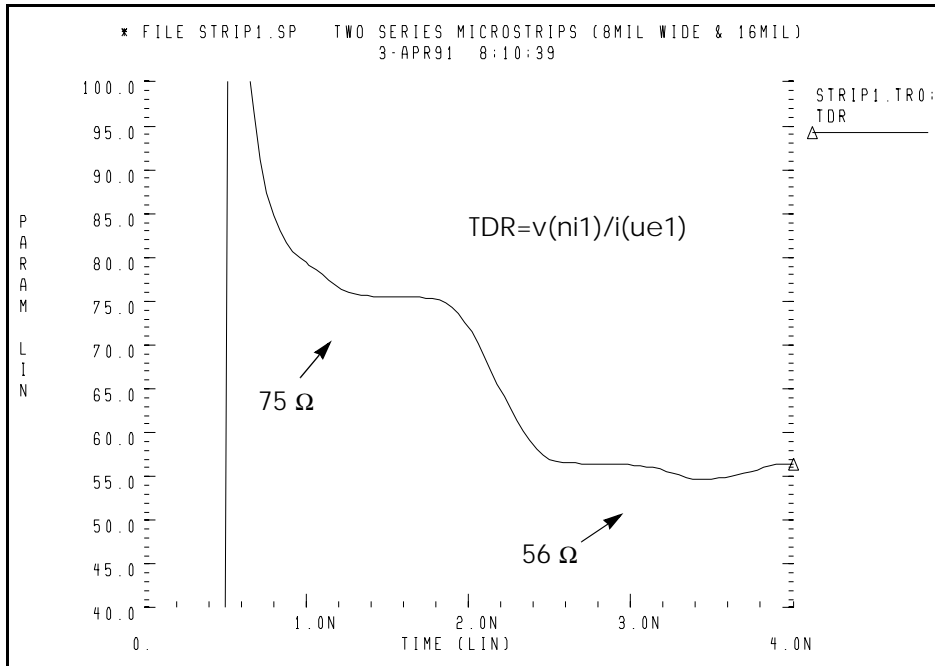
.option acct post list
```

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).

$$v = \frac{1}{\sqrt{L_L C_L}}$$

and

$$Z_0 = \sqrt{\frac{L_L}{C_L}}$$

L_L = inductance per length

C_L = 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.

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, Δx, of the transmission line has a small capacitance associated with it, ΔC, which is the capacitance per length, C_L, times the Δx:

$$\Delta C = C_L \Delta x$$

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, Δx, is charged up in a time Δt:

$$\Delta t = \frac{\Delta x}{v}$$

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):

$$= \frac{V}{I} = \frac{V}{C_L v V} = \frac{1}{C_L v}$$

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:

$$V_L = \frac{\mu_0}{2\pi} Ln\left(\frac{s}{r} - \frac{r}{s}\right)\frac{dI}{dt} \quad \text{[V in mV/inch, l in mA, t in ns]}$$

From this expression, the effective inductance per length of one wire is found to be:

$$L_L = \frac{\mu_0}{2\pi} Ln\left(\frac{s}{r} - \frac{r}{s}\right) \approx 5Ln\left(\frac{s}{r}\right)(s \gg r) \quad \text{[nH/inch]}$$

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:

$$V_1 = L_{11}\frac{dI_1}{dt} + L_{12}\frac{dI_{12}}{dt} + L_{13}\frac{dI_3}{dt} + L_{14}\frac{dI_4}{dt}$$

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:

$$C_L = \frac{85}{Z_0}\sqrt{\varepsilon_r} \quad [\text{pF/inch}]$$

The inductance per length of the signal line (L_L) is:

$$L_L = 0.085 Z_0 \sqrt{\varepsilon_r} \quad [\text{nH/inch}]$$

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
 C11 = 3.8 pF/inch, L11 = 9.6 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:

$$V_1 = L11\frac{dI_1}{dt} + L12\frac{dI_2}{dt} + L13\frac{dI_3}{dt}$$

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_{clock}), its frequency (F_{clock}), and a risetime (τ_{edge}). Figure C-46 illustrates these features.

Figure C-46: Clock Waveform

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:

$$BW = \frac{0.35}{\tau_{edge}} \text{ or } \tau_{edge} = \frac{0.35}{BW}$$

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:

$$T_{period} \approx 2\tau_{edge}$$

For a complex system such as a microprocessor board, the period might be 15 times as long as the risetime:

$$T_{period} \approx 15\tau_{edge}$$

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:

$$F_{clock} = \frac{1}{T_{clock}}$$

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:

$$F_{clock} = \frac{1}{T_{clock}} = \frac{1}{15\tau_{edge}} = \frac{BW}{15 \cdot 0.35} \approx \frac{1}{5}BW \text{ or } BW \approx 5F_{clock}$$

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

F_{clock}	Clock frequency in units of frequency such as Hz
T_{clock}	Clock period in units of time such as secs or ns
t_{edge}	Rise or fall time in units of time such as sec or ns
BW	Bandwidth in units of frequency such as Hz or MHz
F	A repetitive frequency in units of Hz or MHz
f	A sine wave frequency in units of Hz or MHz
ω	An angular frequency in units of radians/sec
t	Time or a conductor thickness, units of sec or length
V(t)	Instantaneous voltage in units of volts
I(t)	Instantaneous current in units of amps or mA
Z(t)	Instantaneous impedance in units of ohms
$Z(\omega)$	Frequency domain, complex impedance in units of ohms
C	Capacitance in units of Farads or microFarads
R	Resistance in units of ohms
L	Inductance in units of Henrys or nanoHenries
v	Speed of light in the medium in units of length/time
d	A length in units such as inches
TD	Time delay in units of time such as sec or nsec
C_L	Capacitance per length in units such as pF/inch
L_L	Inductance per length in units such as nH/inch
R_L	Resistance per length in units such as ohms/inch
ε_r	Relative dielectric constant, $\varepsilon/\varepsilon_0$, dimensionless

Table C-10: Transmission Line Terms

r	Reflection coefficient, dimensionless
DZ	A small change in characteristic impedance
G_L	Conductance per length units of Mhos (Siemens)/length
tan(δ)	Dissipation factor of a material, dimensionless
d	Skin depth of a conductor, units of length such as meter
ρ	Resistivity of a conductor, units of ohm-length
μ_0	Permeability of free space = 4 p x 10-7 Henry/meter
α	Attenuation per length, units of dB/len or nepers/len
w	A conductor width
n	Number of squares in a planar conductor, dimensionless
R_{sq}	Sheet resistance of a planar conductor, units: ohms/sq
ε_0	Permittivity of free space=0.225 pF/inch=0.0885 pF/cm
ε_{eff}	Effective dielectric constant due to mixed dielectrics
h	A dielectric thickness in units of length such as mils
R	Resistance in ohms

Relationships and Rules of Thumb

Time and Frequency Relationships

$$F_{clock} = \frac{1}{T_{clock}}$$

$$BW = \frac{0.35}{\tau_{edge}} \text{ or } \tau_{edge} = \frac{0.35}{BW}$$

$$T_{period} \approx 15\tau_{edge}$$

$$F_{clock} \approx \frac{1}{5}BW \text{ or } BW \approx 5F_{clock}$$

Transmission Line Effects

Transmission line analysis recommended for:

$$BW > \frac{v}{10d} \ = \ \frac{1}{10 \cdot TD}$$

$$\tau_{edge} < \frac{5d}{v} \ = \ 5 \cdot TD$$

On FR4 material,

$$BW > \frac{600}{d} \qquad \text{[BW in MHz, d in inches]}$$

$$F_{clock} > \frac{120}{d} \qquad \text{[}F_{clock}\text{ in MHz, d in inches]}$$

$$\tau_{edge} < \frac{d}{7.5} \qquad \text{[}\tau_{edge}\text{ in ns, d in inches]}$$

Intrinsic Properties

$$Z_0 \ = \ \frac{1}{vC_L} \ = \ \sqrt{\frac{L_L}{C_L}} \qquad \text{[}C_L\text{ is in pF/inch}$$

$$v \ = \ \frac{1}{\sqrt{L_L C_L}}$$

$$C_L \ = \ \frac{1}{vZ_0} \ = \ \frac{\sqrt{\varepsilon_r}}{cZ_0} \ = \ \frac{85}{Z_0}\sqrt{\varepsilon_r} \quad \text{[pF/inch]}$$

$$L_L \ = \ \frac{1}{C_L v^2} \ = \ \frac{7.3\varepsilon_r}{C_L} \quad \text{[}L_L\text{ in nH/inch, }C_L\text{ in pF/inch]}$$

Typical polymers

$$TD = \frac{d}{6} \qquad \text{[TD in ns, d in inches]}$$

$$C_L = \frac{170}{Z_0} \qquad \text{[pF/inch]}$$

$$L_L = 0.172 Z_0 \text{ [nH/inch]}$$

50 ohm lines

$$C_L = 1.7 \sqrt{\varepsilon_r} \quad \text{[pF/inch = 3.4 pF/inch (typical polymer)]}$$

$$L_L = 4.3 \sqrt{\varepsilon_r} \quad \text{[nH/inch = 8.6 nH/inch (typical polymer)]}$$

Transmission line of length d

$$C = C_L d = \frac{TD}{Z_0}$$

$$L = L_L d = Z_0 TD$$

Reflections

Reflection coefficient from Z_1 to Z_2:

$$r = \frac{Z_2 - Z_1}{Z_2 + Z_1}$$

Reflection from a ΔZ, of short length with time delay, TD, $\tau_{edge} > $ TD:

$$r = \frac{\Delta Z}{2 Z_0} \left(\frac{TD}{\tau_{edge}} \right)$$

Reflection from a series lumped L surrounded by a transmission line:

$$r = \frac{L}{2Z_0 \tau_{edge}}$$

Reflection from a lumped C load to ground, on a transmission line:

$$r = \frac{CZ_0}{2\tau_{edge}}$$

Loss and Attenuation

Skin depth of a conductor:

$$\delta = \sqrt{\frac{\rho}{\pi\mu_0 f}} = 500 \cdot \sqrt{\frac{\rho}{f}}$$

[δ in meters, ρ in ohm·meter, *f* in Hz: in copper, at *f* = 1e+9 and ρ =1.7e-8 ohm·meter, δ = 2.0e-6]

Low loss approximation for attenuation per length:

$$\alpha = \frac{1}{2}\left(\frac{R_L}{Z_0} + G_L Z_0\right) \quad \text{[nepers/length]}$$

$$\alpha = 4.34\left(\frac{R_L}{Z_0} + G_L Z_0\right) \text{[dB/length]}$$

Attenuation per length due to just dielectric loss:

$$\alpha = 2.3 f \tan(\delta) \sqrt{\varepsilon_r} \quad \text{[dB/inch, f in GHz]}$$

$$\alpha \approx 0.05 \qquad \text{[dB/inch for FR4 at 1 GHz]}$$

Attenuation per length due to metal, $t < \delta$:

$$\alpha = 43.4 \frac{\rho}{twZ_0} \text{ [dB/inch, with } \rho \text{ in ohm·meter, t in microns, w in mils]}$$

For 1 ounce copper microstrip, 5 mils wide, 50 ohm:

$$\alpha \ = \ 0.01 \qquad \text{[dB/inch]}$$

Attenuation per length due to metal, 50 ohm line, skin depth limited, $t > \delta$:

$$\alpha \ = \ 0.55 \frac{\sqrt{\rho f}}{w} \ \ \text{[dB/inch, with } \rho \text{ in ohm·meter, w in mils, f in GHz]}$$

For 1 ounce copper microstrip, 5 mils wide, 50 ohm, at 1 GHz:

$$\alpha \ = \ 0.15 \qquad \text{[dB/inch]}$$

Physical Design Quantities

For a planar interconnect:

Sheet resistance:

$$R_{sq} \ = \ \frac{\rho}{t} \qquad \text{[ohm/sq]}$$

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:

$$R_L \ = \ 10 \frac{\rho}{tw} \qquad \text{[ohm/inch, } \rho \text{ in ohm·meter, t in microns, w in mils]}$$

Parallel plate, no fringe fields:

$$C_L \ = \ 0.225 \varepsilon_r \left(\frac{w}{h}\right) \qquad \text{[pF/inch]}$$

Microstrip capacitance per length good to ~20%:

$$C_L = 0.45\varepsilon_r\left(\frac{w}{h}\right) \qquad [pF/inch]$$

Microstrip capacitance per length, good to ~5%:

$$C_L = \frac{1.41\varepsilon_{eff}}{\ln\left(\frac{8h}{w} + \frac{w}{4h}\right)}$$

$$\varepsilon_{eff} = \left(\frac{\varepsilon_r + 1}{2}\right) + \left(\frac{\varepsilon_r - 1}{2}\right)\left(1 + \frac{10h}{w}\right)^{-\frac{1}{2}}$$

Stripline capacitance per length good to ~20%:

$$C_L = 0.675\varepsilon_r\left(\frac{w}{h}\right) \qquad [pF/inch]$$

Stripline capacitance per length, good to ~5%:

$$C_L = \frac{0.9\varepsilon_r}{\ln\left(1 + \frac{2h}{w}\right)} \qquad [pF/inch]$$

Inductance per length of one wire in a pair of two parallel wires:

(r = radius, s = center to center spacing, s » r)

$$L_L = 5Ln\left(\frac{s}{r}\right) \quad [nH/inch]$$

Inductance per length for a circular loop:

$$L_L = 26 \qquad [nH/inch\ of\ perimeter]$$

Inductance per length of controlled impedance line, when the return line is a reference plane:

$$L_L = 0.086 Z_0 \sqrt{\varepsilon_r} = \frac{7.3 \varepsilon_r}{C_L} \quad [\text{nH/inch with } C_L \text{ in pF/inch}]$$

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 (σ) (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

$$R_L = \frac{\rho}{wt}$$

　　ρ = 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 (δ). When the signal is a sine wave, the skin depth depends on the conductor's resistivity (ρ), and the sine wave frequency of the current (f):

$$\delta \ = \ \sqrt{\frac{\rho}{\pi\mu_0 f}} \ = \ 500 \cdot \sqrt{\frac{\rho}{f}} \quad [\,\delta \ \text{in meters, r in ohm·meter, f in Hz}\,]$$

A real signal has most of its energy at a frequency of $1/t_{period}$, where t_{period} is the average period. Because most of the loss occurs at this frequency, as a first approximation >> should be used to compute the skin depth. In practice, as mentioned previously, approximating tperiod by $15(\tau_{edge})$ works well. With a 1 ns rise time, the skin depth of copper is 8 microns, assuming $15(\tau_{edge})$ is used for $1/f_{skin}$. 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 (σ), 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:

$$G_L = \sigma \frac{C_L}{\varepsilon_0 \varepsilon_r}$$

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:

$$G_L = 2\pi f \tan(\delta) C_L$$

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:

Figure C-48: Lossy Transmission Line 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:

$$Z_0 = \sqrt{\frac{R_L + j\omega L_L}{G_L + j\omega C_L}}$$

and the propagation phase term, γ, is:

$$\Upsilon = \alpha + j\beta = \sqrt{(R_L + j\omega L_L)(G_L + j\omega C_L)}$$

In the propagation phase term, β is related to the phase velocity by:

$$v = \frac{\omega}{\beta}$$

To first order, when $R_L \ll \omega L_L$ and $G_L \ll \omega C_L$, the characteristic impedance and phase velocity (*v*) are unchanged from their lossless values. However, a new term, the attenuation per length (α) is introduced.

The attenuation per length is approximately:

$$\alpha = \frac{1}{2}\left(\frac{R_L}{Z_0} + G_L Z_0\right) \quad \text{[nepers/length]}$$

$$\alpha = 4.34\left(\frac{R_L}{Z_0} + G_L Z_0\right) \text{[dB/length]}$$

The total attenuation (αd) determines the fraction of the signal amplitude that remains after propagating the distance (d). When α has the units of dB/length, the fraction of signal remaining is:

$$10^{\left(\frac{-\alpha d}{20}\right)}$$

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:

$$\alpha_{\text{dielectric}} = 2.3 f \tan(\delta) \sqrt{\varepsilon_{\text{eff}}} \quad [\text{dB/inch, f in GHz}]$$

The worst case and highest attenuation per length is exhibited by FR4 boards, with tan(δ) 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

Handbook of Electronics Calculations for Engineers and Technicians, McGraw Hill, pages 18-29, 18-23.

Avant!

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:

$$LMIN + XLREF \leq L + XL < LMAX + XLREF$$

$$WMIN + XWREF \leq W + XW < WMAX + XWREF$$

(If XLREF is not specified, XLREF is set to XL. If XWREF is not specified, XWREF 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.

Figure D-1: Automatic Model Selector 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:

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.*nnn* that fits within the length and width ranges given in the element statement. Because the length given in the element statement (5 µm) is not within the 1 to 4 µm 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	AD600J	AD602	AD602J	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

AD712T	AD713	AD713A	AD713B	AD713J
AD713K	AD713S	AD713T	AD734	AD734A
AD734B	AD734S	AD743	AD743A	AD743B
AD743J	AD743K	AD743S	AD744	AD744A
AD744B	AD744C	AD744J	AD744K	AD744S
AD744T	AD745	AD745A	AD745B	AD745J
AD745K	AD745S	AD746	AD746A	AD746B
AD746J	AD746S	AD780	AD780A	AD780B
AD780S	AD797	AD797A	AD797B	AD797S
AD810	AD810A	AD810S	AD811	AD812
AD812A	AD813	AD813A	AD817	AD817A
AD818	AD818A	AD820	AD826	AD826A
AD828	AD828A	AD829	AD829A	AD829J
AD829S	AD830	AD830A	AD830J	AD830S
AD840	AD840J	AD840K	AD840S	AD843
AD843A	AD843B	AD843J	AD843K	AD843S
AD844	AD844A	AD844B	AD844S	AD845
AD845A	AD845B	AD845J	AD845K	AD845S
AD846	AD846A	AD846B	AD846S	AD847
AD847A	AD847J	AD847S	AD848	AD848A
AD848J	AD848S	AD9617	AD9618	AD9621
AD9622	AD9623	AD9624	AD9630	ADG411
ADG411B	ADG411T	ADG412	ADG412B	ADG412T
ADG413	ADG413B	ADG413T	AMP01	AMP02
BUF04	MAT02	MAT03	MAT04	MLT04
MLT04G	OP160	OP160A	OP160F	OP160G
OP176	OP176G	OP177	OP177A	OP177B
OP177E	OP177F	OP177G	OP20	OP200
OP200A	OP200E	OP200F	OP200G	OP20B
OP20C	OP20F	OP20G	OP20H	OP21
OP213	OP215	OP215A	OP215B	OP215C
OP215E	OP215F	OP215G	OP21A	OP21E
OP21F	OP21G	OP21H	OP220	OP220A

OP220C	OP220E	OP220F	OP220G	OP221
OP221A	OP221B	OP221C	OP221E	OP221G
OP249	OP249A	OP249E	OP249F	OP249G
OP260	OP27	OP275	OP275G	OP27A
OP27B	OP27C	OP27E	OP27F	OP27G
OP282	OP282G	OP283	OP285	OP285G
OP290	OP290A	OP290E	OP290F	OP290G
OP292	OP295	OP297	OP297A	OP297E
OP297F	OP297G	OP37	OP37A	OP37B
OP37C	OP37E	OP37F	OP37G	OP400
OP400A	OP400E	OP400F	OP400G	OP400H
OP41	OP41A	OP41B	OP41E	OP41F
OP41G	OP42	OP420	OP420B	OP420C
OP420F	OP420G	OP420H	OP421	OP421B
OP421C	OP421F	OP421G	OP421H	OP42A
OP42E	OP42F	OP42G	OP43	OP43A
OP43B	OP43E	OP43F	OP43G	OP44
OP467	OP467G	OP470	OP482	OP482G
OP490	OP490A	OP490E	OP490F	OP490G
OP492	OP497	OP497A	OP497B	OP497C
OP497F	OP497G	OP61	OP64	OP77
OP77A	OP77B	OP77E	OP77F	OP77G
OP80	OP80B	OP80E	OP80F	OP80G
OP90	OP90A	OP90E	OP90F	OP90G
OP97	OP97A	OP97E	OP97F	PM1012
REF01	REF01A	REF01C	REF01E	REF01H
REF02	REF02A	REF02C	REF02D	REF02E
REF02H	REF05	REF05A	REF05B	REF10
REF10A	REF10B	SSM2017	SSM2017P	SSM2131
SSM2210	SSM2220			

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:

AD4BIT	AD8BIT	ALF155	ALF156	ALF157
ALF255	ALF347	ALF351	ALF353	ALF355
ALF356	ALF357	ALF3741	ALM101A	ALM107
ALM108	ALM108A	ALM111	ALM118	ALM124
ALM124A	ALM139A	ALM1458	ALM1558	ALM158
ALM158A	ALM201A	ALM207	ALM208	ALM208A
ALM224	ALM258	ALM258A	ALM2901	ALM2902
ALM2904	ALM301A	ALM307	ALM308	ALM308A
ALM318	ALM324	ALM3302	ALM339	ALM358
ALM358A	ALM725	ALM741	ALM747	ALM747C
AMC1458	AMC1536	AMC1741	AMC1747	ANE5534P
ANJM4558	ANJM4559	ANJM4560	AOP04	AOP07
AOP14	AOP15B	AOP16B	AT094CNS	ATL071C
ATL072C	ATL074C	ATL081C	ATL082C	ATL084C
ATL092CP	ATL094CN	AUPC1251	AUPC358	GA201
RCFILT	TLINE			

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	T2N2906	T2N2907	T2N2907A	T2N2945A
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
TMPS6595	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
OPA1013E	OPA101E	OPA102	OPA102E	OPA111
OPA111E	OPA121	OPA121E	OPA128	OPA128E
OPA177	OPA177E	OPA2107	OPA2107E	OPA2111

OPA2111E	OPA2541	OPA2541E	OPA2604	OPA2604E
OPA27	OPA27E	OPA27H	OPA27HE	OPA37
OPA37E	OPA404	OPA404E	OPA445	OPA445E
OPA501	OPA501E	OPA511	OPA511E	OPA512
OPA512E	OPA541	OPA541E	OPA602	OPA602E
OPA603X	OPA606	OPA606E	OPA620	OPA620E
OPA620X	OPA621	OPA621E	OPA621X	OPA627
OPA627E	OPA637	OPA637E	RCV420X	UAF42
UAF42E				

Comlinear Device Models

Search path: *$installdir/parts/comline*

Model names:

CLC109	CLC111	CLC400	CLC401	CLC402
CLC404	CLC406	CLC409	CLC410	CLC414
CLC415	CLC420	CLC425	CLC426	CLC428
CLC430	CLC431	CLC432	CLC501	CLC502
CLC505	CLC520	CLC522	CLC532	

Diode Models

Required element syntax: Xyyyyy anode cathode modelname

Search path: *$installdir/parts/dio*

Optional parameters: isat=value, tt=value

Model names:

D12BG11	D12BH11	D12DG11	D12DH11	D12FG11
D12FH11	D12GG11	D12GH11	D12JG11	D12JH11
D1N3016	D1N3017	D1N3018	D1N3019	D1N3020
D1N3021	D1N3022	D1N3023	D1N3024	D1N3025
D1N3026	D1N3027	D1N3028	D1N3029	D1N3030

D1N3031	D1N3032	D1N3033	D1N3034	D1N3035
D1N3036	D1N3037	D1N3038	D1N3039	D1N3040
D1N3041	D1N3042	D1N3043	D1N3044	D1N3045
D1N3046	D1N3047	D1N3048	D1N3049	D1N3050
D1N3051	D1N3821	D1N3822	D1N3823	D1N3824
D1N3825	D1N3826	D1N3827	D1N3828	D1N3829
D1N3830	D1N4001	D1N4002	D1N4003	D1N4004
D1N4005	D1N4006	D1N4007	D1N4148	D1N4149
D1N4150	D1N4370	D1N4371	D1N4372	D1N4446
D1N4447	D1N4448	D1N4449	D1N4728	D1N4729
D1N4730	D1N4731	D1N4732	D1N4733	D1N4734
D1N4735	D1N4736	D1N4737	D1N4738	D1N4739
D1N4740	D1N4741	D1N4742	D1N4743	D1N4744
D1N4745	D1N4746	D1N4747	D1N4748	D1N4749
D1N4750	D1N4751	D1N4752	D1N4753	D1N4754
D1N4755	D1N4756	D1N4757	D1N4758	D1N4759
D1N4760	D1N4761	D1N4762	D1N4763	D1N4764
D1N5221	D1N5222	D1N5223	D1N5224	D1N5225
D1N5226	D1N5227	D1N5228	D1N5229	D1N5230
D1N5231	D1N5232	D1N5233	D1N5234	D1N5235
D1N5236	D1N5237	D1N5238	D1N5239	D1N5240
D1N5241	D1N5242	D1N5243	D1N5244	D1N5245
D1N5246	D1N5247	D1N5248	D1N5249	D1N5250
D1N5251	D1N5252	D1N5253	D1N5254	D1N5255
D1N5256	D1N5257	D1N5258	D1N5259	D1N5260
D1N5261	D1N5262	D1N5263	D1N5264	D1N5265
D1N5266	D1N5267	D1N5268	D1N5269	D1N5270
D1N5271	D1N5272	D1N5333	D1N5334	D1N5335
D1N5336	D1N5337	D1N5338	D1N5339	D1N5340
D1N5341	D1N5342	D1N5343	D1N5344	D1N5345
D1N5346	D1N5347	D1N5348	D1N5349	D1N5350
D1N5351	D1N5352	D1N5353	D1N5354	D1N5355
D1N5356	D1N5357	D1N5358	D1N5359	D1N5360

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:

J108	J109	J110	J111	J112
J113	J2N3330	J2N3460	J2N3824	J2N4391
J2N4392	J2N4393	J2N4856	J2N4857	J2N5457
J2N5458	J2N5459	J2N5460	J2N5461	J2N5462
J2N5463	J2N5465	J309	J511	J557
JSJ74	JSK170	M2N6755	M2N6756	M2N6757
M2N6758	M2N6759	M2N6760	M2N6761	M2N6762
M2N6763	M2N6764	M2N6765	M2N6766	M2N6767
M2N6768	M2N6769	M2N6770	M2N6787	M2N6788
M2N6789	M2N6790	M2N6791	M2N6792	M2N6793
M2N6794	M2N6795	M2N6796	M2N6797	M2N6798
M2N6799	M2N6800	M2N6801	M2N6802	MBUZ10
MBUZ20	MBUZ23	MBUZ24	MBUZ32	MBUZ35
MBUZ36	MBUZ42	MBUZ45	MBUZ46	MBUZ60
MBUZ63	MBUZ64	MBUZ71	MBUZ72A	MBUZ74
MBUZ76	MIRF120	MIRF121	MIRF122	MIRF123
MIRF130	MIRF131	MIRF132	MIRF133	MIRF140
MIRF141	MIRF142	MIRF143	MIRF150	MIRF151
MIRF152	MIRF153	MIRF220	MIRF221	MIRF222
MIRF223	MIRF230	MIRF231	MIRF232	MIRF233
MIRF240	MIRF241	MIRF242	MIRF243	MIRF250
MIRF251	MIRF252	MIRF253	MIRF320	MIRF321
MIRF322	MIRF323	MIRF330	MIRF331	MIRF332
MIRF333	MIRF340	MIRF341	MIRF342	MIRF343
MIRF350	MIRF351	MIRF352	MIRF353	MIRF420
MIRF421	MIRF422	MIRF423	MIRF430	MIRF431
MIRF432	MIRF433	MIRF440	MIRF441	MIRF442
MIRF443	MIRF450	MIRF451	MIRF452	MIRF453
MIRF510	MIRF511	MIRF512	MIRF513	MIRF520
MIRF521	MIRF522	MIRF523	MIRF530	MIRF531
MIRF532	MIRF533	MIRF540	MIRF541	MIRF542
MIRF543	MIRF610	MIRF611	MIRF612	MIRF613

MIRF620	MIRF621	MIRF622	MIRF623	MIRF630
MIRF631	MIRF632	MIRF633	MIRF640	MIRF641
MIRF642	MIRF643	MIRF710	MIRF711	MIRF712
MIRF713	MIRF720	MIRF721	MIRF722	MIRF723
MIRF730	MIRF731	MIRF732	MIRF733	MIRF740
MIRF741	MIRF742	MIRF743	MIRF810	MIRF811
MIRF812	MIRF813	MIRF820	MIRF821	MIRF822
MIRF823	MIRF830	MIRF831	MIRF832	MIRF833
MIRF840	MIRF841	MIRF842	MIRF843	MIRF9020
MIRFF110	MIRFF111	MIRFF112	MIRFF113	MIRFF120
MIRFF121	MIRFF122	MIRFF123	MIRFF130	MIRFF131
MIRFF132	MIRFF133	MIRFF210	MIRFF211	MIRFF212
MIRFF213	MIRFF220	MIRFF221	MIRFF222	MIRFF223
MIRFF230	MIRFF231	MIRFF232	MIRFF233	MIRFF310
MIRFF311	MIRFF312	MIRFF313	MIRFF320	MIRFF321
MIRFF322	MIRFF323	MIRFF330	MIRFF331	MIRFF332
MIRFF333	MIRFF430	MIRFF431	MIRFF432	MIRFF433

Linear Technology Device Models

Search path: *$installdir/parts/lin_tech*

Model names:

113090	LF155	LF155A	LF156	LF156A
LF355	LF355A	LF356	LF356A	LF412
LF412A	LM101A	LM107	LM108	LM108A
LM10C	LM118	LM301A	LM307	LM308
LM308A	LM318	LT1001	LT1001A	LT1006
LT1006A	LT1006S8	LT1007	LT1007A	LT1008
LT1012	LT1012A	LT1012D	LT1012S8	LT1013
LT1013A	LT1013D	LT1022	LT1022A	LT1028
LT1028A	LT1037	LT1037A	LT1055	LT1055A
LT1055S8	LT1056	LT1056A	LT1056S8	LT1057

113090	LF155	LF155A	LF156	LF156A
LT1057A	LT1057S	LT1078	LT1078A	LT1097
LT1101	LT1115	LT1122	LT1178	LT1178A
LT118A	LT1220	LTC1049	LTC1050	LTC1050A
LTC1051	LTC1052	LTC1150	OP05	OP05A
OP05C	OP05E	OP07	OP07A	OP07C
OP07E	OP15A	OP15B	OP15C	OP16A
OP16B	OP16C	OP215A	OP215C	OP27A
OP27C	OP37A	OP37C	OP97	

Intel PCI Speedway Models

Search path: *$installdir/parts/pci*

Model names:

PCI_II_B	PCI_II_T	PCI_II_W	PCI_IN_W	TRACE

Signetics Device Models

Search path: *$installdir/parts/signet*

Model names:

AC109EQ	AC240EQ	AC833OD	AC86EQ	ACTINPUT
CL10X10	CL30X10	DIP14	DIP16	DIP20
DIP24	DIP28	DIP8	DIP80001	SO16
SO20	SO24	SO28	SO8	

Texas Instruments Device Models

Search path: *$installdir/parts/ti*

Model names:

ICL7652	LF347	LF351	LF353	LF411C
LF412C	LM101A	LM107	LM301A	LM307
LM308	LM318	LM324	LM348	LM358
LT1001	LT1008	LT1012	LT1013	LT1028
LT1037	LTC1052	MC1458	MC3403	NE5534
OP-07C	OP-07D	OP-07E	OP-27C	OP-27E
OP-27G	OP-37A	RC4136	RC4558	RC4559
TL022C	TL031	TL032	TL034	TL044C
TL051	TL052	TL054	TL060	TL061
TL062	TL064	TL066	TL070	TL071
TL072	TL074	TL075	TL080	TL081
TL082	TL083	TL084	TL085	TL087
TL088	TL136	TL287	TL288	TL321
TL322	TLC1078	TLC1079	TLC2201	TLC251H
TLC251L	TLC251M	TLC252C	TLC254C	TLC25L2C
TLC25L4C	TLC25M2C	TLC25M4C	TLC2652	TLC2654
TLC271H	TLC271L	TLC271M	TLC272	TLC274
TLC277	TLC279	TLC27L2	TLC27L4	TLC27L7
TLC27L9	TLC27M2	TLC27M4	TLC27M7	TLC27M9
TLE2021	TLE2022	TLE2024	TLE2061	TLE2062
TLE2064	TLE2161	UA741	UA747	UA748

Transmission Line Models

Search path: *$installdir/parts/tline*

Model names:

RCFILT	RG11_U	RG11A_U	RG15_U	RG180B_U
RG188A_U	RG53_U	RG54A_U	RG58A_U	RG58C_U
RG59B_U	RG62_U	RG62B_U	RG71_U	RG71B_U
RG9_U	RG9B_U	TW_SH_U	TW_UN_U	

Xilinx Device Models

Search path: *$installdir/parts/xilinx*

Model names:

FOUTPUT	OUTPUT	XC7236A	XC7272A	XC7336A
XIL_IOB	XIL_IOB4			

Avant!

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 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.

Figure E-1: Encrypted Library Structure

The following input file fragment from the main circuit level selects the *Fast*
library and creates two instances of the *iobuf* circuit.

```
...
.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
...
```

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
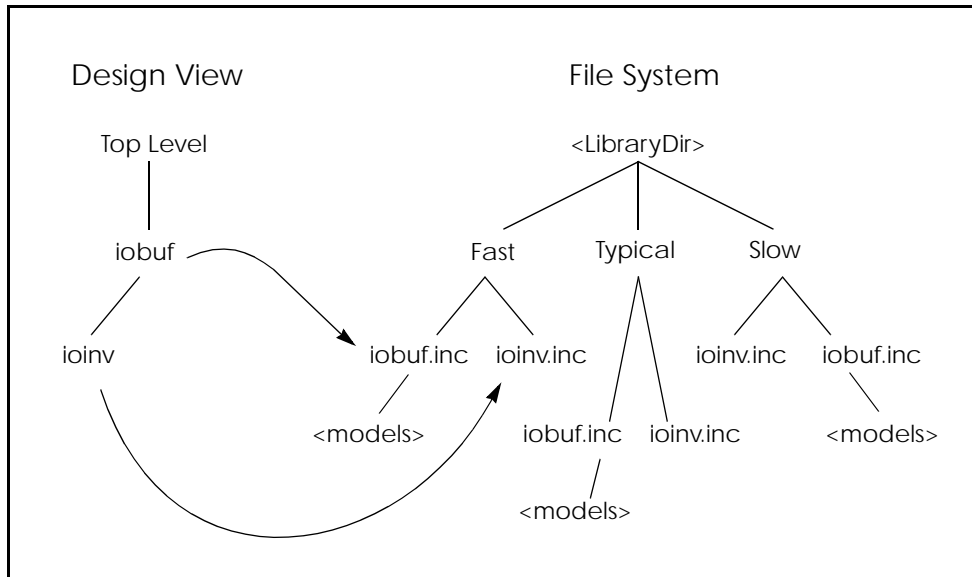
.Model pMod pmos Level=28 Vto=...     $ <FastModels>
.Model nMod nmos Level=28 Vto=...     $ <FastModels>
.UNPROTECT
cPin2 Pin2 0 Cload         $ give you some control
.Ends
```

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*.

```
* 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.

Some versions of Star-Hspice cannot decrypt files with lines longer than 80 characters. Avant! strongly recommends that *all* encrypted files be limited to an 80-character line length, because at encryption time the Star-Hspice version that the customer uses is unknown.

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
```

Avant!

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
Cincap in infilt 0.001
Rdamp infilt clamp 100
Dlow gnd clamp diode_mod
Dhigh clamp vdd diode_mod
Xinv1 clamp inv1out inverter
Rpull clamp inv1out 1x
Xinv2 inv1out inv2out inverter
Routterm 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
```

```
* model definitions
.model pmos_mod pmos level=3
.model nmos_mod nmos level=3
.model diode_mod d

* analysis specifications
.TRAN 10n 1u sweep analog_voltage lin 5 1.0 5.0

* output specifications
.probe TRAN v(in) v(clamp) v(inv1out) v(inv2out) i(dlow)
.measure TRAN falltime  TRIG v(inv2out) VAL=4.5v FALL=1
+                       TARG V(inv2out) VAL=0.5v FALL=1

.end
```

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

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 "
* "runtitle" "* example hspice netlist using a linear cmos amplifier "
* time =    0.
* temperature =   25.0000
*** BEGIN: Saved Operating Point ***
.option
+ gmindc=   1.0000p
.nodeset
```

```
+ clamp =    2.6200
+ in =    0.
+ infilt =    2.6200
+ inv1out =    2.6200
+ inv2out =    2.6199
+ vdd =    5.0000
***    END: Saved Operating Point ***
```

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:hspiceuser  HOSTNAME:hspiceserv  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.pa0

******   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
```

```
+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          iter=   2
vds,vgs,vbs =       2.45           2.93            0.
 gm,gds,gmbs,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
 ******   transient analysis              tnom=  25.000 temp=  25.000 ******
  falltime=  1.1917E-08  targ=  5.6075E-08   trig=  4.4158E-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  =    4.000E+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=  7.5424E-09  targ=  5.3989E-08   trig=  4.6447E-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  =    5.000E+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=  6.1706E-09  targ=  5.3242E-08   trig=  4.7072E-08

  meas_variable = falltime
  mean  =  16.0848n      varian =  1.802e-16
  sigma =  13.4237n      avgdev =   9.2256n
  max   =  39.1488n      min    =   6.1706n


          ***** job concluded
```

```
  ******  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


lic: Release hspice token(s)
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: FLEXlm:v5.12  USER:hspiceuser HOSTNAME:hspiceserv 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:
 init: begin read circuit files, cpu clock= 2.21E+00
      option probe
      option nomod
 init: end read circuit files, cpu clock= 2.23E+00 memory=    145 kb
 init: begin check errors, cpu clock= 2.23E+00
 init: end check errors, cpu clock= 2.24E+00 memory=    144 kb
 init: begin setup matrix, pivot=    10 cpu clock= 2.24E+00
```

```
        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= 1.01016E-07 tot_iter=     186   conv_iter=      54
 tran: time= 2.02642E-07 tot_iter=     338   conv_iter=      98
 tran: time= 3.01763E-07 tot_iter=     495   conv_iter=     145
 tran: time= 4.04254E-07 tot_iter=     668   conv_iter=     198
 tran: time= 5.02594E-07 tot_iter=     841   conv_iter=     248
 tran: time= 6.10102E-07 tot_iter=     983   conv_iter=     289
 tran: time= 7.01850E-07 tot_iter=    1161   conv_iter=     340
 tran: time= 8.01776E-07 tot_iter=    1306   conv_iter=     383
 tran: time= 9.04268E-07 tot_iter=    1481   conv_iter=     436
 tran: time= 1.00000E-06 tot_iter=    1654   conv_iter=     486
 sweep: tran tran2    end,  cpu clock=  3.71E+00 memory=    155 kb
 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 tot_iter=     143   conv_iter=      42
 tran: time= 2.01211E-07 tot_iter=     340   conv_iter=     100
 tran: time= 3.01801E-07 tot_iter=     539   conv_iter=     156
 tran: time= 4.02192E-07 tot_iter=     729   conv_iter=     211
 tran: time= 5.01997E-07 tot_iter=     917   conv_iter=     265
 tran: time= 6.01801E-07 tot_iter=    1088   conv_iter=     314
 tran: time= 7.01801E-07 tot_iter=    1221   conv_iter=     351
 tran: time= 8.01801E-07 tot_iter=    1362   conv_iter=     392
 tran: time= 9.02387E-07 tot_iter=    1515   conv_iter=     435
```

```
 tran: time= 1.00000E-06 tot_iter=    1674   conv_iter=     479
 sweep: tran tran3    end,  cpu clock=  4.57E+00 memory=    155 kb
 parameter: analog_voltage  =    4.00E+00
 dcop: begin dcop,  cpu clock=  4.57E+00
output: ./example.mt0
 sweep: tran tran4    begin, stop_t=  1.00E-06 #sweeps= 101 cpu clock=  4.58E+00
 tran: time= 1.00110E-07 tot_iter=    236   conv_iter=     70
 tran: time= 2.04376E-07 tot_iter=    475   conv_iter=    139
 tran: time= 3.07892E-07 tot_iter=    767   conv_iter=    221
 tran: time= 4.01056E-07 tot_iter=    951   conv_iter=    273
 tran: time= 5.01086E-07 tot_iter=   1250   conv_iter=    353
 tran: time= 6.00965E-07 tot_iter=   1541   conv_iter=    432
 tran: time= 7.03668E-07 tot_iter=   1805   conv_iter=    506
 tran: time= 8.01114E-07 tot_iter=   2046   conv_iter=    571
 tran: time= 9.01005E-07 tot_iter=   2308   conv_iter=    640
 tran: time= 1.00000E-06 tot_iter=   2528   conv_iter=    703
 sweep: tran tran4    end,  cpu clock=  5.83E+00 memory=    155 kb
 parameter: analog_voltage  =    5.00E+00
 dcop: begin dcop,  cpu clock=  5.83E+00
 dcop: end dcop,  cpu clock=  5.84E+00 memory=    155 kb  tot_iter=     23
 output: ./example.mt0
 sweep: tran tran5    begin, stop_t=  1.00E-06 #sweeps= 101 cpu clock=  5.84E+00
 tran: time= 1.00195E-07 tot_iter=    176   conv_iter=     47
 tran: time= 2.00617E-07 tot_iter=    431   conv_iter=    115
 tran: time= 3.00475E-07 tot_iter=    661   conv_iter=    176
 tran: time= 4.00719E-07 tot_iter=    914   conv_iter=    246
 tran: time= 5.04084E-07 tot_iter=   1157   conv_iter=    311
 tran: time= 6.00666E-07 tot_iter=   1347   conv_iter=    363
 tran: time= 7.01830E-07 tot_iter=   1623   conv_iter=    435
 tran: time= 8.02418E-07 tot_iter=   1900   conv_iter=    514
 tran: time= 9.01178E-07 tot_iter=   2161   conv_iter=    585
 tran: time= 1.00000E-06 tot_iter=   2410   conv_iter=    650
 sweep: tran tran5    end,  cpu clock=  7.03E+00 memory=    155 kb
 sweep: parameter       parameter      1 end
>info:         ***** hspice job concluded
 lic: Release hspice token(s)
```

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 *inv1out* vs.Time

Figure F-5: Plot of Voltage on Node *inv2out* vs. Time

Figure F-6: Plot of Current through Diode *dlow* vs. Time

Figure F-7: Plot of User-defined Measured Variable *falltime* vs. Amplifier Input Voltage

Full Simulation Example with Cosmos-Scope

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

```
*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)
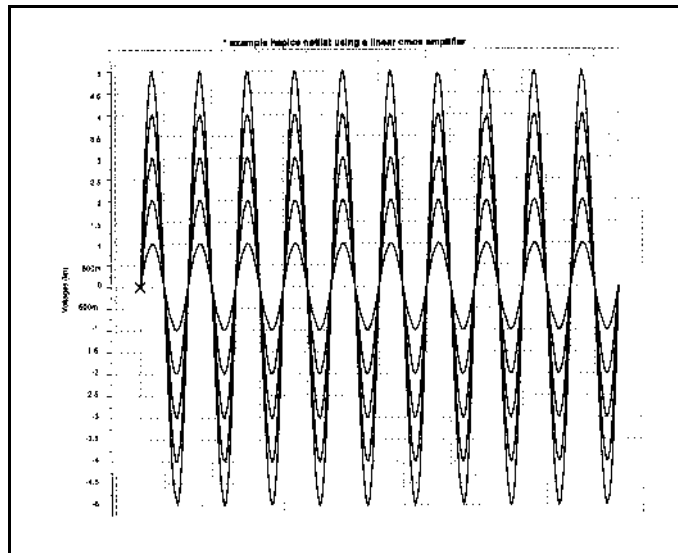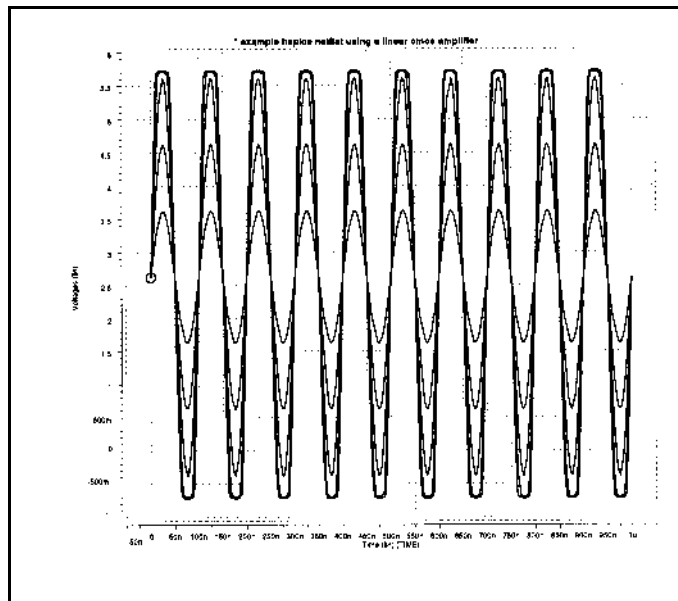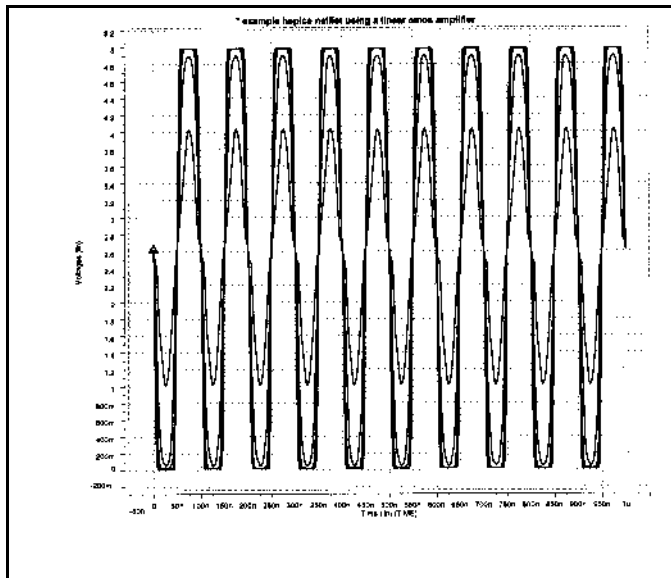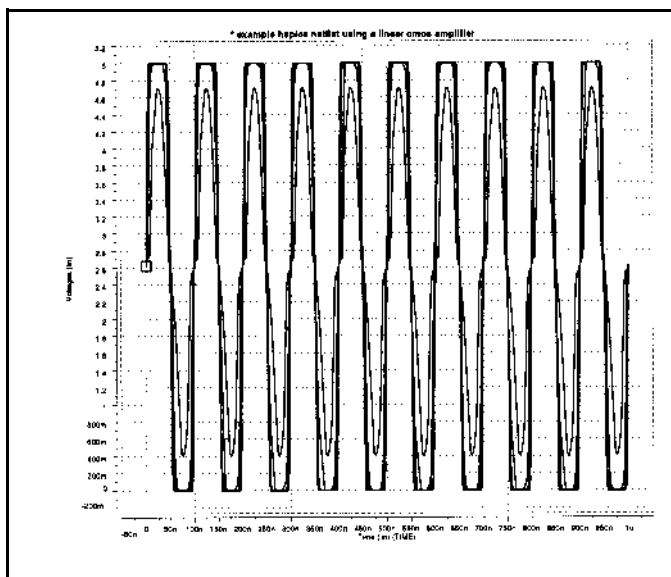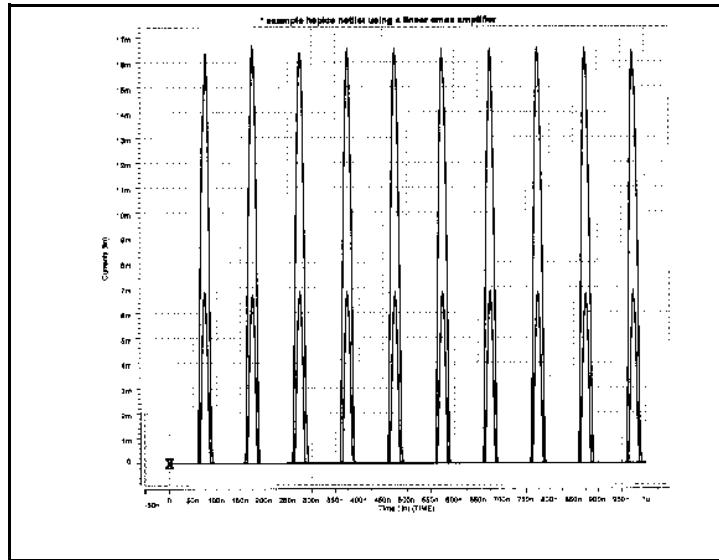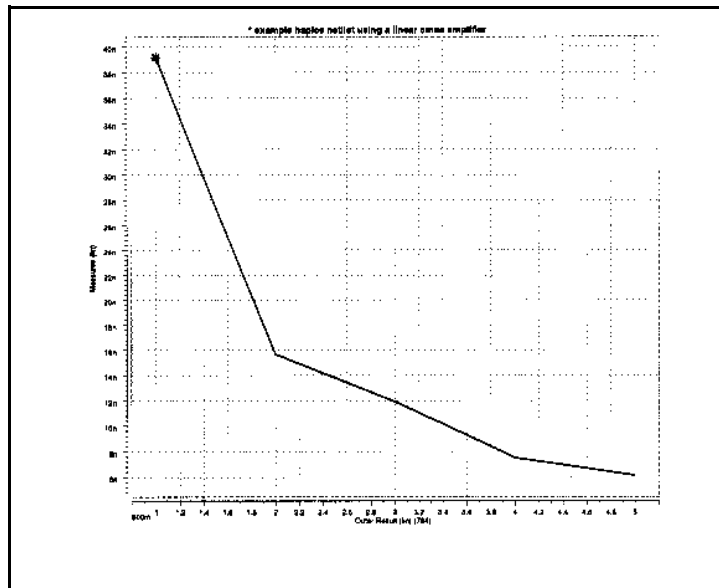
* analysis specifications
.TF v(5) vin
.DC vin -0.20 0.20 0.01 sweep monte=3
.AC dec 10 100k 10meghz
.NOISE v(4) vin 20
.NET v(5) vin rout=10k
.PZ v(5) vin
.DISTRO rc1 20 .9 1m 1.0
.SENS v(4)
.TRAN 5ns 200ns
.FOUR 5meg v(5) v(15)
.TEMP -55 150

* output specifications
.MEAS qa_propdly trig v(1) val=0.09 rise=1
```

```
+                 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
```

```
* model definitions
.MODEL qnl npn(bf=80 rb=100 ccs=2pf tf=0.3ns tr=6ns cje=3pf
+ cjc=2pf va=50 rc=10 trb=.005 trc=.005)
.MODEL qpl pnp(bf=80 rb=100 ccs=2pf tf=0.3ns tr=6ns cje=3pf
+ cjc=2pf va=50 bulk=0 rc=10)
*
.END
```

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.

Figure F-8: Transient Analysis Result: Plot of v(4), v(5), and ITPOWERD(power)

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
ABS element parameter 26-12, 26-18, 26-23, 26-28
abs(x) function 7-9
ABSH option 9-18, 9-36, 10-36, 11-13, 12-7
ABSI option 9-18, 10-35, 10-37
    KCLTEST setting 9-19, 10-26
ABSMOS option 9-18, 10-35, 10-37
    KCLTEST setting 9-19, 10-26
absolute
    power function 7-9
    value function 7-9
    value parameter 26-12, 26-18, 26-23, 26-28
ABSTOL option 9-19, 10-22
ABSV option 9-36, 10-35, 11-13
ABSVAR option 9-41, 9-44, 11-13, 11-16, 11-25
ABSVDC option 9-19, 10-38
AC
    analysis 8-3

    circuit and pole/zero models 29-44
    distortion 12-9
    MOSFETs 20-24
    noise 12-11
    RC network 2-2
  control options 12-7
  magnitude calculation 9-5, 9-37, 12-7
  network analysis 12-14
  optimization 12-4
  output
    calculation method 9-5, 9-37, 12-7
    variables analysis 8-27
  phase calculation 9-5, 9-37, 12-7
  resistance 12-3
    parameter 26-74
    transmission lines C-19
  small signal analysis 9-36
    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 8-14, 9-5
accuracy
  control options 10-36
  simulation time 10-35
  tolerance 10-34
ACCURATE option 9-36, 11-14, 11-25
ACM 17-3
  model parameter 11-25, 17-3, 20-27

G

J

L

M

N

O

S