# Avant!

# Star-Hspice Manual

**Release 1998.2**
**July 1998**

*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.

## Related Documents

The following documents pertain to this guide:

- Star-Hpice,Star-Time, and AvanWaves Installation Guide
- 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.<br><br>In examples, this font indicates system prompts, text from files, and messages printed by the system. |
| `courier italic` | Arguments appear in this font when the value of an argument is a string. The string must be enclosed with quotation marks. |

| Convention | Description |
|---|---|
| *times italic* | Indicates commands, functions, arguments, file names, and variables within a line of text.<br>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:<br>*pin1* [*pin2, ...pinN*]<br>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. |
| .<br>.<br>. | 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. |
| \| | 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://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*

*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 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
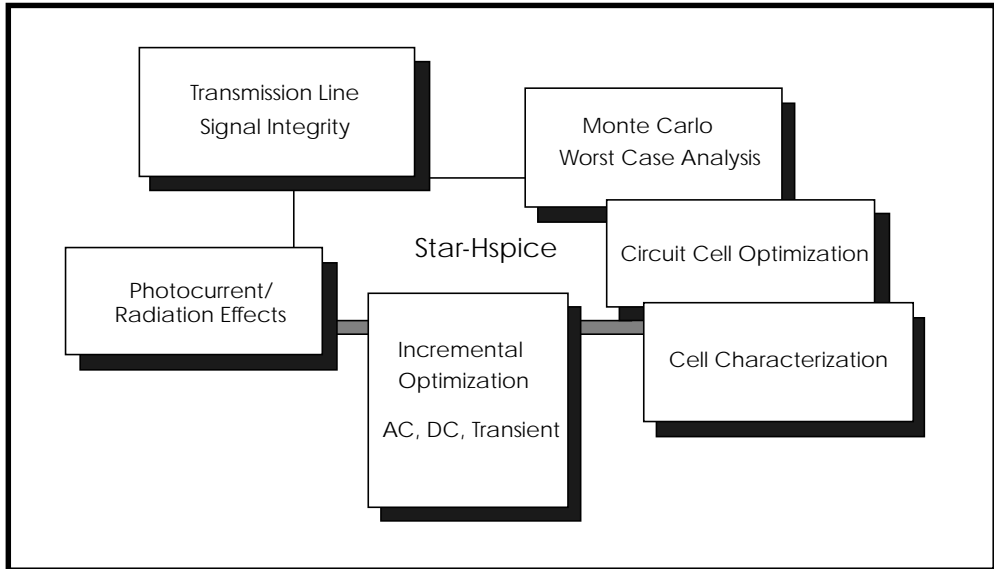


**Figure 1-1: Star-Hspice Design Features**

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
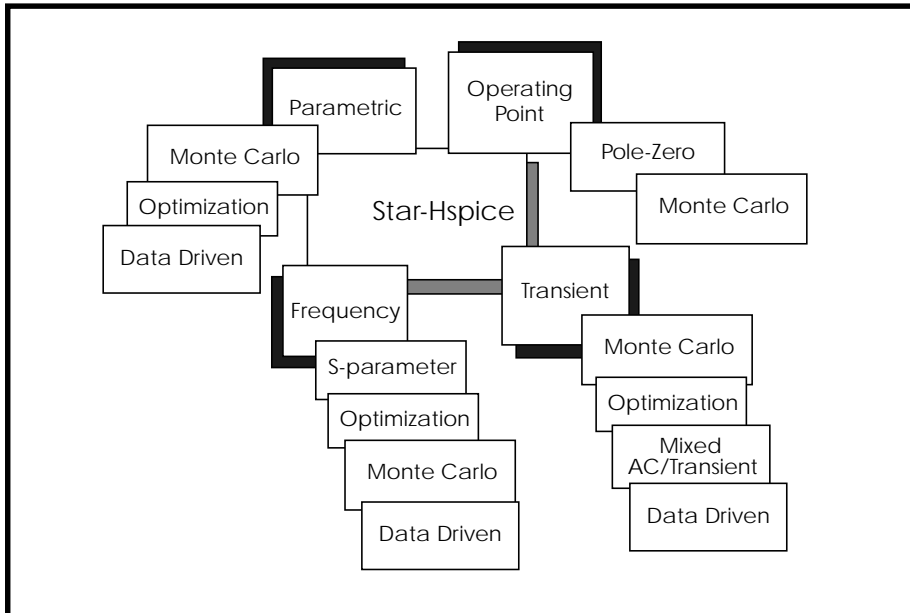


**Figure 1-2: Star-Hspice Circuit Analysis Types**
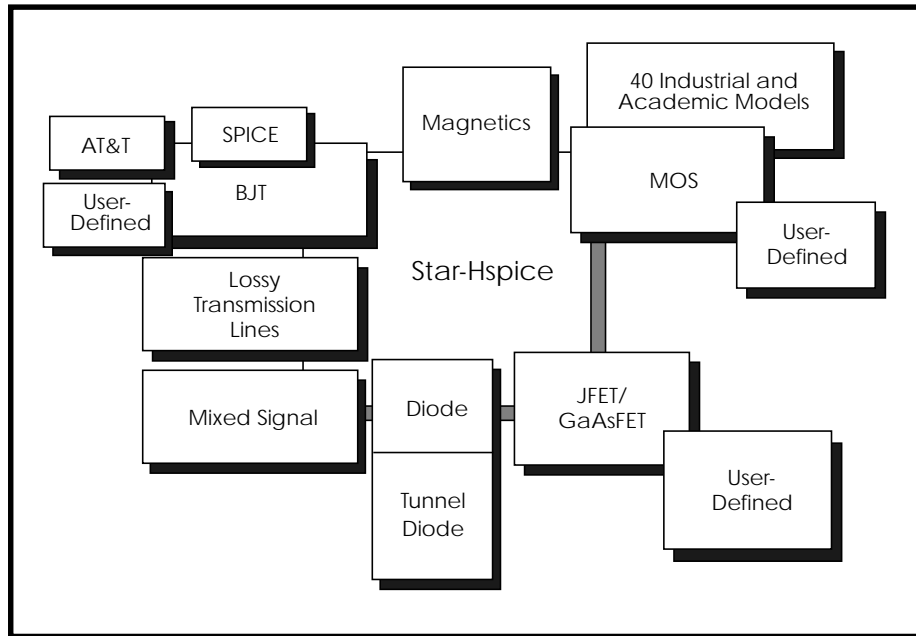
**Figure 1-3: Star-Hspice Modeling Technologies**

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

_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.
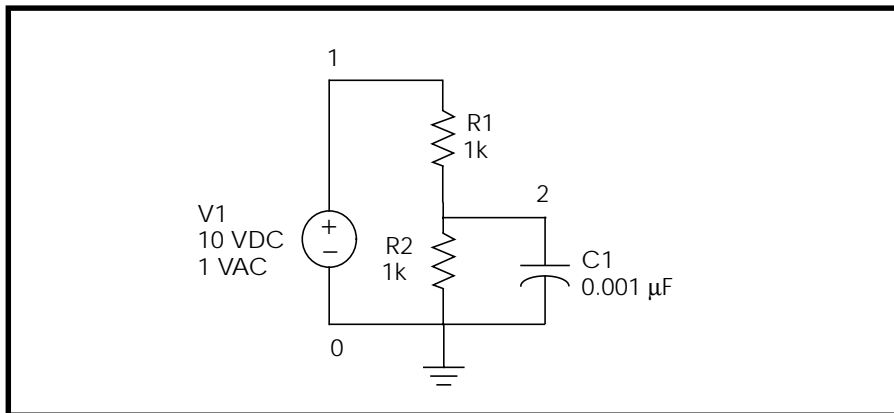


**Figure 2-1: – RC Network Circuit**

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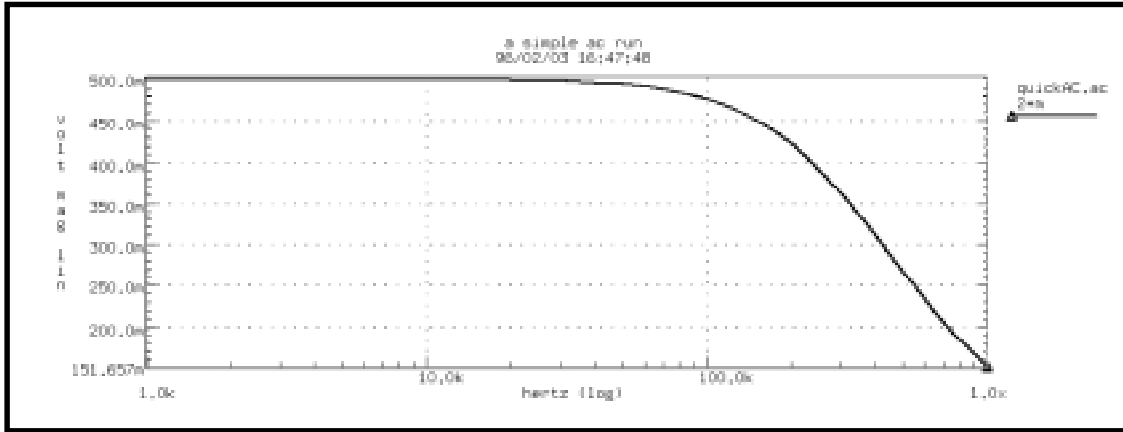
**Figure 2-2: RC Network Node 2 Frequency Response**

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 pulsbe sources and other types of sources is described in Chapter 5, "Using Sources and Stimuli".

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:ob.
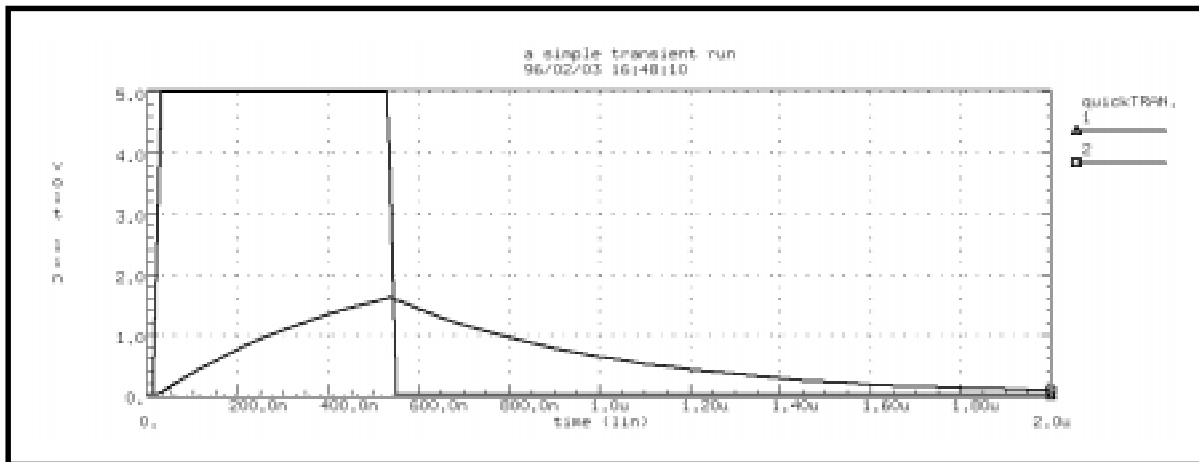
**Figure 2-3: Voltages at RC Network Circuit Node 1 and Node 2**

# Transient Analysis of an Inverter

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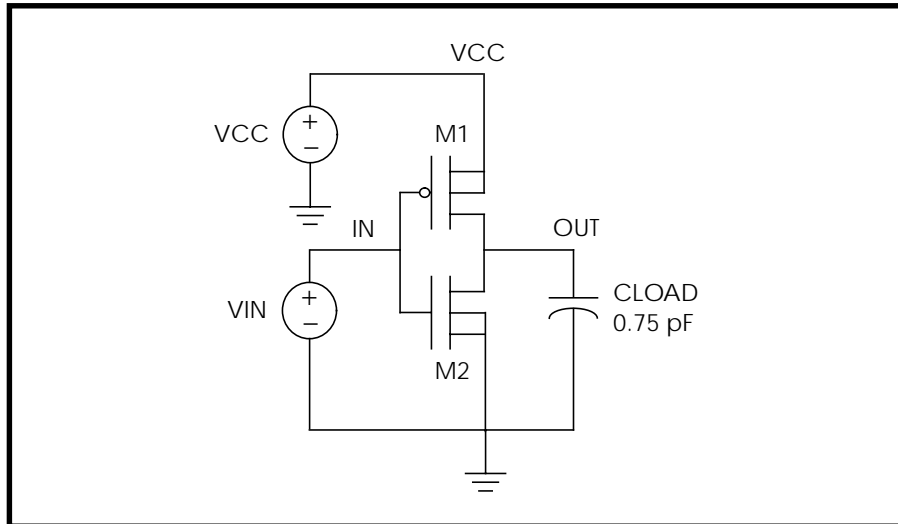


**Figure 2-4: – MOS Inverter Circuit**

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:.
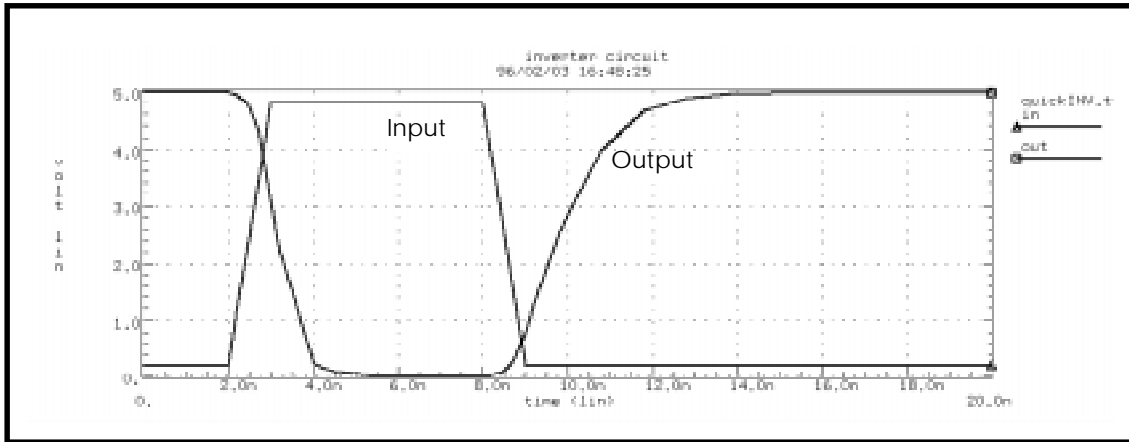


**Figure 2-5: – Voltage at MOS Inverter Node 1 and Node 2**

*Avant!*

# Chapter 3

# Specifying Simulation Input and Controls

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

- Examining the Simulation Structure
- Understanding the Data Flow
- Using the Star-Hspice Command
- Using Standard Input Files
- Using Input Control Statements
- Setting Control Options
- Understanding the Library Types
- Understanding the Library Input
- Comparing the Control Options Default Values

# Examining the Simulation Structure

Figure 2-1: shows the program structure for simulation experiments.
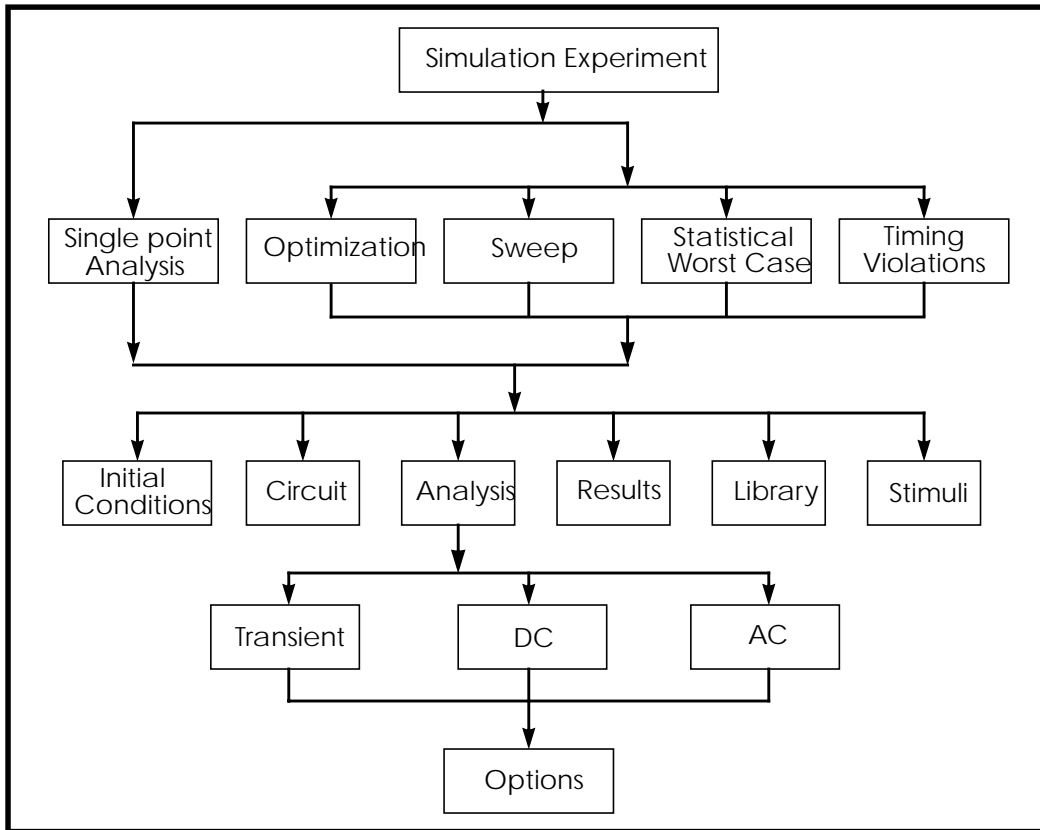


**Figure 3-1: Simulation Program Structure**

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 3-2:.

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 commands and options in various AvanWaves windows. Refer to the *AvanWaves User Guide* for instructions on using AvanWaves.
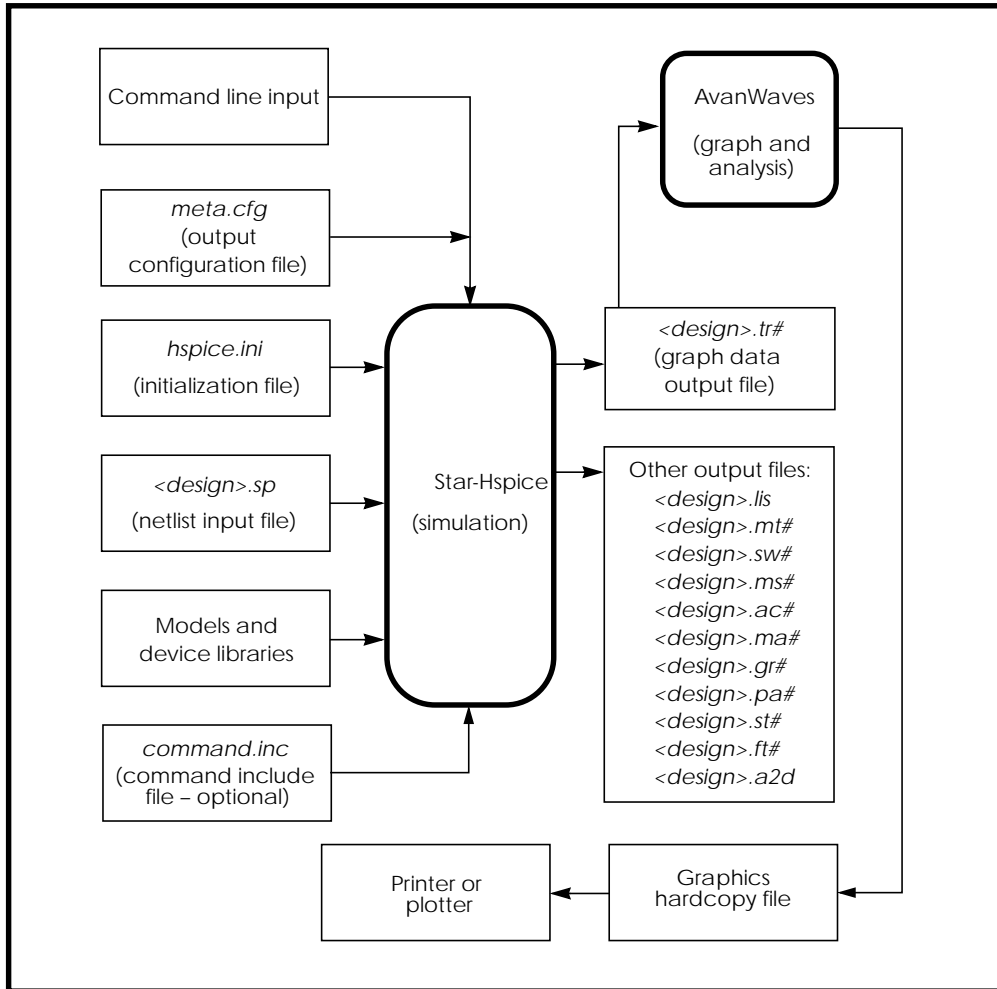
**Figure 3-2: Overview of Star-Hspice Data Flow**

## Simulation Process Overview

Figure 3-3 is a diagram of the Star-Hspice simulation process. The following section summarizes the steps in a typical simulation.

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

**Figure 3-3: Star-Hspice Simulation Process**

Perform these steps to execute a Star-Hspice simulation. These steps and the associated files are described in more detail in this chapter.

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 screen 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 -50°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.

# 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 *<arch>*                 is an argument that overrides the default architecture

Available Star-Hspice command arguments are listed in Table 2-1.

### Table 3-1: Star-Hspice Command Options

| Option | Description |
|---|---|
| -a <arch> | Platform architecture. Choices are:<br>❍ sun4, sol4 (SparcStation, Ultra)    ❍ sgi (SGI)<br>❍ pa (HP 700/800/9000)    ❍ cray (Cray)<br>❍ alpha (DEC ALPHA)    ❍ i386 (Windows 95/NT)<br>❍ 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 screen 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.

# 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 2-2:. These files are described in the following sections.

### Table 3-2: 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.

# Input Netlist File (*<design>.sp*)

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*, 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, 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" below.
- Input filename length, statement length, and equation length are limited to 256 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.

■ 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

An Star-Hspice input netlist file consists of one main program and one or more optional submodules. Use a submodule (preceded by a .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.

Table 2-3: lists the basic statements and elements of an input netlist file.

**Table 3-3: Input Netlist File Statements and Elements**

| Statement or Element | Definition |
|---|---|
| Title | The first line is the input netlist file title. |
| * or $ | Designates comments to describe the circuit |
| .OPTIONS | Sets conditions for simulation |
| Analysis statements and .TEMP | Statements to set sweep variables |
| .PRINT/.PLOT/.GRAPH/.PROBE | Statements to set print, plot and graph variables |
| .IC or .NODESET | Sets initial state; can also be put in subcircuits |
| Sources (I or V) and digital to analog inputs | Sets input stimuli |
| Netlist | Circuit |
| .LIB | Library |
| .INCLUDE | General include files |
| .PROTECT | Turns off output printback |
| .MODEL libraries | Element model descriptions |
| .UNPROTECT | Restores output printback |
| .DELETE LIB | Removes previous library selection |
| .ALTER | Sequence for in-line case analysis |
| .END | Required statement to terminate the simulation |

TITLE
.INCLUDE
.LIB MACROA
.INCLUDE
.ALTER

/////

.ALTER

/////

.END

Main Program 1
//////
,END

Main Program 2
//////
,END

Main Program 3
//////
,END

Concatenating data files for consecutive
simulation of different circuits
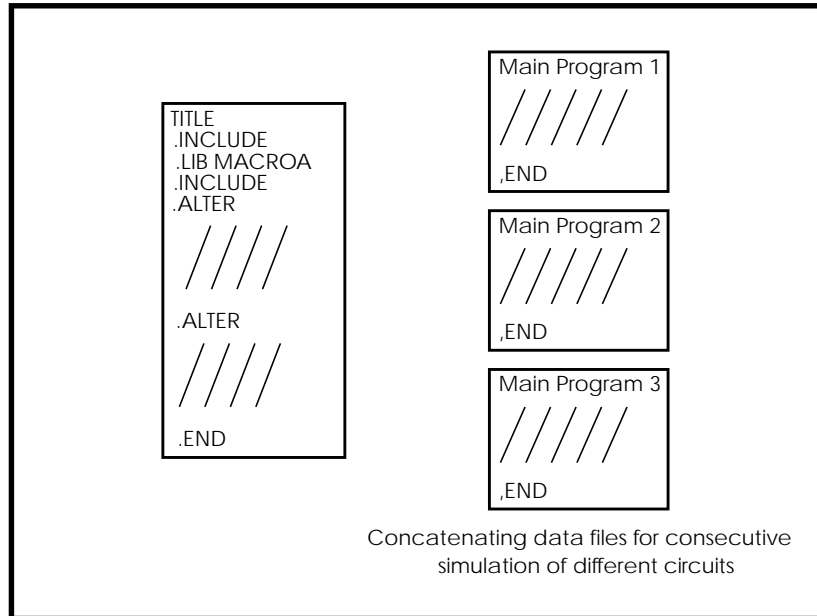
**Figure 3-4: Input Netlist File Structures**

## .TITLE Statement

The .TITLE statement resets the title printed on each subsequent print, plot, probe, or graph statement.

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.

The title is printed verbatim in each section heading of the output listing file of the simulation.

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>
```

### .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 the 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
```

## .GLOBAL Statement

The .GLOBAL statement is 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, override local subcircuit definitions.

## Element 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>
+ <val1 val2 ... valn>
```

where:

*elname*                Element name that cannot exceed 15 characters, and must begin with a specific letter for each element type:

| | |
|---|---|
| C | Capacitor |
| D | Diode |
| E,F,G,H | Dependent current and voltage controlled 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 15 additional alphanumeric characters. Only the first 16 characters of node names are significant. All characters after that are ignored. The following characters are not allowed in node names: = ( ) , . ' [ ] |
|---|---|
| *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. |
| *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. |

## Examples

```
Q1234567  4000 5000 6000 SUBSTRATE BJTMODEL AREA=1.0
```

The example above 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 example above 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 example above 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.

## Comments

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

### Syntax

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

or

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

### Examples

```
*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.

## 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

## 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 or slash (/) 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 Statements" on page 2-24).

## 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 2-5:.
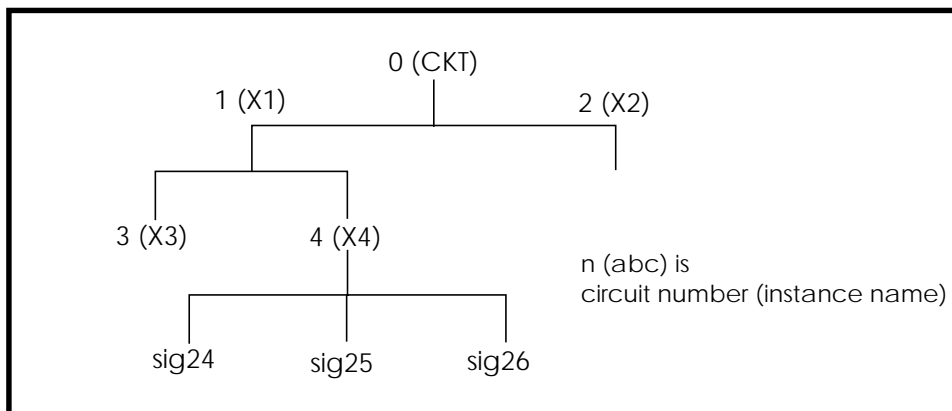
**Figure 3-5: Subcircuit Calling Tree with Circuit Numbers and Instance Names**

The path name of the node named *sig25* in subcircuit X4 in Figure 2-5: 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.

# Using Input Control Statements

This section describes how to use input control statements.

## .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 I/O 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

### 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.

### Example 1

```
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).

### Example 2

```
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 Example 2, 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.

## .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 the chapters on DC sweep, transient, and AC sweep analysis for details about using the .DATA statement in the different types of analysis.

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* | the data name referred to in the .TRAN, .DC or .AC statement |
| *pnami* | 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* | 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.

### 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.

### 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.

## Concatenated .DATA File Statement

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.*

**Syntax**

```
.DATA dataname MER
FILE='file1' p1=1 p2=3 p3=4 p4=4
<FILE='file2'> p1=1
<FILE='file3'>
<FILE='fileout'>
<OUT='fileout'>
.ENDDATA
```

In the above listing, *file1*, *file2*, *file3*, and *fileout* are concatenated into the output file, *fileout*. The data in *file1* is at the top of the file, followed by the data in *file2*, *file3* and *fileout*. The dataname field 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. 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 both *file1* and *fileout*.

## Column Laminated .DATA Statement

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.

**Syntax**
```
.DATA dataname LAM
FILE='file1' p1=1 p2=2 p3=3
<FILE='file2'> p4=1 p5=2
<FILE='file3'> p6=1
<OUT='fileout'>
.ENDDATA
```

This listing takes columns from *file1*, *file2*, and *file3*, and laminates them into the output file, *fileout*. Columns one, two, and three of *file1*, columns one and two of *file2*, and column one of *file3* are designated as the columns to be placed in the output file. There is a limit of 10 files per .DATA statement.

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

## .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 Chapter 9, "Parametric Variation and Statistical Analysis."

**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 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).

# 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 2-4:. Descriptions of the general control options follow the table. Options that are relevant to a specific simulation type are 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 ...*            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 2-4: lists the keywords for the .OPTIONS statement, grouped by their typical usage.

The general control options are described following the table. For descriptions of the options listed under each type of analysis, see the chapter on that type of analysis.

### Table 3-4:  .OPTIONS Keyword Application Table

| GENERAL CONTROL OPTIONS | | MODEL ANALYSIS | DC OPERATING POINT and DC SWEEP ANALYSIS | | 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 |
| BRIEF | CSDF | TNOM | ABSMOS | DCFOR | ACCURATE | DVDT |
| CO | MEASOUT | | ABSTOL | DCHOLD | ACOUT | FS |
| INGOLD | POST | *MOSFETs* | ABSVDC | DCON | CHGTOL | FT |
| LENNAM | PROBE | CVTOL | DI | DCSTEP | CSHUNT, | IMIN, ITL3 |
| LIST | PSF | DEFAD | KCLTEST | DV | GSHUNT | IMAX, ITL4 |
| MEASDGT | SDA | DEFAS | MAXAMP | GMAX | DI | ITL5 |
| NODE | ZUKEN | DEFL | RELH | GMINDC | GMIN | RELVAR |
| NOELCK | | DEFNRD | RELI | GRAMP | GSHUNT, | RMAX |
| NOMOD | *Analysis* | DEFNRS | RELMOS | GSHUNT | CSHUNT | RMIN |
| NOPAGE | ASPEC | DEFPD | RELV | ICSWEEP | MAXAMP | SLOPETOL |
| NOTOP | LIMPTS | DEFPS | RELVDC | NEWTOL | RELH | TIMERES |
| NUMDGT | PARHIER | DEFW | | OFF | RELQ | |
| NXX | SPICE | SCALM | *Matrix* | RESMIN | RELTOL | *Algorithm* |
| OPTLST | | WL | ITL1 | RMAXDC | TRTOL | DVTR |
| OPTS | *Error* | | ITL2 | | VNTOL, ABSV | IMAX |
| PATHNUM | BADCHR | *Inductors* | NOPIV | *Pole/Zero* | | IMIN |

## Table 3-4:  .OPTIONS Keyword Application Table

| GENERAL CONTROL OPTIONS | | MODEL ANALYSIS | DC OPERATING POINT and DC SWEEP ANALYSIS | | TRANSIENT and AC SMALL SIGNAL ANALYSIS | |
|---|---|---|---|---|---|---|
| PLIM | DIAGNOSTIC | GENK | PIVOT, SPARSE | CSCAL | *Speed* | LVLTIM |
| SEARCH | NOWARN | KLIM | | FMAX | AUTOSTOP | MAXORD |
| VERIFY | WARNLIMIT | | PIVREF | FSCAL | BKPSIZ | METHOD |
| | | *BJTs* | PIVREL | GSCAL | BYPASS | MU, XMU |
| *CPU* | *Version* | EXPLI | PIVTOL | LSCAL | BYTOL | |
| CPTIME | H9007 | | SPARSE, PIVOT | PZABS | FAST | *Input, Output* |
| EPSMIN | H95 | *Diodes* | | PZTOL | ITLPZ | INTERP |
| EXPMAX | | DIORSCAL | | RITOL | MBYPASS | ITRPRT |
| LIMTIM | | EXPLI | *Input, Output* | X*n*R, X*n*I | | UNWRAP |
| | | | CAPTAB | NEWTOL | | |
| | | | DCCAP | | | |
| | | | VFLOOR | | | |

A list of default values for options is provided in "Control Options Default Values Comparison" at the end of this chapter.

## Descriptions of General Control .OPTIONS Keywords

Descriptions of the general control options follow. The descriptions are alphabetical by keyword. See the chapters on transient analysis, DC analysis, AC analysis, and models, for descriptions of the options listed under those headings in Table 2-4:.

*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 (default)

        2     enables reporting of MATRIX statistics

*ARTIST=x*            ARTIST=2 enables the Cadence Analog Artist interface. This option requires permit authorization.

| | |
|---|---|
| *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. Default=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 |

| | |
|---|---|
| *BADCHR* | generates a warning when a nonprintable character is found in an input file |
| *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=1 enables printback. NXX is the same as BRIEF. |
| *CDS, SDA* | CDS=2 produces a Cadence WSF ASCII format post-analysis file for Opus™. This option requires permit authorization. SDA is the same as CDS. |
| *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. Default=80. |

| | |
|---|---|
| *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. Default=1e7 (400 days). |
| *CSDF* | selects Common Simulation Data Format (Viewlogic-compatible graph data file format) |
| *DIAGNOSTIC* | logs the occurrence of negative model conductances |
| *EPSMIN=x* | specifies the smallest number that can be added or subtracted on a computer, a constant value. Default=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. |
| 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. |
| *INGOLD=x* | specifies the printout data format. Use INGOLD=2 for SPICE compatibility. Default=0. |

*INGOLD=x* (continued)

Numeric output from Star-Hspice is 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. Default=8.

*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. Default=2001.

*LIMTIM=x*      sets the amount of CPU time reserved for generating prints
                and plots in case a CPU time limit (CPTIME=x) causes
                termination. Default=2 (seconds). This default is normally
                sufficient time for short printouts and plots.

*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. Default=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.

*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. Default=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 permit authorization.

*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 |
| *NOWARN* | suppresses all warning messages except those generated from statements in .ALTER blocks |
| *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. Default=4.0. |
|  | This option does not affect the accuracy of the simulation. |
| *NXX* | same as BRIEF. See 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 above and the 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. |
|---|---|

| | | |
|---|---|---|
| *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 |
| *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=x* | | enables storing of simulation results for analysis using the AvanWaves graphical interface or other methods. POST=2 saves the results in ASCII format. POST=1 saves the results in binary. Set the POST option, and use the .PROBE statement to specify which data you want saved. Default=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*          same as CDS. See CDS.

*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.

*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
```

| | |
|---|---|
| *VERIFY* | same as LIST. See LIST. |
| *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

Default=1.

| | |
|---|---|
| *ZUKEN=x* | If x is 2, enables the Zuken interactive interface. If x is 1, disables it. |

Default=1.

# Understanding the Library Types

This section describes the libraries Star-Hspice uses.

## Discrete Device Library

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 Statements" on page 2-24 and the chapters on specific types of devices for descriptions of element statements.

Optional parameter fields in the element statement can override the internal specification of the model. For example, for op-amp devices, the designer can override offset voltage and gain and offset current. Since the DDL library

devices are based on Star-Hspice circuit level models, the effects of supply voltage, loading, and temperature are automatically compensated for in a simulation.

Star-Hspice accesses DDL models in several ways on most computers:

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

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

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

    ```
    .INCLUDE <path>/T2N2211.inc
    ```

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

## Vendor Libraries

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

## Subcircuit Library Structure

Your library structure must adhere to the Star-Hspice implicit subcircuit
.INCLUDE statement specification feature. This Star-Hspice function allows
you to specify the directory that the subcircuit file (*<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 I/O 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.

# Understanding the Library Input

This section covers the types of library input required by Star-Hspice.

## 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 I/O 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.

## .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 3-6:). The DDL subdirectories are contained in the *parts/* directory.

x1 in out vdd vss buffer_f ⟶ .OPTIONS search='$installdir/parts/vendor'

$installdir/parts/vendor/buffer_f.inc

$installdir/parts/vendor/skew.dat

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

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

$installdir/parts/vendor/buffer.inc

$installdir/parts/vendor/model.dat

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

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

**Note:** The '/usr' directory is in the
Star-Hspice              install

**Figure 3-6: Vendor Library Usage**

## .INCLUDE Statement

**Syntax**

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

where:                specifies:

*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 256 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.

# .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
```

or

```
.LIB libnumber entryname
```

where:              specifies:

*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.

*libnumber*           library numbers, which are a means to reference a library file. These numbers are required where machine independence is needed and libraries reside in various directory structures on different machines. The file unit number permits a library's full path name to be assigned outside the Star-Hspice code. Values from 11-89 can be used for all systems except UNIX, which is restricted to values from 09-14. This number corresponds to the FORTRAN logical unit number to which the library number is attached. It is assigned to the library file when the Star-Hspice job is submitted.

## Examples

```
.LIB 4 lm101
.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's.*

## .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  hardcopy plot model for the .GRAPH statement
PMOS p-channel MOSFET model

|  |  |
|---|---|
| PNP | pnp BJT model |
| R | resistor model |

*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) and BSIM2 (Level=39) models between Star-Hspice releases. Star-Hspice release numbers and the corresponding version numbers are:

| *Star-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.

### Example

```
.MODEL MOD1 NPN BF=50 IS=1E-13 VBF=50 AREA=2 PJ=3, N=1.05
```

# .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 inhibits 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 syntax is:
```
.UNPROTECT
```

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.

# Comparing the Control Options Default Values

This section lists the default values for the Star-Hspice control options.

## General Options Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|---|---|---|---|---|---|---|
| | 9007d | H92B.01 | 93A.02 | 95.3 | 96.1 (dvdt=3) | 96.1 (dvdt=4) |
| acct | 0 | 0 | 1 | 1 | 1 | 1 |
| acout | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| aspec | 0 | 0 | 0 | 0 | 0 | 0 |
| autost | 0 | 0 | 0 | 0 | 0 | 0 |
| badchr | 0 | 0 | 0 | 0 | 0 | 0 |
| bkpsiz | - | - | - | 5000 | 5000 | 5000 |
| bytol | - | - | - | - | 50.00u | 100.00u |
| captab | - | - | 0 | 0 | 0 | 0 |
| co | 78 | 78 | 78 | 78 | 78 | 78 |
| cptime | 10.00x | 10.00x | 10.00x | 10.00x | 10.00x | 10.00x |
| diagno | - | - | 0 | 0 | 0 | 0 |
| epsmin | 1.0e-28 | 1.0e-28 | 1.0e-28 | 1.0e-28 | 1.0e-28 | 1.0e-28 |
| expli | - | - | - | - | 0 | 0 |
| expmax | 80.00 | 80.00 | 80.00 | 80.00 | 80.00 | 80.00 |
| genk | 1 | 1 | 1 | 1 | 1 | 1 |
| h9007 | - | 0 | 0 | 0 | 0 | 0 |
| icsweep | - | - | | 1 | 1 | 1 |
| ingold | 0 | 0 | 0 | 0 | 0 | 0 |

## General Options Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|--------|--------|---------|---------|------|------------------|------------------|
|        | 9007d  | H92B.01 | 93A.02  | 95.3 | 96.1 (dvdt=3)    | 96.1 (dvdt=4)    |
| itrprt | 0      | 0       | 0       | 0    | 0                | 0                |
| klim   | 10.00m | 10.00m  | 10.00m  | 10.00m | 10.00m         | 10.00m           |
| lennam | 8      | 8       | 8       | 8    | 8                | 8                |
| limpts | -      | -       | 2001    | 2001 | 2001             | 2001             |
| limtim | 2      | 2       | 2       | 2    | 2                | 2                |
| list   | 0      | 0       | 0       | 0    | 0                | 0                |
| measdgt | 0     | 0       | 4       | 4    | 4                | 4                |
| measout | 1     | 1       | 1       | 1    | 1                | 1                |
| node   | 0      | 0       | 0       | 0    | 0                | 0                |
| noelck | 0      | 0       | 0       | 0    | 0                | 0                |
| nomod  | 0      | 0       | 0       | 0    | 0                | 0                |
| nopage | 0      | 0       | 0       | 0    | 0                | 0                |
| notop  | 0      | 0       | 0       | 0    | 0                | 0                |
| nowarn | 0      | 0       | 0       | 0    | 0                | 0                |
| numdgt | 4      | 4       | 4       | 4    | 4                | 4                |
| nxx    | 0      | 0       | 0       | 0    | 0                | 0                |
| optlst | -      | -       | 0       | 0    | 0                | 0                |
| opts   | 0      | 0       | 0       | 0    | 0                | 0                |
| parhier | -     | -       | -       | global | global         | global           |
| pathnum | -     | -       | 0       | 0    | 0                | 0                |
| plim   | 0      | 0       | 0       | 0    | 0                | 0                |

## General Options Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|---|---|---|---|---|---|---|
| | 9007d | H92B.01 | 93A.02 | 95.3 | 96.1 (dvdt=3) | 96.1 (dvdt=4) |
| post | 0 | 0 | 0 | 0 | 0 | 0 |
| resmin | - | - | - | 10.00u$^\dagger$ | 10.00u$^\dagger$ | 10.00u$^\dagger$ |
| scale | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| scalm | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| slopetol | - | - | - | 500.00m$^{\dagger\dagger}$ | 500.00m$^{\dagger\dagger}$ | 750.00m$^{\dagger\dagger}$ |
| spice | 0 | 0 | 0 | 0 | 0 | 0 |
| timeres | - | - | | 1.00p | 1.00p | 1.00p |
| tnom | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 | 25.00 |
| unwrap | - | 0 | 0 | 0 | 0 | 0 |
| vfloor | - | - | 500.00n | 500.00n | 500.00n | 500.00n |
| warnlim | - | - | 1 | 1 | 1 | 1 |

$\dagger$  Set resmin to 0.01 ohm for backward compatibility with 93A or earlier.
$\dagger\dagger$ Set .OPTIONS slopetol=0 for backward compatibility with 93A or earlier; slopetol  defaults to 0.75 when dvdt = 4 and lvltim = 1. In all other cases slopetol defaults to 0.5.

## MOSFET Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|---|---|---|---|---|---|---|
| | 9007d | H92B.01 | 93A.02 | 95.3 | 96.1 (dvdt=3) | 96.1 (dvdt=4) |
| defad | 0 | 0 | 0 | 0 | 0 | 0 |
| defas | 0 | 0 | 0 | 0 | 0 | 0 |
| defl | 100.00u | 100.00u | 100.00u | 100.00u | 100.00u | 100.00u |

## MOSFET Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|--------|---------|----------|---------|------|------------------|------------------|
|        | 9007d   | H92B.01  | 93A.02  | 95.3 | 96.1 (dvdt=3)    | 96.1 (dvdt=4)    |
| defnrd | 0       | 0        | 0       | 0    | 0                | 0                |
| defnrs | 0       | 0        | 0       | 0    | 0                | 0                |
| defpd  | 0       | 0        | 0       | 0    | 0                | 0                |
| defps  | 0       | 0        | 0       | 0    | 0                | 0                |
| defw   | 100.00u | 100.00u  | 100.00u | 100.00u | 100.00u       | 100.00u          |
| k2lim  | -       | 0        | 0       | 0    | 0                | 0                |
| wl     | 0       | 0        | 0       | 0    | 0                | 0                |

## Diode Level 3 Options Defaults

| OPTION  | Star-Hspice RELEASE | | | | | |
|---------|-------|---------|--------|------|---------------|---------------|
|         | 9007d | H92B.01 | 93A.02 | 95.3 | 96.1 (dvdt=3) | 96.1 (dvdt=4) |
| diorscal | -    | -       | 0      | 0    | 0             |               |

## DC Solution Options Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|--------|-------|---------|--------|-------|---------------|---------------|
|        | 9007d | H92B.01 | 93A.02 | 95.3  | 96.1 (dvdt=3) | 96.1 (dvdt=4) |
| absh   | 0     | 0       | 0      | 0     | 0             | 0             |
| absi   | 1.00n | 1.00n   | 1.00n  | 1.00n | 1.00n         | 1.00n         |
| absmos | 1.00n | 1.00u   | 1.00u  | 1.00u | 1.00u         | 1.00u         |

## DC Solution Options Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|--------|-------|---------|--------|------|----------------|----------------|
|        | 9007d | H92B.01 | 93A.02 | 95.3 | 96.1 (dvdt=3) | 96.1 (dvdt=4) |
| absvdc | 50.00u | 50.00u | 50.00u | 50.00u | 50.00u | 50.00u |
| accf | 1.3 | - | - | - | - | - |
| converge | - | 0 | 0 | 0 | 0 | 0 |
| cshdc | - | 1.00p | 1.00p | 1.00p | 1.00p | 1.00p |
| cshunt | - | 0 | 0 | 0 | 0 | 0 |
| dccap | 0 | 0 | 0 | 0 | 0 | 0 |
| dcfor | 1 | 0 | 0 | 0 | 0 | 0 |
| dchold | - | 0 | 1 | 1 | 1 | 1 |
| dcon | 0 | 0 | 0 | 0 | 0 | 0 |
| dcstep | 0 | 0 | 0 | 0 | 0 | 0 |
| dctran | 0 | - | - | - | - | - |
| di | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| dv | 1.00k | 1.00k | 1.00k | 1.00k | 1.00k | 1.00k |
| gmax | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| gmin | 1.00p | 1.00p | 1.00p | 1.00p | 1.00p | 1.00p |
| gmindc | 1.00p | 1.00p | 1.00p | 1.00p | 1.00p | 1.00p |
| gramp | 0 | 0 | 0 | 0 | 0 | 0 |
| gshunt | - | 0 | 0 | 0 | 0 | 0 |
| itl1 | 200 | 200 | 200 | 200 | 200 | 200 |
| itl2 | 50 | 50 | 50 | 50 | 50 | 50 |
| kcltest | - | 0 | 0 | 0 | 0 | 0 |

## DC Solution Options Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|---|---|---|---|---|---|---|
| | 9007d | H92B.01 | 93A.02 | 95.3 | 96.1 (dvdt=3) | 96.1 (dvdt=4) |
| maxamp | 0 | 0 | 0 | 0 | 0 | 0 |
| newtol | 0 | 0 | 0 | 0 | 0 | 0 |
| nopiv | 0 | 0 | 0 | 0 | 0 | 0 |
| off | 0 | 0 | 0 | 0 | 0 | 0 |
| pivot | 10 | 10 | 10 | 10 | 10 | 10 |
| pivref | 100.00x | 100.00x | 100.00x | 100.00x | 100.00x | 100.00x |
| pivrel | 100.00n | 100.00u | 100.00u | 100.00u | 100.00u | 100.00u |
| pivtol | 1.00f | 1.00f | 1.00f | 1.00f | 1.00f | 1.00f |
| qvcrit. | 0 | - | - | - | - | - |
| qvt | 26.00m | - | - | - | - | - |
| relh | 50.00m | 50.00m | 50.00m | 50.00m | 50.00m | 50.00m |
| reli | 10.00m | 10.00m | 10.00m | 10.00m | 10.00m | 10.00m |
| relmos | 50.00m | 50.00m | 50.00m | 50.00m | 50.00m | 50.00m |
| relvdc | 1.00m | 1.00m | 1.00m | 1.00m | 1.00m | 1.00m |
| rmaxdc | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |

## Transient Analysis Options Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|---|---|---|---|---|---|---|
| | 9007d | H92B.01 | 93A.02 | 95.3 | 96.1 (dvdt=3) | 96.1 (dvdt=4) |
| absv | 50.00u | 50.00u | 50.00u | 50.00u | 50.00u | 50.00u |

## Transient Analysis Options Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|--------|---------|---------|---------|---------|-------------------|-------------------|
|        | 9007d   | H92B.01 | 93A.02  | 95.3    | 96.1 (dvdt=3)     | 96.1 (dvdt=4)     |
| absvar | 500.00m | 500.00m | 500.00m | 500.00m | 500.00m | 500.00m |
| accurate | -     | 0       | 0       | 0       | 0       | 0       |
| chgtol | 1.00n   | 1.00f   | 1.00f   | 1.00f   | 1.00f   | 1.00f   |
| bypass | 0       | 0       | 0       | 0       | 0       | $1.00^{\dagger}$ |
| cvtol  | 200.00m | 200.00m | 200.00m | 200.00m | 200.00m | 200.00m |
| dcap   | 2       | 2       | 2       | 2       | 2       | 2       |
| delmax | 0       | 0       | 0       | 0       | 0       | 0       |
| dvdt   | 0       | 3       | 3       | 3       | 3       | 4       |
| dvtr   | 1.00k   | 1.00k   | 1.00k   | 1.00k   | 1.00k   | 1.00k   |
| fast   | 0       | 0       | 0       | 0       | 0       | 0       |
| fs     | 250.00m | 250.00m | 250.00m | 250.00m | 250.00m | 250.00m |
| ft     | 250.00m | 250.00m | 250.00m | 250.00m | 250.00m | 250.00m |
| imax   | 8       | 8       | 8       | 8       | 8       | 8       |
| imin   | 3       | 3       | 3       | 3       | 3       | 3       |
| interp | 0       | 0       | 0       | 0       | 0       | 0       |
| itl5   | 0       | 0       | 0       | 0       | 0       | 0       |
| lvltim | 1       | 1       | 1       | 1       | 1       | 1       |
| mbypass | -      | -       | 1.00    | 1.00    | 1.00    | 2.00    |
| mu     | 500.00m | 500.00m | 500.00m | 500.00m | 500.00m | 500.00m |
| maxord | 2       | 2       | 2       | 2       | 2       | 2       |
| method | trap    | trap    | trap    | trap    | trap    | trap    |

## Transient Analysis Options Defaults

| OPTION | Star-Hspice RELEASE | | | | | |
|--------|--------|--------|--------|--------|--------|--------|
| | 9007d | H92B.01 | 93A.02 | 95.3 | 96.1 (dvdt=3) | 96.1 (dvdt=4) |
| relq | 50.00m | 10.00m | 10.00m | 10.00m | 10.00m | 10.00m |
| relv | 1.00m | 1.00m | 1.00m | 1.00m | 1.00m | 1.00m |
| relvar | 300.00m | 300.00m | 300.00m | 300.00m | 300.00m | 300.00m |
| rmax | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 5.00[††] |
| rmin | 1.00n | 1.00n | 1.00n | 1.00n | 1.00n | 1.00n |
| trtol | 7.00 | 7.00 | 7.00 | 7.00 | 7.00 | 7.00 |

†   bypass only defaults to 1 when dvdt = 4. In all other cases it defaults to 0.
††  rmax defaults to 5.00 when dvdt = 4 and lvltim = 1.

*Avant!*

# Chapter 4

# 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
- Selecting Simulation Parameters
- Displaying Simulation Results

# Using Output Statements

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. The .PRINT statement prints numeric analysis results. The .PLOT statement generates low resolution printer plots in the output listing file. The .GRAPH statement generates high resolution plots for supported devices such as HP LaserJet and PostScript printers without using AvanWaves. The .PROBE statement (together with .OPTION PROBE) allows output variables to be saved in all the interface files with no additional output in the listing file. The .MEASURE statement prints numeric results of measured electrical specifications for specific analyses. 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 terminal or printer display of waveforms.

## Output Files

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

### Table 4-1: 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#* |

### Table 4-1: Star-Hspice Output Files and Suffixes

| Output File Type | Extension |
|---|---|
| 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 listed in Table 4-1: are 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 (see *Chapter 1, Introducing Star-Hspice*). *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.

***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.

## 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 10-4 for information about parameters in Star-Hspice.

For element or node output variables defined as parameters, if the parameter name is longer than 16 characters, Star-Hspice substitutes a 0 for the variable, issues a warning, and continues the analysis. The value of the result is 0. For example, in the following statement the parameter name "xptgref.xbug.mxi18" is replaced by 0 because it is longer than 16 characters. This results in a value of 0 for the result "ace":

```
.MEASURE TRAN ace AVG
+ PAR('2*(il(xptgref.xbuf.mxi18))') from 0 to 100
```

Star-Hspice prints a warning that a value of zero is used for a nonexistent output variable, since it does not recognize the long name as a valid output variable name.

## .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.

## .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 above 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 in order 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 timestep 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.

# Selecting Simulation 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 analysis 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 you can use to specify the element templates for output are summarized in *Element Template Output, page -35 in , Specifying Simulation Output*.

### Nodal Voltage Output

#### 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 Output: Voltage Sources

#### Syntax

```
I (<Xzzz.Xyyy>.Vxxx)
```

| *Xyyy* | subcircuit element name, used to call the subcircuit (if any) within which the independent voltage source is defined |
| *Xzzz* | subcircuit element name, used to call the subcircuit (if any) that defines Xyyy |
| *Vxxx* | independent voltage source element name |

**Examples**

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

## Current Output: Element Branches

**Syntax**

```
In (<Xzzz.Xyyy>.Wwww)
```

| where: | specifies: |
| *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. |
| *Xzzz* | subcircuit element name, used to call the subcircuit (if any) within which Xyyy is defined |
| *Xyyy* | subcircuit element name, used to call the subcircuit (if any) within which Wwww is defined |
| *Wwww* | element name |

**Examples**

```
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 4-1 through 4-6 is defined in terms of arrow notation (current direction) and node position number (terminal type).
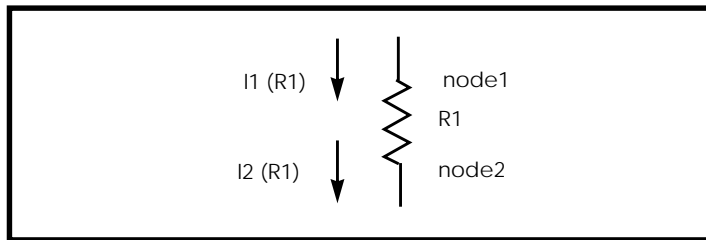


**Figure 4-1: Resistor (node1, node2)**
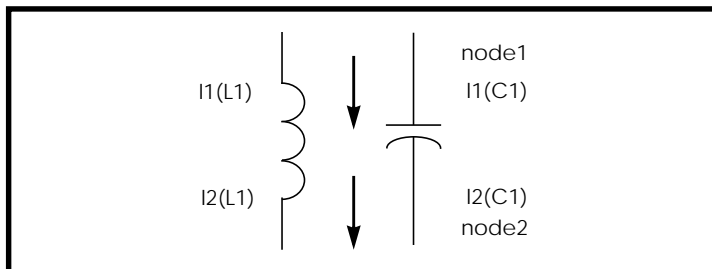


**Figure 4-2: Capacitor (node1, node2); Inductor (node 1, node2)**
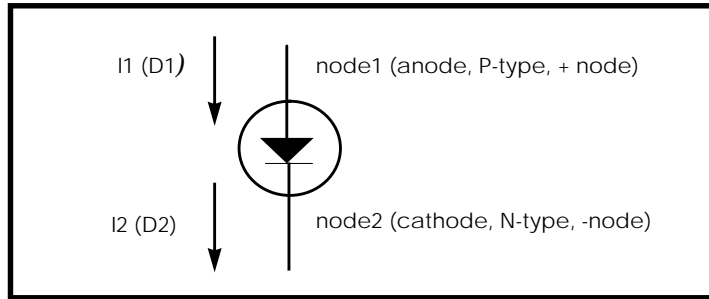
**Figure 4-3: Diode (node1, node2)**
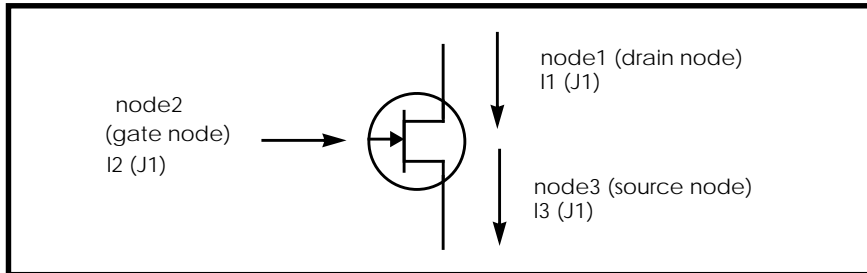


**Figure 4-4: JFET (node1, node2, node3) - n-channel**
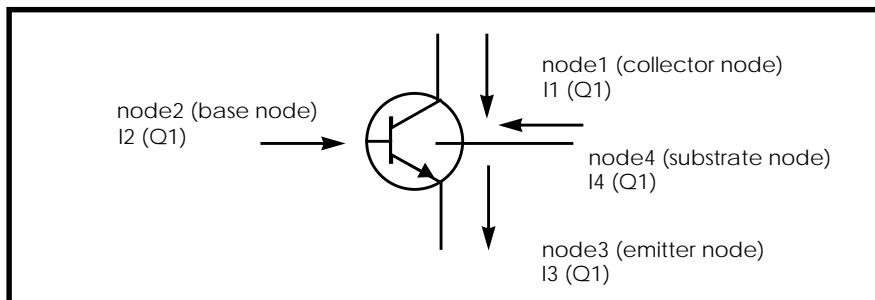


**Figure 4-5: BJT (node1, node2, node3, node4) - npn**
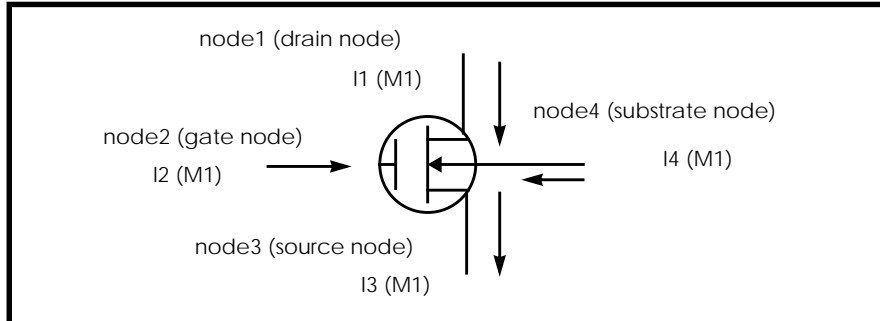
**Figure 4-6: MOSFET (node1, node2, node3, node4) - n-channel**

## AC Analysis Output

This section describes the output for AC analysis.

### 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. See "Group Time Delay Output" on page 9-12 for more information about printing and plotting group time delays.

Also see "AC Analysis Output" on page 4-14 for descriptions and examples of output variables for AC analysis.

### Noise and Distortion Analysis Output Variables

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. |

### Examples
```
.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.*

## 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.

**Examples**

```
.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 |

## .MEASURE Statement

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 the error mode and GOAL parameter, .MEASURE is also used extensively for optimization of circuit component values and curve fitting measured data to model parameters.

The .MEASURE statement has several different formats, depending on the application. You can use it for either DC, AC, or transient analysis.

Fundamental measurement modes are:

- Rise, fall, and delay
- Average, RMS, min, max, peak-to-peak, and integral
- Find-when
- Equation evaluation
- Derivative evaluation
- Integral 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.

## 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* | is the name given 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. |

### TRIG (Trigger) Syntax

```
TRIG trig_var VAL=trig_val <TD=time_delay> <CROSS=c> <RISE=r>
+    <FALL=f>
```

or

```
TRIG AT=val
```

### TARG (Target) Syntax

```
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* | is the 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* | the name of the output variable whose propagation delay is determined with respect to the *trig_var* |
| *time_delay* | the 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*<br>*RISE=r*<br>*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* | a 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. |

**Examples**

```
.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.*

## Average, RMS, MIN, MAX, and Peak-To-Peak Measurements

The average (AVE), RMS, MIN, MAX, and peak-to-peak (PP) measurement
modes report 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. Integral
provides the integral of an output variable over a specified period.

**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_var* divided by the periods of interest |
| | MAX | maximum: reports the maximum value of the *out_var* over the specified interval |
| | MIN | minimum: reports the minimum value of the *out_var* over the specified interval |
| | PP | peak-to-peak: reports the maximum value minus the minimum value of the *out_var* over the specified interval |
| | RMS | root mean squared: calculates the square root of the area under the $out\_var^2$ curve divided by the period of interest |
| | INTEG | integral: reports the integral of *out_var* over the specified interval |

| | |
|---|---|
| *result* | the name given the measured value in the Star-Hspice output. The value is a function of the variable specified (*out_var*) and func. |
| *out_var* | the 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. |

### Examples

```
.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
.MEAS   TRAN   charge INTEG I(cload) FROM=10ns   TO=100ns
```

## 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 *.MEASURE Statement, page -19 in , Specifying Simulation Output* for the definitions of parameters on measure statement.

The syntax is:

    .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*<br>*RISE=r*<br>*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*                     the name given the measured value in the Star-Hspice output

*TD*                         identifies the time at which measurement is to start

*WEIGHT*                     the 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>
```

## 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* | the value of *out_var* at which the derivative is to be found |
| *CROSS=c*<br>*RISE=r*<br>*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 $ERRfun = (GOAL - result)/GOA$. |
| *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* | the variable for which the derivative is to be found |
| *result* | the name given 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 |

## Examples

The following example calculates the derivative of V(out) at 25 ns:

```
.MEAS TRAN slewrate 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
```

## INTEGRAL Function

The INTEGRAL function provides the integral of an output variable over a specified period.

### Syntax

.MEASURE <DC|TRAN|AC> result INTEGRAL out_var1 <FROM=val1> <TO=val2>
+ <GOAL=goalval> <MINVAL=minval> <WEIGHT=weightval>

where:

| | |
|---|---|
| *<DC|AC|TRAN>* | specifies the analysis type of the measurement. If omitted, the last analysis mode requested is assumed. |
| *result* | the name given the measured value in the Star-Hspice output |
| *INTEGRAL* | selects the integral function. It can be abbreviated to INTEG. |
| *FROM* | the lower limit of integration |
| *TO* | the upper limit of integration |
| *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. |
| *WEIGHT* | the calculated error between result and GOAL is multiplied by the weight value. Default=1.0. |

### Example

The following example calculates the integral of I(cload) from t=10 ns to
t = 100 ns:

```
.MEAS TRAN charge INTEG I(cload) FROM=10ns TO=100ns
```

## 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* | the name given the measured result in the output |
| *ERRfun* | ERRfun indicates which error function to use: ERR, ERR1, ERR2, or ERR3. |
| *meas_var* | the name of any output variable or parameter in the data statement. M denotes the *meas_var* in the error equation. |
| *calc_var* | the 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. |

**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

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

**ERR3**

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

The + sign corresponds to a positive M/C ratio. The – sign corresponds to a negative M/C ratio.

*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.*

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.

## 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 in this section.

## Format of Element Template Output

The form is:

```
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 |

### Examples

```
.PLOT TRAN V(1,12) I(X2.VSIN) I2(Q3) DIO1:GD
.PRINT TRAN X2.M1:CGGBO M1:CGDBO X2.M1:CGSBO
```

## 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 |
| VOLT | LX1 | voltage across inductor |

## Inductor

| Name | Alias | Description |
|------|-------|-------------|
| 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 |
| CURR | LX1 | current through source |

## Voltage-Controlled Voltage Source

| Name | Alias | Description |
|------|-------|-------------|
| 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 |
| 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   | photocurrent in diode                                                       |

## BJT

| Name | Alias | Description |
|------|-------|-------------|
| AREA | LV1   | area factor |

## BJT

| Name | Alias | Description |
|------|-------|-------------|
| 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 | photocurrent, base-collector |
| PIBE | LV19 | photocurrent, base-emitter |
| VBE | LX0 | VBE |
| VBC | LX1 | base-collector voltage (VBC) |
| CCO | LX2 | collector current (CCO) |
| CBO | LX3 | base current (CBO) |
| GPI | LX4 | $g_\pi = ib / vbe$, constant vbc |
| GU | LX5 | $g_\mu = ib / vbc$, constant vbe |
| GM | LX6 | $g_m = ic / vbe + ic / vbe$, constant vce |

## BJT

| Name | Alias | Description |
| --- | --- | --- |
| G0 | LX7 | $g_0 =$ ic / vce 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$\Pi$) |
| CAP_IBC | LX20 | cbc internal base-collector capacitance (C$\mu$) |
| 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 | (TF*IBE) / vbc |
| 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 | photocurrent, gate-drain in JFET |
| PIGS | LV17 | photocurrent, 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) |
| 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) |
| 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 | $\Delta L$ (MOS6 amount of channel length modulation) (only valid for Levels 1, 2, 3 and 6) |

## MOSFET

| Name | Alias | Description |
| --- | --- | --- |
| 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) |
| 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) |

## MOSFET

| Name | Alias | Description |
|------|-------|-------------|
| QG | LX14 | gate charge (QG) |
| CQG | LX15 | gate charge current (CQG) |
| QD | LX16 | channel charge (QD) |
| CQD | LX17 | channel charge current (CQD) |
| CGGBO | LX18 | $CGGBO = \partial Qg/\partial Vg$ = CGS + CGD + CGB |
| CGDBO | LX19 | $CGDBO = \partial Qg/\partial Vd$, (for Meyer CGD=-CGDBO) |
| CGSBO | LX20 | $CGSBO = \partial Qg/\partial Vs$, (for Meyer CGS=-CGSBO) |
| CBGBO | LX21 | $CBGBO = \partial Qb/\partial Vg$, (for Meyer CGB=-CBGBO) |
| CBDBO | LX22 | $CBDBO = \partial Qb/\partial Vd$ |
| CBSBO | LX23 | $CBSBO = \partial Qb/\partial Vs$ |
| QBD | LX24 | drain-bulk charge (QBD) |
| – | LX25 | drain-bulk charge current (CQBD) (not used in Star-Hspice releases after 95.3) |
| 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 | $CDGBO = \partial Qd/\partial Vg$ |
| CDDBO | LX33 | $CDDBO = \partial Qd/\partial Vd$ |
| CDSBO | LX34 | $CDSBO = \partial Qd/\partial Vs$ |

## 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) |

# 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 print. These are voltage, current, or element template variables from a DC, AC, TRAN, NOISE, or DISTO analysis. |

### Examples

```
.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 tprinted 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 statements.

```
.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.*

## Print Control Options

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. 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
> 1.23e-01

## Subcircuit Output Printing

The following examples demonstrate how to print or plot voltages of nodes in subcircuit definitions using .PRINT or .PLOT.

*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 subcircuits X2 and X1. The full path name X1.X2.56 specifies that node 56 is within subcircuit X2 that is within subcircuit X1.

# .WIDTH Statement

The syntax is:
```
   .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.

## .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*            the 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. |

### Examples

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>
```

| | |
|---|---|
| *antype* | the 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

*Note:   The .GRAPH statement is not provided in the PC version of Star-Hspice.*

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.

### Syntax

```
.GRAPH antype <MODEL=mname> <unam1=> ov1,
+ <unam2=>ov2, … <unam32=> ov32 (plo,phi)
```

where

| | |
|---|---|
| *antype* | the type of analysis for the specified plots. Analysis types are: DC, AC, TRAN, NOISE, or DISTO. |
| *mname* | the plot modelname referenced by the .GRAPH statement. The .GRAPH statement 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*                  the plot model name referenced by the .GRAPH statements

*PLOT*                    the 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

### 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:<br>        Linear(LIN)       (XSCAL=1)<br>        Logarithm(LOG)  (XSCAL=2) |

| Name(Alias) | Default | Description |
|---|---|---|
| *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:<br>    Linear(LIN)     (YSCAL=1)<br>    Logarithm(LOG)  (YSCAL=2) |

### Examples

```
.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
```

*Avant!*

# 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
- Voltage and Current Controlled Elements
- Voltage Dependent Current Sources — G Elements
- Current Dependent Current Sources — F Elements
- Voltage Dependent Voltage Sources — E Elements
- Dependent Voltage Sources — H Elements
- Digital Files and Mixed Mode — U Elements

# Independent Source Elements

Use source element statements to specify either DC, AC, transient, or 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 keywords:

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

## Independent Source Element Statements

### Syntax

```
Vxxx n+ n- <<DC=> dcval> <tranfun> <AC=acmag, <acphase>>
```

or

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

where:

| *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* | DC source value, *dcval*. The "tranfun" value at time zero overrides the DC value (default=0.0). |
| *tranfun* | transient source function (AC, ACI, AM, DC, EXP, PE, PL, PU, PULSE, PWL, R, RD, SFFM, SIN) |
| *AC* | indicates source is to be used in an AC small-signal analysis |
| *acmag* | AC magnitude |
| *acphase* | AC phase (default=0.0) |
| *M* | multiplier used for simulating multiple parallel current sources. The current value is multiplied by M (default=1.0). |

**Examples**

```
VX 1 0 5V
VB 2 0 DC=VCC
VH 3 6 DC=2 AC=1,90
IG 8 7 PL(1MA 0S 5MA 25MS)
VCC 10 0 VCC PWL 0 0  10NS VCC  15NS VCC  20NS 0
VIN 13 2 0.001 AC 1 SIN (0 1 1MEG)
ISRC 23 21 AC 0.333 45.0 SFFM (0 1 10K 5 1K)
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 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.

### Syntax

```
PULSE <(>v1 v2 <td <tr <tf <pw <per>>>>> <)>
```

or

```
PU <(>v1 v2 <td <tr <tf <pw <per>>>>> <)>
```

where:

| | |
|---|---|
| *v1* | initial value of the voltage or current, before the pulse onset |
| *v2* | pulse plateau value |
| *td* | delay time in seconds from the beginning of transient interval to the first onset ramp (default=0.0 and for td < 0.0, td=0.0 is used as well) |
| *tr* | duration of the onset ramp, from the initial value to the pulse plateau value (reverse transit time) (default=TSTEP) |
| *tf* | duration of the recovery ramp, from the pulse plateau back to the initial value (forward transit time) (default=TSTEP) |
| *pw* | pulse width (the width of the pulse plateau portion of the pulse) (default=TSTEP) |
| *per* | pulse repetition period in seconds (default=TSTEP, the transient timestep) |

**Example 1**

```
VIN 3 0 PULSE (-1 1 2NS 2NS 2NS 50NS 100NS)
V1 99 0 PU lv hv tdlay tris tfall tpw tper
```

The first example specifies a 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. The second example specifies pulse value parameters in the .PARAM statement.

*Single pulse*

| *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 2**
```
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
```
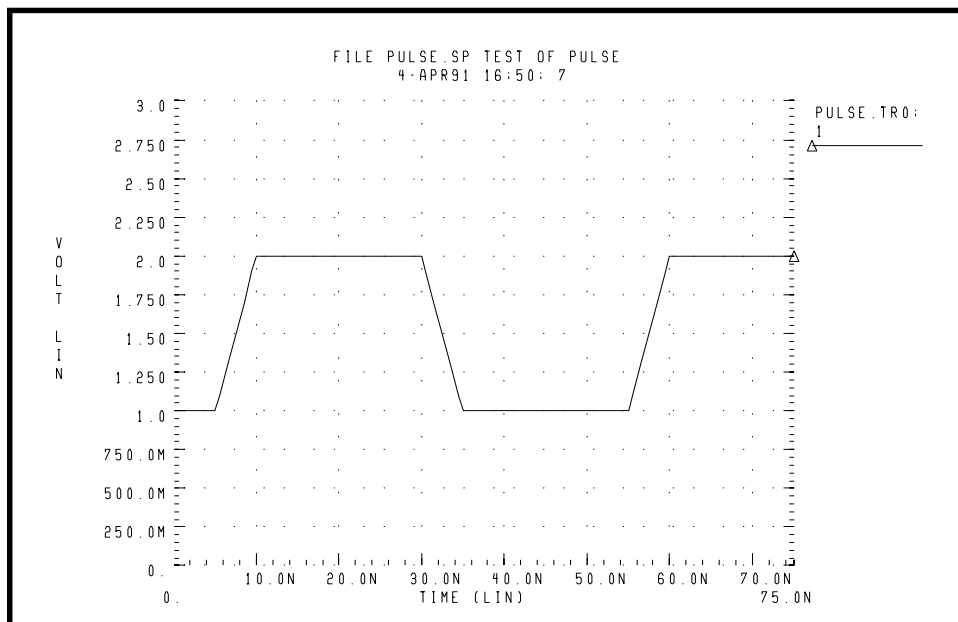


**Figure 5-1: Pulse Source Function**

## 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.

### Syntax

```
SIN <(>vo va <freq <td <θ<φ>>>> <)>
```

where:

| | |
|---|---|
| *vo* | voltage or current offset in volts or amps |
| *va* | voltage or current amplitude in volts or amps |
| *freq* | frequency in Hz (default=1/TSTOP) |
| *td* | delay in seconds (default=0.0) |
| θ | damping factor in 1/seconds (default=0.0) |
| φ | phase delay in degrees (default=0.0) |

### Example

```
VIN 3 0 SIN (0 1 100MEG 1NS 1e10)
```

The example specifies a 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 degrees.

***Waveform shape***

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

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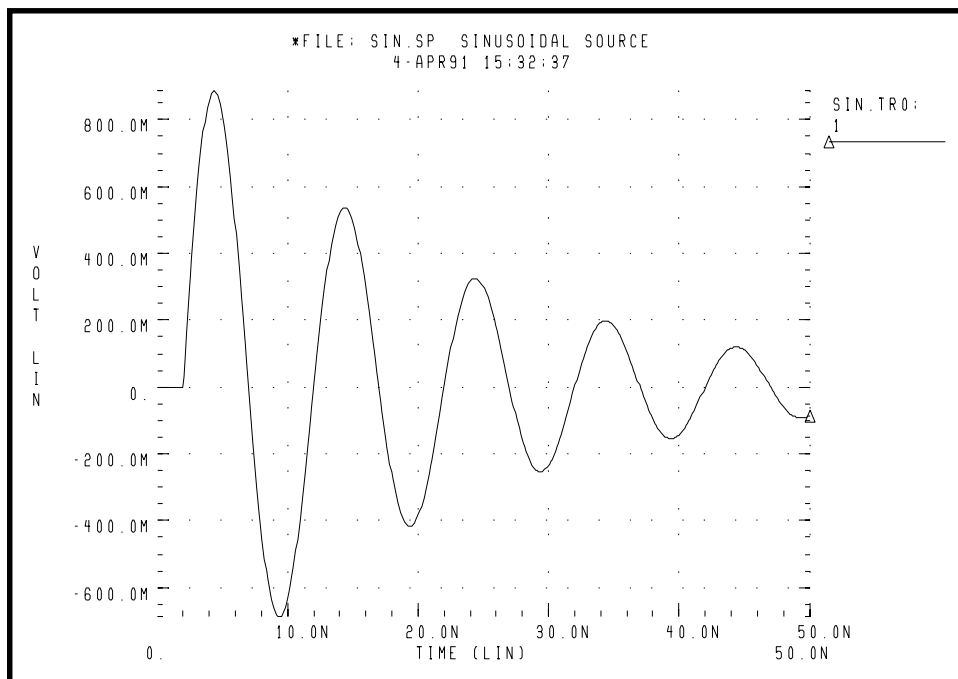


**Figure 5-2: Sinusoidal Source Function**

**Example**

```
*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
```

## Exponential Source Function

### Syntax

```
EXP <(>v1 v2 <td1 <τ1 <td2 <τ2>>>> <)>
```

where:

| | |
|---|---|
| *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) |

### 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.

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

### *Waveform shape*

| *Time* | *Value* |
|---|---|

| | |
|---|---|
| *0 to td1* | $v1$ |

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

*td2 to tstop*
$$v1 + (v2 - v1) \cdot \left[ 1 - Exp\left( -\frac{td2 - td1}{\tau_1} \right) \right] \cdot$$
$$Exp\left[ \frac{-(Time - td2)}{\tau_2} \right]$$
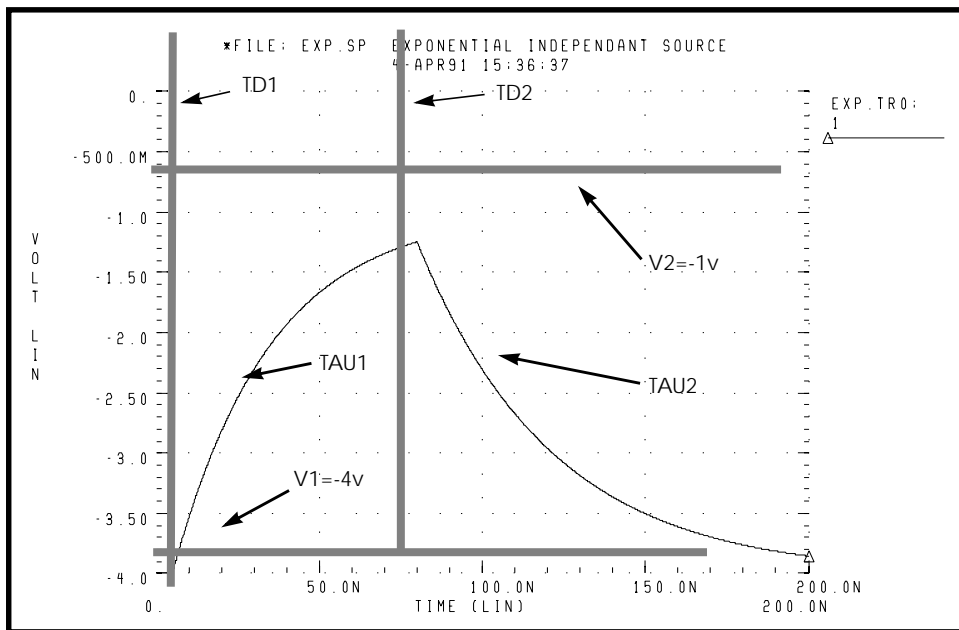


**Figure 5-3: Exponential Source Function**

**Example**

```
*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
```

# Piecewise Linear Source Function

### Syntax

```
PWL <(>t1 v1 <t2 v2 t3 v3…> <R <=repeat>> <TD=delay> <)>
```

### *MSINC and ASPEC*

```
PL <(>v1 t1 <v2 t2 v3 t3…> <R <=repeat>> <TD=delay> <)>
```

where

| | |
|---|---|
| *v1 …* | specifies current or voltage values |
| *t1 …* | specifies segment time values |
| *R* | causes the function to repeat |
| *repeat* | specifies the start point of the waveform which is to be repeated |
| *TD* | is keyword for time delay before piecewise actually starts |
| *delay* | 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) 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.

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
```

```
        FILE PWL.SP REPEATED PIECEWISE LINEAR SOURCE
                    4-APR91 16:35:16
```
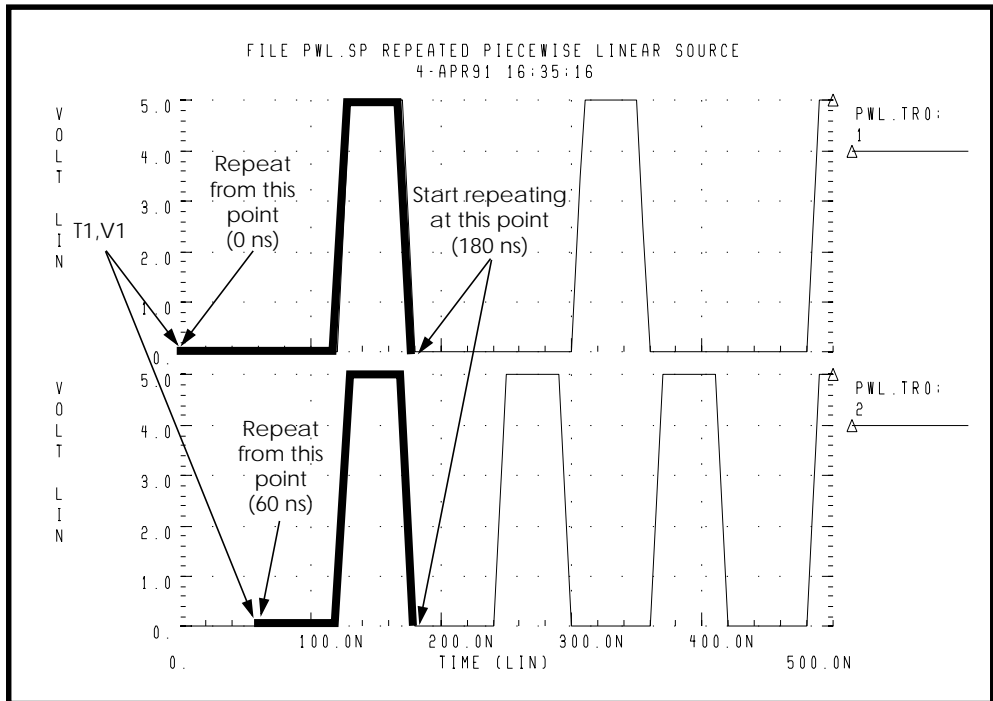
**Figure 5-4: Results of Using the Repeat Function**

## Data Driven Piecewise Linear Source Function

### Syntax

```
PWL (TIME, PV)
.DATA datanam
TIME PV
t1    v1
t2    v2
t3    v3
t4    v4
...   ...
.ENDDATA
```

```
.TRAN DATA=datanam
```

where

| | |
|---|---|
| *TIME* | parameter for a time value provided in a .DATA statement |
| *PV* | parameter for an 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 pv1pv2
0    5v 0v
5n   0v 5v
10n  0v 5v
.ENDDATA
.TRAN DATA=dsrc
.END
```

## Single-Frequency FM Source Function

### Syntax

```
SFFM <(> vo va <fc <mdi <fs>>> <)>
```

where

| | |
|---|---|
| *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 (default=0.0) |
| *fs* | signal frequency in Hz (default=1/TSTOP) |

*Waveform shape*

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

*Note: TSTOP is discussed in the .TRAN statement description.*

### Example

```
*FILE: SFFM.SP THE SINGLE FREQUENCY FM MODULATION
.OPTIONS POST
V 1 0 SFFM (0, 1M, 20K. 10, 5K)
R 1 0 1
.TRAN .0005M .5MS
.END
```
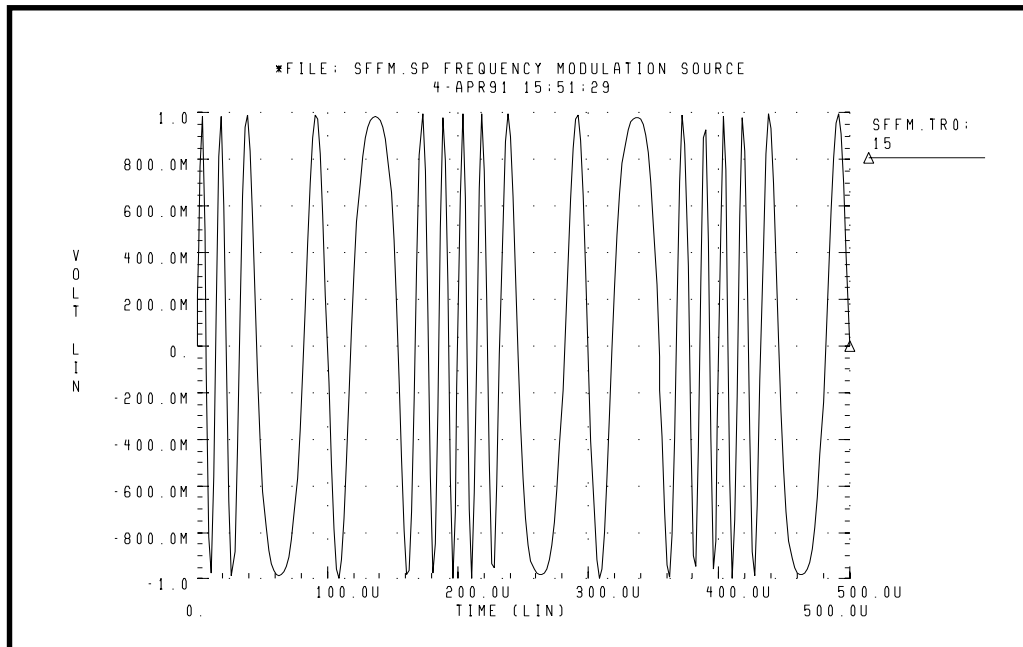
**Figure 5-5: Single Frequency FM Modulation**

## Amplitude Modulation Source Function

### Syntax

```
AM (sa oc fm fc td)
```

where

| | |
|---|---|
| *sa* | signal amplitude (default=0.0) |
| *fc* | carrier frequency (default=0.0) |
| *fm* | modulation frequency (default=1/TSTOP) |
| *oc* | offset constant (default=0.0) |
| *td* | delay time before start of signal (default=0.0) |

$$sourcevalue = sa \cdot \{oc + SIN[2 \cdot \pi \cdot fm \cdot (Time - td)]\} \cdot$$
$$SIN[2 \cdot \pi \cdot fc \cdot (Time - td)]$$

### 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
```
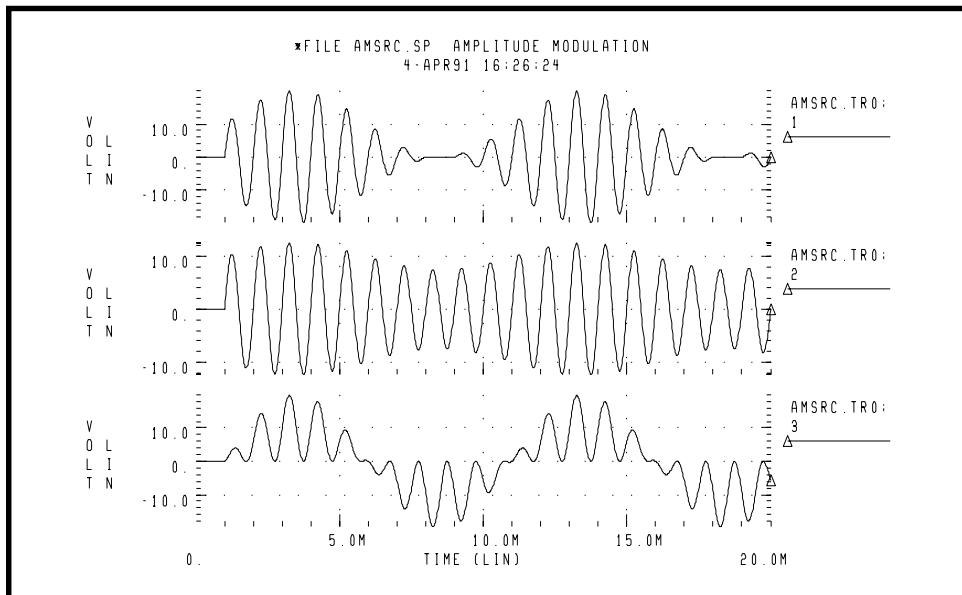


**Figure 5-6: Amplitude Modulation Plot**

# Voltage and Current Controlled Elements

Star-Hspice has four voltage and current controlled elements, known as E, F, G and H 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 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 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 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:

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

*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.*

### One-Dimensional 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.

### Two-Dimensional 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:

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

### Three-Dimensional Example

For example, generate a voltage controlled source that gives the voltage as:

$$
V(1, 0) \ = \ 3 \cdot V(3,2) + 4 \cdot V(7,6)^2 + 5 \cdot V(9,8)^3
$$

from the above defining equation and the three-dimensional polynomial equation:

$$
FA \ = \ V(3,2)
$$

$$
FB \ = \ V(7,6)
$$

$$
FC \ = \ V(9,8)
$$

$$
P1 \ = \ 3
$$

$$
P7 \ = \ 4
$$

$$
P19 \ = \ 5
$$

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 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 an 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 represents 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, page -20 in , Using Sources and Stimuli*). |
| *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 93A. |

| | |
|---|---|
| *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 |

*transconductance*    voltage-to-current conversion factor

*transfactor*    voltage-to-resistance conversion factor

*VCCAP*    the keyword for voltage controlled capacitance element. VCCAP is a reserved word and should not be used as a node name.

*VCCS*    the keyword for voltage controlled current source. VCCS is a reserved word and should not be used as a node name.

*VCR*    the 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

**Examples**

*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)'
```

# 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 | the 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

**Examples**
```
F1 13 5 VSENS MAX=+3 MIN=-3 5
```

This 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
```

This 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.

# 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' in+ in- <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, page -20 in , Using Sources and Stimuli*). |
| *POLY* | polynomial keyword function |
| *PWL* | piecewise linear keyword function |
| *SCALE* | element value multiplier |

| | |
|---|---|
| TC*1,TC2* | first and second order temperature coefficients. The SCALE is updated by temperature:<br>$$\text{SCALEeff} = \text{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$ |
| *TD* | time delay keyword |
| *TRANSFORMER* | the keyword for ideal transformer. TRANS is a reserved word and should not be used as a node name. |
| *VCVS* | the 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 |

**Examples**

*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)'
```

# 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* | the 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:<br>$\qquad$ 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 |

### Examples

```
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).

# Digital Files and Mixed Mode — U Elements

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
```

# Digital Input Element and Model

### Syntax

U<name> <(interface) node> <(lo_ref)node> <(hi_ref)node> <(model) name>
+          SIGNAME = <(digital signal) name>] [IS = (initial state)]

### Example
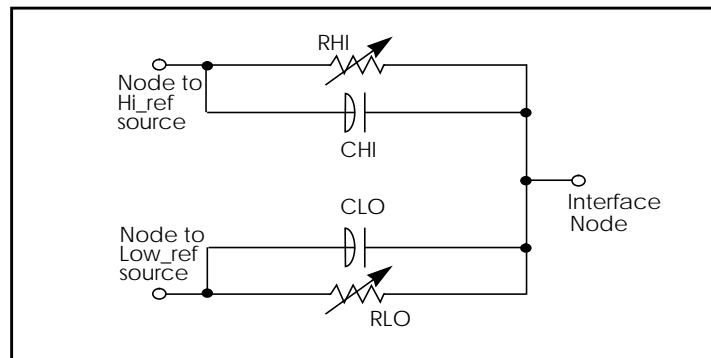
```
UC carry-in VLD2A VHD2A D2A SIGNAME=1 IS=0
VLD2A VLD2A 0 DC lo
VHD2A VHD2A 0 DC hi
```

### Model Form

.MODEL <(model) name> U LEVEL=5 [(model parameters)]

### Examples

```
.MODEL D2A U LEVEL=5 TIMESTEP=0.1NS,
+ 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
```

## Figure 5-7: Digital-to-Analog Converter Element

## Digital-to-Analog Model Parameters

| Names(Alias) | Units | Default | Description |
|---|---|---|---|
| CLO | farad | 0 | Capacitance to low level node |
| CHI | farad | 0 | Capacitance to high level node |
| SONAME | | | State "O" character abbreviation |
| SOTSW | sec | | State "O" switching time |
| SORLO | ohm | | State "O" resistance to low level node |
| SORHI | ohm | | State "O" resistance to high level node |
| S1NAME | | | State "1" character abbreviation |
| S1TSW | sec | | State "1" switching time |
| S1RLO | ohm | | State "1"resistace to low level node |
| S1RHI | ohm | | State "1" resistance to high level node |
| S2NAME | | | State "2" character abbreviation |
| S2TSW | sec | | State "2" switching time |
| S2RLO | ohm | | State "2" resistance to low level node |
| S2RHI | ohm | | State "2" resistance to high level node |
| S19NAME | | | State "19" character abbreviation |
| S19TSW | sec | | State "19" switching time |
| S19RLO | ohm | | Sstate "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) |

## Digital Outputs

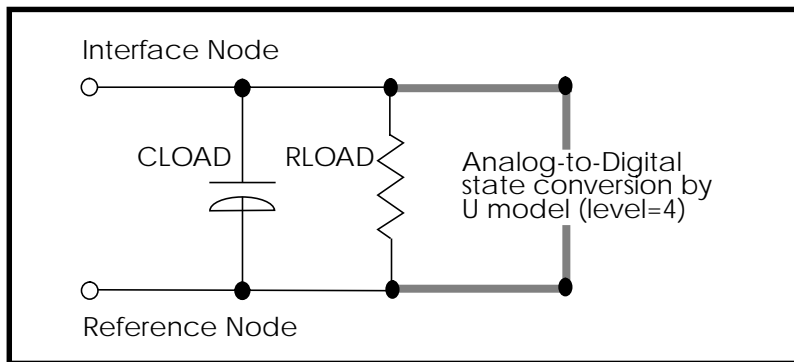### Syntax: analog-to-digital output

```
U<name> <(interface)node> <(reference) node> <(model) name>
+    [SIGNAME = <(digital signal) name>]
```

### Examples

```
  vref VREFA2D 0 DC 0.0V
  uco carry-out_2 VREFA2D a2d signame=12
```



**Figure 5-8: Analog-to-Digital Converter Element**

### Model Form

```
.MODEL < name> U LEVEL=4 [(model parameters)]
```

### Examples

```
* 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
```
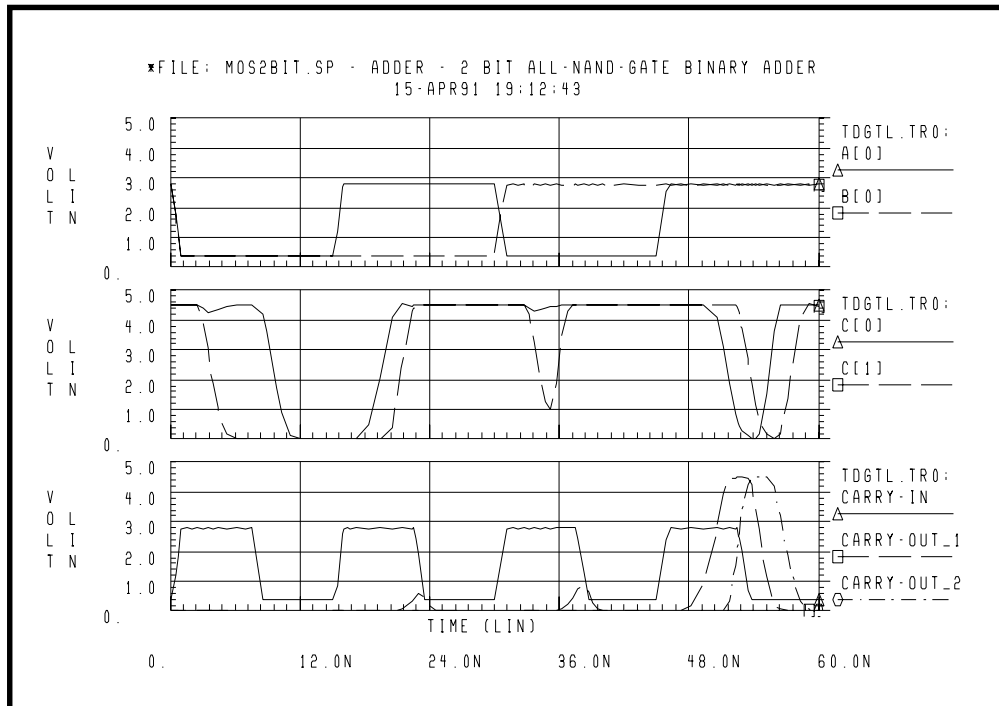
```
+ CLOAD=0.05pf
```

## Analog-to-Digital Output Model Parameters

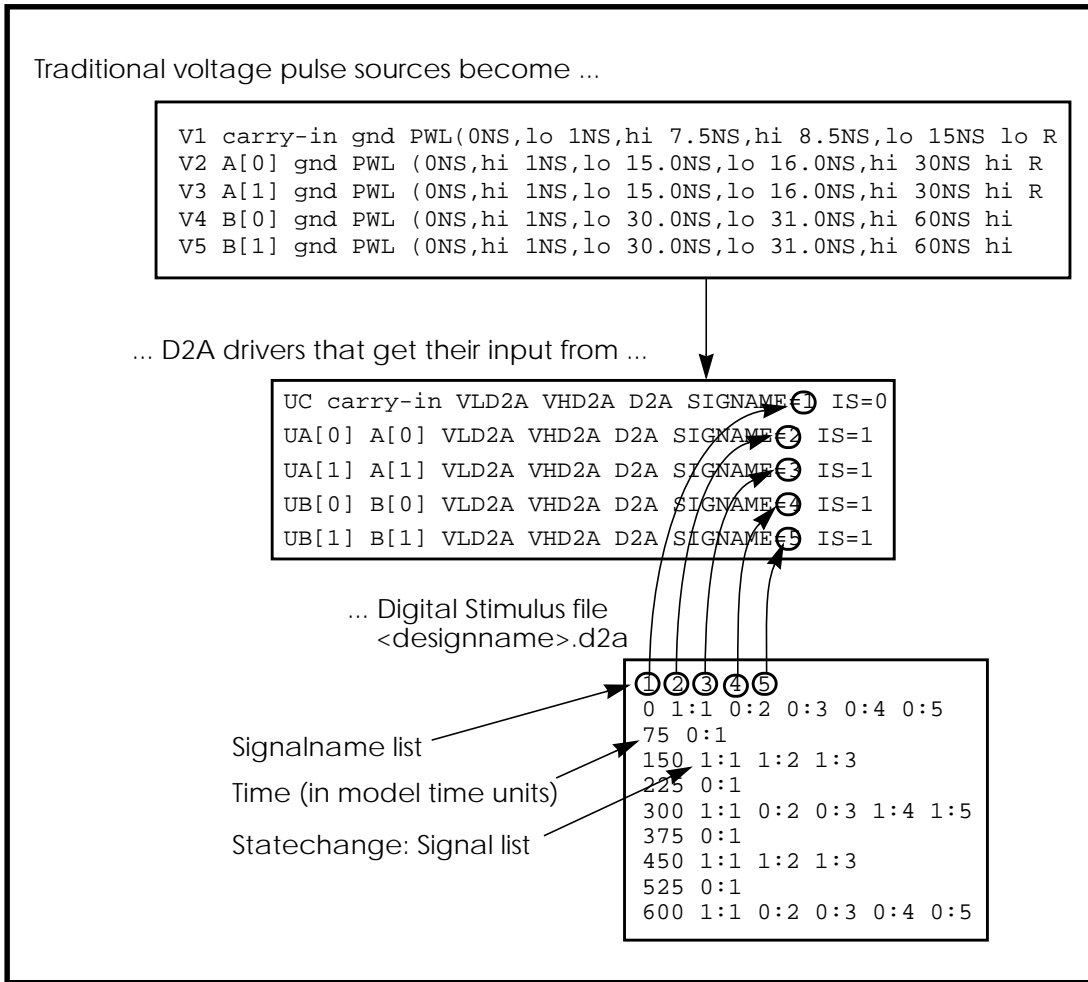| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| RLOAD | ohm | 1/gmin | Output resistor |
| CLOAD | farad | 0 | Output capacitor |
| CHGONLY | | 0 | 0: write each timestep, 1: write upon change |
| SONAME | | | State "0" character abbreviation |
| SOVLO | volt | | State "0" low level voltage |
| SOVHI | volt | | State "0" high level voltage |
| S1NAME | | | State "1" character abbreviation |
| S1VLO | volt | | State "1" low level voltage |
| S1VHI | volt | | Sstate "1" high level voltage |
| S2NAME | | | State "2" character abbreviation |
| S2VLO | volt | | State "2" low level voltage |
| S2VHI | volt | | State "2" high level voltage |
| S19NAME | | | State "19" character abbreviation |
| 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 |

## Two-Bit Adder with Digital I/O

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. SPICE outputs a digital file.



**Figure 5-9: Digital Stimulus File Input**

## Replacing Sources With Digital Inputs

Traditional voltage pulse sources become ...

```
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
```

... D2A drivers that get their input from ...

```
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
```

... Digital Stimulus file
<designname>.d2a

```
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
```

Signalname list

Time (in model time units)

Statechange: Signal list

**Figure 5-10: Digital File Signal Correspondence**

## Example of MOS 2 Bit Adder

```
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
```

## 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 Q — their names, sizes, signal direction, and so on Q — 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', etc. 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 Outputs

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.

**Examples**

```
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 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 56 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.

## The 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
```

## The 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 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]
```

Tne *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]
```

## The 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, bi-directional, 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
```

## The Tunit Statement

The *tunit* statement defines the time unit in 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
```

## The 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.

## The Enable Statement

The *enable* statement specifies the controlling signal(s) of bi-directional signals and is absolutely required for all bi-directional signals. If more than one *enable* statement exists, the last value will overrule the previous ones, with a warning message will be issued.

The syntax is a keyword *enable,* followed by the controlling signal name and the mask that defines the (bi-directional) signals to which *enable* applies.

The controlling signal of bi-directional signals must be an input signal with radix of 1. The bi-directional 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 bi-directional, 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 bi-directional 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.

### The 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
```

## The 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.

**Examples**

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
```

### The 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.

### The 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.

### The 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.

### The 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.

## The 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.

## The 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.

## The 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.

**The 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.

## The 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).*

## The 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

# DC Initialization and 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
- Setting DC Initialization Control Options
- Specifying Accuracy and Convergence
- Reducing DC Errors
- Diagnosing Convergence

# Understanding the Simulation Flow

Figure Figure 6-1 illustrates the simulation flow for Star-Hspice.
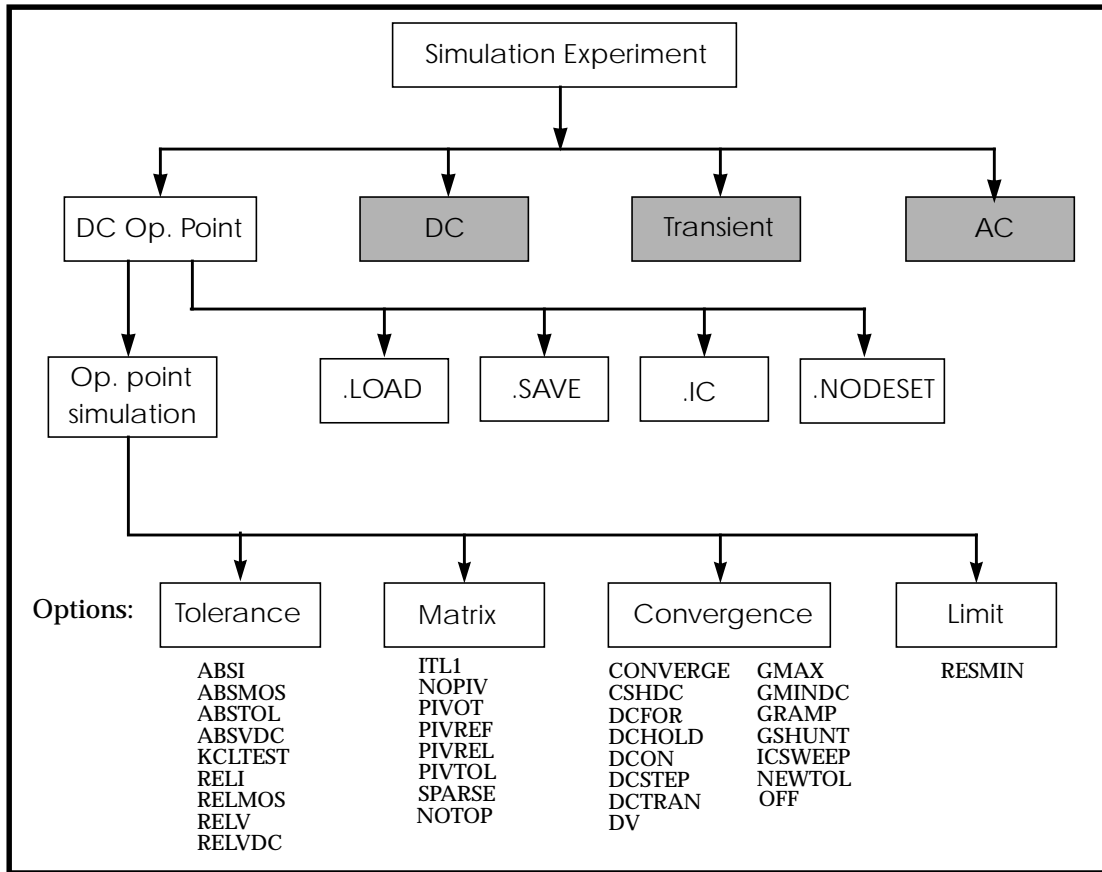


**Figure 6-1: DC Initialization and Operating Point Analysis
Simulation Flow**

# 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.
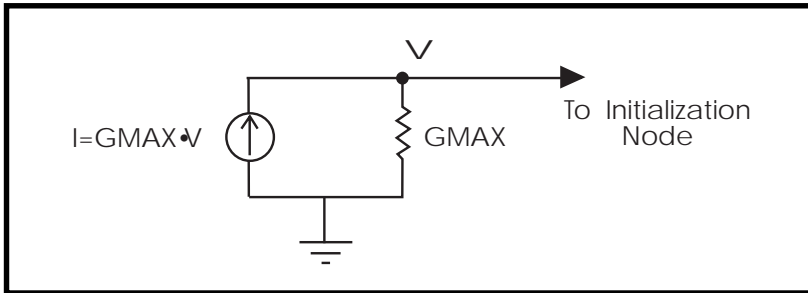
**Figure 6-2: Equivalent Voltage Source: NODESET and .IC**

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 (BJT's, diodes, MOSFET's, JFET's, MESFET's) 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 a .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

## 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 timepoint 0.

**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

### Examples

```
.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
```

# .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.

### 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

### Examples

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
```

### Example 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.
```

## 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.

# 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 ".OPTIONS Statement" on page 3-45.

The following options are associated with controlling DC operating point analysis. They are described in this section.

| | | |
|---|---|---|
| ABSTOL | GRAMP | OFF |
| CAPTAB | GSHUNT | PIVOT |
| CSHDC | ICSWEEP | PIVREF |
| DCCAP | ITL1 | PIVREL |
| DCFOR | KCLTEST | PIVTOL |
| DCHOLD | MAXAMP | RESTOL |
| DCSTEP | NEWTOL | SPARSE |
| DV | NOPIV | |

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 *Chapter 7, Performing Transient Analysis*.

## 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.<br>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, page -23 in , DC Initialization and Point Analysis*.

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. 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. |
| *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 $i_b$s are the node currents.

| | |
|---|---|
| MAXAMP=x | sets the maximum current through voltage defined branches (voltage sources and inductors). If the current exceeds the MAXAMP value, an error is issued. Default=0.0. |

| | |
|---|---|
| *NEWTOL* | calculates one more iterations past convergence for every DC solution and timepoint circuit solution calculated. When NEWTOL is not set, once convergence is determined, the convergence routine is ended and the next program step begun. Default=0. |
| *NOPIV* | prevents Star-Hspice from switching automatically to pivoting matrix factorization when a nodal conductance is less than PIVTOL. NOPIV inhibits pivoting. Also see PIVOT. |
| *OFF* | initializes the terminal voltages of all active devices to zero if they are not initialized to other values. For example, if the drain and source nodes of a transistor are not both initialized using .NODESET or .IC statements or by connecting them to sources, then the OFF option initializes all of the nodes of the transistor to zero. The OFF option is checked before element IC parameters, so if an element IC parameter assignment exists for a particular node, the node is initialized to the element IC parameter value even if it was previously set to zero by the OFF option. (The element parameter OFF can be used to initialize the terminal voltages to zero for particular active devices). |
| | The OFF option is used to help find exact DC operating point solutions for large circuits. |
| *PIVOT=x* | provides different pivoting algorithm selections. These can be used effectively to reduce simulation time and achieve convergence in circuits that produce hard-to-solve matrix equations. The pivot algorithm is selected by setting PIVOT to one of the following values: |

| | |
|---|---|
| 0 | Original nonpivoting algorithm |
| 1 | Original pivoting algorithm |
| 2 | Pick largest pivot in row algorithm |

| | |
|---|---|
| 3 | Pick best in row algorithm |
| 10 | Fast nonpivoting algorithm, more memory required |
| 11 | Fast pivoting algorithm, more memory required than for PIVOT values less than 11 |
| 12 | Pick largest pivot in row algorithm, more memory required than for PIVOT values less than 12 |
| 13 | Fast best pivot: faster, more memory required than for PIVOT values less than 13 |

Default=10.

The fastest algorithm is PIVOT=13, which can improve simulation time by up to ten times on very large circuits. However, the PIVOT=13 option requires substantially more memory for the simulation. Some circuits with large conductance ratios, such as switching regulator circuits, might need pivoting. If PIVTOL=0, Star-Hspice automatically changes from nonpivoting to a row pivot strategy upon detection of any diagonal matrix entry less than PIVTOL. This strategy provides the time and memory advantages of nonpivoting inversion, while avoiding unstable simulations and incorrect results. Use .OPTION NOPIV to prevent pivoting from being used under any circumstances.

For very large circuits, PIVOT=10, 11, 12, or 13 can require excessive memory.

If Star-Hspice switches to pivoting during a simulation, the message "pivot change on the fly" is printed, followed by the node numbers causing the problem. Use .OPTION NODE to obtain a node-to-element cross reference.

SPARSE is the same as PIVOT.

*PIVREF*            pivot reference. Used in PIVOT=11, 12, 13 to limit the size of the matrix. Default=1e+8.

*PIVREL=x*          sets the maximum/minimum row/matrix ratio. Use only for PIVOT=1. Large values for PIVREL can result in very long matrix pivot times. If the value is too small, however, no pivoting occurs. It is best to start with small values of PIVREL, using an adequate but not excessive value for convergence and accuracy. Default=1E-20 (max=1e-20, min=1).

*PIVTOL=x*          sets the absolute minimum value for which a matrix entry is accepted as a pivot. PIVTOL is used as the minimum conductance in the matrix when PIVOT=0. Default=1.0e-15.

                    **Note:** PIVTOL should always be less than GMIN or GMINDC. Values approaching 1 yield increased pivot.

*RESMIN=x*          specifies the minimum resistance value for all resistors, including parasitic and inductive resistances. Default=1e-5 (ohm). Range: 1e-15 to 10 ohm.

*SPARSE=x*          same as PIVOT

# 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 6-1:, 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 6-1:  Absolute and Relative Accuracy Tolerances

| Type | Option | Default |
|------|--------|---------|
| Nodal Voltage Tolerances | ABSVDC | 50 µv |
|  | RELVDC | .001 (.1%) |

**Table 6-1:  Absolute and Relative Accuracy Tolerances**

| Type | Option | Default |
|------|--------|---------|
| Current Element Tolerances | ABSI | 1 nA |
| | RELI | .01 (1%) |
| | ABSMOS | 1 uA |
| | RELMOS | .05 (5%) |

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 6-2: shows an example of the relationship between the RELVDC value and the number of iterations.

**Table 6-2:  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 |
| .05 | 386 | 31.203 | 14.365 | 1.3315 |

**Table 6-2:  RELV vs. Accuracy and Simulation Time for 2 Bit Adder**

| RELVDC | Iteration | Delay (ns) | Period (ns) | Fall time (ns) |
|--------|-----------|------------|-------------|----------------|
| .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 "Testing for Speed, Accuracy and Convergence" on
page 7-18.

## 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-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. 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 Pseudeo
                    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 for DC analysis. GMINDC helps overcome DC convergence problems caused by low values of off conductance for pn junctions and MOSFET devices. GRAMP can be used to reduce GMINDC by one order of magnitude for each step. GMINDC can be set between 1e-4 and PIVTOL. Default=1e-12. |
| | Large values of GMINDC can cause unreasonable circuit response. If large values are required for convergence, a bad model or circuit is suspect. In the event of a matrix floating point overflow, if GMINDC is 1.0e-12 or less, Star-Hspice sets it to 1.0e-11. |

GMINDC is manipulated by Star-Hspice in autoconverge mode, as described in the "Autoconverge Process" section following.

*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 change, in percent, 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=1 (0.01%) for KCLTEST=0, 1e-6 for KCLTEST=1.

*RELMOS=x*          sets the relative drain-to-source current error tolerance, in percent, from iteration to iteration to determine convergence for currents in MOSFET devices. (RELI sets the tolerance for other active devices.) This is the percent 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=5 (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 6-3:.

***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 a .OPTIONS statement has no effect on the autoconverge process. The autoconverge process sets GRAMP independently.

6.  If you specify a value for GMINDC in a .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.

**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

**Figure 6-3: Autoconvergence Process Flow Diagram**

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 6-3:. GMINDC is shown for various elements in Figure 6-4:.
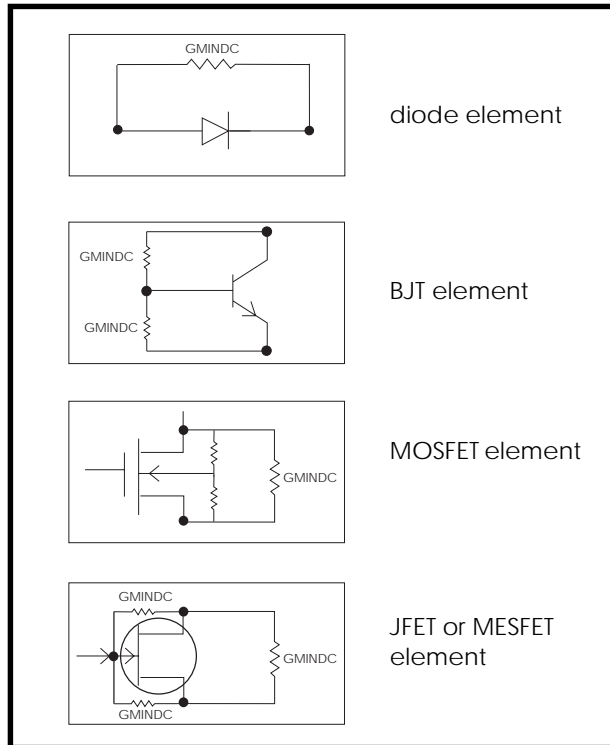


**Figure 6-4: GMINDC Insertion**

# 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 (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 6-5: illustrates how Star-Hspice inserts conductance G in parallel with capacitance Cg to provide current paths around capacitances in DC analysis.
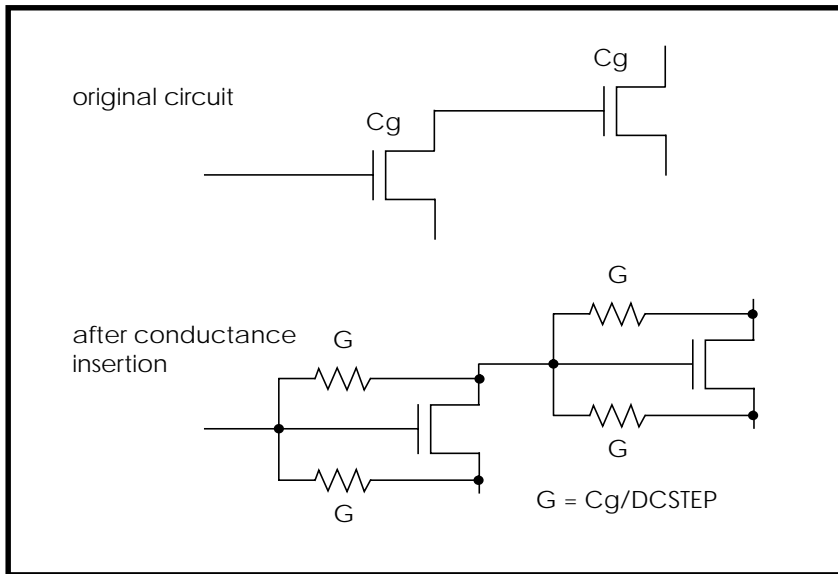
**Figure 6-5: Conductance Insertion**

## 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

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 6-3: 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 6-3:, 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 6-3: 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

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.
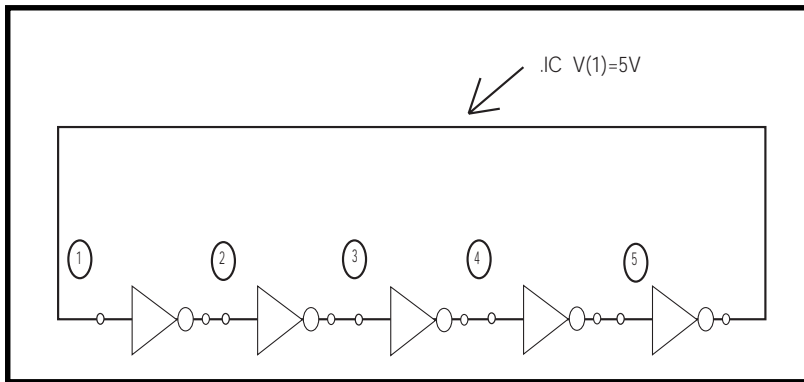
**Figure 6-6: Ring Oscillator**

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.
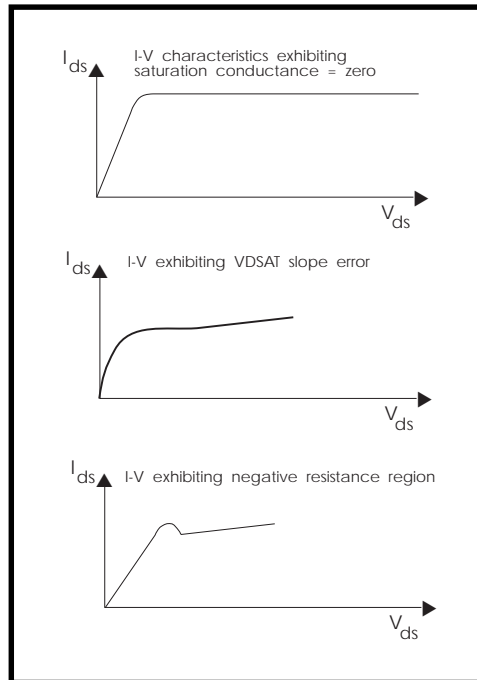
**Figure 6-7: Discontinuous I-V Characteristics**

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 7

# Performing Transient Analysis

Star-Hspice transient analysis computes the circuit solution as a function of time over a time range specified in the .TRAN statement.

This chapter covers the following topics:

- Understanding the Simulation Flow
- Understanding Transient Analysis
- Using the .TRAN Statement
- Understanding the Control Options
- Testing for Speed, Accuracy and Convergence
- Selecting Timestep Control Algorithms
- Performing Fourier Analysis
- Using Data Driven PWL Sources

# Understanding the Simulation Flow

Figure 7-1 illustrates the transient analysis simulation flow for Star-Hspice.
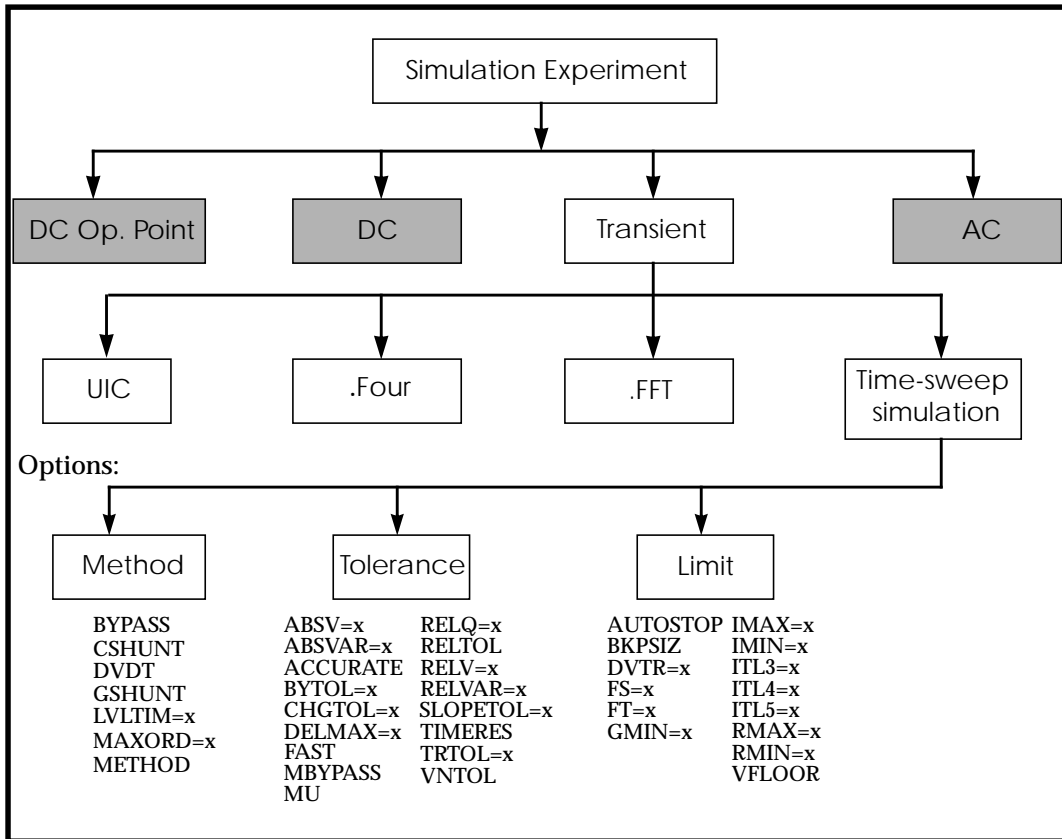


**Figure 7-1: Transient Analysis Simulation Flow**

# 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 points at t=0 and 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.

# 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<br>**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<br>**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="<br>**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. |

**Examples**

The following example performs and prints the transient analysis every 1 ns for 100 ns.

```
.TRAN 1NS 100NS
```

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
```

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
```

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
```

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
```

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=0.

        **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. INTERP typically produces a much smaller *design.tr#* file. |
| *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 *ei*ither 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*          same as VNTOL. See VNTOL.

*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=3DVDT=2RELVAR=0.2
                  ABSVAR=0.2FT=0.2RELMOS=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×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×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,*            the coefficient for trapezoidal integration. The range for MU
                   is 0.0 to 0.5.

*XMU=x*            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, in percent, 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=1 (%) 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 (1%). |
| *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,*        sets the absolute minimum voltage for DC and transient
*ABSV*            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*           Same as MU. See MU.

## 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, page -19 in , Performing Transient Analysis*. 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,*           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. Default=8.0. |
| *IMIN=x,* <br> *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* | same as IMIN. See IMIN. |
| *ITL4=x* | same as IMAX. See IMAX. |
| *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: <br><br> DELMAX=TSTEP×RMAX <br><br> Default=5 when dvdt=4 and lvltim=1, otherwise, default=2. |
| *RMIN=x* | sets the minimum value of Delta (internal timestep). An internal timestep smaller than RMIN×TSTEP results in termination of the transient analysis with the error message |

"internal timestep too small". Delta is decreased by the amount set by the FT option if the circuit has not converged in IMAX iterations. Default=1.0e-9.

*VFLOOR=x*          sets a lower limit for the voltages that are printed in the output listing. All voltages lower than VFLOOR are printed as 0. This only affects the output listing: the minimum voltage used in a simulation is set by VNTOL (ABSV).

## Matrix Manipulation Options

After linearization of the individual elements within a Star-Hspice input netlist file, the linear equations are constructed for the matrix. User-controlled variables affecting the construction and solution of the matrix equation include options PIVOT and GMIN. GMIN places a variable into the matrix that prevents the matrix becoming ill-conditioned.

### Pivot Option

Select the PIVOT option for a number of different pivoting methods to reduce simulation time and assist in both DC and transient convergence. Pivoting reduces the error resulting from elements in the matrix that are widely different in magnitude. The use of PIVOT results in a search of the matrix for the largest element value. This element value then is used as the pivot.

# Testing for Speed, Accuracy and Convergence

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" which starts on page 7-8.

## 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=0.25 | BYPASS=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, page -24 in , Performing Transient Analysis*.

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.

## 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 two options, Gear or Trapezoidal, to convert differential terms into algebraic terms.

### Syntax

Gear algorithm:

```
.OPTION METHOD=GEAR
```

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, 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.
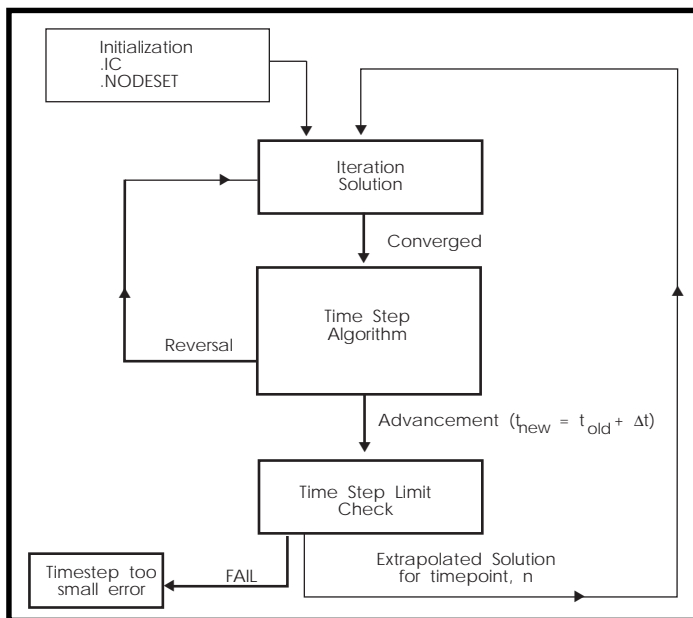
**Figure 7-2: Time Domain Algorithm**
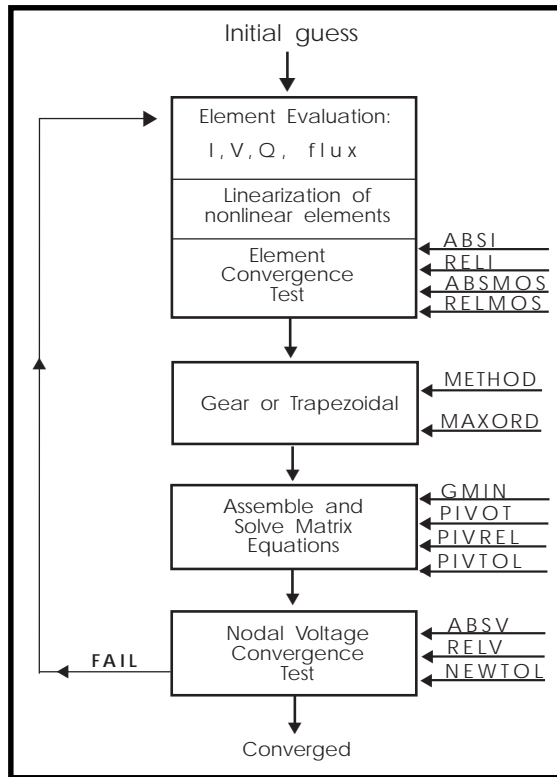
**Figure 7-3: Iteration Algorithm**

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
- Truncation
- DVDT

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.
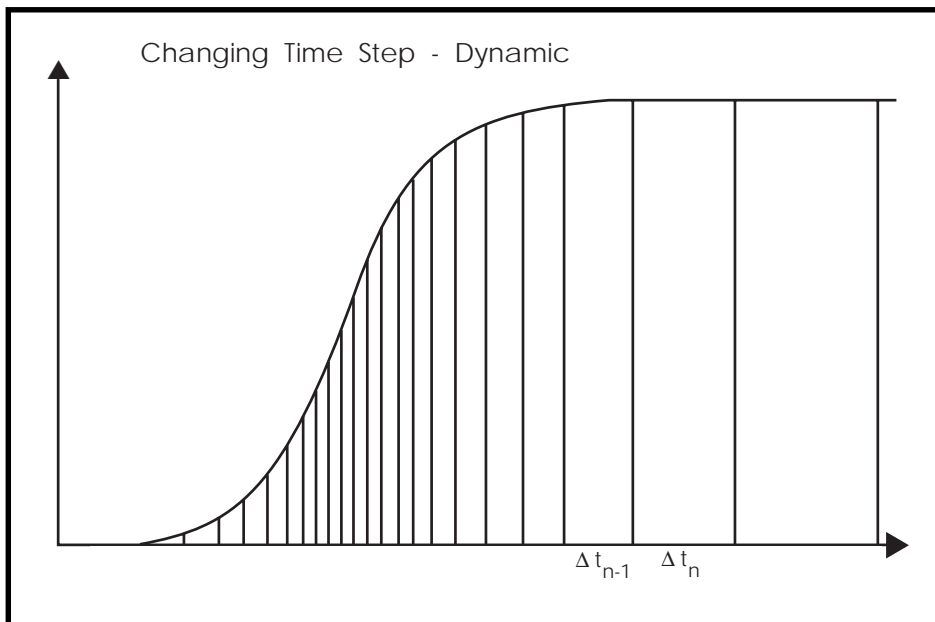


**Figure 7-4: Internal Variable Timestep**

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, page -26 in , Performing Transient Analysis*.

- 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, 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=0.050m

## 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 TSTEP×RMIN.

*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×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 illustrates the flow for Fourier and FFT analysis.
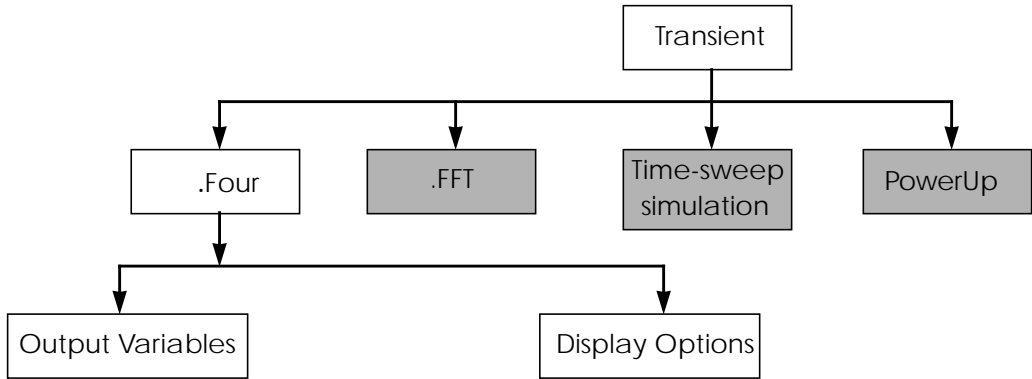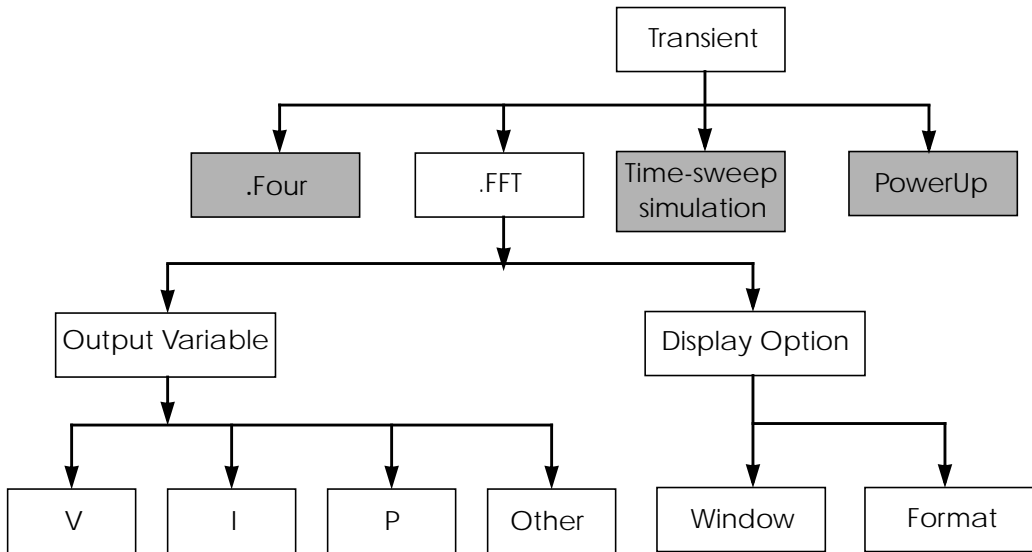


.FOUR Statement



.FFT Statement

**Figure 7-5: Fourier and FFT Analysis**

# .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 *Chapter 25, Performing FFT Spectrum Analysis*.

## .FFT Statement

The syntax of the .FFT statement is shown below. The parameters are described in Table 7-1:.

```
.FFT <output_var> <START=value> <STOP=value> <NP=value>
+   <FORMAT=keyword> <WINDOW=keyword> <ALFA=value> <FREQ=value>
+   <FMIN=value> <FMAX=value>
```

## Table 7-1: .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. |
| 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:<br>NORM=normalized magnitude<br>UNORM=unnormalized magnitude |
| WINDOW | RECT | specifies the window type to be used:<br><br>RECT=simple rectangular truncation window<br>BART=Bartlett (triangular) window<br>HANN=Hanning window<br>HAMM=Hamming window<br>BLACK=Blackman window<br>HARRIS=Blackman-Harris window<br>GAUSS=Gaussian window<br>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.<br>1.0 <= ALFA <= 20.0 |

## Table 7-1: .FFT Statement Parameters

| Parameter | Default | Description |
|---|---|---|
| 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. |

### .FFT Statement Syntax Examples

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 'v(1) + v(2)' from=0.2u stop=1.2u window=harris
```

Only one output variable is allowed in a .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 7-2: describes the output of the Star-Hspice FFT analysis.

### Table 7-2: .FFT 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 | 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 |

*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 *Chapter 25, Performing FFT Spectrum Analysis*.

# Using Data Driven PWL Sources

Data driven PWL sources allow the results of an experiment or a previous simulation to provide one or more input sources for a transient simulation.

### Syntax

```
source n+ n- PWL (param1, param2)
.DATA dataname
param1 param2
val    val
val    val
...    ...
.ENDDATA
.TRAN DATA=dataname
```

where

| | |
|---|---|
| *source* | name of the PWL source |
| n+ n- | positive and negative nodes on the source |
| PWL | piecewise linear source keyword |
| *param1* | parameter name for a value provided in a .DATA statement |
| *param2* | parameter name for a value provided in a .DATA statement |
| .DATA | inline data statement |
| dataname | name of the inline .DATA statement |
| val | value for param1 or param2 |

The named PWL source must be used with a .DATA statement that contains *param1-param2* value pairs. This source allows the results of one simulation to be used as an input source in another simulation. The transient analysis must be data driven.

**Example**

```
vin in 0 PWL(timea, volta)
.DATA timvol
0    0
1    5
20   5
21   0
.enddata
```

In the example above, *vin* is a piecewise linear source with an output of 0 V from time 0 to 1 ns, 5 V from time 1 ns to time 20 ns, and 0 V from time 21 ns onward.

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

*Avant!*

# Chapter 8

# Using the .DC Statement

This chapter describes how to use the .DC statement to perform DC sweep analysis. It covers the following topics:

- Understanding the DC Sweep Flow
- Performing DC Sweeps
- Using Other DC Analysis Statements
- Setting the DC Analysis Control Option

# Understanding the DC Sweep Flow

```
┌──────────────────────────────────────────────────────────────────┐
│                      ┌─────────────────────────┐                   │
│                      │   Simulation Experiment  │                  │
│                      └─────────────────────────┘                   │
│                                                                    │
│   ┌──────────────┐   ┌────────┐   ┌───────────┐   ┌────────┐      │
│   │ DC Op. Point │   │   DC   │   │ Transient │   │   AC   │      │
│   └──────────────┘   └────────┘   └───────────┘   └────────┘      │
│                                                                    │
│  ┌─────────────────┐ ┌───────────────┐ ┌───────────────┐         │
│  │  DC-related AC   │ │ Sweep analysis │ │  Monte Carlo  │         │
│  │ small-signal     │ │                │ │   analysis    │         │
│  │ analysis         │ └───────────────┘ └───────────────┘         │
│  └─────────────────┘                                               │
│    .SENS                                                           │
│    .TF                                                             │
│    .PZ                                                             │
│                                                                    │
│   ┌─────────────┐   ┌─────────────┐   ┌────────────────┐         │
│   │ Sweep range │   │ Sweep type  │   │ Sweep variables │        │
│   └─────────────┘   └─────────────┘   └────────────────┘         │
│        start            linear              temp                  │
│        stop             decade              isource               │
│        increment        octave              vsource               │
│                         poi                 user-parameters       │
└──────────────────────────────────────────────────────────────────┘
```
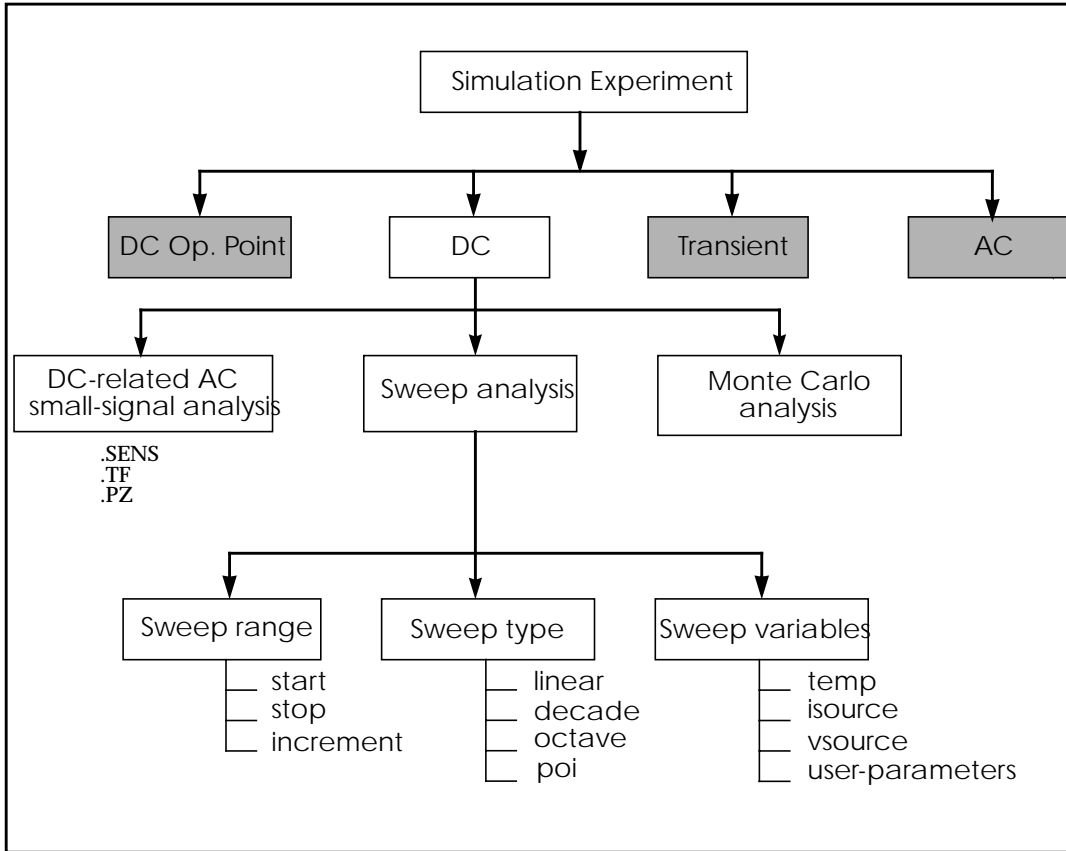
**Figure 8-1: DC Sweep Overview**

# Performing 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 **Note:** 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. |

**Examples**

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.

| | |
|---|---|
| *.OP* | specifies the time or times at which an operating point is to be calculated. See Chapter 5, "DC Initialization and Operating Point Analysis," for information about DC operating point analysis. |
| *.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.

## DC-Related Small-Signal Analyses

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:   IThe .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

### Examples

```
.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

### Examples

```
.PZ   V(10)   VIN
.PZ   I(RL)   ISORC
```

See Chapter 23, "Pole/Zero Analysis," for complete information about pole/zero analysis.

**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 · FSCAL. |
| *FSCAL* | sets the frequency scale. Frequency is multiplied by FSCAL. Default=1e-9. |
| *GSCAL* | sets the conductance scale. Conductances are multiplied by GSCAL. Resistances are divided by GSCAL. Default=1e+3. |
| *ITLPZ* | sets the pole/zero analysis iteration limit. Default=100. |
| *LSCAL* | sets inductance scale. Inductances are multiplied by LSCAL. Default=1e+6. |

The scale factors must satisfy the following relations:

$$GSCA \ = \ CSCAL \cdot FSCAL$$

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

If scale factors are changed, it might be necessary to modify the initial Muller points (X0R, X0I), (X1R, X1I) and (X2R, X2I), even though Star-Hspice multiplies initial values by (1e-9/GSCAL).

| | |
|---|---|
| *PZABS* | sets absolute tolerances for poles and zeros. This option affects the low frequency poles or zeros. It is used as follows: |

If $(|X_{real}| + |X_{imag}| < PZABS)$,

then $X_{real} = 0$ and $X_{imag} = 0$.

It is also used for convergence tests. Default=1e-2.

| | |
|---|---|
| *PZTOL* | sets the relative error tolerance for poles or zeros. Default=1.0e-6. |
| *RITOL* | sets the minimum ratio value for (real/imaginary) or (imaginary/real) parts of the poles or zeros. RITOL is used as follows: |

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

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

Default=1.0e-6.

| | |
|---|---|
| *(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<br>X2R = +.23456e6    X2I = 0.0 |

These initial points are multiplied by FSCAL.

## Setting the DC Analysis Control Option

The following control option is used in DC small signal analysis.

| | |
|---|---|
| *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. |

*Avant!*

# Chapter 9

# AC Sweep and 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 9-1) 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.
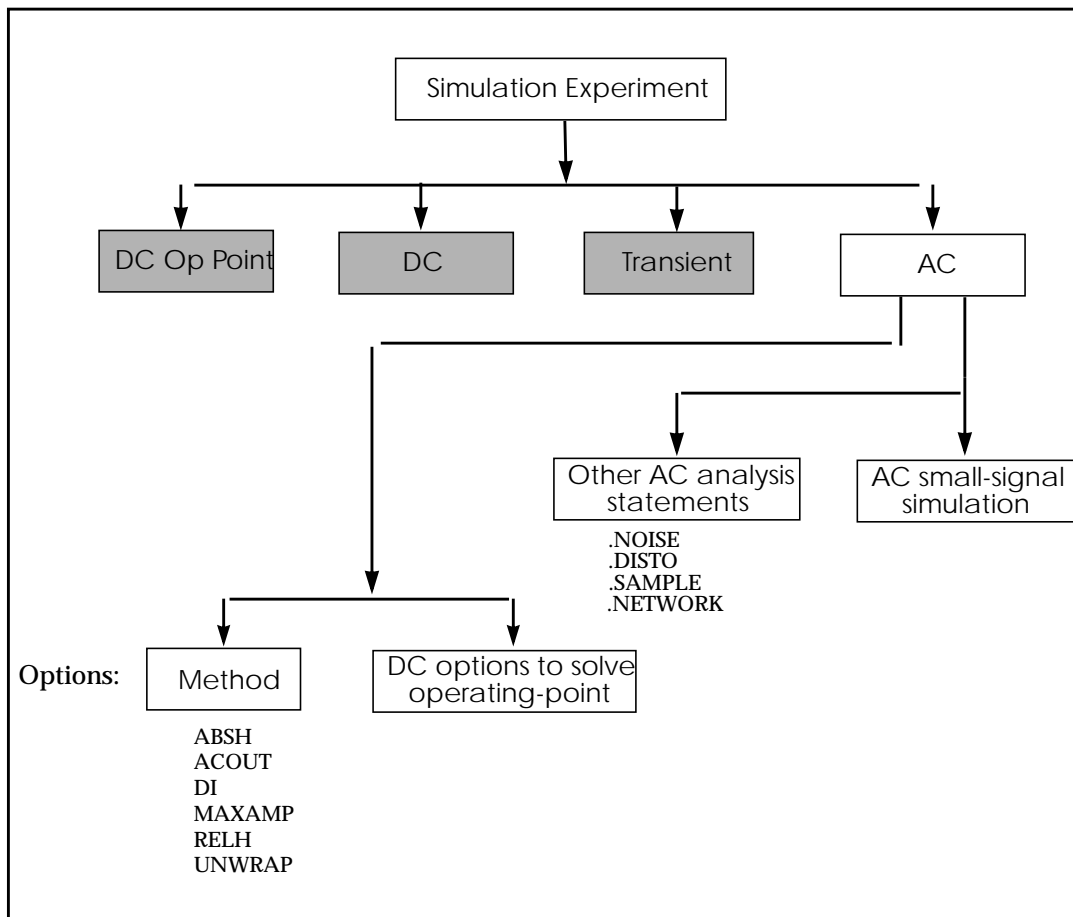


**Figure 9-1: AC Small Signal Analysis Flow**

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 the BJT Models (NPN and PNP)" on page 14-33. MOSFET AC equivalent circuit models are described in *Chapter , Introducing MOSFET*.

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 starstop 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<br>**Note:** If "type" variation is used, the "np" (number of points) is specified instead of "incr". |
| *fstart* | starting frequency<br>**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 10-39 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:<br>　　DEC – decade variation<br>　　OCT – octave variation<br>　　LIN – linear variation<br>　　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.

### Examples

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 *Chapter , Using Sources and Stimuli*).

## 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.<br><br>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.

## 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 9-1:. 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 9-1: AC Output Variable Types.

| Type Symbol | Variable Type  |
|-------------|----------------|
| DB          | decibel        |
| I           | imaginary part |
| M           | magnitude      |
| P           | phase          |
| R           | real part      |
| T           | group delay    |

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

## AC Nodal Voltage Output

### Syntax

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

where:

| | |
|---|---|
| *x* | specifies the voltage output type (see Table 9-1:) |
| *n1, n2* | specfies 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 interstage gain in a circuit (such as an amplifier circuit) and to compare its gain, phase, and magnitude.

The following examples define the AC analysis output variables for the Star-Hspice and then for the SPICE method.

### 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:

$VM(N1,0)$ $= [VR(N1,0)^2 + VI(N1,0)^2]^{0.5}$

$VM(N2,0)$ $= [VR(N2,0)^2 + VI(N2,0)^2]^{0.5}$

$VM(N1,N2)$ $= VM(N1,0) - VM(N2,0)$

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:

$VDB(N1,N2) = 20 \cdot LOG10(VM(N1,0)/VM(N2,0))$

### 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:

$VDB(N1,N2) = 20 \cdot LOG10[VM(N1,N2)]$

## AC Current Output: Independent Voltage Sources

### Syntax

```
Iz (Vxxx)
```

where:

| | |
|---|---|
| *z* | the current output type (see Table 9-1:) |
| *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.Vyyy). |

### Example

```
.PLOT AC IR(V1) IM(VN2B) IP(X1.X2.VSRC)
```

## AC Current Output: Element Branches

### Syntax

```
Izn (Wwww)
```

where:

| | |
|---|---|
| *z* | current output type (see Table 9-1:) |
| *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.Qyyy). |

**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 Output

### 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
```

## AC Network Output

### 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 9-1:). 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 |

**Examples**

```
.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)
```

# 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").<br><br>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.<br><br>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.<br><br>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-3 < skw2 ≤ 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: |

$$FMAX = MAXFLD \cdot FS$$

Default=10.0.

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

> BETA=0    no integrator
>
> 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. |

## Examples

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:<br>R: real part<br>I : maginary part<br>M: magnitude<br>P: phase<br>DB: decibel<br>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.

### Examples

```
.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
```
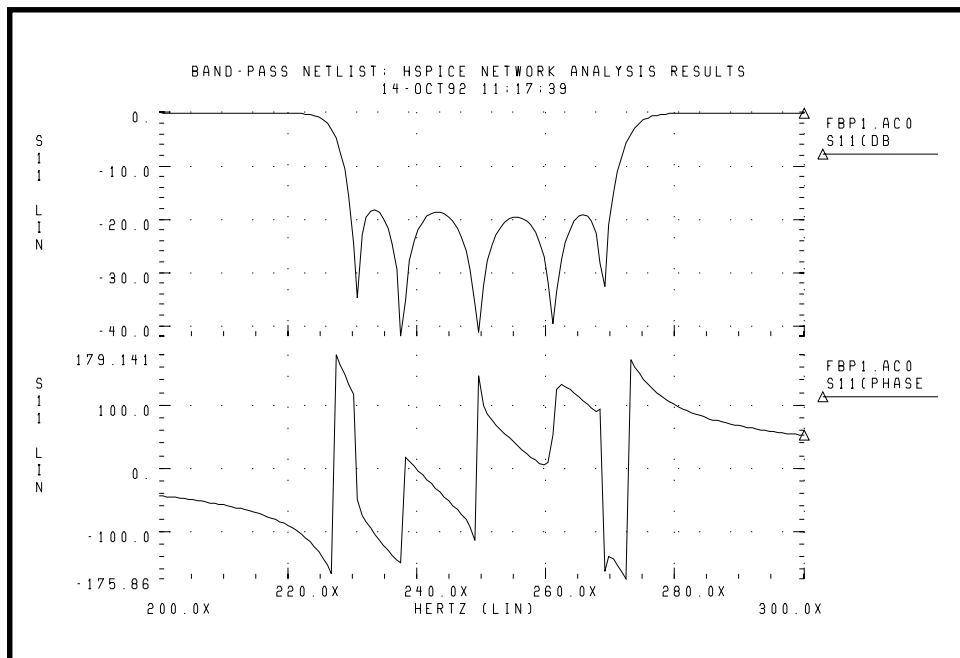
**Figure 9-2: S11 Magnitude and Phase Plots**
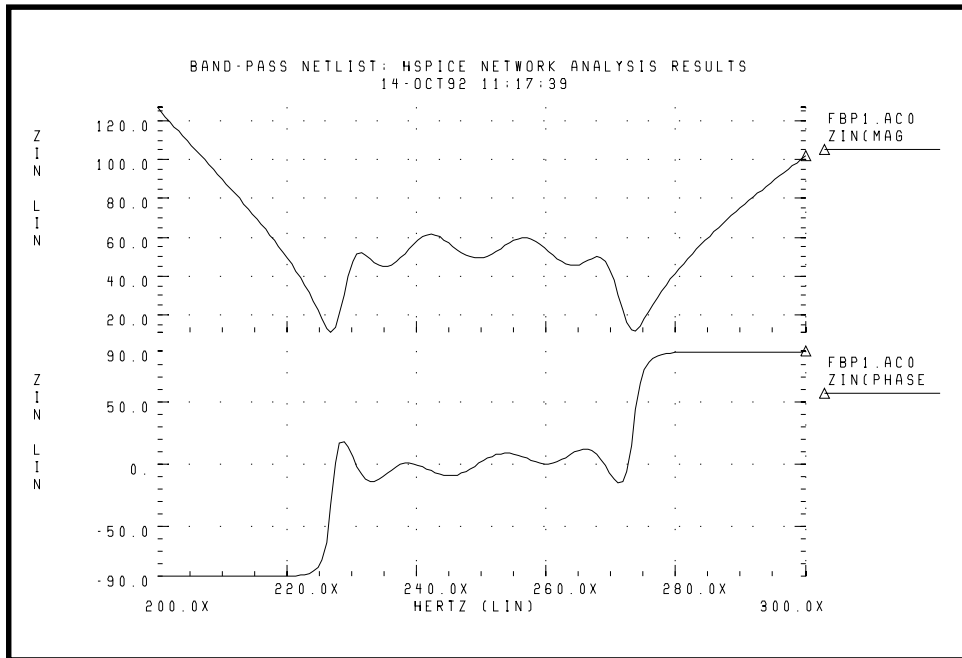
**Figure 9-3: ZIN Magnitude and Phase Plots**

## 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 9-4: shows the importance of the variables used in the
.NET statement. Here, *Isrc* and *Vce* are the DC biases applied to the BJT.
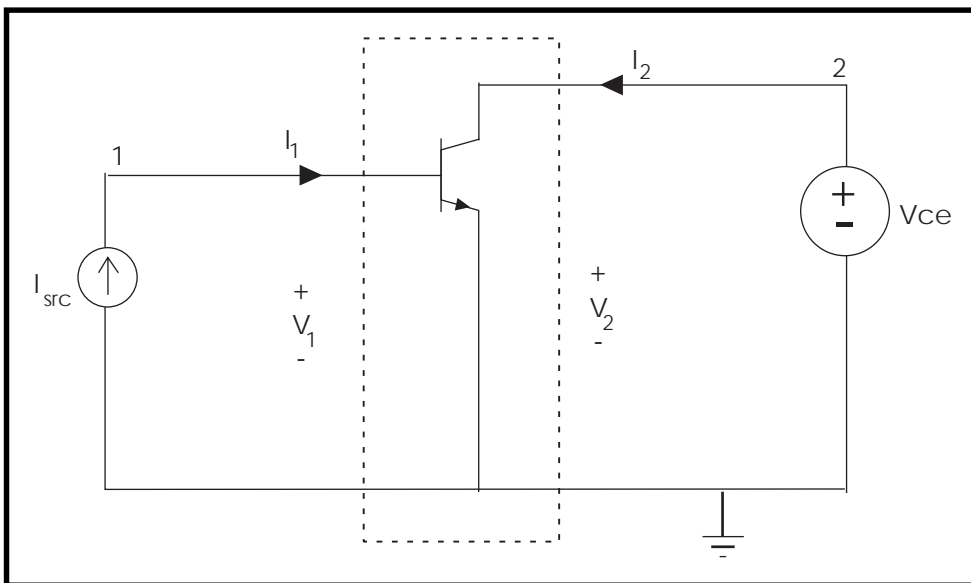


**Figure 9-4: Parameters with .NET V(2) Isrc**

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 9-5:.



Z -parameter: .NET V(C) $I_B$          Y-parameter: .NET I(Vc) $V_{BE}$

H-parameter: .NET I(Vc) $I_B$          S-parameter: .NET V(C) $V_{BE}$
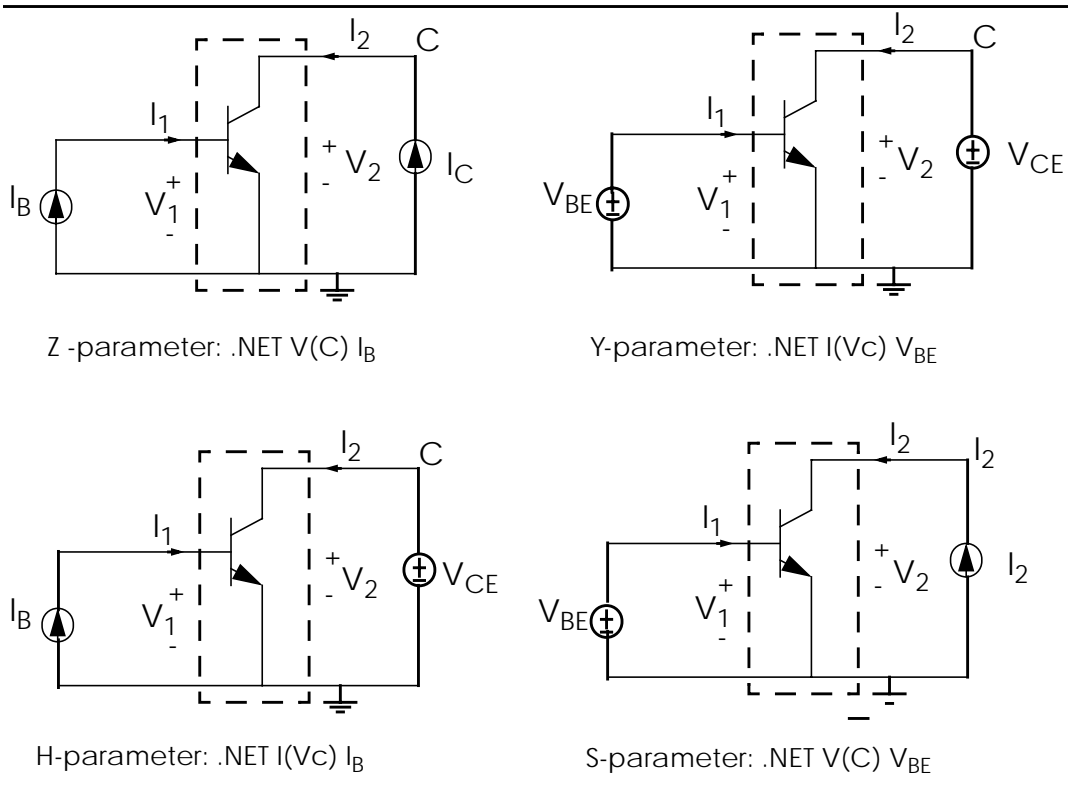
**Figure 9-5: Network Parameter Configurations**

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 10

# Analyzing Electrical Yields

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 the yield by using statistical techniques and observing the effects of element and model parameter variation.

- Specifying Analytical Model Types
- Using Algebraic Expressions
- Using Built-In Functions
- Using Parameters in Simulation
- Varying Parameters and Libraries
- Performing Worst Case Analysis
- Performing Monte Carlo Analysis
- Worst Case and Monte Carlo Sweep Example
- Simulating Circuit and Model Temperatures

# 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.

- 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 *Chapter 19, Performing Cell Characterization*. 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:

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

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

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

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

# Using Algebraic Expressions

You can replace any parameter defined in the netlist by an algebraic expression with quoted strings. Then, use these expressions as output variables in the .PLOT, .PRINT, and .GRAPH statements. Using algebraic expressions can expand your options in the input netlist file. Use algebraic expressions in the following ways:

■ Scaling or changing of element and model parameters

■ Parameterization:

```
.PARAM x=5
```

■ Algebra:

```
.PARAM x='y+3'
```

■ Functions:

```
.PARAM rho(leff,weff)='2+*leff*weff-2u'
```

■ Hierarchical subcircuit algebraic parameter passing

```
.subckt inv in out wp=10u wn=5u qbar_ic=vdd
.ic qbar=qbar_ic
...
.ends
```

■ 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)')
```

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).

### Algebraic Expressions for Output

The syntax is:

```
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.)

# Using Built-In Functions

In addition to simple arithmetic operations (+, -, *, /), Star-Hspice provides a number of built-in functions that you can use in expressions, for example. The Star-Hspice built-in functions are listed in Table 10-1:.

### Table 10-1:  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 sine of x (radians) |
| tanh(x) | hyperbolic tangent | trig | Returns the hyperbolic sine 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: $x^{(\text{integer part of } y)}$ |
| 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)$\log_{10}(\|x\|)$ |

## Table 10-1:  Star-Hspice Built-in Functions

| HSPICE Form | Function | Class | Description |
|---|---|---|---|
| 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" on page 4-35 for more information. |
| v(<Node>), i(<Element>).. . | circuit output variables | various | Returns various circuit values during simulation. See "Transient Sigma Sweep Results" on page 10-53 for more information. |

Star-Hspice reserves the variable names listed in Table 10-2: 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 10-2: 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)
```

# Using Parameters in Simulation

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.

Star-Hspice supports forward referencing of parameter names. That is, you can reference a parameter that you have not yet defined.

The Star-Hspice input:

```
.PARAM Third = Second
.PARAM Second = First
.PARAM First = 1
```

is equivalent to the following, in that all three parameters are set equal to 1:

```
.PARAM First = 1
.PARAM Second = First
.PARAM Third = Second
```

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.

You can use parameters to represent a variable value for any element parameter (such as the width of a transistor), or, in some cases, simulation control parameters (such as the duration of a transient simulation).

```
.PARAM VddP= 3.3v$ Define the parameter values
.PARAM VssP= 0.0v
Vdd Vdd 0 VddP$ Define supplies
Vss Vss 0 VssP
.PARAM StepSize= 1ns
```

```
.PARAM Steps= 10
.TRAN Step=StepSize Start=0ns Stop='StepSize*Steps'
```

All parameter values in Star-Hspice are IEEE double floating point numbers.

*Note:   The statements above show the proper way to define supplies that will be changed using an .ALTER statement. ALTER does not support redefinition of voltage sources—only their parameterized values.*

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 10-3:.

### Table 10-3: – 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 Definition

Parameters in Star-Hspice are names that have associated numeric values. You can use any of the following methods to define parameters:

- Simple parameter            .PARAM <SimpleParam> = 1e-12

- Algebraic defintition       .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         PARAM <mcVar> = Agauss(1.0,0.1)
  function

- .MEASURE                    MEASURE <MeasType> <MeasParam>
  statement                   + <MeasureClause>

### Simple Parameter Value Assignment

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>
```

### Example

```
.PARAM TermValue = 1g
rTerm Bit0 0 TermValue
rTerm Bit1 0 TermValue
rTerm Bit2 0 TermValue
 ...
```

## Algebraic Parameter Definition

Algebraic parameter assignments are 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 in order to invoke the Star-Hspice algebraic processor. A simple expression consists of a single parameter name.

### Syntax

```
.PARAM <ParamName> = '<Expression>'     $ Quotes are mandatory
```

or

```
.PARAM <ParamName1> = <ParamName2>$ Cannot be recursive!
```

### Examples

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

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

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

The syntax is:

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

### Examples

```
.PARAM SqrtOf2       = 'Sqrt(2)'
.PARAM Pi            = '355/113'
.PARAM Pi2           = '2*Pi'
```

Define circuit values defined by a direct algebraic evaluation:

```
r1 n1 0 R='1k/sqrt(HERTZ)' $ Resistance related to frequency.
```

## User-Defined Function Parameters

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>'
```

### 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.

### Syntax

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

where *<SubDefaultsList>* is

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

### Subcircuit Default Definitions Examples

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, the user 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
...
```

### Scoping and Default Overrides Example

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. This section describes the usage of these analysis types with respect to parameter definitions.

### Optimization

For Star-Hspice optimization, define parameters with the .PARAM statement. This statement allows you to specify the parameters to optimize, the optimization range, and an optional incremental value.

The syntax is:

```
.PARAM <ParamName> = <OptParamFunc> (<Init>, <LoLim>, <HiLim>,
+  <Inc>)
```

where:

*OptParmFunc*      Optimization parameter function (string)

*Init*             Initial value of <ParamName> (real)

*LoLim*            Lower limit for <ParamName> (real)

*HiLim*            Upper limit for <ParamName> (real)

*Inc*              Rounds to nearest <Inc> value (real)

### Example

```
.PARAM nWidth = WidOpt ( 5u, 1u, 10u, 0.1u )
...
.TRAN <TranTimeSpec> Sweep Optimize = WidOpt Results
+  = <MeasParamList>
...
```

### Monte Carlo Analysis

The .PARAM statement is also used to define parameters for Star-Hspice Monte Carlo analysis. See for more information.

The syntax is:

```
.PARAM <ParamName> = <MonteFunc> (<MonteParamList>)
```

where:

*MonteFunc*          name of the predefined Monte Carlo function (string). See
                     Table 10-4:.

*MonteParamList*     Monte Carlo function control parameter values (real)

### Table 10-4: Syntax for Parameter Lists for Monte Carlo Analysis Functions

| <MonteFunc> | Description | <MonteParamList> |
|---|---|---|
| UNIF() | Relative variation of a uniform distribution | nominal_val, rel_variation <, multiplier> |
| AUNIF() | Absolute variation of a uniform distribution | nominal_val, abs_variation <, multiplier> |
| GAUSS() | Relative variation of a Gaussian distribution | nominal_val, rel_variation, sigma <, multiplier> |
| AGAUSS() | Absolute variation of a Gaussian distribution | nominal_val, abs_variation, sigma <, multiplier> |
| LIMIT() | Random choice of limit values | nominal_val, abs_variation <, multiplier> |

### Example

```
.PARAM LithOffsetX=AGAUSS(0u,0.1u,3)
...
.TRAN <tran_time_spec> SWEEP MONTE=30
...
```

The statements above produce 30 randomly generated Gaussian-distributed values, centered on 0 $\mu$ and with a three-sigma spread of 0.1 $\mu$.

## 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.

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 hierarchial 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.

### Syntax

```
.MEASURE <MeasType> <ParamName> = <MeasClause>
```

where:

*MeasType*          type of measurement to be performed: AC, DC or TRAN

*MeasClause*        the style of measurement performed: TRIG/TARG, FIND/
                    WHEN, .PARAM, or others

### Example: Measurement Parameter Definition

```
.MEASURE TRAN tSlope Trig v(In) Val = 'Vdd * 0.1' Rise = 1
+ Targ v(In) Val = 'Vdd * 0.9' Rise = 1
...
.TRAN <TranTimeSpec>
...
```

The statements above measure the amount of time it takes for the voltage on the node *In* to change from 10% to 90% of Vdd.

### Example: Measurement Parameter Conflicts

For the input netlist fragment

```
.PARAM tSlope = 1ns
...
.MEASURE TRAN tSlope Trig v(In) Val = 'Vdd * 0.1' Rise = 1
+ Targ v(In) Val = 'Vdd * 0.9' Rise = 1
```

Star-Hspice produces the following error message:

```
**error** measure name tslope conflicts with existing
parameter definition, please rename measure statement
```

The measurement statement

```
.MEASURE TRAN tSlope Trig v(In) Val = 'Vdd * 0.1' Rise = 1
+ Targ v(In) Val = 'Vdd * 0.9' Rise = 1
...
.PARAM tSlope = 1ns
```

produces the following error message:

```
**error**  above line attempts to redefine tslope
```

## 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 HSPICE MOSFET models.

Consider the following example:

### Example 1

```
 ...
 * 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*.

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.

**Example 2**

```
.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 10-5: outlines and compares the three primary methods for configuring libraries to achieve required parameter checking in the case of default MOS transistor widths.

### Table 10-5: 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.<br><br>Single location for default values. | Cannot be used with releases of Star-Hspice prior to Release 95.1. |

## Table 10-5: Methods for Configuring Libraries

| Method | Parameter Location | Pros | Cons |
|--------|-------------------|------|------|
| 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.

The syntax is:

```
.OPTION PARHIER = < GLOBAL | LOCAL >
```

The default setting is GLOBAL. To use the same scoping rules that HSPICE used prior to Release 95.1, set PARHIER to GLOBAL.

Figure 10-1: 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.
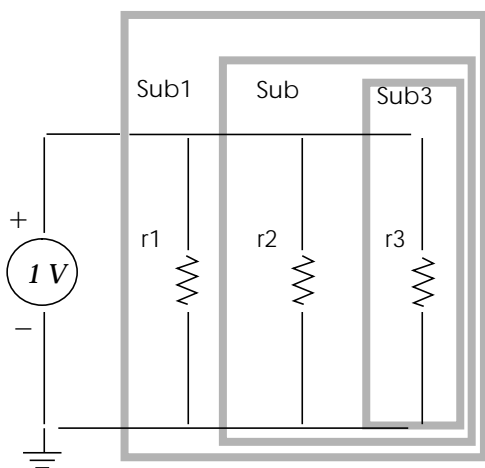
```
TEST OF PARHIER
.OPTION list node post=2
ingold=2
+parhier=<Local|Global>
.PARAM Val=1
x1 n0 0 Sub1
.SubCkt Sub1 n1 n2 Val=1
    r1 n1 n2 Val
    x2 n1 n2 Sub2
.Ends Sub1
.SubCkt Sub2 n1 n2 Val=2
    r2 n1 n2 Val
    x3 n1 n2 Sub3
.Ends Sub2
.SubCkt Sub3 n1 n2 Val=3
    r3 n1 n2 Val
.Ends Sub3
```

**Figure 10-1: Hierarchical Parameter Passing Problem**

There are two possible solutions for the total resistance of the chain: 0.3333 $\Omega$ and 0.5455 $\Omega$.

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 $\Omega$. Under local scoping rules, the lower level assignments prevail, and the total is 0.5455 $\Omega$ (one, two and three ohms in parallel).

The preceding example produces the results in Table 10-6:, based on the setting of the local/global PARHIER option:

**Table 10-6: PARHIER=LOCAL versus PARHIER=GLOBAL Results**

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

## Anticipating and Detecting 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.

### Solving Parameter Passing Problems

The easiest way to eliminate differences in your simulations between Release 95.1 and prior releases is to make sure global scoping rules are in effect (that is, make sure .OPTION PARHIER is *not* set to LOCAL). Global scoping rules are the default.

To set the default to GLOBAL for just a particular netlist, add the following line to the netlist:

```
.OPTIONS PARHIER = GLOBAL
```

To set the default to GLOBAL (which is the default if no PARHIER setting is specified) for all simulations, make the change in the Star-Hspice initialization file ($*installdir/hspice.ini*).

*Note:   This parameter is not recognized by releases prior to 95.1 and aborts the run if put into the hspice.ini file for previous releases.*

Remember, for Star-Hspice simulation, the value of the last definition of a parameter is the final definition of the parameter.

## Hierarchical Circuits, Parameters, and Models

This section describes hierarchical circuits, parameters, and models.

### Subcircuits

Reusability is an important factor in circuit simulation efficiency. Subcircuits and cell macros are reusable circuits. To expand the utility of a cell macro, parameters are also used. SPICE provides the basic subcircuit, but has no provisions for the consistent naming of nodes. Star-Hspice provides a simple method for naming subcircuit nodes and elements: simply prefix the node or element with the subcircuit call name.
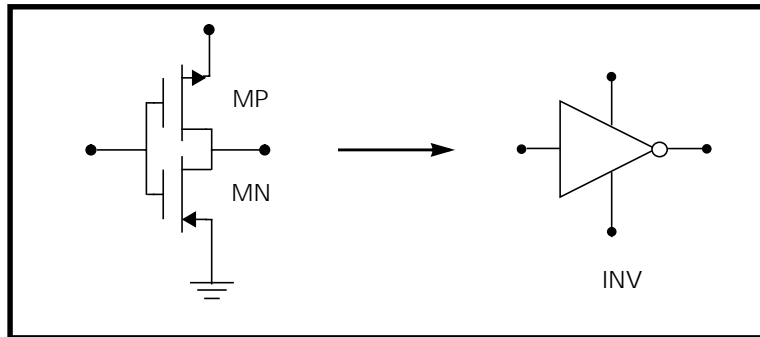
**Figure 10-2: Circuit-to-Subcircuit Representation**

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
```

The MOSFETs MP and MN inside the cell macro INV called by X1 are accessed using their hierarchical names X1.MP and X1.MN. For example, to print the current through the MOSFETs:

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

## The Multiply Parameter

The most basic subcircuit parameter is the M multiply parameter. This is a keyword common to all elements (except for voltage sources) and subcircuits. 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, 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.
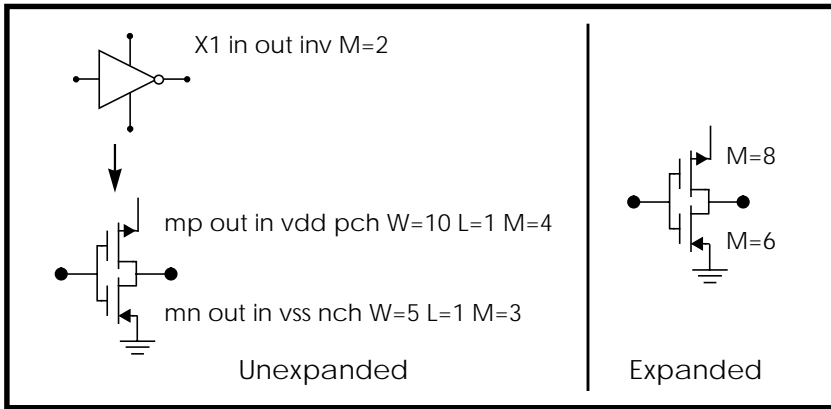
**Figure 10-3: Hierarchical Parameters Simplify Flip-Flop Initialization**

### 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
```
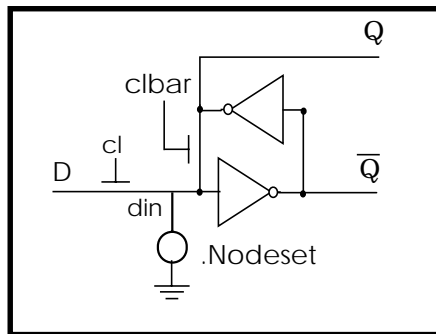


**Figure 10-4: D Latch with Nodeset**

**Parameters in 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.
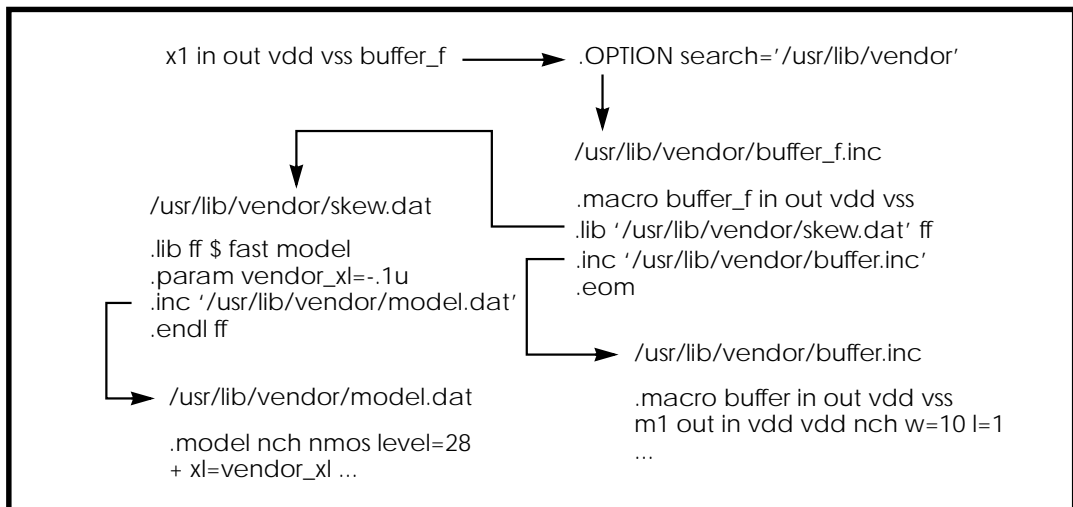


**Figure 10-5: Vendor Library Usage**

# Varying Parameters and Libraries

Use the .ALTER statement to run simulations of the same basic circuit but using different library parameters.

## Using the .ALTER Statement

Use the .ALTER statement to specify that a simulation be rerun using different:

- Circuit topologies
- Models
- Library components
- Elements
- Parameter values
- Options
- Test vectors
- Source stimulus (must parameterize)
- Analysis variables (must parameterize)
- Print and plot commands (must parameterize)

### 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.

# .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 |

# 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 10-7: shows whether deviations are added to or subtracted from the average.

### Table 10-7: 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 10-6:.
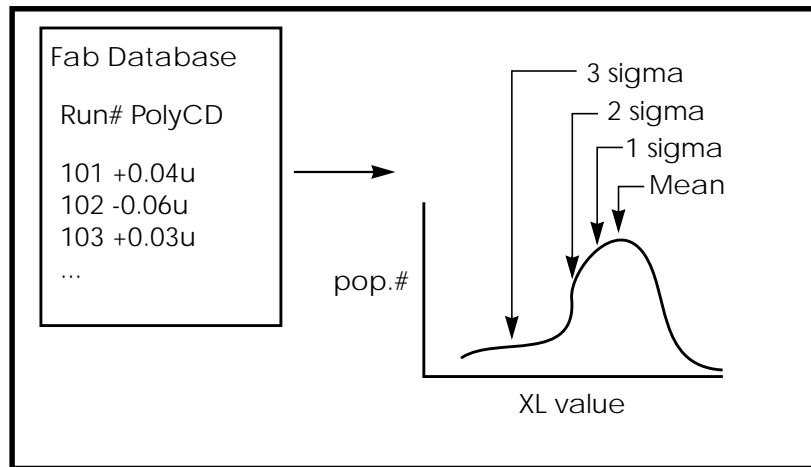
**Figure 10-6: Historical Records for Skew Parameters in a MOS Process**

## 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.
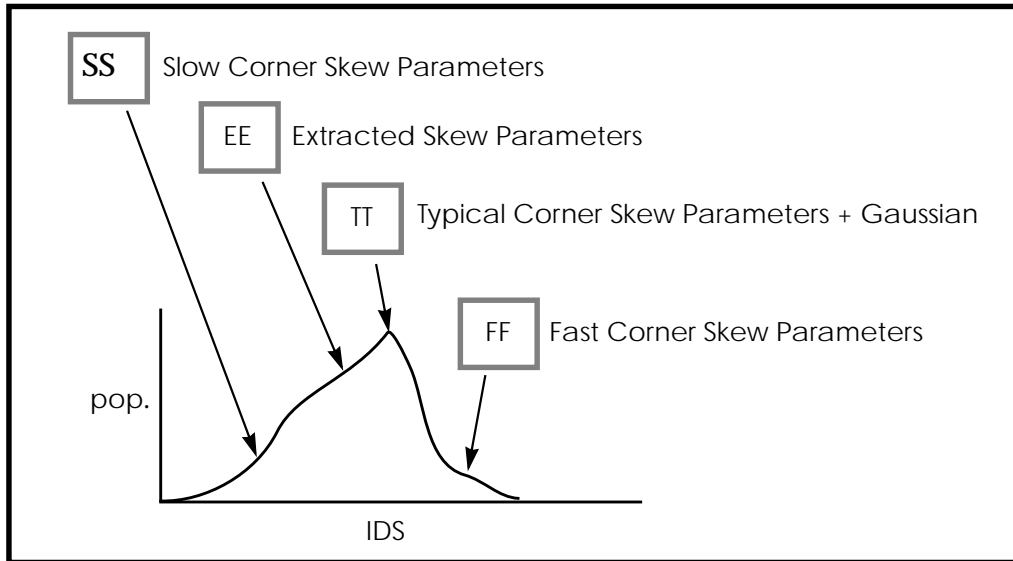
**Figure 10-7: Worst Case Corners Library File for a CMOS Process Model**

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 of Skew Parameter Overlay for a Model

```
.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 10-8: shows the importance of the four size levels.*
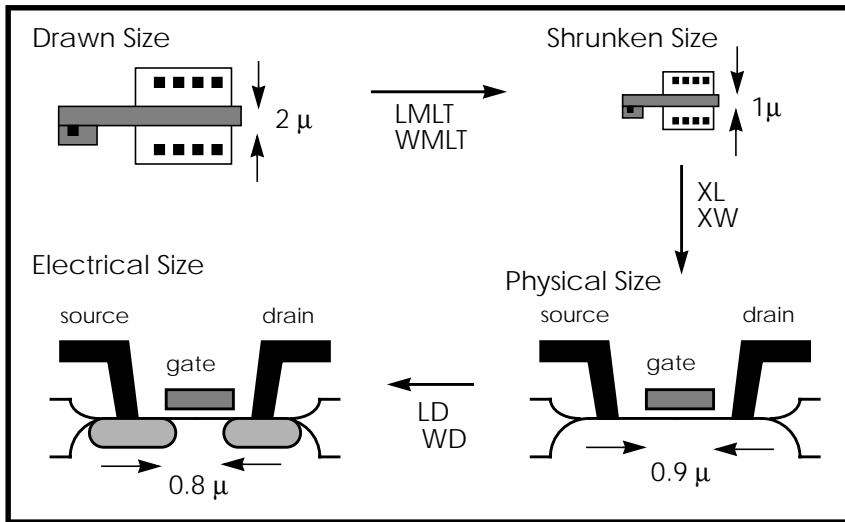


**Figure 10-8: Device Model from Drawn to Electrical Size**

# 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. |
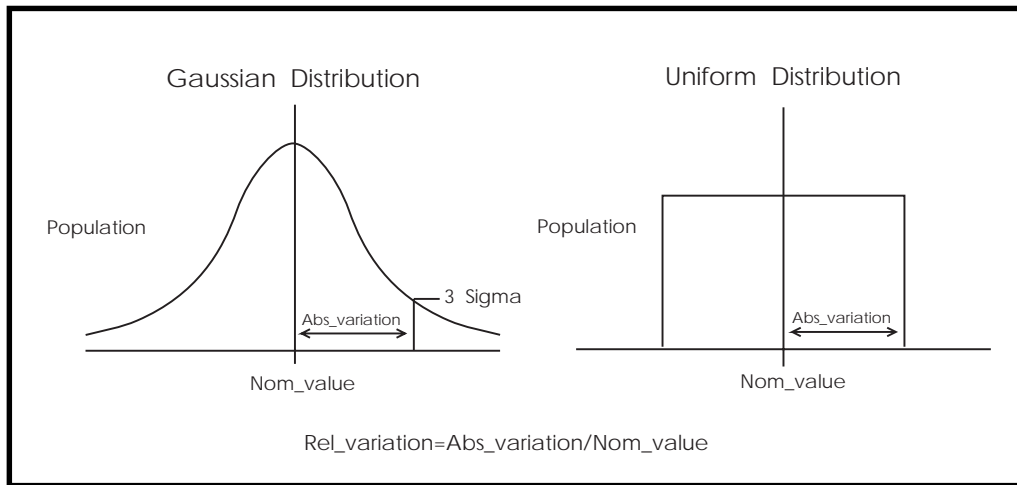


**Figure 10-9: Monte Carlo Distribution**

## 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

### Example 1: 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
```
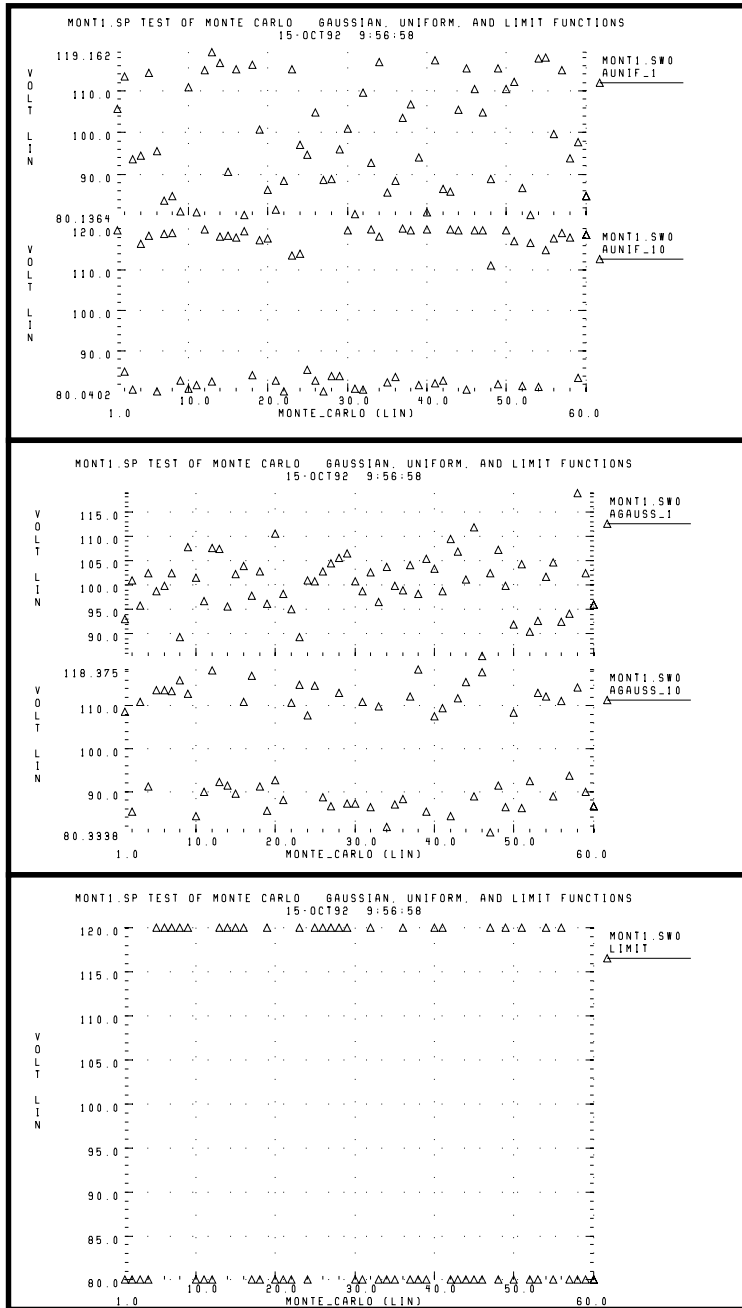
## Figure 10-10: Gaussian, Uniform, and Limit Functions

## Example 2: 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.
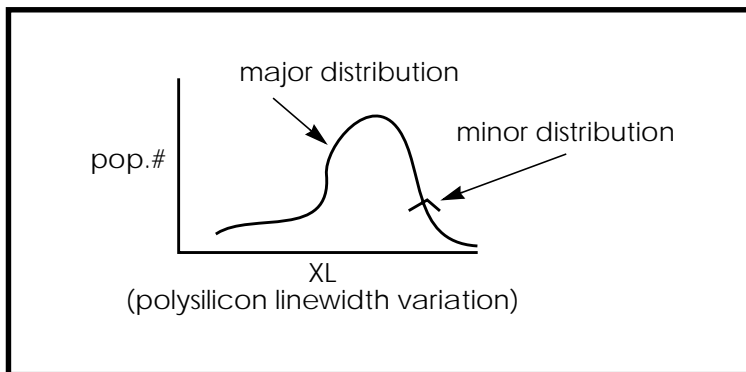


## Figure 10-11: Major and Minor Distribution of Manufacturing Variations

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
```

## Example 3: 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
```
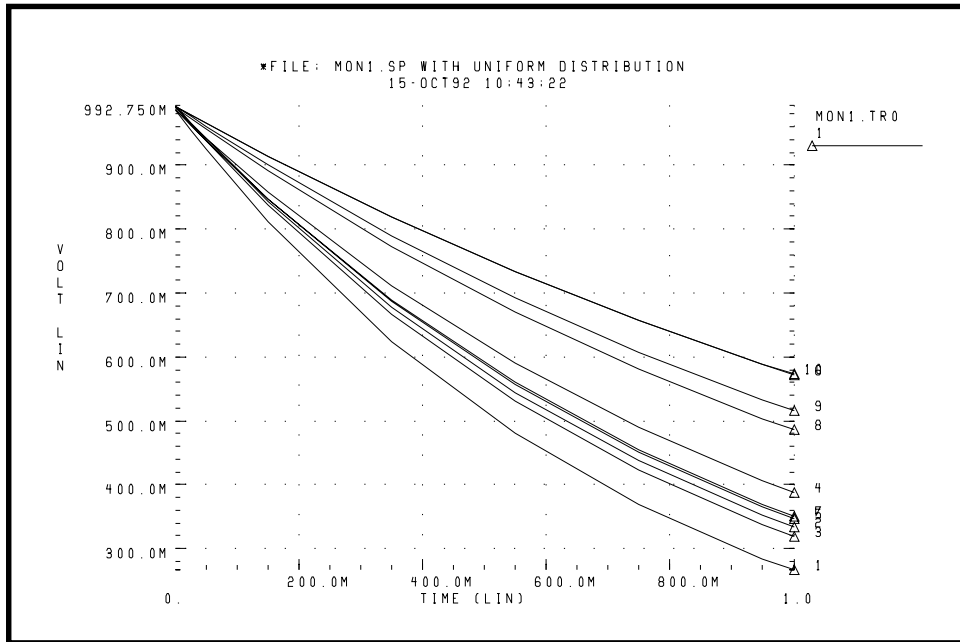
**Figure 10-12: Monte Carlo Analysis of RC Time Constant**

## Example 4: 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.
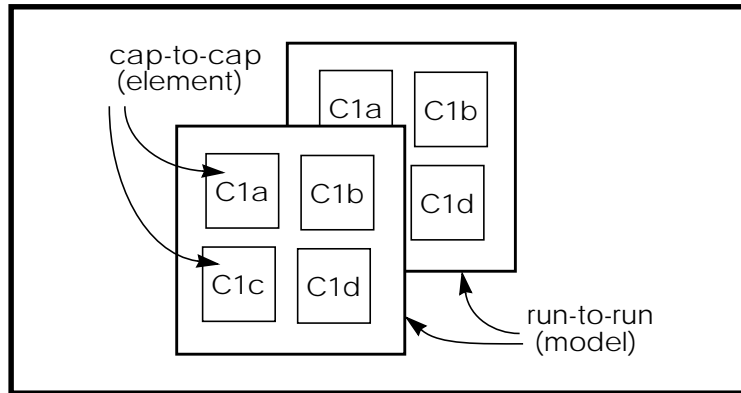
**Figure 10-13: Monte Carlo Distribution**

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

$$\text{Leff} = \text{Ldrawn} - 2 \cdot \text{DEL}$$

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

```
C1a 1 0 CMOD  W=ELPOLY  L=ELPOLY
C1b 1 0 CMOD  W=ELPOLY  L=ELPOLY
C1C 1 0 CMOD  W=ELPOLY  L=ELPOLY
C1D 1 0 CMOD  W=ELPOLY  L=ELPOLY

$ 10U 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.

```
C1a 1 0 CMOD  SCALE=ELCAP
C1b 1 0 CMOD  SCALE=ELCAP
C1C 1 0 CMOD  SCALE=ELCAP
C1D 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 10-14: 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.
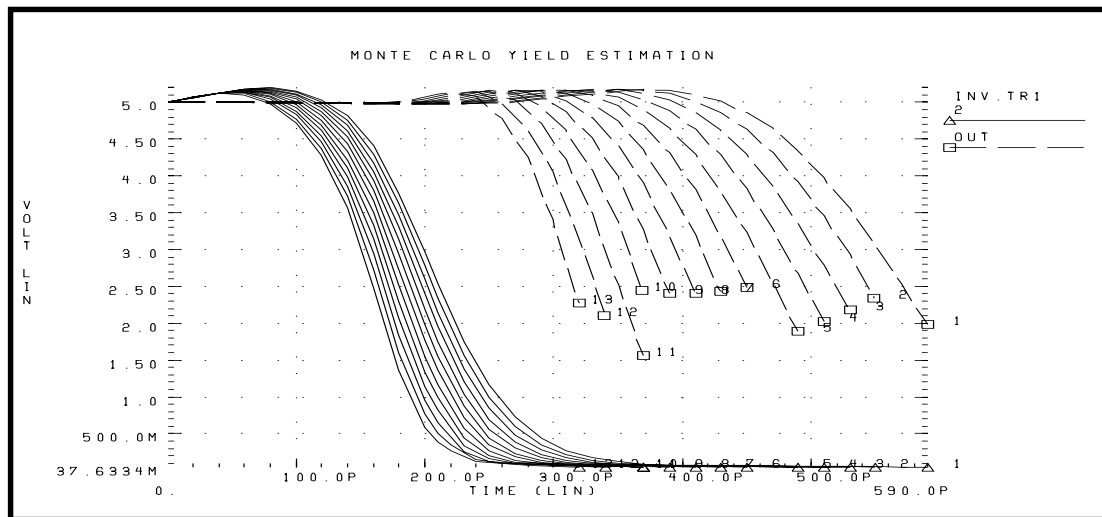
**Figure 10-14: Sweep of Skew Parameters from -3 Sigma to +3 Sigma**

You can view the transient family of curves by plotting the .MEASURE output
file. The plot in Figure 10-15: shows the measured pair delay and the total
dissipative power against the parameter SIGMA.
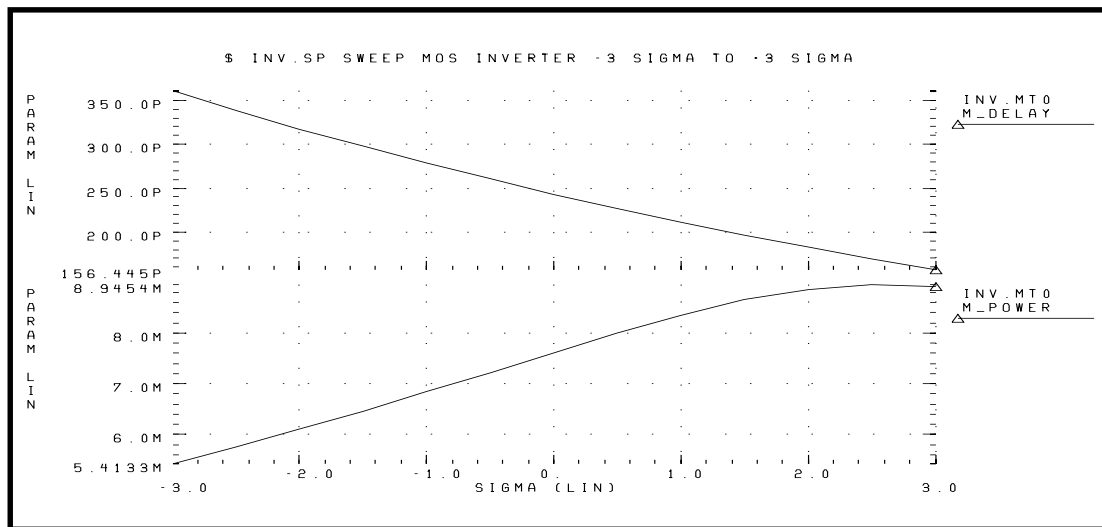
### Figure 10-15: Sweep MOS Inverter, Pair Delay and Power:
### -3 Sigma to 3 Sigma

## Monte Carlo Results

The output of the Monte Carlo analysis is evaluated in this section. The plot in Figure 10-16: 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.
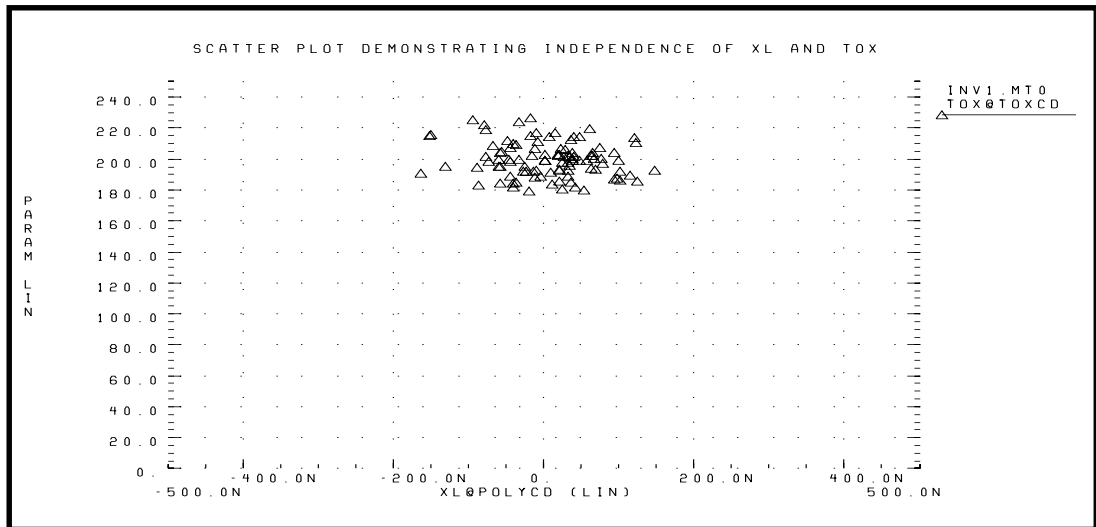


**Figure 10-16: Scatter Plot, XL and TOX**

The next graph (see Figure 10-17:) 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.
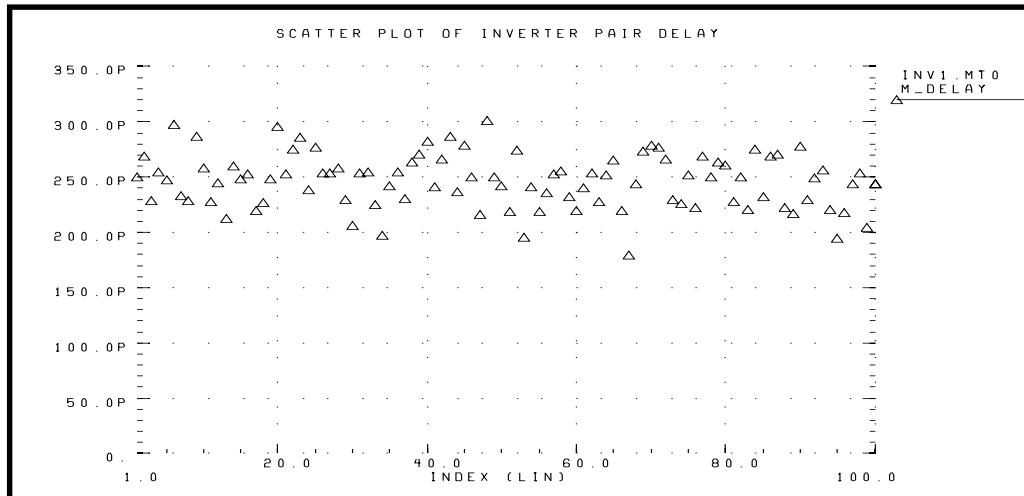


**Figure 10-17: Scatter Plot of Inverter Pair Delay**

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 10-18: demonstrates the expected sensitivity of output pair delay to channel length variation (polysilicon variation).
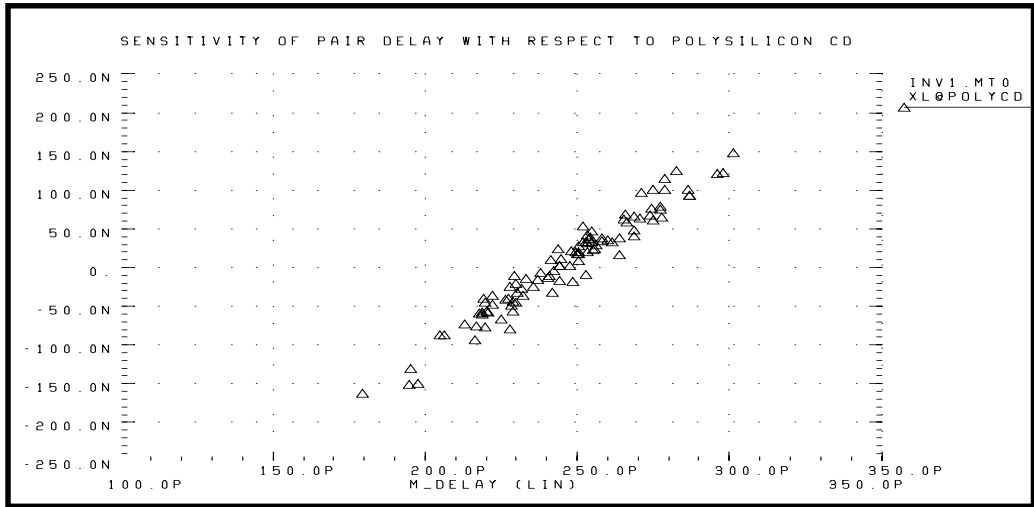
**Figure 10-18: Sensitivity of Delay with Poly CD (XL)**

Now, the parameter TOX is plotted against pair delay (Figure 10-19:). 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.
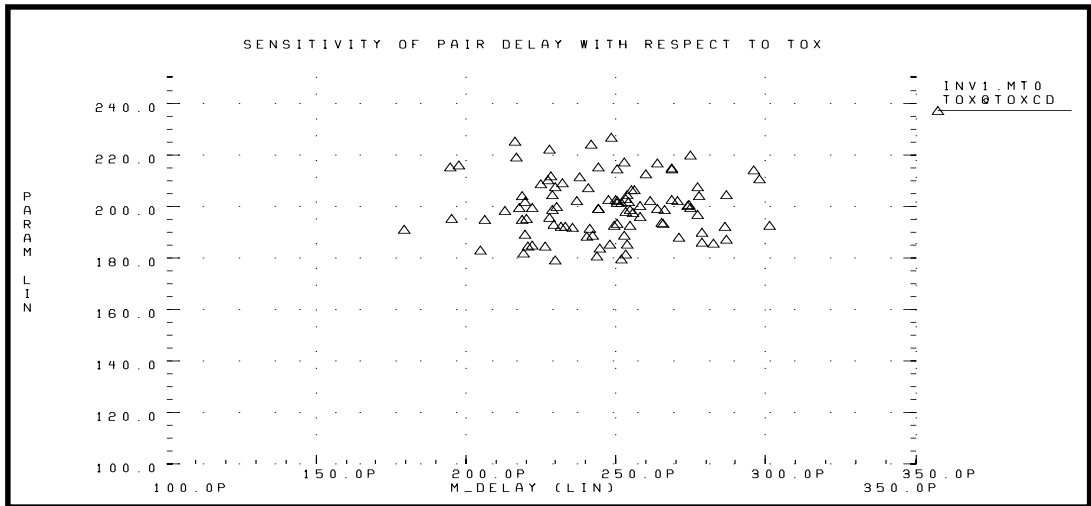


**Figure 10-19: Sensitivity of Delay with TOX**

The plot in Figure 10-20: 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.
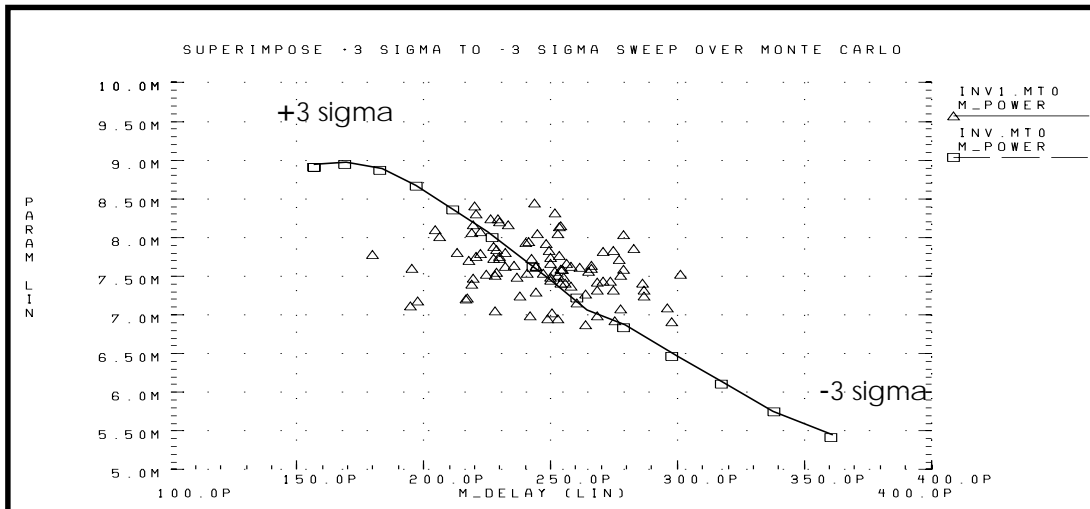


**Figure 10-20: Superimpose Sigma Sweep over Monte Carlo**

Now take the Monte Carlo plot and superimpose the assumed part grades from marketing studies (Figure 10-21:). 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.
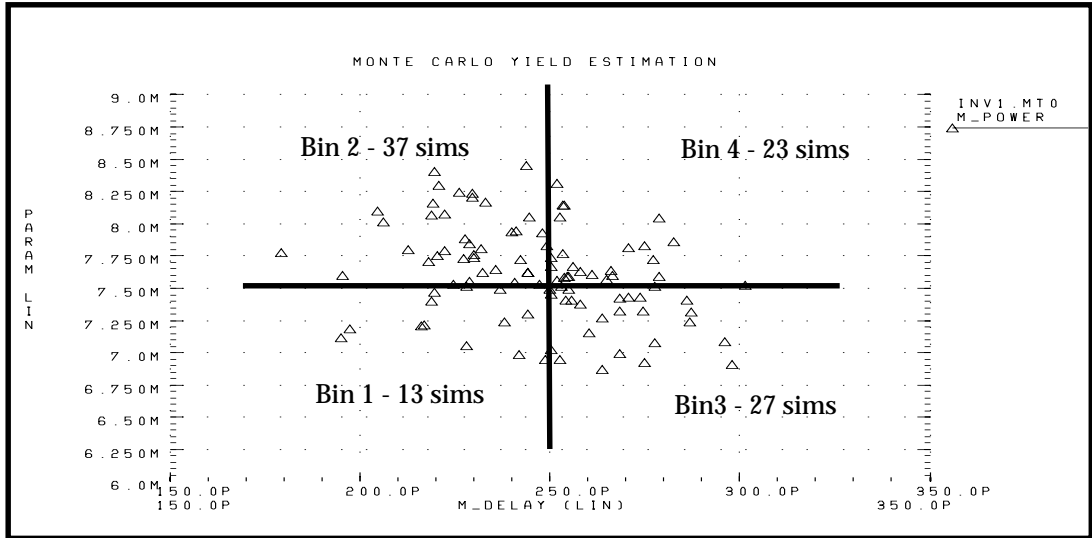


**Figure 10-21: Speed/Power Yield Estimation**

# Simulating Circuit and Model Temperatures

Temperature affects all electrical circuits. The key temperature parameters associated with circuit simulation are (see Figure 10-22:):

- 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.
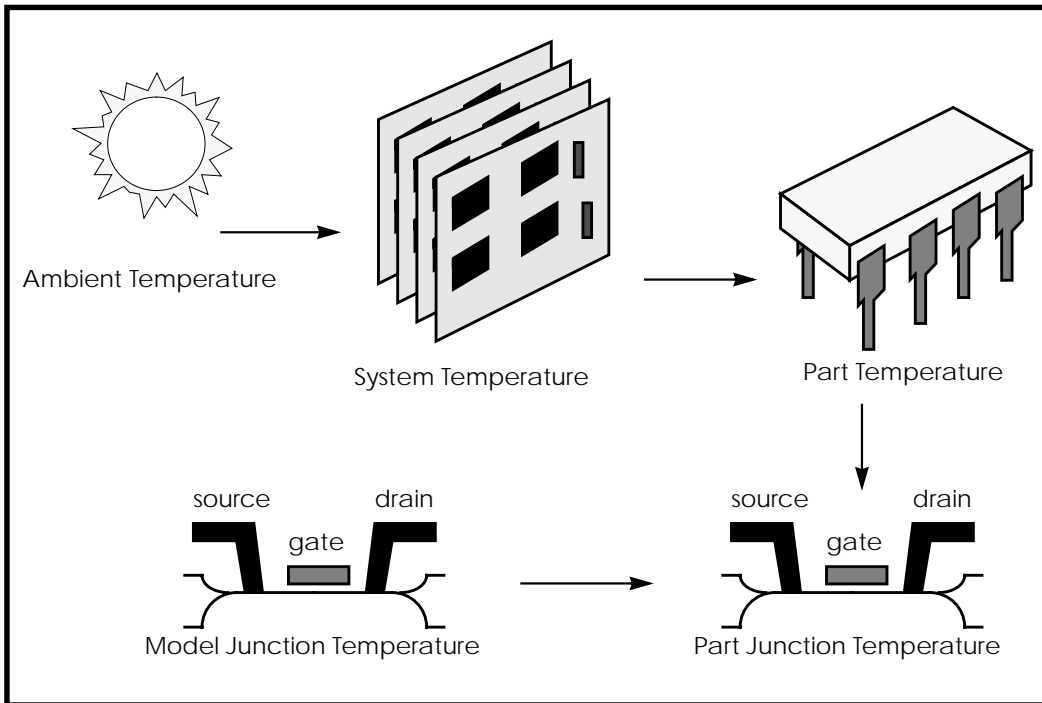
**Figure 10-22: Part Junction Temperature Sets System Performance**

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 I/O 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 1

```
.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 2**

```
.TEMP 100
D1 N1 N2 DMOD DTEMP=30
D2 NA NC DMOD
R1 NP NN 100 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).

*Avant!*

# Chapter 11

# Optimizing Performance

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.

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.

This chapter covers the following topics:

- Performing Optimization
- Understanding the Statements Syntax
- Examining Optimization Examples
- Performing Timing Analysis

# Performing 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.

Avant!'s Device Model Builder (DMB) product automates curve-fit
optimization for the development of most DC MOSFET models.

## 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.

# Understanding the Statements 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 *Chapter 4, Specifying Simulation Output*.

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 Syntax (.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 Syntax

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 $\pm$ ($n$·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 Syntax

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 $\Delta x$ = 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:<br>$\Delta par\_val = DIFSIZ \cdot MAX(par\_val, PARMIN)$ |

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. |

# Examining 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
.ENDDATA
.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 11-1:.
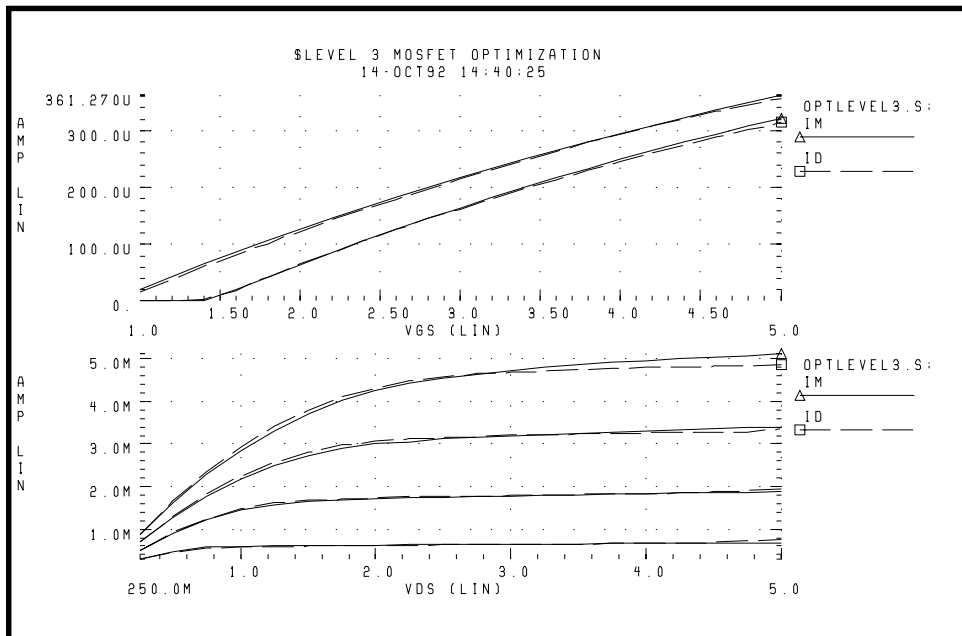
**Figure 11-1: Level 3 MOSFET Optimization**

## 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 11-2:.

## 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
```
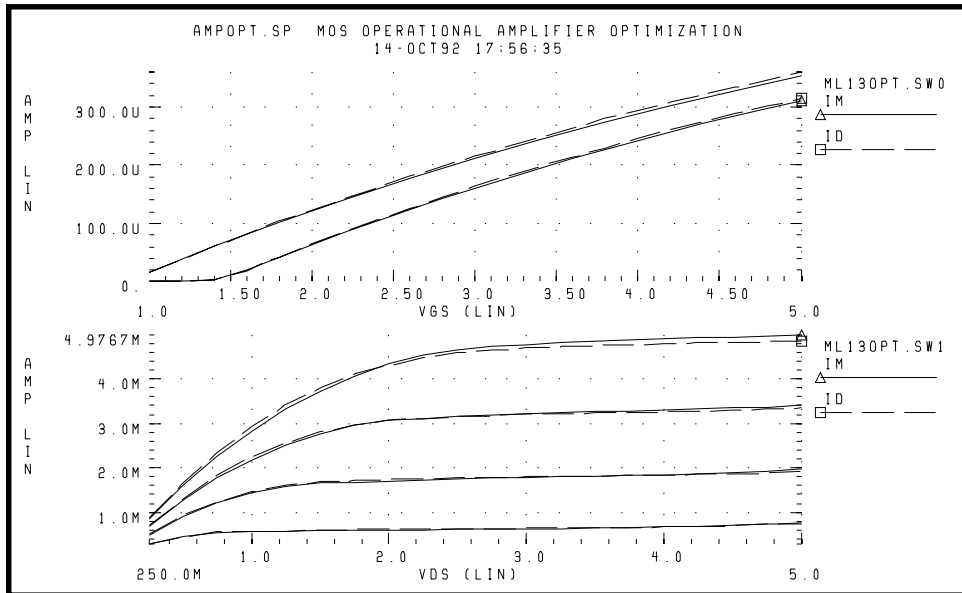
**Figure 11-2: Level 13 MOSFET Optimization**

## 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 Levenberg-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
```
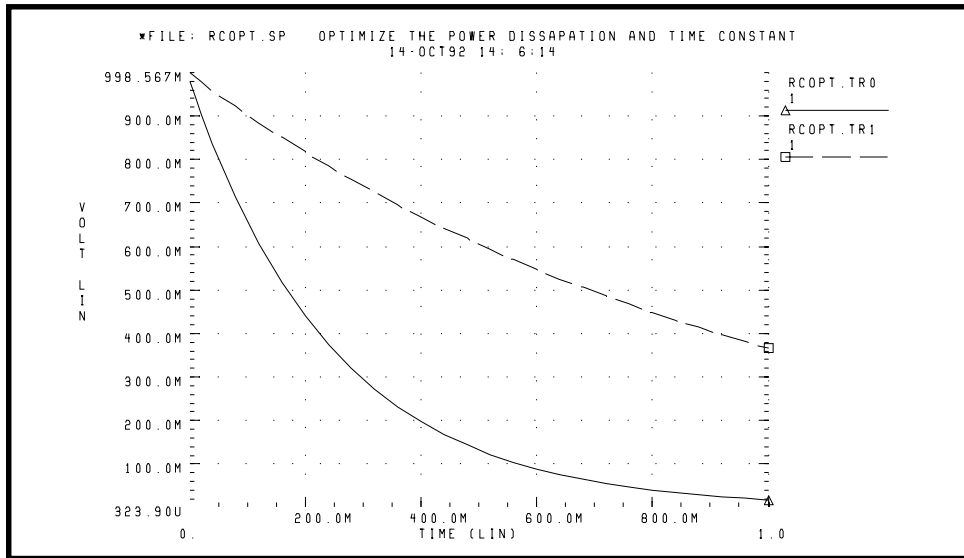
**Figure 11-3: Power Dissipation and Time Constant (VOLT)**
**RCOPT.TR0 = Before Optimization, RCOPT.TR1 = Optimized Result**
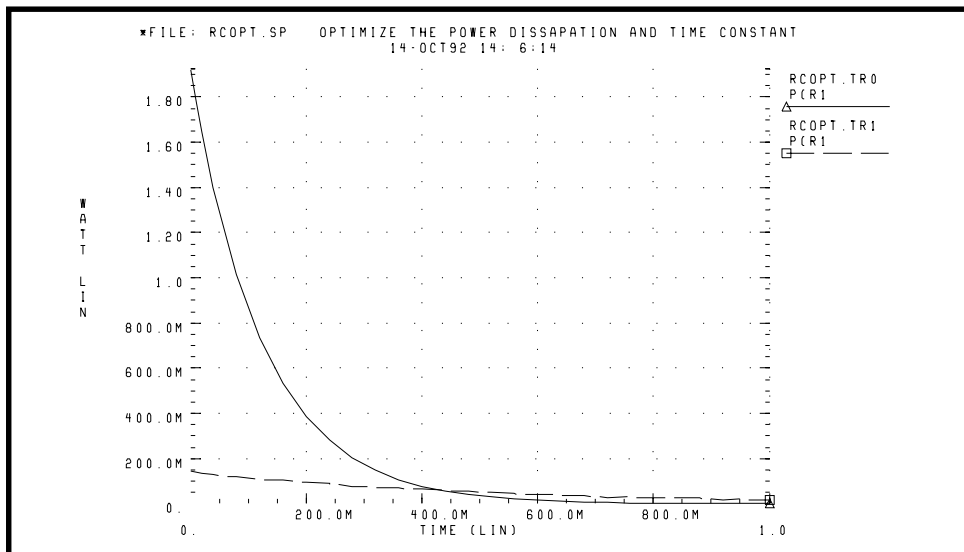


**Figure 11-4: Power Dissipation and Time Constant (WATT)**
**RCOPT.TR0 = Before Optimization, RCOPT.TR1 = Optimized Result**

## 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 I/O 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
```
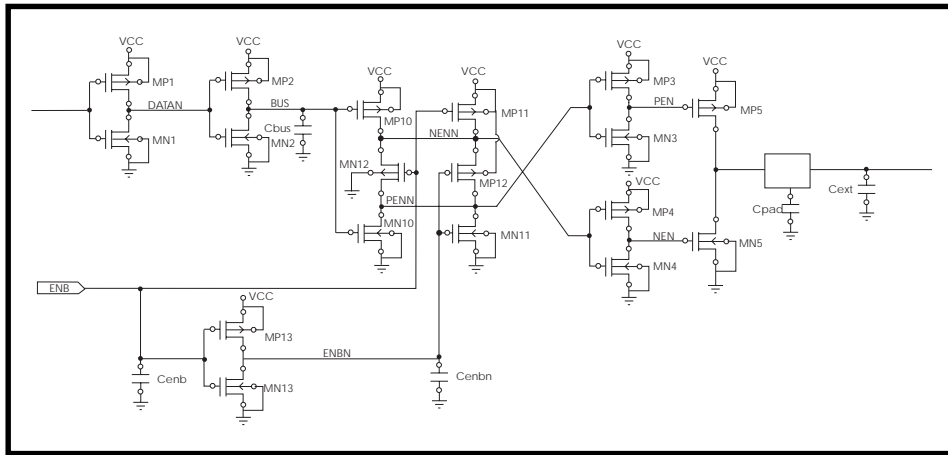
**Figure 11-5: Tristate Buffer Optimization Circuit**
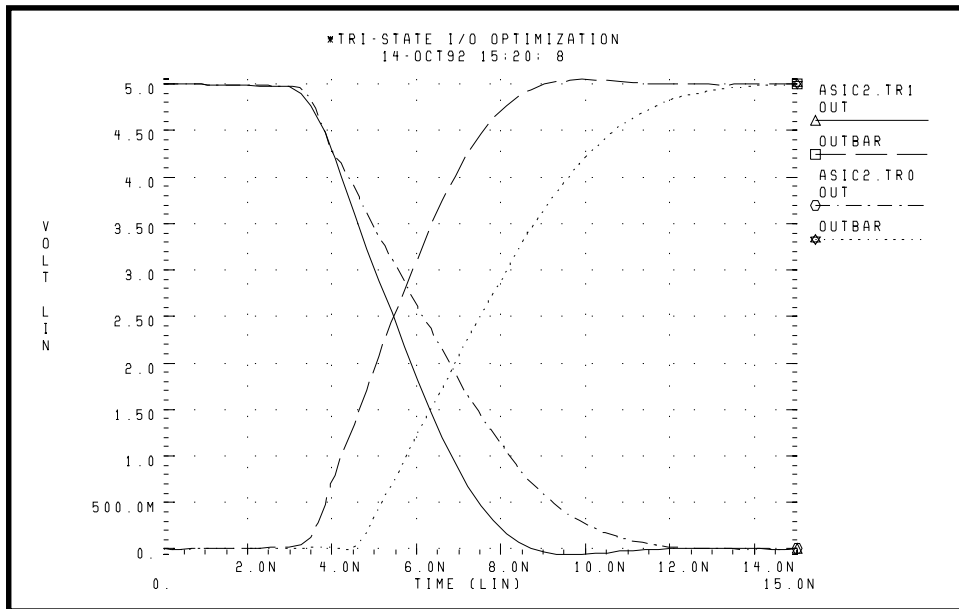


**Figure 11-6: Tristate I/O Optimization**
**ACIC2B.TR0 = Before Optimization, ACIC2B.TR1 = Optimized Result**

## 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
```
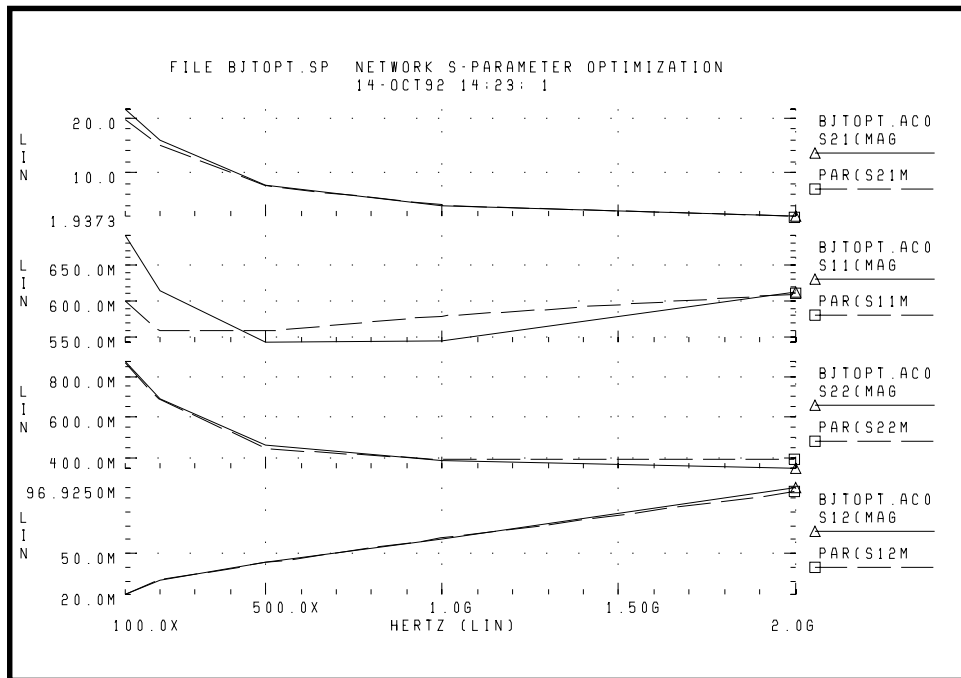
**Figure 11-7: BJT-S Parameter Optimization**

## 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
```
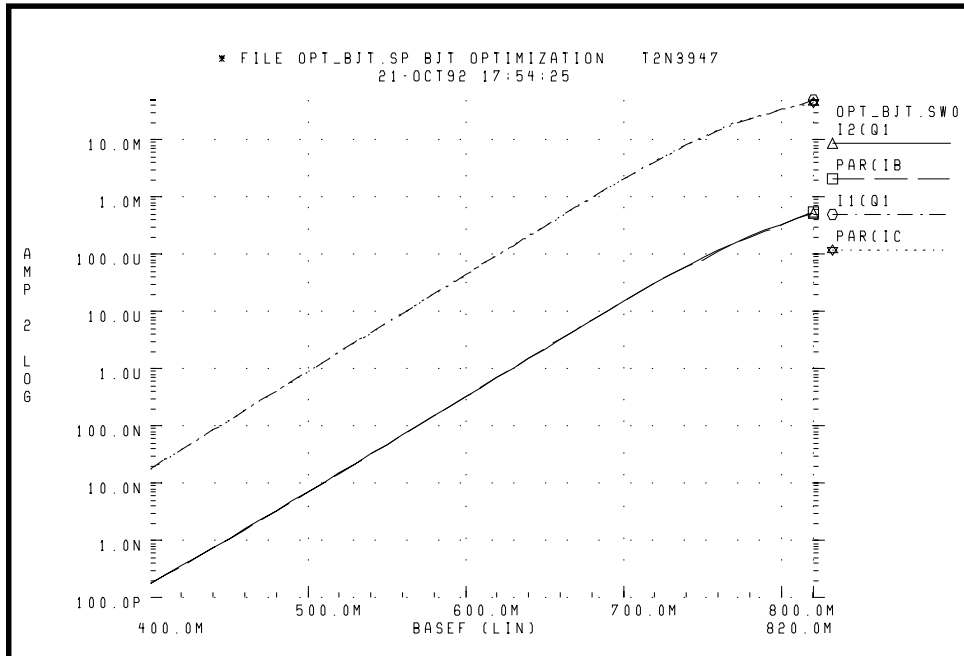


**Figure 11-8: BJT Optimization Forward Gummel Plots**
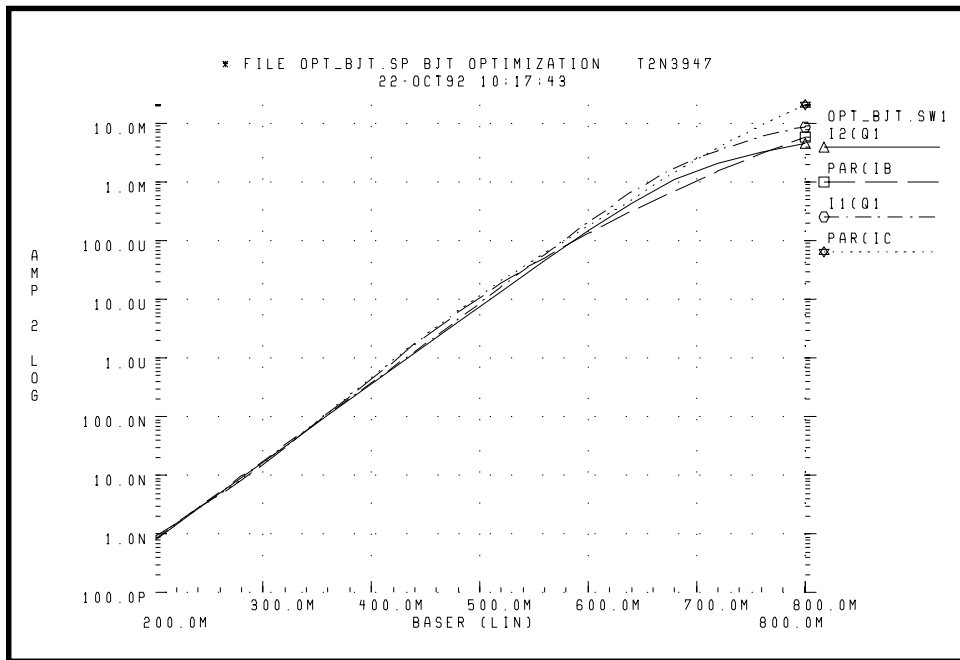
**Figure 11-9: BJT Optimization Reverse Gummel Plots**

## 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
```
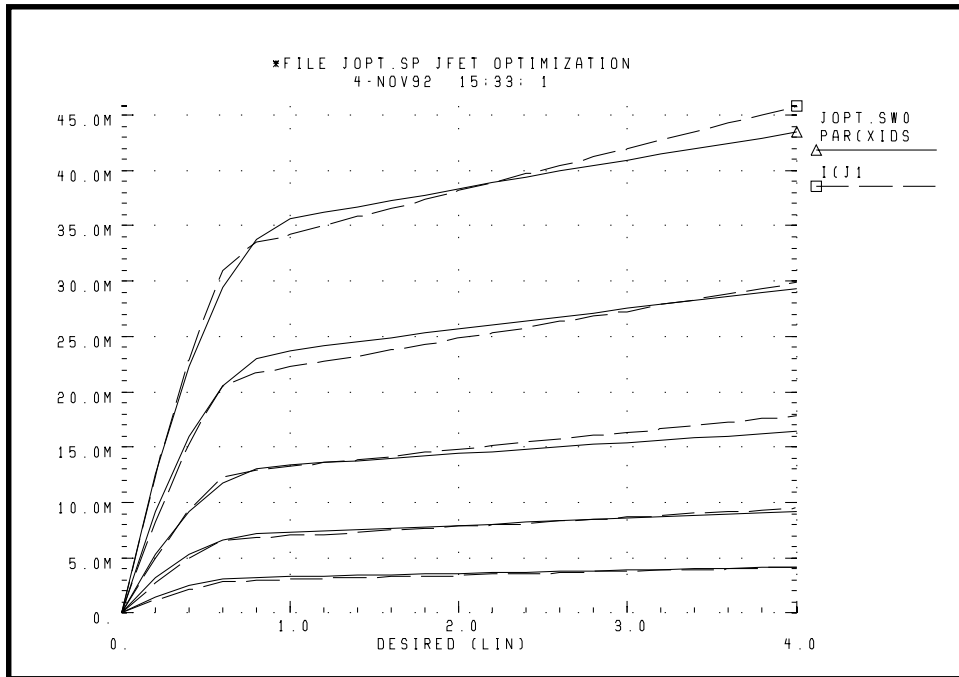


**Figure 11-10: JFET Optimization**

## 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
```
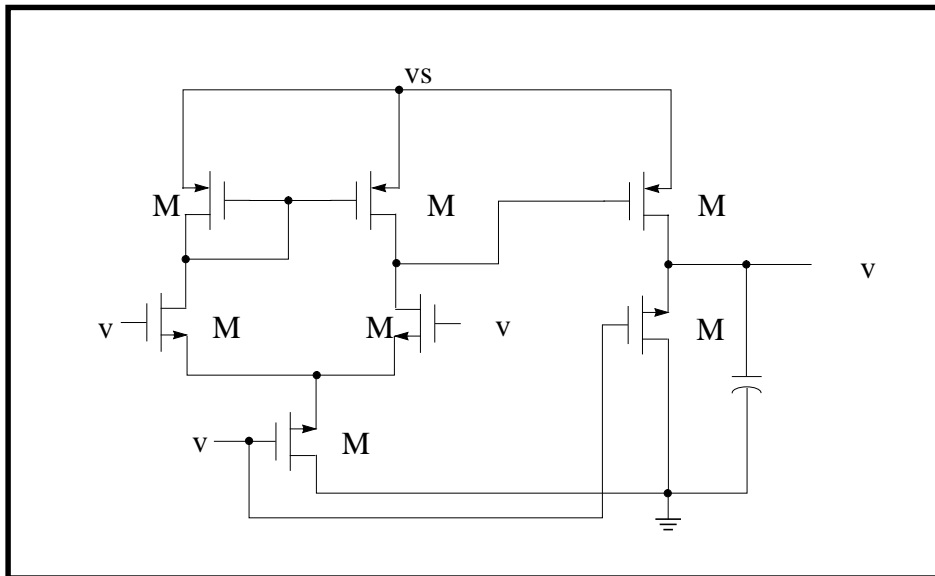


**Figure 11-11: CMOS Op-amp**
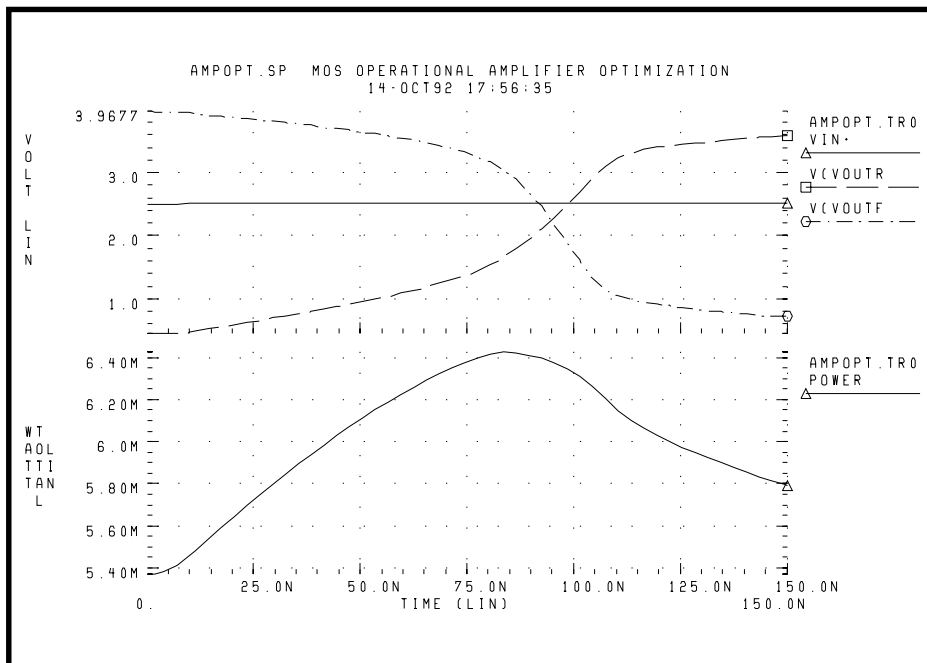
**Figure 11-12: Operational Amplifier Optimization**

# 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.

*Avant!*

# Chapter 12

# Using Passive Devices

This chapter describes element and model statements for passive devices. It includes statements for resistors, inductors, capacitors, and assorted magnetic elements and models.

Resistors, inductors, and capacitors are of two types:

- A simple, linear element with a value that depends on temperature, initialization, and scaling.
- An element that refers to a model statement

Use the set of passive elements and model statements to construct a wide range of board and integrated circuit level designs. Passive elements 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 covers the following topics:

- Using the Element Statement
- Using the Resistor Element
- Using the Capacitor Element
- Using the Linear Inductor Element
- Using Magnetics

# Using the Element Statement

The element statement specifies the type of element used. It has fields for the element name, the connecting nodes, a component value, and optional parameters.

**Syntax**

```
NAME node1, node2 …. nodeN <model reference>
value <optional parameters>
```

NAME                    Specifies the type and name of element. The first letter in the name field identifies the element as a specific type. For example, C=capacitor, L=inductor, or R=resistor. The remaining letters give the element a unique name.

node1 ... nodeN         Specifies how the element is connected in the netlist

value                   Gives the value of the element. For example, C1 2 0 10uF.

<model reference>       Refers to a model when the basic element value does not provide an adequate description

## Element Parameters

Element parameters within the element statement describe the device type, device terminal connections, associated model reference, element value, DC initialization voltage or current, element temperature, and parasitics.

The following tables include the complete set of parameters for all the available elements. The netlist field includes the element name (for example, Rxxx or Cxxx), the connecting nodes (for example, n1, n2), and any associated element value parameters (for example, rval in the resistor statement or cval in the capacitor statement).

## Table 12-1: R, L, C, and K Element Parameters

|  | **Resistor** | **Capacitor** | **Inductor** | **Mutual Inductor** |
|---|---|---|---|---|
| netlist | Rxxx, n1, n2 mname, rval | Cxxx, n1, n2, mname, cval | Lxxx, n1, n2 mname, lval | Kxxx, Lyyy, Lzzz, kval |
| temperature | DTEMP, TC1, TC2 | DTEMP, TC1, TC2 | DTEMP, TC1, TC2 | |
| geometric (scaling) | L, M, W, SCALE | L, M, W, SCALE | M, SCALE | |
| parasitics | C | | | |
| initialization | | IC(v) | IC(i) | |
| AC | AC | | | |

Table 12-2 shows two nonlinear elements. Use nonlinear elements to specify nonlinear inductors and capacitors in a polynomial equation. Specify either a nonlinear or linear element equation with the netlist parameters POLY, C0, and C1.

## Table 12-2: Nonlinear L and C Element Parameters

|  | **Nonlinear Inductor** | **Nonlinear Capacitor** |
|---|---|---|
| netlist | Lxxx, n1, n2, POLY, C0, C1 | Cxxx, n1, n2, POLY, C0, C1 |
| geometric | M | M |
| initialization | IC(i) | IC(v) |

Geometric parameters describe the dimensions of the element.

DTEMP is an element temperature parameter. It represents the difference between the general circuit simulation temperature and the element temperature. Element temperature = circuit temperature + DTEMP.

The magnetic winding element has a secondary resistance parasitic parameter (R) to model secondary parasitics.

The element statement provides initialization parameters for capacitor, inductor, magnetic winding, and transmission line elements to initialize circuits for DC analysis.

## Table 12-3: Magnetic and Transmission Line Element Parameters

| Function | Magnetic Winding | Mutual Coupling |
|---|---|---|
| netlist | Lxxx, n1, n2 | Kxxx, Lyyy, mname |
| geometric | NT | |
| parasitics | R | |
| initialization | IC(i) | MAG |

## Model Statement

Capacitor, resistor, and mutual coupling element statements have associated model capacitor, wire, and magnetic core model statements. The capacitor, wire, and magnetic core model statements use the following syntax:

### General form

```
.MODEL mname modeltype <keyword=value>
```

| mname | Refers to the model reference name specified in the associated element statement |
|---|---|
| modeltype | Specifies the model type: R for wire model, C for capacitor model, L for magnetic core model |

## Model Parameters

The model statement for each element has associated parameters to specify temperature, geometric dimensions, primary and parasitic resistance, and capacitive and magnetic values. You can set both resistance and capacitor values in the wire model, to model interconnect.

## Table 12-4: Passive Model Parameters

| Function | Wire Model | Capacitor | Saturable Core |
|----------|------------|-----------|----------------|
| Temperature | TC1C, TC1R, TC2C, TC2R, TREF | TC1, TC2, TREF | |
| Geometric | DLR, DW, L, SHRINK, THICK, W | DEL, L, SHRINK, THICK, W | AC, LC, LG, TC |
| Resistance | RAC, RES, RSH | | |
| Capacitance | BULK, CAP, CAPSW, COX, DI | CAP, CAPSW, COX, DI | |
| Magnetic | | | BS, BR, HC, HCR, HS |

# Using the Resistor Element

The resistor element uses the following element statement formats:

### Format

The general format of the resistor element is:
```
Rxxx n1 n2 <mname> rval <TC1 <TC2>> <SCALE=val>
<M=val> <AC=val>
+ <DTEMP=val> <L=val> <W=val> <C=val>

Rxxx n1 n2 <mname> R=val <TC1=val> <TC2=val >
<SCALE=val> <M=val> <AC=val>
 + <DTEMP=val> <L=val> <W=val> <C=val>
Rxxx n1 n2 R='equation'
```

If you specify mname, the resistor value specification is optional.

| | | |
|---|---|---|
| AC | AC resistance used in the AC analysis. Default=Reff. | |
| | | $ACeff = AC \cdot SCALE$ (element) / M |
| C | Capacitance connected from node n2 to bulk. Default=0.0, if C is not specified in the model. | |
| | | $Ceff = C \cdot SCALE$ (element) $\cdot M$ |
| DTEMP | Temperature difference between the element and the circuit. Default=0.0. | |
| *L* | Resistor length. Default = 0.0, if L is not specified in the model. | |
| | | $Lscaled = L \cdot SHRINK \cdot SCALE$ (option) |
| | SHRINK is a model parameter that specifies the fabrication shrink factor. | |

| | |
|---|---|
| *M* | Multiplier that simulates 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. |
| mname | Model name. Use this name in elements to reference the model. |
| n1 | Positive terminal node name |
| n2 | Negative terminal node name |
| R=equation | Equation that descibes the resistor value as a function of any node voltages, branch currents, and any independent variables such as time, frequency (Hertz), or temperature. |
| R=val | Resistance value |

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

| | |
|---|---|
| Rxxx | Resistor element name. Must begin with "R", which can be followed by up to 15 alphanumeric characters. |
| rval | Resistor value. Can be a value or parameter that can be evaluated. |
| *SCALE* | Element scale factor for resistance and capacitance. Default=1.0. |
| TC1 | First order temperature coefficient for resistor |
| TC2 | Second order temperature coefficient for resistor |
| W | Resistor width. Default=0.0, if W is not specified in the model. SHRINK is a model parameter. |

$$Wscaled = W \cdot SHRINK \cdot SCALE \text{ (option)}$$

### Examples

```
R1 Rnode1 Rnode2 100
RC1 12 17 1K TC=0.001, 0  1.2
R4 33 0 45 RTC1 RTC2  7
Rxxx  98999999 87654321  1 AC=1e10
```

### 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: $OT = RX^2 \cdot inr$

**Resistor Temperature Equations**

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

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

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

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

| | |
|---|---|
| $\Delta t$ | t - tnom |
| $t$ | Element temperature in °K: t = circuit temp + DTEMP + 273.15 |
| $tnom$ | Nominal temperature in °K: tnom = 273.15 + TNOM |

# Wire RC Model

The 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.

**Format**

The format of the wire element is:

```
.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

        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 is specified for CRATIO, Hspice displays a
                                  warning, sets CRATIO to 0.5, and continues the
                                  analysis.
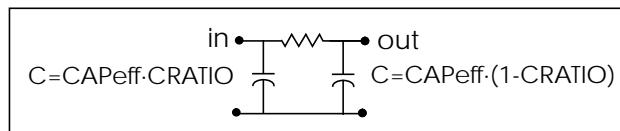


A resistor referred to as a wire model behaves like an elementary transmission
line if you specify an optional capacitor from node n2 to a bulk or ground node
in the model statement. The bulk node functions as a ground plane for the wire
capacitance.

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, the model name is taken. 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<br>DLReff=DLR · SCALM |
| DW | m | 0 | difference between drawn width and actual width<br>DWeff=DW · SCALM |
| L | m | 0 | default length of wire<br>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 |
| THICK | m | 0 | dielectric thickness |
| TREF | deg C | TNOM | temperature reference for model parameters |

| Name(Alias) | Units | Default | Description |
|-------------|-------|---------|-------------|
| W | m | 0 | default width of wire<br>Wscaled=W · SHRINK · SCALM |

### 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 to RESMIN and a warning message is issued.

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

## 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.

$$CAPeff \ = \ M \cdot SCALE(element) \cdot [Leff \cdot Weff \cdot COX + 2 \cdot (Leff + Wef_{|}$$

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, that 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

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

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

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

3. If only the model capacitance CAP is specified, then:

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

If the capacitance is specified and the bulk node is not specified, then capacitance is not evaluated and a warning message is issued.

# Using the Capacitor Element

General form
```
Cxxx n1 n2 <mname> cval <TC1 <TC2>> <SCALE=val>
<IC=val> <M=val> <W=val>
+ <L=val> <DTEMP=val>
```

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

or
```
Cxxx n1 n2 C='equation' CTYPE = 0 or 1
```

If a model is chosen for the capacitor, then the specification of CAPVAL is optional.

| | |
|---|---|
| *Cxxx* | capacitor element name. Must begin with a "C", which can be followed by up to 15 alphanumeric characters. |
| *n1* | positive terminal node name |
| *n2* | negative terminal node name |
| *mname* | model name |
| *C* | capacitance in farads at room temperature |
| cval | capacitance value. Can be a value or parameter that can be evaluated. |
| *TC1* | first order temperature coefficient |

| | |
|---|---|
| *TC2* | second order temperature coefficient |
| *M* | multiplier used to simulate multiple parallel devices. Default=1.0. |
| *W* | capacitor width |

Wscaled = W · SHRINK · SCALE (option)

| | |
|---|---|
| *L* | capacitor length |

Lscaled = L · SHRINK · SCALE (option)

| | |
|---|---|
| *DTEMP* | element temperature difference with respect to circuit temperature. Default=0.0. |
| *C=equation* | The capacitor value can be described as a function of any node voltages, branch currents, and any independent variables such as time, frequency (Hertz), or temperature. |

| | |
|---|---|
| *CTYPE* | If capacitance C is a function of v(n1, n2), set CTYPE=0. If C is not a function of v(n1, n2), set CTYPE=1. The capacitance charge is calculated differently for the two types. CTYPE must be set properly to provide correct simulation results. C as a function of multiple variables is not recommended. Default=0. |
| *SCALE* | element scale factor |

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

| | |
|---|---|
| *IV* | initial voltage across capacitor in volts. This value is used as the DC operating point voltage. |

To avoid syntactic conflicts, if a capacitor model exists using the same name as a parameter for cval in the element statement, the model name is taken. In the following example, C1 assumes that CAPXX refers to the model and not the parameter.

```
.PARAMETER  CAPXX=1
C1  1  2  CAPXX
.MODEL CAPXX C CAP=1
```

## Capacitance Model

### General form

.MODEL mname C parameter=value

| | |
|---|---|
| *mname* | model name |
| *C* | specifies a capacitance model |
| *parameter* | any model parameter name |

## Capacitance Parameters

### Table 12-5: 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 <br> DELeff = DEL · SCALM |
| DI | | 0 | relative dielectric constant |
| L | m | 0 | default length of capacitor <br> 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 |
| TREF | deg C | TNOM | reference temperature |
| W | m | 0 | default width of capacitor <br> Wscaled = W · SHRINK · SCALM |

## Parameter Limit Checking

HSPICE writes a warning message to the output listing file if a capacitive element value exceeds 0.1 farad. This feature eases identification of elements with missing units or wrong values, particularly those in automatically produced netlists.

## Effective Capacitance Calculation

A model can be associated with a capacitor in 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.

$$\backslash Peff = M \cdot SCALE(element) \cdot$$
$$[Leff \cdot Weff \cdot COX + 2 \cdot (Leff + Weff) \cdot CAPS$$

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

$$Peff = CAP \cdot SCALE(element) \cdot$$

## Capacitance Temperature Equation

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

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

| | |
|---|---|
| $\Delta t$ | t - tnom |
| $t$ | element temperature in degrees Kelvin |
| | t=circuit temp + DTEMP + 273.15 |
| $tnom$ | nominal temperature in degrees Kelvin |
| | tnom=273.15 + TNOM |

# Polynomial Capacitor Elements

General form

Cxxx  n1 n2 POLY c0 c1 … <IC=v>

| | |
|---|---|
| Cxxx | capacitor element name. Must begin with a "C", which can be followed by up to 15 alphanumeric characters. |
| *n1, n2* | node names |
| *POLY* | keyword identifies capacitor as nonlinear polynomial. It is a reserved name. |
| c0 c1 … | coefficients of a polynomial describing the element value. The capacitance is described as a function of the voltage across the capacitor. The value is: |

$$\text{capacitance} = c0 + c1 \cdot v + c2 \cdot v…$$

*IC*                           initial voltage across capacitor in volts. If the input
                              netlist file contains a .IC statement, the initial
                              conditions in the .IC statement override initial
                              conditions specified in element statements.

# Using the Linear Inductor Element

General form
```
Lxxx n1 n2 lval <TC1 <TC2>> <SCALE=val> <IC=val>
<M=val> <DTEMP=val> <R=val>
```

or
```
Lxxx n1 n2 L=val <TC1=val > <TC2=val> <SCALE=val>
<IC=val> <M=val> <DTEMP=val> + <R=val>
```

or
```
Lxxx n1 n2 L='equation' LTYPE = 0 or 1 <R=val>
```

| | |
|---|---|
| *Lxxx* | inductor element name. The name must begin with an "L", followed by up to 15 alphanumeric characters. |
| *n1* | positive terminal node name |
| *n2* | negative terminal node name |
| *TC1* | first order temperature coefficient |
| *TC2* | second order temperature coefficient |
| *SCALE* | element scale factor. Default=1.0. |
| *IC* | initial current through the inductor in amperes |
| *L* | inductance in henries at room temperature |
| | $Leff = L \cdot SCALE \text{ (element)} / M$ |

| *M* | multiplier used to simulate multiple paralleled inductors. Default=1.0. |
|---|---|
| *DTEMP* | element and circuit temperature difference. Default=0.0 |
| *R* | resistance in ohms of the inductor element |
| *equation* | The inductor value can be described as a function of any node voltages, branch currents, and any independent variables such as TIME, frequency (HERTZ), or temperature (TEMPER). The type of variable L depends upon is indicated by the parameter "LTYPE". Most commonly L depends upon I(Lxxx), which is assumed with the default of LTYPE=0 as explained below. |
| *LTYPE* | If inductance L is a function of I(Lxxx), set LTYPE to 0. Otherwise, set LTYPE to 1. The inductance flux is calculated differently depending on the value of LTYPE. LTYPE must be set properly to provide correct simulation results. Defining L as function of multiple variables is not recommended. Default=0. |

**Example**

```
LLINK   42   69   1UH
LSHUNT 23 51 10U  .001 0  15  IC=15.7MA
```

**Parameter Limit Checking**

HSPICE writes a warning message to the output listing file if an inductive element value exceeds 0.1 henry. This feature eases identification of elements with missing units or wrong values, particularly those in automatically produced netlists.

**Inductance Temperature Equation**

The effective inductance as a function of temperature is provided by the following equation:

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

| | | |
|---|---|---|
| $\Delta t$ | t - tnom | |
| *t* | element temperature in degrees Kelvin | |
| | | t=circuit temp + DTEMP + 273.15 |
| *tnom* | nominal temperature in degrees Kelvin | |
| | | tnom=273.15 + TNOM |

## Mutual Inductor Element

### General form

```
Kxxx Lyyy Lzzz kvalue
or
Kxxx Lyyy Lzzz K=val
```

| | |
|---|---|
| *Kxxx* | mutual (coupled) inductor element name. Must begin with a "K", which can be followed by up to 15 alphanumeric characters. |
| *Lyyy, Lzzz* | element names of coupled inductors |
| *kvalue, K* | the coefficient of mutual coupling. The absolute value of kvalue must be greater than 0 and less than or equal to 1. If kvalue is negative, the direction of coupling is reversed. This reversal is equivalent to reversing the polarity of either of the coupled inductors. If kvalue is a parameter, the syntax of kvalue should be K=parameter. |

### Example

```
K43 LAA LBB 0.9999
KXFTR L1 L4 K=0.87
```

Using the "dot" convention, a "dot" appears on the first node of each inductor.

Use the mutual inductor statement to specify coupling between more than two inductors. Also, automatically calculate second order coupling effects using the GENK and KLIM options. For example, if you specify three inductors, HSPICE automatically calculates the coupling between L3 and L1, if the coupling between L1, L2 and L2, L3 is given.

### Example

```
K1 L1 L2 .98
K2 L2 L3 .87
```

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 HSPICE issues an error message and terminates.*

## Polynomial Inductor Element

General form

```
Lxxx n1 n2 POLY c0 c1 … <IC=val>
```

| | |
|---|---|
| *Lxxx* | inductor element name. Must begin with an "L", which can be followed by up to 15 alphanumeric characters. |
| *n1, n2* | node names |
| *POLY* | keyword to identify the inductor as nonlinear polynomial |
| *c0 c1 ...* | coefficients of a polynomial describing the inductor value |
| *IC* | initial current through the inductor in amperes |
| *lval* | The inductance is described as a function of the instantaneous current, i, through the inductor. The value is computed as: |

$$lval = c0 + c1 \cdot i + c2 \cdot i^2$$

# Using Magnetics

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 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.

## Magnetic Winding Element

### General form

```
Lxxx  n1  n2  NT=val <R=val> <IC=val>
```

| *Lxxx* | the name of the winding, which must begin with an "L" followed by up to 15 alphanumeric characters |
| --- | --- |
| *n1, n2* | node names |
| *NT* | number of turns |
| *R* | DC resistance |
| *IC* | the initial current through the inductor in amperes. The "dot" convention is used to determine the direction of the turns. |

### Example

```
L3 4 5 NT=50 R=.01
LDRIVE 1 2 NT=100
```

# Mutual Core Statement

## General form

```
Kxxx Lyyy <Lzzz
… <Laaa>>
mname
<MAG=val>
```

| | |
|---|---|
| *Kxxx* | the saturable core element name. Must begin with a "K", which can be followed by up to 15 alphanumeric characters. |
| *Lyyy, Lzzz,* | the names of the windings about the Kxxx element. At least one winding must |
| *Laaa* | be specified. There is no limit to the total number of windings. The "dot" |
| | convention is used to determine the direction of the turns. |
| *mname* | model name reference |
| *MAG* | initial magnetization of the core. Can be set to -1, 0 or +1, where +1, -1 correspond to positive or negative values of model parameter BS in model "mname". Default=0.0. |

## Example

```
K2 L1 CHOKE
K5 L3 L5 L7 T1 MAG=1
```

## Magnetic Core Model

### General form

.MODEL mname L (<pname1=val1>…)

| | |
|---|---|
| *mname* | model name. Elements refer to the model by this name. |
| *L* | identifies a saturable core model |
| *pname1* | each saturable 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 12-1: 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 metawaves 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

The magnetic core model parameters are described in Table 12-6.

### Table 12-6: Definitions of 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 |
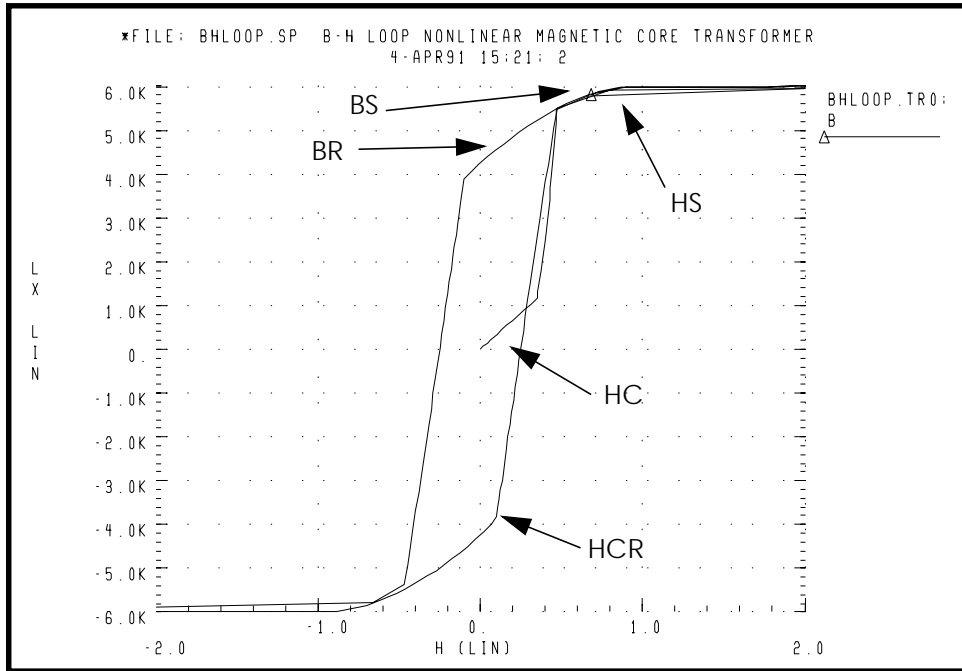
**Figure 12-1: Magnetic Saturable Core Model**

## Jiles-Atherton Ferromagnetic Core Model

The Jiles-Atherton ferromagnetic core model is based on domain wall motion, including both bending and translation. The hysteresis-free (anhysteretic) magnetization curve is described by a modified Langevin expression. This leads to

$$m_{an} = M\dot{S} \cdot \left( \coth\left(\frac{h_e}{A}\right) - \frac{A}{h_e} \right)$$

$$h_e = h + ALPHA \cdot m_{an}$$

where

$m_{an}$ is the magnetization level, if domain walls could move freely.

$h_e$ is the effective magnetic field.

is the magnetic field.

*MS* is a model parameter that represents the saturation magnetization.

*A* is a model parameter that characterizes the shape of the anhysteretic magnetization.

*ALPHA* is a 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 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. 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 bend 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 $\mu_0$, the permeability of free space, is $4\pi \cdot 10^{-7}$, and the units of $h$ and $m$ are in amp/meter. Then the units of $b$ would be in tesla (Wb/meter$^2$).

## Core Model Statement

### General Form
.MODEL mname CORE (LEVEL=1 <keyword = val> ... )

## Core Model Parameters

The core model parameters are listed in Table 12-7.

### Table 12-7: *Magnetic Model Parameter Definitions*

| 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 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 anisotropy parameter |

### Magnetic Core Outputs

### Table 12-8: Magnetic Core Element Outputs

| Output Variable | Description |
|---|---|
| LX1 | magnetic field, *h* (oersted) |
| LX2 | magnetic flux density, *b* (gauss) |
| LX3 | slope of the magnetization curve, $\frac{am}{dh}$ |
| LX4 | bulk magnetization, m (amp/meter) |
| LX5 | slope of the anhysteretic magnetization curve, $\frac{am_{an}}{dh}$ |
| LX6 | anhysteretic magnetization, $m_{an}$ (amp/meter) |
| LX7 | effective magnetic field, $h_e$ (amp/meter) |

### Jiles-Atherton Model Examples

#### Example 1 – 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.

Table 12-2 shows the *b-h* curves for three values of ALPHA.

Table 12-3 shows the *b-h* curves for three values of A.

Table 12-4 shows the *b-h* curves for three values of K.

#### 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
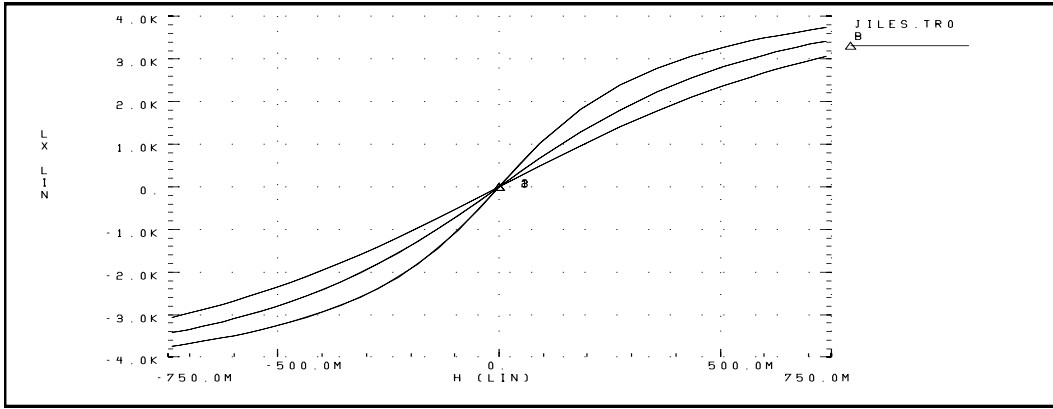


**Figure 12-2: Variation of Anhysteretic b-h Curve: the Slope Increases as ALPHA Increases**
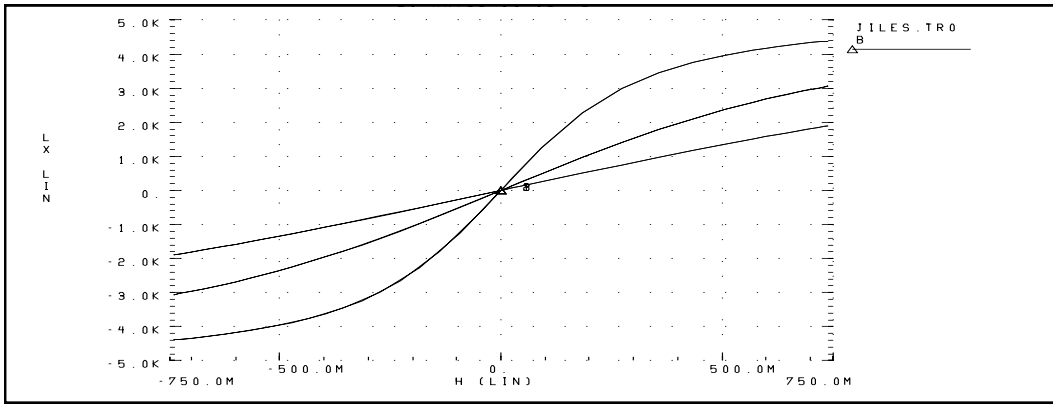


**Figure 12-3: Variation of Anhysteretic b-h Curve: the Slope Decreases as A Increases**
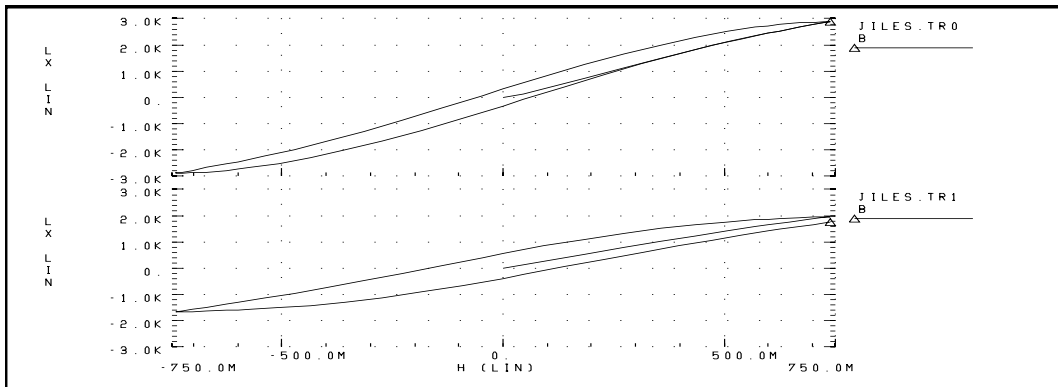
**Figure 12-4: Variation of Hysteretic b-h Curve: as K Increases, the
Loop Widens and Rotates Clockwise**

### Example 2 – 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 12-5: demonstrates the effects of
hysteresis on the inductance of the core.

### 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
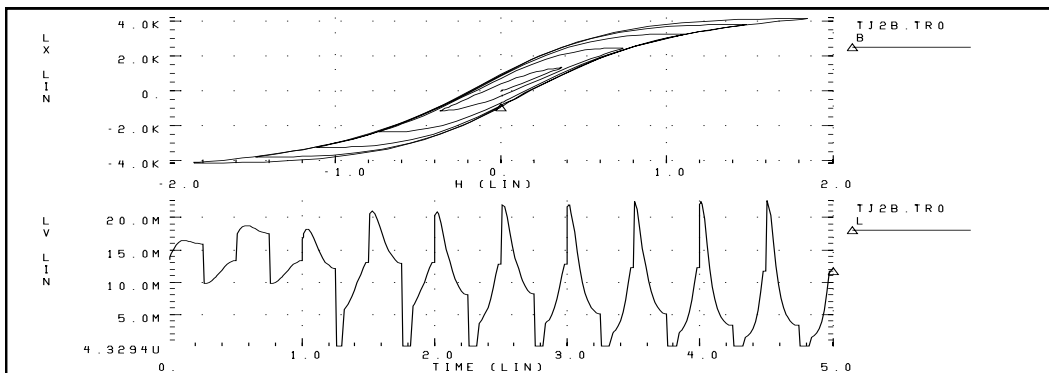


## Figure 12-5: Hysteresis Curve and Inductance of a Magnetic Core

### Example 3 – Optimization of Parameter Extraction

This example demonstrates the usage of optimization in the parameter extraction of the Jiles-Atherton model. Figure 12-6: shows the plots of the core output before and after optimization.

**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
```
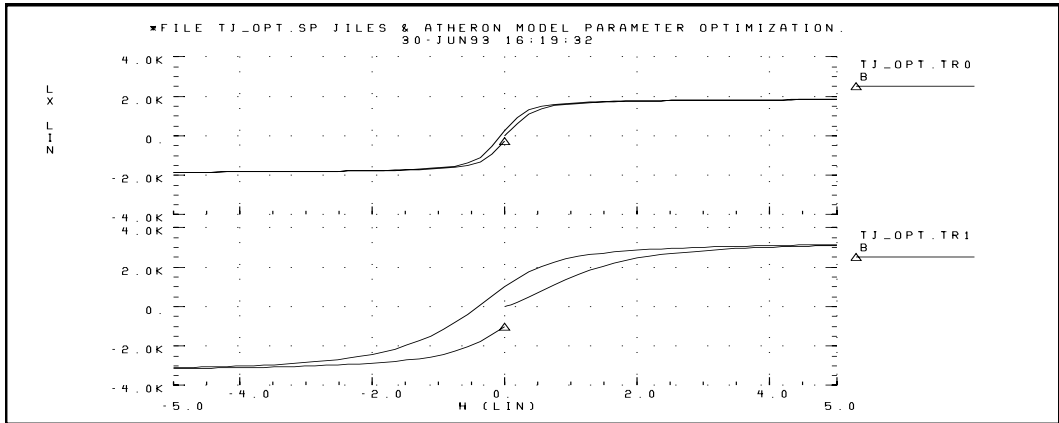
**Figure 12-6: Output Curves Before Optimization (top), and After Optimization (bottom)**

*Avant!*

# Chapter 13

# 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 Chapter 15, Introducing MOSFET; Chapter 16, Selecting a MOSFET Model; and Chapter 22, Performing Behavioral Modeling 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 element and model parameters and scaling effects for the geometric and nongeometric junction diodes.The following topics are covered in this chapter:

- Understanding the Diode Types
- Using Model and Element Statements
- Specifying Junction Diodes
- Calculating Temperature Effects
- Using 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. Use the geometric parameter to 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 allows you to 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 Model and Element 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.

## 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.

### 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.

## 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.

## 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 Hspice automatically places in the circuit). You can specify GMIN and GMINDC in the .OPTIONS statement.

The diode control options follow:

| | |
|---|---|
| Capacitance | DCAP, DCCAP |
| Conductance | GMIN, GMINDC |
| Geometry | SCALM, SCALE |

# Specifying Junction Diodes

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 13-1: – 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 |

# Diode Element

General form

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

or

```
Dxxx nplus nminus mname <area_val <periphery_val>> <OFF>
<IC=vd> <M=val>
```

or

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

| | |
|---|---|
| AREA | Area of the diode. It modifies saturation currents, capacitances, and resistances. Area factor for LEVEL=1 model is not affected by the option SCALE. Default=1.0. Affects IK, IKR, JS, CJO, and RS. |
| | For LEVEL=3, |
| | AREAeff = AREA · M · SCALE2 · SHRINK2, or |
| | AREAeff = Weff · Leff · M |
| | The effective area overrides model parameter AREAeff calculated from model parameter AREA. If unspecified, AREA is calculated from W, L. |
| DTEMP | The difference between element temperature and the circuit temperature. Default=0.0. |
| Dxxx | Diode element name. Must begin with a **D**, which can be followed by a maximum of 15 alphanumeric characters. |

| | |
|---|---|
| IC=vd | Initial voltage across the diode element. Interacts with the UIC option in the .TRAN statement and is overridden by the .IC statement. |
| L | Length of diode in meters (Level 3 only) |
| | `Leff = L   SCALE   SHRINK + XWeff` |
| LM | Length of metal capacitor in meters (Level 3 only). Overrides model parameter LM in the model. Default=0.0. |
| | `LMeff = LM   SCALE   SHRINK.` |
| LP | Length of polysilicon capacitor in meters (Level 3 only). Overrides model parameter LP in model. Default=0.0. |
| | `LPeff = LP   SCALE   SHRINK.` |
| M | Multiplier factor to simulate multiple diodes. All currents, capacitances, and resistances are affected by M=val. Default=1. |
| mname | Model name. It can be up to 16 characters long. |
| nplus | Positive terminal (anode) node name. It can be up to 16 characters long. Series resistor is attached to this terminal. |
| nminus | Negative terminal (cathode) node name. It can be up to 16 characters long. |
| OFF | Switch that sets initial condition to OFF for the element in DC analysis. Default=ON. |
| PJ | Periphery of junction. Overrides PJ in model. Calculated from W, L if specified. Affects JSW and CJP model parameters. Default=0.0. |
| | For LEVEL=1, |

```
                          PJeff = PJ    M
```

For LEVEL=3,

```
                          PJeff = PJ    SCALE    M    SHRINK
                          PJeff = (2    Weff + 2    Leff)    M
```

|       |                                                                           |
|-------|---------------------------------------------------------------------------|
| W     | Width of diode in meters (Level 3 only)                                   |
|       | `Weff = W    SCALE    SHRINK + XWeff`                                      |
| WM    | Width of metal capacitor (Level 3 only). Overrides WM in the model. Default=0.0 |
|       | `WMeff = WM    SCALE    SHRINK`                                            |
| WP    | Width of polysilicon capacitor in meter (Level 3 only). Overrides WP in model. Default=0.0 |
|       | `WPeff = WP    SCALE    SHRINK`                                            |

## Examples

The following example shows how to connect a diode called DCLMMMP between node 3 and substrate. The diode has a voltage of 0.2 V at timepoint 0 in a transient analysis. The model statement with the model reference name DMOD contains the diode model parameters.

```
DCLMMMP 3 substrate DMOD 3 IC=0.2
```

## 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:

```
AREAeff = AREA·M·SCALE²·SHRINK²
PJeff= PJ·SCALE·M·SHRINK
```

or, if W and L are specified,

```
AREAeff = Weff ·Leff·M
PJeff = (2·Weff + 2·Leff)·M
```

where

```
Weff =  W·SCALE·SHRINK + XWeff
Leff =  L·SCALE ·SHRINK + XWeff
```

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

```
JSWeff= PJeff·(JSW/SCALM)
CJPeff=PJeff·(CJP/SCALM)
```

To determine the polysilicon and metal capacitor dimensions, multiply each by
SCALE or by SCALM if specified as model parameters.

```
LMeff= LM·SCALE·SHRINK
WMeff= WM·SCALE·SHRINK
LPeff= LP·SCALE·SHRINK
WPeff= WP·SCALE·SHRINK
XPeff= XP·SCALM
XMeff= XM·SCALM
```

You can determine the effective scaled model parameters, IBeff, IKeff, IKReff, IBVeff, RSeff, and CJO as follows:

```
IKeff= AREAeff·IK
IKReff= AREAeff·IKR
```

$$IBVeff=(AREAeff \cdot IBV)/SCALM^2$$

$$ISeff= IS \cdot (AREAeff/SCALM^2)$$

$$RSeff= RS/(AREAeff \cdot SCALM^2)$$

$$CJOeff=AREAeff \cdot (CJO/SCALM^2)$$

### Diode Current

Figure 13-1 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.
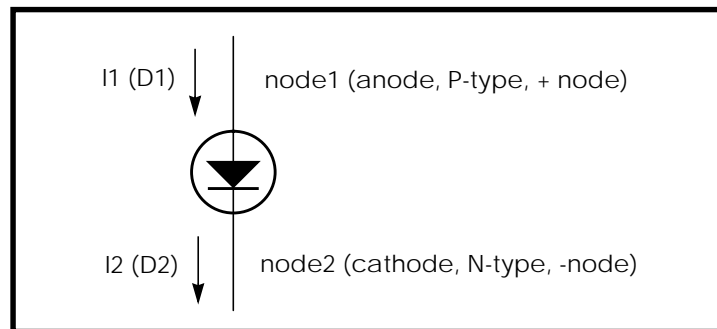


**Figure 13-1: Diode Current Convention**

## Diode Equivalent Circuits

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. Hspice assumes capacitance effects to be separate from the id equations.
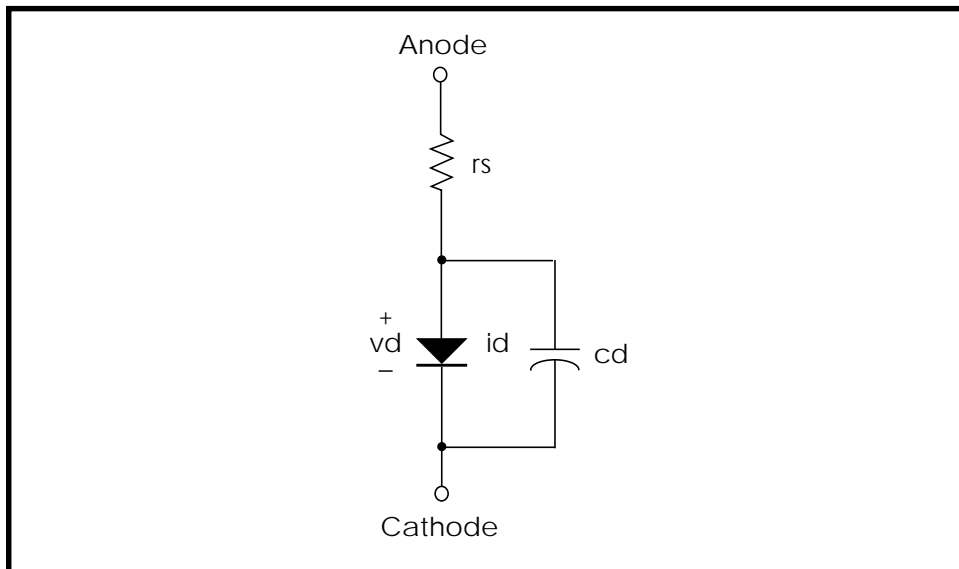


**Figure 13-2: Equivalent Circuit, Diode Transient Analysis**
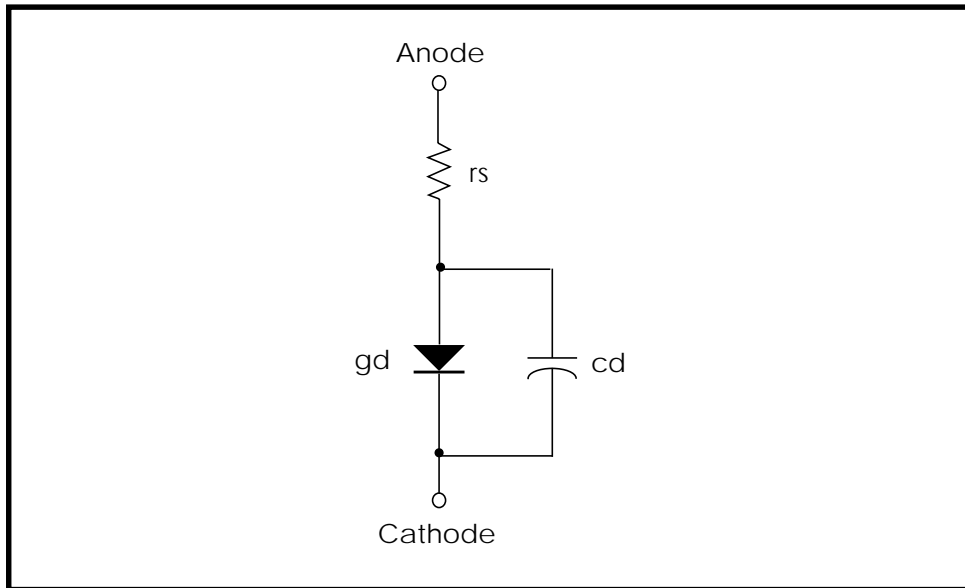
**Figure 13-3: Equivalent Circuit, Diode AC Analysis**
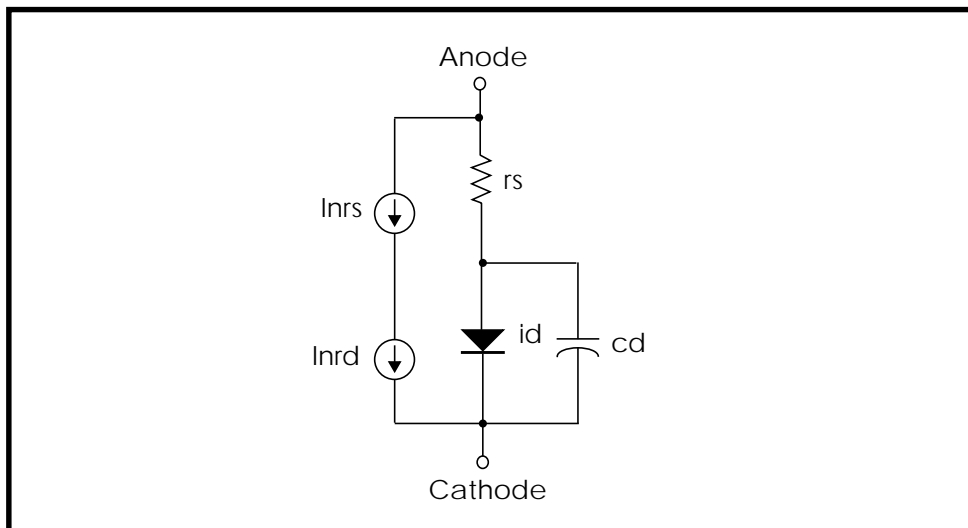


**Figure 13-4: Equivalent Circuit, Diode AC Noise Analysis**

## Using the Junction Model Statement

This section describes how to use the junction model statement.

### Format

The format 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 | Type of diode. The diode model includes three diode types: |
| | LEVEL=1 = junction diode |
| | LEVEL=2 = Fowler-Nordheim |
| | LEVEL=3 = geometric processing for junction diode |
| keyword | Model parameter keyword such as CJO or IS |

### Examples

Examples of the jumction model statements are:

```
.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=1XP  =0.0EG  =1.1
+ XOI  =0.0XOM =0.0XM  =0.0
+ WP   =0.0WM  =0.0LP  =0.0
+ LM   =0.0AF  =1.0JSW =0.0
+ PB   =0.65PHP =0.8M   =0.2994
+ FC   =0.95FCS =0.4MJSW=0.5
```

```
+ TT   =2.446e-9BV  =4.65RS  =19
+ IS   =1.485e-11CJO =1.09e-9CJP =0.0
+ PJ   =0.0N   =1.615IK  =0.0
+ IKR  =1.100e-2IBV =2.00e-2
```

## 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  shows the junction diode model parameters and their function.)

### Table 13-2: 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 |

### Junction DC Parameters Level 1 and 3

Table  shows the junction DC parameters for Levels one and three:

#### Table 13-3: Junction DC parameters for Level  and 3

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| AREA | | 1.0 | Junction area<br>For LEVEL=1<br>AREAeff = AREA · M, unitless<br>For LEVEL=3<br>AREAeff=AREA · SCALM$^2$ · SHRINK$^2$ · M unit = meter$^2$<br>If you specify W and L:<br>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.<br>EXPLIeff = EXPLI · AREAeff |
| IB | amp | 1.0e-3 | Current at breakdown voltage<br>For LEVEL=3<br>IBVeff = IBV · AREAeff / SCALM$^2$ |
| IBV | amp | 1.0e-3 | Current at breakdown voltage<br>For LEVEL=3<br>IBVeff = IBV · AREAeff / SCALM$^2$ |
| IK (IKF, JBF) | amp/ AREAeff | 0.0 | Forward knee current (intersection of the high- and low-current asymptotes)<br>IKeff = IK · AREAeff. |
| IKR (JBR) | amp/ AREAeff | 0.0 | Reverse knee current (intersection of the high- and low-current asymptotes)<br>IKReff = IKR · AREAeff. |

## Table 13-3: Junction DC parameters for Level  and 3

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| IS (JS) | amp/ AREAeff | 1.0e-14 | If you use an IS value less than EPSMIN, the program resets the value of IS to EPSMIN and displays a warning message.<br>EPSMIN default=1.0e-28<br>If the value of IS is too large, the program displays a warning<br>For LEVEL=1<br>ISeff = AREAeff · IS<br>For LEVEL=3<br>ISeff = AREAeff · IS/SCALM$^2$ |
| JSW (ISP) | amp/ PJeff | 0.0 | Sidewall saturation current per unit junction periphery<br>For LEVEL=1<br>JSWeff = PJeff · JSW<br>For LEVEL=3<br>JSWeff = PJeff · JSW/SCALM |
| L | | | Default length of diode<br>Leff = L · SHRINK · SCALM+ XWeff |
| LEVEL | | 1 | Diode model selector<br>LEVEL=1 or LEVEL=3 selects junction diode model<br>LEVEL=2 selects Fowler-Nordheim model |
| N | | 1.0 | Emission coefficient |
| PJ | | 0.0 | Junction periphery<br>For LEVEL=1<br>PJeff = PJ · M, unitless<br>For LEVEL=3<br>PJeff = PJ · SCALM · M · SHRINK, meter<br>If W and L are specified<br>PJeff = (2 · Weff + 2 · Leff) · M, meter |

## Table 13-3: Junction DC parameters for Level  and 3

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| RS | ohms<br><br>or<br><br>ohms/m$^2$<br>See<br>Note. | 0.0 | Ohmic series resistance<br>For LEVEL=1<br>RSeff = RS/AREAeff<br>For LEVEL=3<br>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<br>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$^2$.*

### Junction Capacitance Parameters

Table 13-4 shows the junction capacitance parameters:

#### Table 13-4: 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/SCALM$^2$ |
| CJP | F/PJeff | 0.0 | Zero-bias junction capacitance per unit junction periphery (PJ) |
| (CJSW) | | | 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, | V | 0.8 | Area junction contact potential |
| VJ, PHA) | | | |

## Table 13-4: Junction Capacitance Parameters

| Name(Alias) | Units | Default | Description |
| --- | --- | --- | --- |
| PHP | V | PB | Periphery junction contact potential |
| TT | s | 0.0 | Transit time |

### Metal and Poly Capacitor Parameters Level=3

Table 13-5 shows the metal and poly capacitor parameters for Level 3:

## Table 13-5: 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.<br>LMeff = LM · SCALM · SHRINK |
| LP | m | 0.0 | Use this parameter if LP is not specified in the element statement.<br>LPeff = LP · SCALM · SHRINK |
| WM | m | 0.0 | Use this parameter if WM is not specified in the element statement.<br>WMeff = WM · SCALM · SHRINK |
| WP | m | 0.0 | Use this parameter if WP is not specified in the element statement.<br>WPeff = WP · SCALM · SHRINK |
| XM | m | 0.0 | XM accounts for masking and etching effects:<br>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<br>XPeff = XP · SCALM |

## Noise Parameters LEVEL=1 and 3

Table 13-5 shows the metal and poly capacitor parameters for Level 3

### Table 13-6: Noise Parameters Level 1 and 3

| Name(Alias) | Units | Default | Description |
|-------------|-------|---------|------------------------|
| AF          |       | 1.0     | flicker noise exponent |
| KF          |       | 0.0     | flicker noise coefficient |

# Calculating Temperature Effects

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 13-7: 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 |

## Temperature Effect Parameters LEVEL=1 and 3

### Table 13-8: Temperature Effect Parameters

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

# Table 13-8: Temperature Effect Parameters

| 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 <br> 1.17 - silicon <br> 0.69 - Schottky barrier diode <br> 0.67 - germanium <br> 1.52 - gallium arsenide |
| GAP1 | eV/° | 7.02e-4 | 7.02e-4 - silicon (old value) <br> 4.73e-4 - silicon <br> 4.56e-4 - germanium <br> 5.41e-4 - gallium arsenide |
| GAP2 | | 1108 | 1108 - silicon (old value) <br> 636 - silicon <br> 210 - germanium <br> 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. |
| TREF | | 25.0 | Model reference temperature (Level 1 or 3 only) |

### Table 13-8: Temperature Effect Parameters

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 Diode Equations

Table 13-9 shows the diode equation variable definition:

**Table 13-9: 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 13-10 shows the equation quantity definition:

**Table 13-10: 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 |
| $\Delta t$ | 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 |

## 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**

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

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.

## Diode Capacitance Equations

The diode capacitance is modeled by cd in Figure 13-4. The capacitance, cd, is a combination of diffusion capacitance, (cdiff), depletion capacitance, (cdep), metal, (cmetal), and poly capacitances, (cpoly).

$$cd \; = \; cdiff + cdep + cmetal + cpoly$$

### 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.

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

## 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 which are selected using .OPTIONS DCAP.

### *DCAP=1*

The junction bottom area capacitance formula is:

**$vd < FC \cdot PB$**

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

**$vd \; FC \cdot PB$**

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

The junction periphery capacitance formula is:

**$vd < FCS \cdot PHP$**

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

**$vd \; FCS \cdot PHP$**

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

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 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 $FC \cdot CGDeff$ or $FC \cdot CGSeff$, with proper fall-off when forward bias exceeds PB ($FC \geq 1$).

## Metal and Poly Capacitance Equations (LEVEL=3 Only)

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

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

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

## Noise Equations

Figure 13-4 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 Element

The format of the Fowler-Nordheim diode element is:

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

| | |
|---|---|
| Dxxx | Diode element name. Must begin with the letter **D**, and can be followed by a maximum of 15 alphanumeric characters. |
| nplus | Positive (anode) terminal node name (can be up to 16 characters) |
| nminus | Negative (cathode) terminal node name (can be up to 16 characters) |
| mname | Model name. Must reference a LEVEL=2 model for a Fowler- Nordheim diode element. |
| W | Width of diode in units of meter. Overrides W in the LEVEL=2 model. Default=0.0. <br><br> `Weff = W · SCALE · SHRINK + XWeff` |
| L | Length of diode in units of meter. Overrides L in the LEVEL=2 model. Default=0.0. <br><br> `Leff = L · SCALE · SHRINK + XWeff` |

| OFF | Sets initial condition to OFF for this element in DC analysis. Default=ON. |
| --- | --- |
| IC=vd | Initial voltage across this diode element. Interacts with the UIC option on the .TRAN statement and overridden by the .IC statement. |
| M | Multiplier factor to simulate multiple diodes. M affects all currents and capacitances. Default=1.0. |

**Example**

```
*FILE: /TUN.SP
.OPTION GMINDC=1E-22 GMIN=1E-22 PIVTOL=1E-23
D1 1 0 TMOD   W=5E-4   L=5E-4
.MODEL TMOD D LEVEL=2 EF=3E8 JF=2E-6  TOX=100
.TRAN .2 5S
VD 1 0 PL 5V 0S    10V 5S
.PRINT V(1) I(D1)
.END
```

The SCALE element parameter scales the length and width of Fowler-Nordheim diode models.

SCALM is used as the scaling factor if length and width are specified as model parameters.

If both element and model parameters are specified, the element parameter is used along with the SCALE scaling factor. The effective area is then determined as the product of the scaled effective length and width.

## Fowler-Nordheim Diode Model Parameters LEVEL=2

Table 13-11 shows the Fowler-Nordheim diode model parameters for Level 2:

### Table 13-11: 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 |

### 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  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 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.

### Scaled Diode Subcircuit Definition

The scaled diode subcircuit definition converts the National Semiconductor scaled diode model to a form a model usable in 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:    All the parameters used in the following model must have a value which 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!*

<div align="right">

# Chapter 14

# BJT Models

</div>

IThe bipolar-junction transistor (BJT) model in HSPICE is an adaptation of the integral charge control model of Gummel and Poon.

The 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 the BJT Model
- Using the BJT Element
- Understanding the BJT Model Statement
- Using the BJT Models (NPN and PNP)
- Understanding BJT Capacitances
- Modeling Various Types of Noise
- Using the BJT Quasi-Saturation Model
- Using Temperature Compensation Equations
- Converting National Semiconductor Models

# Using the BJT Model

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.

## Model Selection

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 reference name is given as MOD1 in the following example. 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.

## Control Options

Control options affecting the BJT model are: DCAP, GRAMP, GMIN, and
GMINDC. DCAP selects the equation which 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.

### Table 14-1: BJT Options

| capacitance | DCAP, DCCAP |
|---|---|
| conductance | GMIN, GMINDC, GRAMP |

Override global depletion capacitance equation selection that uses the .OPTION
DCAP=<*val*> statement in a BJT model by including DCAP=<val> in the
BJT's .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.

# Using the BJT Element

The BJT element parameters specify the connectivity of the BJT, normalized geometric specifications, initialization, and temperature parameters.

### Table 14-2: BJT Element Parameters

| Type | Parameters |
|------|-----------|
| *netlist* | Qxxx, mname, nb, nc, ne, ns |
| geometric | AREA, AREAB, AREAC, M |
| initialization | IC (VBE, VCE), OFF |
| temperature | DTEMP |

### General form

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

or

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

| | |
|---|---|
| Qxxx | BJT element name. Must begin with a "Q", which can be followed by up to 15 alphanumeric characters. |
| nc | collector terminal node name |
| nb | base terminal node name |
| ne | emitter terminal node name |
| ns | substrate terminal node name, optional. Can be set in the model with BULK= Node name. |
| mname | model name reference |
| aval | value for AREA |

| OFF | sets initial condition to OFF for this element in DC analysis. Default=ON. |
| IC=vbeval, | initial internal base to emitter voltage (vbeval) or initial internal collector to |
| vceval | emitter voltage (vceval). Overridden by the .IC statement. |
| M | multiplier factor to simulate multiple BJTs. All currents, capacitances, and resistances are affected by M. |
| DTEMP | the difference between element and circuit temperature (default= 0.0) |
| AREA | emitter area multiplying factor that affects resistors, capacitors, and currents (default=1.0) |
| AREAB | base area multiplying factor that affects resistors, capacitors, and currents (default=AREA) |
| AREAC | collector area multiplying factor that affects resistors, capacitors, and currents (default=AREA) |

### Examples

```
Q100 CX BX EX QPNP AREA=1.5 AREAB=2.5 AREAC=3.0
Q23 10 24 13  QMOD IC=0.6,5.0
Q50A 11 265 4 20 MOD1
```

## 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.

## BJT Current Convention

The direction of current flow through the BJT is assumed for example purposes in Figure 13-1. 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.
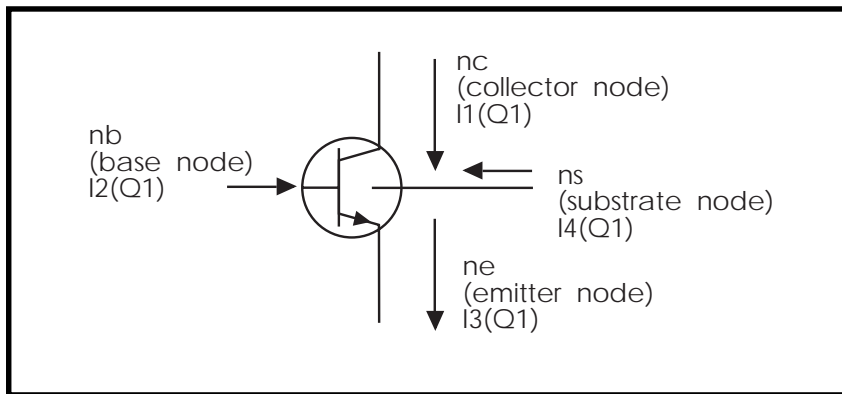


*Figure 13-1 – BJT Current Convention*

## BJT Equivalent Circuits

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 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. The partial derivatives of ib and ic with respect to the terminal voltages vbe and vbc are used instead. 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**

$$gm = \frac{\partial ic}{\partial vbe}\bigg|_{vce\,=\,con.}$$

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

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

The ib and ic equations account for all DC effects of the BJT.
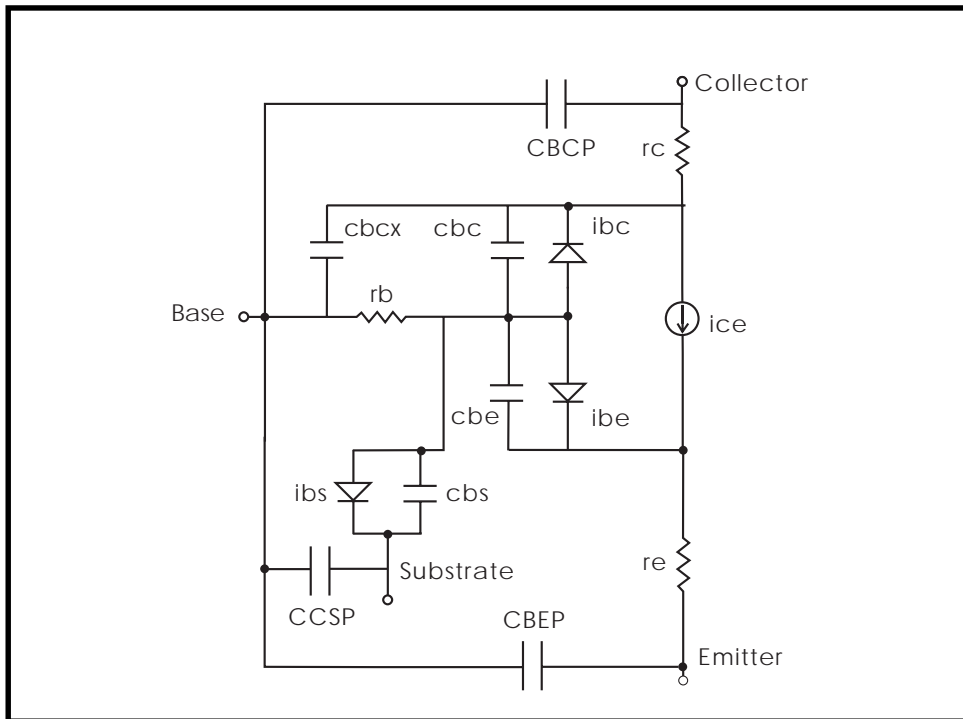


**Figure 13-1: Lateral Transistor, BJT Transient Analysis**
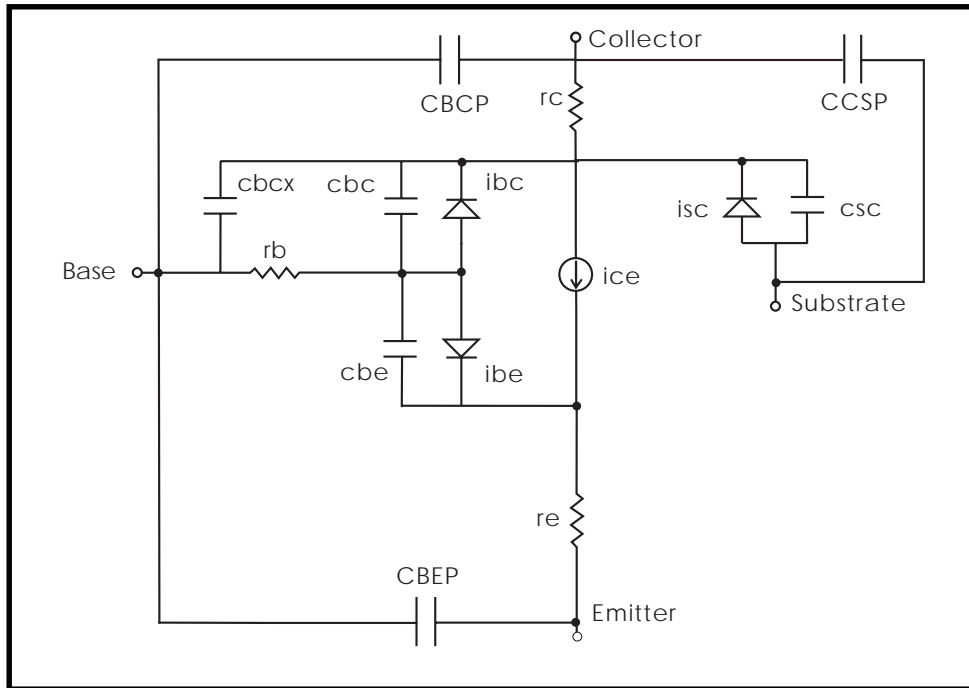
**Figure 13-2: Vertical Transistor, BJT Transient Analysis**
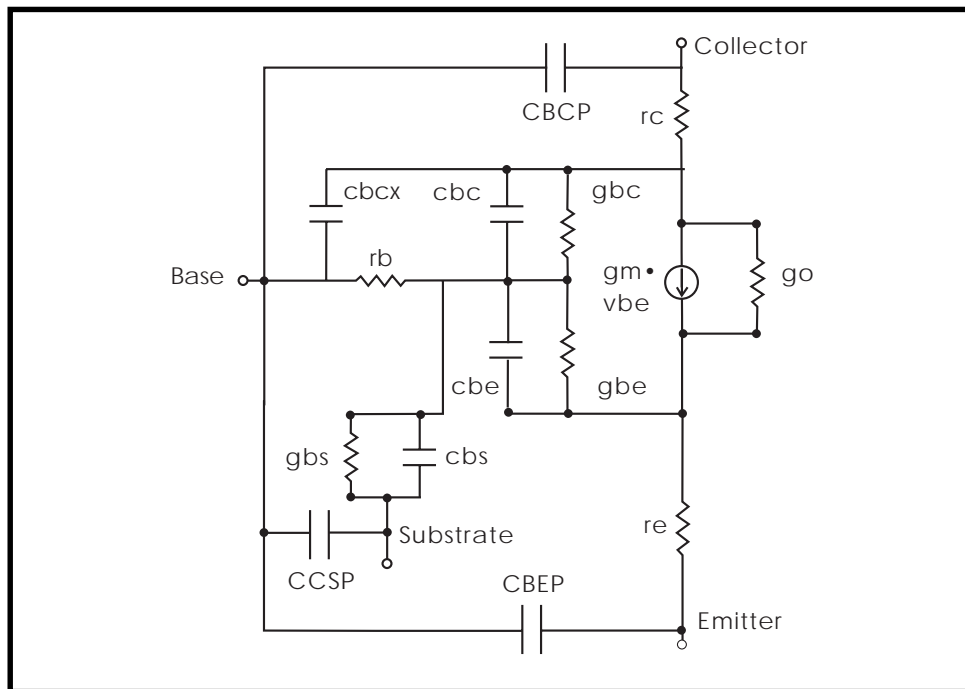
**Figure 13-3: Lateral Transistor, BJT AC Analysis**
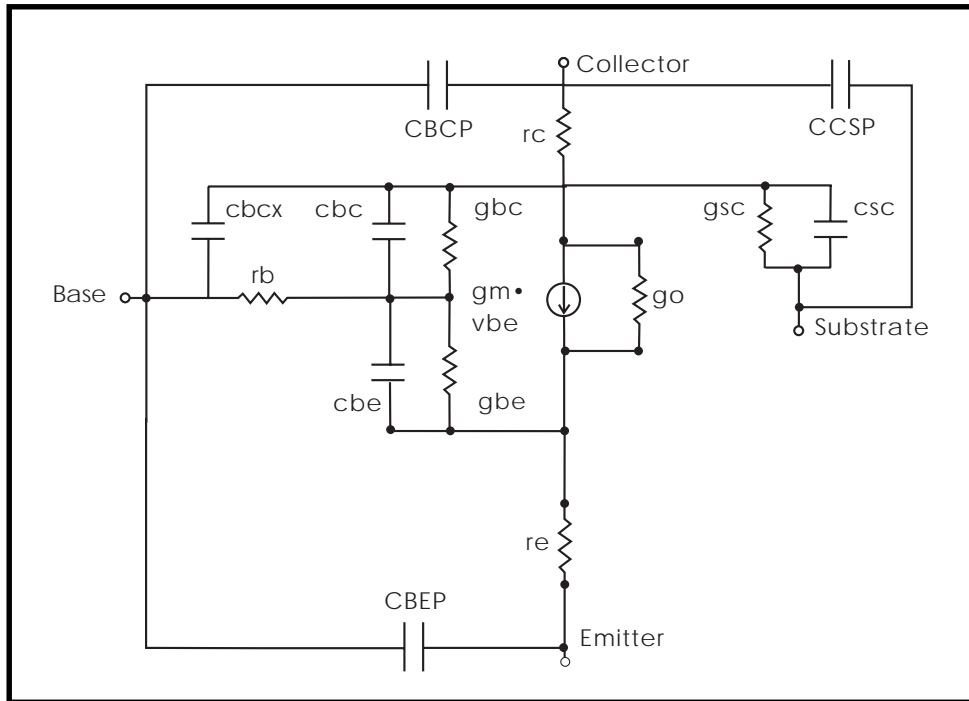
**Figure 13-4: Vertical Transistor, BJT AC Analysis**
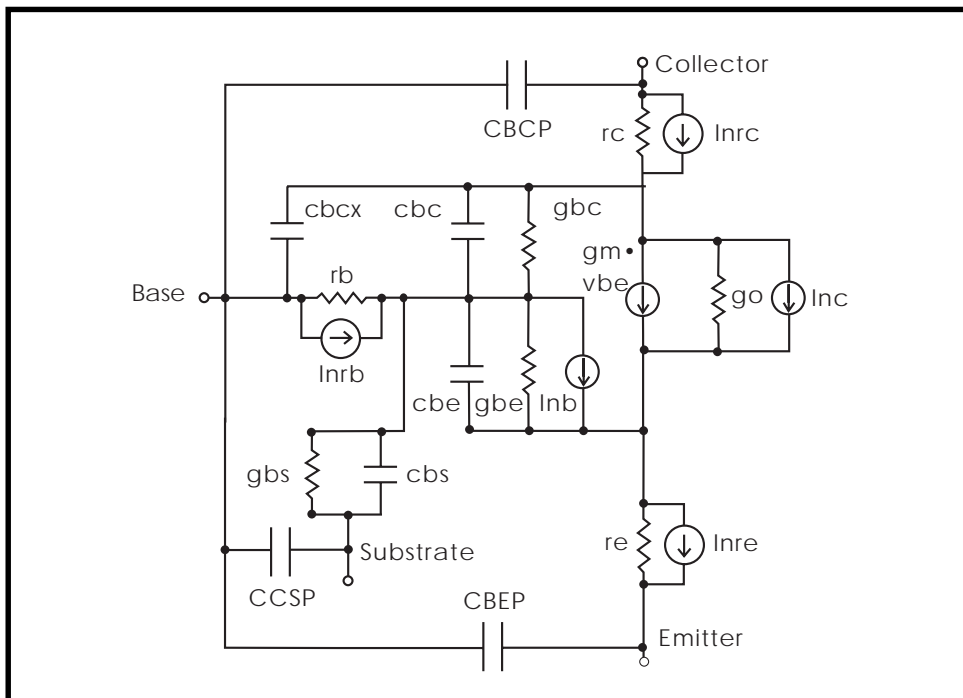
**Figure 13-5: Lateral Transistor, BJT AC Noise Analysis**
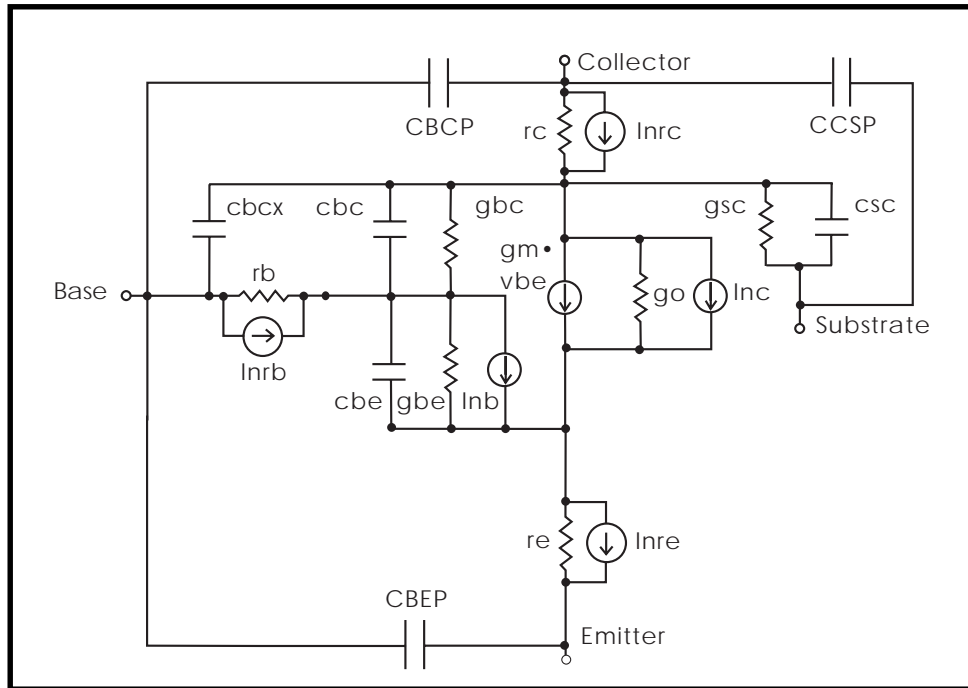
**Figure 13-6: Vertical Transistor, BJT AC Noise Analysis**

**Table 13-2: 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 |

## Table 13-2: Equation Variable Names

| Variable | Definitions |
|----------|-------------|
| gbe | forward base conductance |
| gm | transconductance |
| 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 13-3: Equation Constants

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

# Understanding the BJT Model Statement

### General form

```
.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     F   = 153.40622
+ BR = 40.00000     IS =   1.6339e-14 VAF= 103.40529
+ VAR= 17.77498     IKF=   1.00000     IS =   4.6956e-15
+ 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
```

## BJT Basic Model Parameters

To permit the use of model parameters from earlier versions of HSPICE, many of the model parameters have aliases, which are included in the model parameter list in "BJT Basic DC Model Parameters" on page 14-19. 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, typically 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 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 13-4: – 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 |

## BJT Basic DC Model Parameters

| Name(Alias) | Units | 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. |
| IBC | amp | 0.0 | reverse saturation current between base and collector. If both IBE and IBC are specified, HSPICE uses them in place of IS to calculate DC current and conductance, otherwise IS is used.<br><br>$IBCeff = IBC \cdot AREAB \cdot M$<br><br>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.<br><br>$EXPLIeff = EXPLI \cdot AREAeff$ |
| IBE | amp | 0.0 | reverse saturation current between base and emitter. If both IBE and IBC are specified, HSPICE uses them in place of IS to calculate DC current and conductance, otherwise IS is used.<br><br>$IBEeff = IBE \cdot AREA \cdot M$ |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| IS | amp | 1.0e-16 | transport saturation current. If both IBE and IBC are specified, HSPICE uses them in place of IS to calculate DC current and conductance, otherwise IS is used. |
| | | | 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 |
| 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 |

## Low Current Beta Degradation Effect Parameters

| | | | |
|---|---|---|---|
| ISC (C4, JLC) | amp | 0.0 | base-collector leakage saturation current. If ISC is greater than 1e-4, then: |
| | | | $ISC = IS \cdot ISC$ |
| | | | otherwise: |
| | | | $ISCeff = ISC \cdot AREAB \cdot M$ |
| | | | AREAC replaces AREAB, depending on vertical or lateral geometry. |
| ISE (C2, JLE) | amp | 0.0 | base-emitter leakage saturation current. If ISE is greater than 1e-4, then: |
| | | | $ISE = IS \cdot ISE$ |
| | | | otherwise: |
| | | | $ISEeff = ISE \cdot AREA \cdot M$ |
| NC (NLC) | | 2.0 | base-collector leakage emission coefficient |
| NE (NLE) | | 1.5 | base-emitter leakage emission coefficient |

## Base Width Modulation Parameters

| 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. |

## High Current Beta Degradation Effect Parameters

| IKF IK,JBF) | amp | 0.0 | corner for forward Beta high current roll-off. Use zero to indicate an infinite value. |
|---|---|---|---|
| | | | $IKFeff = IKF \cdot AREA \cdot M$ |
| IKR (JBR) | amp | 0.0 | corner for reverse Beta high current roll-off. Use zero to indicate an infinite value |
| | | | $IKReff = IKR \cdot AREA \cdot M$ |
| NKF | | 0.5 | exponent for high current Beta roll-off |

## Parasitic Resistance Parameters

| IRB (JRB,IOB) | amp | 0.0 | base current, where base resistance falls half-way to RBM. Use zero to indicate an infinite value. |
|---|---|---|---|

$$\text{IRBeff} = \text{IRB} \cdot \text{AREA} \cdot \text{M}$$

| RB | ohm | 0.0 | base resistance |
|---|---|---|---|

$$\text{RBeff} = \text{RB} / (\text{AREA} \cdot \text{M})$$

| RBM | ohm | RB | minimum high current base resistance |
|---|---|---|---|

$$\text{RBMeff} = \text{RBM} / (\text{AREA} \cdot \text{M})$$

| RE | ohm | 0.0 | emitter resistance |
|---|---|---|---|

$$\text{REeff} = \text{RE} / (\text{AREA} \cdot \text{M})$$

| RC | ohm | 0.0 | collector resistance |
|---|---|---|---|

$$\text{RCeff} = \text{RC} / (\text{AREA} \cdot \text{M})$$

## Junction Capacitor Parameters

| CJC | F | 0.0 | base-collector zero-bias depletion capacitance |
|---|---|---|---|

Vertical: $\text{CJCeff} = \text{CJC} \cdot \text{AREAB} \cdot \text{M}$

Lateral: $\text{CJCeff} = \text{CJC} \cdot \text{AREAC} \cdot \text{M}$

| CJE | F | 0.0 | base-emitter zero-bias depletion capacitance (vertical and lateral): |
|---|---|---|---|

$$CJEeff = CJE \cdot AREA \cdot M$$

| *CJS (CCS, CSUB)* | F | 0.0 | zero-bias collector substrate capacitance |
|---|---|---|---|

$$\text{Vertical: } CJSeff = CJS \cdot AREAC \cdot M$$
$$\text{Lateral: } CJSeff = CJS \cdot AREAB \cdot M$$

| *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 |

## Parasitic Capacitances

| CBCP | F | 0.0 | external base-collector constant capacitance |

$$CBCPeff = CBCP \cdot AREA \cdot M$$

| CBEP | F | 0.0 | external base-emitter constant capacitance |

$$CBEPeff = CBEP \cdot AREA \cdot M$$

| CCSP | F | 0.0 | external collector substrate constant capacitance (vertical) or base substrate (lateral) |

$$CCSPeff = CCSP \cdot AREA \cdot M$$

## Transit Time Parameters

| ITF (JTF) | amp | 0.0 | TF high-current parameter |

$$ITFeff = ITF \cdot AREA \cdot M$$

| PTF |  | 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. Use zero to indicate an infinite value. |
| XTF |  | 0.0 | TF bias dependence coefficient |

## Noise Parameters

| | | | |
|---|---|---|---|
| AF | | 1.0 | flicker-noise exponent |
| KF | | 0.0 | flicker-noise coefficient |

## BJT LEVEL=2 Model Parameters

| | | | |
|---|---|---|---|
| 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: $QCOeff = QCO \cdot AREAB \cdot M$ |
| | | | Lateral: $QCOeff = QCO \cdot AREAC \cdot M$ |
| RC | ohm | 0.0 | resistance of the epitaxial region under equilibrium conditions |
| | | | $RCeff = RC/(AREA \cdot M)$ |
| VO | V | 0.0 | carrier velocity saturation voltage. Use zero to indicate an infinite value. |

## 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 13-5: – 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 |

### Temperature Effect Parameters

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| BEX | | 2.42 | VO temperature exponent (Level 2 only) |
| BEXV | | 1.90 | RC temperature exponent (Level 2 only) |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| CTC | 1/° | 0.0 | temperature coefficient for zero-bias base collector capacitance. TLEVC=1 enables CTC to override the default HSPICE temperature compensation. |
| CTE | 1/° | 0.0 | temperature coefficient for zero-bias base emitter capacitance. TLEVC=1 enables CTE to override the default HSPICE temperature compensation. |
| CTS | 1/° | 0.0 | temperature coefficient for zero-bias substrate capacitance. TLEVC=1 enables CTS to override the default HSPICE temperature compensation. |
| EG | eV | | energy gap for pn junction<br><br>for TLEV=0 or 1, default=1.11;<br><br>for TLEV=2, default=1.16<br><br>1.17 - silicon<br>0.69 - Schottky barrier diode<br>0.67 - germanium<br>1.52 - gallium arsenide |
| GAP1 | eV/° | 7.02e-4 | first bandgap correction factor (from Sze, alpha term)<br><br>7.02e-4 - silicon<br>4.73e-4 - silicon<br>4.56e-4 - germanium<br>5.41e-4 - gallium arsenide |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| GAP2 | | 1108 | second bandgap correction factor (from Sze, beta term) |
| | | | 1108 - silicon<br>636 - silicon<br>210 - germanium<br>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 | $1/°$ | 0.0 | first order temperature coefficient for BF |
| TBF2 | $1/°^2$ | 0.0 | second order temperature coefficient for BF |
| TBR1 | $1/°$ | 0.0 | first order temperature coefficient for BR |
| TBR2 | $1/°^2$ | 0.0 | second order temperature coefficient for BR |
| TIKF1 | $1/°$ | 0.0 | first order temperature coefficient for IKF |
| TIKF2 | $1/°^2$ | 0.0 | second order temperature coefficient for IKF |
| TIKR1 | $1/°$ | 0.0 | first order temperature coefficient for IKR |
| TIKR2 | $1/°^2$ | | second order temperature coefficient for IKR |
| TIRB1 | $1/°$ | 0.0 | first order temperature coefficient for IRB |
| TIRB2 | $1/°^2$ | 0.0 | second order temperature coefficient for IRB |
| TISC1 | $1/°$ | 0.0 | first order temperature coefficient for ISC<br><br>TLEV=3 enables TISC1. |
| TISC2 | $1/°^2$ | 0.0 | second order temperature coefficient for ISC<br><br>TLEV=3 enables TISC2. |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| TIS1 | 1/° | 0.0 | first order temperature coefficient for IS or IBE and IBC |
| | | | TLEV=3 enables TIS1. |
| TIS2 | 1/°$^2$ | 0.0 | second order temperature coefficient for IS or IBE and IBC |
| | | | TLEV=3 enables TIS2. |
| TISE1 | 1/° | 0.0 | first order temperature coefficient for ISE |
| | | | TLEV=3 enables TISE1. |
| TISE2 | 1/°$^2$ | 0.0 | second order temperature coefficient for ISE |
| | | | TLEV=3 enables TISE2. |
| TISS1 | 1/° | 0.0 | first order temperature coefficient for ISS |
| | | | TLEV=3 enables TISS1. |
| TISS2 | 1/°$^2$ | 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/° | 0.0 | first order temperature coefficient for MJC |
| TMJC2 | 1/°$^2$ | 0.0 | second order temperature coefficient for MJC |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| TMJE1 | 1/° | 0.0 | first order temperature coefficient for MJE |
| TMJE2 | 1/°$^2$ | 0.0 | second order temperature coefficient for MJE |
| TMJS1 | 1/° | 0.0 | first order temperature coefficient for MJS |
| TMJS2 | 1/°$^2$ | 0.0 | second order temperature coefficient for MJS |
| TNC1 | 1/° | 0.0 | first order temperature coefficient for NC |
| TNC2 | | 0.0 | second order temperature coefficient for NC |
| TNE1 | 1/° | 0.0 | first order temperature coefficient for NE |
| TNE2 | 1/°$^2$ | 0.0 | second order temperature coefficient for NE |
| TNF1 | 1/° | 0.0 | first order temperature coefficient for NF |
| TNF2 | 1/°$^2$ | 0.0 | second order temperature coefficient for NF |
| TNR1 | 1/° | 0.0 | first order temperature coefficient for NR |
| TNR2 | 1/°$^2$ | 0.0 | second order temperature coefficient for NR |
| TNS1 | 1/° | 0.0 | first order temperature coefficient for NS |
| TNS2 | 1/°$^2$ | 0.0 | second order temperature coefficient for NS |
| TRB1 (TRB) | 1/° | 0.0 | first order temperature coefficient for RB |
| TRB2 | 1/°$^2$ | 0.0 | second order temperature coefficient for RB |
| TRC1 (TRC) | 1/° | 0.0 | first order temperature coefficient for RC |
| TRC2 | 1/°$^2$ | 0.0 | second order temperature coefficient for RC |
| TRE1 (TRE) | 1/° | 0.0 | first order temperature coefficient for RE |
| TRE2 | 1/°$^2$ | 0.0 | second order temperature coefficient for RE |
| TRM1 | 1/° | TRB1 | first order temperature coefficient for RBM |
| TRM2 | 1/°$^2$ | TRB2 | second order temperature coefficient for RBM |
| TTF1 | 1/° | 0.0 | first order temperature coefficient for TF |
| TTF2 | 1/°$^2$ | 0.0 | second order temperature coefficient for TF |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| TTR1 | $1/°$ | 0.0 | first order temperature coefficient for TR |
| TTR2 | $1/°^2$ | 0.0 | second order temperature coefficient for TR |
| TVAF1 | $1/°$ | 0.0 | first order temperature coefficient for VAF |
| TVAF2 | $1/°^2$ | 0.0 | second order temperature coefficient for VAF |
| TVAR1 | $1/°$ | 0.0 | first order temperature coefficient for VAR |
| TVAR2 | $1/°^2$ | 0.0 | second order temperature coefficient for VAR |
| TVJC | $V/°$ | 0.0 | temperature coefficient for VJC. TLEVC=1 or 2 enables TVJC to override the default HSPICE temperature compensation. |
| TVJE | $V/°$ | 0.0 | temperature coefficient for VJE. TLEVC=1 or 2 enables TVJE to override the default HSPICE temperature compensation. |
| TVJS | $V/°$ | 0.0 | temperature coefficient for VJS. TLEVC=1 or 2 enables TVJS to override the default 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 the BJT Models (NPN and PNP)

This section describes the NPN and PNP BJT models.

## Transistor Geometry — Substrate Diode

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. Vertical and lateral transistor geometries are illustrated in the following figures.
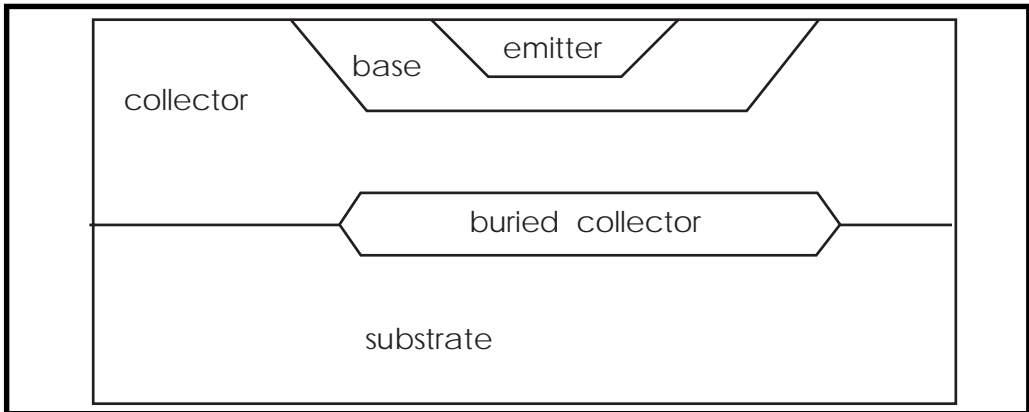


**Figure 13-7: Vertical Transistor (SUBS = +1)**
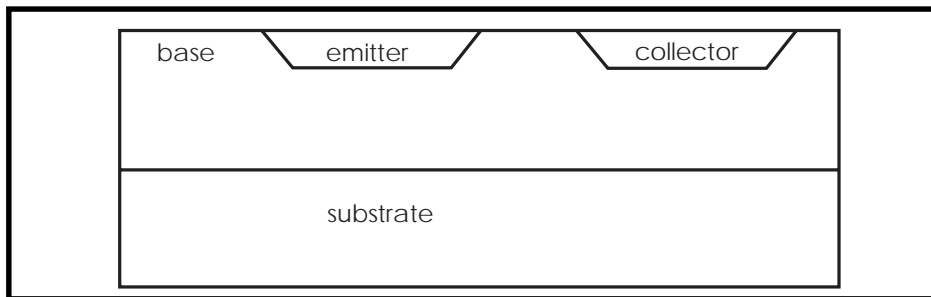


**Figure 13-8: Lateral Transistor (SUBS = -1)**

In Figure 13-10, the views from the top demonstrate how IBE is multiplied by either base area, AREAB, or collector area, AREAC.
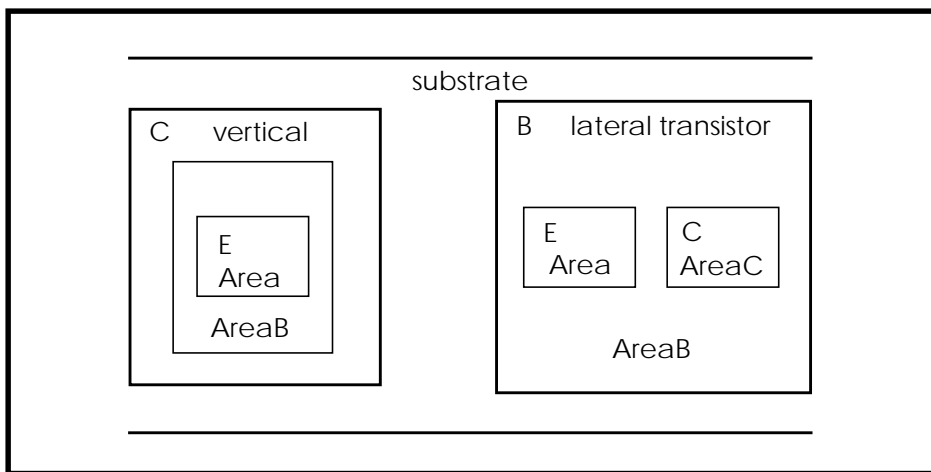


**Figure 13-9: Base, AREAB, Collector, AREAC**

# DC Model Equations

These 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:

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

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

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

**Current Equations - IBE and IBC**

If IBE and IBC are specified, instead of IS:

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

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

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

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

$$IBCeff = IBC \cdot AREAB \cdot M \text{ Vertical}$$

$$IBCeff = IBC \cdot AREAC \cdot M \text{ Lateral}$$

$$IBEeff = IBE \cdot AREA \cdot M \quad \text{Vertical or Lateral}$$

$$ISCeff = ISC \cdot AREAB \cdot M \text{ Vertical}$$

$$ISCeff = ISC \cdot AREAC \cdot M \text{ Lateral}$$

$$ISEeff = ISE \cdot AREA \cdot M \quad \text{Vertical or Lateral}$$

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.

## Substrate Current Equations

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

### Vertical Transistors

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

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

### Lateral Transistors

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

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

If both IBE and IBC are *not* specified:

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

If both IBE and IBC are specified:

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

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

## 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

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

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

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

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

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

## Variable Base Resistance Equations

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:

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

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

# Understanding BJT Capacitances

This section describes BJT capacitances.

## 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.*

### Base-Emitter Diffusion Capacitance

Determine diffusion capacitance as follows:

**ibe ≤ 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:

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

## Base-Emitter Depletion Capacitance

There are two different equations for modeling the depletion capacitance. The proper equation is selected by specification of the option DCAP in the OPTIONS statement.

### DCAP=1

The base-emitter depletion capacitance is determined as follows:

**vbe < FC . VJE**

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

**vbe ≥ FC . VJE**

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

### 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).

## 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.

### Base Collector Diffusion Capacitance

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

where the internal base-collector current ibc is:

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

### 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**

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

### DCAP=2

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

**vbc < 0**

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

**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.

## 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.

### 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

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

### 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

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

### 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.

# Modeling Various Types of Noise

Equations for modeling BJT thermal, shot, and flicker noise are as follows.

### 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 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 which 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 13-10: 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.
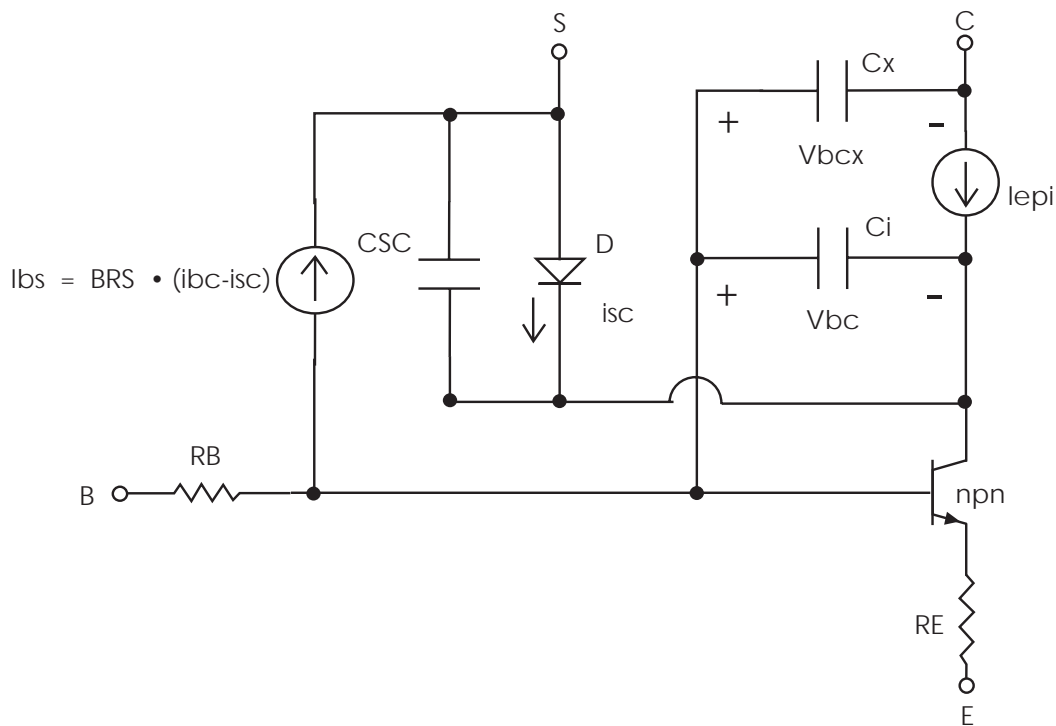
**Figure 13-10: Vertical npn Bipolar Transistor (SUBS=+1)**
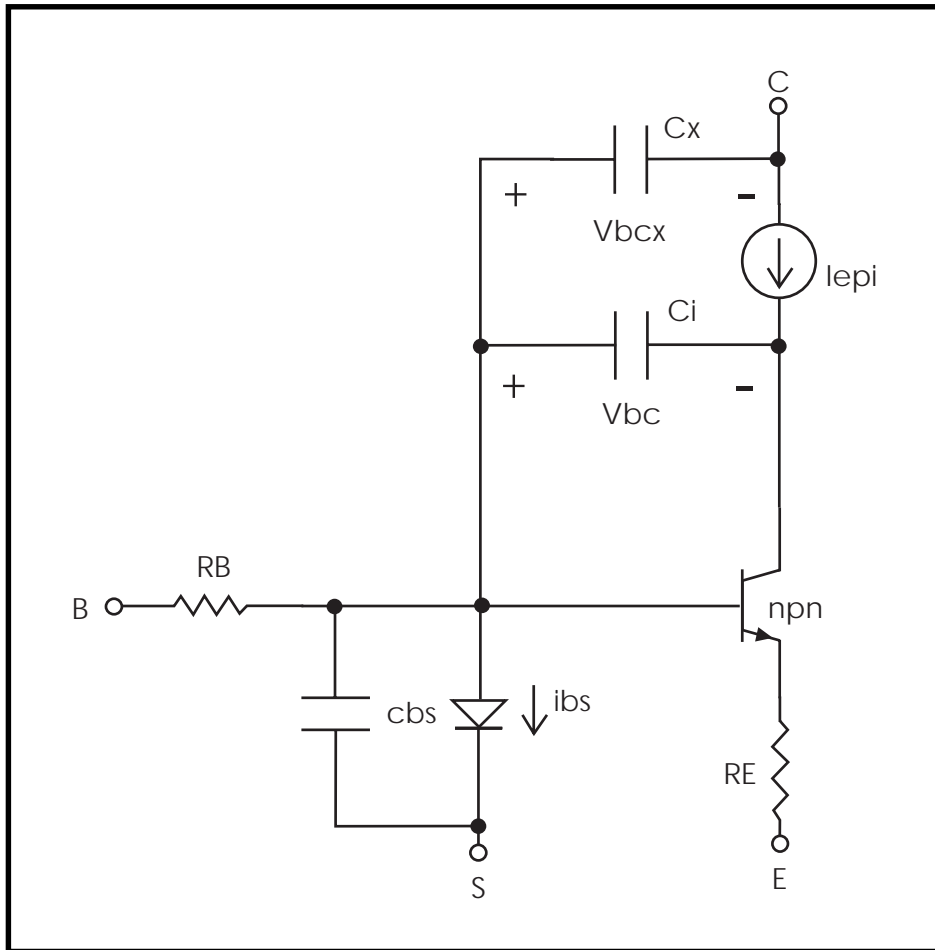
**Figure 13-11: Lateral npn Bipolar Transistor (SUBS=-1)**

## Epitaxial Current Source Iepi

The epitaxial current value, Iepi, is determined by the following 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

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

## 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 following:

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

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
```
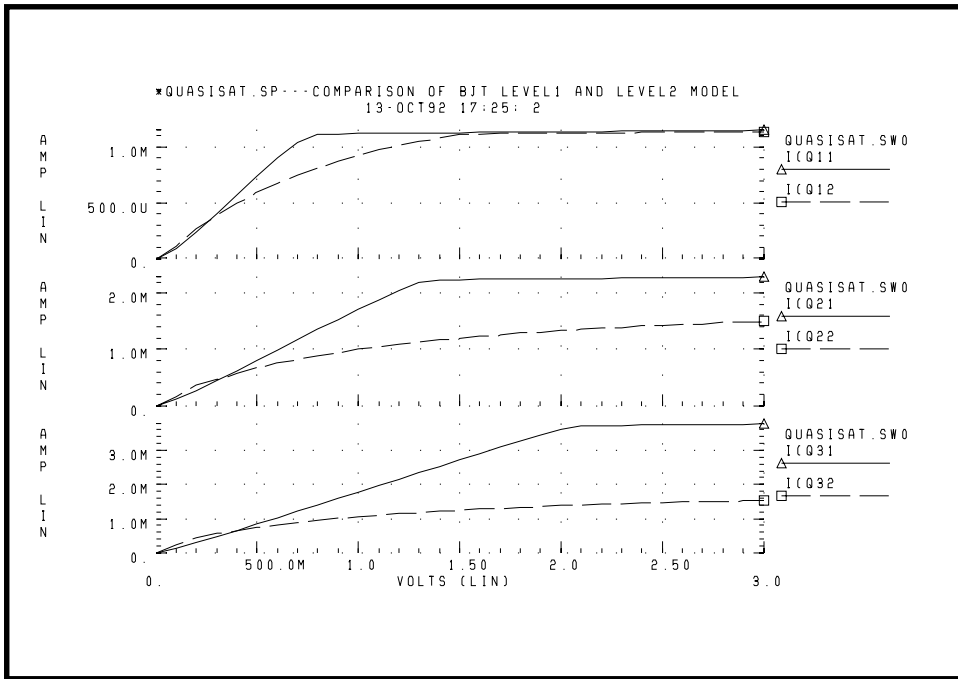


**Figure 13-12: Comparison of BJT Level 1 and Level 2 Model**

# Using Temperature Compensation Equations

This section describes how to use temperature compensation equations.

## Energy Gap Temperature Equations

To determine energy gap for temperature compensation, use the following 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**

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

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

## 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.

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

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

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

**TLEV=0, 1 or 3**

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

**TLEV=2**

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

## **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 follows:

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

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

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

## 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 follows:

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

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

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

The following parameters are also modified when corresponding temperature coefficients are specified, regardless of the TLEV value.

$$BF(t) = BF \cdot (1 + TBF1 \cdot \Delta t + TBF2 \cdot \Delta t^2)$$

$$BR(t) = BR \cdot (1 + TBR1 \cdot \Delta t + TBR2 \cdot \Delta t^2)$$

$$VAF(t) = VAF \cdot (1 + TVAF1 \cdot \Delta t + TVAF2 \cdot \Delta t^2)$$

$$VAR(t) = VAR \cdot (1 + TVAR1 \cdot \Delta t + TVAR2 \cdot \Delta t^2)$$

$$ITF(t) = ITF \cdot (1 + TITF1 \cdot \Delta t + TITF2 \cdot \Delta t^2)$$

$$TF(t) = TF \cdot (1 + TTF1 \cdot \Delta t + TTF2 \cdot \Delta t^2)$$

$$TR(t) = TR \cdot (1 + TTR1 \cdot \Delta t + TTR2 \cdot \Delta t^2)$$

$$NF(t) = NF \cdot (1 + TNF1 \cdot \Delta t + TNF2 \cdot \Delta t^2)$$

$$NR(t) = NR \cdot (1 + TNR1 \cdot \Delta t + TNR2 \cdot \Delta t^2)$$

$$NE(t) = NE \cdot (1 + TNE1 \cdot \Delta t + TNE2 \cdot \Delta t^2)$$

$$NC(t) = NC \cdot (1 + TNC1 \cdot \Delta t + TNC2 \cdot \Delta t^2)$$

$$NS(t) = NS \cdot (1 + TNS1 \cdot \Delta t + TNS2 \cdot \Delta t^2)$$

$$MJE(t) = MJE \cdot (1 + TMJE1 \cdot \Delta t + TMJE2 \cdot \Delta t^2)$$

$$MJC(t) = MJC \cdot (1 + TMJC1 \cdot \Delta t + TMJC2 \cdot \Delta t^2)$$

$$MJS(t) = MJS \cdot (1 + TMJS1 \cdot \Delta t + TMJS2 \cdot \Delta t^2)$$

## 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 for 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 for TLEV=2*

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

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

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

## Parasitic Resistor Temperature Equations

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

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

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

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

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

## BJT LEVEL=2 Temperature Equations

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

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

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

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

# Converting National Semiconductor Models

National Semiconductor's SNAP circuit simulator has a scaled BJT model that is not the same as that used by HSPICE. To use this model with 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.

## Scaled BJT Subcircuit Definition

This subcircuit definition converts the National Semiconductor scaled BJT model to a form usable in HSPICE. The .PARAM parameter inside the .SUBCKT represents the .MODEL parameter in the National circuit simulator. Therefore, the ".MODEL mname" statement must be replaced by a .PARAM statement. The model name must be changed to SBJT.

*Note:  All the parameters used in the following model must have a value which 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=110E-18 ICS=5.77E-18NE=1.02 NC=1.03
+ ME=3.61MC=1.24EG=1.12NSUB=0
+ CJE=1E-15CJC=1E-15 CSUB=1E-15EXE=0.501
+ EXC=0.222ESUB=0.709PE=1.16PC=0.37
+ PSUB=0.698 RE=75RC=0.0RB=1.0
+ TRE=2E-3 TRC=6E-3 TRB=1.9E-3VA=25
+ FTF=2.8E9 FTR=40E6 BR=1.5TCB=5.3E-3
```

```
+ TCB2=1.6E-6 BF1=9.93 BF2=45.7 BF3=55.1
+ BF4=56.5 BF5=53.5BF6=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
```

*Avant!*

# Chapter 14

# Using JFET and MESFET Models

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 have also proven 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 Element Statements
- Using JFET and MESFET Model Statements
- Generating Noise Models
- Using the Temperature Effect Parameters
- Understanding the 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, creating a Schottky diode. Both technologies control the flow of carriers by modulating the gate diode depletion region. These field effect devices are referred to as 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 HSPICE GaAs model Level=3 (See *A MESFET Model for Use in the Design of GaAs Integrated Circuits, IEEE Transactions on Microwave Theory*) assumes that GaAs device velocity saturates at very low drain voltages. The HSPICE model has been further enhanced to include drain voltage induced threshold modulation and user-selectable materials constants. These features allow use of 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 Vluume ED-34*) in 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 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* | Meta-Software 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* | Meta 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 |

**Examples**
```
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

J2  7 1 4 JM1
.MODEL JM1 NJF (VTO=-1.5, BETA=5E-3, CGS=5P, CGD=1P,
CAPOP=1  ALPHA=2)

J3  8 3 5 JX
.MODEL JX PJF (VTO=-1.2, BETA=.179M, LAMBDA=2.2M
+   CGS=100P CGD=20P CAPOP=1 ALPHA=2)
```

The first example selects the n channel MESFET model, LEVEL=3. It uses the
SAT, ALPHA, and CAPOP=1 parameter. The second example selects an n-
channel JFET and the third example selects a p-channel JFET.

---

# 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:

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, GMINDCconductance options

SCALM              model scaling option

DCCAP                 invokes capacitance calculation in DC analysis.

### Table 14-1: 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 Element Statements

The JFET and MESFET element statement contains netlist parameters for connectivity, the model reference name, dimensional geometric parameters, in addition to initialization and temperature parameters. The parameters are listed in Table 14-2.

### Table 14-2:  JFET Element Parameters

| Function | Parameter |
|---|---|
| netlist | Jxxx, mname, nb, nd, ng, ns |
| geometric | AREA, L, M, W |
| initialization | IC=(vds, vgs), OFF |
| temperature | DTEMP |

### Syntax

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

| | |
|---|---|
| *Jxxx* | JFET or MESFET element name. The name must begin with a "J" followed by up to 15 alphanumeric characters. |
| *nb* | bulk terminal node name or number |
| *nd* | drain terminal node name or number |
| *ng* | gate terminal node name or number |
| *ns* | source terminal node name or number |

| | |
|---|---|
| *mname* | model name. the name must reference a JFET or MESFET model. |
| *AREA* | the AREA multiplying factor. It affects the BETA, RD, RS, IS, CGS, and CGD model parameters. If AREA is not specified but Weff and Leff are greater than zero then: |
| | AREA=Weff/Leff, ACM=0 |
| | AREA=Weff · Leff, ACM=1 |
| | AREAeff=M · AREA |
| | Default = 1.0 |
| *W=val* | FET gate width |
| *L=val* | FET gate length |
| | Leff = L · SCALE + LDELeff |
| *OFF* | sets initial condition to OFF for this element in DC analysis. Default = ON. |
| | Weff = W · SCALE + WDELeff |
| *IC=vdsval,* | initial condition for the drain-source voltage (vdsval), or for the gate-source |
| *M=val* | Multiplier factor to simulate multiple JFETs. All currents, capacitances, and resistances are affected by M. |
| *vgsval* | voltage (vgsval). This condition can be overridden by the IC statement. |
| *DTEMP* | device temperature difference with respect to circuit temperature. Default = 0.0. |

## Examples

```
J1 7 2 3 JM1
jmes xload gdrive common jmodel
```

## 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; 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.

## JFET Current Convention

The direction of current flow through the JFET is assumed in the following diagram. Either I(Jxxx) or I1(Jxxx) syntax can be used when printing the drain current. I2 references the gate current and I3 references the source current. Jxxx is the device name.
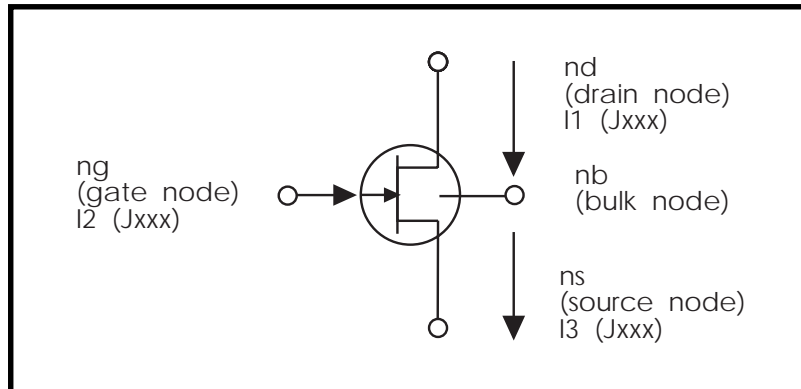


**Figure 14-1:  JFET Current Convention, N-Channel**

Figure 14-1: represents the HSPICE current convention for an n channel JFET. 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

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

$$gm = \frac{\partial(ids)}{\partial(vgs)}\bigg|_{vds\,=\,const.}$$

**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 14-2: are modeled by these ideal diode equations:

$$igd = ISeff \cdot \left(e^{\frac{vgd}{N \cdot vt}} - 1\right) \qquad vgd > -10 \cdot N \cdot vt$$

$$igd = -ISeff \qquad\qquad\qquad vgd \le -10 \cdot N \cdot vt$$

$$igs = ISeff \cdot \left(e^{\frac{vgs}{N \cdot vt}} - 1\right) \qquad vgs > -10 \cdot N \cdot vt$$

$$igs = -ISeff \qquad\qquad\qquad vgs \le -10 \cdot N \cdot vt$$
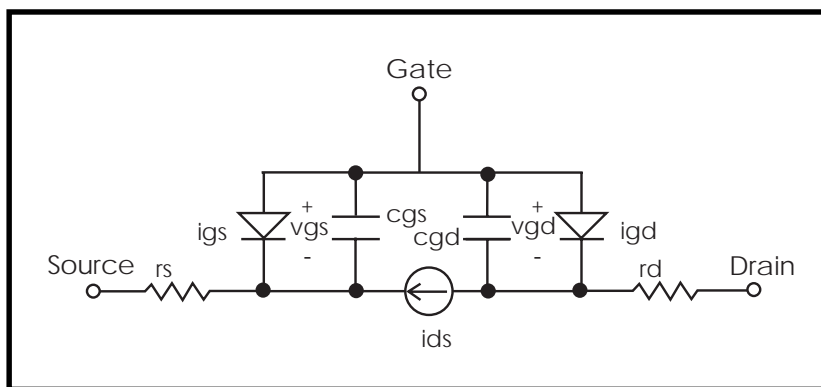


**Figure 14-2: JFET/MESFET Transient Analysis**

*Note:   For DC analysis, the capacitances are not part of the model.*
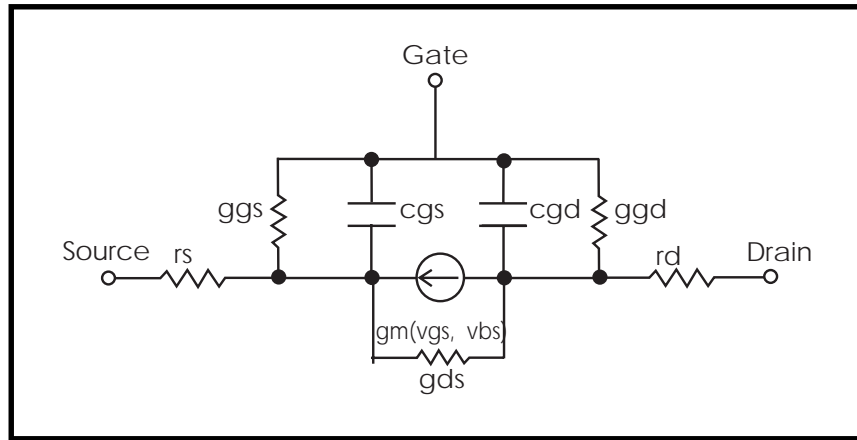
**Figure 14-3:  JFET/MESFET AC Analysis**
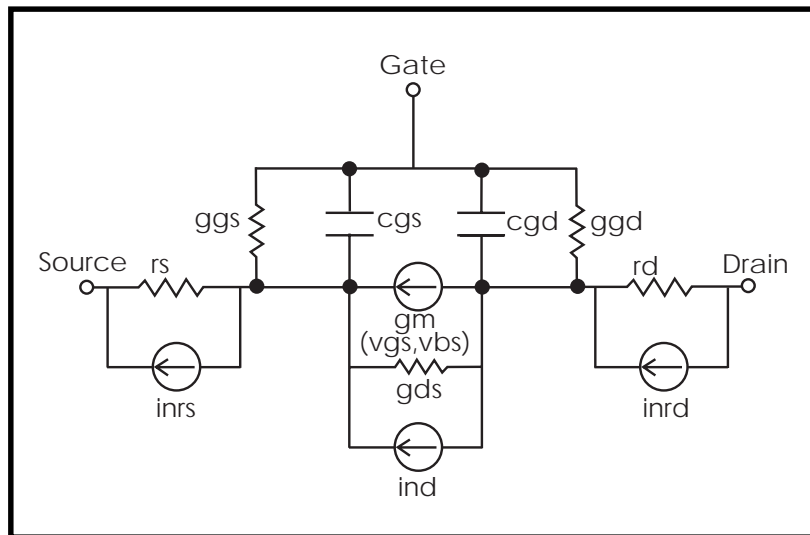


**Figure 14-4: JFET/MESFET AC Noise Analysis**

### Table 14-3: 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 |
| $\varepsilon_o$ | vacuum permittivity = 8.854e-12 F/m |
| k | 1.38062e-23 (Boltzmann's constant) |
| q | 1.60212e-19 (electron charge) |
| t | temperature in °K |
| Dt | t - tnom |

### Table 14-3:  Equation Variable Names and Constants

| Variable/<br>Quantity | Definitions |
|---|---|
| tnom | nominal temperature of parameter measurements in °K (user-input in °C). Tnom = 273.15 + TNOM |
| vt(t) | $k \cdot t/q$ |
| vt(tnom) | $k \cdot tnom/q$ |

# Using JFET and MESFET Model Statements

**General Form**

.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. Charge storage is modeled by nonlinear depletion-layer capacitances for both gate junctions which 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 14-4:   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 HSPICE area calculations (see the ACM section). If W and L are specified, AREA becomes: |
| | | | ACM=0    AREA=Weff/Leff |
| | | | ACM=1    AREA=Weff · 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 · 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 · AREAeff |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| L | m | 0.0 | default length of FET. Override this parameter using the element L. |
| | | | Leff = L · SCALM + LDELeff |
| LDEL | m | 0.0 | difference between drawn and actual or optical device length |
| | | | LDELeff = LDEL · SCALM |
| LDIF | m | 0 | distance of the lightly doped region from heavily doped region to transistor edge |
| 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 – Meta-software improvement to CAPOP=1 |
| CALPHA | ALPHA | | saturation factor for capacitance model (CAPOP=2 only) |
| CAPDS | F | 0 | drain to source capacitance for TriQuint model |

$$\text{CAPDSeff=CAPDS} \cdot \frac{\text{Weff}}{\text{Leff}} \cdot \text{M}$$

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 |
| *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)<br>　　0.50 - step junction<br>　　0.33 - linear graded junction |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| *PB* | V | 0.8 | gate junction potential |
| *TT* | s | 0 | transit time<br>– 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 $$BETAeff = BETA \cdot \frac{Weff \cdot}{Leff}$$ |
| 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 $$BETAeff = BETA \cdot \frac{Weff \cdot}{Leff}$$ |
| 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) |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| NG | | 0.0 | gate subthreshold factor (typical value=1) |
| 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 |
| | | | $Aeff = A \cdot SCALM$ |
| ALPHA | 1/V | 2.0 | saturation factor |
| BETA | amp /V $^2$ | 1.0e-4 | transconductance parameter, gain |
| | | | $$BETAeff \;=\; BETA \cdot \frac{Weff \cdot}{Leff}$$ |
| 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$^3$ | 1.552e 16 | 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 allows you to select between the SPICE unitless gate area calculations and the HSPICE area calculations. The ACM=0 method (SPICE) uses the ratio of W/L to keep AREA unitless. The ACM=1 model (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:   It is important to remember that 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  FC· PB**

$$cgd = TT \cdot \frac{\partial igd}{\partial vgd} + CGDeff \cdot \frac{1 - FC \cdot (1 + M) + M \cdot \frac{vgd}{PB}}{(1 - FC)^{M+1}}$$

**vgs  FC· PB**

$$cgs = TT \cdot \frac{\partial igs}{\partial vgs} + CGSeff \cdot \frac{1 - FC \cdot (1 + M) + M \cdot \frac{vgs}{PB}}{(1 - FC)^{M+1}}$$

### DCAP=2 (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  0**

$$cgd = TT \cdot \frac{\partial igd}{\partial vgd} + CGDeff \cdot \left(1 + M \cdot \frac{vgd}{PB}\right)$$

**vgs  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 $\cdot$ CGDeff or FC $\cdot$ CGSeff, with proper fall-off when forward bias exceeds PB (FC $\geq$ 1).

## Gate Capacitance CAPOP=1

Gate capacitance CAPOP=1 is a charge conserving symmetric capacitor model most often used for MESFET model Level 3.

$$Cgs = \frac{CGS}{4\sqrt{1 - \frac{vnew}{PB}}} \cdot \left[1 + \frac{veff - vte}{\sqrt{(veff - vte)^2 + (0.2)^2}}\right] \cdot$$

$$\left|+ \frac{vds}{\sqrt{vds^2 + \left(\frac{1}{ALPHA}\right)^2}}\right| + \left|\frac{CGD}{2} \cdot \left(1 - \frac{vds}{\sqrt{vds^2 + \left(\frac{1}{ALPHA}\right)^2}}\right)\right|$$

$$Cgd = \left(\frac{CGS}{4\sqrt{1 - \frac{vnew}{PB}}} \cdot \left[1 + \frac{veff - vte}{\sqrt{(veff - vte)^2 + (0.2)^2}}\right] \cdot \left[1 - \frac{vds}{\sqrt{vds^2 + \left(\frac{1}{ALPI}\right.}}\right.\right.$$

$$\left(\frac{CGD}{2} \cdot \left[1 + \frac{vds}{\sqrt{vds^2 + \left(\frac{1}{ALPHA}\right)^2}}\right]\right)$$

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 some 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=1implementation 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 | Units | 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 <br> 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)

The following figures show comparisons of CAPOP=1 and CAPOP=2. Note in Figure 14-5 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 14-6 how the Cgs-Cgd characteristic curve "flips over" below threshold for CAPOP=1, while for CAPOP=2, it is well-behaved.
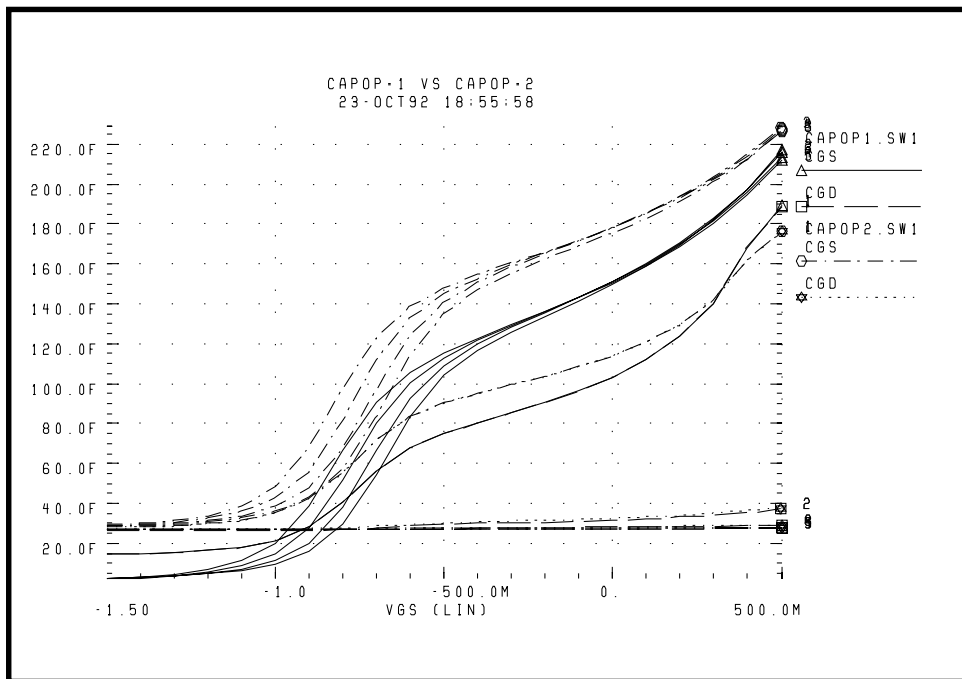
**Figure 14-5:  CAPOP=1 vs. CAPOP=2. Cgs, Cgd vs. Vgs for Vds=0, 1, 2, 3, 4**
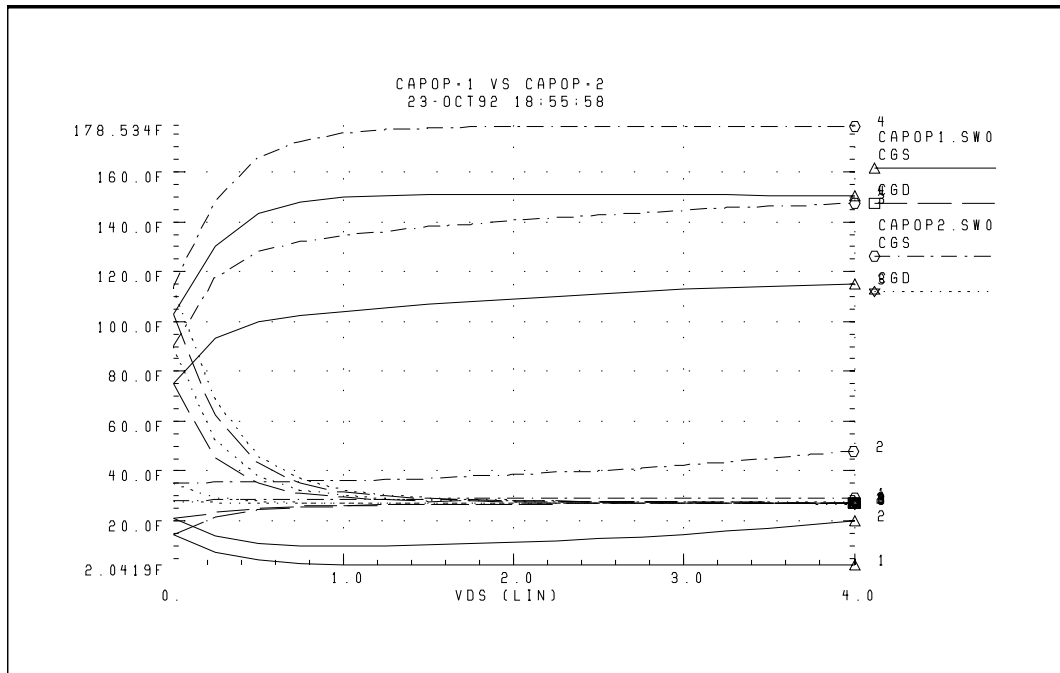
**Figure 14-6: CAPOP=1 vs. CAPOP=2. Cgs, Cgd vs. Vds for Vgs = -1.5, -1.0, -0.5, 0**

## 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 following 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 following equations:

$$vgst \ = \ vgs - VTO$$

**vgst<0** Channel pinched off

$$ids \ = \ 0$$

**0<vgst ≤ vds, vgs 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 following equations:

**vds>0**Forward region

If model parameters VP and VTO are not specified they are calculated as follows:

$$VP = -\frac{q \cdot NCHAN \cdot Aeff^2}{2 \cdot D \cdot \varepsilon_o}$$

$$VTO = VP + VBI$$

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.*

# Generating Noise Models

## 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 $V^2$ 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 14-4:. 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:

$$ind = channel\ thermal\ noise + flicker\ noise$$

If the model parameter NLEV is less than 3, then:

$$channel\ thermal\ noise = \left( \frac{8 \cdot k \cdot t \cdot gm}{3} \right)^{1/2}$$

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, 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**

$channel\ thermal\ noise\ =$

$$\left( \frac{8kt}{3} \cdot BETAeff \cdot (vgs - VTO) \cdot \frac{1 + a + a^2}{a} \cdot GDSNOI \right)^{1/2}$$

where

$$\alpha\ =\ 1 - \frac{vds}{vgs - VTO}, \quad \textit{Linear region}$$

$$\alpha\ =\ 0 \qquad\qquad \textit{Saturation region}$$

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 |

# Using the Temperature Effect Parameters

Table 14-5: 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 can be either 0, 1, or 2 while TLEVC can be either 0, 1, 2, or 3.

### Table 14-5: 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<br><br>1.17 - silicon<br>0.69 - Schottky barrier diode<br>0.67 - germanium<br>1.52 - gallium arsenide |
| GAP1 | eV/° | 7.02e-4 | first bandgap correction factor, from Sze, alpha term<br><br>7.02e-4 - silicon<br>4.73e-4 - silicon<br>4.56e-4 - germanium<br>5.41e-4 - gallium arsenide |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| GAP2 | | 1108 | second bandgap correction factor, from Sze, beta term |
| | | | 1108 - silicon<br>636 - silicon<br>210 - germanium<br>204 - gallium arsenide |
| M (MJ) | | 0.50 | grading coefficient for gate-drain and gate-source diodes |
| | | | 0.50 - step junction<br>0.33 - linear graded junction |
| N | | 1.0 | emission coefficient for gate-drain and gate-source diodes |
| TCV (VTOTC) | 1/° | 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/° | 0.0 | temperature coefficient for PB. TLEVC=1 or 2 overrides the default temperature compensation. |
| TRD (TDR1) | 1/° | 0.0 | temperature coefficient for drain resistance |
| TRG (TRG1) | 1/° | 0 | temperature coefficient for gate resistance |
| TRS (TRS1) | 1/° | 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 following 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**

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

**TLEV=2**

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

### 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 follows:

$$BETA(t) = BETA \cdot \left(\frac{t}{tnom}\right)^{BEX} \qquad \textit{If BETATCE=0}$$

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 following equations:

$$RD(t) \; = \; RD \cdot (1 + TRD \cdot \Delta t)$$

$$RS(t) \; = \; RS \cdot (1 + TRS \cdot \Delta t)$$

$$RG(t) \; = \; RG \cdot (1 + TRG \cdot \Delta t)$$

# Understanding the 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 Cicuit Simulation in SPICE'*.

There are a few differences from the original implementation. The 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 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 Names**

Table 14-6 shows the Hspice alias printout names.

**Table 14-6: Alias Hspice Printout Names**

| | | |
|---|---|---|
| Q | VGEXP | |
| GAMMA | GAMDS<br>original | sign opposite of TriQuint's |
| VTOTC | TCV<br>original | sign opposite of TriQuint's |
| TRG1 | TRG | |
| TRD1 | TRD | |
| TRS1 | TRS | |

## TOM Model Parameters

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| *BETATCE* | | | temperature coefficient for BETA<br>If betatce is set to a nonzero value: |

$$BETA(temp) \ = \ BETA(tnom) \cdot 1.01^{(BETATCE \cdot (}$$

The more common HSPICE Beta temperature update is:

$$BETA(temp) \ = \ BETA(tnom) \cdot \left(\frac{temp}{tnom}\right)^{BE}$$

*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 15

# Introducing MOSFET

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, refer to Chapter 16, "Selecting a MOSFET Model".

The following topics are covered in this chapter:

- Understanding MOSFET Models
- Selecting Models
- Using Nonplanar and Planar Technologies
- Using a MOSFET Diode Model
- Using MOS Diode Equations
- Using Common Threshold Voltage Equations
- Performing MOSFET Impact Ionization
- 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, more detailed models, such as the Level 6 IDS model or one of the BSIM models (Level 13, 39, or 49) can be used.

For precision modeling of integrated circuits, the BSIM models take into account 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 take into account 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. Level 6 can be used 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**

```
M3 3 2 1 0 PCH <parameters>
.MODEL PCH PMOS LEVEL=13 <parameters>
```

The above 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.

## 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 HSPICE.

Not all MOSFET models are available in the PC version of HSPICE, Table 15-1 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 | |
| 4 | Grove-Frohman: Level 2 model derived from SPICE 2E.3 | X | |

| Level | MOSFET Model Description | All Platforms including PC | All Platforms except PC |
|-------|-------------------------|---------------------------|-------------------------|
| 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 |
| 24 ** | GE-Intersil advanced | | X |
| 25 ** | CASMOS (Rutherford) | | X |
| 26 ** | Sierra 2 | | X |

| Level | MOSFET Model Description | All Platforms including PC | All Platforms except PC |
|-------|-------------------------|-------------------------|------------------------|
| 27 | SOSFET | | X |
| 28 | BSIM derivative; Meta-software 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 |
| 46 *** | SGS-Thomson MOS Level 3 | | X |
| 47 | BSIM3 Version 2.0 | | X |
| 49 | BSIM3 Version 3 | | |
| 50 | Philips MOS9 | | |

* not officially released
** equations are proprietary – no documentation will be provided
*** requires a license and equations are proprietary – no documentation will be provided

## MOSFET Capacitor Selection

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*         BSIM 2 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 15-1: CAPOP=4 Selections

| MOS Level | Default CAPOP | CAPOP=4 selects: |
|-----------|---------------|-------------------|
| 2 | 2 | 11 |
| 3 | 2 | 12 |

### Table 15-1: CAPOP=4 Selections

| MOS Level | Default CAPOP | CAPOP=4 selects: |
|-----------|---------------|------------------|
| 13, 28, 39 | 13 | 13 |
| others | 2 | 11 |

The proprietary models, as well as Level 5, 17, 21, 22, 25, 31, 33, and the SOS model Level 27, have their own built-in capacitance routines.

## MOS Diode Selection

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* | META 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, and HSPICE uses the MOSFET model parameters to select the appropriate model for the given width and length.

The 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, you must 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, for example, 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 illustrates calling 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
```

## 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=0. |

| | |
|---|---|
| *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.

## Unit Scaling

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 only applies 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 15-2: 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.

# MOSFET Element Syntax

This section describes the MOSFET element syntax.

## General form

    Mxxx nd ng ns <nb> mname <L=val> <W=val> <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

    Mxxx nd ng ns <nb> mname lval wval …

or

    .OPTION WL
    Mxxx nd ng ns <nb> mname wval lval …

| | |
|---|---|
| *Mxxx* | MOSFET element name. The name must begin with an "M" followed by up to 15 alphanumeric characters. |
| *ng* | gate terminal node name |
| *ns* | source terminal node name |
| *nb* | bulk terminal node name Can be set by BULK parameter in model statement. |
| *nd* | drain terminal node name |
| *mname* | model name reference |
| | **Note:** If the model name includes a period (.), the HSPICE automatic model selector does not work properly for that model. Do not use periods in model names if you intend to use the automatic model selector. |
| *L* | channel length. This option overrides DEFL in OPTIONS statement. Default=DEFL.<br>    Lscaled = L · SCALE. The maximum value of Lscaled is 0.1 m. |

| | |
|---|---|
| *W* | channel width. This option overrides DEFW in OPTIONS statement. Default= DEFW.<br>Wscaled = W · SCALE |
| *AD* | drain diffusion area. Overrides DEFAD in the OPTIONS statement. Default=DEFAD only when ACM=0. (See "Using a MOSFET Diode Model" for effective ADeff). |
| *AS* | source diffusion area. Overrides DEFAS in the OPTIONS statement. Default=DEFAS only when ACM=0. (See "Using a MOSFET Diode Model" for effective ASeff). |
| *PD* | perimeter of the drain junction, including the channel edge. Overrides DEFPD in OPTIONS statement. ACM=0 and ACM=1 Default=DEFPD. ACM=2, 3 Default=0.0 (See "Using a MOSFET Diode Model"). |
| *PS* | perimeter of the source junction, including the channel edge. Overrides DEFPS in OPTIONS statement. ACM=0 and ACM=1 Default=DEFPD. ACM=2, 3 Default=0.0 (See "Using a MOSFET Diode Model"). |
| *NRD* | number of squares of drain diffusion for resistance calculations. Overrides DEFNRD in .OPTIONS statement. ACM=0 and ACM=1: default=DEFNRD. ACM=2: default=0.0 (see "Using a MOSFET Diode Model"). |
| *NRS* | number of squares of source diffusion for resistance calculations. Overrides DEFNRS in .OPTIONS statement. ACM=0 and ACM=1: default=DEFNRS. ACM=2, 3: default=0.0 (see "Using a MOSFET Diode Model"). |
| *RDC* | additional drain resistance due to contact resistance. (Units are ohm;<br>Default = 0.0)<br>**Note:** A value assigned for RDC in the element statement overrides any value for RDC as a model parameter. |
| *RSC* | additional source resistance due to contact resistance. (Units are ohm; Default=0.0) |

|  | **Note:** A value assigned for RSC in the element statement overrides any value for RSC as a model parameter. |
|---|---|
| *OFF* | sets the initial condition to OFF for this element in the DC analysis, or for the first timepoint in the transient analysis. Default=ON.<br>**Note:** This command does not work for depletion devices. |
| *M* | multiple device option. MOSFET channel width, diode leakage, capacitors, and resistors are altered by this parameter. Simulates multiple parallel devices. Default=1.0. |
| *vbs* | initial condition for the voltage across the external bulk and source terminals. Overridden by the IC statement. |
| *vds* | initial condition for the voltage across the external drain and source terminals. Overridden by the IC statement. |
| *vgs* | initial condition for the voltage across the external gate and source terminals. Overridden by the IC statement. |
| *DTEMP* | device temperature difference from circuit temperature. Default=0.0. |
| *GEO* | source/drain sharing selector for ACM=3. Default=0.0 (see ACM=3 section). |
| *DELVTO* | zero-bias threshold voltage shift. Default=0.0. |

*Note:   SCALE defaults to 1.0 meter. To enter parameter PD=val with units in microns, for example, set SCALE to 1e-6. Then if PD=5 is entered, HSPICE sets PD=5e-6 meters, or 5 microns.*

**Examples**

```
M1 24 2 0 20 TYPE1

M31 2 17 6 10 MODM L=5U W=2U
M31 2 16 6 10 MODM 5U 2U
Or
```

```
 .OPTION WL
M31 2 16 6 10 MODM 2U 5U

M1 2 9 3 0 MOD1 L=10U W=5U AD=100P AS=100P PD=40U
PS=40U
M1 2 9 3 0 MOD1 10U 5U 2P 2P
```

The first example specifies a MOSFET element connected between nodes 24, 2, 0, and 20. It calls a MOSFET model statement which has a model reference name called TYPE1. The .OPTION WL reverses the order of the width and length parameters in the MOSFET element statement.

The element statement parameters previously listed are summarized below. You can specify the geometric parameters, except for M, in the options statements. Element parameter values always override .OPTION or .MODEL parameter settings.

### Table 15-3: MOSFET Element Parameters

| Function | Parameters |
|----------|-----------|
| geometric | AD, AS, L, M, PD, PS, W |
| initialization | IC=Vds, Vgs, Vbs, OFF |
| netlist | Mxxx, nd, ng, ns, nb, mname |
| resistance | NRD, NRS, RDC, RSC |
| temperature | DTEMP |

### Table 15-4: Variables and Constants.

| Variable | Definition |
|----------|-----------|
| cbd | bulk to drain capacitance |
| cbs | bulk to source capacitance |
| cbg | gate to bulk capacitance |

**Table 15-4: Variables and Constants.**

| Variable | Definition |
|----------|------------|
| cgd | gate to drain capacitance |
| cgs | gate to source capacitance |
| f | frequency |
| gbd | bulk to drain dynamic conductance |

## Equation Variables

This section lists the equation variables and constants.

**Table 15-5: Equation Variables and Constants**

| Variable/Quantity | Definition |
|-------------------|------------|
| 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 |
| inrs | source resistor equivalent noise circuit |
| rd | drain resistance |
| rs | source resistance |

## **Table 15-5: Equation Variables and Constants**

| Variable/Quantity | Definition |
|---|---|
| vsb | source to bulk voltage |
| vds | drain to source voltage |
| vgs | gate to source voltage |
| $\Delta t$ | 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 °K |
| tnom | tnom = TNOM + 273.15. This variable represents the nominal temperature of parameter measurements in °K (user input in °C). |
| vt | $k \cdot t / q$ |
| vt(tnom) | $k \cdot tnom / q$ |

### **MOSFET Current Convention**

Figure 15-1: 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 15-1: MOSFET Current Convention, N-channel**

## MOSFET Equivalent Circuits

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 15-2 through Figure 15-4 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 15-2:  Equivalent Circuit, MOSFET Transient Analysis**

**Figure 15-3: Equivalent Circuit, MOSFET AC Analysis**

**Figure 15-4: Equivalent Circuit, MOSFET AC Noise Analysis**

# 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.

## Field Effect Transistor

The metal gate MOSFET is a nonisoplanar metal-oxide-semiconductor field effect transistor as illustrated in Figure 15-5 and Figure 15-6.

**Figure 15-5:  Field Effect Transistor**

Looking at the actual geometry, from source-to-drain, Figure 15-6 shows a perspective of the nonisoplanar metal-oxide semiconductor field effect transistor.



**Figure 15-6:  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 15-7.) The cut A-B shows a drain contact (Figure 15-8).



**Figure 15-7: Isoplanar Silicon Gate Transistor**

**Figure 15-8:  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 15-9), which includes the contacts.

**Figure 15-9:  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 15-10).

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 15-10:  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 |

## General MOSFET Model Statement

This is the general form for all model specifications. All related parameter levels are covered in their respective sections.

### General form

```
.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. |

### Examples

```
.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 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.

## MOSFET Diode Model Selection

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 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 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.

## 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. Use of these parameters also enhances the convergence properties of the diode model.

# 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<br>JSscaled = JS/SCALM$^2$<br>– for ACM=1, unit is amp/m and<br>JSscaled = JS/SCALM. |
| JSW | amp/m | 0 | sidewall bulk junction saturation current<br>JSWscaled = JSW/SCALM. |
| IS | amp | 1e-14 | bulk junction saturation current. For the option<br>ASPEC=1, default=0. |
| N | | 1 | emission coefficient |
| NDS | | 1 | reverse bias slope coefficient |
| VNDS | V | -1 | reverse diode current transition point |

## Capacitance Model Parameters

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| CBD | F | 0 | zero bias bulk-drain junction capacitance. Used only<br>when CJ and CJSW are 0. |
| CBS | F | 0 | zero bias bulk-source junction capacitance. Used only<br>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$^2$<br>– for ACM=1 the unit is F/m and CJscaled = CJ/SCALM<br>– default for option ASPEC=0 is<br><br>$$CJ = \left(\frac{\varepsilon si \cdot q \cdot NSUB}{2 \cdot PB}\right)^{1/2}$$ |
| CJSW (CJP) | F/m | 0 | zero-bias sidewall bulk junction capacitance CJSWscaled = CJSW/SCALM<br>– default = 0 |
| CJGATE | F/m | CSJW | zero-bias gate-edge sidewall bulk junction capacitance<br>(ACM=3 only)<br>CJGATEscaled=CJGATE/SCALM<br>Default = CJSW for HSPICE releases later than H9007D.<br>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 |

## 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 ACM 1. |
| 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 ACM 1. |
| 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 |

### 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) <br><br> HDIFwscaled = HDIF · SCALM |
| LD (DLAT,LATD) | m | | lateral diffusion into channel from source and drain diffusion. <br> If LD and XJ are unspecified, LD default=0.0. <br> When LD is unspecified, but XJ is specified, LD is calculated from XJ. LD default=0.75 · XJ. <br> For Level 4 only, lateral diffusion is derived from LD·XJ. <br> LDscaled = LD · SCALM |
| LDIF | m | 0 | length of lightly doped diffusion adjacent to gate (ACM=1, 2) <br> LDIFscaled = LDIF · SCALM |
| WMLT | | 1 | width diffusion layer shrink reduction factor |
| XJ | m | 0 | metallurgical junction depth <br> XJscaled = XJ · SCALM |
| XW (WDEL, DW) | m | 0 | accounts for masking and etching effects <br> XWscaled = XW · SCALM |

## ACM=0 MOS Diode

The following listing illustrates typical parameter value settings for a MOSFET diode that is designed with a MOSFET that has a channel length of 3 μm and a channel width of 10 μm.

**Figure 15-11: – ACM=0 MOS Diode**

**Example**

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) |

## Effective Areas and Peripheries Calculations

For ACM=0, the effective areas and peripheries are calculated as follows:

$$Deff = M \cdot AD \cdot WMLT^2 \cdot SCALE$$

$$ASeff = M \cdot AS \cdot WMLT^2 \cdot SCALE^2$$

$$PDeff = M \cdot PD \cdot WMLT \cdot SCALE$$

$$PSeff = M \cdot PS \cdot WMLT \cdot SCALE$$

## Effective Saturation Current Calculations

For ACM=0, 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,

$$isbd = M \cdot IS$$

### Drain Diode Saturation Current

Define:

$$val = JSscaled \cdot ADeff + JSWscaled \cdot PDeff$$

If val > 0 then,

$$isbd = val$$

Otherwise,

$$isbd = M \cdot IS$$

## Effective Drain and Source Resistances

For ACM=0, the effective drain and source resistances are calculated as follows:

### 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,

$$RDeff = \frac{RD + RDC}{M}$$

## ACM=1 MOS Diode

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 15-12:  ACM=1 MOS Diode**

### Example

The listings below are typical parameter value settings for a transistor with

LD=0.5 μmW=10 μmL=3 μmLDIF=0.5 μm

| | |
|---|---|
| *CJ* | 1e-10 F/m of gate width<br>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<br>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 |

### Effective Areas and Peripheries Calculations

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.*

### Effective Saturation Current Calculations

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,

$$isbd = M \cdot IS$$

**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 $\geq$ 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.*

## ACM=2 MOS Diode

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 15-13:  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) |

## Effective Areas and Peripheries Calculations

For ACM=2, the effective areas and peripheries are calculated as follows:

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.*

### Effective Saturation Current Calculations

For ACM=2, 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,

$$isbd = M \cdot IS$$

## Effective Drain and Source Resistances

For ACM=2, the effective drain and source resistances are calculated as follows.

### 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,

$$RDeff \ = \ \frac{RDC}{M} + \frac{HDIFeff \cdot RSH + (LDscaled + LDIFscaled) \cdot RD}{Weff}$$

## ACM = 3 MOS Diode

The ACM=3 is used to model MOS diodes of the stacked devices properly. Also, 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 15-14: – Stacked Devices and Corresponding GEO Values**

## Effective Areas and Peripheries Calculations

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

$$|gsbs| = |isbs|, \text{ and } |gdbd| = |isbd|$$

## MOS Diode Capacitance Equations

Each MOS diode capacitance is the sum of diffusion and depletion capacitance. The diffusion capacitance is evaluated in both 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.

The bias-dependent depletion capacitance must be calculated 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,

$$
vbs \; = \; TT \cdot \frac{\partial ibs}{\partial vbs} + C0BS \cdot \left( 1 - \frac{vbs}{PB} \right)^{-\mathit{u}}
$$

$$
C0BS\_SW \cdot \left( 1 - \frac{vbs}{PHP} \right)^{-MJSW}
$$

For vbs > 0,

$$
vbs \; = \; TT \cdot \frac{\partial ibs}{\partial vbs} + C0BS \cdot \left( 1 + MJ \cdot \frac{v_{\mathit{t}}}{P} \right.
$$

$$
C0BS\_SW \cdot \left( 1 + MJSW \cdot \frac{vbs}{PHP} \right.
$$

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,

$$\text{)}bs \ = \ TT \cdot \frac{\partial ibs}{\partial vbs} + M \cdot CBS \cdot \left(1 + MJ \cdot \frac{vl}{P}\right)$$

## 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,

$$\text{)}bd \ = \ TT \cdot \frac{\partial ibd}{\partial vbd} + C0BD \cdot \left(1 + MJ \cdot \frac{vl}{P}\right)$$
$$C0BD\_SW \cdot \left(1 + MJSW \cdot \frac{vbd}{PHP}\right)$$

Otherwise, if (ADeff · CJscaled + PDeff · CJSWscaled) ≤ 0, then:
 For vbd < 0,

$$apbd \ = \ TT \cdot \frac{\partial bd}{\partial vbd} + M \cdot CBD \cdot \left(1 - \frac{vbd}{PB}\right)^{-m}$$

For vbd > 0,

$$apbd \ = \ TT \cdot \frac{\partial ibd}{\partial vbd} + M \cdot CBD \cdot \left(1 + MJ \cdot \frac{vb}{Pl}\right)$$

# 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 | $cm^3$ | | 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 |

## Calculation of 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 $\Phi ms$ is determined by:

$$\Phi ms = -\frac{eg}{2} - type \cdot \frac{PHI}{2} - 0.05$$

where type is +1 for n-channel and -1 for p-channel.

For Poly-Gate (TPG=±1), the work function is determined by:

If the model parameter NGATE is not specified,

$$\Phi ms \ = \ type \cdot \left( - \ TPG \cdot \frac{eg}{2} - \frac{PHI}{2} \right)$$

Otherwise,

$$\Phi ms \ = \ type \cdot \left[ - \ TPG \cdot vt \cdot \ln\left( \frac{NGATE \cdot 1e6}{ni} \right) - \frac{PHI}{2} \right]$$

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·I_impact
I_db = I_db_diode + (1-IIRAT)·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.

## 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 |

## 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

$$ALPHAeff = ALPHA + LALPHA \cdot 1e\text{-}6 \cdot \left( \frac{1}{Leff} - \frac{1}{LREFeff} \right)$$

$$WALPHA \cdot 1e\text{-}6 \cdot \left( \frac{1}{Weff} - \frac{1}{WREFe} \right)$$

$$VCReff = VCR + LVCR \cdot 1e\text{-}6 \cdot \left( \frac{1}{Leff} - \frac{1}{LREFeff} \right)$$

$$WVCR \cdot 1e\text{-}6 \cdot \left( \frac{1}{Weff} - \frac{1}{WREFej} \right)$$

where

$$LREFeff = LREF + XLREF - 2 \cdot LD \text{ and}$$

$$WREFeff = WREF + XWREF - 2 \cdot WD$$

## Effective Output Conductance

The element template output allows gds to be output directly, for example,

```
.PRINT I(M1) gds=LX8(M1)
```

However, 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 15-15:  Drain, Source, and Bulk Currents for vgs=3, with IIRAT=0.5**

## Cascode Example

Drain to bulk impact ionization current limits the use of cascoding to increase output impedance. The following cascode example shows the affect of changing IIRAT. When IIRAT is less than 1.0, the drain to bulk current lowers the output impedance of the cascode stage.



**Figure 15-16: 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
```

# MOS Gate Capacitance Models

Capacitance model parameters can be used 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 piece-wise linear model for all MOS levels. The charge conservation model is also available for MOSFET model Levels 2, 3, 4, 5, 6, 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.

## Capacitor Model Selection

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). Others 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* | BSIM 2 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 15-6: 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.

## Introduction to 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

$$\frac{dQ1}{dV1} + \frac{dQ2}{dV1} = 0$$

$$\frac{dQ1}{dV1} + \frac{dQ1}{dV2} = 0$$

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 15-17: MOS Capacitances**

In Figure 15-17:, 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 the following:

$$CGG = \frac{dQg}{dVG}$$

$$CGD = -\frac{dQg}{dVD}$$

$$CDG = -\frac{dQD}{dVG}$$

CGG represents input capacitance: a change in gate voltage requires a current equal to CGG×dVG/dt into the gate terminal. CGD represents Miller feedback: a change in drain voltage gives a current equal to CGG×dVG/dt out of the gate terminal. CDG represents Miller feedthrough, capacitive current out of the drain due to a change in gate voltage.

To see how CGD might not be equal to CDG, the following example presents a simplified model with no bulk charge, with gate charge a function of VGS only, and 50/50 partition of channel charge into QS and QD:

$$QG = Q(vgs)$$

$$QS = -0.5 \cdot Q(vgs)$$

$$QD = -0.5 \cdot Q(vgs)$$

$$QB = 0$$

As a result of this:

$$CGD = -\frac{dQG}{dVD} = 0$$

$$CGD = -\frac{dQD}{dVG} = 0.5 \cdot \frac{dQ}{dvgs}$$

Therefore, in this model there is Miller feedthrough, but no feedback.

## 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 QD and so on 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 |

## 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 that is 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 LX18= CGG, LX19= -CGD, and so forth, 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
```

## Example Gate Capacitance Calculation

This example shows a gate capacitance calculation in detail for a BSIM model. TOX is chosen so that

$$\frac{eox}{tox} = 1e-3F/m^2$$

Vfb0, phi, k1 are chosen so that vth=1v. The AC sweep is chosen so that $2 \cdot \pi \cdot freq = 1e6s^{-1}$ for the last point.

### 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

$$Leff = 0.6u$$

$$\frac{eox}{tox} = 1e-3F/m^2$$

$$Cap = \frac{Leff \cdot Weff \cdot eox}{tox} = 60e - 15F$$

BSIM equations for internal capacitance in saturation with xqc=0.4:

$$body = 1 + 0.5 \cdot \left(1 - \frac{1}{(1.744 + 0.8364 \cdot (PHI0 + vsb))}\right) \cdot \frac{K1}{\sqrt{(PHI0 + vsb)}}$$

$$1 + 0.5 \cdot \left(1 - \frac{1}{(1.744 + 0.8364)}\right) = 1.30$$

$$cgg = Cap \cdot \left(1 - \frac{1}{(3 \cdot body)}\right) = Cap \cdot 0.7448 = 44.69F$$

$$cgd = 0$$

$$cdg = \left(\frac{4}{15}\right) \cdot Cap = 16F$$

$$cdd = 0$$

$$\text{Gate-drain overlap} = (ld + meto) \cdot Weff \cdot \frac{eox}{tox} = 20e - 15F$$

Adding the overlaps,

$$cgg = 44.69F + 2 \cdot 20F = 84.69F$$

$$cgd = 20F$$

$$cdg = 36F$$

$$cdd = 20F$$

$$\text{Drain-bulk diode cap } cj \cdot ad = (0.5e - 4) \cdot (200e - 12) = 1$$

Adding the diodes,

$$cgg = 84.69F$$

$$cgd = 20F$$

$$cdg = 36F$$

$$cdd = 30F$$

### 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 HSPICE results match.

## Plotting Gate Capacitances

This input file shows how to plot gate capacitances as a function of bias. The .OPTION DCCAP needs to be set to turn on capacitance calculations for a DC sweep. The model used is the same as for the above 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 15-18: Gate Capacitance**

## 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 "CAPOP=3 — Gate Capacitances (Simpson Integration)" on page 15-90). 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.

# MOS Gate Capacitance Model Parameters

## 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:<br>COXscaled = COX/$SCALM^2$ |
| TOX | m | 1e-7 | represents the oxide thickness, calculated from COX when COX is input. Program uses default if COX is not specified. For TOX>1, unit is assumed to be Angstroms. There can be a level-dependent default that overrides. |

## 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.<br>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.<br>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.<br>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.<br><br>LDscaled = LD · SCALM<br><br>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<br><br>WDscaled = WD · SCALM |

### 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 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) |

### Charge Conservation Parameter, 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. |

## XQC & XPART Specification for CAPOP=4, 9, 11, 12 and 13

Parameter rule for gate capacitance charge sharing coefficient, XQC & XPART, in the saturation region:

- If neither XPART or XQC is specified, the 0/100 model is used.
- If both XPART and XQC are specified, XPART overrides XQC.
- If XPART is specified:

XPART=0 → 40/60

XPART=0.4 → 40/60

XPART=0.5 → 50/50

XPART=1 → 0/100

XPART = any other value less than 1 → 40/60

XPART >1 → 0/100If XQC is specified:

XQC=0 → 0/100

XQC=0.4 → 40/60

XQC=0.5 → 50/50

XQC=1 → 0/100

XQC = any other value less than 1 → 40/60

XQC>1 → 0/100

The only difference is the treatment of the parameter value 0.

After XPART/XQC is specified, the gate capacitance is ramped from 50/50 at Vds=0 volt (linear region) to the value (with Vds sweep) in the saturation region specified by XPART/XQC. This charge sharing coefficient ramping will assure the smoothness of the gate capacitance characteristic.

## 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,

$$CGBOeff \ = \ M \cdot Leff \cdot CGBOscaled$$

Otherwise,

$$CGBOeff \ = \ 2 \cdot WDscaled \cdot Leff \cdot COXscaled \cdot M$$

### Gate to Source Overlap Capacitance

If CGSO is specified,

$$CGSOeff \ = \ Weff \cdot CGSOscaled$$

Otherwise,

$$CGSOeff \ = \ Weff \cdot (LDscaled + METOscaled) \cdot COXscaled$$

### Gate to Drain Overlap Capacitance

If CGDO is specified,

$$CGDOeff \ = \ Weff \cdot CGDOscaled$$

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.

## 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, $vgs \leq vth - \dfrac{PHI}{2}$**

$$cgs = 0$$

**Depletion, vgs $\leq$ vth**

$$cgs = CF5 \cdot cap + \frac{cap \cdot (vgs - cth)}{0.75 \cdot PHI}$$

**Strong Inversion Saturation Region, vgs > vth and vds $\geq$ 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 15-19: CAPOP=0 Capacitances**

## 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 $\geq$ *0.1*)**

**Accumulation, vgs $\leq$ vth**

$$cgd \;=\; CF5 \cdot cap \cdot G^- \cdot D^+$$

**Saturation Region, vgs < vth + vds**

$$cgd \;=\; CF5 \cdot cap \cdot D^+$$

**Strong Inversion, vgs $\geq$ 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 15-20: CAPOP=1 Capacitances**

## **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 $\leq$ 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 $\geq$ *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 $\neq$ 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 15-21: CAPOP=2 Capacitances**

## 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 often intractable or 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:

$$q(v) \ = \ \int\limits_{0}^{v} C(v) dv$$

For the calculation of current:

$$i(t) \ = \ \frac{dq(v)}{dt} \ = \ \left(\frac{d}{dt}\right)\int\limits_{0}^{v} C(v) dv$$

For small intervals:

$$I(n+1) \ = \ \frac{dq(v)}{dt} \ = \ \frac{1}{t(n+1) - t(n)} \int\limits_{V(n)}^{V(n+1)} C(v) dv$$

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. The charge can be estimated accurately by using 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. The OPTIONS settings RELV (relative voltage tolerance), RELMOS (relative current tolerance for MOSFETs), and CVTOL (capacitor voltage tolerance) might have to be modified. The default of 0.5 is a good nominal value for CVTOL. The option CVTOL sets the number of integration steps with the formula:

$$n \ = \ \frac{|V(n + 1) - V(n)|}{CVTOL}$$

The effect of using a large value for CVTOL is to decrease 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, in some instances severely.

## 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 HSPICE the above 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 HSPICE an empirical equation is used to make the transition from 50/50 (linear region) to 40/60 (saturation region) smoothly.

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 which is invoked by setting CAPOP=4.

In the following example the charge conservation capacitances CAPOP=4 and the improved charge conservation capacitance CAPOP=9 for the model Level 3 only is 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 15-22:  CAPOP=4, 9 Capacitances for Level 3 Model**

**Figure 15-23: 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 15-24: Charge Pump Circuit**

**Figure 15-25: 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 becomes zero when the input pulse goes to zero. Consequently, the model that provides voltage closer to zero for node 20 conserves 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 HSPICE. The results indicate that 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 15-26: Charge Conservation Test Circuit**

## CAPOP=5 — Gate Capacitance

Use CAPOP=5 for no capacitors, and HSPICE will not calculate gate capacitance.

## 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 · (vth + vfb) > vgs

$$cgs = 0$$

### 0.5 · (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 = $ *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**

$$vc = vth - vfb + \frac{1}{2} \cdot vds + vsb$$

## CAPOP=13 — BSIM 1-based Charge-Conserving Gate Capacitance Model

See "Level 13 BSIM Model" on page 16-104.

## CAPOP=39 — BSIM2 Charge-Conserving Gate Capacitance Model

See "Level 39 BSIM2 Model" on page 16-183.

## Effective Length and Width for AC Gate Capacitance Calculations

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, model parameters LDAC and WDAC can be substituted for LD and WD in Leff and Weff calculations. LD and WD are still used 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 HSPICE parameters XL, LD, XW, and WD. They should not be used with other parameters such as DL0 and DW0.

# Using Noise Models

This section describes how to use noise models.

## 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) |

## Noise Equations

The HSPICE MOSFET model noise equations have a selector parameter NLEV that is used to select either the original SPICE flicker noise or an equation proposed by Gray and Meyer.

Thermal noise generation in the drain and source resistors is modeled by the two sources inrd and inrs (units amp/(Hz)$^{1/2}$), as shown in Figure 15-4. 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, 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

$$a = 1 - \frac{vds}{vdsat} \qquad \textit{Linear region}$$

$$a = 0 \qquad\qquad \textit{Saturation region}$$

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$^2$· (channel thermal noise)$^2$* |
| *FN, V$^2$/Hz* | output flicker noise: *FN = RX$^2$· (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 HSPICE temperature compensation. |
| CTP | 1/°K | 0.0 | junction sidewall capacitance CJSW temperature coefficient. Set TLEVC to 1 to enable CTP to override default 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.<br><br>1.17 – silicon<br>0.69 – Schottky barrier diode<br>0.67 – germanium<br>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<br>4.73e-4 – silicon<br>4.56e-4 – germanium<br>5.41e-4 – gallium arsenide |
| GAP2 | °K | 1108 | second bandgap correction factor (from Sze, beta term) |
| | | | 1108 – silicon<br>636 – silicon<br>210 – germanium<br>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 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 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 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. |
| | | | **W**hen 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/°K | 0.0 | temperature coefficient for drain resistor |
| TRS | 1/°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. |

## 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 modelling for various device sizes. The default value of each sensitivity parameter is zero to ensure backward compatibility.

**Table 15-7:**

| 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 |

# Temperature Equations

This section describes how to use temperature equations.

## Energy Gap Temperature Equations

To determine energy gap for temperature compensation use the following 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:**

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

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

## 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 15-29.

## 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:**

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

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

**and TLEV=2:**

$$dpbdt = -\frac{\left[egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left(2 - \dfrac{tnom}{tnom + GAP2}\right) - PB\right]}{tnom}$$

$$dphpdt = -\frac{\left[egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left(2 - \dfrac{tnom}{tnom + GAP2}\right) - PHP\right]}{tnom}$$

## 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 follows:

$$PHI(t) = 2 \cdot vt(t) \cdot \ln\left(\frac{NSUB}{ni}\right)$$

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 for TLEV=0 or 1:**

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

**and for TLEV=2:**

$$dphidt = -\frac{\left[egnom + 3 \cdot vt(tnom) + (EG - egnom) \cdot \left(2 - \dfrac{tnom}{tnom + GAP2}\right) - PHI\right]}{tnom}$$

## 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:**

$$VTO(t) = VTO + \left(1 + \frac{GAMMA}{2 \cdot PHI^{1/2}}\right) \cdot dphidt \cdot \Delta t$$

$$vbi(t) = VTO(t) - GAMMA \cdot (PHI(t))^{1/2}$$

## Mobility Temperature Equations

The MOS mobility temperature equations are:

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

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

$$F1(t) = F1 \cdot \left(\frac{t}{tnom}\right)^{F1EX}$$

## Channel Length Modulation Temperature Equation

The LAMBDA is modified with temperature if model parameter LAMEX is specified.

$$LAMBDA(t) = LAMBDA \cdot (1 + LAMEX \cdot \Delta t)$$

## Diode Resistance Temperature Equations

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 16

# Selecting a MOSFET Model

Now that you know more about MOSFET models from Chapter 15, "Introducing MOSFET." it will be easier for you to choose which type of models you require for your needs.

This chapter lists the 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
- Level 47 BSIM3 Version 2 MOS Model
- Level 49 BSIM3 Version 3 MOS Model
- Level 50 Philips MOS9 Model
- Comparing MOS Models

# 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=$0.75 \cdot XJ$ |
| | | | $LD_{scaled} = LD \cdot SCALM$ |
| 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| XJ | m | 0.0 | metallurgical junction depth: $XJ_{scaled} = XJ \cdot SCALM$ |
| XL (DL, (LDEL) | m | 0.0 | accounts for masking and etching effects: $XL_{scaled} = XL \cdot SCALM$ |
| *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 15-52). |
| *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 15-52). |
| *VTO (VT)* | V | | zero-bias threshold voltage. If not specified, it is calculated. (See "Common Threshold Voltage Parameters" on page 15-52). |

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, $v_{gs} \le v_{th}$

$$I_{ds} = 0.0$$

### 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 15-52 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: |
| | | | $ECRIT = 100 \cdot (VMAX / UO)$ |
| 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 |
| | | | $KP = UO \cdot COX$ |
| 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 15-52). |
| 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$^{-2}$·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 15-52). |
| VTO(VT) | V | | zero-bias threshold voltage. If it is not specified, it is calculated (see "Common Threshold Voltage Parameters" on page 15-52). |
| 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. |
| THETA | V$^{-1}$ | 0.0 | mobility modulation. THETA is used only when MOB=7. A typical value in this application is THETA=5e-2. |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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) | low-field bulk mobility. This parameter is calculated from KP if KP is input. |
| | | 250 (P) | |
| UTRA | | 0.0 | transverse field coefficient |

> **Note:** SPICE does not use UTRA. 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 $\eta$. To include the narrow width effect, specify the model parameter DELTA. The short-channel effect is included through the effective $\gamma$. To include short-channel effects, the model parameter XJ must be greater than zero. Then:

$$\gamma = GAMMA \cdot \left\{ 1 - \frac{XJ_{scaled}}{2 \cdot L_{eff}} \cdot \left[ \left( 1 + \frac{2 \cdot W_s}{XJ_{scaled}} \right)^{1/2} + \left( 1 + \frac{2 \cdot W_d}{XJ_{scaled}} \right)^{1/2} - 2 \right] \right\}$$

The depletion widths, $W_s$ and $W_d$, are determined by:

$$W_s = \left[ \frac{2 \cdot E_{si}}{q \cdot NSUB} \cdot (PHI + v_{sb}) \right]^{1/2}$$

$$W_d = \left[ \frac{2 \cdot E_{si}}{q \cdot NSUB} \cdot (PHI + v_{ds} + v_{sb}) \right]^{1/2}$$

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 15-52).

## 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 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 $\lambda$ if you do not specify the model parameter LAMBDA.

**LAMBDA>0**

$$\lambda = LAMBDA$$

**VMAX>0, NSUB >0, and LAMBDA$\leq$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 $\leq$ 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, HSPICE limits the value of Le as follows:

If Le < xwb, then Le is replaced by:

$$\frac{xwb}{1 + \frac{xwb - Le}{xwb}}$$

where:

$$xwb = X_d \cdot PB^{1/2}$$

## Subthreshold Current, $I_{ds}$

This region of operation is characterized by the fast surface states model parameter, NFS. For NFS>0 the model determines the modified threshold voltage (von) as follows:

$$von = v_{th} + fast$$

where:

$$fast = vt \cdot \left[ \eta + (PHI + v_{sb})^{1/2} \cdot \frac{\partial \gamma}{\partial v_{sb}} + \frac{\gamma}{2 \cdot (PHI + v_{sb})^{1/2}} + \frac{q \cdot NFS}{COX} \right]$$

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<br><br>$DEL_{scaled} = DEL \cdot SCALM$ |
| LD (DLAT, (LATD) | m | | lateral diffusion into channel from source and drain diffusion,<br><br>If LD and XJ are unspecified, LD Default= 0.0.<br><br>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<br><br>$LREF_{scaled} = LREF \cdot SCALM$ |
| LMLT | | 1.0 | length shrink factor |
| WD | m | 0.0 | lateral diffusion into channel width from bulk<br><br>$WD_{scaled} = WD \cdot SCALM$ |
| 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| WREF | m | 0.0 | channel width reference |
| | | | $\text{WREF}_{scaled} = \text{WREF} \cdot \text{SCALM}$ |
| XJ | m | 0.0 | metallurgical junction depth |
| | | | $\text{XJ}_{scaled} = \text{XJ} \cdot \text{SCALM}$ |
| XL (DL, LDEL) | m | 0.0 | length bias accounts for masking and etching effects |
| | | | $\text{XL}_{scaled} = \text{XL} \cdot \text{SCALM}$ |
| XW (DW, WDEL) | m | 0.0 | width bias accounts for masking and etching effects |
| | | | $\text{XW}_{scaled} = \text{XW} \cdot \text{SCALM}$ |

## 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 15-52). |
| 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. 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 15-52). |
| VTO (VT) | V | | zero-bias threshold voltage. This parameter is calculated if not specified (see "Common Threshold Voltage Parameters" on page 15-52). |
| WIC | | 0.0 | sub-threshold model selector |
| WND | $\mu m/V$ | 0.0 | ND width sensitivity |
| WN0 | $\mu m$ | 0.0 | N0 width sensitivity |

## Mobility Parameters

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| THETA | $V^{-1}$ | 0.0 | mobility degradation factor |
| UO (UB,UBO) | $cm^2/(V \cdot s)$ | 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, $v_{gs} \le v_{th}$

$$I_{ds} = 0 \qquad \textit{(See subthreshold current)}$$

#### 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, 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, HSPICE computes them (see "Common Threshold Voltage Parameters" on page 15-52).

### Saturation Voltage, $v_{dsat}$

For the Level 3 model, 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 ($\Delta L$) 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:

$$I_{ds} = \frac{I_{ds}}{1 - \frac{\Delta L}{L_{eff}}}$$

In order to prevent the denominator from going to zero, HSPICE limits the $\Delta L$ value as follows:

If

$$\Delta L > \frac{L_{eff}}{2}$$

then

$$\Delta L = L_{eff} - \frac{\left( \frac{L_{eff}}{2} \right)^2}{\Delta L}$$

## 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$_{gs}$ von**

$$I_{ds} = I_{ds}(v_{gs}, vde, v_{sb})$$

*Note:   The model does not use the modified threshold voltage in strong inversion.*

If WIC=3, the model calculates subthreshold current differently. In this case, the I$_{ds}$ 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 -117).

N0$_{eff}$ and ND$_{eff}$ are functions of effective device width and length.

# Compatibility Notes

This section describes compatibility issues.

## HSPICE versus SPICE3

Differences between HSPICE and Berkeley SPICE3 can arise in the following situations:

## Small XJ

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. HSPICE imposes the limit $f_s \le 1.0$, while SPICE3 allows $f_s > 1.0$.

## ETA

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 HSPICE, multiply ETA by 815/814.

## NSUB Missing

When NSUB is missing in SPICE3, the KAPPA equation becomes inactive. In 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 HSPICE model.

### LD Missing

If LD is missing, HSPICE uses the default 0.75·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 $k = 1.3806226e\text{-}23 J \cdot K^{-1}$

Electron charge    $e = 1.6021918e\text{-}19 C$

Permittivity of silicon dioxide $\varepsilon_{ox} = 3.45314379969e\text{-}11 F/m$

Permittivity of silicon $\varepsilon_{si} = 1.035943139907e\text{-}10 F/m$

## Example of 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,

$$beta = 1.2e-3 \cdot (1.251551)^{-1.5} = 0.570545e-4$$

$$v_{tm} = 0.8 - (1.5e - 3) \cdot 75 = 0.6875$$

$$egarg = 9.399920 \qquad vtherm = 3.215466e - 2$$

$$phi(t) = 0.64 \cdot 1.251551 - 0.3238962 = 0.4770964$$

$$I_{ds} = beta \cdot 0.5 \cdot \frac{(2 - vt)^2}{1 + \dfrac{0.2}{\sqrt{phi(t)}}} = 5.724507e - 4$$

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 "Using Common Threshold Voltage Equations" on page 15-52).

# 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| UB (UO) | $cm^2/$ (V·s) | 0.0 | low field bulk mobility |
| 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 HSPICE MOS5 and is identical to AMI SPICE MOS Level 4.

*ZENH=0*         This depletion model is revised in HSPICE (from previous depletion mode) and is identical to AMI SPICE MOS Level 5.

The HSPICE enhancement and depletion models are basically identical to the AMI models. However, certain aspects have been revised to enhance performance. Using the HSPICE enhancement and depletion models provides access to HSPICE features as described below.

The 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 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 (HSPICE). All MOSFET models follow this convention.

The 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, $v_{gs} \leq v_{th}$

$$I_{ds} = 0 \qquad \text{(See "" on page -42)}$$

### On Region, $v_{gs} \ v_{th}$

$$I_{ds} = \beta \cdot \left\{ \left( v_{gs} - v_{bi} - \frac{vde}{2} \right) \cdot vde - \frac{2}{3} \cdot \gamma \cdot [(\Phi_f + vde + v_{sb})^{3/2} - (\Phi_f + v_{sb})^{3/2}] \right\}$$

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 HSPICE program computes $\gamma_0$ using DNB and ignores the GAMMA model parameter.*

The effective body effect ($\gamma$), including the device size effects, is computed as follows.

$$\gamma = \gamma_0 \cdot (1 - scf) \cdot (1 + ncf)$$

If SCM $\leq 0$,

$$scf = 0$$

otherwise,

$$scf = \frac{XJ}{L_{eff}} \cdot \left\{ \left[ 1 + \frac{2xd}{XJ} \cdot (SCM \cdot v_{ds} + v_{sb} + \Phi_f)^{1/2} \right]^{1/2} - 1 \right\}$$

If NWM $\leq 0$,

$$ncf = 0$$

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:

$$UB_{eff} = \frac{1}{\dfrac{1}{UB} + \dfrac{FRC \cdot (v_{gs} - v_{th})}{TOX} + \dfrac{vde}{VST \cdot L_e} + FSB \cdot v_{sb}^{1/2}}$$

where

$$L_e = L_{eff} \qquad \textit{linear region}$$

$$L_e = L_{eff} - \Delta L \; \textit{saturation region}$$

The channel length modulation effect ($\Delta L$) is defined in the following section.

## Channel Length Modulation

The Level 5 model includes the channel length modulation effect by modifying the $I_{ds}$ current as follows:

$$I_{ds} = \frac{I_{ds}}{1 - \dfrac{\Delta L}{L_{eff}}}$$

where

$$\Delta L = 1e4 \cdot \left[ \frac{2.73e3 \cdot XJ}{DNB \cdot \ln\left(\dfrac{1e20}{DNB}\right)} \right]^{1/3} \cdot [(v_{ds} - v_{dsat} + PHI)^{1/3} - PHI^{1/3}]$$

The $\Delta L$ is in microns, assuming XJ is in microns and DNB is in 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$_{fb}$  vde >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 $\gamma$ 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 $\gamma$, including small device size effects, is computed as follows:

$$\gamma = \gamma_0 \cdot (1 - scf) \cdot (1 + ncf)$$

where

**If SCM $\leq$0,**

$$scf = 0$$

otherwise,

$$scf = \frac{XJ}{L_{eff}} \cdot \left\{ \left[ 1 + \frac{2xd}{XJ} \cdot (SCM \cdot v_{ds} + v_b + \Phi_d)^{1/2} \right]^{1/2} - 1 \right\}$$

**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), 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 $\Delta L$ parameter is in microns, assuming XJ is in microns and na1 is in cm$^{-3}$.

## Subthreshold Current, $I_{ds}$

When device leakage currents become important for operation near or below the normal threshold voltage, the subthreshold characteristics are considered. The 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 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, $v_{gs}$-$v_{fb}$ 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 of Level 5 Model

```
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 I$_{DS}$: 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, -11 and Channel Length Modulation, -16 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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. |
| 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·s) | 600 (N) | This parameter is the low field bulk mobility. It is calculated from KP if KP is supplied. |
| | | 250 (P) | |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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. |
| 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 |
| 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 |

| 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 |
| | | | 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$^{-2}$·V$^{-1}$ | 0.0 | fast surface state density |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 15-52). |
| WEX | | | weak inversion exponent |
| 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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, described below.

### UPDATE=0

This is the original Level 6 model in 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 16-88.

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 cm2/(V · s) (N-ch)
UTRA (F3)= 0.5
```

### UPDATE=0 then,

```
TOX     = 1000
UO (UB)= 600 cm2/(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:

$$vde = \min\left(\frac{vds}{vfa}, vsat\right), UPDATE = 1 \ or \ 2$$

$$vde = \min(vds, vfa \cdot vsat), \ UPDATE = 0$$

The saturation voltage in the impact ionization equation is as follows:

$$vdsat = vfa \cdot vsat, \quad UPDATE= 1 \ or \ 2$$

$$vdsat = vsat, \quad\quad UPDATE= 0$$

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 $\gamma$ is computed differently depending on the specified model parameters.

### Single-Gamma, VBO=0

When model parameter VBO is zero, the single-gamma model is used. In this case the model treats the parameter LGAMMA as a junction depth. It then modifies the GAMMA parameter for short-channel effect by the scf factor, which is computed using the Poon and Yau formulation. In this case LGAMMA is multiplied by the SCALM option.

$$scf = 1 - \frac{LGAMMA}{leff} \cdot \left\{ \left[ 1 + \frac{2 \cdot LAMBDA}{LGAMMA} \cdot (PHI + vsb)^{1/2} \right]^{1/2} - 1 \right\}$$

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 $\gamma$, 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: HSPICE calculates model parameters such as VTO, PHI, and GAMMA, if they are not user-specified (see "Common Threshold Voltage Parameters" on page 15-52).*

## 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 16-1 shows the variation of vth as a function of vsb for Multi-Level Gamma.



**Figure 16-1: 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 $\gamma i$ and $\gamma b$ are effective values of GAMMA and LGAMMA, respectively. The model computes them as $\gamma$ 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 $\leq$VFDS,**

if vsb≤VBO,

$$vbi = VTO - \gamma i \cdot (PHI)^{1/2} + (\eta - 1) \cdot (PHI + vsb) - \frac{LDscaled}{Leff} \cdot VSH - \frac{\varepsilon si}{COX \cdot Leff} \cdot FDS$$

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,

$$vbi = VTO - \gamma b \cdot (PHI)^{1/2} + (\gamma i - \gamma b) \cdot [(VBO + PHI)^{1/2} - (PHI)^{1/2}] + (\eta - 1)$$

$$\cdot PHI + vsb - \frac{LDscaled}{Leff} \cdot VSH - \frac{\varepsilon si}{COX \cdot Leff} \cdot [(FDS - UFDS) \cdot VFDS + U$$

**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, $\gamma$ is switched from $\gamma i$ to $\gamma b$, the ids, vsat, and conductances are not continuous. This problem is demonstrated in the following example. To correct the discontinuity problem, specify model parameter UPDATE=1. The next section discusses this improvement.

### Example of 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 16-2: Variation of IDS, VTH and VDSAT for UPDATE=0**

**Figure 16-3: 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 vsb is less than VBO, $\gamma$ is set to $\gamma i$ and used in ids 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 $\gamma b$ must be used instead of $\gamma i$ in vsat computation. Since vsat = vgs - vth (drain), the threshold voltage at drain is computed using $\gamma i$ 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 of 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 16-4: Variation of IDS, VTH and VDSAT for UPDATE=2**

**Figure 16-5: 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 $\gamma i$ and $\gamma b$ 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 $\gamma i$ and $\gamma b$, 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:

$$le \; = \; \min(vds, vfa \cdot vsa, \; UPDATE\text{=}0$$

$$ds \; = \; \min(vds/vfa, vsa, \; UPDATE\text{=}1,2$$

## 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  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 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 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 16-75. 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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.

$$factor = \cfrac{1}{1 + \cfrac{COX}{F1 \cdot \varepsilon ox} \cdot (vgs - cth) + \cfrac{vde}{vc} + F2 \cdot (vsb + PHI)^{1/2}}$$

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:

$$ids = \cfrac{ids}{1 + F1 \cdot \left(vgs - vth - \cfrac{vde}{2}\right) + \cfrac{UTRA}{Leff} \cdot vde}$$

### 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. 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 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 | Meta-Software's channel length modulation |
| CLM = 6 | Meta-Software'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 |

$$L = vfu^{(2 \cdot MCL)} \cdot LAMBDA \cdot$$

$$[(vds - vfa \cdot vsat + KCL \cdot vsb + PHI)^{1/2} - (KCL \cdot vsb + PHI)^{1/2}]$$

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 16-75 for definitions of vfa and vfu.

### CLM=4, Wang's Equation

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| A1 | m | 0.2 | junction depth |
| | | | A1scaled = A1   SCALM |
| DND | cm$^{-3}$ | 1e20 | drain diffusion concentration |

### Linearly-Graded Depletion Layer

$$\Delta L = \left[ \frac{2.73e5 \cdot A1scaled}{DNB \cdot \ln\!\left(\dfrac{DND}{DNB}\right)} \right]^{1/3} \cdot [(vds - vdsat + PHI)^{1/3} - PHI^{1/3}]$$

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, 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, HSPICE ∆L 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 (∆L) 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. 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, 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 | $F/m^2$ | 3.45314 e-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·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 15-52). |
| LND | $\mu$m/V | 0.0 | ND length sensitivity |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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) |
| WIC | | 0.0 | sub-threshold model selector |
| WND | μm/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 15-52). |
| VTO(VT) | V | | zero-bias threshold voltage. This parameter is calculated if not specified (see "Common Threshold Voltage Parameters" on page 15-52). |

## 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 $V^{-1}$. |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| UEXP (F2) | | 0.0 | critical field exponent in mobility degradation |
| UTRA | m/V | 0.0 | transverse field coefficient (mobility) |
| UO (UB, UBO) | $cm^2$/ (V·s) | 600 (N) | low field bulk mobility. This parameter is calculated from KP (BETA) if KP (BETA) is input. |
| | | 250 (P) | |

## 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 $\gamma$, $\Phi$, 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 $\gamma$ value is adjusted for short-channel effect the same way as the Level 2 model. Also, $\Phi$ 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 $\gamma$ is modified for short-channel effect, the same as in the Level 2 model, to get effective $\gamma$.

### 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 $\leq$ 0 or MOB=7,

$$vdsat \; = \; vsat$$

*vsat* is computed as in the Level=2 model (See page -15).

### 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 $\qquad (vgs - vth) > \dfrac{\varepsilon_{si} \cdot UCRIT}{COX}$

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,
$$u_{eff} = \frac{UO}{1 + \dfrac{UTRA}{L_{eff}} \cdot vde}$$

**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**

$$\Delta L = \cfrac{L_{eff}}{1 + \cfrac{(1 + LAM1 \cdot L_{eff}) \cdot (1 + vde)^{A1}}{LAMBDA \cdot (vds - vde)}}$$

and

$$ids = \cfrac{ids}{1 - \cfrac{\Delta L}{L_{eff}}}$$

## 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 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 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 HSPICE capacitor model (from capacitor element) simulates capacitors generated with interconnect. The 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 HSPICE, the keyletter for the MOSFET device is S for SPICE BSIM and M for HSPICE. (Refer to the example of 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 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 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 | $\mu$m | 0.0 | difference between drawn poly and electrical |
| DW0 | $\mu$m | 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 |
| WETA | $\mu$m | 0.0 | width sensitivity |
| K1 | $V^{1/2}$ | 0.5 | root-vsb threshold coefficient |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 | $\mu m$ | 0.0 | length sensitivity |
| WK2 | $\mu m$ | 0.0 | width sensitivity |
| MUS | $cm^2/(V \cdot s)$ | 600 | high drain field mobility |
| LMS (LMUS) | $\mu m \cdot cm^2/(V \cdot s)$ | 0.0 | length sensitivity |
| WMS (WMUS) | $\mu m \cdot cm^2/(V \cdot s)$ | 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 |
| WMUZ | $\mu m \cdot cm^2/(V \cdot s)$ | 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 |
| WNB | | 0.0 | width sensitivity |
| ND0 | | 0.0 | Vds reduction to low field weak inversion gate drive coefficient |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 |
| 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| WX2M (WX2MZ) | $\mu m \cdot cm^2 / (V^2 \cdot s)$ | 0.0 | width sensitivity |
| X2MS | $cm^2 / (V^2 \cdot s)$ | 0.0 | Vbs reduction to high drain field mobility |
| LX2MS | $\mu m \cdot cm^2 / (V^2 \cdot s)$ | 0.0 | length sensitivity |
| WX2MS | $\mu m \cdot cm^2 / (V^2 \cdot s)$ | 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 |
| 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 |
| 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 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 |
| 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 |
| 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 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/°K | 0.0 | flat-band voltage temperature coefficient |
| TREF | °C | 25 | temperature at which parameters are extracted. This parameter defaults to the option *TNOM*, which defaults to 25 °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 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.

**Examples:**

$$VFB0 \ = \ -0.350v$$

$$LVFB \ = \ -0.1v\mu$$

$$WVFB \ = \ 0.08v \cdot \mu$$

$$Leff \ = \ 1 \cdot 10^{-6}m \ = \ 1\mu$$

$$Weff \ = \ 2 \cdot 10^{-6}m \ = \ 2\mu$$

$$LREFeff \ = \ 2 \cdot 10^{-6}m \ = \ 2\mu$$

$$WREFeff \ = \ 1 \cdot 10^{-5}m \ = \ 10\mu$$

$$zvfb \; = \; VFB0 + LVFB \cdot \left( \frac{1}{Leff} - \frac{1}{LREFeff} \right) + WVFB \cdot \left( \frac{1}{Weff} - \frac{1}{WREFeff} \right)$$

$$zvfb \; = \; -0.35v + -0.1v \cdot \mu \cdot \left( \frac{1}{1\mu} - \frac{1}{2\mu} \right) + 0.08v \cdot \mu \cdot \left( \frac{1}{2\mu} - \frac{1}{10\mu} \right)$$

$$zvfb \; = \; -0.35v - 0.05v + 0.032v$$

$$zvfb \; = \; -0.368v$$

## .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 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*

9007Blevel 13 BSIM model introduced: no changes

9007Dremoves the K2 limit

92Achanges the TOX parameter default from 1000 A to 200 A

92Badds the K2LIM parameter, which specifies the K2 limit

93Aintroduces gds constraints

93A.02VERSION parameter introduced

95.1fixes nonprinting TREF and incorrect GMBS problems

96.1Flatband 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 $\leq$ 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 $\geq$ 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

$$xu1 = zu1 - zx2u1 \cdot vsb + zx3u1 \cdot (vds - VDDM), \quad UPDATE=2$$

$$xu1 = \frac{zu1 - zx2u1 \cdot vsb + zx3u1 \cdot (vds - VDDM)}{Leff}, \quad UPDATE=0, 1$$

## 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 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 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 $\leq$ vtho, vgs $\leq$ zvfb - vsb**

$$Qg \ = \ cap \cdot (vgs - zvfb + vsb)$$

$$Qb \ = \ -qg$$

$$Qs \ = \ 0$$

$$Qd \ = \ 0$$

**Subthreshold Region, vgs $\leq$ vtho, vgs > zvfb - vsb**

$$= \ \frac{cap \cdot zk1}{2} \cdot \left\{ [(zk1)^2 + 4(vgs - zvfb + vsb)]^{1/2} - zk \right.$$

$$Qb \ = \ -qg$$

$$Qs \ = \ 0$$

**50/50 Channel-Charge Partitioning for Drain and Source, XPART=.5**

**Triode Region, vgs > vtho, vds $\leq$ 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 $\le$ 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

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.

## Example Calculations Using Level 13 Equations

To verify the equations, it is helpful to do very simple tests using 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 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 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 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.

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/HSPICE Parameter Differences

A cross-reference table for UCB's BSIM1 and Meta-Software's Level 13 model parameters is provided for comparison. Units are given in brackets. The HSPICE parameter name is given only if it differs from the SPICE name. The model specifies units for HSPICE parameters only if they differ from SPICE's. HSPICE aliases are in parentheses. Note that some HSPICE aliases match the SPICE names.

An asterisk (*) in front of a UCB SPICE name denotes an incompatibility between the HSPICE name and the UCB SPICE name (that is, the HSPICE alias does not match, or units are different).

Even when there is a difference in parameter name between 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 16-1: – Comparison of HSPICE Parameters with UCB SPICE 2 and 3

| UC Berkeley SPICE 2, 3 | Meta-Software HSPICE |
| --- | --- |
| VFB [V] | VFB0 (VFB) |
| PHI [V] | PHI0 |
| K1 [$V^{1/2}$] | same |
| K2 | same |

## **Table 16-1: – Comparison of HSPICE Parameters with UCB SPICE 2 and 3**

| UC Berkeley SPICE 2, 3 | Meta-Software 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 [°C] | TREF |
| * VDD [V] | VDDM |
| CGDO [F/m] | CGDOM (CGDO) |

## Table 16-1: – Comparison of HSPICE Parameters with UCB SPICE 2 and 3

| UC Berkeley SPICE 2, 3 | Meta-Software HSPICE |
|---|---|
| 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 HSPICE, there are defaults for the parameters.

### Parasitics

ACM > 0 invokes HSPICE parasitic diodes. ACM=0 (default) is SPICE style.

## Temperature Compensation

The model reference temperature TNOM's default is 25˚C in HSPICE unless ".OPTION SPICE" is set, causing TNOM to default to 27˚C. This option also sets some other SPICE compatibility parameters. 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, HSPICE Level 13 does provide for temperature analysis. The default analysis temperature is 25˚C in 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 HSPICE netlist to change the HSPICE analysis temperature.

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.

### Example of 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=-5
VB=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 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, 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.*

**General form**

```
.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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 |
| 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). |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 of NMOS and PMOS SOSFETs

```
*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 HSPICE for SOS/SOI applications, several approaches are currently available. 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 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.
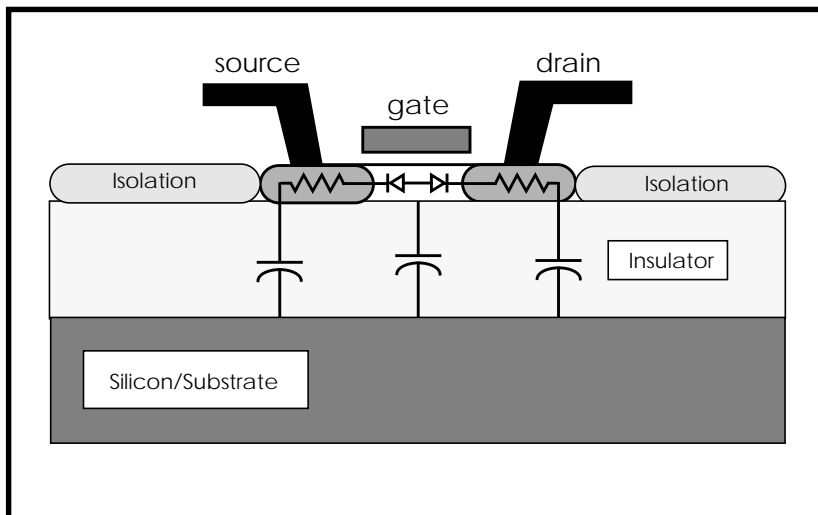
**Figure 16-6: – Non-fully depleted SOI Model**

### Example for Non-Fully Depleted Case

```
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 existance 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
```
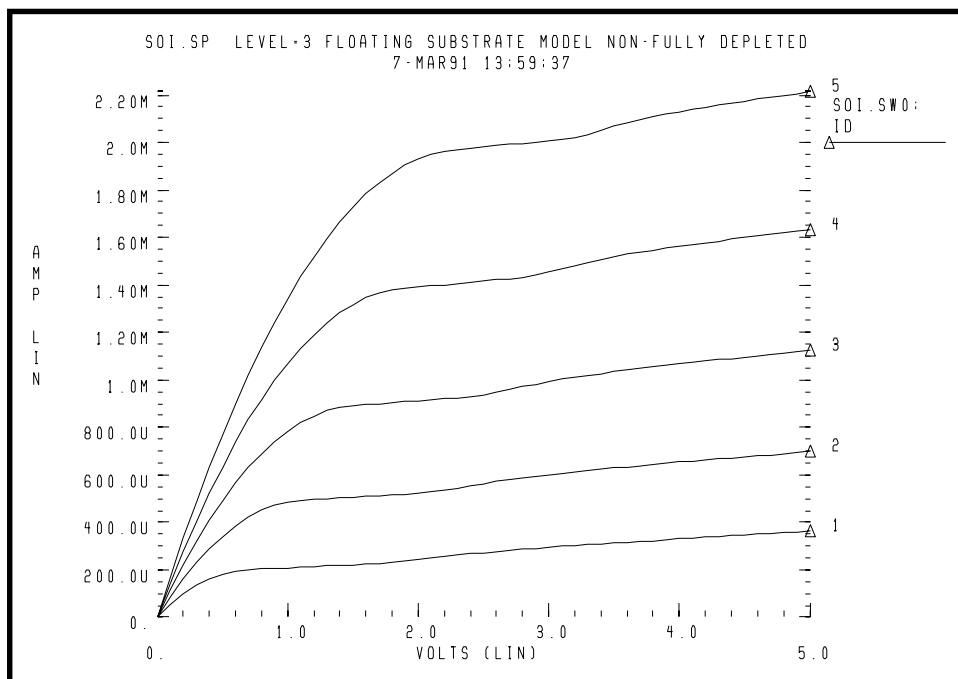
**Figure 16-7: – Level 3 Floating Bulk Model**

## 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 fo 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 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 |

| Name (Alias) | Units | Default | Description |
|---|---|---|---|
| LB2 | μm | 0.0 | length sensitivity |
| 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 | $V^{1/2} \cdot$μm | 0.0 | length sensitivity |
| WGAMN | $V^{1/2} \cdot$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 |
| MUZ | $cm^2/V \cdot s$ | 600 | low drain field first order mobility |

| Name (Alias) | Units | Default | Description |
|---|---|---|---|
| LMUZ | $\mu m \cdot cm^2 / V \cdot s$ | 0.0 | length sensitivity |
| WMUZ | $\mu m \cdot cm^2 / V \cdot s$ | 0.0 | width sensitivity |
| N0 | | 200 | low field weak inversion gate drive coefficient (value of 200 for N0 disables weak inversion calculation) |
| LN0 | $\mu m$ | 0.0 | length sensitivity |
| WN0 | $\mu m$ | 0.0 | width sensitivity |
| NB0 | | 0.0 | Vsb reduction to low field weak inversion gate drive coefficient |
| LNB | $\mu m$ | 0.0 | length sensitivity |
| WNB | $\mu m$ | 0.0 | width sensitivity |
| ND0 | | 0.0 | Vds reduction to low field weak inversion gate drive coefficient |
| LND | $\mu m$ | 0.0 | length sensitivity |
| WND | $\mu m$ | 0.0 | width sensitivity |
| PHI0 | V | 0.7 | two times the Fermi potential |
| LPHI | $V \cdot \mu m$ | 0.0 | length sensitivity |
| WPHI | $V \cdot \mu m$ | 0.0 | width sensitivity |
| TOXM (TOX) | $\mu m$ (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 | $\mu m / V$ | 0.0 | length sensitivity |
| WU0 | $\mu m / V$ | 0.0 | width sensitivity |
| U1 | 1/V | 0.0 | drain field mobility reduction factor |

| Name (Alias) | Units | Default | Description |
|---|---|---|---|
| LU1 | μm/V | 0.0 | length sensitivity |
| WU1 | μm/V | 0.0 | width sensitivity |
| 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 |

| Name (Alias) | Units | Default | Description |
|---|---|---|---|
| LX2U0 | $\mu m/V^2$ | 0.0 | length sensitivity |
| WX2U0 | $\mu m/V^2$ | 0.0 | width sensitivity |
| 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<br>XLREFscaled = XLREF · SCALM |
| WD | m | 0.0 | lateral diffusion into channel from bulk along width<br><br>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<br><br>XLscaled = XL · SCALM |
| XW (DW, WDEL) | m | 0.0 | accounts for masking and etching effects<br><br>XWscaled = XW · SCALM |
| WREF | m | 0.0 | reference channel width<br><br>WREFscaled = WREF · SCALM |
| XWREF | m | 0.0 | difference between physical (on wafer) and drawn reference channel width<br>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 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.

**Examples**

$$VFB0 = -0.350v$$

$$LVFB = -0.1v\mu$$

$$WVFB = 0.08v \cdot \mu$$

$$Leff = 1 \cdot 10^{-6}m = 1\mu$$

$$Weff = 2 \cdot 10^{-6}m = 2\mu$$

$$LREFeff = 2 \cdot 10^{-6}m = 2\mu$$

$$WREFeff = 1 \cdot 10^{-5}m = 10\mu$$

$$zvfb = VFB0 + LVFB \cdot \left(\frac{1}{Leff} - \frac{1}{LREFeff}\right) + WVFB \cdot \left(\frac{1}{Weff} - \frac{1}{WREFeff}\right)$$

$$zvfb = -0.35v + -0.1v \cdot \mu \cdot \left(\frac{1}{1\mu} - \frac{1}{2\mu}\right) + 0.08v \cdot \mu \cdot \left(\frac{1}{2\mu} - \frac{1}{10\mu}\right)$$

$$zvfb = -0.35v - 0.05v + 0.032v$$

$$zvfb = -0.368v$$

## 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,

$Leff = L + XL - 2 \cdot LD$     $Weff = W + XW - 2 \cdot WD$

*Note:  If LDAC and WDAC are included in the .MODEL statement,*

$Leff = L + XL - 2 \cdot LDAC$     $Weff = W + XW - 2 \cdot WDAC$

### General form

```
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.

$xbs = (zphi - vbs)^{1/2}$

$xeta = zeta + zx2e \cdot vbs + zx3e \cdot vds$

$vth = zvfb + zphi + zk1 \cdot xbs - zk2 \cdot xbs^2 - xeta \cdot vds$

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

$$_{ls}, v_{gst}, v_{bs}) = (v_{gst} - body/2 \cdot v_{ds}) \cdot \frac{v_{ds}}{(1 + (xu1 + zx3u1 \cdot v_{ds}) \cdot}$$

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,

$$= beta \cdot (v_{gst} - body/2 \cdot v_{ds}) \cdot \frac{v_{ds}}{(1 + (zu1 + zx3u1 \cdot v_{ds}) \cdot}$$

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 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 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 (HSPICE). All MOSFET models follow this convention.

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 $\bar{\gamma}$ is proportional to both the substrate doping and, to first order, the implant depth. In this model level, the "amplification" of the body effect due to deep implant is accounted for by an empirical parameter, BetaGam.

Model parameters that start with L or W represent geometric sensitivities. In the model equations, a quantity denoted by zX (X being the variable name) is determined by three model parameters: the large-and-wide channel case value X and length and width sensitivities LX and WX, according to zX=X+LX/Leff+WX/Weff. For example, the zero field surface mobility is given by

$$zUO = UO + \frac{LUO}{leff} + \frac{WUO}{weff}$$

*Note:*  *This model uses mostly micrometer units rather than the typical meter units. Units and defaults are often unique in Level 38. The $I_{ds}$ derivatives that give small signal gains gm, gds, and gmbs are calculated using the finite difference method. The options SCALM and DERIV are ineffective for this model.*

## Level 38 Model Parameters

The Level 38 model parameters follow.

### Basic Model Parameters

| 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. |
| 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 | $1.7 \cdot XJ$ | 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. |

| Name (Alias) | Units | Default | Description |
|---|---|---|---|
| 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 |
| 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 |

| Name (Alias) | Units | Default | Description |
|---|---|---|---|
| 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 |

| Name (Alias) | Units | Default | Description |
|---|---|---|---|
| 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 |
| WUO(WUB) | cm$^2$· µm/ (V·s) | 0.0 | UO sensitivity to effective channel width |
| FRCEX(F1EX) | | 0.0 | temperature coefficient for *FRC* |

| Name (Alias) | Units | Default | Description |
|---|---|---|---|
| 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**

$$
ids = \beta1 \cdot \left\{ q \, . \, zKI0 \cdot NI \cdot vde + cav \cdot \left[ (vgs - vfb) \cdot vde - \frac{vde^2}{2} \right] \right.
$$

$$
\left. -\frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Icrit \right\}
$$

**Enhancement, vgs-vfb  vde >0**

$$
ids = \beta1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde - \frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Ic \right.
$$

$$
+ \beta \cdot \left[ (vgs - vfb) \cdot vde - \frac{vde^2}{2} \right]
$$

**Partial Enhancement, vgs-vfb<vde**

$$
ids = \beta1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde + cav \cdot \left[ (vgs - vfb) \cdot vde - \frac{vde^2}{2} \right] \right.
$$

$$
\left. -\frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Icrit \right\}
$$

$$+ \left( \frac{1}{2}\beta - \frac{1}{2}\beta 1 \cdot cav \right) \cdot (vgs - vfb)^2$$

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 $\bar{\gamma}$ 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:

$$vth \ = \ vfb - \beta d \cdot [vch - \bar{\gamma} \cdot (Phid + vsb)^{1/2} + vcrit]$$

where

$$vfb \ = \ zVTO - zETA \cdot vds + \beta d \cdot (vch - \gamma_0 \cdot Phid^{1/2})$$

$$vcrit \ = \ \left(\gamma - \frac{\gamma}{zBetaGam}\right) \cdot (Phid + vsbc)^{1/2} \text{ for vsb} > \text{vsbc; 0 otherwise.}$$

$$\beta d \ = \ \frac{UH \cdot cav}{zUO \cdot cox}$$

$$vch \ = \ \frac{q \cdot NI}{cav}$$

$$\gamma_0 = \frac{(2 \cdot \varepsilon si \cdot q \cdot na1)^{1/2}}{cav}$$

$$na1 = \frac{nd \cdot DNB}{nd + DNB}$$

$$nd = \frac{NI}{DP \cdot 1e\text{-}4}$$

The effective $\overline{\gamma}$, including small device size effects, is computed as follows:

$$\overline{\gamma} = \frac{\gamma}{zBetaGam} \text{ for vsb>vsbc, and } = g \text{ otherwise.}$$

$$\gamma = \gamma_0 \cdot (1 - scf) \cdot (1 + ncf)$$

where

**If SCM $\leq$ 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 $\leq$ 0,**

$$ncf = 0$$

otherwise,

$$ncf = \frac{NWM \cdot xd \cdot (Phid)^{1/2}}{Weff}$$

where

$$xd = \left( \frac{2 \cdot \varepsilon si}{q \cdot DNB} \right)^{1/2}$$

The body effect transition point is calculated as follows:

$$Vsbc = \frac{qDP^2}{2\varepsilon si} \left( \frac{NI}{DP \cdot 1e-4} - DNB \right) + zDVSBC + TDVSBC \cdot (t-tnom) - Phid$$

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:

$$vsat = vgs - vfb + vch + \frac{\gamma^2}{2} \cdot \left\{ 1 - \left[ 1 + \frac{4}{\gamma^2} \cdot (vgs - vfb + vch + vsb + Phid) \right]^{1/2} \right\}$$

$$vdsat = vsat$$

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:

$$UBeff = \cfrac{1}{\cfrac{1}{zUO} + \cfrac{(zFRC + zVFRC \cdot vde + zBFRC \cdot vsb) \cdot (vgs - vfb)}{TOX} + \frac{1}{V}}$$

where

$$Le = Leff \qquad \text{Linear region}$$

$$Le = Leff - \Delta L \quad \text{Saturation region}$$

and at elevated temperatures

$$zFRC(t) = zFRC(tnom) \cdot \left(\frac{t}{tnom}\right)^{FRCEX}$$

The $\Delta L$ is the channel length modulation effect, defined in the next section. Note that $v_{fb}$ assumes the role of $v_{th}$ in the Level 5 mobility equation. The degradation parameters are semi-empirical and grouped together according to their (linearized) mathematical dependencies instead of physical origin to better provide parameter extraction.[7.]

## Channel Length Modulation

The channel length modulation effect is included by modifying the ids current as follows:

$$ids = \cfrac{ids}{1 - \cfrac{\Delta L}{Leff}}$$

where

$$\Delta L = 1e4 \cdot \left[ \frac{2.73e3 \cdot XJ}{na1 \cdot \ln\left(\frac{1e20}{na1}\right)} \right]^{1/3} \cdot [(vds - vdsat + PHI)^{1/3} - PHI^{1/3}]$$

The $\Delta L$ 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

$$ids = \beta1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde + cav \cdot \left[ (von - vfb) \cdot vde - \frac{vde^2}{2} \right] \right.$$

$$\left. - \frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Icrit \right\}$$

$$+ \frac{1}{2} \cdot \left( \beta \cdot e^{\frac{vgs - von}{fast}} - \beta1 \cdot cav \right) \cdot (von - vfb)^2$$

**Full Enhancement, vgs-vfb  -vde > 0**

$$
ids = \beta1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde - \frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/} \right.
$$

$$
+ \beta \cdot \left[ (von - vfb) \cdot vde - \frac{vde^2}{2} \right] \cdot e^{\frac{vgs - von}{fast}}
$$

**Depletion, vgs-vfb  < 0**

$$
ids = \beta1 \cdot \left\{ q \cdot zKI0 \cdot NI \cdot vde + cav \cdot \left[ (vgs - vfb) \cdot vde - \frac{vde^2}{2} \right] \right.
$$

$$
-\frac{2}{3} \cdot cav \cdot \bar{\gamma} \cdot [(vde + vsb + Phid)^{3/2} - (vsb + Phid)^{3/2}] + Icrit \right\} \cdot e^{\frac{vgs - von}{fast}}
$$

## 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=2.9VST=8000UHSAT=0
* CHANNEL LENGTH MODULATION
+ XJ= 0.1
* OXIDE THICKNESS AND CAPACITANCE
+ TOX=165 CGSO=0 CAPOP=2
* CHANNEL IMPLANT
+ NI=1.5e12KCS=3 DP=0.25
*.END
```

# Level 39 BSIM2 Model

The BSIM2 (Berkeley Short-Channel IGFET Model 2)[8,9] is available in HSPICE as Level 39. Meta-Software's implementation of this model is based on Berkeley SPICE 3E2.

Provide input to the model by assigning model parameters, as for other 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 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 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, 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 Meta-Software 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 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 |

| Name (Alias) | Units | Default | Description |
|---|---|---|---|
| MU3B | $cm^2/$ $V^3{\cdot}s$ | 0 | sensitivity of empirical parameter to $V_{bs}$ |
| 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. |
| AIB | $V^{-1}$ | 0 | sensitivity of impact ionization coefficient to $V_{bs}$. |

| Name (Alias) | Units | Default | Description |
|---|---|---|---|
| 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 HSPICE! |
| WDF | m | - | default width. NOT USED IN 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 HSPICE parameters that can be used with HSPICE's BSIM2. See "Gate Capacitance Modeling" on page 16-272 and "MOSFET Diode Model Selection" on page 15-29 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. |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| CGSO | F/m | - | gate-source overlap capacitance. |
|  |  |  | This parameter is calculated if not specified and if LD or METO, and TOX are. |
| 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 HSPICE.

## Other HSPICE Model Parameters Affecting BSIM2

The following HSPICE MOS model parameters are needed to use some 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 "Effective Length and Width for AC Gate Capacitance Calculations" on page 15-103, "Drain and Source Resistance Model Parameters" on page 15-32, "Impact Ionization Model Parameters" on page 15-55, and "Temperature Parameters" on page 15-107. See ".MODEL VERSION Changes to BSIM2 Models" on page 16-197 for information about how the .MODEL statement VERSION parameter changes the BSIM2 model depending on the model version number.

### 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. $LD_{scaled} = LD \cdot SCALM$ |

| 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. |
| 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]*·μμ 2 | 0 | P*x* is Meta-Software'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,

$$\beta' = \beta_0 + \beta_1\tanh\left(MU2\frac{V_{ds}}{V_{dsat}}\right) + \beta_3 V_{ds} - \beta_4 V_{ds}^2$$

$$\beta_0 = \frac{W_{eff}}{L_{eff}} MU \cdot C_{ox} \ ,$$

$$\beta_1 = \beta_S - (\beta_0 + \beta_3 VDD - \beta_4 VDD^2),$$

$$\beta_i = \frac{W_{eff}}{L_{eff}} MUi \cdot C_{ox}, \qquad i = S, 3, 4 \ ,$$

$$a = 1 + \frac{gK1}{2\sqrt{PHI - V_{bs}}},$$

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, $V_{dsat}(V_{geff}(VGLOW)) \approx 0$), 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 $V_{gs} = V_{th} + VGLOW$ and $V_{gs} = V_{th} + VGHIGH$.

## 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. Meta-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 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 Meta-Software 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 HSPICE unless ".OPTION SPICE" is set. In this case TNOM defaults to 27˚ C. This option also sets some other SPICE compatibility parameters. 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 HSPICE.

In contrast to UCB SPICE's BSIM models, HSPICE Level 39 does provide for temperature analysis. The default analysis temperature is 25˚C in HSPICE. Set ".TEMP #" in the HSPICE netlist to change the HSPICE analysis temperature (TEMP as a model parameter is NOT USED). HSPICE provides temperature adjustment of key model parameters, as explained later.

## Parasitics

ACM > 0 invokes HSPICE MOS source-drain parasitics. ACM=0 (default) is SPICE style. See "HSPICE Enhancements" on page 16-200.

## 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 "". See also "Modeling Guidelines and Removal of Mathematical Anomalies" on page 16-203.

Other CAPOPs can be chosen. CAPOP=13 (recommended) selects Meta's BSIM1-based charge-conserving capacitance model that is in common usage with 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 16-104 *for more information.*

## Unused Parameters

SPICE model parameters DELL (S/D diode length reduction) and WDF (default device width) are not used in HSPICE. The function of DELL in SPICE 3E cannot be determined. A default width can be specified in HSPICE on the .OPTION line as DEFW (which defaults to 100μ).

## .MODEL VERSION Changes to BSIM2 Models

HSPICE provides a VERSION parameter to the .MODEL statement, which allows portability of Level 13 BSIM and Level 39 BSIM2 models between 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*

92Alevel 39 BSIM2 model introduced: no changes

92Bno changes

93Aintroduces gds constraints,
fixes WMU3B parameter defect,
fixes PMU0 parameter "O" versus "0" defect, and
introduces MU4 parameter defect

93A.02VERSION parameter introduced,
fixes MU4 parameter defect

95.1fixes defects that cause PMUSB, LDAC, WDAC parameter problems
fixes GMBS defect when gds constraints are used

96.1limited ETA+ ETAB · vb5≥0

## Prevention of Negative Output Conductance

HSPICE internally protects against conditions in the Level 13 model that cause convergence problems due to negative output conductance. The constraints imposed are:

$$MU2 \geq 0 \qquad ND \geq 0 \qquad AI \geq 0$$

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 ($V_{gs}<V_{bs}+\text{VFB}$):

$$Q_g = C_{ox}W_{eff} \cdot L_{eff}(V_{gs} - V_{bs} - VFB)$$

$$Q_b = -Q_g$$
$$Q_d = 0$$

Subthreshold region ($V_{bs} + \text{VFB}<V_{gs}<V_{th} + \text{VGLOW}$):

$$Q_g = C_{ox}W_{eff} \cdot L_{eff}(V_{gs} - V_{bs} - VFB)$$

$$\cdot \left[ 1 - \frac{V_{gs} - V_{bs} - VFB}{V_{gs} - V_{bs} - VFB - V_{gst}} + \frac{1}{3} \left\{ \frac{V_{gs} - V_{bs} - VFB}{V_{gs} - V_{bs} - VFB - V_{gst}} \right\}^2 \right]$$

$$Q_b = -Q_g$$

$$Q_d = 0$$

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 ($V_{ds} < V_{dsat}$):

$$Q_g = \frac{2}{3} C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst} \cdot \left[ \frac{3 \left( 1 - \frac{V_{ds}}{V_{dsat}} \right) + \left( \frac{V_{ds}}{V_{dsat}} \right)^2}{2 - \frac{V_{ds}}{V_{dsat}}} \right] + Q_{bulk}$$

$$Q_b = -Q_{bulk}$$

$$Q_d = -\frac{1}{3} C_{ox} W_{eff} \cdot L_{eff} \cdot V_{gst}$$

$$\cdot \left[ \frac{3\left(1 - \frac{V_{ds}}{V_{dsat}}\right) + \left(\frac{V_{ds}}{V_{dsat}}\right)^2}{2 - \frac{V_{ds}}{V_{dsat}}} + \frac{\frac{V_{ds}}{V_{dsat}}\left(1 - \frac{V_{ds}}{V_{dsat}}\right) + 0.2\left(\frac{V_{ds}}{V_{dsat}}\right)^2}{\left(2 - \frac{V_{ds}}{V_{dsat}}\right)^2} \right] + Q_{bulk}$$

## 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

$$V_{to} = V_{bi} + K1 \cdot \sqrt{PHI} - K2 \cdot PHI$$
$$= VFB + PHI + K1 \cdot \sqrt{PHI} - K2 \cdot PHI,$$

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 HSPICE adjustments to capacitances and parasitic diodes and resistors are effective.

### Alternate Gate Capacitance Model

Select CAPOP=13 for Meta-Software's charge-conserving capacitance model, widely used with LEVEL=13 (BSIM1) and LEVEL=28 (improved BSIM1). See "Level 13 BSIM Model" on page 16-104 for more details.

### Impact Ionization

You can select 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*).

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 "Impact Ionization Equations" on page 15-55 for more details.

## Parasitic Diode for Proper LDD Modeling

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 "MOSFET Diode Model Selection" on page 15-29 for more details.

## Skewing of Model Parameters

The BSIM2 model file, like any other 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 Chapter 9, "AC Sweep and Signal Analysis.".

## HSPICE Optimizer

The BSIM2 model, like any other HSPICE model, can be tied into the HSPICE optimizer for fitting to actual device data.

For more information, see Chapter 11, "Optimizing Performance."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 ≠ if AI0 ≠ 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 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 ≤ 0, and VGHIGH ≥ 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 Meta-Software's
ATEM characterization software and the HSPICE optimizer.
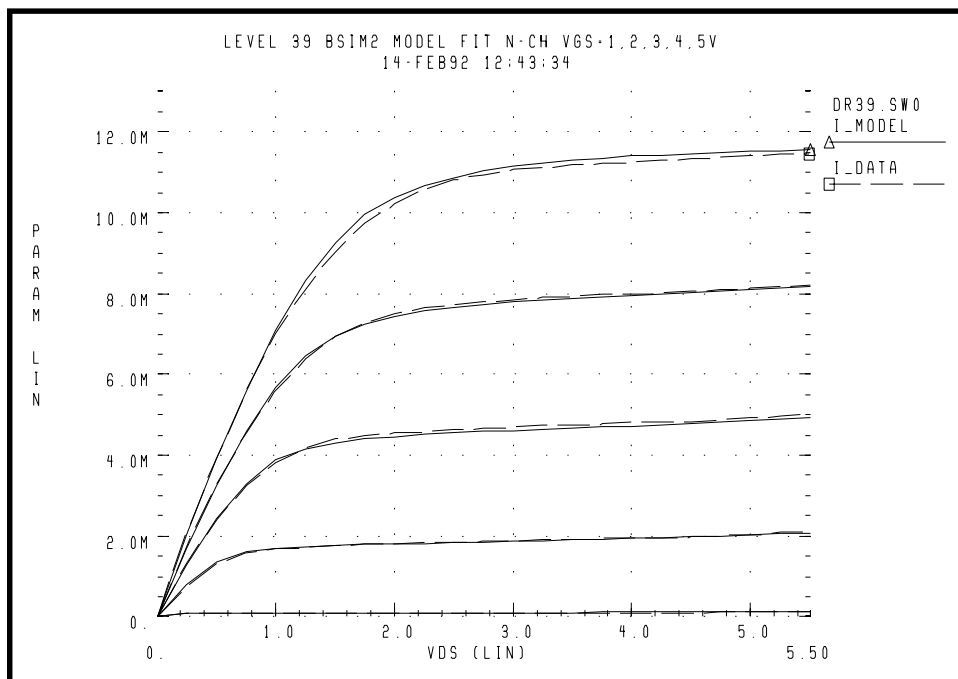


**Figure 16-8: − I$_{DS}$ vs.V$_{ds}$ for V$_{gs}$ = 1, 2, 3, 4, 5V; BSIM2 Model vs. Data**
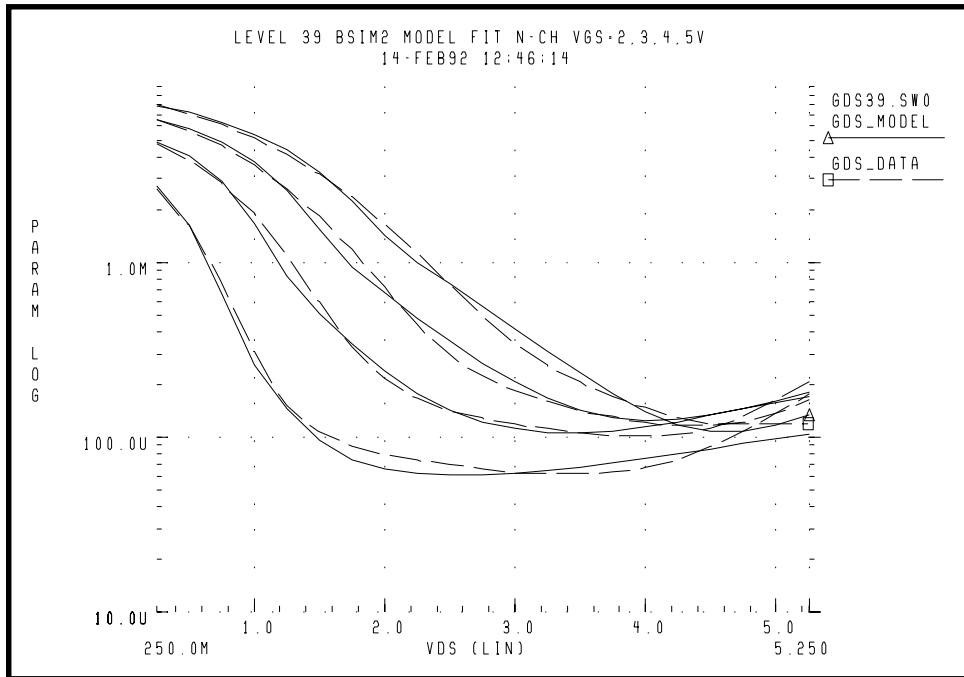
**Figure 16-9: – $g_{ds}$ vs. $V_{ds}$ for $V_{gs}$ = 2, 3, 4, 5V; BSIM2 Model vs. Data, LOG scale**
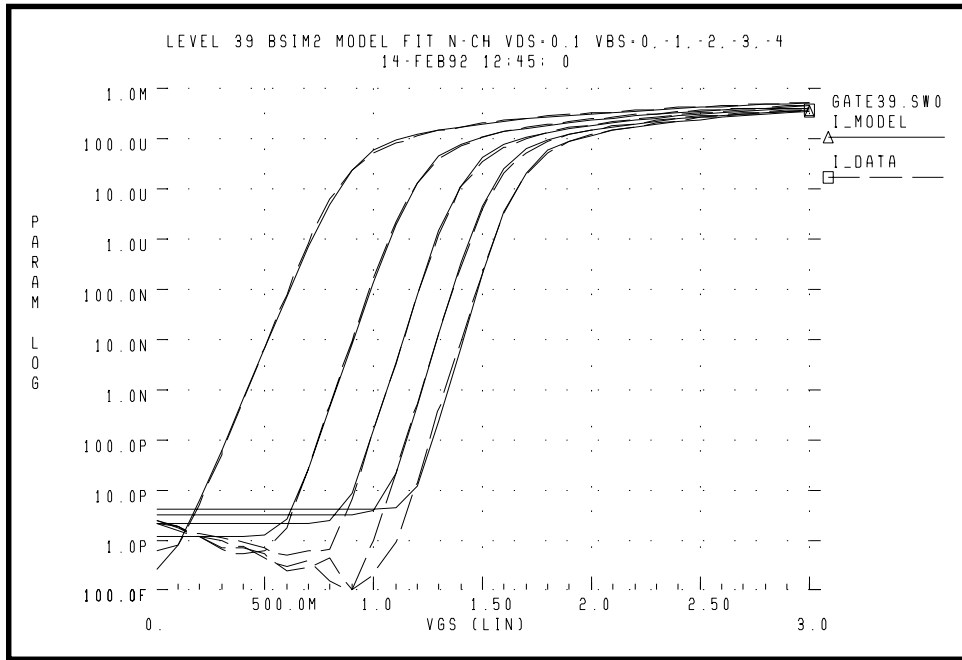
**Figure 16-10: – I$_{DS}$ vs. V$_{gs}$ for V$_{ds}$ = 0.1V, V$_{bs}$ = 0, -1, -2, -3, -4V, Showing Subthreshold Region; Model vs. Data**
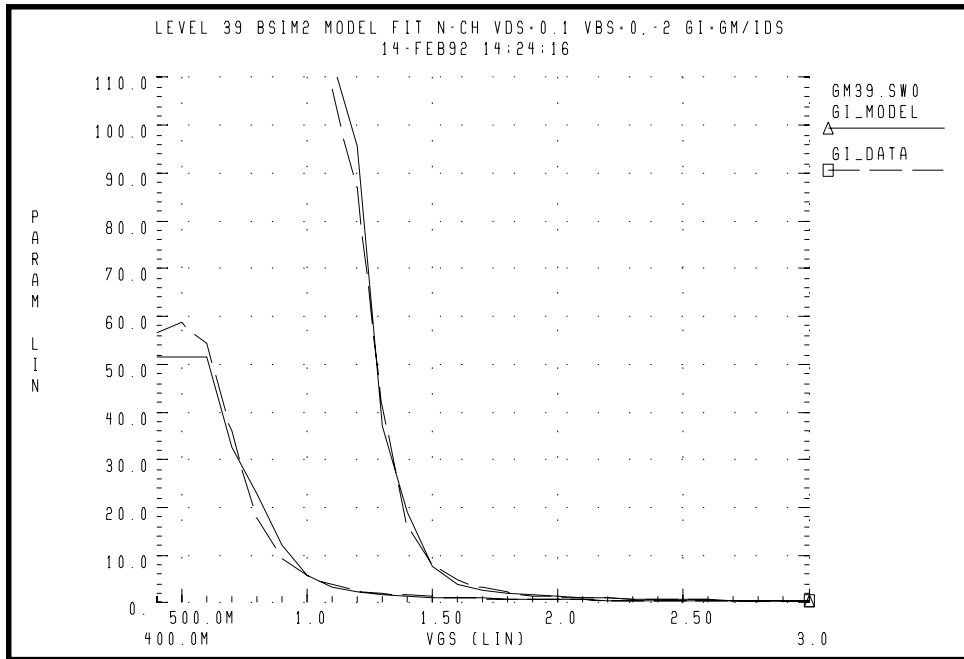
**Figure 16-11: – $_{gm}$/$I_{DS}$ vs. $V_{gs}$ for $V_{ds}$ = 0.1V, $V_{bs}$ = 0, -2V; BSIM2 Model vs. Data**

## Example of 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
```

**Meta-Software 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**

HSPICE Level 40 is a Hewlett-Packard amorphous silicon thin-film transistor model.

## Model Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| 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 | Comments |
|------|-------|---------|----------|
| 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 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$^{o}$ C in HSPICE, but is 27$^{o}$ C in some other simulators. It is a matter of choice whether or not to set the nominal simulation temperature to 27$^{o}$ C, by adding .OPTION TNOM=27 to the netlist. Although the *reference* temperature of the HP a-Si TFT model is fixed at 312$^{o}$ K (or 38.85 $^{o}$C), the behavior of the model adjusts to other *simulation* temperatures that are user specified or provided by 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 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 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 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 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 HSPICE.

### Element DELVTO

DELVTO and DTEMP on the element line can be used with Level 40.

### Example HSPICE model and element statement:

```
.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 $\varepsilon 0$.

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 $Cfm = \dfrac{(\varepsilon 0 \cdot E1 \cdot E2)}{((T2 \cdot E1) + (T1 \cdot E2))}$

If $T1 = 0$ and $T2 \neq 0$, then $Cfm = \dfrac{(\varepsilon 0 \cdot E2)}{T2}$

If $T2 = 0$ and $T1 \neq 0$, then $Cfm = \dfrac{(\varepsilon 0 \cdot E1)}{T1}$

$kp = uo \cdot Cfm \cdot 10^{-4}$

TEMP is the HSPICE device simulation temperature, specified in $^{o}$C, but converted to $^{o}$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 ≤ von)

If $NFS = 0$ and $vgs \le von$, then

$$Cgdi = 0$$

$$Cgsi = 0$$

$$Ids = GO \cdot f(vgs, (DEFF \cdot vds))$$

$$gm = GO$$

$$gds = GO \cdot DEFF$$

## Noncutoff Region (NFS ≠ 0)

If $vgs > von$, then

$$vgsx = vgs$$

If $vgs \le von$, 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 $\geq$ 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

$$fdrain = \frac{1}{\left(1 + \left(\dfrac{vdsx}{vdsc}\right)\right)}$$

$$dfddvg = -dfgdvg \cdot \frac{((fdrain^2) \cdot vdsx)}{(vdsc \cdot fgate)}$$

$$dfddvd = \frac{-(fdrain^2)}{vdsc}$$

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 16-12: shows the topology of the Level 40 model.
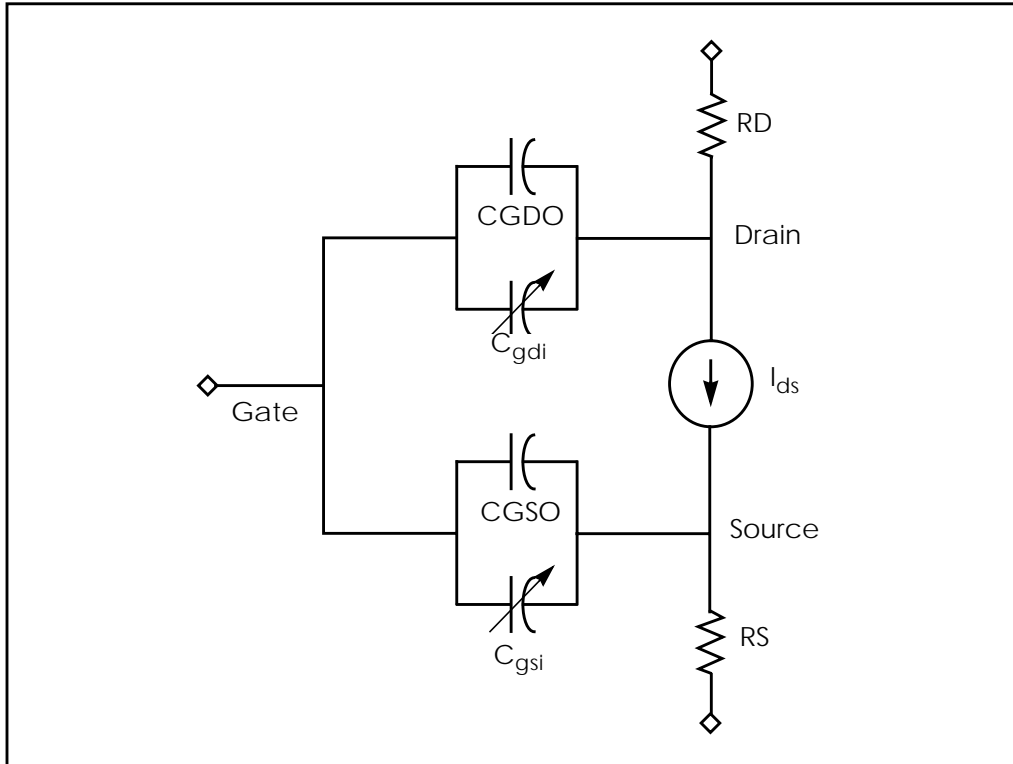


**Figure 16-12: Level 40 HP a-Si TFT Topology**

# Level 47 BSIM3 Version 2 MOS Model

The BSIM3 version 2.0 MOS model from UC Berkeley is available as the Level 47 HSPICE model.

## Level 47 Model Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| 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 "BSIM3 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 |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| TOX | m | 150e-10 | gate oxide thickness |
| 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 "BSIM3 Model Equations" | surface potential under strong inversion |
| XT | m | 1.55e-7 | doping depth |
| VBM | V | -5.0 | maximum substrate bias |
| VBX | V | See "BSIM3 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, below) | 0.067 | low field mobility at T = $TREF$ (0.067 for n-channel, 0.025 for p-channel) |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| UA | m/V | 2.25e-9 | first-order mobility degradation coefficient |
| UA1 | m/V | 4.31e-9 | temperature coefficient of *UA* |
| 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 |

$$LITL = \left( \frac{\varepsilon_{si} T_{ox} X_j}{\varepsilon_{ox}} \right)^{1/2}$$

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| VGLOW | V | -0.12 | lower bound of the weak-strong inversion transition region |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| VGHIGH | V | 0.12 | upper bound of the weak-strong inversion transition region |
| 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) |
| A2 | | 1.0 | second nonsaturation factor (1.0 for NMOS, 0.08 for PMOS) |
| UTE | | -1.5 | mobility temperature exponent |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| 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 HSPICE

The HSPICE Level 47 model uses the same model parameters for source/drain diode current, capacitance, and resistance as do the other HSPICE MOS levels. The model parameter ACM controls the choice of source/drain equations.

The 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 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 Chapter 11, "Optimizing Performance."

## 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 HSPICE is *TREF*, which is supported as a model parameter in all 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 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 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 HSPICE and SPICE3 as follows:

**Example**
```
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 HSPICE MOS parasitic models, specified by ACM.

17. Level 47 uses the common 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 HSPICE equations for Leff and Weff are

$$L_{eff} = L + XL - (2 \cdot LD)$$

$$W_{eff} = W + XW - (2 \cdot WD)$$

The UCB SPICE3 equations used for BSIM3 are

$$L_{eff} = L - (2 \cdot DL)$$

$$W_{eff} = W - (2 \cdot DW)$$

The units for these parameters are meters, with defaults of zero.

HSPICE uses the standard 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 HSPICE .MODEL statement, 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 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

$V_{th0}$, $K1$, $K2$, $\phi_s$, $N_{lx}$, $K3$, $W_0$, $T_{ox}$, $V_{bi}$, $D_{vt0}$, $D_{vt1}$, $D_{vt2}$, $N_{peak}$, $N_{sub}$, $\Upsilon_1$, $\Upsilon_2$, $V_{bx}$, $V_{bm}$, $V_{bi}$, $X_t$, $TREF$

$$V_{th} = V_{th0} + K1(\sqrt{\phi_s - V_{bs}} - \sqrt{\phi_s}) - K2V_{bs} + K1\left(\sqrt{1 + \frac{N_{lx}}{L_{eff}}\sqrt{\frac{\phi_s}{\phi_s - V_{bs}}}} - 1\right)\sqrt{\phi_s}$$

$$+ \ (K3 + K3B \cdot V_{bs}) \cdot \left(\frac{T_{ox}}{W_{eff} + W_0}\right)\phi_s - \Delta V_{th}$$

$$T_{ratio} = \frac{(TEMP + DTEMP + 273.15)}{(TREF + 273.15)}$$

$$\Delta V_{th} = \theta_{th}(L_{eff}) \cdot (V_{bi} - \phi_s)$$

$$\theta_{th}(L_{eff}) = D_{vt0} \cdot \left[exp\left(\frac{-D_{vt1} \cdot L_{eff}}{2l_t}\right) + 2\,exp\left(\frac{-D_{vt1} \cdot L_{eff}}{l_t}\right)\right]$$

$$l_t = \sqrt{3 \cdot T_{ox} \cdot X_{dep}} \cdot (1 + D_{vt2} \cdot V_{bs})$$

$$X_{dep} = \sqrt{\frac{2 \cdot \varepsilon_{si} \cdot (\phi_s - V_{bs})}{q \cdot N_{peak}}}$$

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*

$\mu_0$,   $U_a$,   $U_b$,   $U_c$

$$\mu_{\text{eff}} = \frac{\mu_0}{1 + U_a \cdot \left(\dfrac{V_{gs} + V_{\text{th}}}{T_{\text{ox}}}\right) + U_b \cdot \left(\dfrac{V_{gs} + V_{\text{th}}}{T_{\text{ox}}}\right)^2 + U_c \cdot V_{bs}}$$

## 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,

$$A_{\text{bulk}} = \left(1 + \frac{K1 \cdot A_0 \cdot L_{\text{eff}}}{(L_{\text{eff}} + T1) \cdot T1s \cdot 2}\right)\Big/ (1 + KETA \cdot V_{bs})$$

For BULKMOD = 2,

$$A_{\text{bulk}} = \left( \frac{K1 \cdot A_0 \cdot L_{\text{eff}}}{(L_{\text{eff}} + T1) \cdot \sqrt{\phi_s} \cdot 2} \right) \Big/ (1 + KETA \cdot V_{bs})$$

$$T1 = 2 \cdot \sqrt{X_j \cdot X_{\text{dep}}}$$

For $V_{\text{bs}}$ <= 0,

$$T1s = \sqrt{\phi_s - V_{bs}}$$

For $V_{\text{bs}}$ >= 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:

$V_{\text{dsat}}$ is the solution of $Tmpa * V_{\text{dsat}} * V_{\text{dsat}} - Tmpb * V_{\text{dsat}} + Tmpc = 0$

$$V_{\text{dsat}} = \left( Tmpb - \sqrt{Tmpb^2 - 4 \cdot Tmpa \cdot Tmpc} \right) \Big/ (2 \cdot Tmpa)$$

$$Tmpa = A_{\text{bulk}} \cdot (A_{\text{bulk}} \cdot W_{\text{eff}} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot R_{ds} - 1 + 1/Pfactor)$$

$$Tmpb = V_{gst} \cdot (2/Pfactor - 1) + (A_{\text{bulk}} \cdot E_{\text{sat}} \cdot L_{\text{eff}}) + (3 \cdot A_{\text{bulk}} \cdot V_{gst} \cdot W_{\text{eff}} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot R_{ds})$$

$$Tmpc = (V_{gst} \cdot E_{\text{sat}} \cdot L_{\text{eff}}) + (V_{gst}^2 \cdot 2 \cdot W_{\text{eff}} \cdot v_{\text{sat}} \cdot C_{\text{ox}} \cdot R_{ds})$$

## 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,

$$I_{ds} = u0 \cdot \sqrt{\frac{q \cdot \varepsilon_{si} \cdot N_{peak}}{2 \cdot \phi_s}} \cdot \frac{W_{eff}}{L_{eff}} \cdot V_{tm}^2 \cdot \left[ 1 - exp\left(\frac{-V_{ds}}{V_{tm}}\right) \right] \cdot exp\left(\frac{V_{gs} - V_{th} - V_{off} + DIBL \cdot V_{ds}}{n \cdot V_{tm}}\right)$$

## Transition Region (for subthMod = 2 only)

### Model Parame*ters*

$V_{gshigh}, \; V_{gslow}$

$$I_{ds} = (1-t)^2 \cdot I_{dslow} + 2 \cdot (1-t) \cdot t \cdot I_p + t^2 \cdot I_{dshigh}$$

$$t = \left( \frac{V_p - V_{gslow}}{V_{gslow} - 2 \cdot V_p + V_{gshigh}} \right) \cdot \left( \sqrt{1 + \frac{(V_{gslow} - 2 \cdot V_p + V_{gshigh})(V_{gs} - V_{th} - V_{gslow})}{(V_p - V_{gslow})^2}} - 1 \right)$$

$$V_p = \frac{(g_{mhigh} \cdot V_{gshigh} - g_{mlow} \cdot V_{gslow}) - (I_{dshigh} - I_{dslow})}{g_{mhigh} - g_{mlow}}$$

$$I_p = I_{dslow} + g_{mlow} \cdot (V_p - V_{gslow})$$

## Temperature Compensation

### Model Parameters

$A_t, \; U_{a1}, \; U_{b1}, \; U_{c1}, \; KT1, \; KT2, \; UTE$

$$V_{th}(temp) = V_{th}(tref) + (KT1 + KT2 \cdot V_{bs}) \cdot (T_{ratio} - 1)$$

$$u0(temp) = u0(tref) \cdot (T_{ratio})^{UTE}$$

$$V_{sat}(temp) = V_{sat}(tref) - A_t \cdot (T_{ratio} - 1)$$

$$U_a(temp) = U_a(tref) + U_{a1} \cdot (T_{\text{ratio}} - 1)$$

$$U_b(temp) = U_b(tref) + U_{b1} \cdot (T_{\text{ratio}} - 1)$$

$$U_c(temp) = U_c(tref) + U_{c1} \cdot (T_{\text{ratio}} - 1)$$

## Sample 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
```

# Level 49 BSIM3 Version 3 MOS Model

The BSIM3v3 MOS model from UC Berkeley has been installed as Level 49 in Star-Hspice. Level 49 is compliant with the Berkeley release except for some differences in default values and the absence of the non-quasi static (NQS) capacitance model.

Level 49 also provides enhancements to the standard BSIM3 model including the Star-Hspice junction and noise models and can be parametrized. Parameterization is useful for modeling process skew, either by worst-case corners or Monte Carlo analysis. See Chapter 9, "Parametric Variation and Statistical Analysis," for information on worst-case and Monte Carlo analysis. Specific level 49 enhancements and differences between level 49 and Berkeley-BSIM3v3 are discussed below.

## Version

The model parameter, VERSION, selects among the various Berkeley releases of BSIM3v3 as follows:

- Version 3.0 Berkeley (October 31, 1995) default for HSPICE96.1,96.2,96.3
- Version 3.1 Berkeley (December 12, 1997) default for HSPICE97.1,97.2
- Version 3.11 Berkeley (December 12, 1997) default for HSPICE97.4
- Version 3.1 is maintained for compatibility with the older Star-Hspice version. In order to realize the 97.4 enhancements and bug fixes make sure that VERSION is set or defaults to 3.11.

## Model Parameter Defaults

The following parameter default values deviate from the Berkeley BSIM3v3 default values and should be set explicitly in the model card:

VERSION, SFVTFLAG, ACM, NQS=0, XPART, CAPMOD, JS, CJ, CJSW

## Star-Hspice Enhancements

The performance of level 49 has been improved by reducing model equation complexity, replacing some calculations with spline functions, and optimizing the compiler. The result is a reduction in simulation time of up to 35% (for both 3.11 and 3.1 versions) compared to previous releases while maintaining accuracy to 5 digits or better. The use of spline functions can be disabled by setting the model parameter, SFVTFLAG=0, in the model card.The default value, SFVTFLAG=1, enables the spline functions.

In the 12/10/96 release of BSIM3v3, Berkeley offers the BSIM1 capacitance model as CAPMOD=0. In the 3.1 and 3.11 versions in Star-Hspice we offer a modified BSIM1 capacitance model based on the Star-Hspice CAPOP=13 model. The current default is CAPMOD=0 for VERSION=3.1, 3.11 and CAPMOD=1 for VERSION=3.0.

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 VTHO please set the model parameter VFBFLAG=1. The capacitance model CAPMOD=0 will calculate the threshold voltage as Vth = vfb + phi +k1 * sqrt(phi - vbs), where vfb is identical to the flatband voltage used in the drain current calculations. This setting will show capacitance dependence on VTH0. The VFBFLAG default value is 0.

Either Berkeley BSIM3v3 or standard Star-Hspice MOSFET junction diode models can be used in Level 49. Specifying ACM=0,1,2,3 will invoke the standard Star-Hspice junction model (see pg.15-19 to 15-36). The default ACM is zero.

The Berkeley BSIM3v3 junction model can be invoked by setting ACM=10,11,12,13. In addition, setting ACM=10-13 provides external source and drain parasitic resistances. These parasitic resistance equations are calculated using the corresponding ACM=0-3 parasitic resistance equations (see pg.15-19 to 15-36).

*Note:*  *ACM=10-13 only apply Star-Hspice area calculation equations to the parasitic resistors, Rs and Rd.*

Star-Hspice equations are not used in calculating areas and perimeters for the Berkeley junction models. Note the following parameters are ignored when using ACM=0-3: NJ, CJSWG, PBSWG, MJSWG. The following parameters are ignored when using ACM=10-13: PHP, CJGATE.

The Star-Hspice noise equations (see pg. 15-71 to 15-73) are invoked when the parameter NLEV is specified in the model card. If NLEV is not specified, the Berkeley noise equations are invoked.

Printback of all model parameters with units is now enabled. The printback also indicates whether Berkeley or Star-Hspice junction diodes are invoked and which parameters are not used (e.g. CJGATE is not used when ACM=0-3).

## Level 49 Model Parameters

The Level 49 model parameters are described in this section.

### Model Flags

| Name | Units | Default | Comments |
|---|---|---|---|
| *VERSION* | - | 3.11 | Selects from BSIM3 Versions 3.0, 3.1, 3.11 (version=3.11 is the HSPICE97.4 equivalent to BSIM3v3 version 3.1) |
| *PARAMCHK* | - | 0 | PARAMCHK=1 will check some model parameters for range compliance |
| *BINFLAG* | - | 0 | Uses wref, lref when set > 0.9 |
| *MOBMOD* | - | 1 | mobility model selector |
| *CAPMOD* | - | 0 | Selects from Charge models 0,1,2 (default deviates from BSIM3v3 = 2) |
| *NOIMOD* | - | 1 | Berkeley noise model flag |
| *NLEV* | - | 0 (off) | Star-Hspice noise model flag (non-zero overrides NOIMOD) (Star-Hspice specific) |

| NQSMOD | - | 0 (off) | NQS Model is not supported in Level 49 |
| SFVTFLAG | - | 1 (on) | Spline function for Vth (Star-Hspice specific) |
| VFBFLAG | - | 0 (off) | UFB selector for CAPMOD=0 (Star-Hspice specific) |

### Basic Model Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| TOX | m | 150e-10 | gate oxide thickness |
| XJ | m | 0.15e-6 | junction depth |
| NGATE | cm$^{-3}$ | infinite | Poly gate doping concentration |
| VTH0 (VTHO) | V | 0.7 | threshold voltage of long channel device at $V_{bs} = 0$ and small $V_{ds}$ (typically 0.7 for n-channel, - 0.7 for p-channel) |
| NSUB | cm$^{-3}$ | 6.0e16 | substrate doping concentration |
| NCH | cm$^{-3}$ (see Note 6) | 1.7e17 | peak doping concentration near interface |
| NLX | m | 1.74e-7 | lateral nonuniform doping along channel |
| K1 | V$^{1/2}$ | 0.50 | 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 |
| W0 | m | 2.5e-6 | narrow width effect coefficient |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *DVT0W* | 1/m | 0 | narrow width coefficient 0, for Vth, at small L |
| *DVT1W* | 1/m | 5.3e6 | narrow width coefficient 1, for Vth, at small L |
| *DVT2W* | 1/V | -0.032 | narrow width coefficient 2, for Vth, at small L |
| *DVT0* | - | 2.2 | short channel effect coefficient 0, for $V_{th}$ |
| *DVT1* | - | 0.53 | short channel effect coefficient 1, for $V_{th}$ |
| *DVT2* | 1/V | -0.032 | short channel effect coefficient 2, for $V_{th}$ |
| *ETA0* | - | 0.08 | subthreshold region DIBL (Drain Induced Barrier Lowering) coefficient |
| *ETAB* | 1/V | -0.07 | subthreshold region DIBL coefficient |
| *DSUB* | - | DROUT | DIBL coefficient exponent in subthreshold region |
| *VBM* | V | -3.0 | maximum substrate bias, for $V_{th}$ calculation |
| *U0* | cm$^2$/V/ sec | 670 nmos 250 pmos | low field mobility at T = *TREF* = *TNOM* |
| *UA* | m/V | 2.25e-9 | first-order mobility degradation coefficient |
| *UB* | m$^2$/V$^2$ | 5.87e-19 | second-order mobility degradation coefficient |
| *UC* | 1/V | -4.65e-11 or -0.0465 | body bias sensitivity coefficient of mobility -4.65e-11 for MOBMOD=1,2 or, -0.0465 for MOBMOD = 3 |
| *A0* | - | 1.0 | bulk charge effect coefficient for channel length |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *AGS* | 1/V | 0.0 | gate bias coefficient of Abulk |
| *B0* | m | 0.0 | bulk charge effect coefficient for channel width |
| *B1* | m | 0.0 | bulk charge effect width offset |
| *KETA* | 1/V | -0.047 | body-bias coefficient of bulk charge effect |
| *VOFF* | V | -0.08 | offset voltage in subthreshold region |
| *VSAT* | m/sec | 8e4 | saturation velocity of carrier at T = *TREF* = *TNOM* |
| A1 | 1/V | 0 | first nonsaturation factor |
| A2 | - | 1.0 | second nonsaturation factor |
| *RDSW* | ohm · μm | 0.0 | parasitic source drain resistance per unit width |
| *PRWG* | 1/V | 0 | gate bias effect coefficient of RDSW |
| *PRWB* | $1/V^{1/2}$ | 0 | body effect coefficient of RDSW |
| *WR* | - | 1.0 | width offset from Weff for Rds calculation |
| *NFACTOR* | - | 1.0 | subthreshold region swing |
| *CIT* | $F/m^2$ | 0.0 | interface state capacitance |
| *CDSC* | $F/m^2$ | 2.4e-4 | drain/source and channel coupling capacitance |
| *CDSCD* | $F/Vm^2$ | 0 | drain bias sensitivity of CDSC |
| *CDSCB* | $F/Vm^2$ | 0 | body coefficient for CDSC |
| *PCLM* | - | 1.3 | coefficient of channel length modulation values $\leq$ 0 will result in an error message and program exit. |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *PDIBLC1* | - | 0.39 | DIBL (Drain Induced Barrier Lowering) effect coefficient 1 |
| *PDIBLC2* | - | 0.0086 | DIBL effect coefficient 2 |
| *PDIBLCB* | 1/V | 0 | body effect coefficient of DIBL effect coefficients |
| *DROUT* | - | 0.56 | length dependence coefficient of the DIBL correction parameter in $R_{out}$ |
| *PSCBE1* | V/m | 4.24e8 | substrate current induced body effect exponent 1 |
| *PSCBE2* | V/m | 1.0e-5 | substrate current induced body effect coefficient 2 |
| *PVAG* | - | 0 | gate dependence of Early voltage |
| *DELTA* | V | 0.01 | effective Vds parameter |
| *ALPHA0* | m/V | 0 | the first parameter of impact ionization current |
| *BETA0* | V | 30 | the second parameter of impact ionization current |
| *RSH* | 0.0 | ohm/square | source/drain sheet resistance in ohm per square |

### AC and Capacitance Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *XPART* | - | 1 | charge partitioning rate flag (default deviates from BSIM3V3=0) |
| *CGSO* | F/m | p1 (see Note1) | non-LDD region source-gate overlap capacitance per unit channel length |
| *CGDO* | F/m | p2 (see Note2) | non-LDD region source-gate overlap capacitance per unit channel length |
| *CGBO* | F/m | 0 | gate-bulk overlap capacitance per unit channel length |
| *CGS1* | F/m | 0.0 | lightly doped source-gate overlap region capacitance |
| *CGD1* | F/m | 0.0 | lightly doped drain-gate overlap region capacitance |
| *CKAPPA* | F/m | 0.6 | coefficient for lightly doped region overlap capacitance fringing field capacitance |
| *CF* | F/m | (see Note3) | fringing field capacitance |
| *CLC* | m | 0.1e-6 | constant term for the short channel model |
| *CLE* | - | 0.6 | exponential term for the short channel model |

## Length and Width Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *WINT* | m | 0.0 | width offset fitting parameter from I-V without bias |
| *WLN* | - | 1.0 | power of length dependence of width offset |
| *WW* | $m^{WWN}$ | 0.0 | coefficient of width dependence for width offset |
| *WWN* | - | 1.0 | power of width dependence of width offset. |
| *WWL* | $m^{WWN}$ $*m^{WLN}$ | 0.0 | coefficient of length and width cross term for width offset |
| *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 |
| *LINT* | m | 0.0 | length offset fitting parameter from I-V without bias |
| *LL* | $m^{LLN}$ | 0.0 | coefficient of length dependence for length offset |
| *LLN* | - | 1.0 | power of length dependence of length offset |
| *LW* | $m^{LWN}$ | 0.0 | coefficient of width dependence for length offset |
| *LWN* | - | 1.0 | power of width dependence of length offset |
| *LWL* | $m^{LWN}$ $*m^{LLN}$ | 0.0 | coefficient of length and width cross term for length offset |
| *DLC* | m | LINT | length offset fitting parameter from CV |
| *DWC* | m | WINT | width offset fitting parameter from CV |

### Temperature Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *KT1* | V | 0.0 | temperature coefficient for Vth |
| *KT1L* | m-V | 0.0 | temperature coef. for channel length dependence of Vth |
| *KT2* | - | 0.022 | body bias coefficient of Vth temperature effect |
| *UTE* | - | -1.5 | mobility temperature exponent |
| *UA1* | m/V | 4.31e-9 | temperature coefficient for UA |
| *UB1* | $(m/V)^2$ | -7.61e-18 | temperature coefficient for UB |
| *UC1* | $m/V^2$ | -5.69e-11 | temperature coefficient for UC |
| *AT* | m/sec | 3.3e4 | temperature coefficient for saturation velocity |
| *PRT* | ohm-um | 0 | temperature coefficient for RDSW |
| *XTI* | - | 3.0 | junction current temperature exponent |

### Bin Description Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *LMIN* | m | 0.0 | maximum channel length |
| *LMAX* | m | 1.0 | maximum channel length |
| *WMIN* | m | 0.0 | minimum channel width |
| *WMAX* | m | 1.0 | maximum channel width |
| *BINUNIT* | - | 0 | assumes weff,leff,wref,lref units are in microns when BINUNIT=1 or meters otherwise |

### Process Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *GAMMA1* | $V^{1/2}$ | see Note 8 | body effect coefficient near the surface |
| *GAMMA2* | $V^{1/2}$ | see Note 9 | body effect coefficient in the bulk |
| *VBX* | V | see Note 10 | VBX at which the depletion region width equals XT |
| *XT* | m | 1.55e-7 | doping depth |
| *VBI* | V | see Note 11 | drain and source junction built-in potential |

### Noise Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *NIOA* | - | 1.0e20 nmos 9.9e18 pmos | body effect coefficient near the surface |
| *NOIB* | - | 5.0e4 nmos 2.4e3 pmos | body effect coefficient in the bulk |
| *NOIC* | - | -1.4e-12 nmos 1.4e-12 pmos | VBX at which the depletion region width equals XT |
| *EM* | V/m | $4.1e^{7}$ | doping depth |
| *AF* | - | 1.0 | drain and source junction built-in potential |
| *KF* | - | 1.0 | flicker exponent |
| *EF* | - | 0.0 | flicker noise parameter |

### Junction Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *ACM* | - | 0 | Area calculation method selector (Star-Hspice specific) |
| *JS* | A/m$^2$ | 0.0 | bulk junction saturation current (Default deviates from BSIM3v3 = 1.0e$^{-4}$) |
| *JSW* | A/m | 0.0 | sidewall bulk junction saturation current |
| *NJ* | - | 1 | emission coefficient (not used with ACM=3) |
| *CJ* | F/m$^2$ | 5.79e$^{-4}$ | zero-bias bulk junction capacitance (Default deviates from BSIM3v3 = 5.0e$^{-4}$) |
| *CJSW* | F/m | 0.0 | zero-bias sidewall bulk junction capacitance (Default deviates from BSIM3v3 = 5.0e$^{-10}$) |
| *CJSWG* | F/m | CJSW | zero-bias gate-edge sidewall bulk junction capacitance (not used with ACM=0-3) |
| *PB, PHIB* | V | 1.0 | bulk junction contact potential |
| *PBSW* | V | 1.0 | sidewall bulk junction contact potential |
| *PBSWG* | V | PBSW | gate-edge sidewall bulk junction contact potential (not used with ACM=0-3) |
| *MJ* | - | 0.5 | bulk junction grading coefficient |
| *MJSW* | - | 0.33 | sidewall bulk junction grading coefficient |
| *MJSWG* | - | MJSW | gate-edge sidewall bulk junction grading coefficient (not used with ACM=0-3) |

### NQS Parameters

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *ELM* | - | 5.0 | Elmore constant (Star-Hspice currently does not support the NQS model) |

### 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:

$$V_{bx} = \phi_s - \frac{q n_{ch} X_t^2}{2\varepsilon_{si}}$$

11. If $V_{bi}$ is not given, it is calculated using:

$$V_{bi} = \frac{k_B T}{q} \log\left(\frac{n_{ch} N_{DS}}{n_i^2}\right)$$

where $N_{DS}$ is $1e20/c^2$

Level 49 uses the BSIM3.V3 capacitance model. However, it does not support the nonquasistatic (NQS) capacitance model. The Level 49 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 49, for compatibility with SPICE3.

The default room temperature is 25°C in Star-Hspice, but is 27°C in SPICE3. If the BSIM3 model parameters are specified at 27°C, TNOM=27 should be added to the model, so that the model parameters are interpreted correctly. It is a matter of choice whether or not to set the nominal simulation temperature to 27, by adding .OPTION TNOM=27 to the netlist. Add this option when testing Star-Hspice versus SPICE3.

DELVTO and DTEMP on the element line can be used with Level 49.

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 ...
```

The option SCALM does not affect the parameters unique to this model, but it does affect the common MOS parameters, such as CJ, CJSW, JS, and JSW.

There are three ways for the BSIM3 model to calculate $V_{th}$ using:

- *K1* and *K2* values that are user specified
- *GAMMA1*, *GAMMA2*, *VBM*, and *VBX* values entered in the .MODEL statement
- *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.

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 Star-Hspice parameters XL and XLREF are handled in a manner consistent with other models, and they produce shifts in parameter values without disrupting the continuity across bin boundaries.

## Level 49 Equations

The effective channel length and width used in all model equations are:

$$L_{eff} = L_{drawn} - 2dL$$

$$W_{eff} = W_{drawn} - 2dW$$

$$W_{eff}' = W_{drawn} - 2dW'$$

where the unprimed $W_{eff}$ is bias-dependent, and the primed quantity is bias-independent.

$$dW = dW' + dW_g V_{gsteff} + dW_b(\sqrt{\phi_s - V_{bseff}} - \sqrt{\phi_s})$$

$$dW' = W_{int} + \frac{W_L}{L^{WLN}} + \frac{W_W}{W^{WWN}} + \frac{W_{WL}}{L^{WLN} W^{WWN}}$$

$$dL = L_{int} + \frac{L_L}{L^{LLN}} + \frac{L_W}{W^{LWN}} + \frac{L_{WL}}{L^{LLN} W^{LWN}}$$

*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*

## Example .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
```

## Example 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 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 Philips document does not describe built-in parasitic diodes. Standard HSPICE parasitic diode equations, using parameters JS, JSW, N, CJ, CJSW, CJGATE, MJ, MJSW, PB, PHP, ACM, HDIF have been added. The older parameter IS is not used.

## Level 50 Model Parameters

| Name | Units | Default(N) | Default(P) | Comments |
|------|-------|-----------|-----------|----------|
| 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 |
| 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 |

| Name | Units | Default(N) | Default(P) | Comments |
|------|-------|-----------|-----------|----------|
| 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* |
| 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* |

| Name | Units | Default(N) | Default(P) | Comments |
|------|-------|-----------|-----------|----------|
| 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* |
| 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* |

| Name | Units | Default(N) | Default(P) | Comments |
|------|-------|-----------|-----------|----------|
| 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* |
| 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* |

| Name | Units | Default(N) | Default(P) | Comments |
|------|-------|-----------|-----------|----------|
| 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 |
| NTR | J | 24.4e-21 | 21.1e-21 | thermal noise coefficient |
| NFR | $V^2$ | 70.0e-12 | 21.4e-12 | flicker noise coefficient |

## Using the Philips MOS9 Model in HSPICE

1.  Set LEVEL=50 to identify the model as the Philips MOS Model 9.

2.  The default room temperature is 25 $^{\circ}$C in HSPICE, but is 27 $^{\circ}$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 HSPICE. The default for TR is 21.0 $^{\circ}$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 the table above be set in the .MODEL statement.

## Example 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
```

# 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 | Meta-Software proprietary model, based on BSIM1 |
| Level 39 | BSIM2 |

## History and Motivation

This section describes the history of and motivation for using MOS models in Star-Hspice.

### HSPICE Model Enhancements

In order to satisfy the needs of customers, Avant! modified the standard SPICE models. 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 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 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 Meta-Labs.

### Level 39

The BSIM2 model was developed by Duster, Jeng, Ko, and Hu, and released in SPICE3 in 1991. It is designed for deep sub-micron 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 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 HSPICE Parameters with UCB SPICE 2 and 3" on page 16-126.

# 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. 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. 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 HSPICE parameters ALPHA and VCR, the impact ionization current is assigned to IDB, which is essential for cascode simulation. The 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 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 HSPICE for compatibility, using XL, LD, XW, WD is recommended instead. In 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 HSPICE, several choices of capacitance models are available; the range of choices and the default varies with the model chosen. The default for Levels 2 and 3 is still the Meyer model, but you can also select a charge-conserving Ward-Dutton model.

## Level Comparisons

### Table 16-2:

| 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 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 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 byimproved 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.
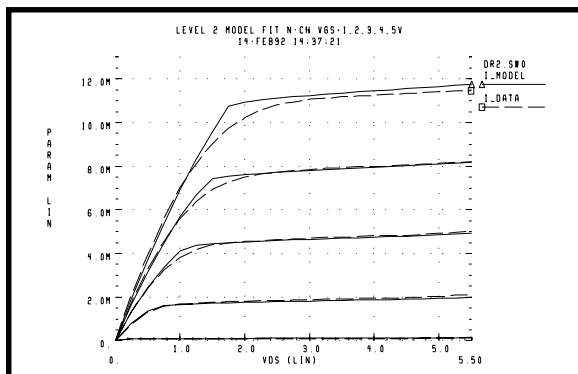


**Figure 16-13: – Level 2 Ids Model vs. Data Curves**
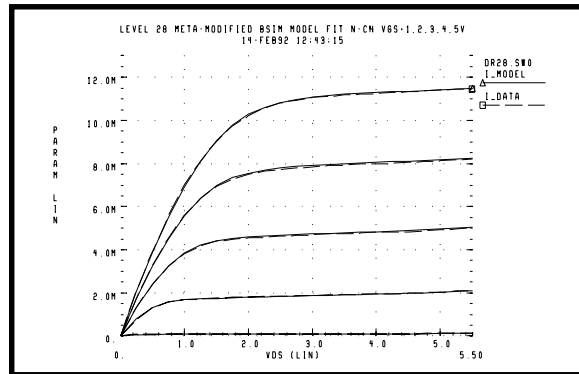
**Figure 16-14: – Level 28 Ids Model vs. Data Curves**
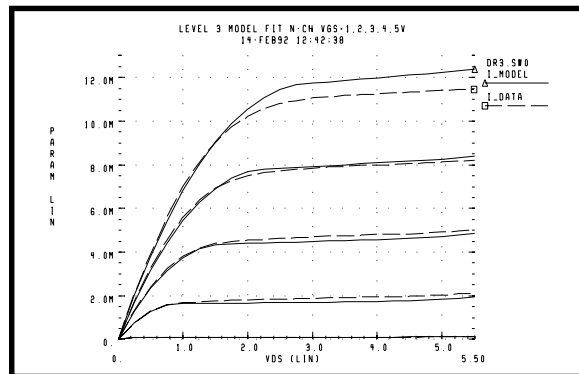


**Figure 16-15: – Level 3 Ids Model vs. Data Curves**

## Level 13, 28, 39 - Ids Model vs. Data

- Ids vs. Vds at Vgs= 1, 2, 3, 4, 5, Vbs=0
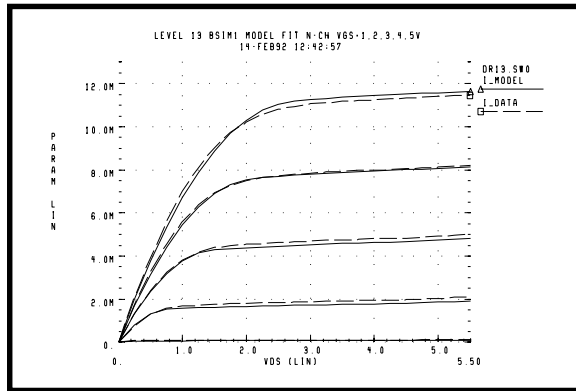
**Figure 16-16: – Level 13 Ids vs. Vds Curves**
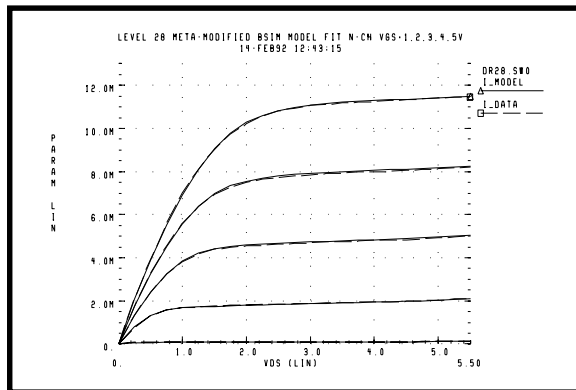


**Figure 16-17: – Level 28 Ids vs. Vds Curves**
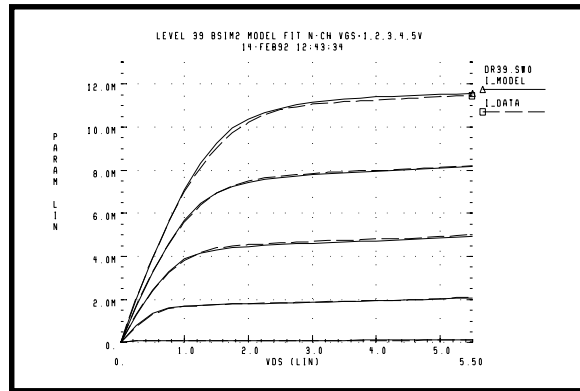
**Figure 16-18: – Level 39 Ids vs. Vds Curves**

## 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.
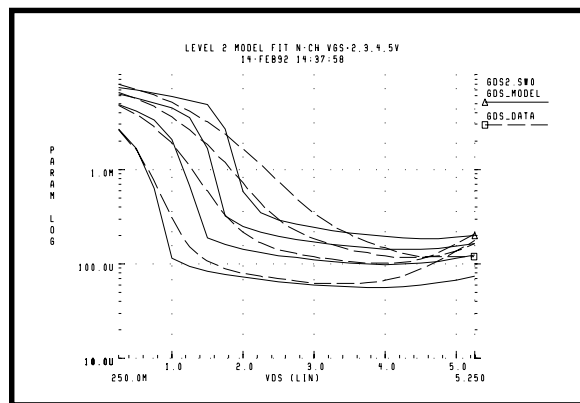


**Figure 16-19: – Level 2 gds vs. Vds Curves**
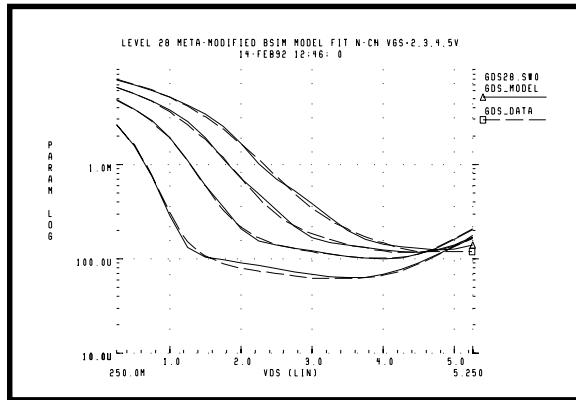
**Figure 16-20: – Level 28 gds vs. Vds Curves**
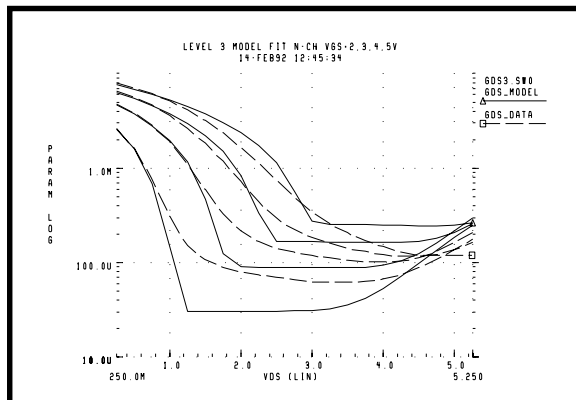


**Figure 16-21: – Level 3 gds vs. Vds Curves**

### 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.
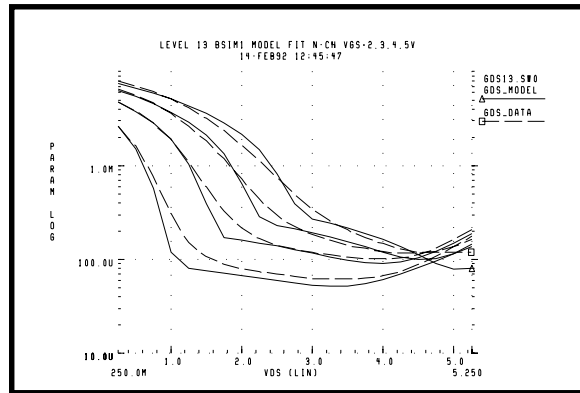
**Figure 16-22: – Level 13 gds vs. Vds Curves**
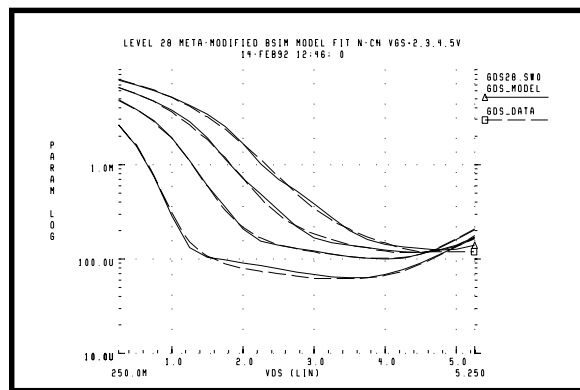


**Figure 16-23: – Level 28 gds vs. Vds Curves**
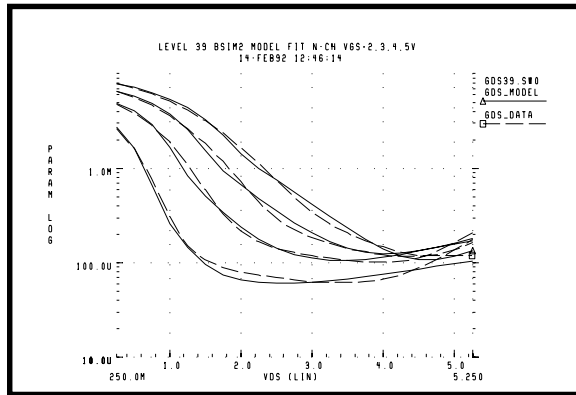
**Figure 16-24: – Level 39 gds vs. Vds Curves**

## Level 2, 3, 28 - Ids Model vs. Data

- ■ Ids -vs.- Vgs at Vds=0.1, Vgs =0, -1, -2, -3, -4
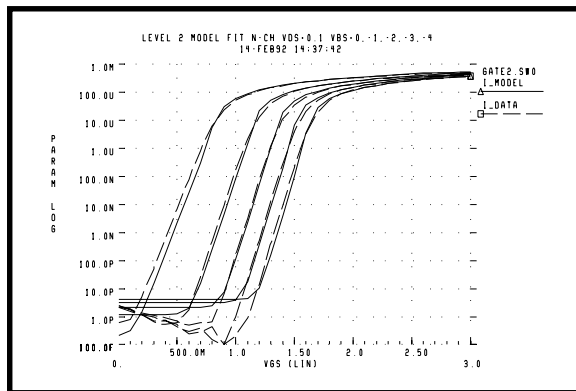


**Figure 16-25: – Level 2 Ids vs. Vgs Curves**
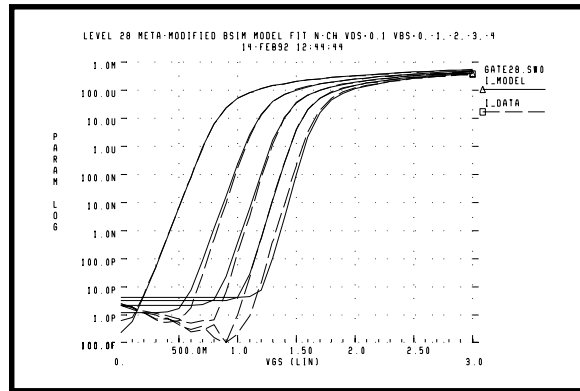
**Figure 16-26: – Level 28 Ids vs. Vgs Curves**
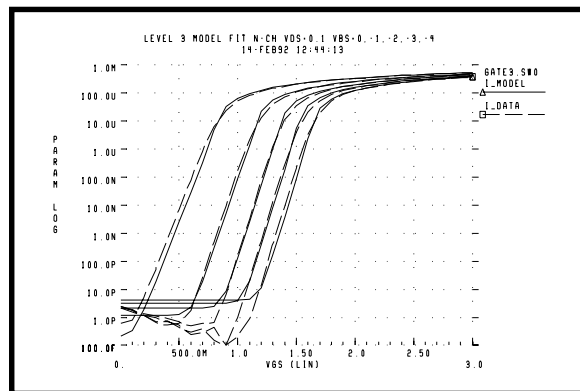


**Figure 16-27: – Level 3 Ids vs. Vgs Curves**

## Level 13, 28, 39 - Ids Model vs. Data

■ Ids -vs.- Vgs at Vds=0.1, Vbs =0, -1, -2, -3, -4
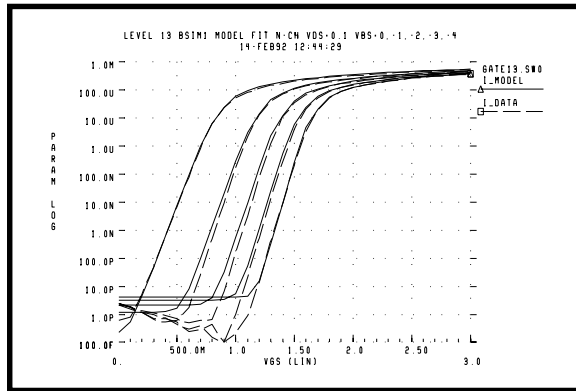
**Figure 16-28: – Level 13 Ids vs. Vgs Curves**
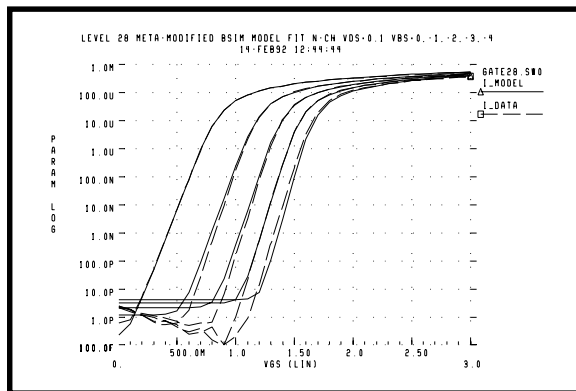


**Figure 16-29: – Level 28 Ids vs. Vgs Curves**
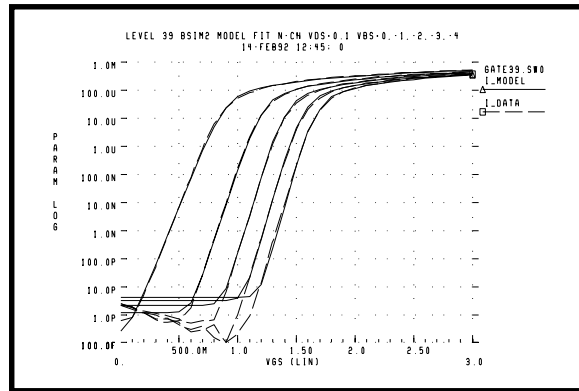
**Figure 16-30: – Level 39 Ids vs. Vgs Curves**

## 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.
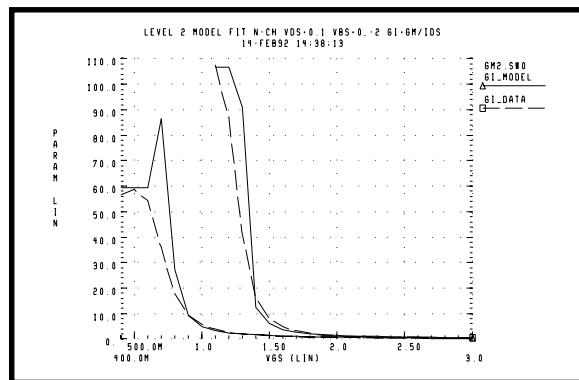


**Figure 16-31: – Level 2 gm/Ids vs. Vgs Curves**
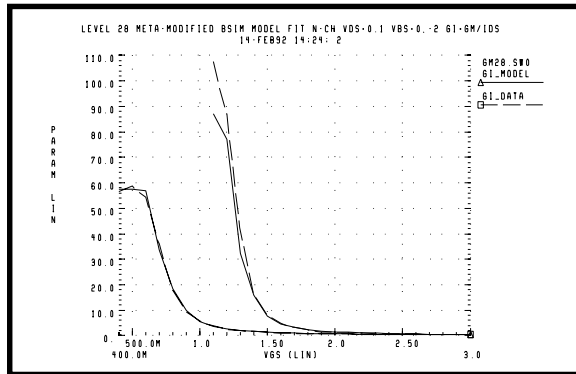
**Figure 16-32: – Level 28 gm/Ids vs. Vgs Curves**
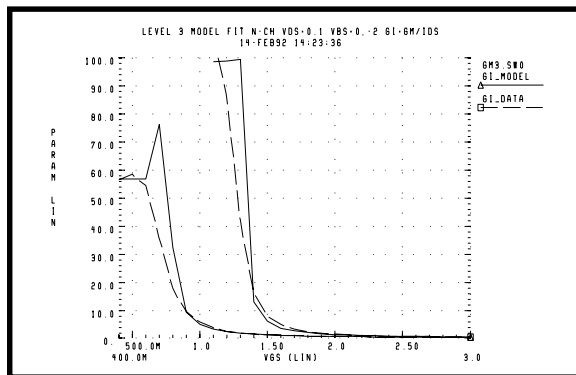


**Figure 16-33: – Level 3 gm/Ids vs. Vgs Curves**

### 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.
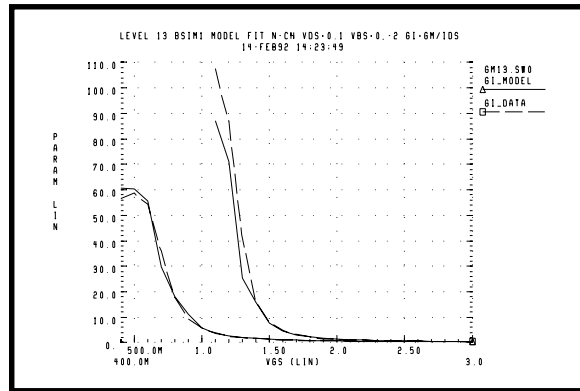
**Figure 16-34: – Level 13 gm/Ids vs. Vgs Curves**
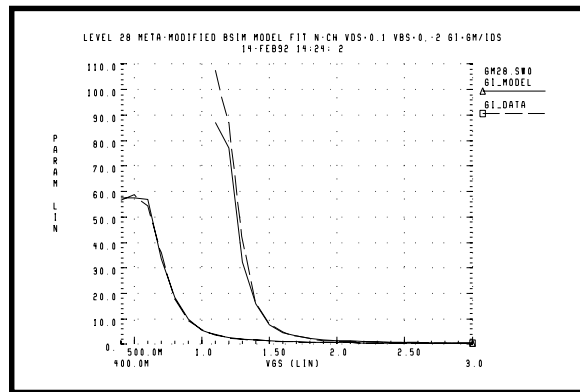


**Figure 16-35: – Level 28 gm/Ids vs. Vgs Curves**
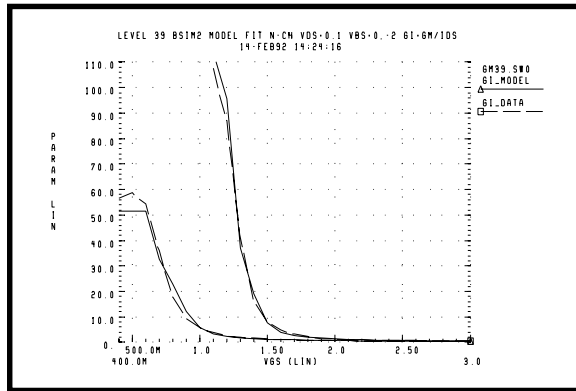
**Figure 16-36: – Level 39 gm/Ids vs. Vgs Curves**

## 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.
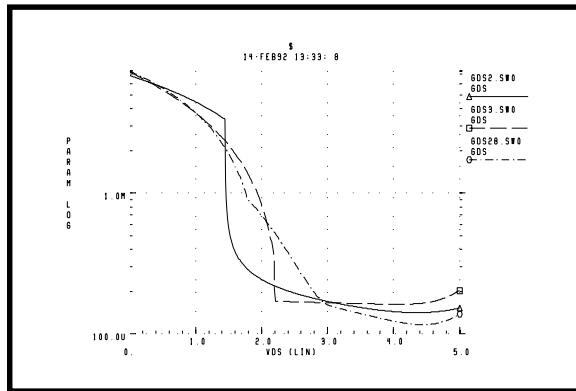


**Figure 16-37: – Levels 2, 3, 28 gds vs. Vds Curves**
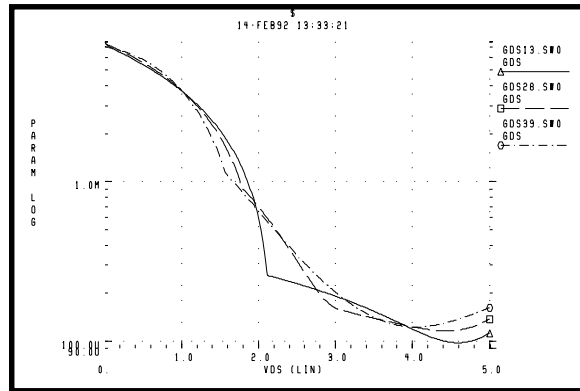
**Figure 16-38: – Levels 13, 28, 39 gds vs. Vds Curves**

## 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.
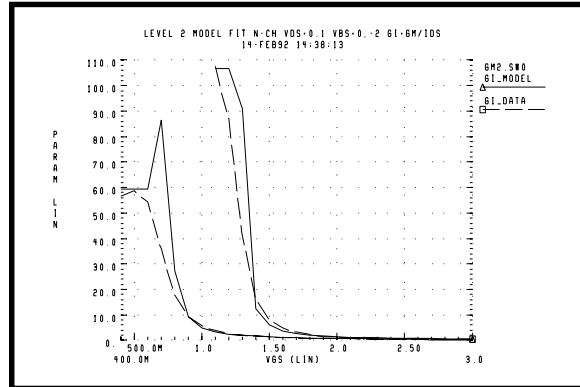


**Figure 16-39: – Level 2 gm/Ids vs. Vgs Curves**
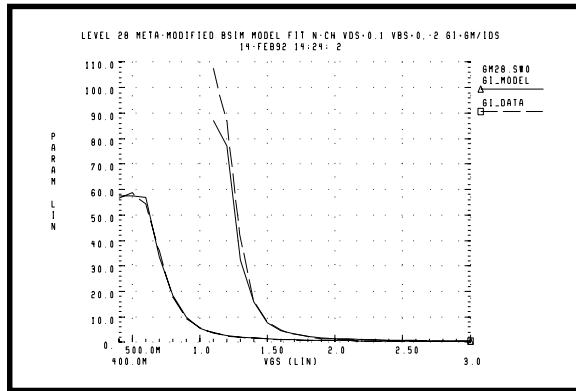
**Figure 16-40: – Level 28 gm/Ids vs. Vgs Curves**

### 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.
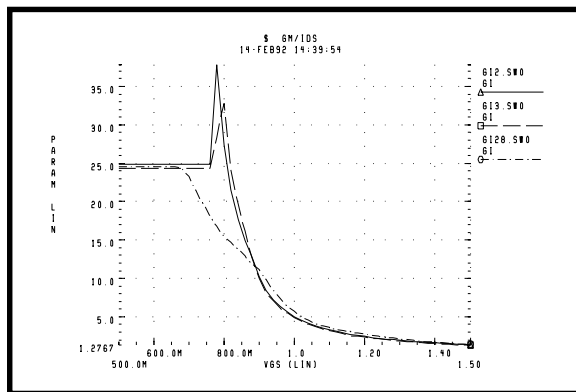


**Figure 16-41: – Levels 2, 3, 28 gm/Ids vs. Vgs Curves**
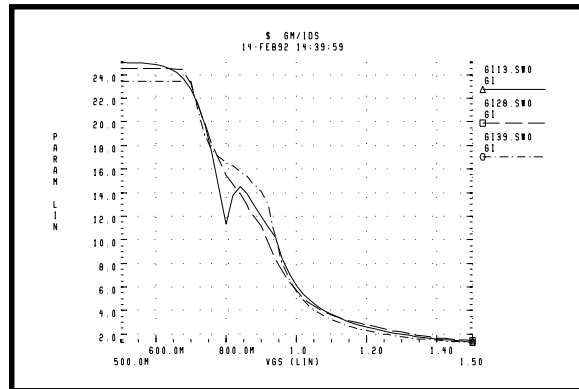
## Figure 16-42: – Levels 13, 28, 39 gm/Ids vs. Vgs Curves

**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, No3, pp. 313-315, 1985.

6. Ballay, N. et. al., "Analytic Modeling of Depletion-Mode MOSFET with Short- and Narrow-Channel Effects,*" IEEE PROC, V*ol.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 17

# Using the Bipolar Transistor Model -VBIC

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 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

■ Avalance multiplication in collector junction, parasitic capacitances neof base-emitter overlap in double poly BJTs, and self heating.

This chapter covers the following topics:

■ Understanding the History of VBIC

■ Examining VBIC Equations

■ Performing Noise Analysis

■ Using VBIC

# Understanding the History of VBIC

VBIC was developed by engineers at several companies and is described in the following publication, which provides a major reference to equations of the model:

> *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.*

The detailed equations for all elements are given in the above 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 except that self-heating and excess phases are not implemented or enabled.

The large signal equivalent circuit for VBIC is shown in Figure 17-1. Capacitors CBCO, CBEO and resistors RCX, RBX, RE, and RS are linear elements, all other elements of the equivalent circuit are nonlinear.

# Examining VBIC Equations

Table 17-1 lists the parameters for the model that you can set. Table 1-1 also contains the default values for the parameters. The same names are used for parameters in the table and in the above referenced publication.

**Figure 17-1: 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.

# Performing 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.

**Examples**

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
```

### Table 17-1: Default Model Parameters for BJT, Level 4

| Name | Units | Default | Comments |
| --- | --- | --- | --- |
| *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 |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *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 |
| *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 |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *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 |
| *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 |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| 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 |
| TNF | 1/K | 0 | Temperature coefficient of NF |
| TR | s | 1e-11 | Reverse transit time |
| TREF (TNOM) | ºC | 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 |

| Name | Units | Default | Comments |
|------|-------|---------|----------|
| *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 |

# Using VBIC

The following are tips you can follow when using VBIC.

1. Set Level to 4 to identify the model as a VBIC bipolar junction transistor model.

2. Self-heating has not been enabled for this model, so model parameters RTH and CTH have no influence.

3. The Level 4 model does not scale with any area terms, and does not yet scale with M.

4. The HSPICE option SCALM is ignored for this model.

5. Setting these parameters to zero infers a value of infinity: HRCF, IKF, IKP, IKR, ITF, VEF, VER, VO, VTF.

6. 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 HSPICE.

7. The default room temperature is 25 in 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 HSPICE versus other simulators that use 27 as the default room temperature.

8. Pole-zero simulation of this model is not supported.

9. Excess phase has not yet been implemented.

10. 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 $1.0e^{-3.}$

11. This implementation is a preliminary version only. It is not to be used for new designs, and has not yet been optimized for speed and memory usage.

*Avant!*

# Chapter 18

# 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 18-1 illustrates the model selection method.
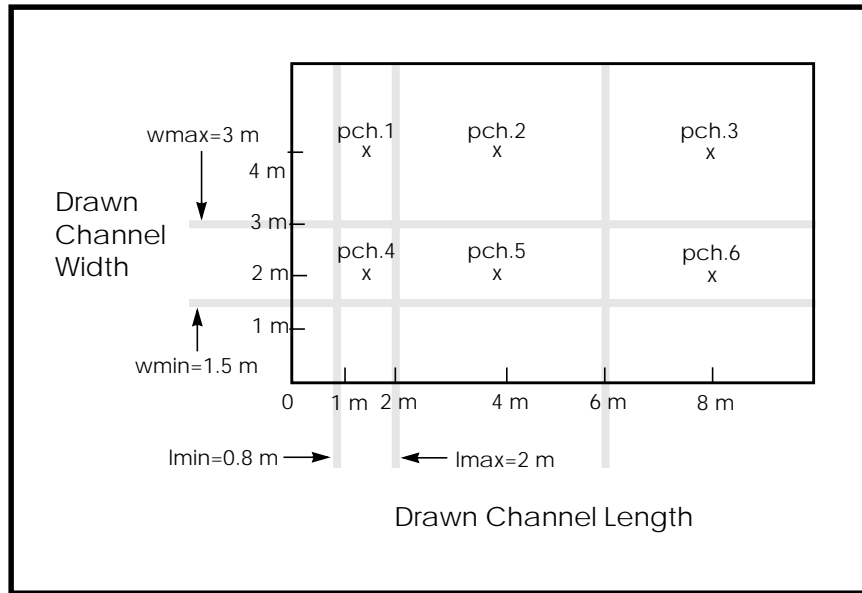
**Figure 18-1: Automatic Model Selector Method**

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 |

| | | | | |
|---|---|---|---|---|
| 113090 | LF155 | LF155A | LF156 | LF156A |
| LT1028A | LT1037 | LT1037A | LT1055 | LT1055A |
| LT1055S8 | LT1056 | LT1056A | LT1056S8 | LT1057 |
| 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 | PCI_II_B |

## 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!*

# Chapter 19

# 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.

- Performing 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.

- Using Digital File Input Stimuli
  You can use logic state transition tables to produce the input stimuli for the characterization. The D2A model in Star-Hspice provides a 28-state logic simulator interface for rapid figuring of a cell characterization testbed.

# 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
```
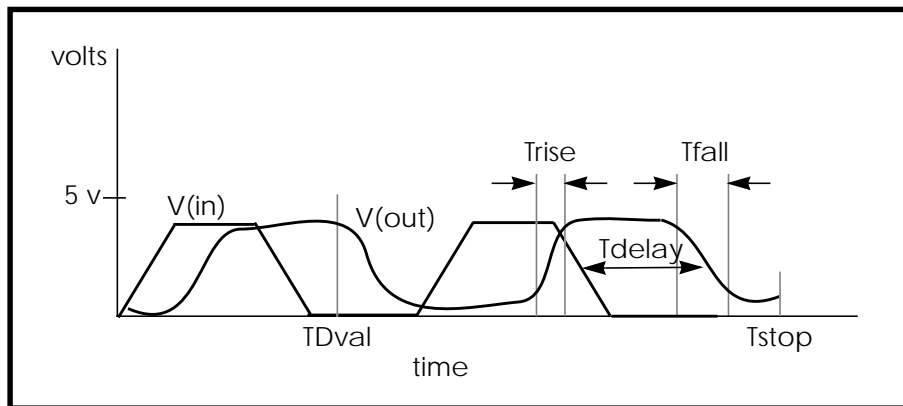


**Figure 19-1: Rise, Fall, and Delay Time Demonstration**

## 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'
```
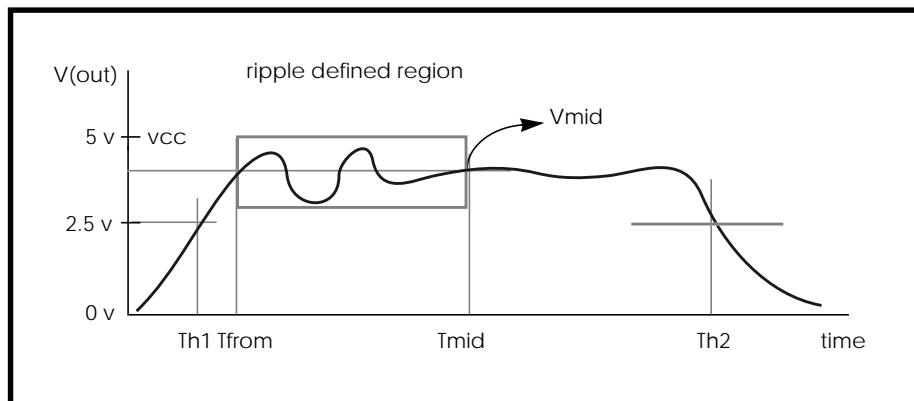


**Figure 19-2: Waveform to Demonstrate Ripple Calculation**

## 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 10-33.

**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
```
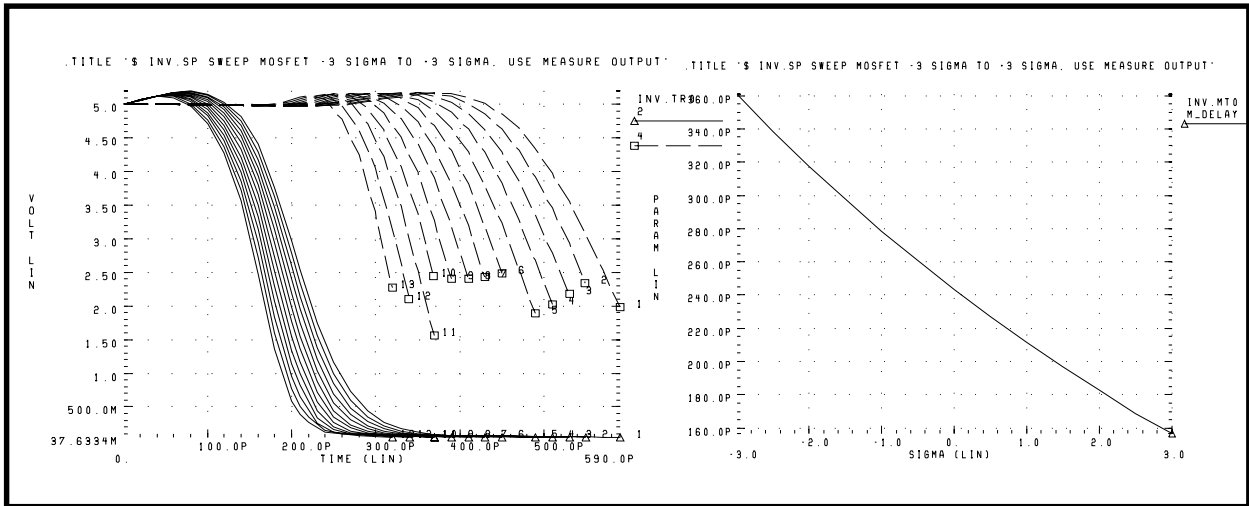


**Figure 19-3: Inverter Pair Transfer Curves and Sigma Sweep vs. Delay**

## Delay versus Fanout

This example sweeps the subcircuit multiplier to quickly generate a family of
five load curves. By buffering the input source with one stage, more accurate
results are obtained. The 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
```
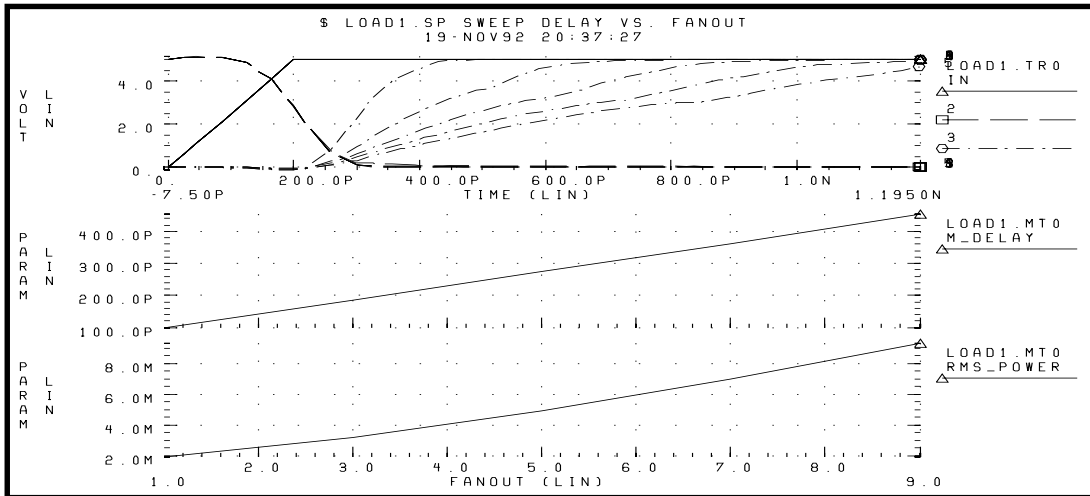
**Figure 19-4: Inverter Delay and Power versus Fanout**

## 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
```
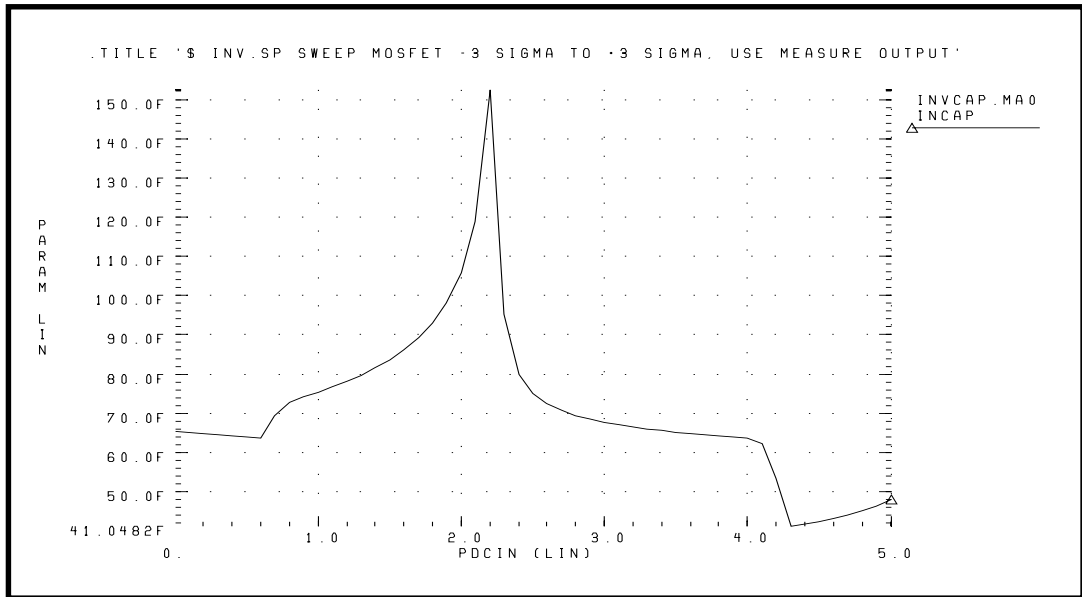
**Figure 19-5: Graph of Pin Capacitance versus Inverter Input Voltage**

## 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
```
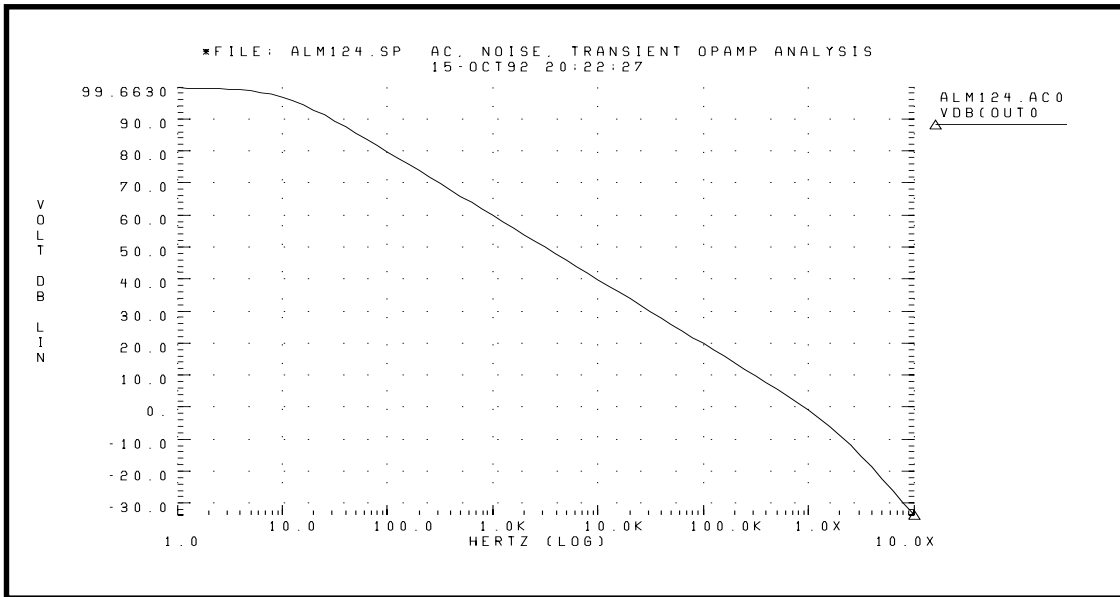


**Figure 19-6: Magnitude Plot of Op-Amp Gain**

# Performing Data Driven Analysis

Data driven analysis allows you to simultaneously modify any number of parameters, then perform an operating point, DC, AC, or transient analysis. The parameter value array is either included in the simulation input file directly or is stored as an external ASCII file. The .DATA statement associates the parameters with the value array, and it replaces the .PARAM statement.

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

*Note:  The .DATA statement format is almost the same as the measure output format. It does not require the "+" sign to continue a line.*

The syntax is:
```
.DATA dataname pname1 pname2 ... pnamen
val11 val12 ... val1n
    …
valm1 valm2 ... valmn
.ENDDATA
```
or
```
.DATA dataname Merge file=filename pname1=column1
pname2=column2 ...
.ENDDATA
```

The first general form specifies *n* parameters with *m* iterations. The number of parameters is not limited, and this number determines the number of values taken per iteration. There is no requirement that all parameter values be on the same physical line. The .DATA statement syntax is covered in greater detail in *Chapter 4, Specifying Simulation Output*.

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
```

The .MEASURE statement computes rise, fall, and propagation delay times. For cell characterization, many measurements are necessary because of changing specifications for load capacitance, fanout, temperature, and so on. This process can be very time consuming. Use the AUTOSTOP option to stop the transient analysis when all the rise, fall, and delays specified in the .MEASURE statements are calculated. This option saves a great deal of CPU time.

## Cell Characterization Example

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 19-7: and Figure 19-8: are identical except that their input signals are complementary. The circuit in Figure 19-7: calculates the rise time and the low-to-high propagation delay time. The circuit in Figure 19-8: 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.
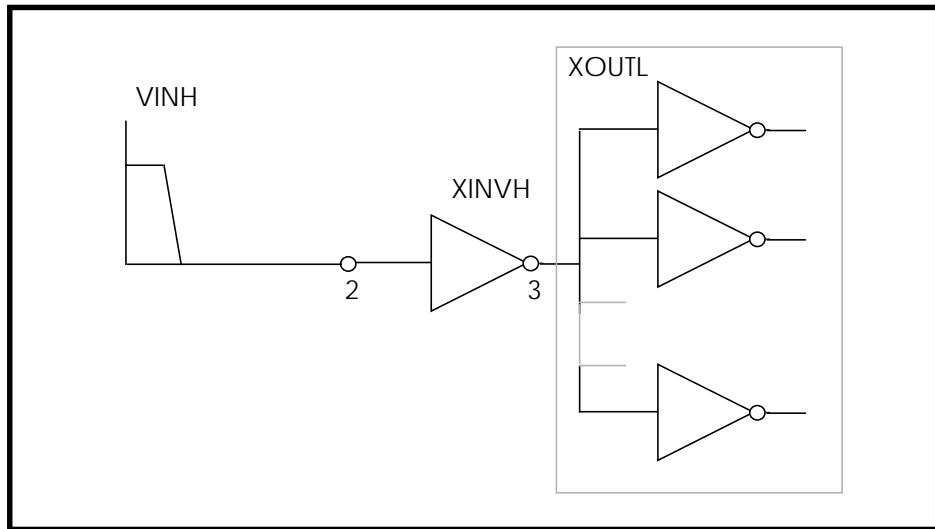
**Figure 19-7: Cell Characterization Circuit 1**
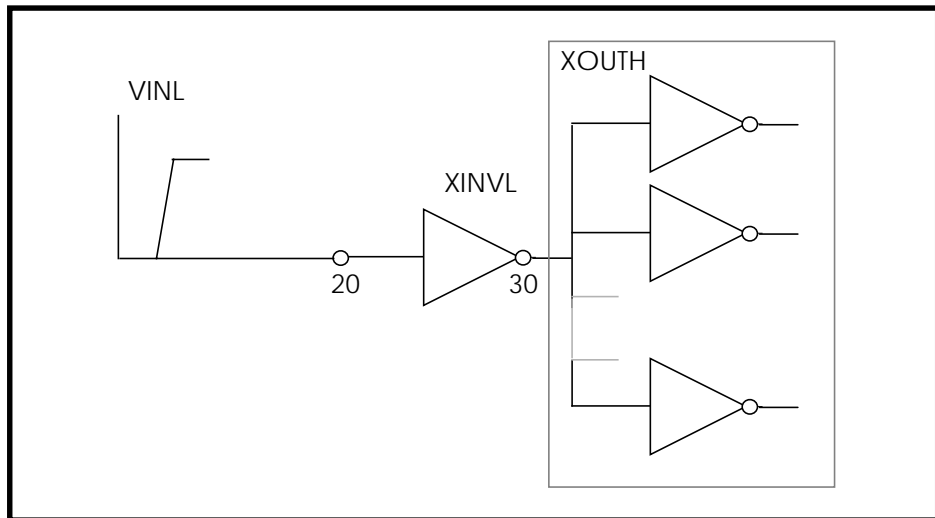


**Figure 19-8: Cell Characterization Circuit 2**

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, $V(3) = 0.1\text{VDD}$) and ends when the voltage at node 3 is equal to $0.9 \cdot \text{VDD}$ (that is, $V(3) = 0.9\text{VDD}$).

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
```
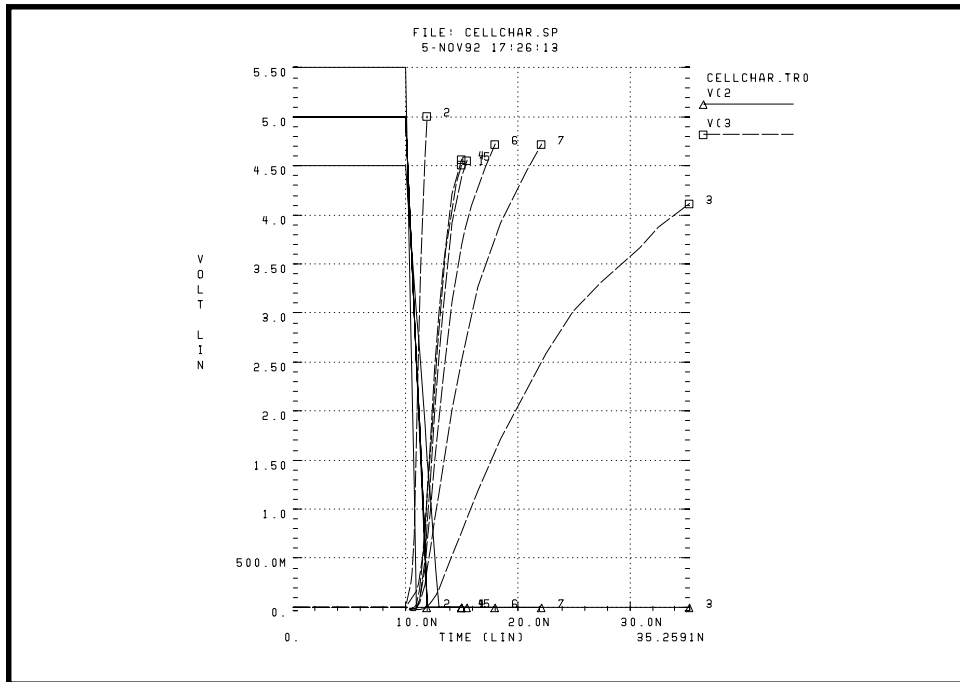
**Figure 19-9: Plotting the Simulation Outputs**
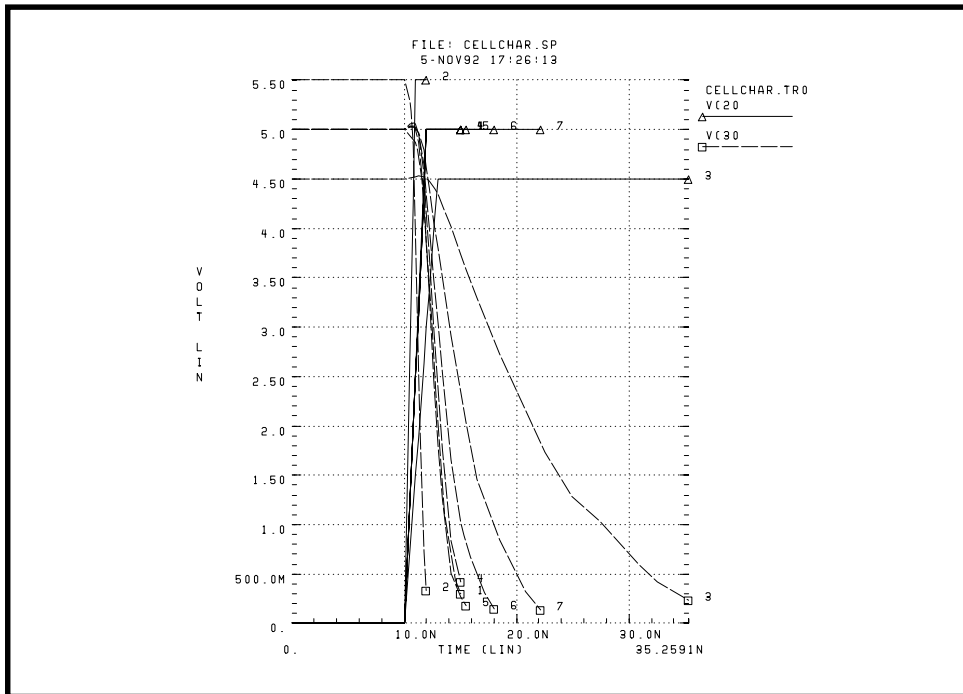
**Figure 19-10: Verifying the Measure Statement Results by the Plots**

# Using Digital File Input Stimuli

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. The example outputs a digital file.
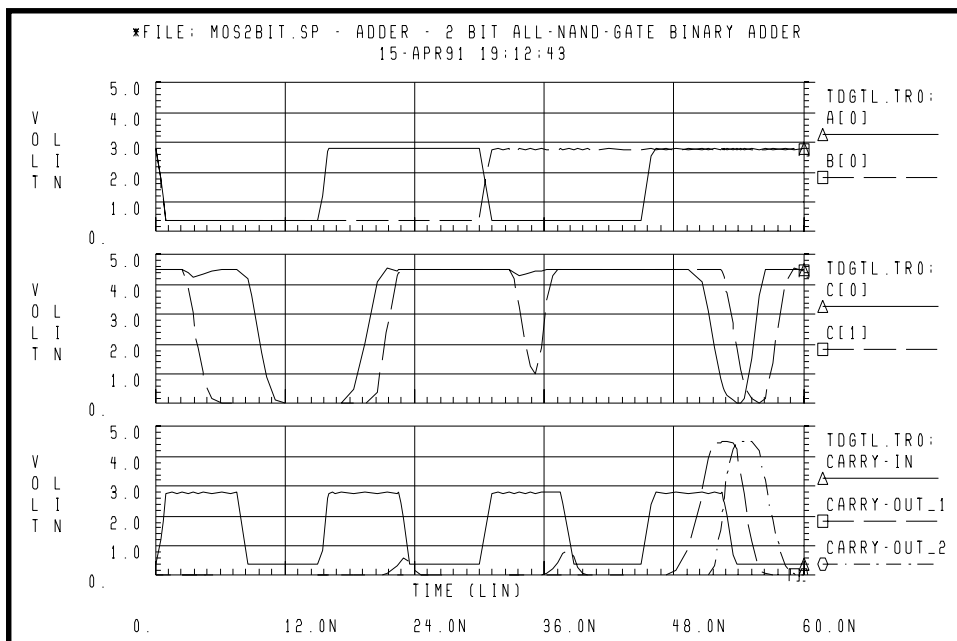


**Figure 19-11: Digital Stimuli File Input**

The simulation above uses the digital-to-analog interface model. The example, found in the demo directory *$installdir/demo/hspice/cchar/tdgtl.sp*, shows a way of generating stimuli using an external stimuli file produced by a logic simulator (in this case, the Viewsim simulator).

Top-down design generally starts with a system-level hardware description language (HDL) description of the circuit. This is decomposed to the logic cell level, and cells are then synthesized into transistor level circuits. Since simulation has been done at the logic level, it is possible to capture all of the basic input stimuli to the cell. With the integrated 28-state logic interface, Star-Hspice enables you to reuse the logic output as circuit simulator stimuli input.

## Replacing Sources With Digital Inputs

Traditional voltage pulse sources become ...

```
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
```

... D2A drivers that get their input from ...

```
UC carry-in VLD2A VHD2A D2A SIGNAME①  IS=0
UA[0] A[0] VLD2A VHD2A D2A SIGNAME②  IS=1
UA[1] A[1] VLD2A VHD2A D2A SIGNAME③  IS=1
UB[0] B[0] VLD2A VHD2A D2A SIGNAME④  IS=1
UB[1] B[1] VLD2A VHD2A D2A SIGNAME⑤  IS=1
```

... Digital Stimuli 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
```

**Figure 19-12: Digital File Signal Correspondence**

### Example

```
.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
```

_Avant!_

# Chapter 20

# 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 you 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-peed components use Star-Hspice already.
- The complexity can be hidden from the systems 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 20-1:.

### Table 20-1:  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 20-1: demonstrates the analog behavior of a digital gate simulated in the Star-Hspice circuit simulator.
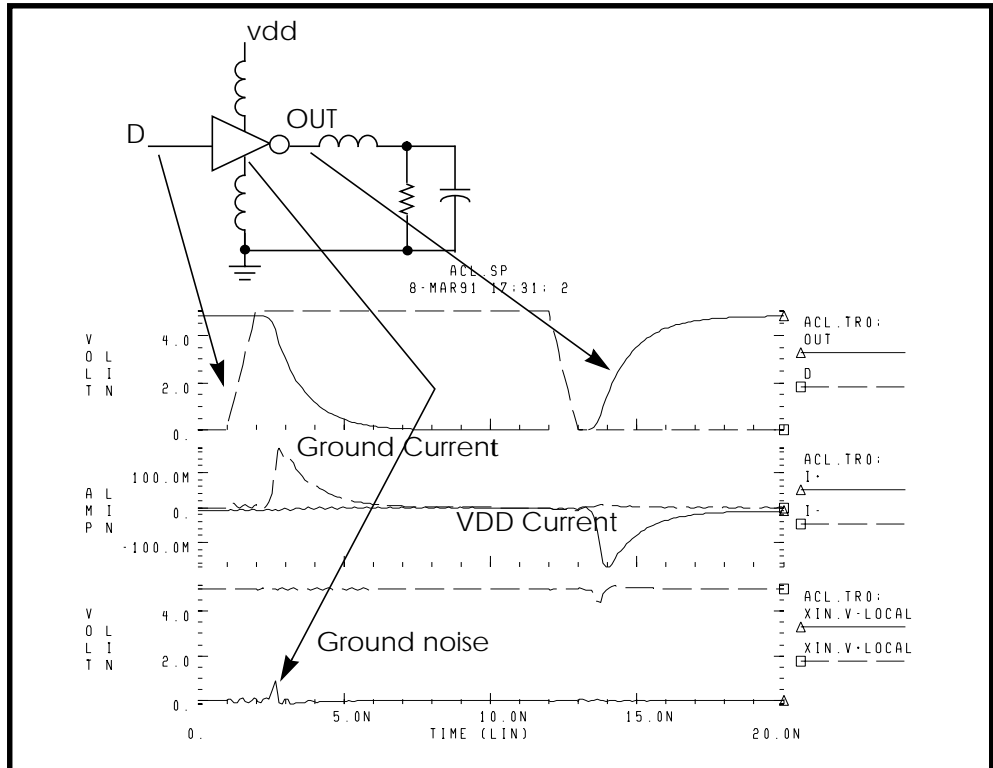
**Figure 20-1: Simulation of Output Buffer with 2 ns Delay and 1.8 ns Rise and Fall Times**

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.
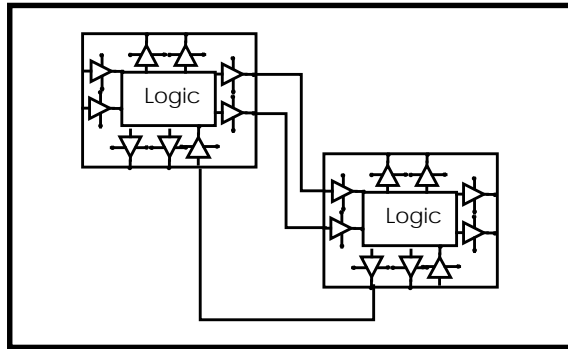


**Figure 20-2: Analog Drivers and Wires**

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.
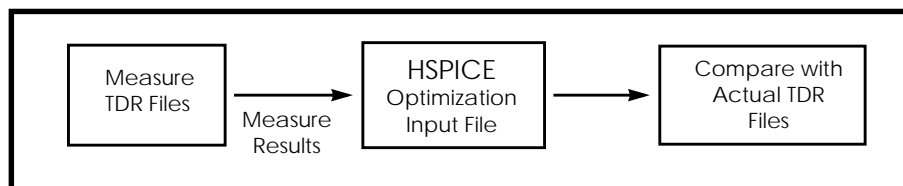


**Figure 20-3: Optimization Process**
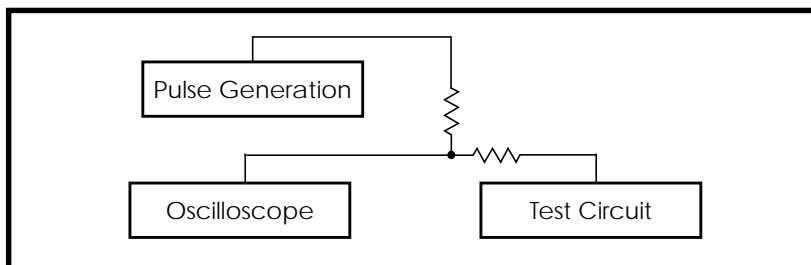
**Figure 20-4: General Method for TDR Optimization**

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
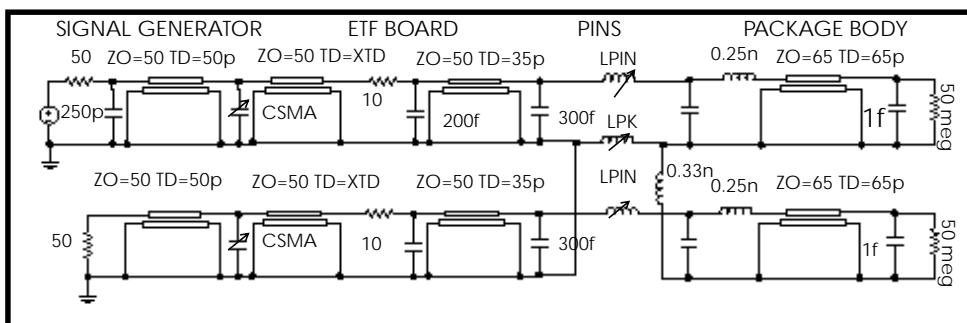


**Figure 20-5: SPICE Model for Package-Plus-Test Fixture Optimized Parameters: XTD, CSMA, LPIN, and LPK**

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 20-6 through 20-9.

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
```
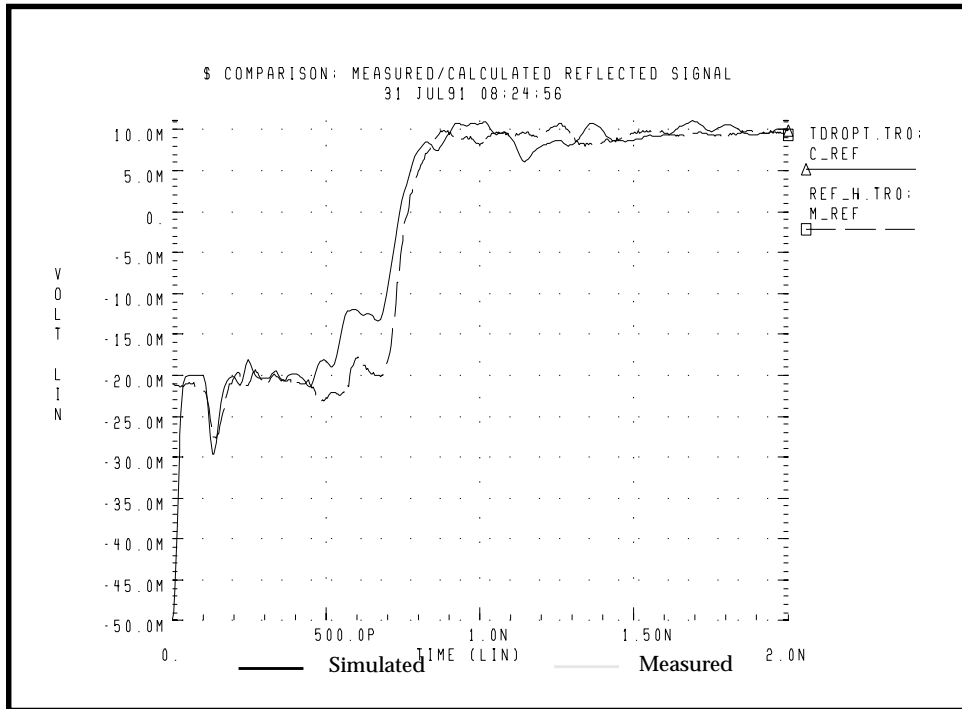
**Figure 20-6: Reflected Signals Before Optimization**
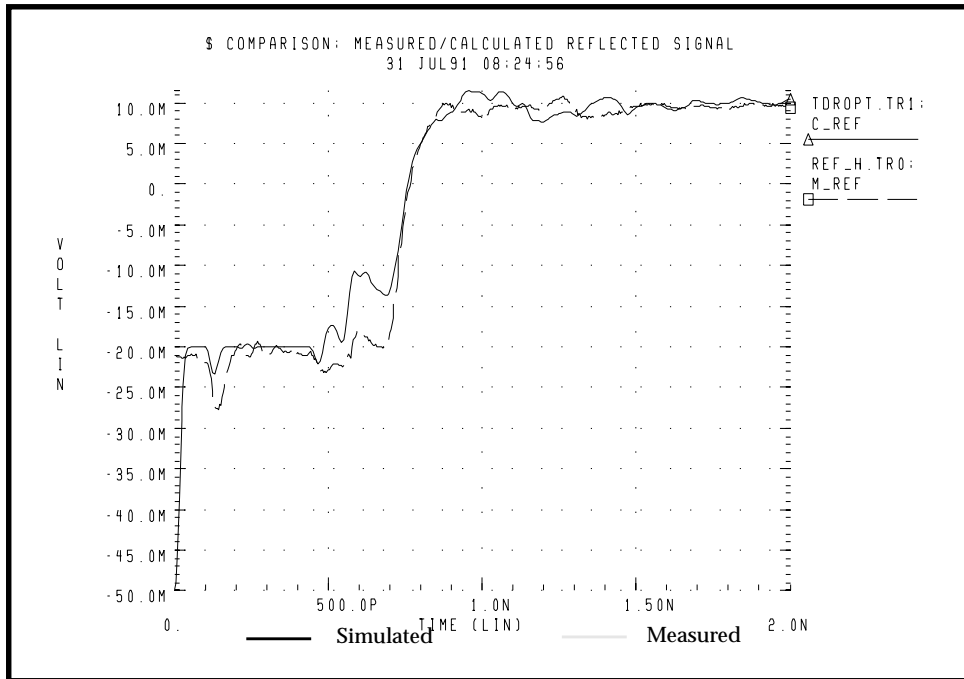
**Figure 20-7: Reflected Signals After Optimization**
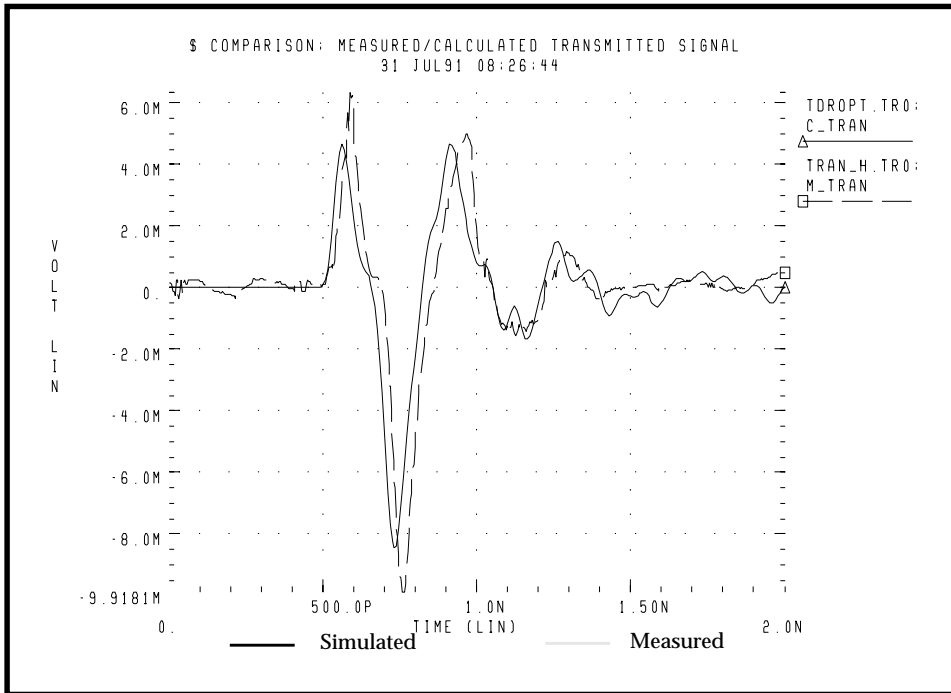
**Figure 20-8: Transmitted Signals Before Optimization**
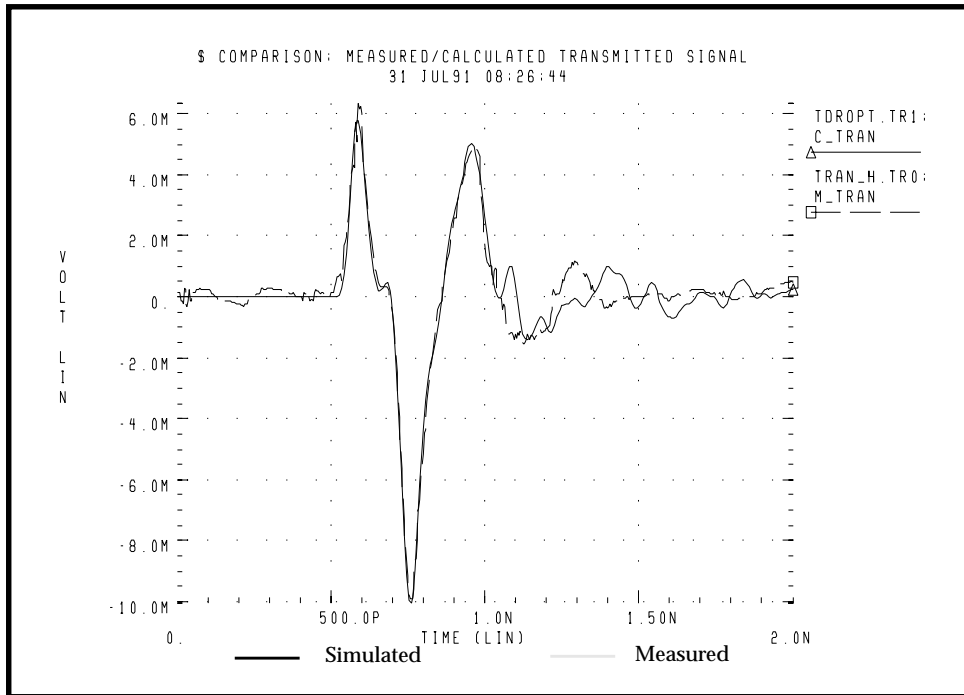
**Figure 20-9: Transmitted Signals after Optimization**

# Simulating Circuits with Signetics Drivers

The Signetics I/O buffer library is distributed with Star-Hspice in the *$installdir/parts/signet* directory. These are high performance parts used in backplane design. The transmission line model used describes two conductors.
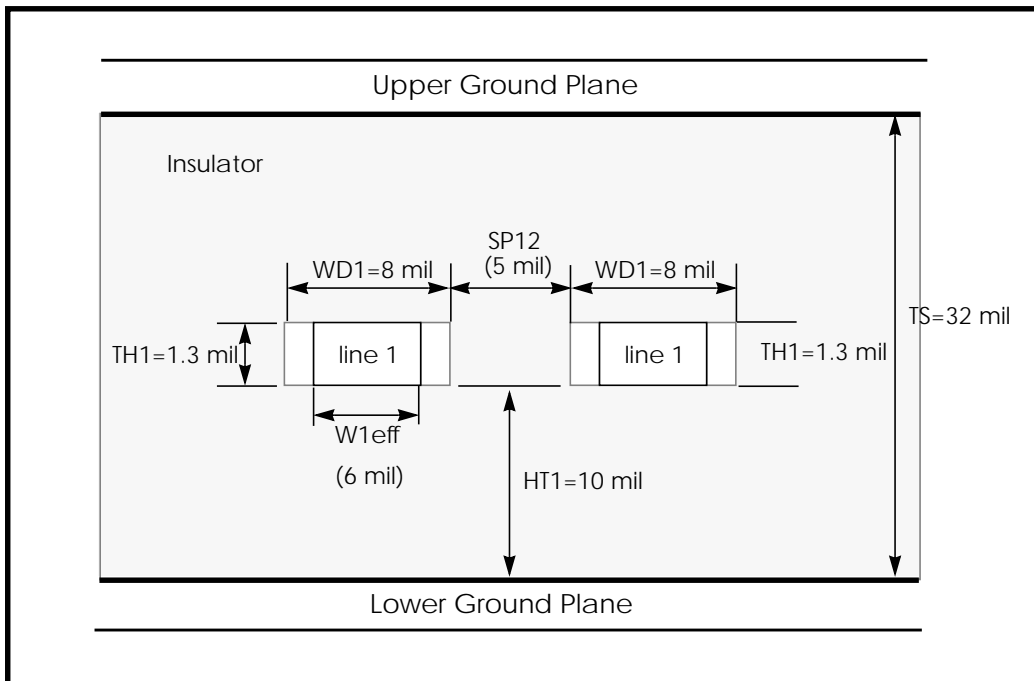


**Figure 20-10:  Planar Transmission Line DLEV=0: Microstrip Sea of Dielectric**

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.
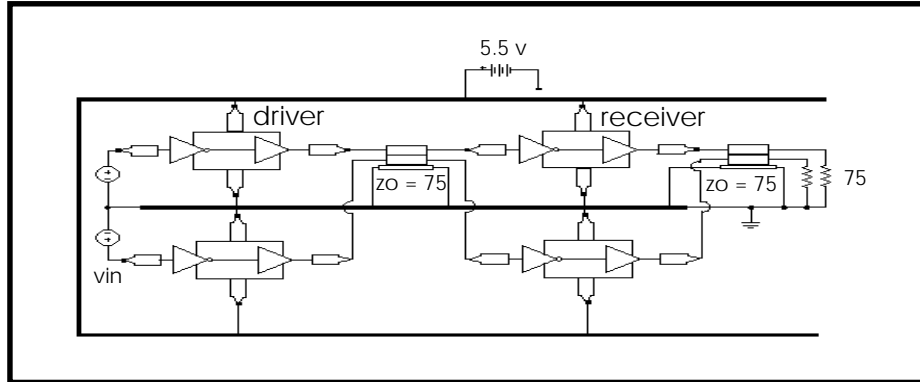
**Figure 20-11: I/O Drivers/Receivers with Package Lead Inductance, Parallel 4" Lossy Microstrip Connectors**

# Example of Connecting I/O Chips With Tlines

```
.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
```
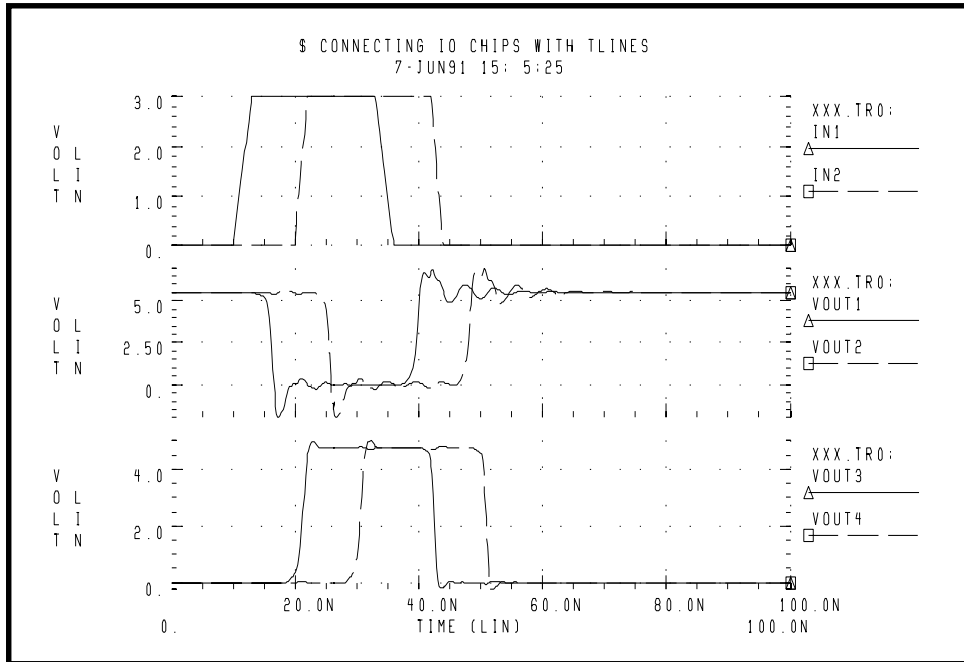
**Figure 20-12: Connecting I/O Chips with Transmission Lines**

# 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 (5 V enables) |
| PDOWN (IOB4 only) | pad pull-down 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.

## Example of 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.
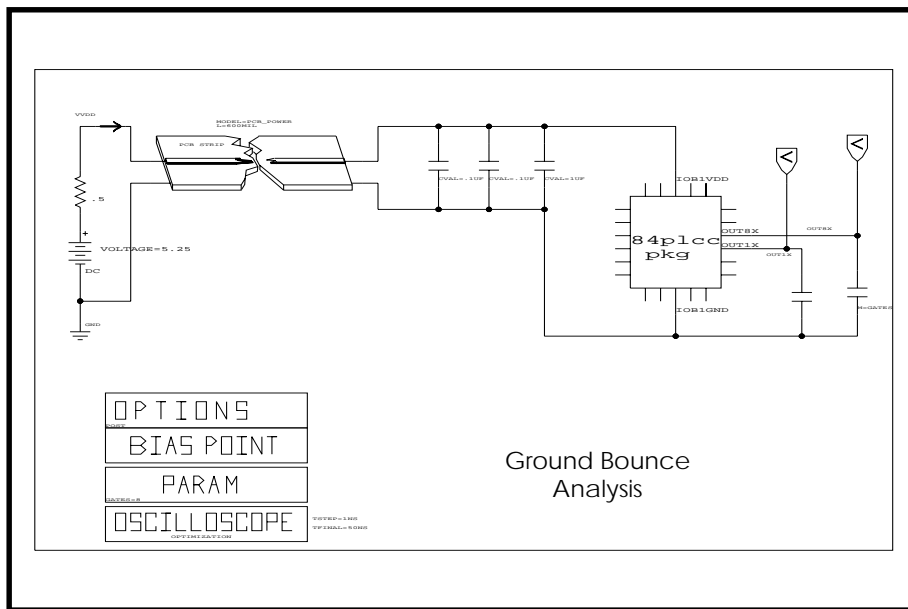
**Figure 20-13: Ground Bounce Simulation**

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
```
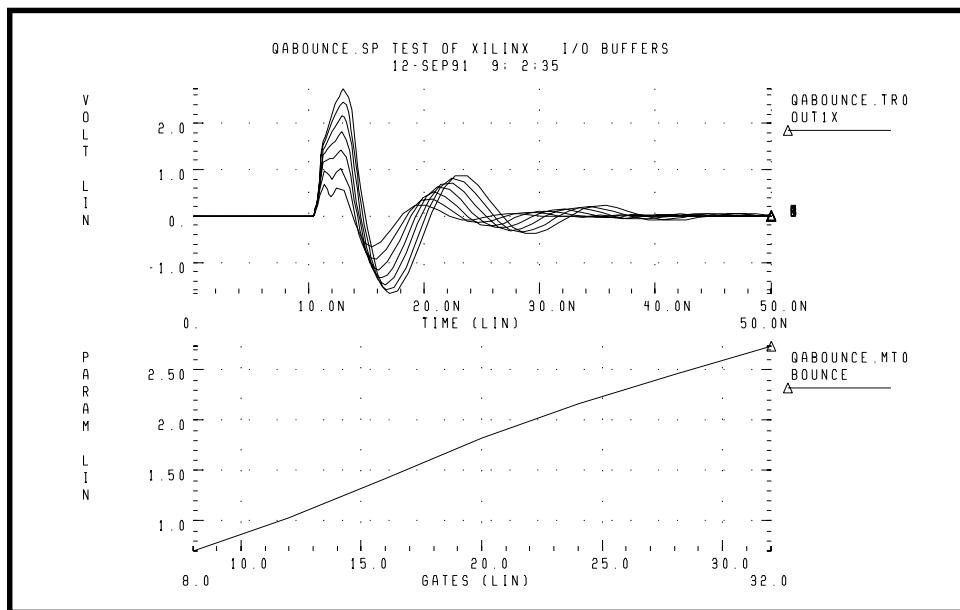
**Figure 20-14: Results of Ground Bounce Simulation**

## Example of 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.
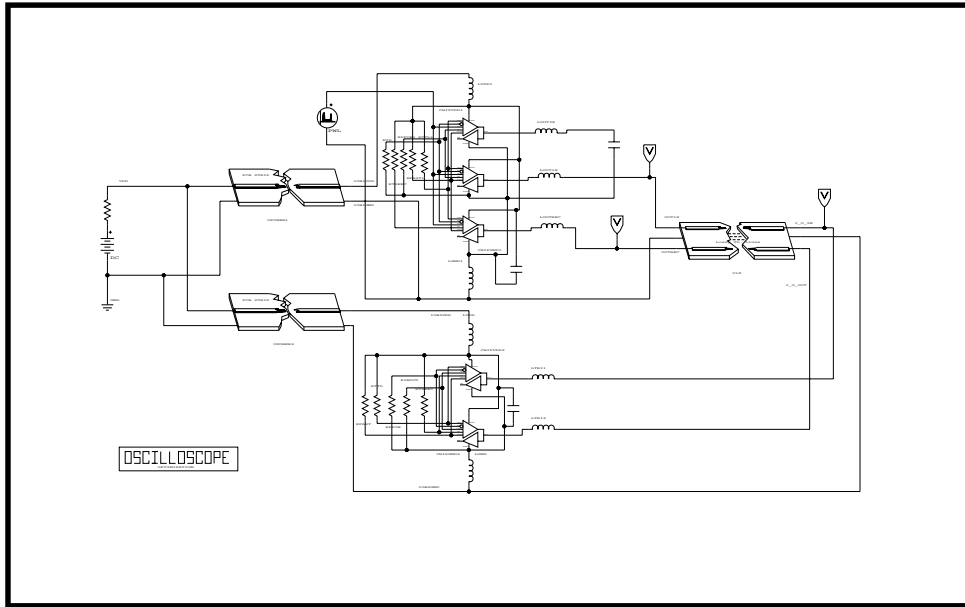
**Figure 20-15: Coupled Noise Simulation**

## Coupled Noise Star-Hspice File

```
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
```
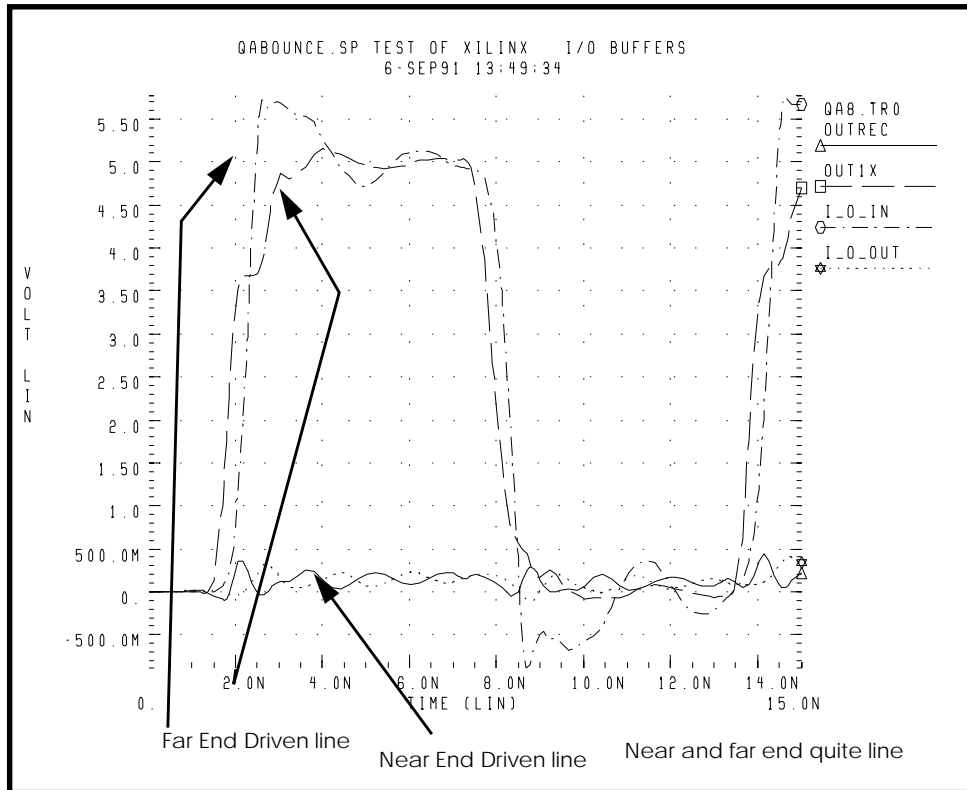


**Figure 20-16: Results of Coupled Noise Simulation**

## Sample of 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 20-17:). 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.
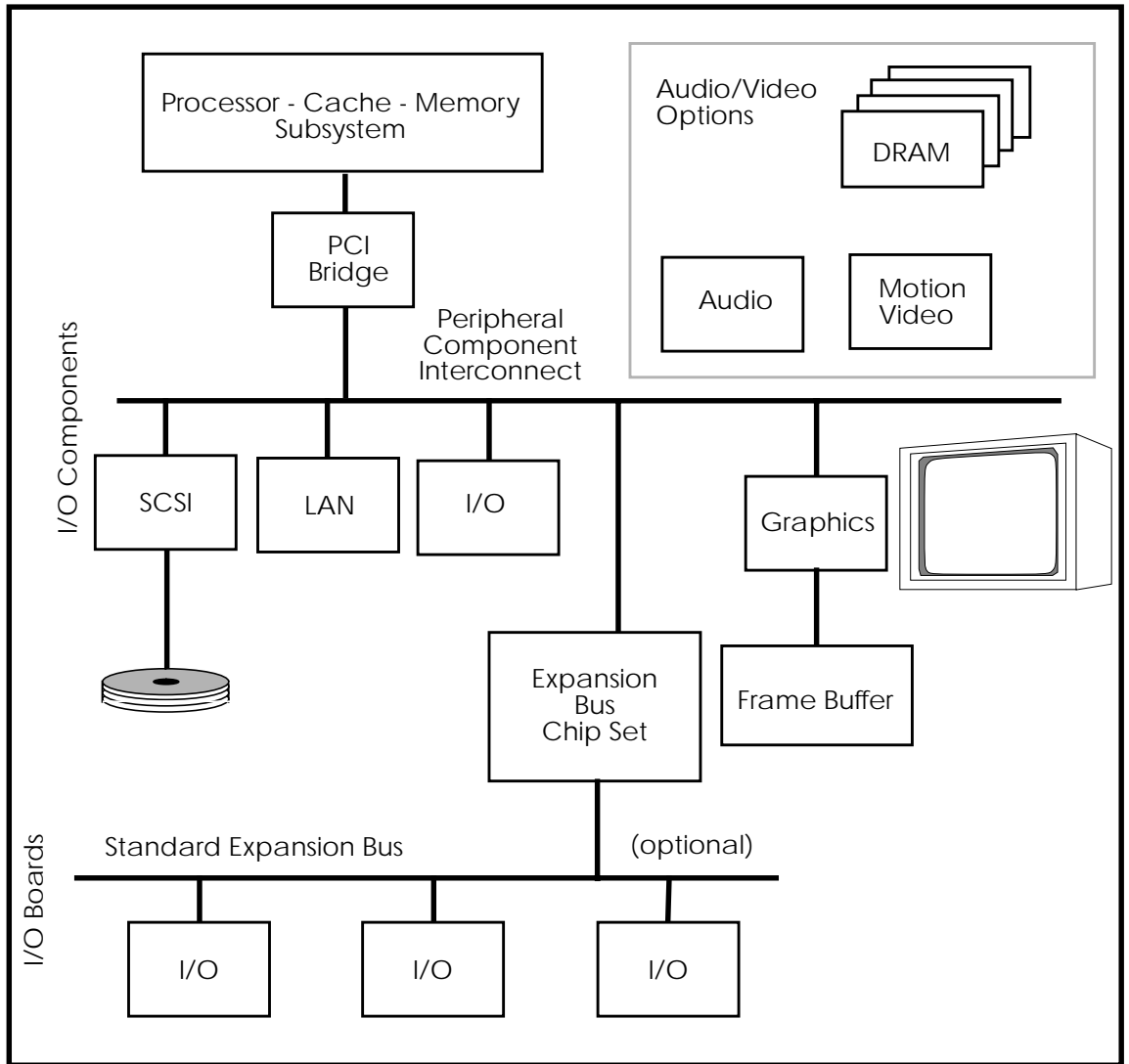
**Figure 20-17: PCI System Block Diagram**

## 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.

## The PCI Speedway Star-Hspice Model

Intel's PCI Speedway is a recommended method for interconnecting PCI devices, as shown in Figure 20-18:. The Speedway is not the only way to interconnect PCI devices, but it has proven to be a robust solution.
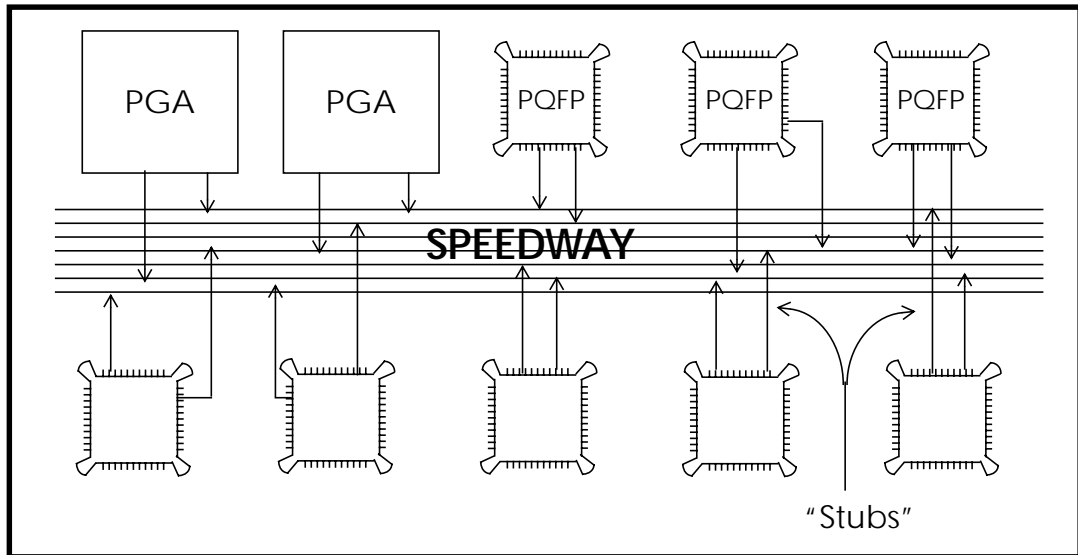
**Figure 20-18:  PCI Speedway**

Figure 20-18: 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 20-2:. 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 20-2: 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.

# Using the 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 20-49. The file is broken into six major sections, as listed below:

- Parameters
- Star-Hspice Control/Analysis Statements
- Measure Commands
- PCI Driver Selection
- PCI Speedway Subsections
- File Alter Commands

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 below:

```
.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 20-19:.
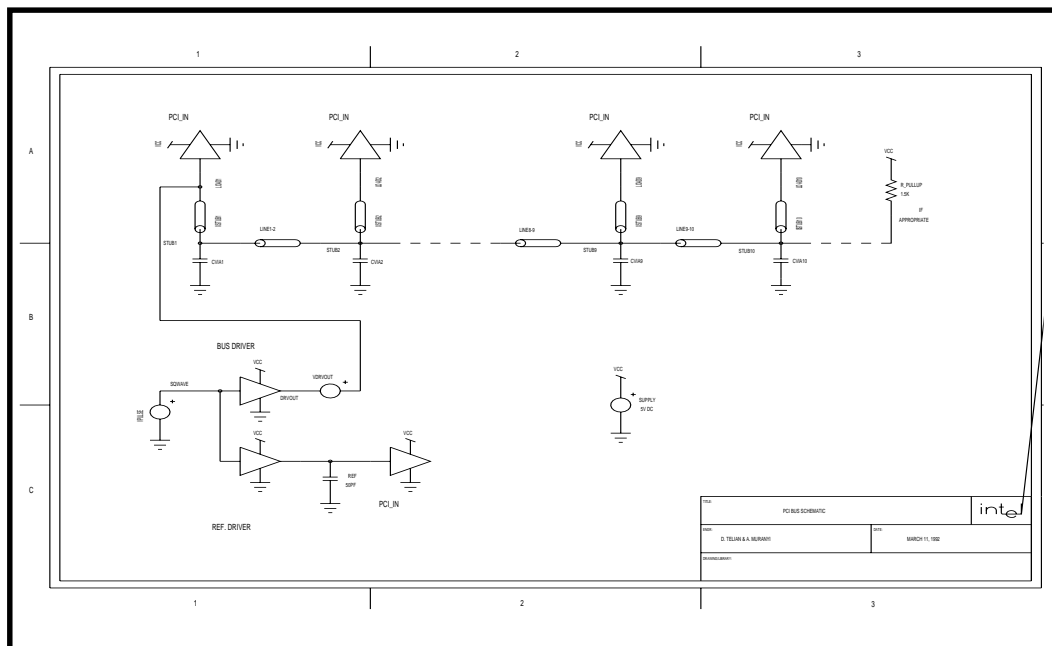


**Figure 20-19:  PCI Speedway Circuit Schematic**

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
********************************************************************
```

## Star-Hspice/Excel Graph Conversion Macro

To develop a PCI bus, you must run simulations of the board architecture to ensure that the *T_prop* timing matches the PCI specification. However, it is a long and tedious process to manually extract the important *T_prop* timings from the Star-Hspice data file (*.mt0*) produced from running these simulations, especially if there are many Monte Carlo iterations. An Excel macro has been developed to automate this process and extract the *T_prop* timings from the Star-Hspice data file. The macro creates a bar graph in Excel from the *T_prop* timings according to the specifications given.

There are two programs; the first is a demo called *PCIDEMO.XLA*. This demo shows tables of impedance calculations made on the Speedway, describes how the Star-Hspice/Excel graph conversion program works, and shows the *T_prop* graphs made on the Speedway using the conversion program. The second program is the Star-Hspice/Excel graph conversion program called *D_XLCONV.XLA*.

If you have the disk with the two programs, use the following steps to access the data on the disk:

1. Verify that Microsoft Excel™ is installed on your computer.

2. Start Microsoft® Windows™ and run "D_XLCONV.XLA" using either File-Run or the file manager.

   You can open Excel first and then open the conversion program inside Excel, but it is not necessary. The ".XLA" extensions are associated with Excel and automatically open Excel when they are run.

3. Enter the file name of your Star-Hspice data file.

   It is not necessary to add the ".MT0" extension to the file name since the program assumes the file is in the current directory and searches for it there. If your file is not in the current directory, Excel lets you know it cannot find it. If Excel cannot find the file, enter the path with the file name (ex: C:\DATA\EXAMPLE.MT0). You only need to enter the path one time because the program stores the path in memory. To erase this path from memory and switch back to using the current directory, type a period in front of the file name when you enter it.

4. Enter the combination of rising and falling edges your data file has in the main section and its alters.

   For example, if the first run in the *.SP file is a rising edge and the three alters are all falling edges, you would type in "RFFF."

5. Enter what position you are driving from in your simulation runs. Simply enter the number of the position (ex: "1").

   At this point, the program opens your Star-Hspice data file and prompts you for the size of divisions you want to make for your X-axis on the graph(s) to be created.

6. Choose the X-axis division size.

   Your choice affects the level of detail you want for the actual data that make up your bar graph. To use the default value (0.1 ns), press the 'enter' key. To change the tick-mark divisions on the X or Y axis, select the scale you want to change, choose "Format - Scale" and change the major and/or minor unit accordingly.

7. If your data file has one or more alters, the program prompts you to make a separate graph for each alter, or to combine all your alters into one graph. If you press 'return', Excel assumes you want a separate graph for each alter. If you want to combine all your alters into one graph, type in "2" and press 'return.'

8. After your graph has been created, enter the title of the graph.

   Type in the title you want and press 'return.' The charts are in the order of how they were run in your HSPICE *.SP file. The first run is the chart with the lowest number and the last alter is the chart with the highest number.

9. To rerun the Star-Hspice/Excel graph conversion program without exiting Excel, press "CTRL - D."

*Note:   This Star-Hspice/Excel graph conversion program automatically sets up your graph "Page Setup" for printing. The total number of runs is displayed in the top right hand corner of each graph when printed.*

## 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:

```
┌─────────────────────────────────────────────────────────────┐
│              ┌─────────────────────────────────┐             │
│              │   PCI.SP (HSPICE Circuit File)   │             │
│              └─────────────────────────────────┘             │
│                              │                               │
│                              ▼                               │
│              ┌─────────────────────────────────┐             │
│              │   Submit to HSPICE for simulation │           │
│              └─────────────────────────────────┘             │
│                              │                               │
│                              ▼                               │
│         ┌───────────────────────────────────────────┐       │
│         │ PCI.TR0 (plot file), PCI.MT0 (measure file) │     │
│         └───────────────────────────────────────────┘       │
│                              │                               │
│                              ▼                               │
│   ┌───────────────────────────────────────────────────────┐ │
│   │ Use MetaWaves to view PCI.TR0, or list PCI.MTO to see results │
│   └───────────────────────────────────────────────────────┘ │
│                              │                               │
│                              ▼                               │
│        ┌─────────────────────────────────────────────┐      │
│        │   Edit PCI.SP and re-simulate, if desired    │      │
│        └─────────────────────────────────────────────┘      │
└─────────────────────────────────────────────────────────────┘
```
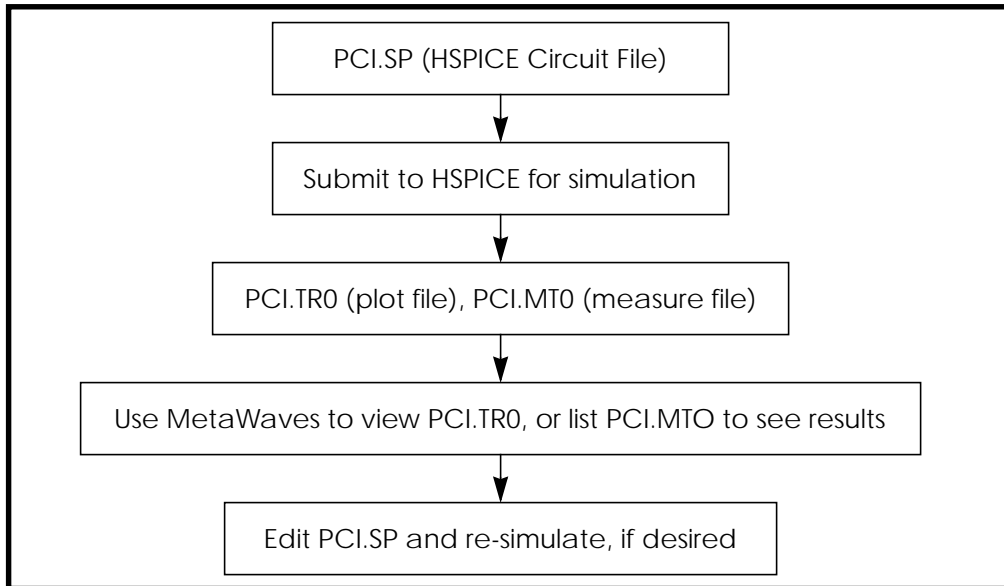
### Figure 20-20: PCI Simulation Process

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.

**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
```

**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
```

**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
```

**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

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
```
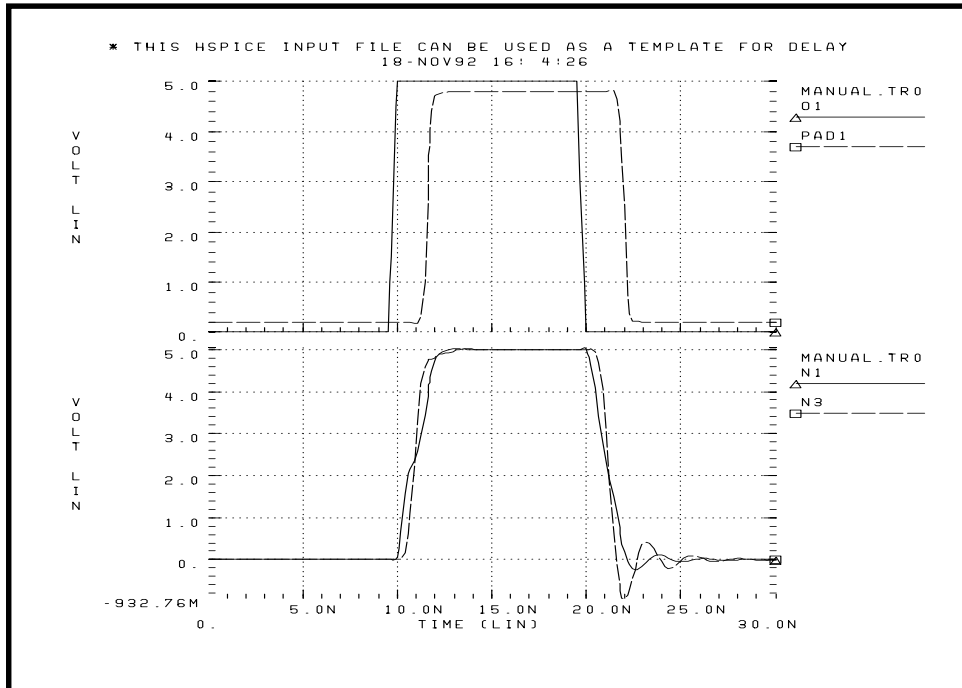


**Figure 20-21: Delay Template Output**

# 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
```
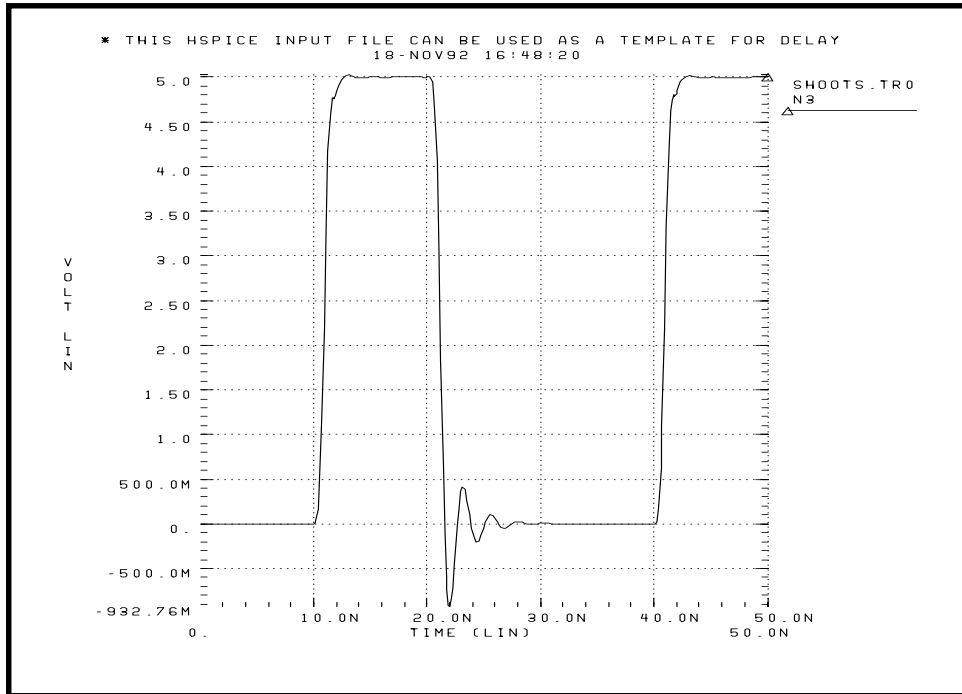
**Figure 20-22: Overshoot Template Example**

# Chapter 21

# 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 which 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. Other electrical elements which should be thought of as transmission lines include traces on printed circuit boards and multichip modules (MCMs) and within integrated circuits.

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 covers these topics:

- Selecting Wire Models
- Performing HSPICE Interconnect Simulation
- Understanding the Transmission Line Theory
- References

# Selecting Wire Models

Various terms are used for electrical interconnections between nodes in a circuit. Common terms are

- Wire
- Trace
- Conductor
- Line

The term "transmission line" or "interconnect" generally can be used to mean any of the above terms.

Many applications model electrical properties of interconnections between nodes by their equivalent circuits and integrate them into the system simulation to make accurate predictions of system performance. The choice of electrical model to simulate the behavior of interconnect must take into account 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

Choices for circuit models for interconnects are:

- No model at all. Use a common node to connect two elements.
- Lumped models with R, L, and C elements, as described in *Chapter 12, Using Passive Devices*. 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)

As a rule of thumb, follow Einstein's advice, "Everything should be made as simple as possible, but no simpler." 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. When compatibility with conventional versions of SPICE is required, use one of the discrete lumped models or the T element. The best choice of a transmission line model is determined by the following factors:

*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 21-1: is a guide to selecting a model based on the above factors.
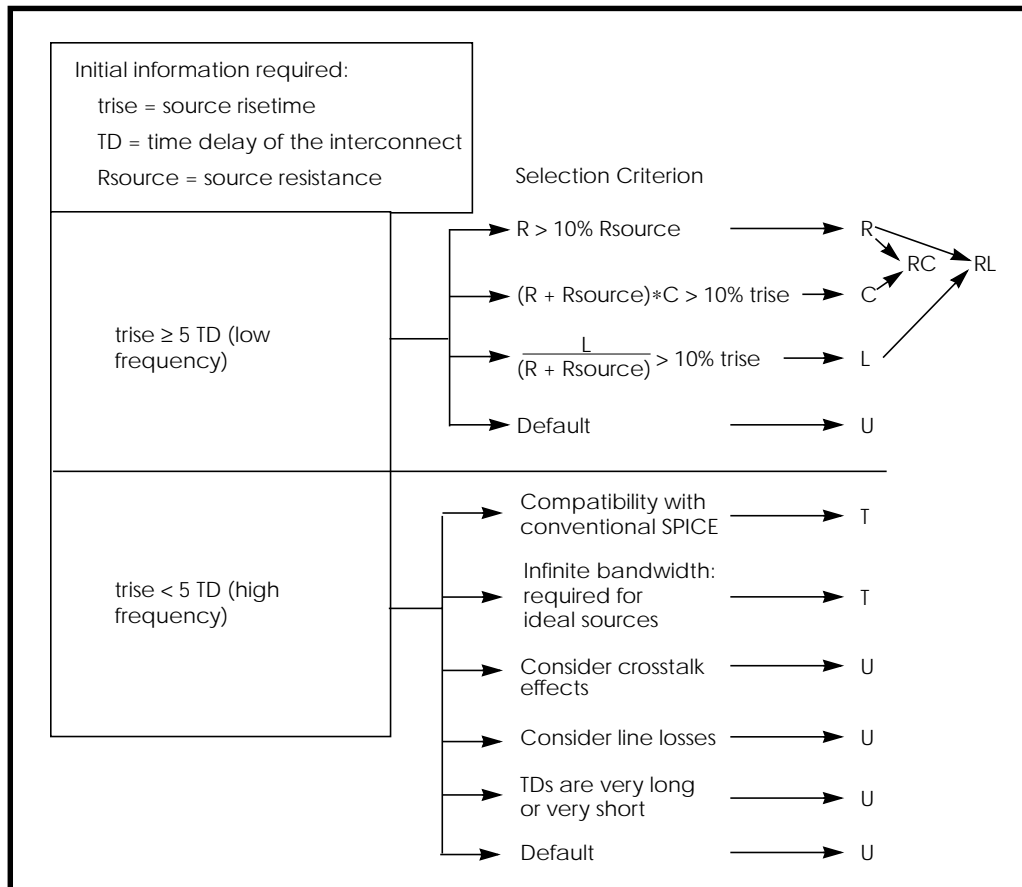
**Figure 21-1:  Wire Model Selection Chart**

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. Thus, Star-Hspice offers both a more flexible definition of the conventional SPICE T element and more accurate U element lossy simulations.
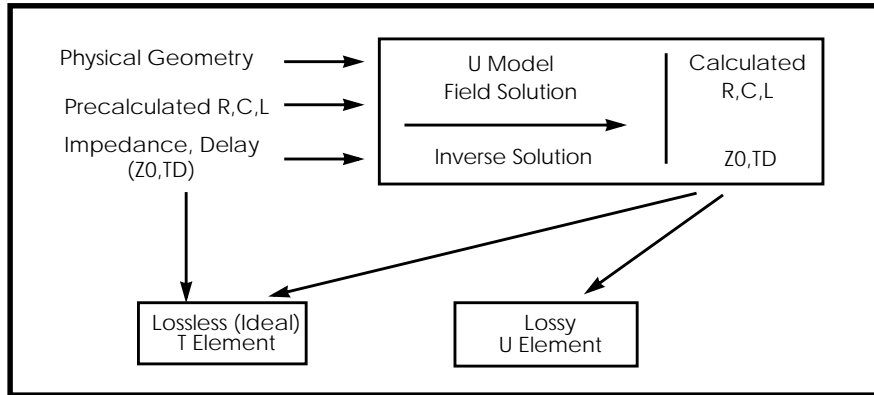
**Figure 21-2: U Model, T Element and U Element Relationship**

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.

## 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 as not to be 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.

## Selection of 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 fashion. 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 21-1:.

### Table 21-1: 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 illustrated in Figure 21-3:.



Ideal Element Circuit                          Lossy Element Circuit
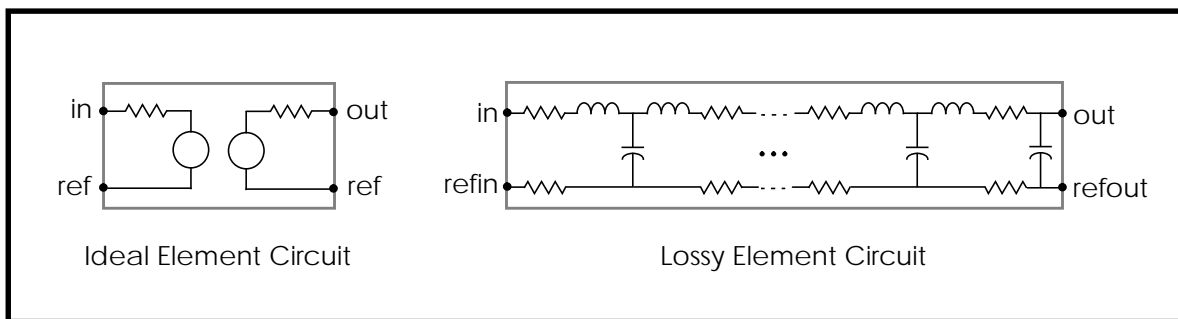
### Figure 21-3: Ideal versus Lossy Transmission Line Model

Because the ideal element represents the complex impedance as a resistor, the transmission line impedance is constant, even at DC values. On the other hand, you may need to prefilter the lossy element if ideal piecewise linear voltage sources are used to drive the line.

## U Model Selection

The U model allows three different description formats: geometric/physical, precomputed, and electrical. This model 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. This accommodates 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. This model 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.
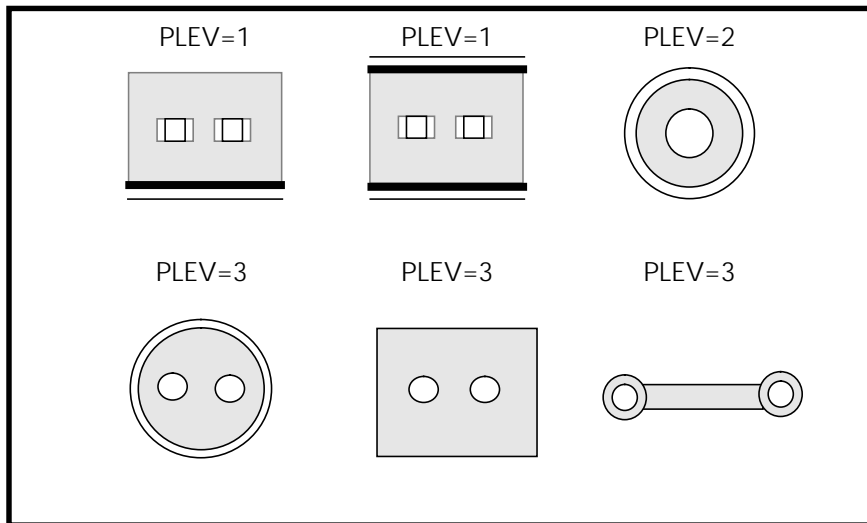
**Figure 21-4: U Model geometric Structures**

## Transmission Line Usage 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 21-2:. 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 HSPICE Interconnect Simulation

This section provides details of the requirements for T line or U line simulation.

## Ideal T Element Statement

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.
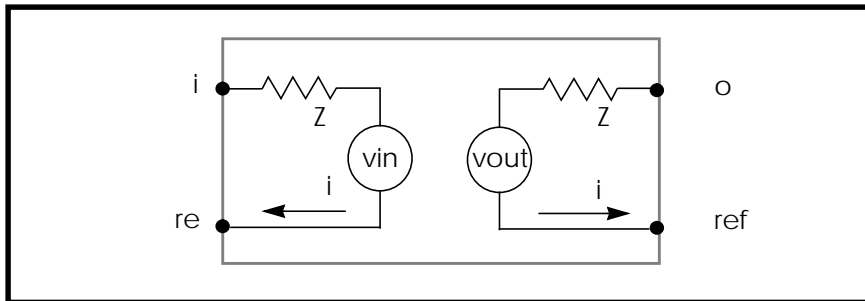


**Figure 21-5:  Ideal Element Circuit**

The input and output of the ideal transmission line have the following relationships:

$$Vin\big|_t = V(out-refout)\big|_{t-TD} + (iout \times Z_0)\big|_{t-TD}$$

$$Vout\big|_t = V(in-refin)\big|_{t-TD} + (iin \times Z_0)\big|_{t-TD}$$

### T Element Statement Syntax

The syntax is:
```
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>
```

or

```
Txxx in refin out refout mname L=val
```

| | |
|---|---|
| *F* | frequency at which the transmission line has electrical length NL |
| *IC* | initial conditions keyword |
| *i1* | initial branch current for input port |
| *i2* | initial branch current for output port |
| *in* | signal node ("in" side) |
| *L* | physical length of the transmission line (meter) default = 1 meter |
| *NL* | normalized electrical length of the transmission line with respect to the wavelength in the line at the frequency specified with the F parameter. Default=0.25, which corresponds to a quarter-wave frequency. |
| *mname* | U model reference name |
| *out* | signal node ("out" side) |
| *TD* | transmission delay (sec/meter) |

$$TDeff=TD{\cdot}L \text{ or}$$

$$TDeff=NL/F \text{ or}$$

$$TDeff=TD \text{ (computed from U model)}{\cdot}L$$

| | |
|---|---|
| *refin, refout* | ground references for input and output |
| *Txxx* | transmission line (lossless) element. Must begin with a "T", which may be followed by up to 15 alphanumeric characters. |
| *v1* | initial voltage across input port |
| *v2* | initial voltage across output port |
| *Z0* | characteristic impedance |

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. If the full signal and reference are required, a U element should be used. However, as a crude approximation, two T elements may be used as shown in Figure 21-6. Note that in this figure, the two lines are completely uncoupled, so that only the delay and impedance values are correctly modeled.
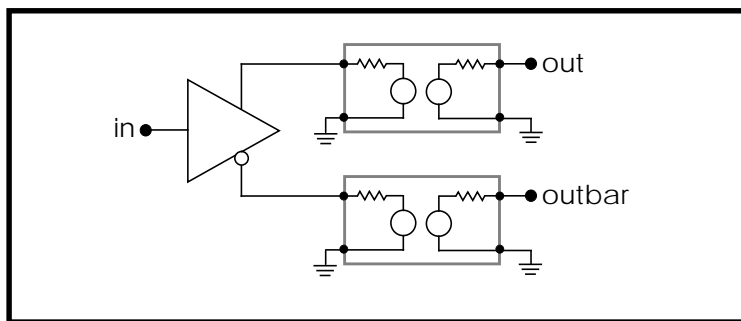


**Figure 21-6: Use of Two T Elements for Full Signal and Reference**

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. Very short lines can usually be replaced by a single R, L, or C element (see Figure 21-1).

## 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 *Chapter 12, Using Passive Devices*. The interconnect properties may be specified in three ways:

■ The R, L, C, and G (conductance) parameters may be directly specified in matrix form (ELEV = 2).

■ Common electrical parameters, such as characteristic impedance and attenuation factors (ELEV = 3) may be provided.

■ The geometry and the material properties of the interconnect may be specified (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 have been optimized via measurements and comparisons with several different electromagnetic field solvers.

The Star-Hspice U element geometric model can handle from one to five uniformly spaced transmission lines, all at the same height. Also, the transmission lines may 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. Two kinds of lossy lines can be made, 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 typically 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.

## U Element Statement Syntax

The syntax is:

One wire with ground reference:
```
Uxxx in refin out refout mname L=val <LUMPS=val>
```

Two wires with ground reference:
```
Uxxx in1 in2 refin out1 out2 refout mname L=val <LUMPS=val>
```

Two or more wires with ground reference:
```
Uxxx in1 ... inn refin out1 ... outn refout mname L=val
<LUMPS=val>
```

| | |
|---|---|
| *Uxxx* | lossy transmission line element name |
| *in1, inn* | input signal nodes 1 through *n* |
| *refin, refout* | input or output reference name |
| *out1, outn* | output signal node 1 through *n* |
| *mname* | lossy transmission line model name |
| *L=val* | element length in meters |
| *LUMPS=val* | number of lumps (lumped-parameter sections) in the element |

## Lossy U Model Statement

The schematic for a single lump of the U model, with LLEV=0, is shown in Figure 21-7:. If LLEV is 1, the schematic includes inductance in the reference path as well as capacitance to HSPICE ground. See "Reference Planes and HSPICE Ground" for more information about LLEV=1 and reference planes.
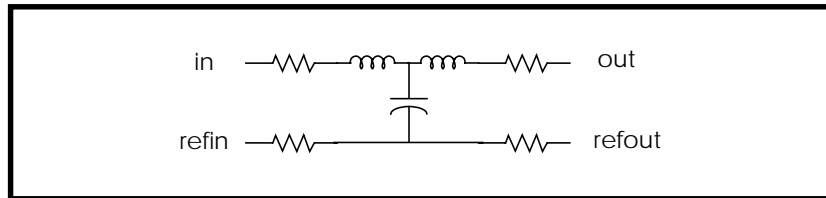


**Figure 21-7:  Lossy Line with Reference Plane**

HSPICE netlist syntax for the U model is shown below. Model parameters are listed in Tables 21-2 and 21-3.

### 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 21-2:) or a loss parameter, such as RHO or NLAY (see Table 21-3:). |

Figure 21-8: 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.
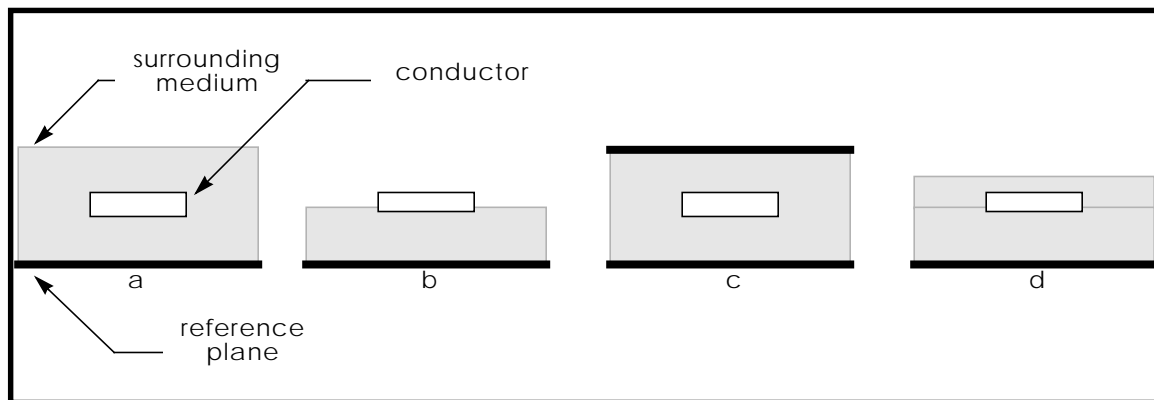


**Figure 21-8: Dielectric and Reference Plane Configurations:
a) sea, DLEV=0, b) microstrip, DLEV=1, c) stripline, DLEV=2d) overlay,
DLEV=3**

## Lossy U Model Parameters for Planar Geometric Models (PLEV=1, ELEV=1)

## Common Planar Model Parameters

The parameters for U models are shown in Table 21-2:.

### Table 21-2: U Element Physical Parameters

| Parameter | Units | Default | Description |
|---|---|---|---|
| LEVEL | | req* | (=3) required for lossy transmission lines model |
| ELEV | | req | electrical model (=1 for geometry) |
| DLEV | | | dielectric model<br>(=0 for sea, =1 for microstrip, =2 stripline, =3 overlay; default is 1) |
| PLEV | | req | transmission line physical model (=1 for planar) |
| LLEV | | | omit or include the reference plane inductance<br>(=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 |
| TH | m | | thickness of all conductors |
| THB | m | | reference plane thickness |
| TS | m | | distance between reference planes for stripline<br>(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 |

*\* Required – must be specified in the input*

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 21-9:.
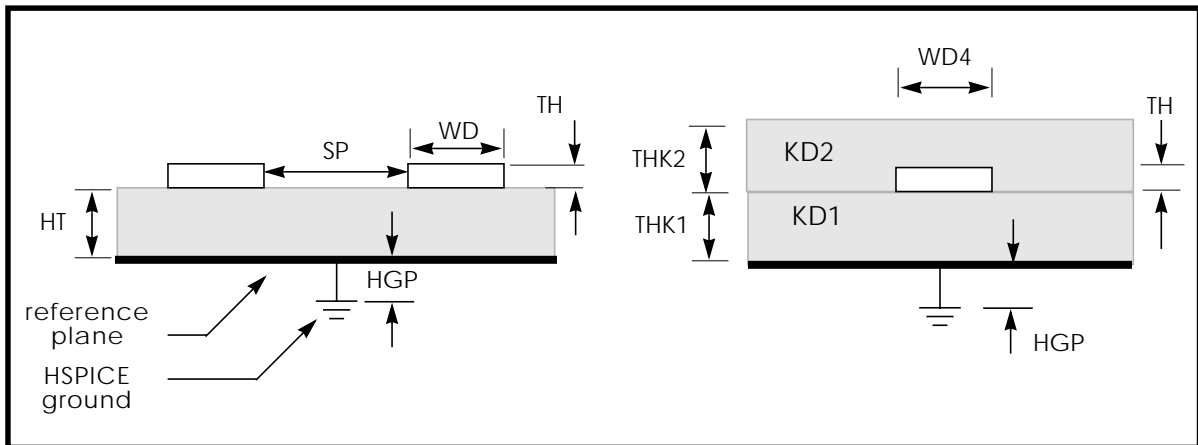


**Figure 21-9: U Element Conductor Dimensions**

### Loss Parameters

Loss parameters for the U model are shown in Table 21-3:.

### Table 21-3: 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 21-4: lists the recommended minimum and maximum values for U element parameter variables.

### Table 21-4: 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 values outside the recommended ranges are used. 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 21-10: 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.
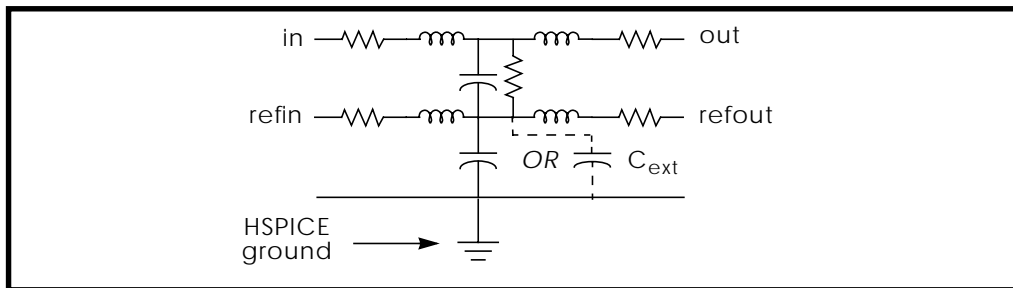


**Figure 21-10: Schematic of a U Element Lump when LLEV=1**

The model reference plane is not necessarily the same as HSPICE ground. For example, a printed circuit board with transmission lines might have a separate reference plane above a chassis. 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 height HGP above 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 any one of the parameters *tstep*, *delmax*, and *trise* is set, 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 21-27: on page -45. 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 should reduce 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 21-64 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 21-11:).
- DLEV=1 – microstrip dual dielectric. This planar conductor has a single reference plane and two dielectric layers (Figure 21-12:).
- DLEV=2 – stripline. This planar conductor has an upper and lower reference plane (Figure 21-13:). 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 21-14).
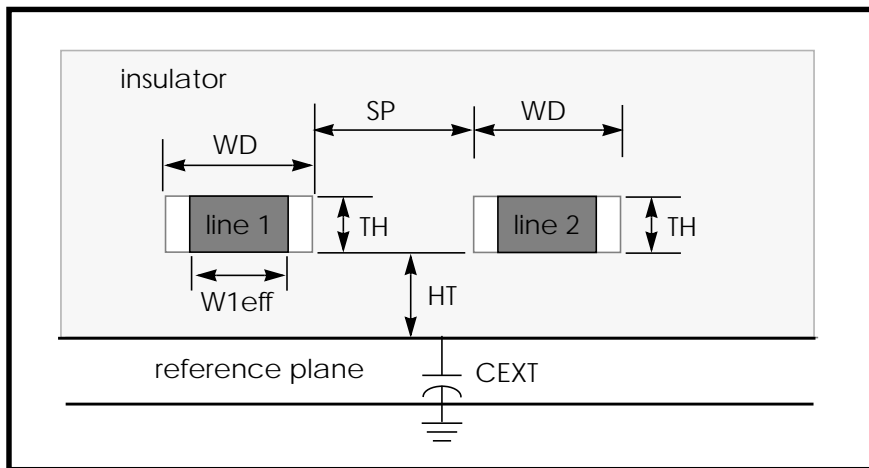
**Figure 21-11: Planar Transmission Line, DLEV=0, Sea of Dielectric**
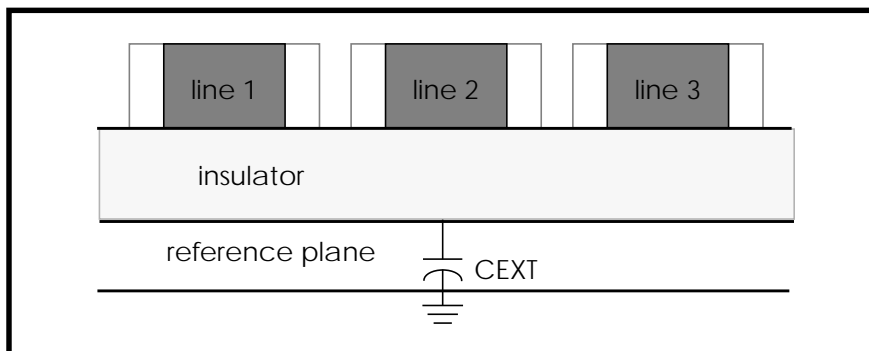


**Figure 21-12: Planar Transmission Line, DLEV=1, Microstrip**
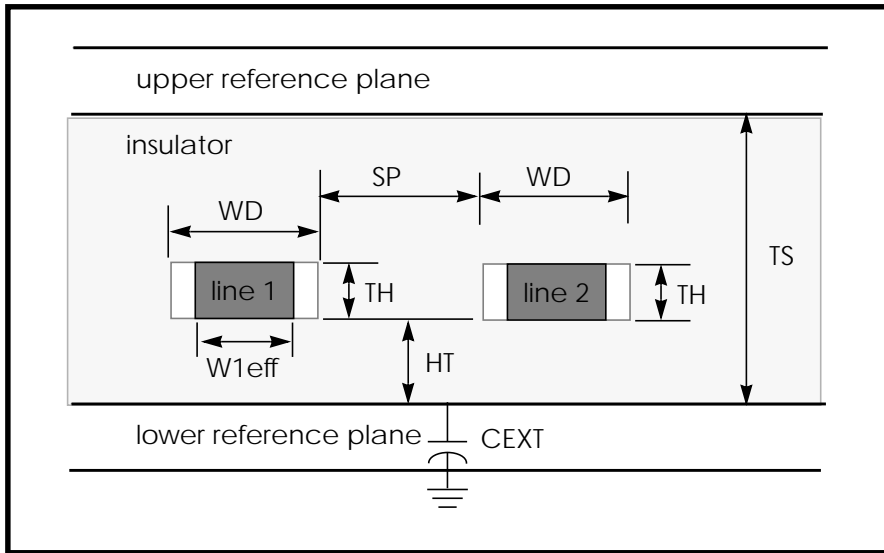
**Figure 21-13:  Planar Transmission Line, DLEV=2, Stripline**
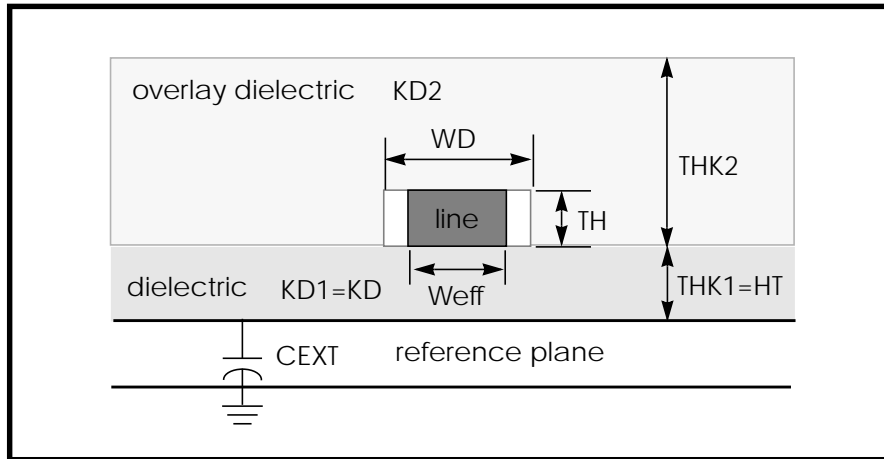


**Figure 21-14: Planar Transmission Line, DLEV=3, Overlay Dielectric**

### ELEV=1 Parameters

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| DLEV | — | 1.0 | 0: microstrip sea of dielectric<br>1: microstrip layered dielectric<br>2: stripline |
| NL | — | 1 | number of conductors |
| NLAY | — | 1.0 | layer algorithm:<br>1: DC cross section only<br>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<br>$0 \leq THK2 < 3 \cdot HT$ (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<br>$TS=TH+2 \cdot HT$ (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 |
| 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 |

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| KD2 | | KD | relative dielectric constant of overlay dielectric for DLEV=3<br><br>$1 < KD1 < 4 \cdot KD$ |
| 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)
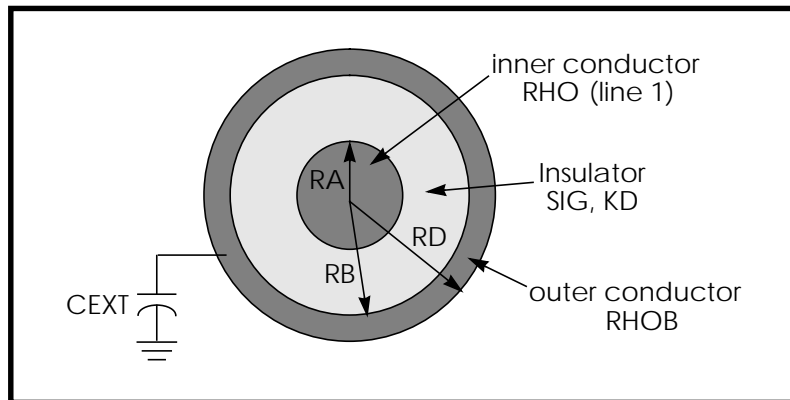


**Figure 21-15: Geometric Coaxial Cable**

## 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)
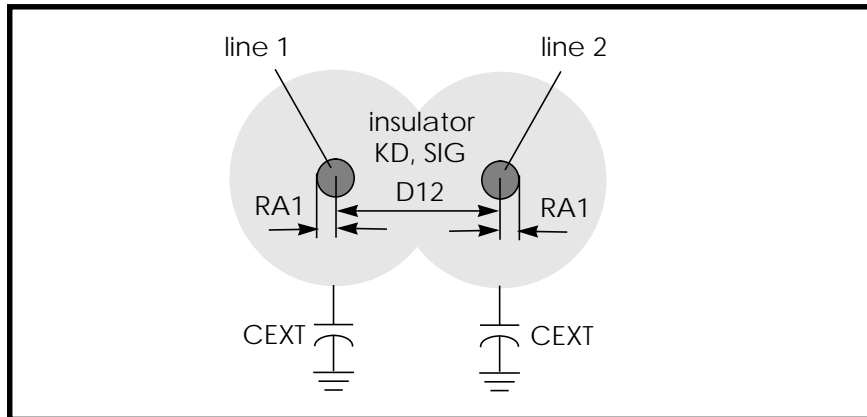


**Figure 21-16: Geometric Embedded Twinlead, DLEV=0, Sea of Dielectric**
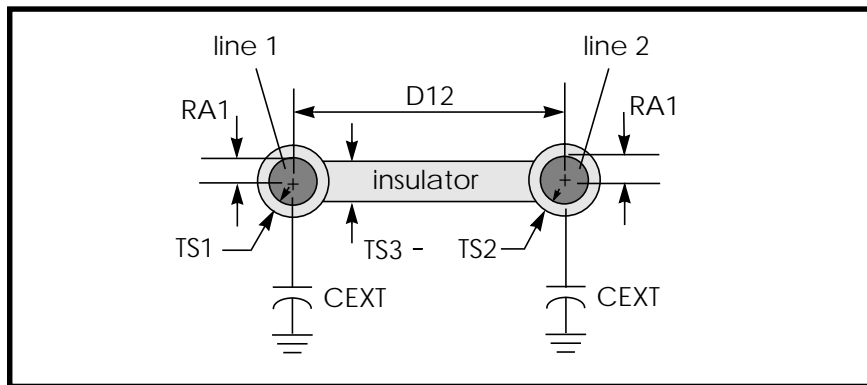


**Figure 21-17: Geometric Twinlead, DLEV=1, with Insulating Spacer**
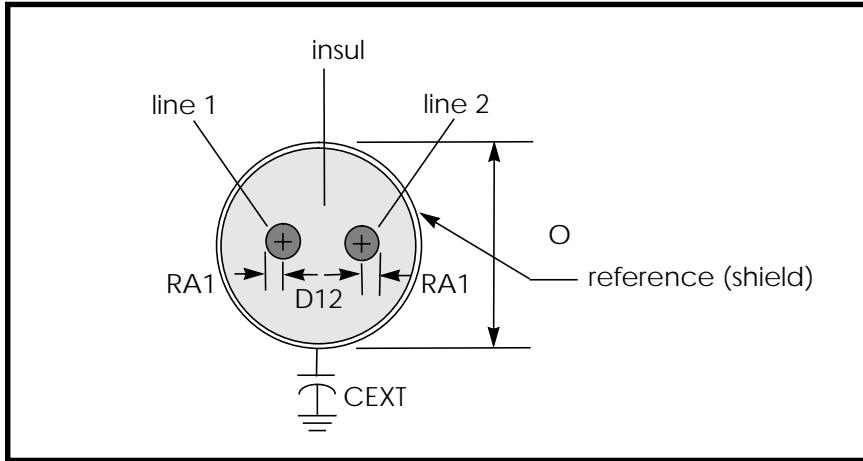
**Figure 21-18: Geometric Twinlead, DLEV=2, Shielded**

## Geometric Twinlead Parameters (ELEV=1)

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| DLEV | — | 0.0 | 0: embedded twinlead<br>1: spacer twinlead<br>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 21-19: 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 21-20:.
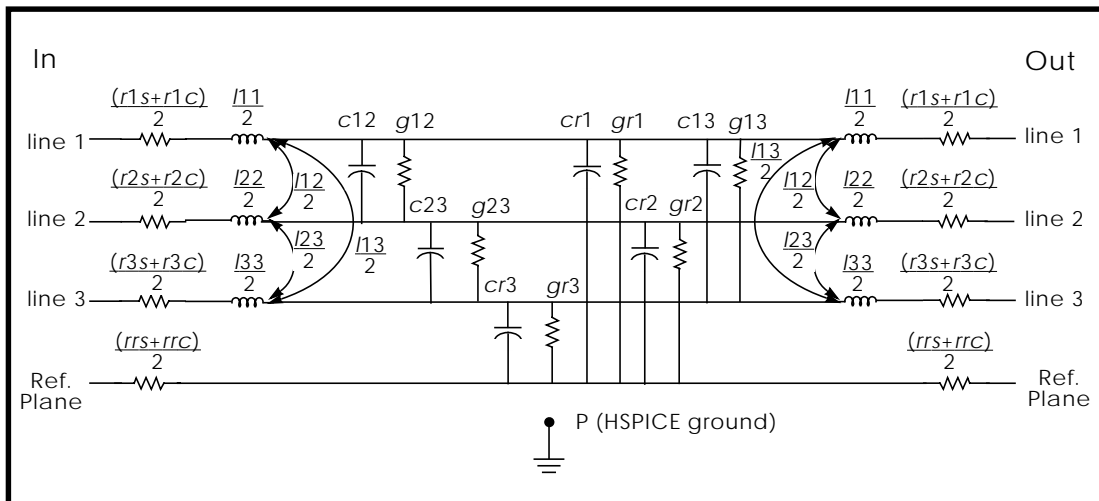


**Figure 21-19: Precomputed Components for Three Conductors and a Reference Plane**

| | 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 |
| 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 |

LLEV=1 parameter only

### Figure 21-20: ELEV=2 Model Keywords for Conductor PLEV=1

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 21-21:, demonstrates the three definitions of capacitance. The branch capacitances are given in Star-Hspice notation.
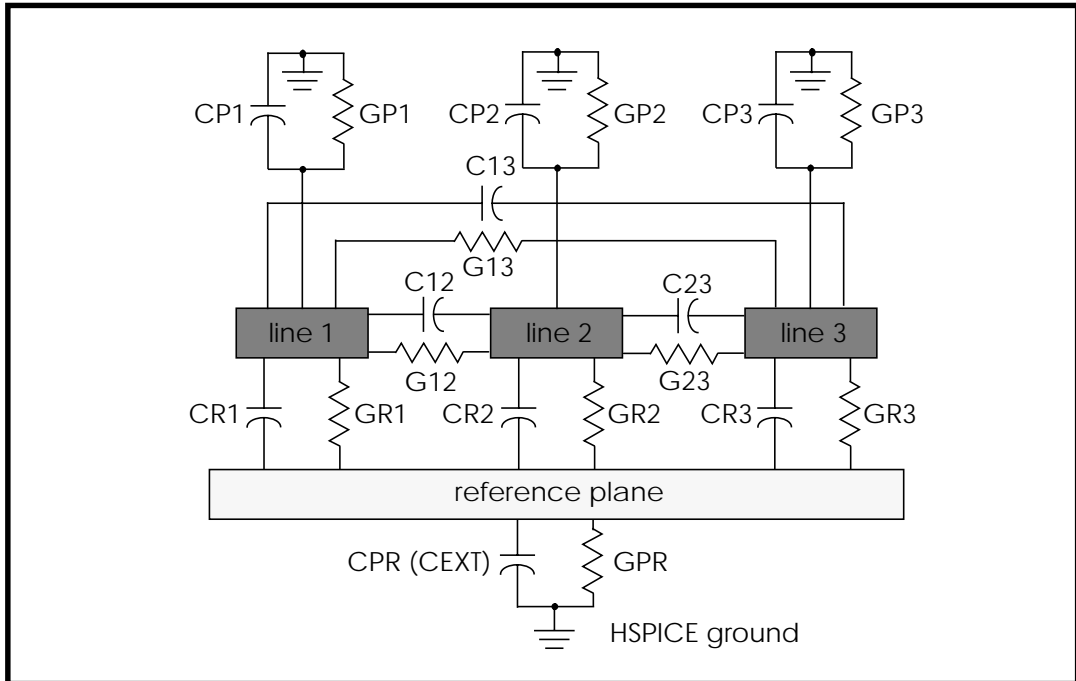


**Figure 21-21: Single-Lump Circuit Capacitance**

The branch and Maxwell matrixes are completely derivable from each other. The "O.C.G." ("other conductors grounded") matrix is derivable from either the Maxwell matrix or the branch matrix. Thus:

$$CX = \sum_{X \neq Y} CXY$$

Cjk    = CX on diagonal
         = –CXY off diagonal

The matrixes 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:

$$Cjk = \begin{bmatrix} CR & -CR1 & -CR2 & -CR3 \\ -CR1 & C1 & -C12 & -C13 \\ -CR2 & -C12 & C2 & -C23 \\ -CR3 & -C13 & -C23 & C3 \end{bmatrix}$$

The branch capacitances also may be obtained from the Maxwell matrixes. 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.

$$
= \begin{vmatrix} CR & -CR1 & -CR2 & -CR3 \\ -CR1 & C1 & -C12 & -C13 \\ -CR2 & -C12 & C2 & -C23 \\ -CR3 & -C13 & -C23 & C3 \end{vmatrix} \cdot jw \cdot \begin{vmatrix} 0.0 \\ 1.0 \\ 0.0 \\ 0.0 \end{vmatrix} = jw \cdot \begin{vmatrix} -CR \\ C1 \\ -C1 \\ -C1 \end{vmatrix}
$$

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<br>2: coax.<br>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 $\sqrt{}$(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.  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 21-5: 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)<br><br>L=ZK/(VREL · CLIGHT) |
| *ZK, CAPL, CLEN* | C=CAPL/CLEN<br><br>L=C · ZK$^2$ |
| *CAPL, CLEN, DELAY, DELEN* | VREL=DELEN/(DELAY · CLIGHT) |
| CAPL, CLEN, VREL | LC=CAPL/CLEN<br><br>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 21-6: 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 ATI < 1) |
| AT1 = 0.758 | v(out)/v(in) = 0.758 = AT1 (for ATI > 1) |

The attenuation factor is used to compute the exponential loss parameter and linear resistance.

$$\alpha \;=\; \frac{ln((v(in))/(v(out)))}{ATlin}$$

$$LR \;=\; 2 \cdot \alpha \cdot \sqrt{(LL)/(LC)}$$

## 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.

### Example 1 – Three Coupled Lines, Stripline Configuration

Figure 21-22: 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 21-23:.
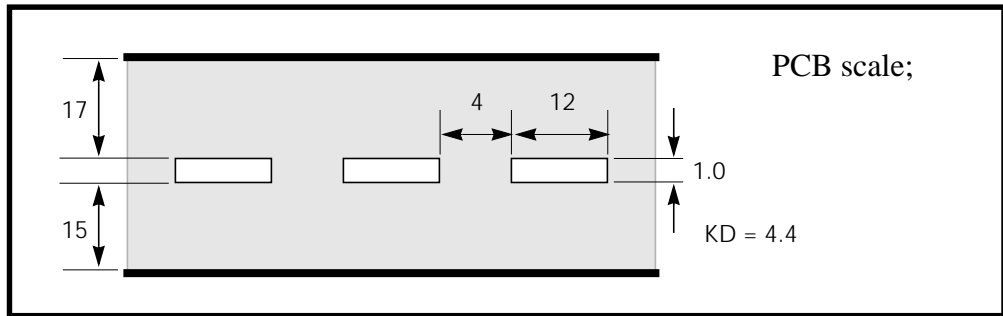
**Figure 21-22: Three Coupled Striplines (PCB Scale)**
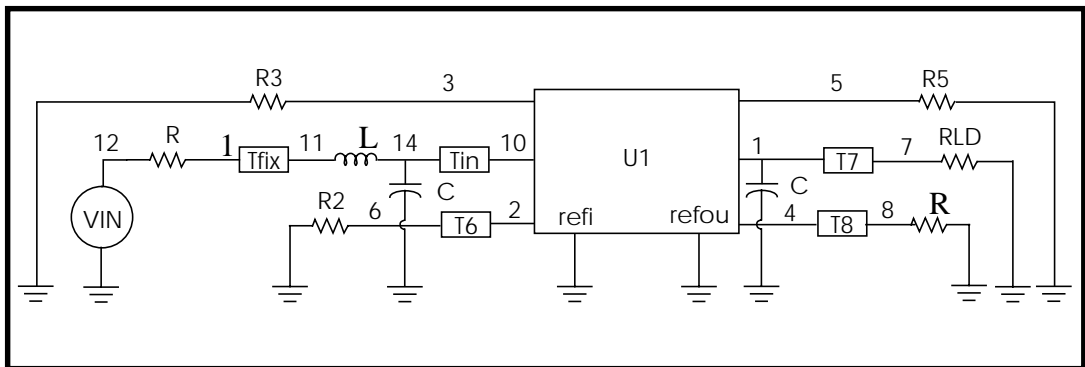


**Figure 21-23: Schematic Using the Three Coupled Striplines U Model**

The 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 21-24, 21-25, and 21-26 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.
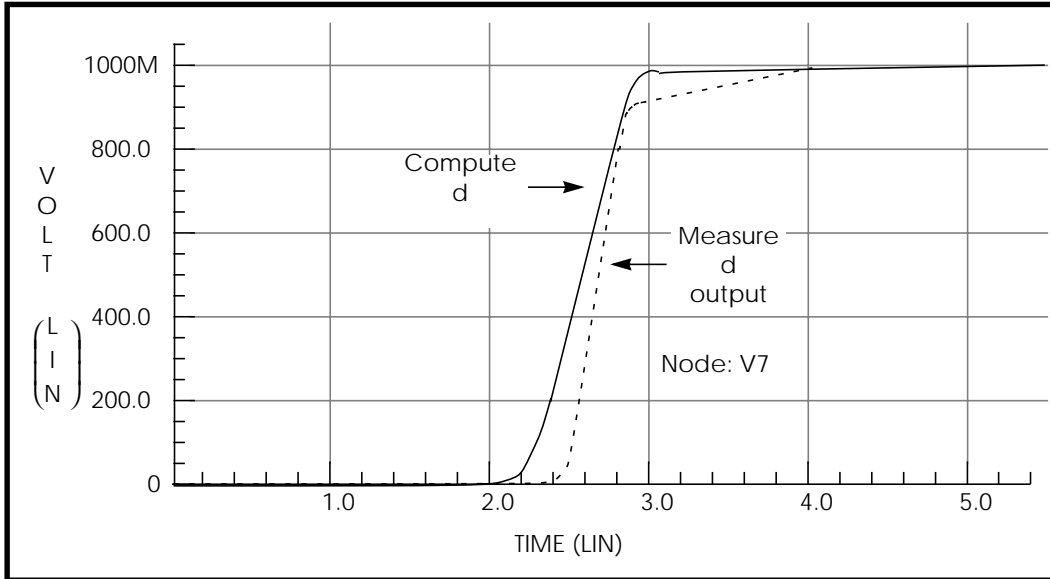


**Figure 21-24: Measured versus Computed Through-Line Response**

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 21-25: 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.
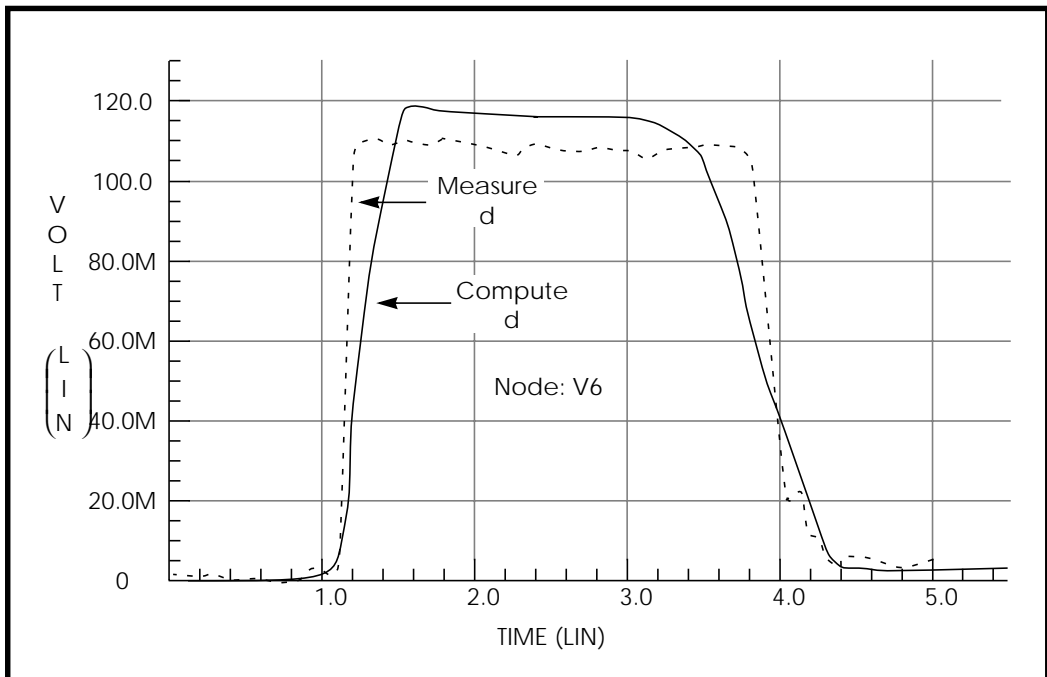


**Figure 21-25: Measured Versus Computed Backward Crosstalk Response**
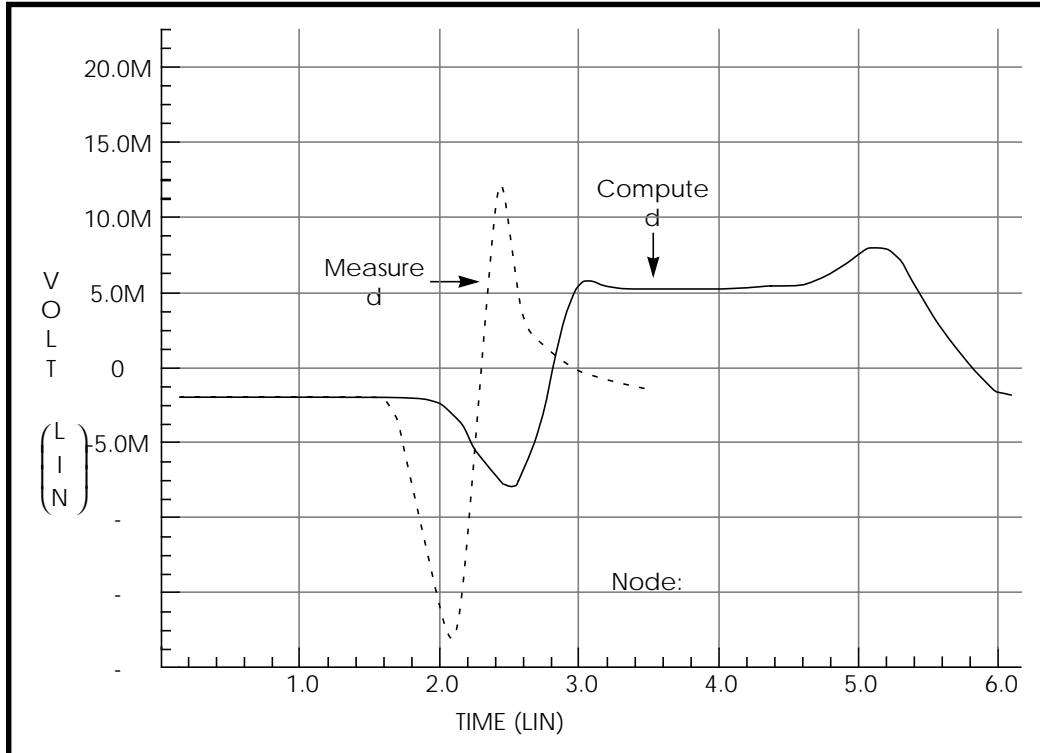
**Figure 21-26:  Measured Versus Computed Forward Crosstalk Response**

Figure 21-26: 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 21-27: shows the main line response at Node 7 of Figure 21-23: 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.
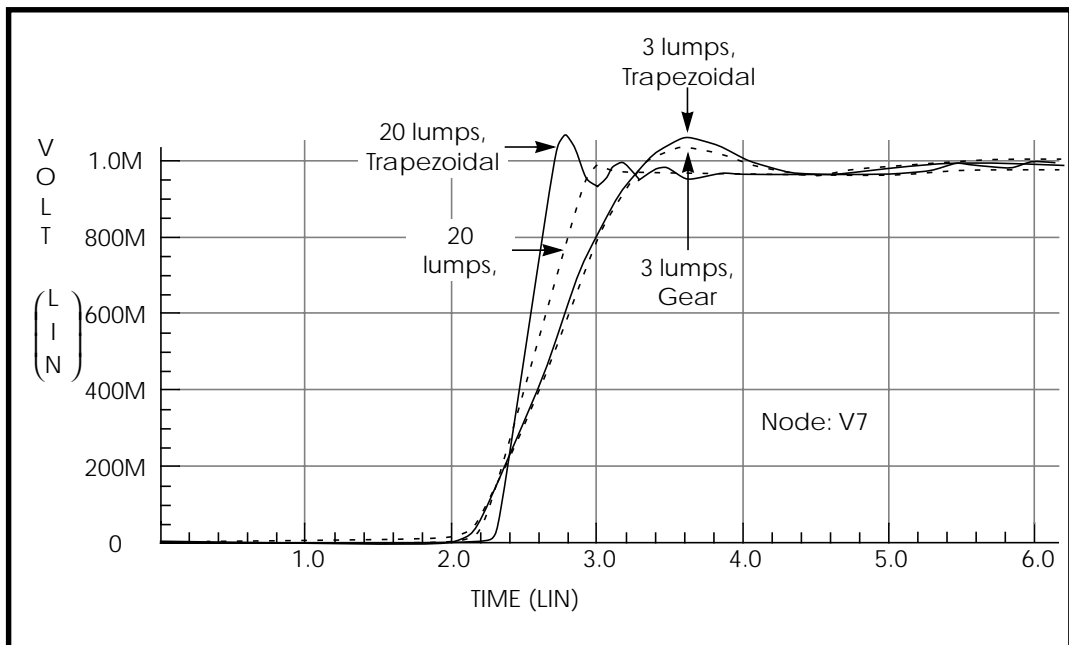


**Figure 21-27:  Computed Responses for 20 Lumps and 3 Lumps, Gear and Trapezoidal Integration Methods**

## Example 2 – 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 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 21-28: shows the IC-scale coupled line geometry.
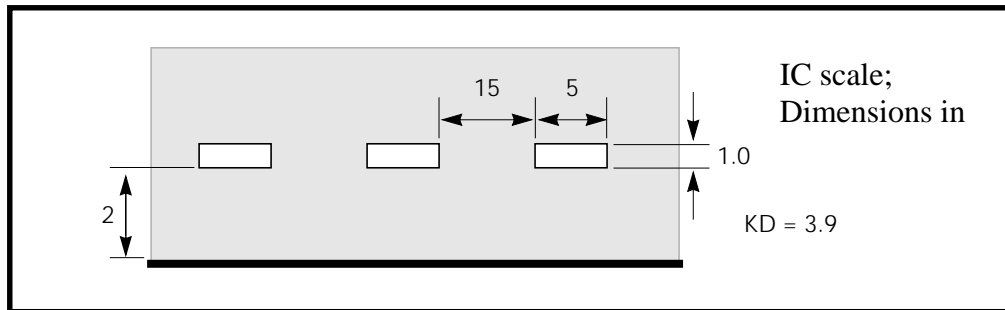


**Figure 21-28:  Three Coupled Lines with One Reference Plane in a Sea of Dielectric (IC Scale)**

Figure 21-29: shows one lump of the lumped-parameter schematic for the three-conductor stripline configuration of Figure 21-28:. This is the internal circuitry Star-Hspice creates to represent one U element instantiation. The internal elements are described in "HSPICE Output for Example 2" on page 21-51.
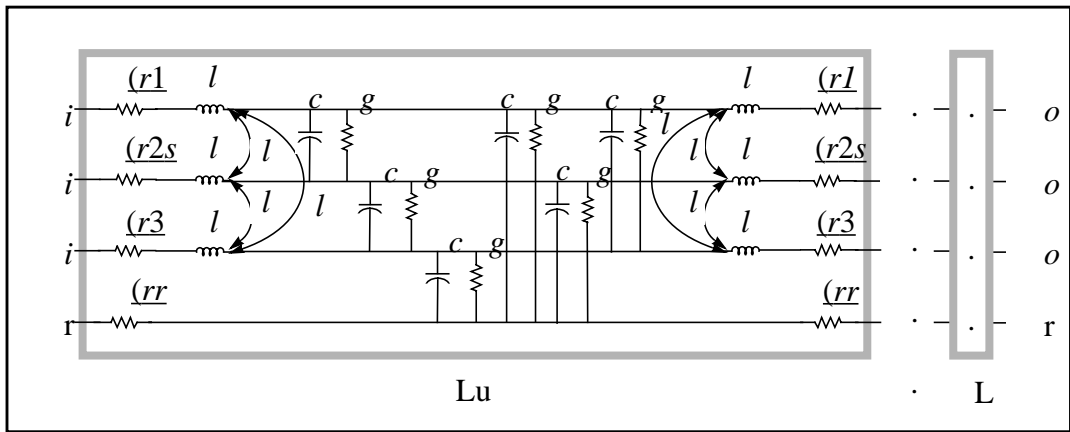
**Figure 21-29: – Schematic for Three Coupled Lines with One Reference Plane**

Figure 21-30: shows a schematic using the U element of Figure 21-29:. In this simple circuit, a pulse drives a three-conductor transmission line source terminated by 50Ω resistors and loaded by 1pF capacitors.
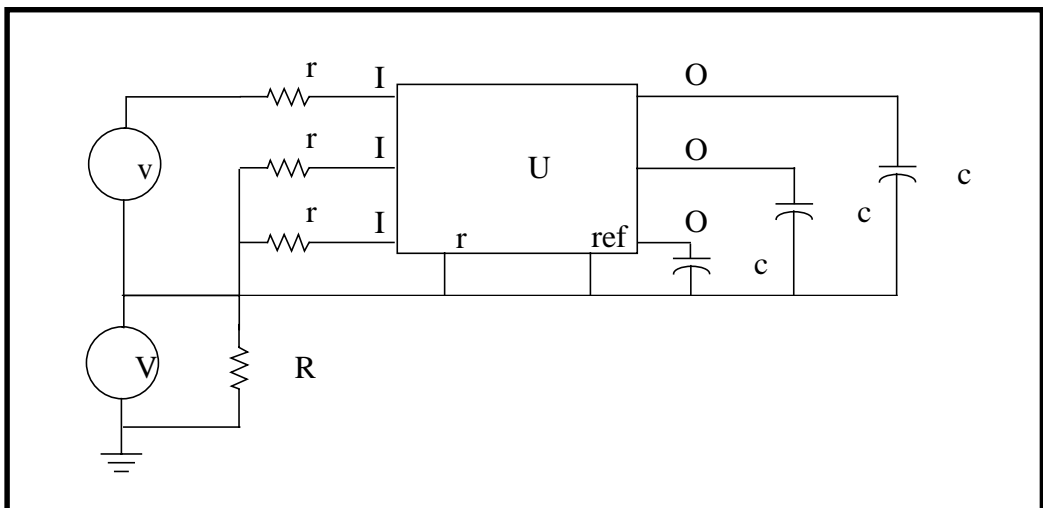


**Figure 21-30:  Schematic Using the Three Coupled Lines U Model**

The Star-Hspice input file for the U element solution is shown below.

```
.Tran 0.1ns 20ns
.Options Post Accurate NoMod Brief Probe
Vss Vss 0 0v
Rss Vss 0 1x
vIn1 In1 Vss Pwl 0ns 0v 11ns 0v 12ns 5v 15ns 5v 16ns 0v
rIn1 In1 In10 50
rIn2 Vss In20 50
rIN3 Vss In30 50
u1 In10 In20 In30 Vss Out1 Out2 Out3 Vss IcWire L=200um
cIn1 Out1 Vss 1pF
cIn2 Out2 Vss 1pF
cIn3 Out3 Vss 1pF
.Probe v(Out1) v(Out2) v(Out3)
.Model IcWire U Level=3 Dlev=0 Nl=3 Nlay=2 Plev=1 Elev=1
Llev=0 Ht=2u
+ Wd=5u Sp=15u Th=1u Rho=2.8e-8 Kd=3.9
.End
```

The HSPICE U element uses the conductor geometry to create length-independent RLC matrixes 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 21-31, Figure 21-32, and Figure 21-33 show the through and coupled responses computed by Star-Hspice using the U element equations.
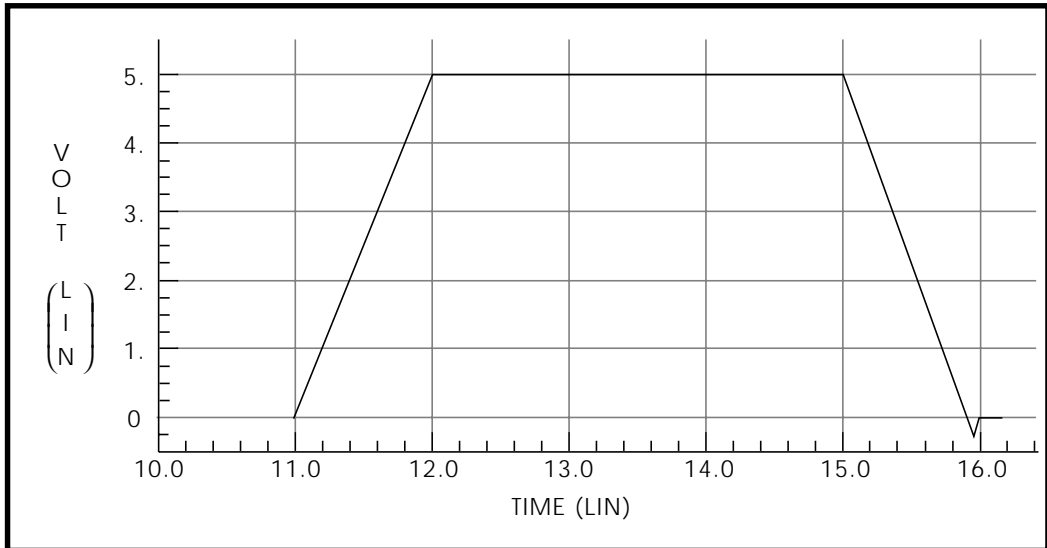
**Figure 21-31:  Computed Through-Line Response**

Figure 21-32 shows the nearest coupled line response. This response only occurs during signal transitions.
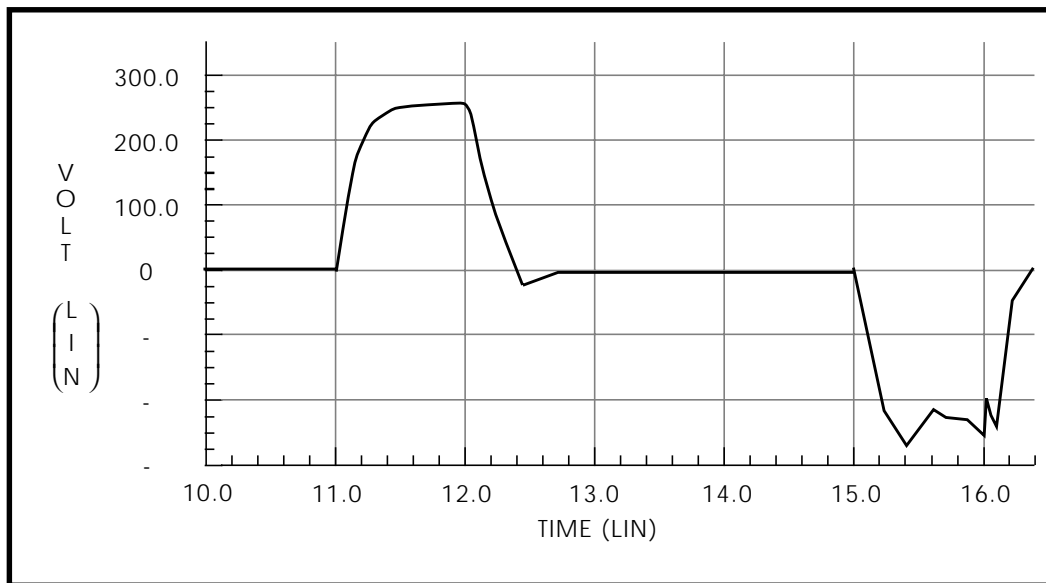
**Figure 21-32: Computed Nearest Coupled-Line Response**

Figure 21-33: shows the third coupled line response. The predicted response is about 1/100000 of the main line response.
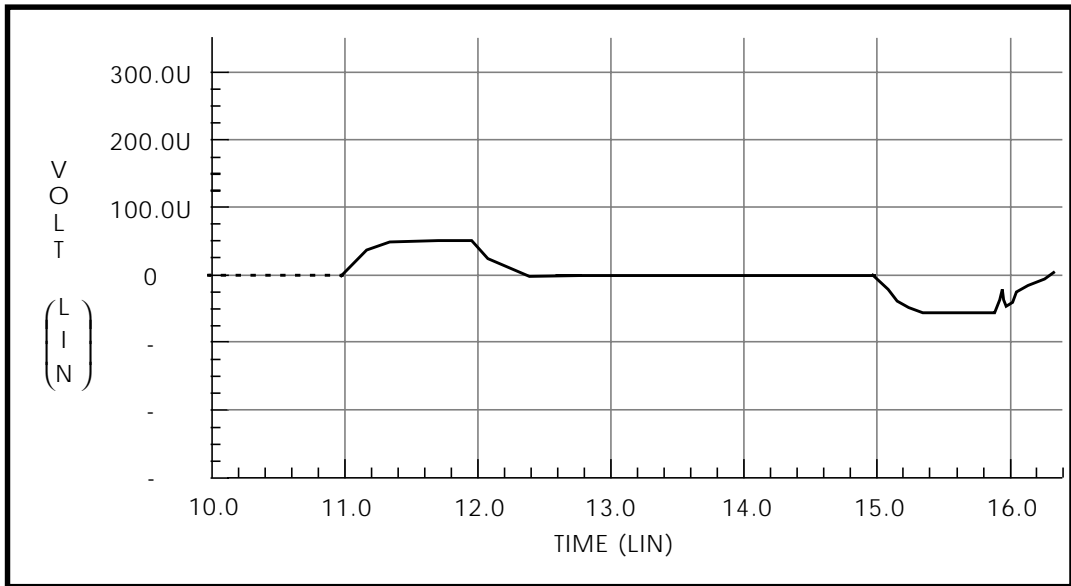
**Figure 21-33:  Computed Furthest Coupled-Line Response**

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.

### HSPICE Output for Example 2

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 21-2 and 21-3.)

### 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

| | |
|---|---|
| c*ij* | 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 |
| g*ij* | 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 |
| l$ij$ | inductance/m from conductor $i$ to conductor $j$ |
| l$jj$ | 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) |
| r$i$s | skin resistance/m of conductor $i$ (if NLAY = 2) |
| r$i$c | core resistance/m of conductor $i$ (if NLAY = 2, zero if skin depth > 50% of th) |
| r$jj$ | resistance/m of conductor $j$ (if NLAY = 1) |
| skin | skin depth |
| skinb | skin depth of the reference plane |
| 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 r$jj$ when NLAY = 1, or by r$i$s + r$i$c 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 21-4: 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 Matrixes

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 21-34: shows the capacitance and inductance matrixes for the three-line, buried microstrip IC-scale example shown in Figure 21-28:.

| | |
|---|---|
| **Capacitance (pF/m)** | 176  -5.02  -0.65<br>-5.02  178  -5.02<br>-0.65  -5.02  176 |
| **Inductance (nH/m)** | 246　　6.97　　1.11<br>6.97　　243　　6.97<br>1.11　　6.97　　246 |

**Figure 21-34:  Capacitance and Inductance Matrixes for the Three-Line, IC-Scale Interconnect System**

The capacitance matrices in Figure 21-34: 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 21-35: shows the equivalent capacitances, in terms of Star-Hspice parameters.
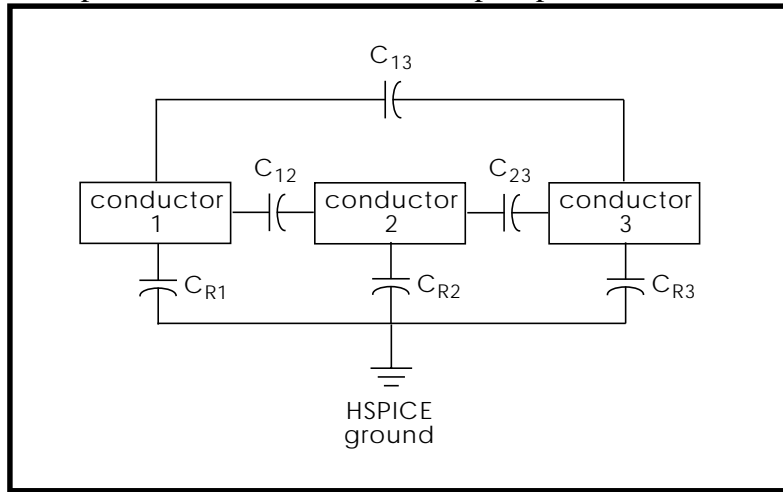


**Figure 21-35:  Conductor Capacitances for Example 2**

The capacitances of Figure 21-35: are those shown in Figure 21-29:. The HSPICE nodal capacitance matrix of Figure 21-34: is shown below, using the capacitance terms that are listed in the HSPICE output.

$$
\begin{bmatrix}
C_{R1} + C_{12} & -C_{12} & -C_{13} \\
-C_{12} & C_{R2} + C_{12} & -C_{23} \\
-C_{13} & -C_{23} & C_{R3} + C_{13}
\end{bmatrix}
$$

The off-diagonal terms are the negative of the coupling capacitances (to conform to the sign convention). The diagonal terms require some computation, for example,

$$C_{11} \quad = C_{R1} + C_{12} + C_{13}$$

$$= 170.28 + 5.02 + 0.65$$

$$= 175.95, \text{ or } 176 \text{ pF/m}$$

Note that the matrix values on the diagonal in Figure 21-34: 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 Figure 21-34:. 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. Figure 21-34: represents very small coupling because the line spacing is large (about seven substrate heights).

Table 21-7: shows the parameters for the stripline shown in Figure 21-36:.
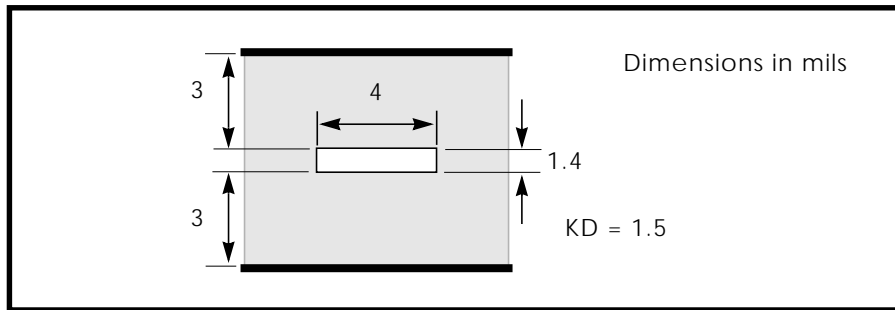


**Figure 21-36: Stripline Geometry Used in MCM Technology**

**Table 21-7: Capacitance and Inductance for the Single Line MCM-Scale Stripline**

| Capacitance (pF/m) | 164.4 |
|---|---|
| Inductance (nH/m) | 236.5 |

### Example 3 – Five Coupled LInes, Stripline Configuration

This example shows a five-line interconnect system in a PCB technology. Table 21-8: shows the matrix parameters for the line configuration of Figure 21-37:.

**Figure 21-37:  Five Coupled Lines on a PCB**
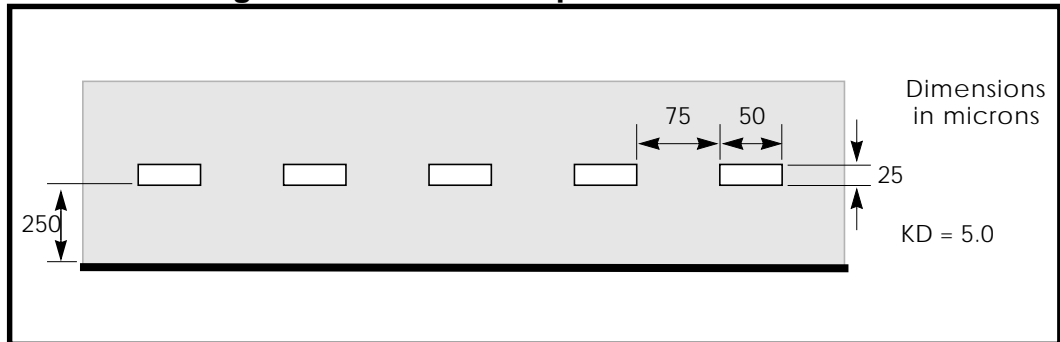


**Table 21-8:  Capacitance and Inductance for the Five-Line PCB-Scale Interconnect System**

| 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 21-38: 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
```
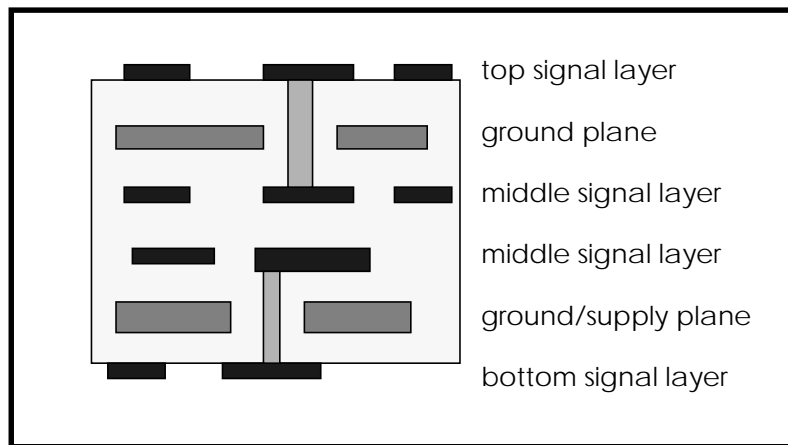


**Figure 21-38: Six-Layer Printed Circuit Board**

**Coax Models**

The following examples are for standard coax. These are obtained from commonly available tables[2]. (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 21-39: 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.
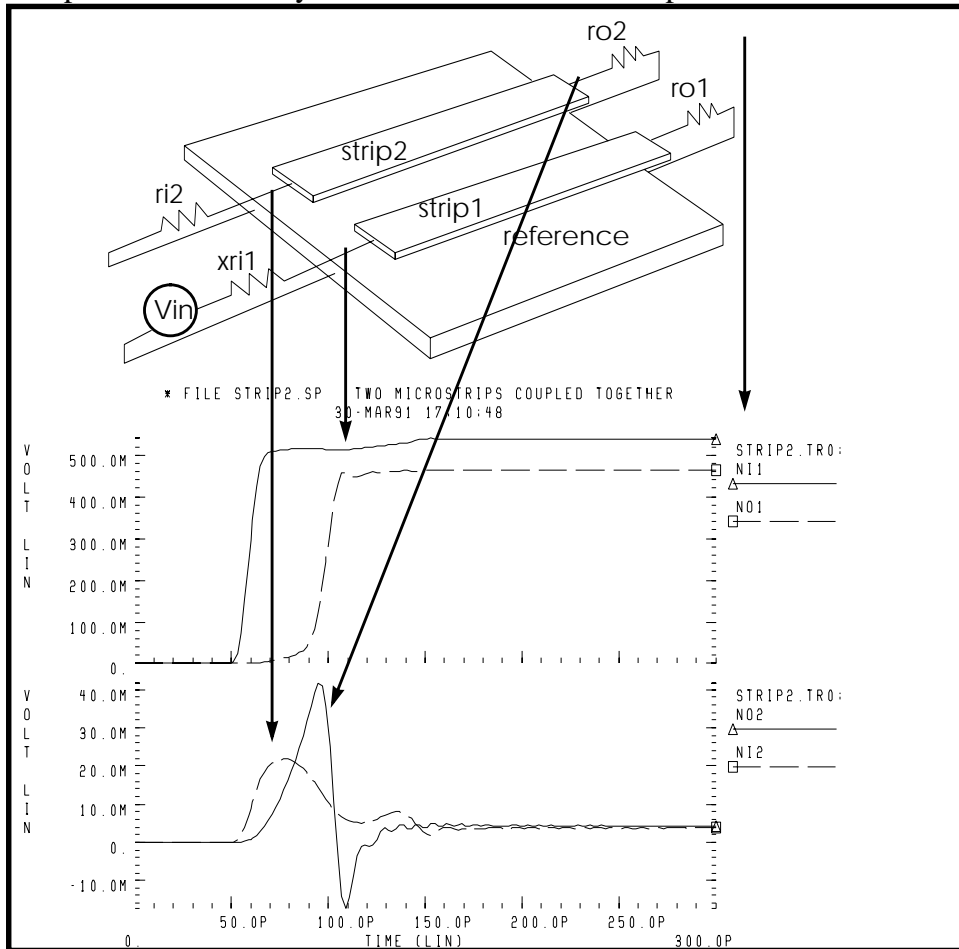


**Figure 21-39: Two Coupled Microstrips Geometrically Defined as LSI Metallization**

### 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
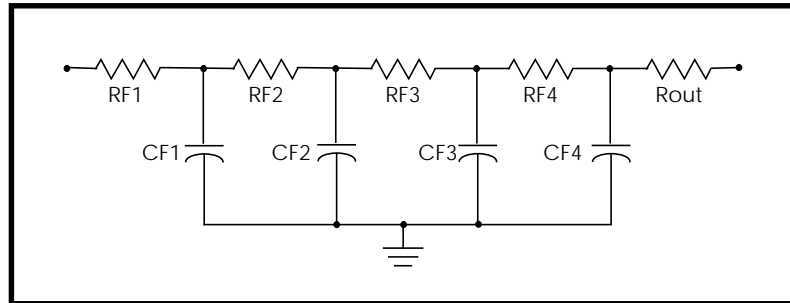
**Figure 21-40: Circuit Diagram of an RC Filter**

## 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 21-41: shows the input and output voltages for the filter.
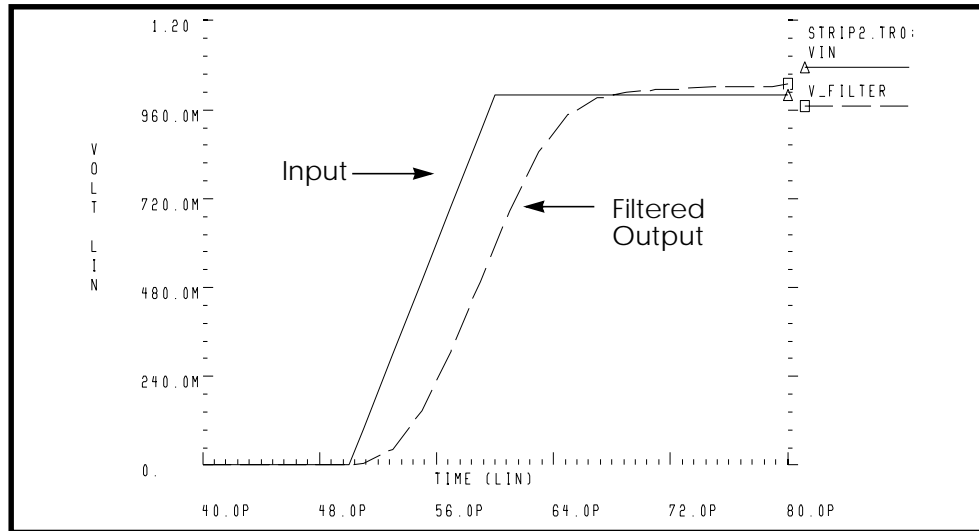
**Figure 21-41: Multistage RC Filter Input and Output**

## 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 21-42, 21-43, and 21-44. 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.
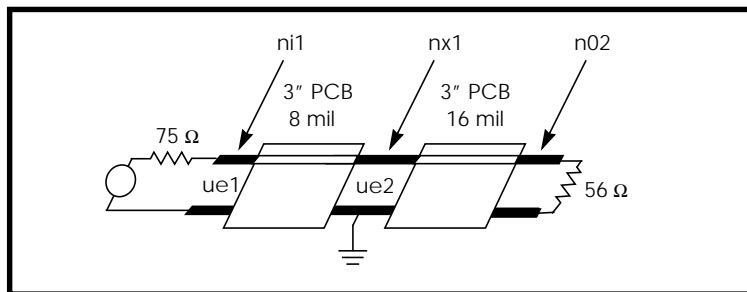
**Figure 21-42: Mathematical Transmission Line Structure**
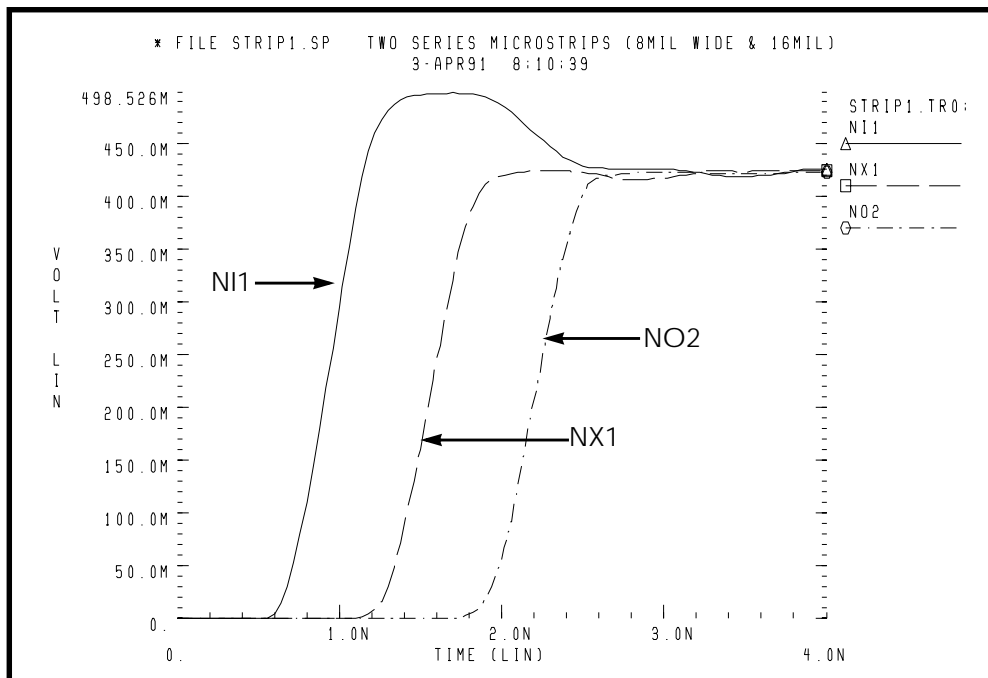


**Figure 21-43: Waveforms in Mismatched Transmission Line Structure**
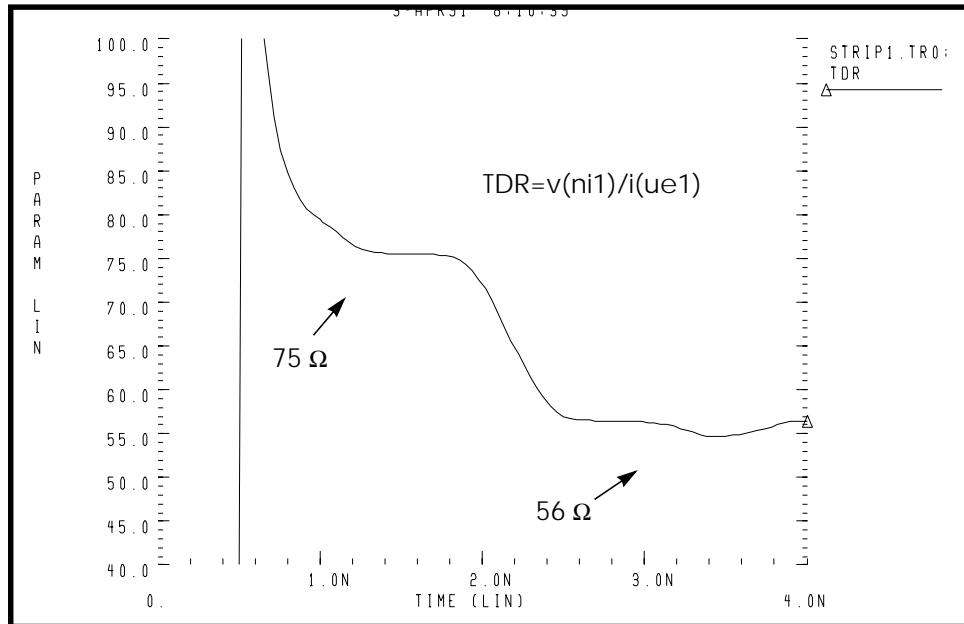
**Figure 21-44: –Impedance from TDR at Input**

**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
```

## The W Element

The W-element – Multiconductor Lossy Frequency Dependent Transmission
Line provides advanced modeling capabilities for transmission lines.

It supports the following types of analyses:

- AC
- DC
- Transient

### Syntax now conforms to generic Spice format

The W-element card has been extended to accept generic Spice format
(parameters come *after* the list of terminal nodes). Previously, the number of
signal conductors, *N*, was placed *before* the list of nodes, which was not
compatible with some automatic netlist generation tools. Presently, the nodes
and parameters in the W-element card can be mixed freely. The old format is
still supported. Refer to "Element Card" on page 21-76 of this chapter for the
W-element card description.

### Improved time-step reversal control

The W element was modified to provide higher accuracy and fewer time-step reversals with Star-Hspice's time-step control algorithms that support time-step reversal.

### Improved *RLGC* matrix error check

The Maxwellian admittance matrices—$C_0$, $G_0$, and —$G_d$ are now checked for strict diagonal dominance. This permits for detecting input errors before the simulation started. Refer to "RLGC Matrices" on page 21-80 of this chapter for information on Maxwellian matrices.

### Support for parameter sweeps

W-element parameters can now be swept in ac, dc, and transient parameter sweeps.

### Support for optimization

Star-Hspice optimization can now be performed on W-element parameters.

### Support for Monte-Carlo statistical analysis

Monte-Carlo statistical analysis can now be performed on W-element parameters.

### Now accepts U model

The W-element has been extended to fully accept all U-model modes, including:

- *RLGC* input for up to 5 coupled conductors;
- Geometric input (planar, coax, twinlead);
- Measured-parameter input;
- Skin effect.

This provides backward compatibility with the U element, and eliminates the need to use field solver for simple geometries. Parameterization, algebraics, optimization, and Monte-Carlo statistical analysis of U-model parameters can

now be used with the W element. Refer to "Using W-Element with the U Model" on page 21-85 of this chapter for the detailed description of the W element to U model link.

### Platforms Supported

Sun SunOS 4, Sun Solaris, HP PA, IBM RS6000, DEC Alpha, SGI, PC Windows.

## Input

W element inputs **R**, **L**, **G**, **C**, $\mathbf{R_s}$ (skin-effect), and $\mathbf{G_d}$(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 element can also handle frequency-independent (*RLGC*) and lossless (*LC*) lines. *RC* lines are not supported.

W element also accepts U .model, thus providing backward compatibility with the U element and simple geometric/measured-parameter interface. Future releases will also input tabulated frequency samples of *RLGC* matrices to model arbitrary frequency dependence, and will have an integrated field solver to provide a general geometric interface.

## Features

- W element is based on a novel, very fast, accurate and robust transmission-line simulation method.
- Typical time-domain error is 0.0001%. Speed-up of up to 10,000 times over Star-Hspice's U element (lumped segmentation model) and up to 1000 times over Spice3 convolution model (see Table I).
- Uses the exact analytical solution for ac and dc analyses.
- No limit on the number of coupled conductors.
- No restrictions on the structure of input RLGC matrices, all matrices can be full.
- Accuracy does not depend on the transient speed, line length, or amount of loss, coupling or frequency dependence.

- No spurious ringing as that produced by U element (see Fig. 1).
- Frequency-dependent loss is accurately modeled in the transient analysis.
- Requires no manual adjustments (such as the number of lumps in the U element), gives accurate results with large time step.
- In Star-Hspice, lines can be terminated with arbitrary nonlinear devices.

**Table 21-9: Hspice Transient Runtime (Four-conductor line and Sun Sparc 20 workstation)**

| Line Model | Relative Runtime | Runtime | Memory |
|------------|------------------|---------|--------|
| W element | 1 | 0.17 s | 23 Kbytes |
| Single lumped resistors | 0.69 | 0.12 s | 153 Kbytes |
| Single lumped *RLGC* segment | 0.92 | 0.16 s | 172 Kbytes |
| U element | 23,513 | 3,827 s | 9,354 Kbytes |
| Spice3 convolution model | 687 | | |
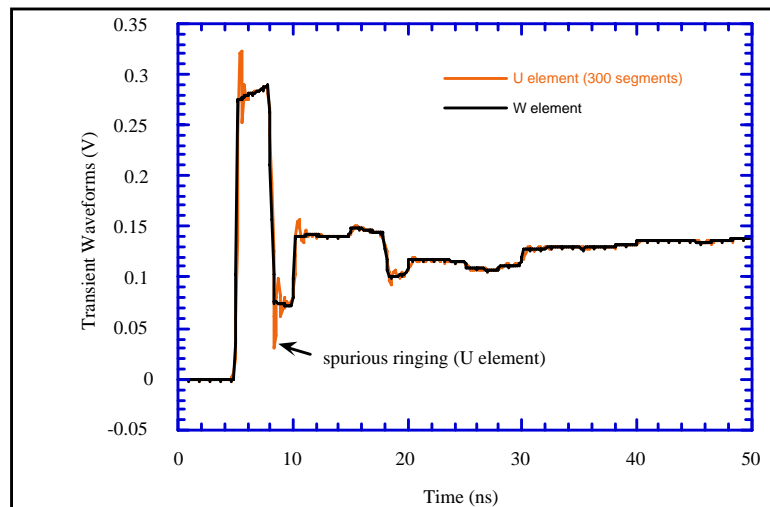


**Figure 21-45: Fig. 1.  Spurious ringing in U element, four-conductor lossy line**

## Technology

W element employs novel state-of-the-art transmission line simulation method,[1] very fast, accurate and robust.

ac and dc analyses use the exact analytical solution.  Transient analysis is based on the open-loop transmission line characterization, frequency-domain difference approximation, and indirect numerical integration.

The method introduces neither internal nodes nor current variables, and does not use computationally expensive convolution, thus resulting in the maximum efficiency.  At the same time, frequency-domain approximation allows the method to handle directly arbitrary frequency dependence.

The open-loop characterization assures simplicity of characteristic responses and, as a result, very accurate approximation.  The typical maximum relative error in the full frequency/time range from zero to infinity is under.

## Element Card

The element card specifies the number of line conductors, the terminal nodes, line length, and name of a U `.model` or the file that contains *RLGC* matrices. Future releases will also support entering *RLGC* matrices in W `.model` card.

### Format
```
Wxxxx  N=num_of_conductors
+      node1.1 ... node1.N node1'
+      node2.1 ... node2.N node2'
+      RFGCfile=RLGC_file_name or Umodel=U-model_name
+      1=line_length
```

---

1.

**Parameters**

| | |
|---|---|
| N | Number of signal conductors (does not include the reference conductor). |
| node1.1 ... node1.N | Nodes for the near-end signal-conductor terminals (see Fig. 2). |
| node1' | Node for the near-end reference-conductor terminal (see Fig. 2). |
| node2.1 ... node2.N | Nodes for the far-end signal-conductor terminals (see Fig. 2). |
| node2' | Node for the far-end reference-conductor terminal (see Fig. 2). |
| RLGCfile | Name of the file with *RLGC* parameters (see "RLGC File" on page 21-80 of this chapter). |
| Umodel | Name of U .model (see "Using W-Element with the U Model" on page 21-85 of this chapter). |
| l | Transmission-line length. |

**N+1 conductor line**

$[i_2]_1$  $[v_1]_1$   R(f), L(f), G(f), C(f)   $[v_2]_1$   $[i_2]_1$

1.1   $[i_2]_2$   $[v_1]_2$   **Signal**   $[v_2]_2$   $[i^2]_2$   2.1

1.1   $[v_1]_2$   $[v_2]_2$   2.2

$[i_2]_N$   $[v_1]_N$   **conductors**   $[v_2]_N$   $[i^2]_N$

1.1   +   **Reference conductor**   +   2.N

1'   −   −   2'

0   *l*   *x*
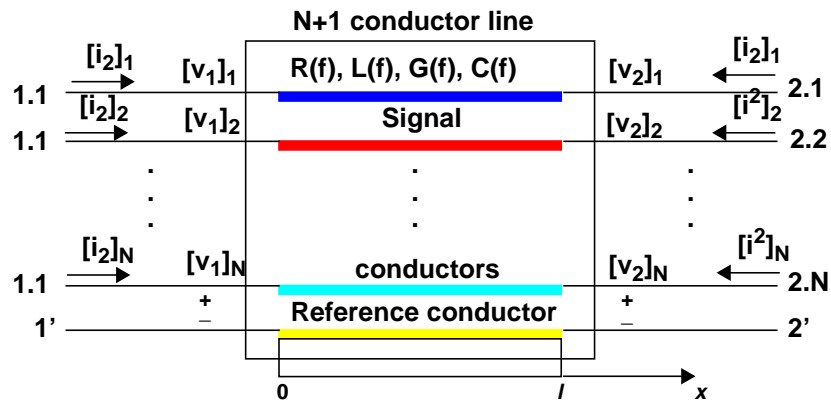
## Figure 21-46: Terminal Node Numbering

The order of parameters in the W-element card does not matter, 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 and Umodel can be specified in a single W-element card.

### Example, U model

The following Star-Hspice netlist is for a four-conductor line shown in Fig. 3.

```
* 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
```
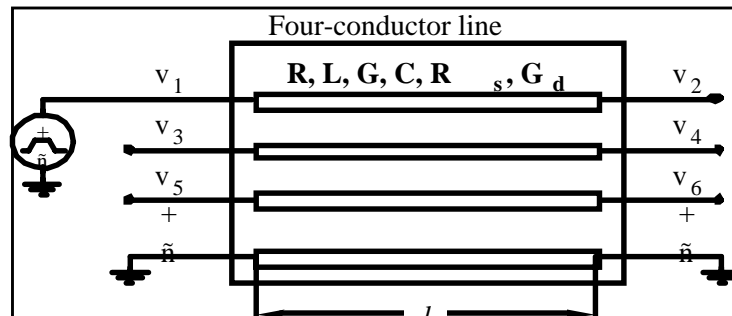


Fig. 3.  Four-conductor line.

## Example, *RLGC* file

The same circuit as in the above example but with the input parameters taken from *RLGC* file.

```
* 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
```

# RLGC File

*RLGC* text file contains frequency-dependent *RLGC* matrices per unit length.

## *RLGC* Matrices

*RLGC* matrices relate the voltage vector across the line, **v**, to the current vector along the line, **i**, in the Telegrapher's equations[1]

$$\frac{\partial}{\partial x} v(x,\omega) \;=\; (R(\omega) + j\omega L(\omega)) i(x, \omega)$$

$$\frac{\partial}{\partial x} i(x,\omega) \;=\; (G(\omega) + j\omega C(\omega)) v(x, \omega)$$

Presently, the frequency dependence is specified by the skin-effect matrix, $R_s$,

$$R(f) \;=\; R_0 + \sqrt{f R_s} \qquad\qquad\qquad \textbf{(1)}$$

and dielectric-loss matrix, $G_d$,

$$G(f) \;=\; G_0 + f G_d \qquad\qquad\qquad \textbf{(2)}$$

The inductance and capacitance matrices are currently constant

Future releases will also input tabulated frequency samples of *RLGC* matrices to specify an arbitrary frequency dependence.

W element has no restrictions on the structure of *RLGC* matrices. All matrices, including $R_0$ and $R_s$ can be full.

Since *RLGC* matrices are symmetric, *only the lower-triangular parts of the matrices are specified in the RLGC file*. The diagonal terms of $L_0$ and $C_0$ are positive nonzero. The diagonal terms of $R_0$, $R_s$, $G_0$, and $G_d$ are nonnegative (could be zero). Off-diagonal terms of impedance matrices $L_0$, $R_0$, and $R_s$ are

---

1. Normal, boldface lower-case, and boldface upper-case symbols denote scalars, vectors, and matrices, respectively.

nonnegative.  Off-diagonal terms of admittance matrices $\mathbf{C_o}$, $\mathbf{G_o}$, and $\mathbf{G_d}$ are nonpositive.  Off-diagonal terms of all matrices could be zero.

The elements of admittance matrices are related to the self/mutual admittances (as those inputted by U element) as follows

$$Y_{ii} = \sum_{j=1}^{N} Y_{ij}^{(self)/(mutual)}$$

$$Y_{ij} = Y_{ij}^{mutual}, i \neq j$$

where $\mathbf{Y}$ stands for $\mathbf{C_o}$, $\mathbf{G_o}$, or $\mathbf{G_d}$ (see an example in "RLGC matrices" on page 21-86 of this chapter).  The elements of the impedance matrices $\mathbf{L_o}$, $\mathbf{R_o}$, and $\mathbf{R_s}$ are the same as self/mutual impedances.

Since 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, i.e. admittance matrices are strictly diagonally dominant.

## Order of *RLGC* matrices

*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). |
| $\mathbf{L_o}$ | dc inductance matrix per unit length. |
| $\mathbf{C_o}$ | dc capacitance matrix per unit length. |
| $\mathbf{R_o}$(optional) | dc resistance matrix per unit length. |
| $\mathbf{G_o}$(optional) | dc shunt conductance matrix per unit length. |

**R$_s$**(optional)    Skin-effect resistance matrix per unit length,

$$R(f) \;=\; R_0 + \sqrt{f\,R_s}$$

**G$_d$**(optional)    Dielectric-loss conductance matrix per unit length,

$$G(f) \;=\; G_0 + f\,G_d$$

Optional parameters may be omitted and default to zero. If one of the optional parameters is specified, all preceding parameters, even if they are zero, have to be specified.

## Comments

An asterisk '*' comments out everything till the end of its line.

## Separators

Numbers can be separated by any combination of the following characters: space, tab, newline, ',', ';', '(', ')', '[', ']', '{' or '}'.

## Example

The following is example.rlc *RLGC* file from the four-conductor line example in "Example, U model" on page 21-78 in this chapter.

```
* 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 ($10^{-12}$). Star-Hspice simulation results are plotted in Fig. 4. As can be observed, 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.
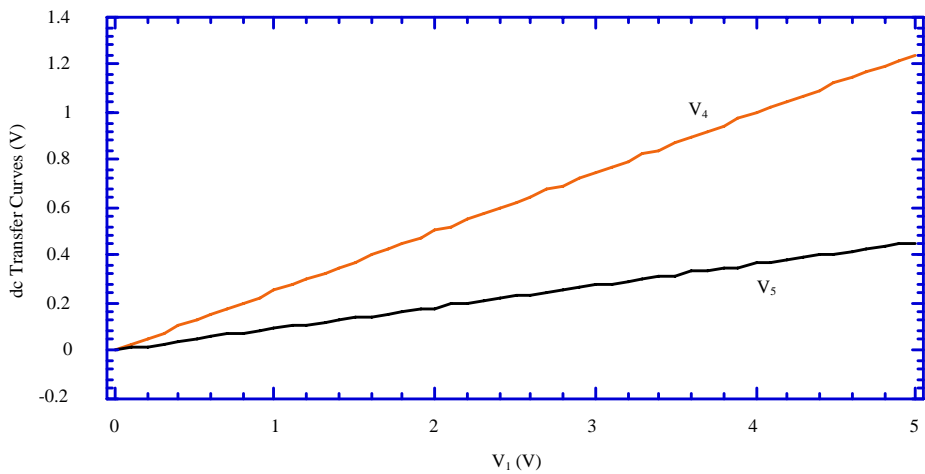
(a)
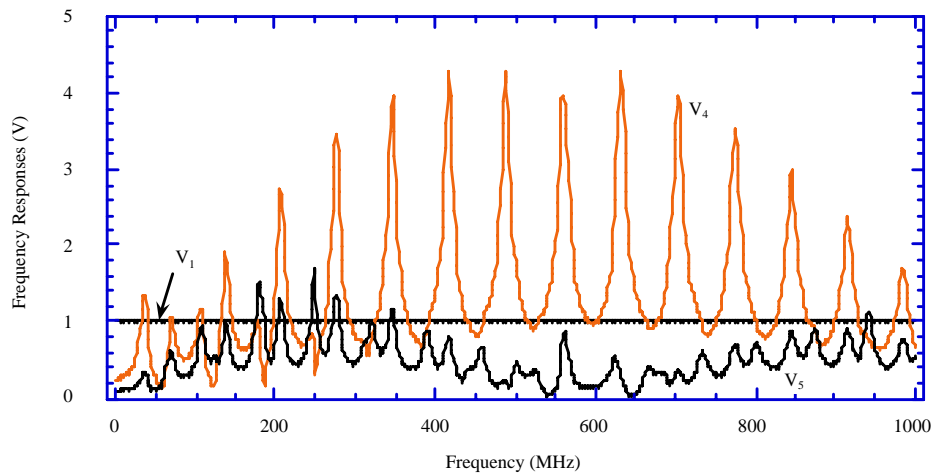


(b)
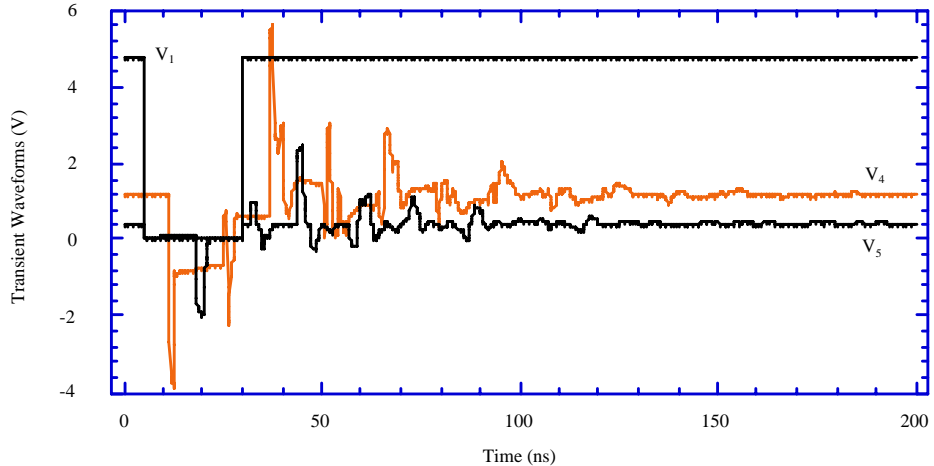
(c)

**Figure 21-47: Star-Hspice simulation results.  (a) dc sweep, (b) ac response, and (c)  transient waveforms.**

## Using W-Element with the U Model

The W element accepts U .model, thus providing backward compatibility with the U element, and taking advantage of U-model's geometric and measured-parameter  interfaces.

To use W element with U model, specify `Umodel=U-model_name` instead of `RLGCfile=RLGC_file_name` on the W-element card (see the "Example, U Model" section on page 3-7).

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 external lumped capacitor in Elev=1 and 3. Refer to Star-Hspice User's Manual, vol. III, section 20, for information on U model.

### *RLGC* matrices

*RLGC* matrices in W-element's *RLGC* file are in Maxwellian format, and in the U model, the matrices are in self/mutual format (refer to "RLGC Matrices" on page 21-80 in this chapter for conversion information). When using the U model, the W element performs the conversion internally. The following four-conductor example illustrates how U-model's *RLGC* matrices are related to W-element's *RLGC* matrices and used by the W element.

W-Element      U-Model Parameters
Parameters

$$\mathbf{L_0} = \begin{bmatrix} L11 & & \\ L12 & L22 & \\ L13 & L23 & L33 \end{bmatrix}$$

$$\mathbf{C_0} = \begin{bmatrix} Cr1 + C12 + C13 & & \\ -C12 & Cr2 + C12 + C23 & \\ -C13 & -C23 & Cr3 + C13 + C23 \end{bmatrix}$$

$$\mathbf{G_0} = \begin{bmatrix} Gr1 + G12 + G13 & & \\ -G12 & Gr2 + G12 + G23 & \\ -G13 & -G23 & Gr3 + G13 + G23 \end{bmatrix}$$

$$\mathbf{G_d} = \begin{bmatrix} 0 & & \\ 0 & 0 & \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{R_0} = \begin{cases} \begin{bmatrix} R11+Rrr & & \\ 0 & R22+Rrr & \\ 0 & 0 & R33+Rrr \end{bmatrix}, \text{Nlay}=1 \text{ (no skin effect)} \\[2em] \begin{bmatrix} R1c+Rrc & & \\ 0 & R2c+Rrc & \\ 0 & 0 & R3c+Rrc \end{bmatrix}, \text{Nlay}=2 \text{ (skin effect present)} \end{cases}$$

$$\mathbf{R_s} = \begin{cases} \begin{bmatrix} 0 & & \\ 0 & 0 & \\ 0 & 0 & 0 \end{bmatrix}, \text{Nlay}=1 \text{ (no skin effect)} \\[2em] \frac{1}{\sqrt{f_{skin}}}\begin{bmatrix} R1s+Rrs & & \\ 0 & R2s+Rrs & \\ 0 & 0 & R3s+Rrs \end{bmatrix}, \text{Nlay}=2 \text{ (skin effect present)} \end{cases}$$

## Dielectric-loss matrix

Since U model does not input the dielectric loss matrix $\mathbf{G_d}$, **W** element defaults $\mathbf{G_d}$ to zero when using the U-model input.  In future releases, **W** element will have its own .model with $\mathbf{G_d}$ capability.  Presently, use *RLGC* file to specify nonzero $\mathbf{G_d}$.

## Skin-effect matrix

The skin-effect resistance $\mathbf{R_s}$ is used differently by U and W elements.  For W element, the $\mathbf{R_s}$ matrix specifies the square-root dependence of the frequency-dependent resistance,

$$R(f) = R_0 + \sqrt{f}R_s$$

For U element, **R** is the value of skin resistance at the frequency , and, for the simulation, U element uses constant resistance

$$R = R_c + R_s,$$

where the core resistance $R_c$ is equivalent to the W-element's dc resistance $R_0$. The frequency at which U element computes $R_s$

$$f_{skin} = \frac{1}{15 \ RISETIME}.$$

If .option RISETIME is not specified, U element uses Tstep from .tran card instead.

For U models with *RLGC* input, Elev=2, W element uses $R_s$ values unchanged as specified in the U model by the user. For the geometric input, $\sqrt{\frac{f}{f_{skin}}}$ v=1, $R_s$ is computed internally by U model, and W element divides it by

to obtain the $R_s$ of (1). Therefore, for Elev=1, $R_s$ reported in the U-model printout will not be the same as $R_s$ actually used by the W element. The measured-parameter input, Elev=3, does not support skin effect.

## Units

Star-Hspice normally uses SI units for all values. The SI units for W element parameters are:

$$t[s], \ f[Hz], \ l[m], \ R_o\left[\frac{\Omega}{m}\right], \ R_s\left[\frac{\Omega}{m\sqrt{Hz}}\right], \ G_o\left[\frac{S}{m}\right], \ G_d\left[\frac{S}{m \, Hz}\right], \ L_o\left[\frac{H}{m}\right], \ \text{and} \ C_o\left[\frac{F}{m}\right].$$

W element uses no internal absolute constants, thus allowing for consistent use of any other units. However, this is not in general recommended in Star-Hspice, as it uses absolute constants optimized for SI units.

## ac Analysis

Frequency-domain responses of transmission lines oscillate (see Fig. 4(b)) with a period of

$$\frac{1}{propagation delay}$$

because of the periodic cancellation of incident and reflected waves (the comb-filter effect).  The best ac results are obtained with linear frequency scale.

## Time-Step Control

W element will provide accurate results with just 1-2 time steps per excitation transient (0.1 ns in the above example).  The element supports Star-Hspice's iteration count (the option LVLTIM=0) and DVDT (LVLTIM=1 or 3) time step control algorithms.  The LTE (LVLTIM=2) algorithm is not currently supported. Star-Hspice's default time-step control algorithm is DVDT.

W element limits the maximum time step by the smallest transmission line delay in the circuit.

W element supports the TLINLIMIT option as does T element.  The default value of TLINLIMIT=0 enables special breakpoint building which 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 while the wave is propagating along the line. Star-Hspice starts to increase the time step, and when the wave finally reaches the termination, the accuracy is compromised.  To prevent this, for longer lines excited with short pulses, .option DELMAX should be set to limit the time step *f* to 0.5-1 of the excitation transient.

# .option RISETIME

.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 excitation transient actually is, will only worsen the simulation accuracy.

# Computing $R_s$ and $G_d$

If the values of $\mathbf{R}(f)$ and $\mathbf{G}(f)$ are known at certain frequencies, $\mathbf{R_s}$ and $\mathbf{G_d}$ are computed by fitting the samples of $\mathbf{R}(f)$ and $\mathbf{G}(f)$ to the expressions (1) and (2). For instance, if we know $\mathbf{R}(0)$ and $\mathbf{R}(f_1)$, then from (1) we can compute

$$R_0 = R(0), \ R_s = \frac{1}{\sqrt{f_1}}(R(f_1) - R_0) \tag{3}$$

Similarly, if we know $\mathbf{G}(0)$ and $\mathbf{G}(f_1)$, then from (2) we obtain

$$G_0 = G(0), \ G_s = \frac{1}{f_1}(G(f_1) - G_0) \tag{4}$$

### Example

Given a three-conductor line with the following,

$$R(0) = \begin{bmatrix} 10 & 0 \\ 0 & 15 \end{bmatrix} \frac{\Omega}{m} \text{ and}$$

$$R(1\,GHz) = \begin{bmatrix} 500 & 0 \\ 0 & 700 \end{bmatrix} \frac{\Omega}{m}$$

compute $R_s$.

From (3) we compute

$$R_0 = R(0) = \begin{bmatrix} 10 & 0 \\ 0 & 15 \end{bmatrix} \frac{\Omega}{m}$$

$$R_s = \frac{1}{\sqrt{f_1}}(R(f_1) - R_0) = \frac{1}{10^9}\left(\begin{bmatrix} 500 & 0 \\ 0 & 700 \end{bmatrix} - \begin{bmatrix} 10 & 0 \\ 0 & 15 \end{bmatrix}\right) = \begin{bmatrix} 0.015495 & 0 \\ 0 & 0.021662 \end{bmatrix}\frac{\Omega}{m\sqrt{Hz}}$$

For a larger number of frequency samples, the constrained least-square fitting[1] or method of averages[2] can be used. The dc values of $R(f)$ and $G(f)$ should be always matched exactly. Future releases of **W** element will directly input the frequency samples and perform the fitting internally.

## Computing RLGC Parameters from Cable Specifications

Standard cable specifications may not provide all of the *RLGC* parameters, but specify instead characteristic impedance and attenuation at certain frequencies.

The high-frequency characteristic impedance is

$$Z_c = \sqrt{\frac{L}{C}} \qquad\qquad (5)$$

And the formula for the high-frequency attenuation is

$$\alpha = e^{-\frac{Rl}{2Z_c}}$$

---

1. S. J. Leon, *Linear Algebra with Applications,* 3rd ed.  New York, NY:  Macmillan Publishing Company, 1990.
2. I. N. Bronshtein and K. A. Semendyaev, "Spravochnik po matematike.  Moskva:  Nauka, 1967, str. 578.  In Russian.

The shunt conductance of the cable dielectric, *G,* is usually sufficiently low to be neglected.

## Example

Compute W-element *RLGC* parameters for Category 5 UTP (twisted-pair) cables. We will need the following parameters from the cable specifications.

Mutual capacitance: *C*=45.9 *pF/m.*

DC resistance: $R_0 = 0.0938\ \Omega/m$.

Characteristic impedance at high-frequency: $Z_c = 100\ \Omega$.

Attenuation at high-frequency $f = 100\ MHz$ at cable length $l = 305\ m$: $\alpha_{dB} = 67\ dB$.

From (5), we can compute the inductance as

$$L = CZ_c^2 = 45.9 \cdot 10^{-12}\ 100^2 = 459\ nH/m.$$

The high-frequency attenuation

$$\alpha = 10^{-\frac{\alpha_{dB}}{20}} = 10^{-\frac{67}{20}} = 0.02529.$$

From (6), the high-frequency resistance is

$$R(f = 100\ MHz) = -\frac{2Z_c}{l}\ln(\alpha) = -\frac{2 \cdot 100}{305}\ln(0.02529) = 5.058\ \Omega/m.$$

Finally, from (4) we compute the skin-effect resistance

$$R_s = \frac{1}{\sqrt{f}}\left(R(f) - R_0\right) = \frac{1}{\sqrt{100 \cdot 10^6}}(5.058 - 0.0938) = 4.964 \cdot 10^{-4}\ \frac{\Omega}{m\sqrt{Hz}}.$$

## Future Enhancements

■ Input *RLGC* matrices in a .model as alternative to *RLGC* file.

■ Input tabulated frequency samples of RLGC matrices to model arbitrary frequency dependence.

■ Incorporate a field-solver into W element to implement a general geometric interface.

## When to Use Transmission Line Models

Interconnects should be modeled as transmission lines if the interconnect delay greatly exceeds the signal transient time.  This is always the case for communication and instrumentation cables, PC-board interconnects, and majority of multichip-module (MCM) interconnects.  Only critical on-chip interconnects, over several millimeters in length and with transient speeds in excess of 100 ps, should be modeled as transmission lines.

## A Note on Transmission Lines

To illustrate the physical processes of wave propagation and reflection in transmission lines,[1] let us consider the line with simple terminations excited with the voltage step as shown in Figure 21-48.

---

1.  A good source on transmission lines is:  H. B. Bakoglu, *Circuits, Interconnections and Packaging for VLSI*.  Reading, MA: Addison-Wesley Publishing Company, 1990.
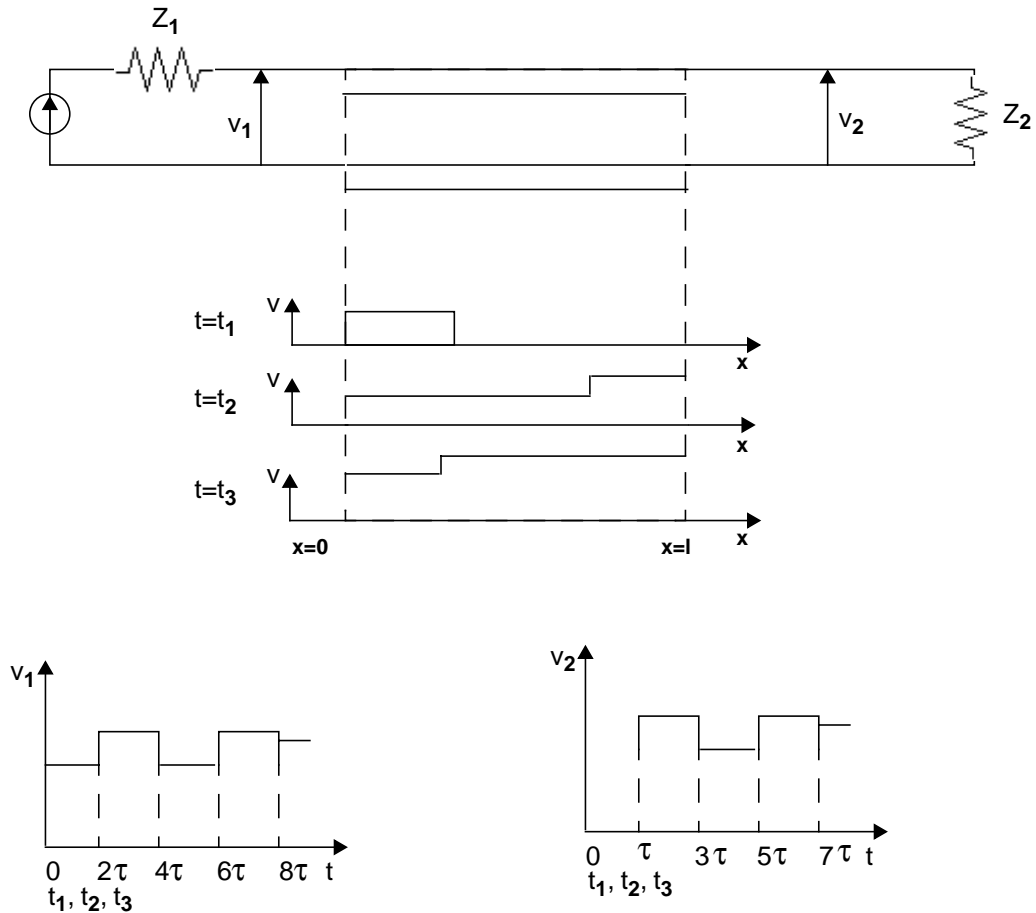
**Figure 21-48: Propagation of a voltage step in transmission line.**

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 has arrived at the far end of the transmission line, 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 has arrived back at the near end, reflected again, and is again propagating in the forward direction.  The voltage at the source end is the sum of the attenuated voltage from the source $e_1$, the backward wave, and reflected forward wave.
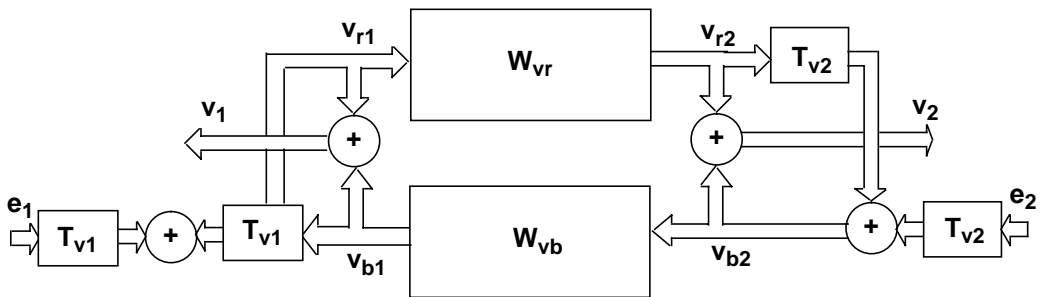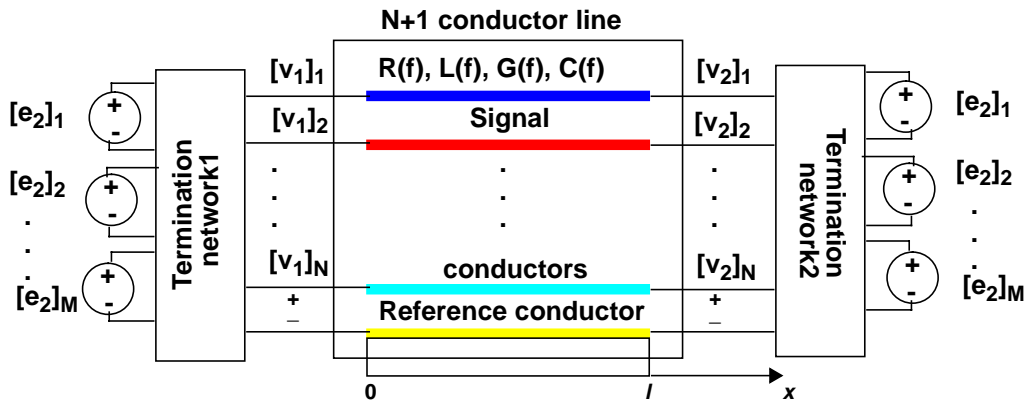
**Figure 21-49: System model for transmission lines.**

The process in Figure 21-49 can be summarized as follows.

■ Signals from the excitation sources attenuate 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 so on in a loop.

■ The voltage at any point along the line, including the terminals, is a superposition of the forward and backward propagating waves.

This process can be represented by the system diagram in Figure 21-49.[2] $\mathbf{W_{vr}}$ and $\mathbf{W_{vb}}$ are the forward and backward matrix propagation functions for voltage waves; $\mathbf{T_1}$, $\mathbf{T_2}$ and $\Gamma_1$, $\Gamma_2$ stand for the near- and far-end matrix transmission and reflection coefficients. 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 allows one to write the macrosolution for a distributed system without complicated mathematical derivations.

As can be observed from Figure 21-49, transmission line along with terminations forms a feedback system. 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 21-47(b).

An important special case occurs when the line terminates in another line. The system diagram representation of a line-to-line junction is shown in
. It can be used for the solution of multilayered plane-wave propagation problems, for the analysis of many common waveguide structures, and for the derivation of generalized transmission and reflection coefficient formulas and scattering parameter formulas.

2. Dmitri Kuznetsov, *Transmission Line Modeling and Transient Simulation*, 1993, htttp://www.comports.com/vdm/book.html.
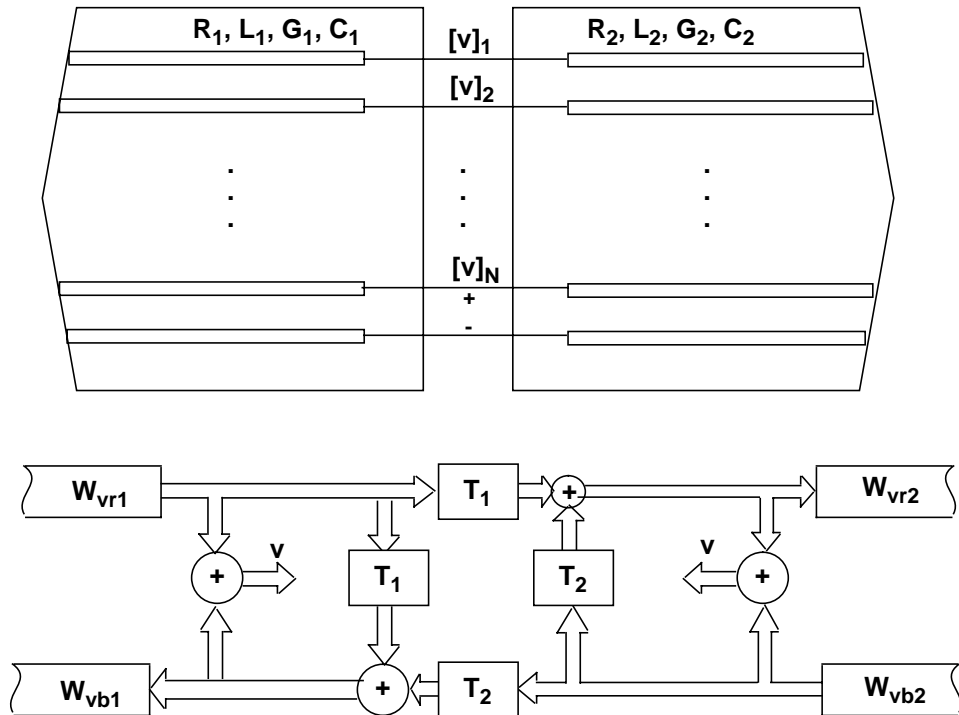
**Figure 21-50: System model for a line-to-line junction.**

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 21-51). Lines with frequency-dependent parameters, and, therefore, all real lines, do not contain the frequency-independent attenuation component. Figure 21-51 also lists some common sources of frequency dependence.
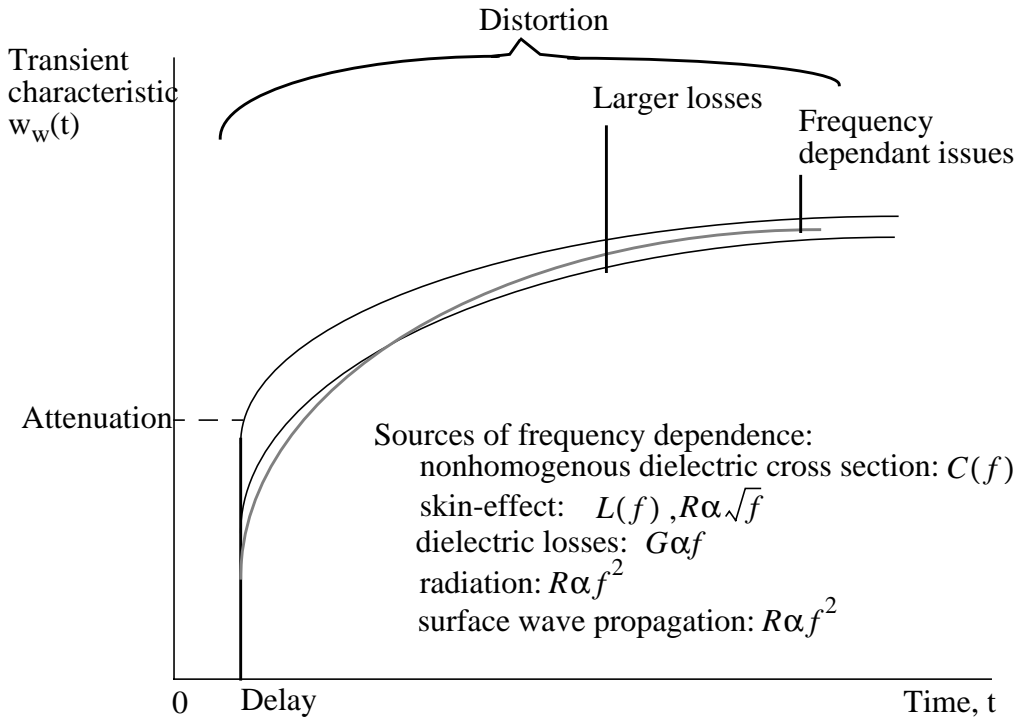
**Figure 21-51: Transient characteristic (the unit-step response) of a propagation function.**

# 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 21-52:.
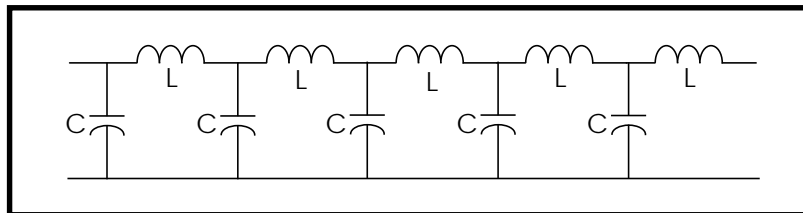


**Figure 21-52: Equivalent Circuit Model of a Lossless Transmission Line**

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 21-53: shows the equivalent circuit model of a lossy transmission line, with distributed "lumps" of R, L, and C elements.
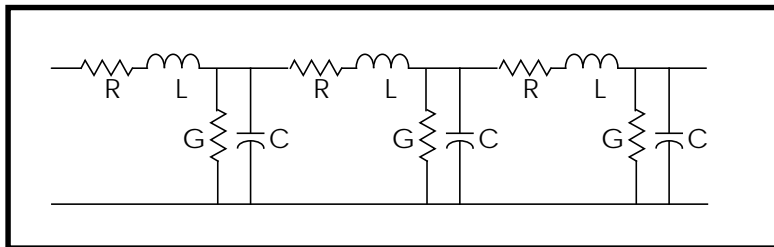


**Figure 21-53: Equivalent Circuit Model of a Lossy Transmission Line**

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 $90^o$ in the negative or direction to generate the voltage across the capacitor. The impedance of an inductor rotates the current $90^o$ in the positive direction to generate the voltage across the inductor. For a resistor, the current and voltage have the same phase.

In the frequency domain, when all signals are sine waves in the time domain, the impedance of a capacitor and an inductor is frequency dependent, decreasing with frequency for a capacitor and increasing with frequency for an inductor. The impedance of a resistor is constant with frequency.

In the real world of finite dimensions and engineered materials, ideal circuit elements have parasitics associated with them, which cause them to behave in complex ways that are very apparent at high frequencies.

## Characteristic Impedance

A controlled impedance transmission line is a pair of conductors that have a uniform cross section and uniform distribution of dielectric materials down their length. A short segment, $\Delta x$, of the transmission line has a small capacitance associated with it, $\Delta C$, which is the capacitance per length, $C_L$, times the $\Delta x$:

$$\Delta C = C_L \Delta x$$

When a voltage signal is introduced at one end, the voltage between the conductors induces an electric that propagates the length of the line at the speed of light in the dielectric. As the voltage signal moves down the line, each new section of line charges up. The new section of line, $\Delta x$, is charged up in a time $\Delta t$:

$$\Delta t = \frac{\Delta x}{v}$$

If the voltage (V) moves down the line at a constant speed and the capacitance per length is uniform throughout the line, then the constant voltage applied to the front end draws a constant charging current (I):

$$I = \frac{\Delta Q}{\Delta t} = \frac{\Delta CV}{\Delta t} = \frac{C_L \Delta x V}{\Delta t} = C_L v V$$

This constant voltage with constant current has the behavior of a constant impedance (Z):

$$Z = \frac{V}{I} = \frac{V}{C_L v V} = \frac{1}{C_L v}$$

The impedance is determined by the speed of the signal and the capacitance per length of the pair of conductors, both intrinsic properties of the line. This intrinsic impedance is termed the characteristic impedance of the line ($Z_0$).

If a measurement is made at one end of the line in a short time compared to the round trip time delay, the line behaves like a resistor with a resistance equal to the characteristic impedance of the line. Transmission line effects are only important when rise times are comparable or shorter than the round trip time delay.

For example, if the rise time of a device is 1 ns, and it drives an interconnect trace in FR4 which is longer than three inches, the load on the device during the risetime is purely resistive. For CMOS devices, which are used to drive high resistance loads, the typical 50 ohm resistance they see initially can significantly distort the waveform from what is expected.

It is only during the initial surge of the voltage that a transmission line behaves as a constant impedance, with a value equal to its characteristic impedance. For this reason the characteristic impedance of a line is also called the surge impedance. The surge time during which the impedance is constant is the round trip time of flight, or twice the time delay. Reflections from the far end complicate the electrical behavior of the line after the surge time.

The instantaneous impedance measured at the front end of a transmission line is a complicated function of time. It depends on the nature of the terminations at the far end. When the line is shunted to ground with a resistor of value equal to the characteristic impedance of the line, there is no reflection back, and the front end of the line behaves as a resistive load. When the termination at the far end is open, the impedance at the front end starts out at the characteristic impedance and eventually, after multiple reflections, approaches an infinite impedance. During some periods the instantaneous impedance may be zero. These transient effects are fully simulated with T elements and U elements in Star-Hspice.

## Inductance

### Mutual Inductance and Self Inductance

The most confusing, subtle and important parameter in high-speed packaging and interconnect design is inductance. It plays a key role in the origin of simultaneous switching noise, also called common ground inductance, and a key role in crosstalk between transmission line structures.

## Operational Definition of Inductance

Consider an inductor to be any section of circuit element which carries current: an interconnect trace, a ground plane, a TAB lead frame, a lead in a DIP package, the lead of a resistor or a pin in a connector. An inductor does not have to be a closed circuit path, but can be a small section of a circuit path.

A changing current passing through an inductor generates a voltage drop. The magnitude of the voltage drop ($\Delta V$) 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:

$$V_1 = L_{11}\frac{dI_1}{dt} + L_{12}\frac{dI_{I2}}{dt} + L_{13}\frac{dI_3}{dt} + L_{14}\frac{dI_4}{dt}$$

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 ($\tau_{edge}$). Figure 21-54: illustrates these features.
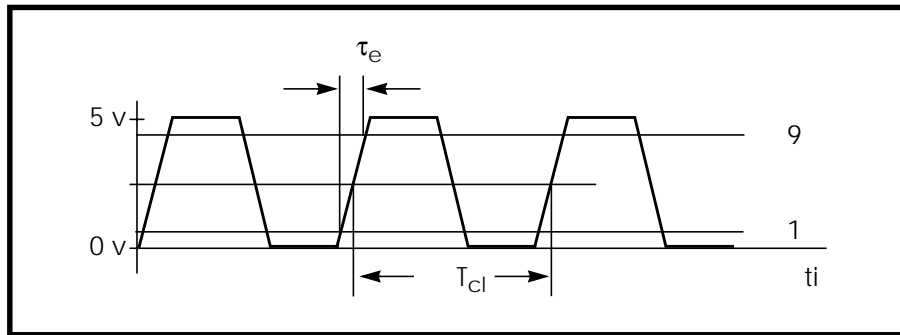


**Figure 21-54: Clock Waveform**

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 21-10: 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 |
| $\omega$ | 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 |
| $\varepsilon_r$ | relative dielectric constant, $\varepsilon/\varepsilon_0$, dimensionless |
| r | reflection coefficient, dimensionless |
| DZ | a small change in characteristic impedance |
| $G_L$ | conductance per length units of Mhos (Siemens)/length |
| $\tan(\delta)$ | dissipation factor of a material, dimensionless |
| d | skin depth of a conductor, units of length such as meter |
| $\rho$ | resistivity of a conductor, units of ohm-length |

## Table 21-10: Transmission Line Terms

| | |
|---|---|
| $\mu_0$ | permeability of free space = 4 p x 10-7 Henry/meter |
| $\alpha$ | 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 |
| $\varepsilon_0$ | permittivity of free space=0.225 pF/inch=0.0885 pF/cm |
| $\varepsilon_{eff}$ | 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

$$TD = \frac{d}{6} \qquad \text{[TD in ns, d in inches]}$$

$$C_L = \frac{170}{Z_0} \qquad \text{[pF/inch]}$$

$$L_L = 0.172Z_0 \quad \text{[nH/inch]}$$

*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 $\Delta Z$, 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:

$$r = \frac{L}{2 Z_0 \tau_{edge}}$$

Reflection from a lumped C load to ground, on a transmission line:

$$r = \frac{C Z_0}{2 \tau_{edge}}$$

## Loss and Attenuation

Skin depth of a conductor:

$$\delta = \sqrt{\frac{\rho}{\pi\mu_0 f}} = 500 \cdot \sqrt{\frac{\rho}{f}}$$

[$\delta$ in meters, $\rho$ in ohm·meter, $f$ in Hz: in copper, at $f$ = 1e+9 and $\rho$ =1.7e-8 ohm·meter, $\delta$ = 2.0e-6]

Low loss approximation for attenuation per length:

$$\alpha = \frac{1}{2}\left(\frac{R_L}{Z_0} + G_L Z_0\right) \quad \text{[nepers/length]}$$

$$\alpha = 4.34\left(\frac{R_L}{Z_0} + G_L Z_0\right)\text{[dB/length]}$$

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, $t < \delta$:

$$\alpha = 43.4\frac{\rho}{t w Z_0} \text{ [dB/inch, with $\rho$ in ohm·meter, t in microns, w in mils]}$$

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, $t > \delta$:

$$\alpha = 0.55\frac{\sqrt{\rho f}}{w} \quad \text{[dB/inch, with $\rho$ in ohm·meter, w in mils, f in GHz]}$$

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:

$$R_L = n \cdot R_{sq} \qquad \text{[ohm/inch, R}_{sq} \text{ in ohm/sq]}$$

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%:

$$C_L = 0.45\varepsilon_r\left(\frac{w}{h}\right) \qquad \text{[pF/inch]}$$

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:

$$L_L = 26 \qquad [nH/inch \text{ of perimeter}]$$

Inductance per length of controlled impedance line, when the return line is a reference plane:

$$L_L = 0.086Z_0\sqrt{\varepsilon_r} = \frac{7.3\varepsilon_r}{C_L} \qquad [nH/inch \text{ with } C_L \text{ in } pF/inch]$$

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 dependance to the characteristic impedance and a frequency dependance to the speed of propagation, often called dispersion. Both the first order and second order effects of series resistance generic to lossy transmission lines are simulated using the Star-Hspice U model.

## The Physical Basis of Loss

The origin of loss is the series resistance of the conductors and the dielectric loss of the insulation. Conductor resistance is considered in two parts, the DC resistance and the resistance when skin depth plays a role. The dielectric loss of

the insulation, at low frequency, is described by the material conductivity ($\sigma$) (SIG in Star-Hspice), and at high frequency, by the dissipation factor, $\tan(\delta)$. These material effects contribute a shunt conductance to ground (G).

In a planar interconnect such as a microstrip or stripline, the resistance per length of the conductor $R_L$ is

$$R_L = \frac{\rho}{wt}$$

$\rho$ = bulk resistivity of the conductor
w = the line width of the conductor
t = thickness of the conductor

**Example 1: FR4, 5 mil wide line, half ounce copper:**

$R_L$ = 0.24 ohm/inch

**Example 2: Cofired ceramic, 4 mil wide, 0.75 mils thick Tungsten:**

$R_L$ = 2.7 ohm/inch

**Example 3: Thin film copper, 1 mil wide, 5 microns thick:**

$R_L$ = 3.6 ohm/inch

## Skin Depth

At high frequency, the component of the electric field along the conductor, which drives the current flow, does not penetrate fully into the depth of the conductor. Rather, its amplitude falls off exponentially. This exponential decay length is called the skin depth ($\delta$). When the signal is a sine wave, the skin depth depends on the conductor's resistivity ($\rho$), and the sine wave frequency of the current (f):

$$\delta = \sqrt{\frac{\rho}{\pi\mu_0 f}} = 500 \cdot \sqrt{\frac{\rho}{f}} \text{ [} \delta \text{ in meters, r in ohm·meter, f in Hz]}$$

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 ($\sigma$), typically specified as between $10^{-12}$ and $10^{-16}$ siemens/cm, is often an upper limit, rather than a true value. It is also very temperature and humidity sensitive. The shunt conductance per length ($G_L$) depends on the geometrical features of the conductors in the same way as the capacitance per length ($C_L$). It can be written as:

$$G_L = \sigma \frac{C_L}{\varepsilon_0 \varepsilon_r}$$

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:
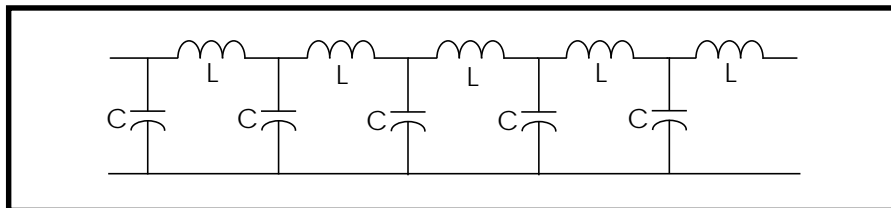


**Figure 21-55: Lossless Transmission Line Model**

In the lossy transmission line model, the series resistance and dielectric conductance are introduced into the equivalent circuit model:
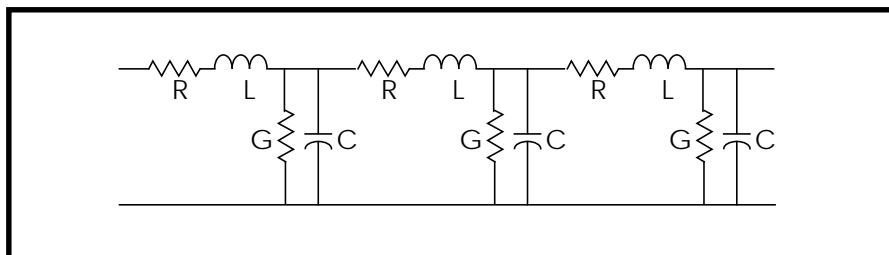
**Figure 21-56: Lossy Transmission Line Model**

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, $\gamma$, is:

$$\Upsilon = \alpha + j\beta = \sqrt{(R_L + j\omega L_L)(G_L + j\omega C_L)}$$

In the propagation phase term, $\beta$ is related to the phase velocity by:

$$v = \frac{\omega}{\beta}$$

To first order, when $R_L \ll \omega L_L$ 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 ($\alpha$) 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 ($\alpha$d) determines the fraction of the signal amplitude that remains after propagating the distance (d). When $\alpha$ has the units of dB/length, the fraction of signal remaining is:

$$10^{\left(\frac{-\alpha d}{20}\right)}$$

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:

$$\alpha_{dielectric} = 2.3 f \tan(\delta) \sqrt{\varepsilon_{eff}} \quad \text{[dB/inch, f in GHz]}$$

The worst case and highest attenuation per length is exhibited by FR4 boards, with $\tan(\delta)$ of the order of 0.02 and a dielectric constant of 5. The attenuation at 1 GHz is about 0.1 dB/inch. For an interconnect 10 inches long, this is 1 dB of attenuation, which would leave about 90% of the signal remaining, comparable to the attenuation offered by a conductor with 1 ohm/inch resistance.

When the resistance per length is larger than 1 ohm/inch— for example in cofired ceramic and thin film substrates, and the dissipation factor is less than 0.005, the attenuation from the conductor losses can be on the order of 10 times greater than dielectric loss. In these applications, the dielectric losses can be ignored.

## Integrating Attenuation Effects in Star-Hspice

All of the first order effects of attenuation are automatically simulated with the U element in Star-Hspice.

With ELEV=1, the inputs can be the cross sectional geometry and the material properties of the conductor, bulk resistivity (RHO), the relative dielectric constant of the insulation (KD), and the conductivity of the dielectric (SIG). From these features, the equivalent capacitance per length, inductance per length, series resistance per length, and conductance per length are calculated by Star-Hspice.

With ELEV=2, the equivalent capacitance per length, inductance per length, series resistance per length, and conductance per length are input directly using estimates, measurements or third-party modeling tools.

Star-Hspice automatically generates a model for the specified net composed of a series of lumped elements that resembles the model for a lossy transmission line. The parameter WLUMPS controls the number of lumped elements included per wavelength, based on the estimated rise time of signals in the simulation.

The attenuation effects previously described are a natural consequence of this model. The U element allows realistic simulations of lossy transmission lines in both the AC and the transient domain.

# References

2.   *Handbook of Electronics Calculations for Engineers and Technicians*, McGraw Hill, pages 18-29, 18-23.

*Avant!*

# Chapter 22

# 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 Subcircuits
- Using Voltage and Current Controlled Elements
- Dependent Current Sources — G Elements
- Dependent Voltage Sources — F Elements
- Dependent Voltage Sources — E Elements
- Dependent Voltage Sources – H Elements
- Referencing Digital Files
- 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-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-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-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-Star-Hspice. You can calibrate the behavioral elements from a full transistor circuit using the built-in OPTIMIZE function.
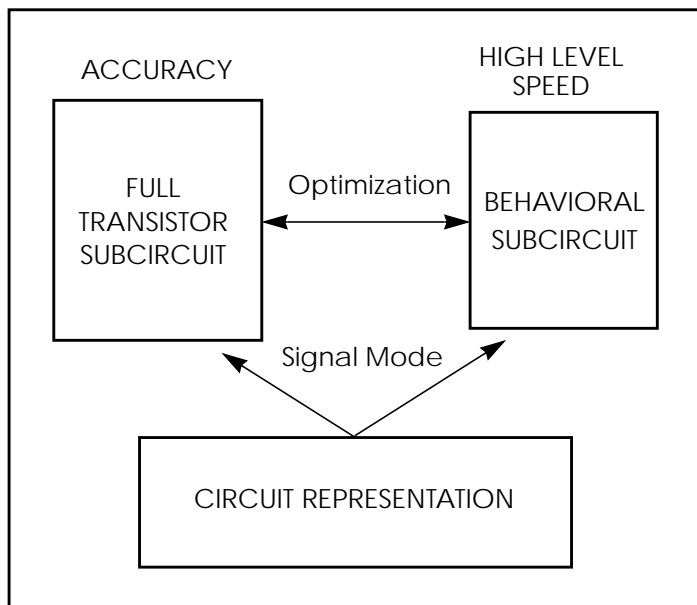
**Figure 22-1: Netlisting by Signal Mode**

### 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 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-Star-Hspice provides a simple method for the naming of the subcircuit nodes and elements. To use this, you prefix the node or element with the subcircuit call name.
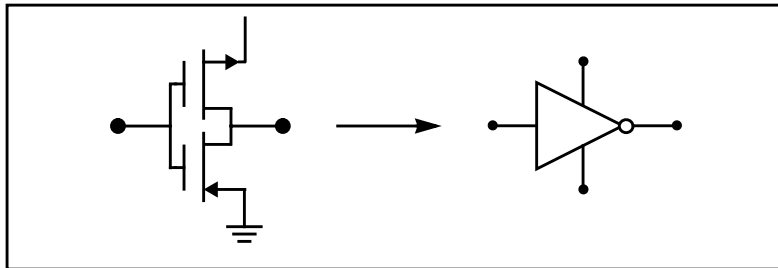


**Figure 22-2: Subcircuit Representation**

```
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.
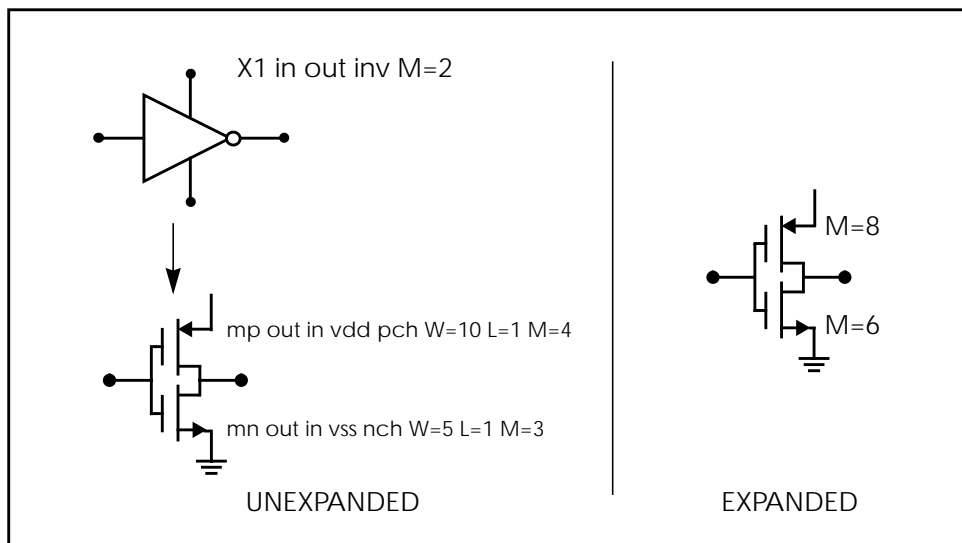


**Figure 22-3: Using Hierarchical Parameters Simplifies Flip-flop Initialization**

**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
```
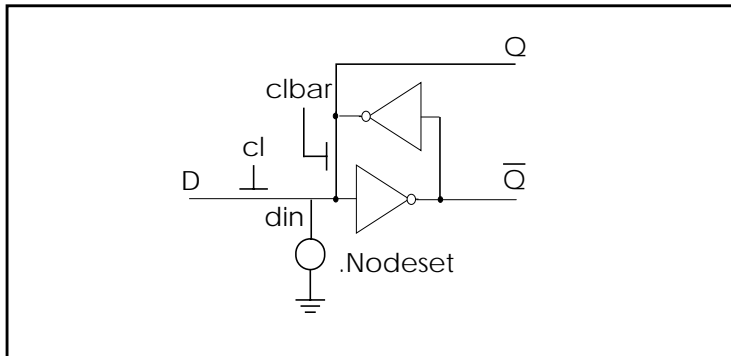


**Figure 22-4: D Latch with Nodeset**

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.

# .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 …* | snode 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. |

## Examples of Subcircuit Test

```
*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>
```

**Examples**
```
.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:              specifies:

*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.*

## Undefined Subcircuit Search

When a subcircuit call is in a data file that does not contain the subcircuit description, Star-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-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-Star-Hspice searches the system directories for the file called *inv.inc* and when found, places it in the calling data file.

# Using Voltage and Current Controlled Elements

In Star-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:

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

*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-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-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:

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

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 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-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 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 an Star-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 …*         the polynomial coefficients. When one coefficient is
                   specified, Star-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 -14.

*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-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 Star-Hspice releases prior to Release H93A. Use this method if you need 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 HPICE 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*             the keyword for voltage controlled capacitance element. VCCAP is an Star-Hspice keyword and should not be used as a node name.

*VCCS*               the keyword for voltage controlled current source. VCCS is an Star-Hspice keyword and should not be used as a node name.

*VCR*                 the keyword for voltage controlled resistor element. VCR is an 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

**Examples**

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 — 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 an 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 an 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-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 -14. |
| *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 |

### Examples

```
$ 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.

# 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' in+ in- <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 an 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* | the keyword for ideal op-amp element. OPAMP is an Star-Hspice keyword 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, 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 -14. |
| *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 |
|------|--------------------|
| *TRANS-* | the keyword for ideal transformer. TRANSFORMER is an Star-Hspice keyword<br>*FORMER* and should not be used as a node name. |
| *VCVS* | the keyword for voltage controlled voltage source. VCVS is an 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 |

**Examples**

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) = 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 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*          the keyword for current controlled voltage source. CCVS is an 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 an 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 -14. |
| *POLY* | polynomial dimension. If POLY(ndim) is not specified, a one-dimensional polynomial is assumed. Ndim must be a positive number. |
| *PWL* | piecewise linear function keyword |
| *SCALE* | element value multiplier |
| *TC1,TC2* | first and second order temperature coefficients. The SCALE is updated by temperature: |

$$\text{SCALEeff} \ = \ \text{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

|  | |
|---|---|
| *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 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).

# Referencing Digital Files

The U element can reference digital input and digital output models. If Star-Star-Hspice is being used by Viewlogic's Viewsim mixed mode simulator, the digital input comes from Viewsim. The state information comes from a digital file if Star-Star-Hspice is being run in standalone mode. Digital outputs are handled in a similar fashion. In this digital file mode, the input file is '*<designname>.d2a*' and the output file is named '*<designname>.a2d*'.

A2D and D2A statements accept the "\" 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-Star-Hspice issues an error message.
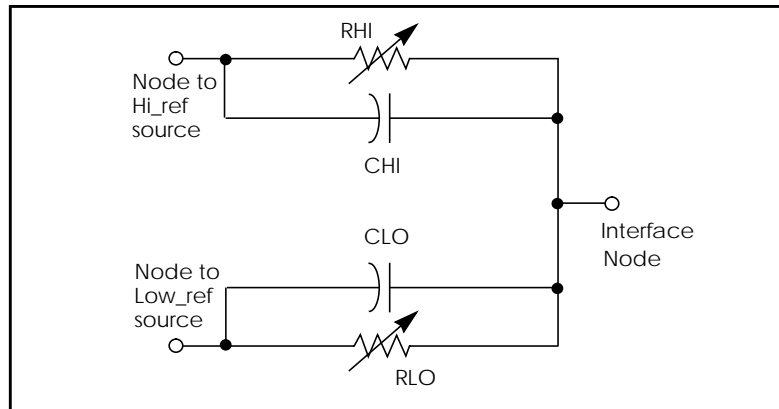


**Figure 22-5: Digital- to-Analog Converter Element**

## Digital Input Element and Model

The syntax is:

    U<name> <(interface) node> <(lo_ref)node> <(hi_ref)node> <(model) name>
    +        SIGNAME = <(digital signal) name>] [IS = (initial state)]

### Example

```
UC carry-in VLD2A VHD2A D2A SIGNAME=1 IS=0
VLD2A VLD2A 0 DC lo
VHD2A VHD2A 0 DC hi
```

The syntax is:

```
.MODEL <(model) name> DINPUT [(model parameters)]
```

### Examples

```
.MODEL D2A U LEVEL=5 TIMESTEP=0.1NS,
+ 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
```

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
```

### Digital to Analog Input Model Parameters

| Names(Alias) | Units | Default | Description |
|---|---|---|---|
| CLO | farad | 0 | capacitance to low level node |

| Names(Alias) | Units | Default | Description |
|---|---|---|---|
| CHI | farad | 0 | capacitance to high level node |
| SONAME | | | state "O" character abbreviation |
| SOTSW | sec | | state "O" switching time |
| SORLO | ohm | | state "O" resistance to low level node |
| SORHI | ohm | | state "O" resistance to high level node |
| S1NAME | | | state "1" character abbreviation |
| S1TSW | sec | | state "1" switching time |
| S1RLO | ohm | | state "1"resistace to low level node |
| S1RHI | ohm | | state "1" resistance to high level node |
| S2NAME | | | state "2" character abbreviation |
| S2TSW | sec | | state "2" switching time |
| S2RLO | ohm | | state "2" resistance to low level node |
| S2RHI | ohm | | state "2" resistance to high level ode |
| S19NAME | | | state "19" character abbreviation |
| 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) |

# Analog to Digital Output

General Form

U<name> <(interface)node> <(reference) node>  <(model) name>

+          [SIGNAME = <(digital signal) name>]

## Examples

```
vref VREFA2D 0 DC 0.0V
uco carry-out_2 VREFA2D a2d signame=12
```
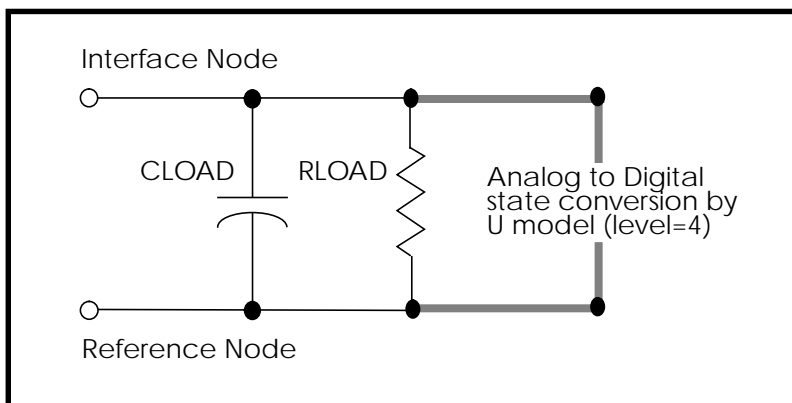


**Figure 22-6: – Analog to Digital Converter Element**

The syntax is:

.MODEL  < name> U LEVEL=4 [(model parameters)]

## Examples

```
* 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
```

### Analog to Digital Output Model Parameters

| Name(Alias) | Units | Default | Description |
|---|---|---|---|
| RLOAD | ohm | 1/gmin | output resistor |
| CLOAD | farad | 0 | output capacitor |
| CHGONLY | | 0 | 0: write each timestep, 1: write upon change |
| SONAME | | | state "0" character abbreviation |
| SOVLO | volt | | state "0" low level voltage |
| SOVHI | volt | | state "0" high level voltage |
| S1NAME | | | state "1" character abbreviation |
| S1VLO | volt | | state "1" low level voltage |
| S1VHI | volt | | state "1" high level voltage |
| S2NAME | | | state "2" character abbreviation |
| S2VLO | volt | | state "2" low level voltage |
| S2VHI | volt | | state "2" high level voltage |
| S19NAME | | | state "19" character abbreviation |
| S19VLO | volt | | state "19" low level voltage |
| S19VHI | volt | | state "19" high level voltage |
| TIMESTEP | sec | 1E-9 | digital input file step-size |

## Two Bit Adder with Digital I/O

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. SPICE outputs a digital file.
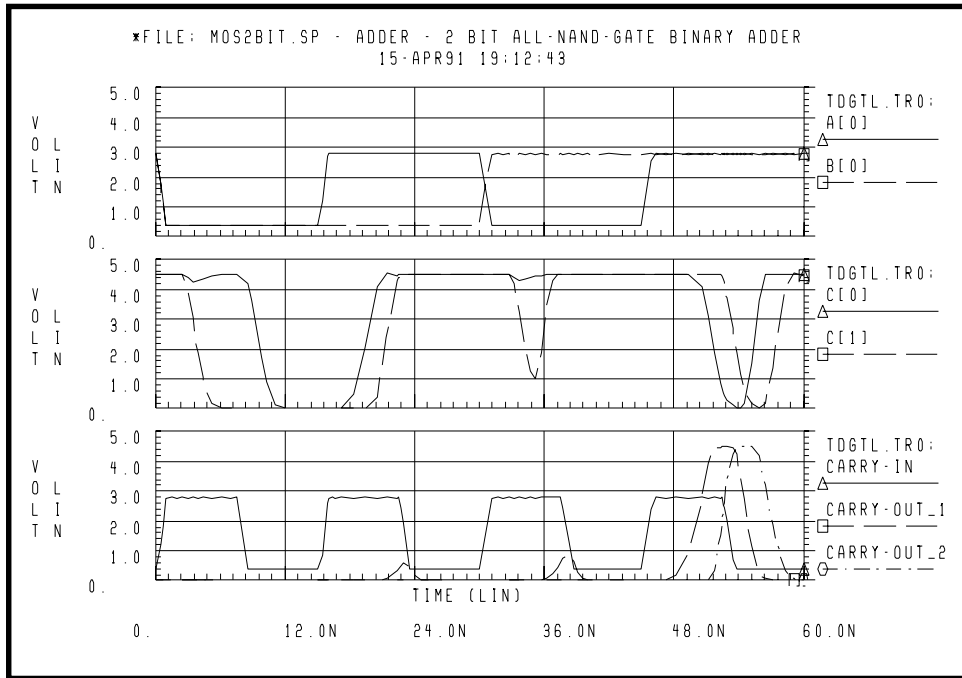
**Figure 22-7: Digital Stimulus File Input**

## Replacing Sources With Digital Inputs



Traditional voltage pulse sources become ...

```
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
```

... D2A drivers that get their input from ...

```
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
```

... 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
```

**Figure 22-8: Digital File Signal Correspondence**

## Example of MOS 2 Bit Adder

```
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
```

---

# 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 of AND/NAND Gates

```
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
```
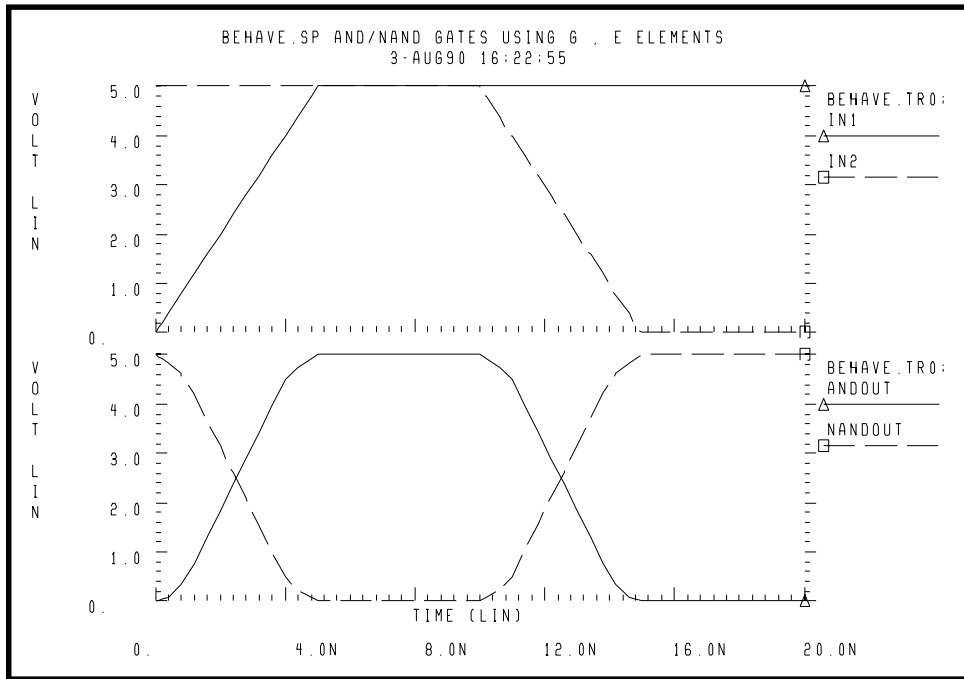
**Figure 22-9: NAND/AND Gates**

## Behavioral D-Latch

In this example, a D flip-flop is modeled by one input NAND gates and NPWL/ PPWL functions.
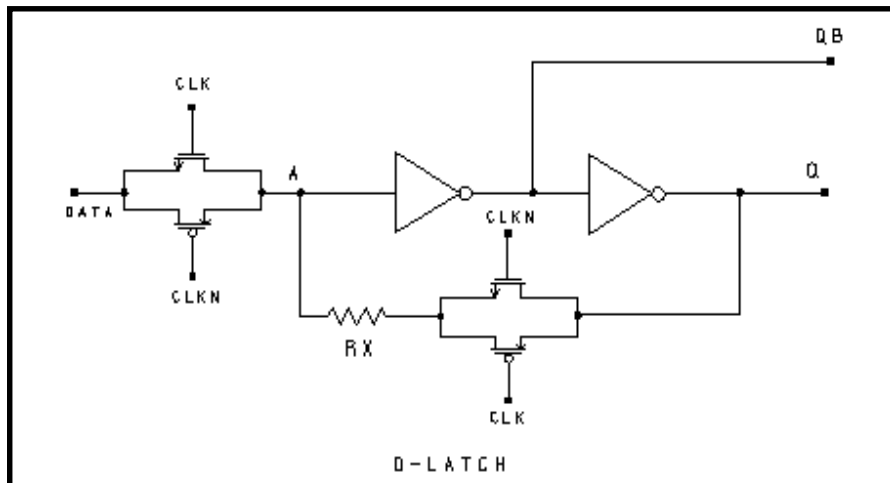


**Figure 22-10: D-Latch Circuit**

### Example of a D-Latch

```
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
```
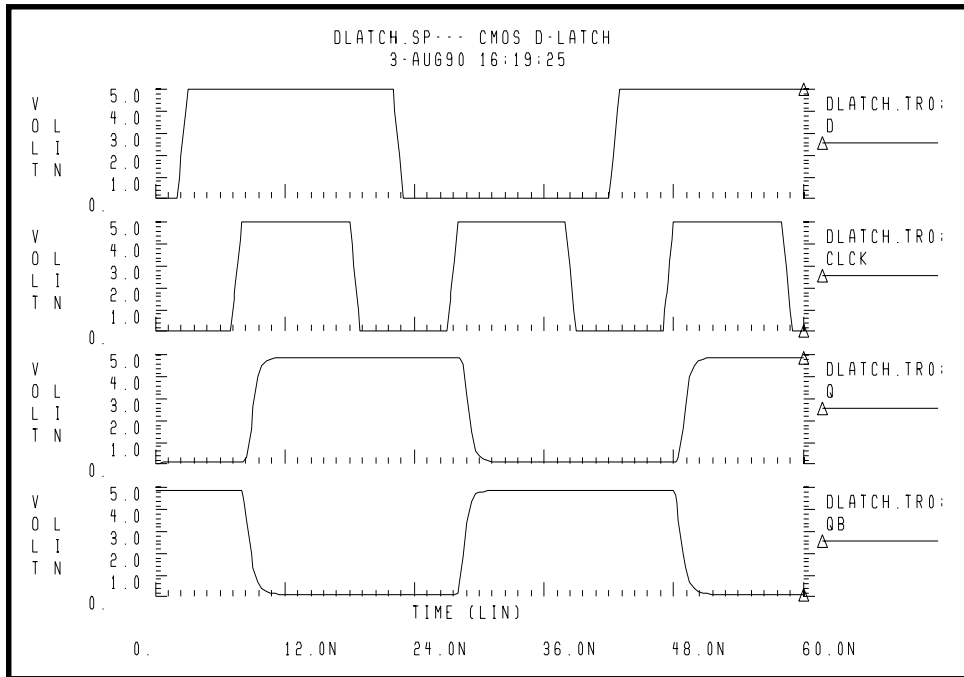


**Figure 22-11: D-Latch Response**

# 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.
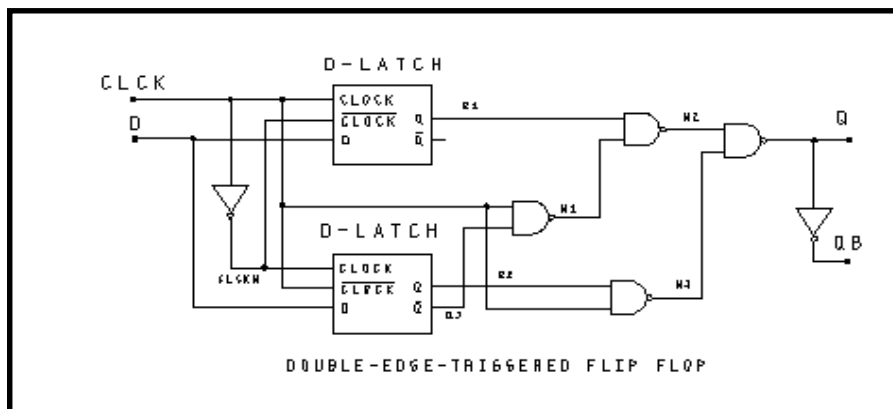


**Figure 22-12: Double Edge Triggered Flip-Flop Schematic**

## Example of a Double Edge Triggered Flip-Flop

```
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
```
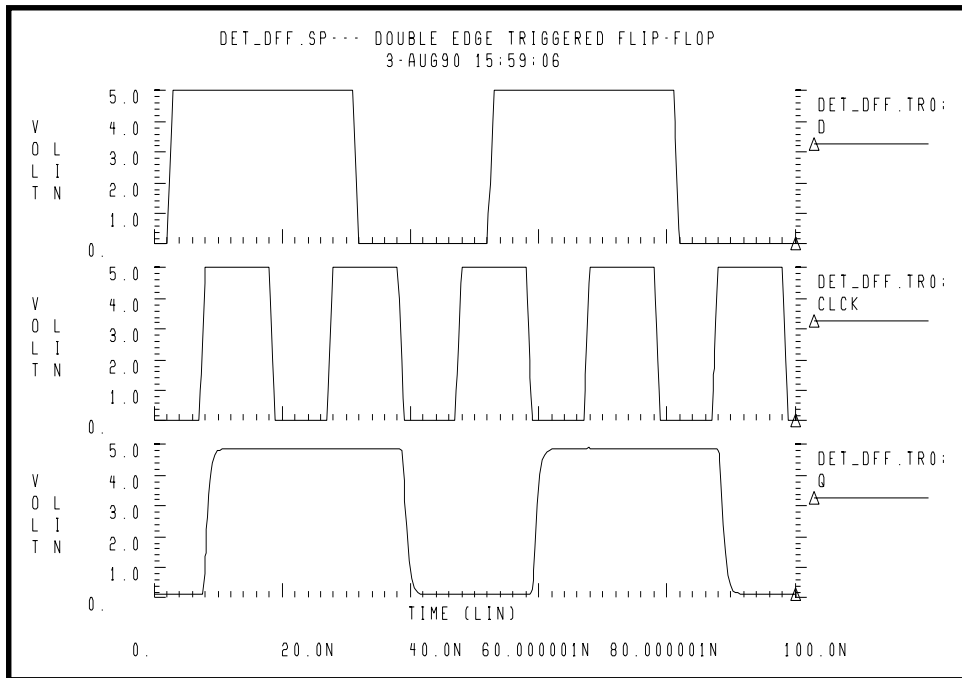
**Figure 22-13: Double Edge Triggered Flip-Flop Response**

# 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.
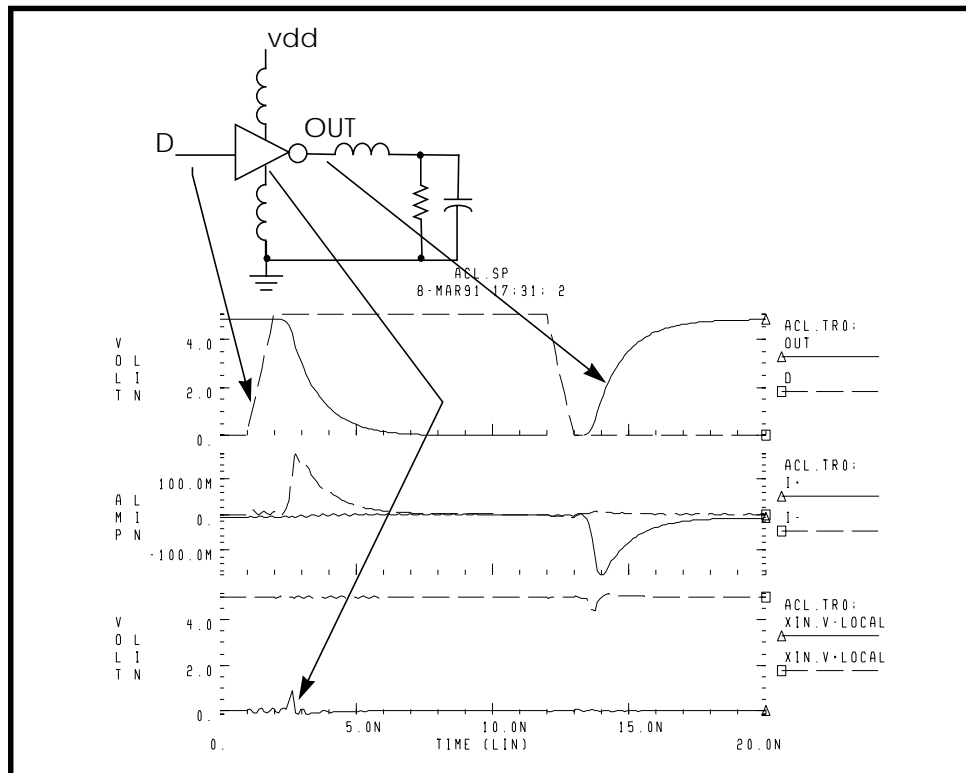


**Figure 22-14: ACL Family Output Buffer**

Star-Star-HspiceStar-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.
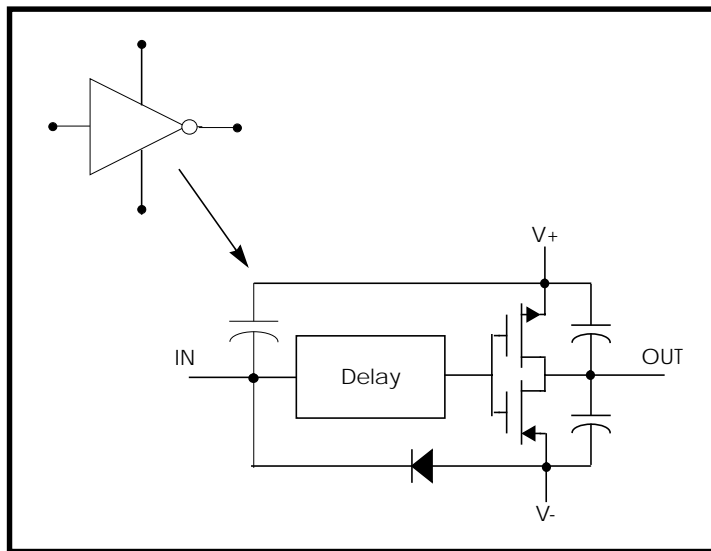


**Figure 22-15: Inverter**

## 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:
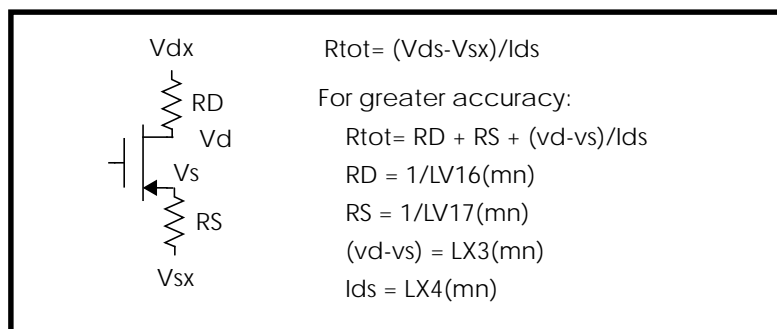


**Figure 22-16: VIN versus VOUT**

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.
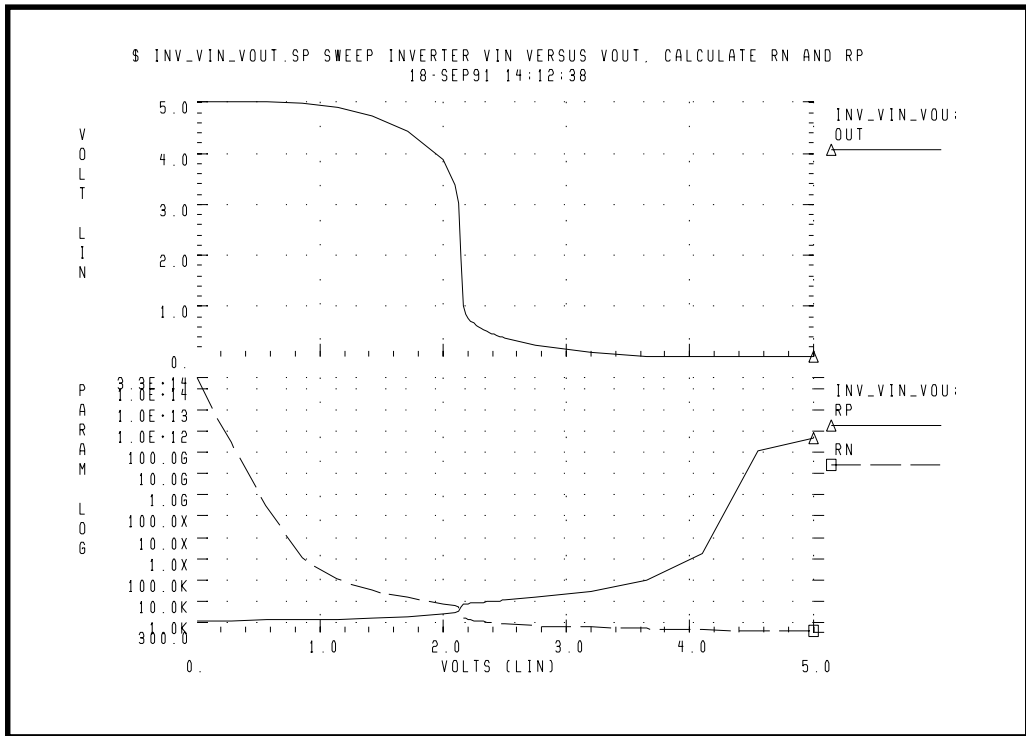
**Figure 22-17: RP and RN as a Function of VIN**

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 22-18: 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 CMOS Inverter Subcircuit

```
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 22-19: 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 22-20: 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         lookup table H element source
  rectifier

- 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 22-21: Integrator Circuit**

## Example of Integrator Circuit

```
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 22-22: 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 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 22-23: Differentiator Circuit**

## Example of a Differentiator Circuit

```
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
```

```
          DIFF.SP  DIFFERENTIATOR CIRCUIT
              19-MAR91 15:45:42
```

**Figure 22-24: 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 22-25: 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 of Tunnel Diode

```
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 22-26: Tunnel Diode Characteristic**

## Behavioral Silicon Controlled Rectifier

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 of Silicon Controlled Rectifier (SCR)

```
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 22-27: 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 Model

```
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 22-28: Triode Family of Curves**

## Behavioral Amplitude Modulator

This example uses a G element as an amplitude modulator with pulse waveform carrier.

## Example of Amplitude Modulator

```
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 22-29: Amplitude Modulator Waveforms**

# Behavioral Data Sampler

A behavioral data sample follows.

## Example Behavioral Sampling with E Element

```
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 22-30: Sampled Data**

# Using Op-Amps, Comparators, and Oscillators

Thsi 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)

General form
```
 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.<br>COMP=0 internal compensation (Default)<br>COMP=1 external compensation |
| DEF | | | default model selector. Allows choice of three default settings.<br>0= generic (0.6 MHz bandwidth) (Default)<br>1= ua741 (1.2 MHz bandwidth)<br>2= mc4560 (3 MHz bandwidth) |

| Names(Alias) | Units | Default | Description |
|---|---|---|---|
| DELPHS | degrees | | 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*<br><br>(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.<br>If AV1K is greater than zero, the unity gain frequency is calculated from AV1K and FREQ is ignored. |
| IB | amp | | input bias current. The amount of current required to bias the input differential transistors. This is generally a fundamental electrical characteristic. Typical values are between 20 and 400 nA. |
| *IBOS* | amp | | input bias offset current, also called input offset current. This is the amount of unbalanced current between the input differential transistors. Generally a fundamental electrical characteristic. Typical values are 10% to 20% of the IB. |

| Names(Alias) | Units | Default | Description |
|---|---|---|---|
| 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. |
| 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. |

| Names(Alias) | Units | Default | Description |
|---|---|---|---|
| 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. |
| 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. |

| Names(Alias) | Units | Default | Description |
|---|---|---|---|
| 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 |

| Parameter | Description | Defaults | | |
|-----------|-------------|----------|----------|----------|
| | | DEF=0 | DEF=1 | DEF=2 |
| CMRR | common mode rejection ratio | 96 db | 106 db | 90 db |
| | | 63.1k | 199.5k | 31.63k |
| COMP | compensation level selector | 0 | 0 | 0 |
| 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 |

| Parameter | Description | Defaults | | |
| --- | --- | --- | --- | --- |
| | | DEF=0 | DEF=1 | DEF=2 |
| 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 G$\Omega$ feedback resistor for the AC analysis in place of the 10 k$\Omega$ 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 15v. 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 $\mu$s (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 Amplifier Model

```
$$ 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 22-31: Op-Amp Circuit**

## Example Op-Amp

```
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 22-32: AC Analysis Response**

**Figure 22-33: Transient Analysis Response[1.]**

## Inverting Comparator with Hysteresis

An inverting comparator is modelled by a piecewise linear VCVS.



**Figure 22-34: Inverting Comparator with Hysteresis**

Two reference voltages corresponding to volow and vohigh of Ecomp characteristic are:

$$Vreflow \; = \; \frac{Volow \cdot Rb}{Rb + Rf}$$
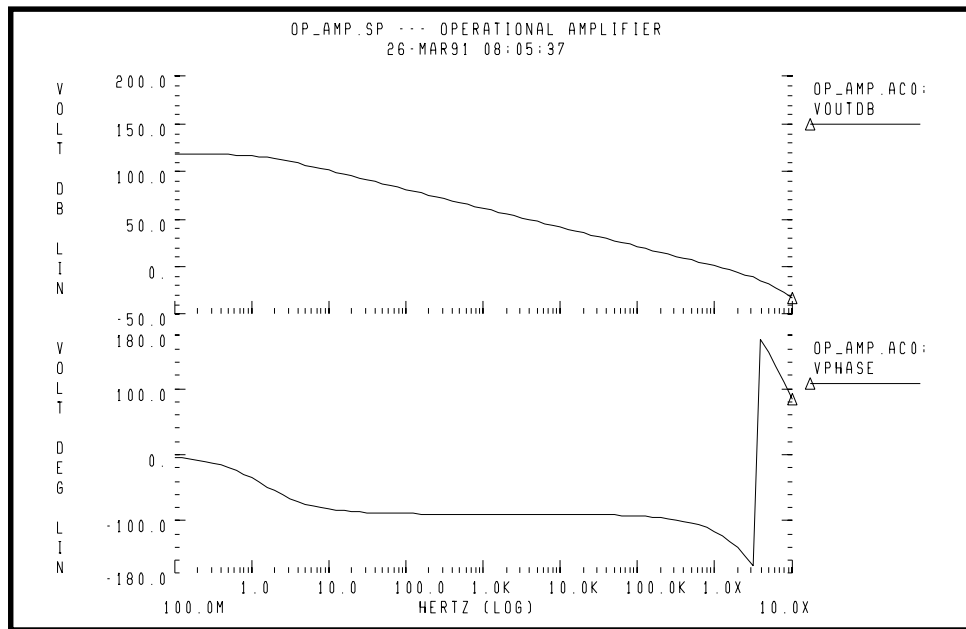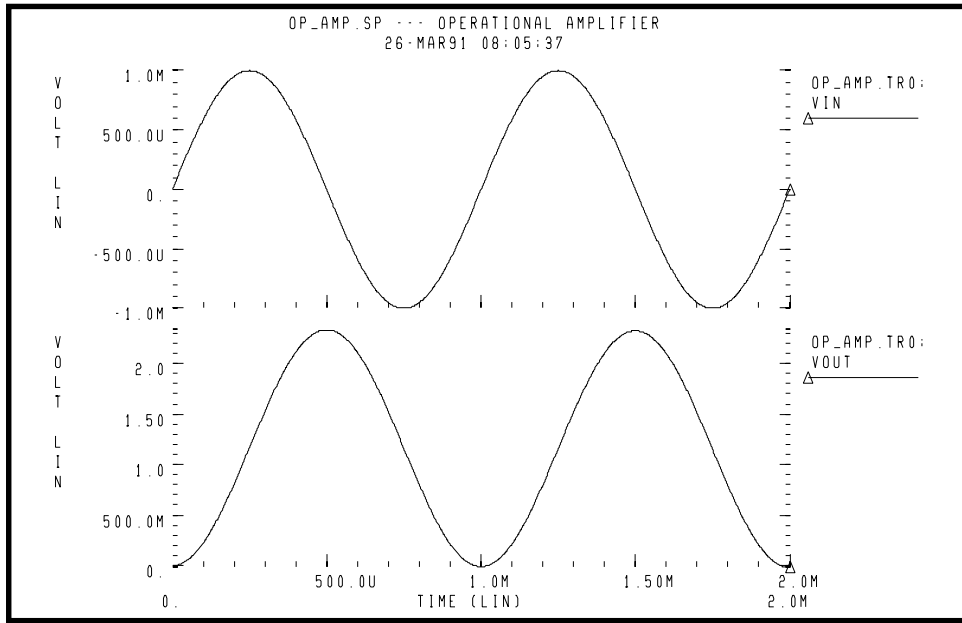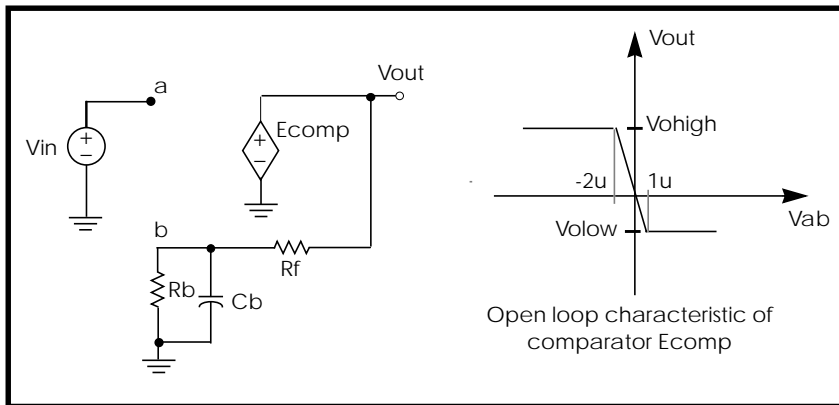
$$Vrefhigh \; = \; \frac{Vohigh \cdot Rb}{Rb + Rf}$$

When Vin exceeds Vrefhigh, the output Vout goes to Volow. For Vin less than Vreflow, the output goes to Vohigh.

## Example Inverting Comparator with Hysteresis

```
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
```
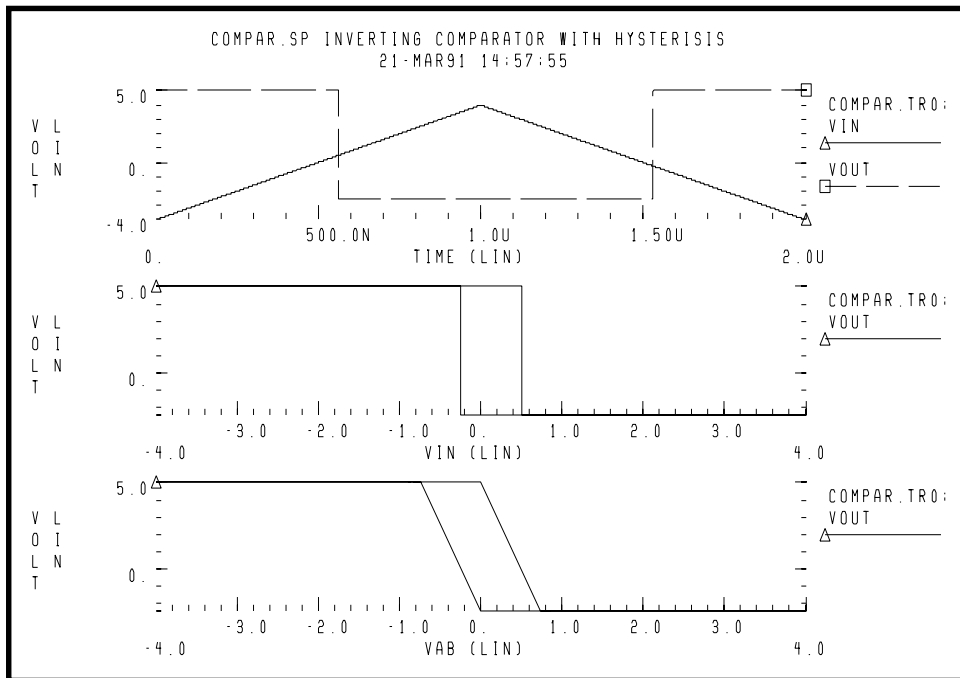
**Figure 22-35: Response of Comparator**

## Voltage Controlled Oscillator

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 Voltage Controlled Oscillator

```
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
```
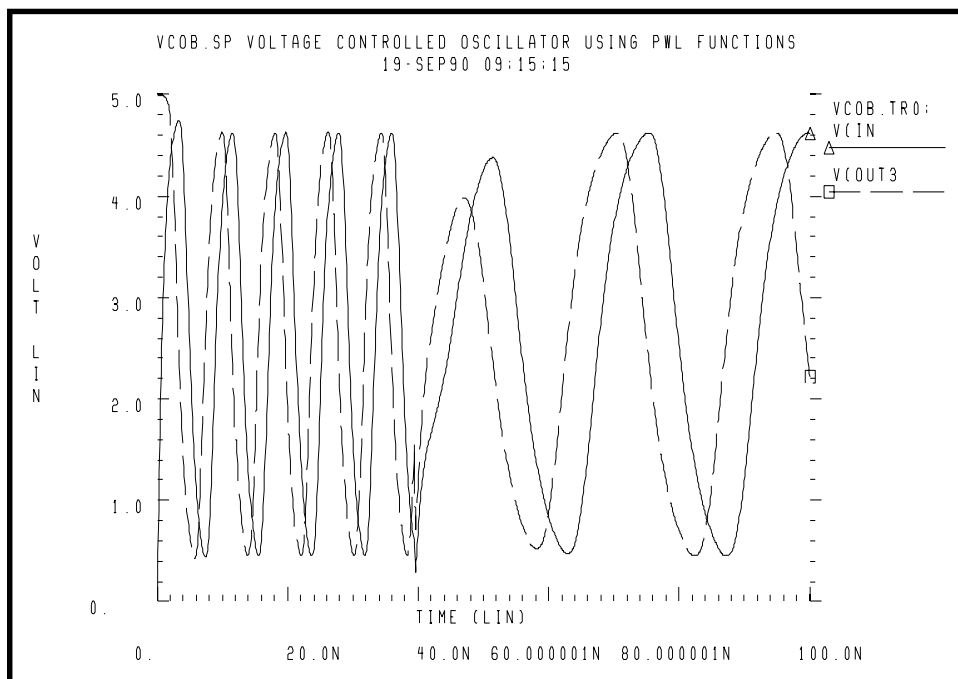
**Figure 22-36: Voltage Controlled Oscillator Response**

## 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 Voltage Variable Capacitance

```
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
```
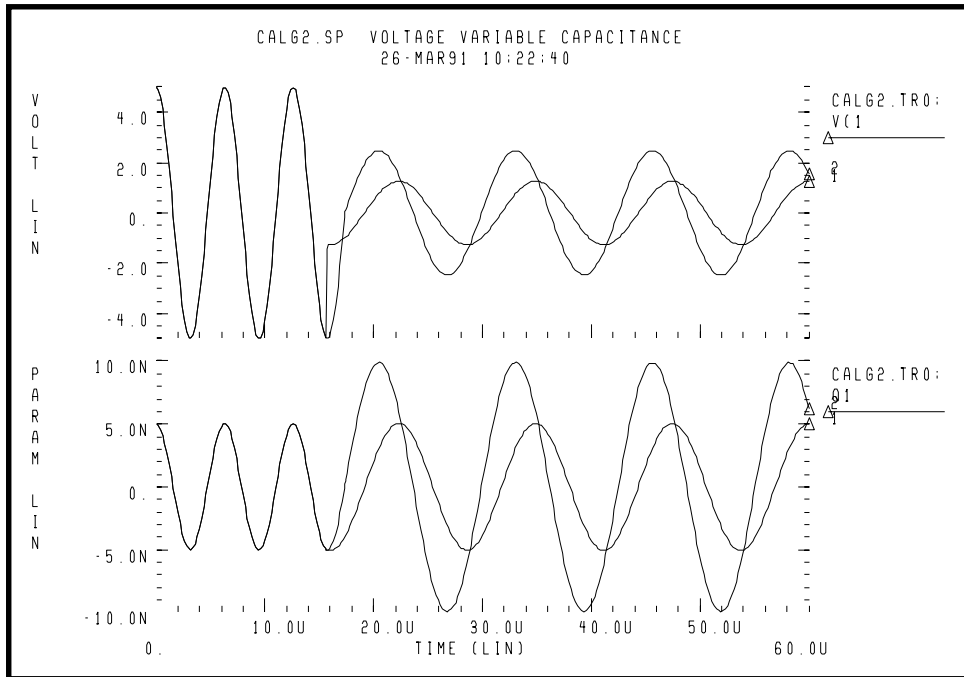
**Figure 22-37: Correct Result Corresponding to CTYPE=1**
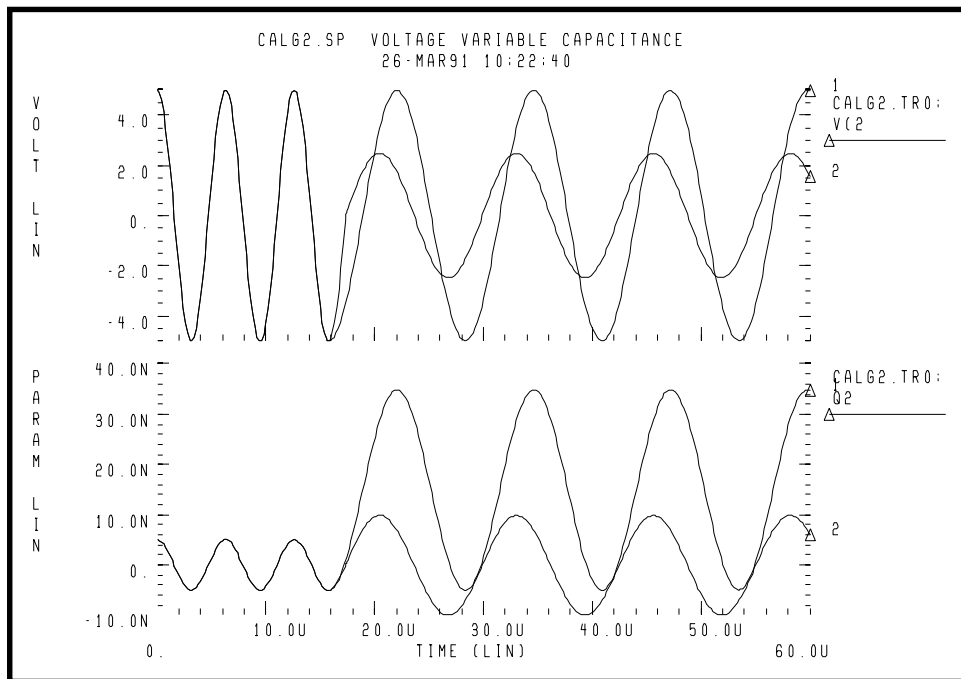
**Figure 22-38: Incorrect Result Corresponding to CTYPE=0**

# 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.
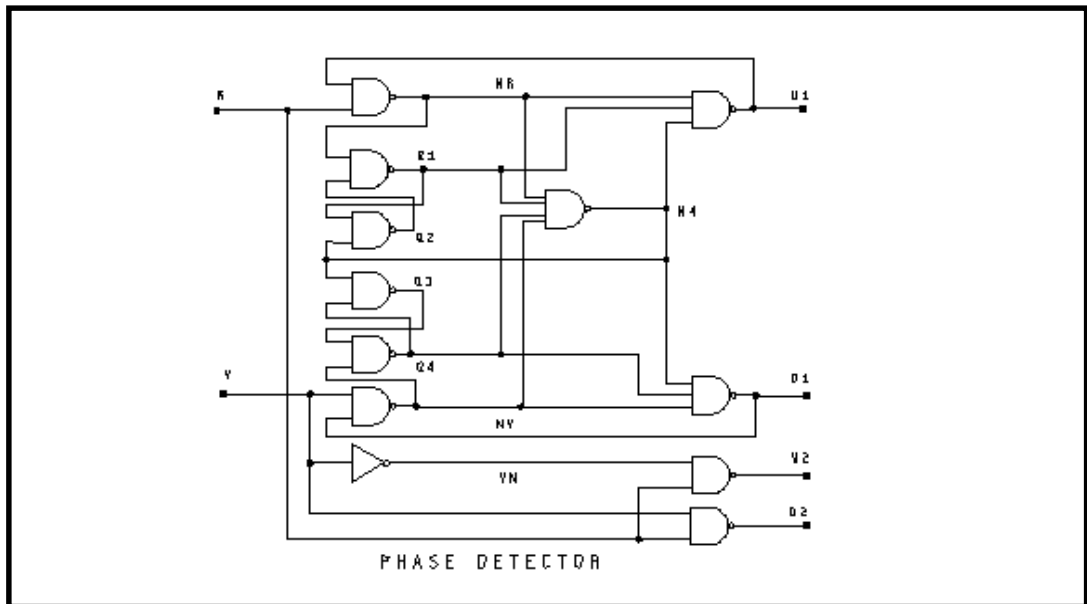


**Figure 22-39: Circuit Schematic of Phase Detector**

## Example Phase Detector with Behavioral NAND Gates

```
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
```
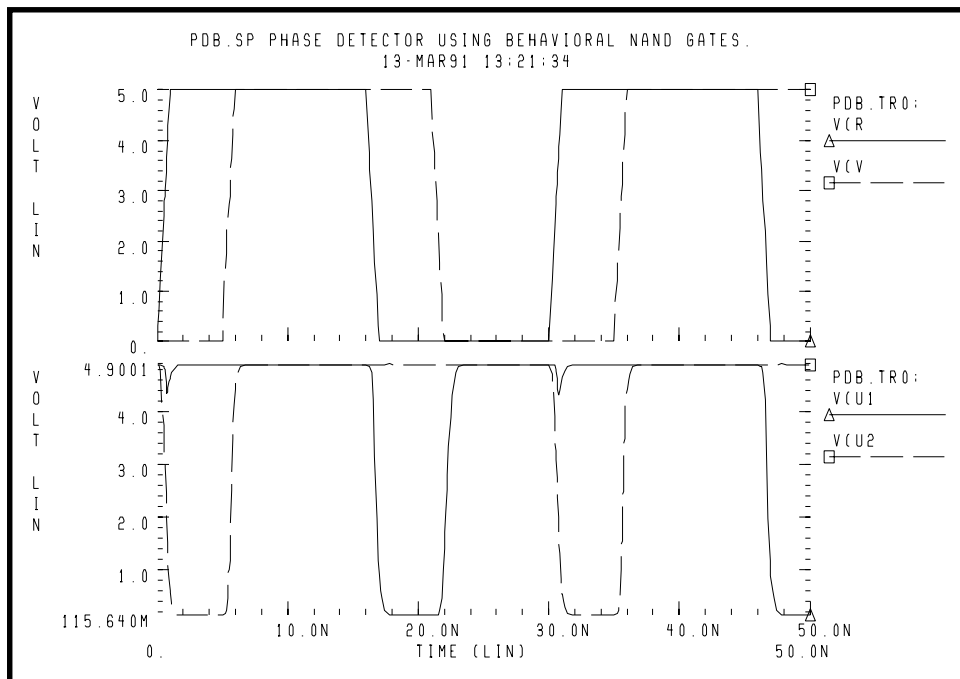
**Figure 22-40: Phase Detector Response**

## PLL BJT Behavioral Modeling
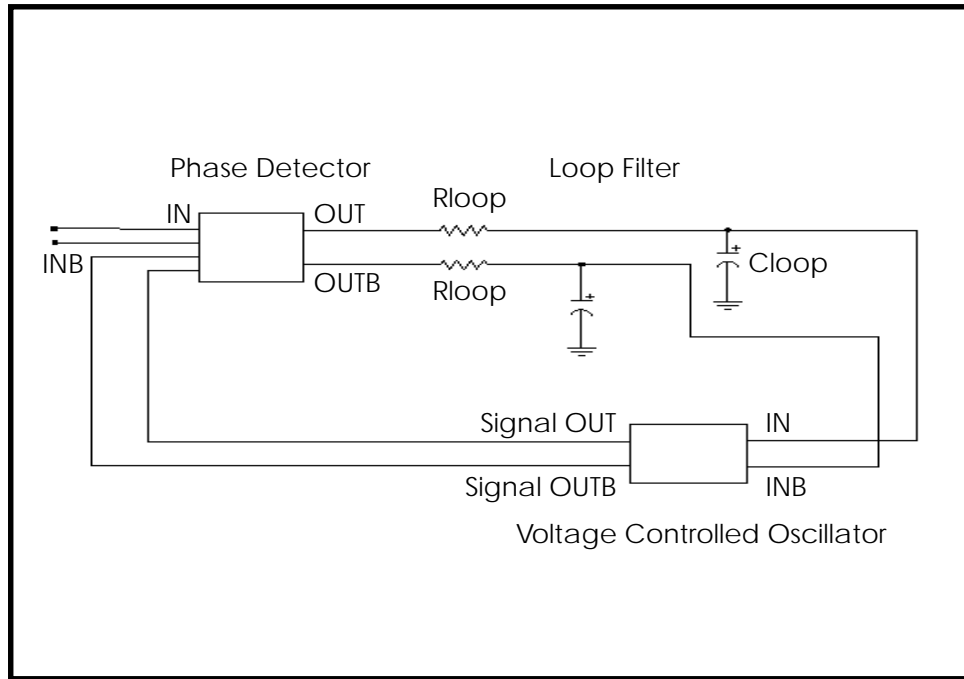


**Figure 22-41: PLL Schematic**

### Example Phase Locked Loop

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 an 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 22-42:) 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.
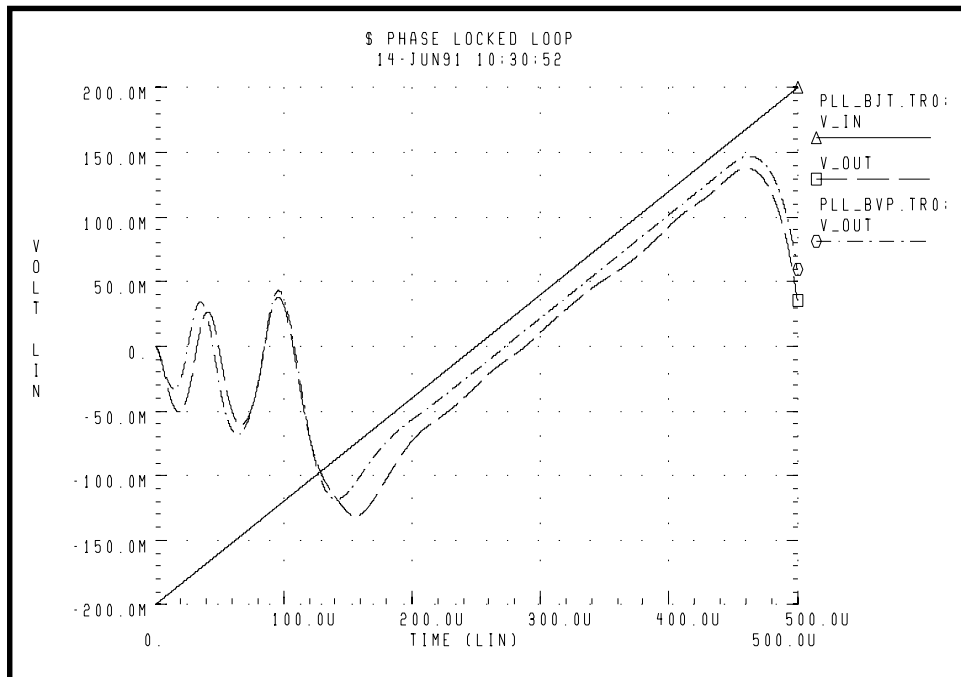


**Figure 22-42: Behavioral (PLL_BVP Curve) vs. Bipolar (PLL_BJT Curve) Circuit Simulation**

## Example 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
```

## Example 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
```
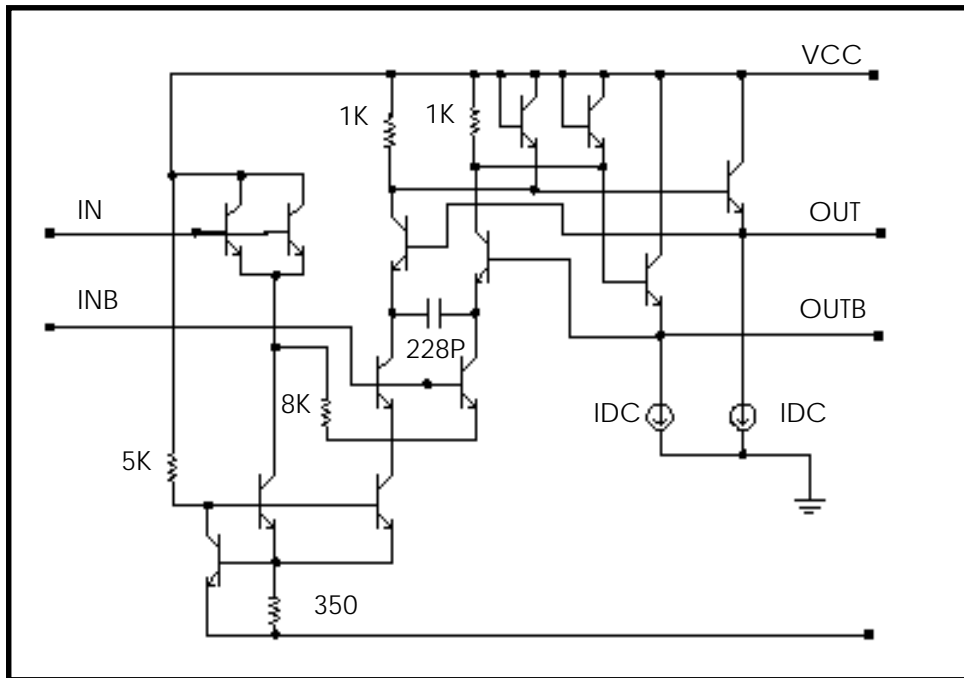


**Figure 22-43: Voltage Controlled Oscillator Circuit**

## BJT Level Phase Detector

### Example Phase Detector

```
$ 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 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
```
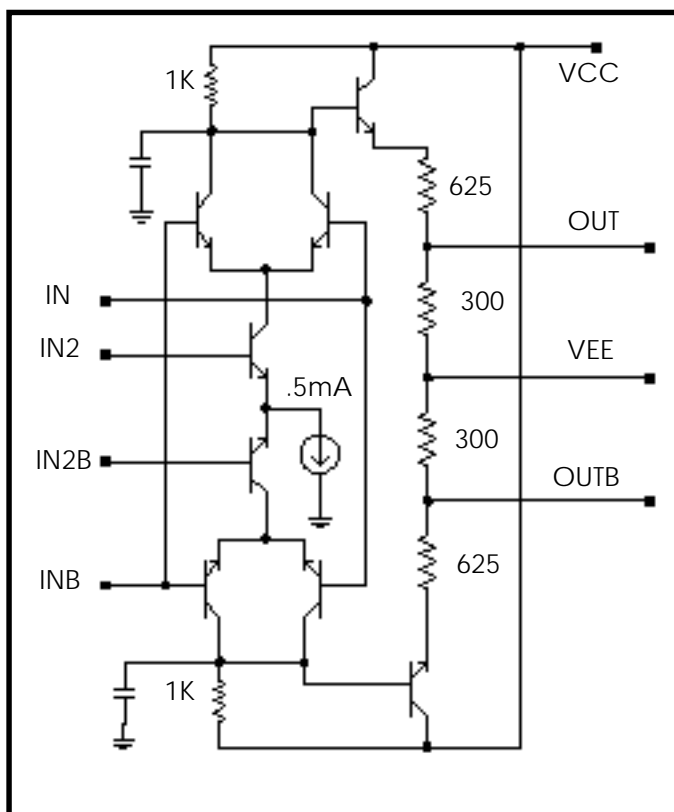


**Figure 22-44: Phase Detector Circuit**

# 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!*

# Chapter 23

# Using Meta I/O

Meta I/O converts I/O buffer models between Star-Hspice format and the I/O Buffer Information Specification (IBIS) Version 2.1 format. Meta I/O is not part of Star-Hspice, but is an optional product that operates with the following software:

- Star-Hspice, Versions 95.2 and higher
- Meta Encryptor, Version 1.0
- SPICE to IBIS, Version 1.0, from the University of North Carolina

This chapter explains how to use Meta I/O, gives an example of a Meta I/O application, and describes Meta I/O commands. The following topics are covered in this chapter:

- Understanding the Design Flow
- Getting Started with IBIS
- Using the Meta I/O Commands

# Understanding the Design Flow

Meta I/O Version features include the following:

- Intellectual property protection with IBIS buffer description format
- Library management for improved productivity
- Automatic library conversion and Star-Hspice verification
- Optional Meta Encryption for models other than I/O buffers

You must have the full Star-Hspice product set on your system for Meta I/O to run successfully.

Figure 23-1: shows the design data flow for the IC vendor. The process starts with Star-Hspice or SPICE 2g6 transistor level buffer models. These models are converted by the "SPICE to IBIS" program into IBIS description files. The IBIS description files can then be converted into Star-Hspice behavioral buffer models and verified against the original transistor buffers. Optionally, any Star-Hspice subcircuit can be encrypted with the Meta Encryptor, which is a separate Avant! product.

Figure 23-2: shows the design data flow for the system designer. The process starts with Star-Hspice IBIS templates and IBIS description files from the IC vendor. These files are converted by the "IBIS to Star-Hspice" program into Star-Hspice behavioral level models.
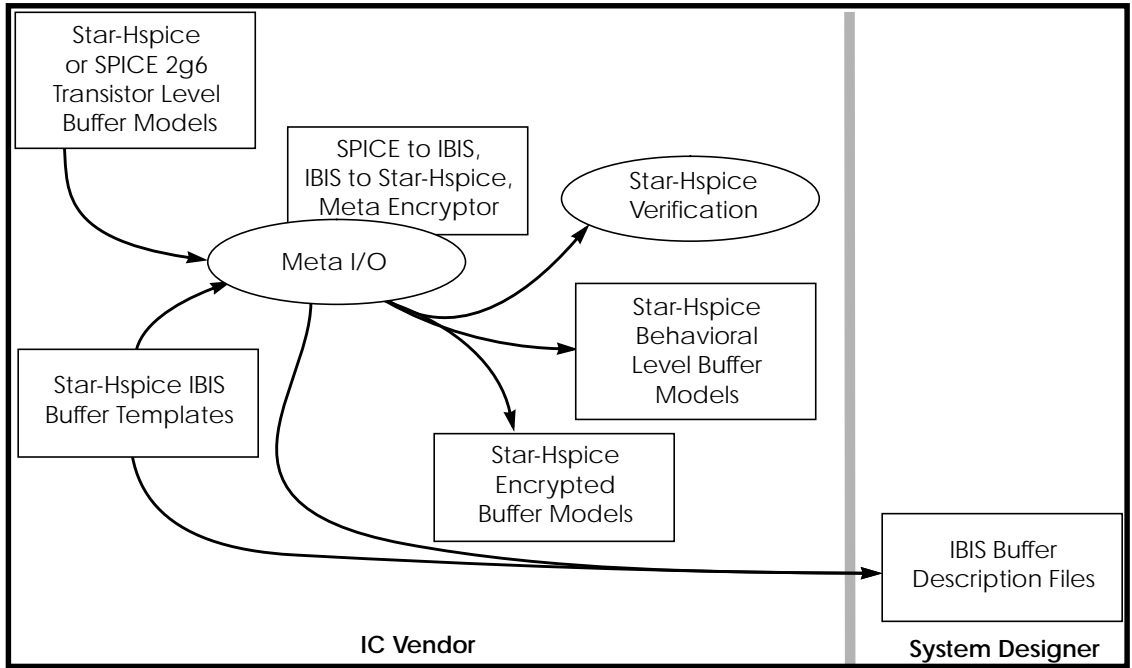
**Figure 23-1: IC Vendor Design Data Flow Using Meta I/O**
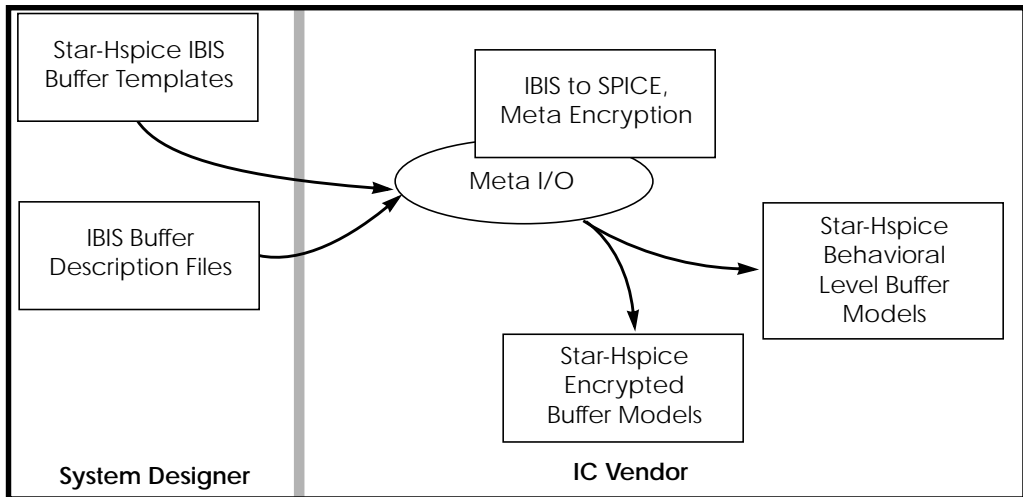


**Figure 23-2: – System Design Data Flow Using Meta I/O**

# Getting Started with IBIS

The example in this section introduces the basic concepts of the Meta I/O product. This example shows how you can use Meta I/O to:

- Select and convert an IBIS description to a set of Star-Hspice behavioral buffers
- Verify the final results with Star-Hspice

The example design is a simple three-buffer component named IbisTest.

Figure 23-3: illustrates the library structure for the IbisTest example. *IbisLib*, *Templates*, *Star-HspiceLib*, and *VerDir* are directories used in the example.
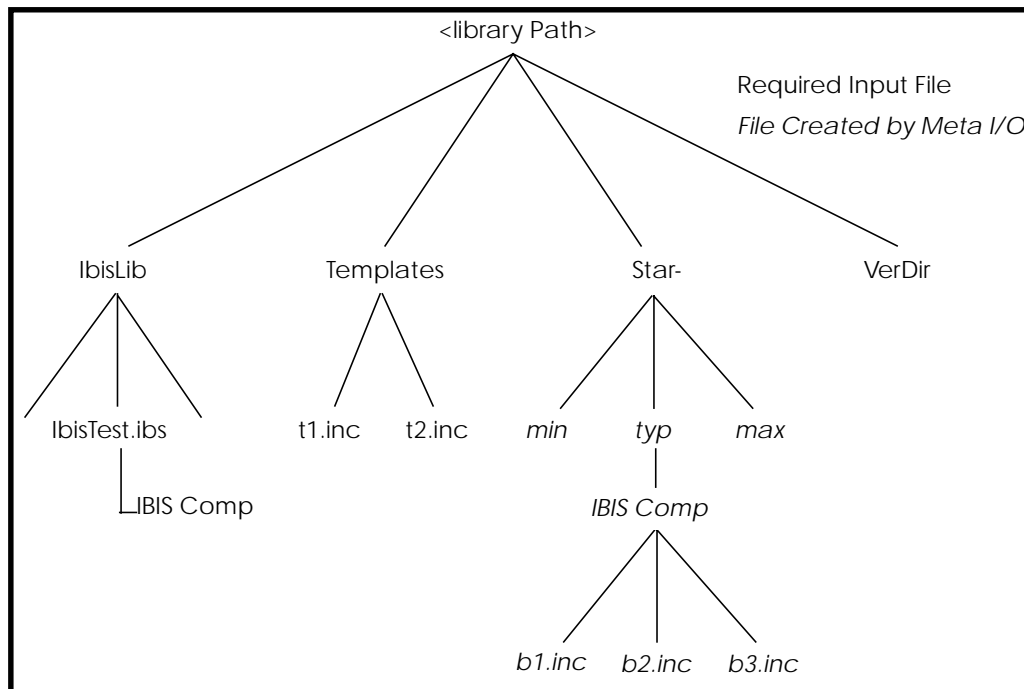


**Figure 23-3: – IbisTest Example Library Structure**

## The Meta I/O Script

Meta I/O derives its control information from a Meta I/O script file that contains all commands required to convert and verify an IBIS-based component library. The script is a sequence of Meta I/O commands in an ASCII text file that you can create using a text editor. The script for this example is presented later in this chapter. All of the Meta I/O commands also are described later in this chapter.

## Conversion Function Setup

Set the conversion function to "IBIS to Star-Hspice". Use the SetConvFunc command with the i2h keyword:

```
SetConvFunc      i2h;
```

Select the desired data from the IBIS description file. Use the SetConvType command with the min keyword:

```
SetConvType      typ;
```

## Library Directory Setup

Use the *SetIbisDescripDir* command to select the source of the IBIS description file. This command accepts a string parameter that defines the directory containing the IBIS description file for the IBIS to Star-Hspice process.

Use the *SetIbisTemplateDir* command to select the source directory for the behavioral level buffer model template files. This command accepts a string parameter that defines the directory containing the template files for the IBIS to Star-Hspice process.

Use the *SetStar-HspiceDir* command to select the destination for the behavioral level buffer model files. This command accepts a string parameter that defines the directory to hold the files created during the IBIS to Star-Hspice process.

Use the *SetVerDir* command to select the location for running the Star-Hspice verification process. This command accepts a string parameter that defines the directory to hold the Star-Hspice intermediate files during verification.

The section of the IbisTest script that sets up directories is:

```
  SetIbisDescripDir./lib/IbisLib;
SetIbisTemplateDir   ./lib/Templates;
SetStar-HspiceDir    ./lib/Star-HspiceLib;
SetVerDir            ./lib/Verify;
```

## Template Selection

Meta I/O provides a method of connecting a given IBIS buffer model to an executable Star-Hspice behavioral template with the *SetDefTemplate* and *TemplateMap* template selection commands.

Use the *SetDefTemplate* command to define the filename for all IBIS buffer model types that do not have a special template model. In this example, Meta I/O expects to find the d2a.inc template file in the directory specified by the value of the SetIbisTemplateDir function.

Use the TemplateMap command to specify the filename for specific IBIS buffer model types that require a special template model. In this example, Meta I/O expects to find the *d2a.inc* template file in the directory specified by the value of the *SetIbisTemplateDir* function and uses that template file to convert all I/O_open_drain type models.

```
  SetDefTemplate d2a.inc;
  TemplateMap I/O_open_drain  d3.inc;
```

For further information on *SetDefTemplate* and Requirements, please see the to"Using the Meta I/O Commands" on page 23-10.

## File Selection In Meta I/O

Use the *SetFileMask*, *SetInFileSuffix*, *SetOutFileSuffix, and SelectFiles*
commands to select the single SPICE transistor level model:

```
SetFileMask        IbisTest;
SetInFileSuffix    .ibs;
SetOutFileSuffix   .inc;
SelectFiles;
```

Use the *SetFileMask* command to describe which files to convert. Meta I/O
allows the use of file wildcards in filename, to select multiple files for
conversion. In this example, no wildcards are used.

Use the *SetInFileSuffix* command to select only files that end in .ibs.

Use the *SetOutFileSuffix* command to produce output files that have the suffix
.inc.

Use the *SelectFiles* command to iterate through the library directory and place
each file that matches the select criteria in its internal selected files list.

The selection process is additive, which allows you to set a selection criteria,
select files, change criteria, and append new files to the existing list.

Use the *UnselectFiles* command to clear the file selection list.

## Conversion

The command Iterate causes Meta I/O to process each file in the selection list
according to the conversion function specified earlier. As each file is converted,
warning and error messages are echoed to the screen (through stdout) providing
a record of the actions taken.

You can select a set of files and then issue many different conversion or
verification functions on that list without reselecting the files for each operation.

## IbisTest Script Listing

```
# IbisTest Conversion Example
#
SetConvFunc          i2h;
SetConvType          typ;
SetIbisDescripDir    ./lib/IbisLib;
SetIbisTemplateDir   ./lib/Templates;
SetStar-HspiceDir    ./lib/Star-HspiceLib;
SetVerDir            ./lib/Verify;
SetDefTemplate       d2a.inc;
SetFileMask          IbisTest;
SetInFileSuffix      .ibs;
SetOutFileSuffix     .inc;
SelectFiles;
Iterate;
UnselectFiles;
```

## Running the Example

Enter the following command to run the IbisTest example:

```
metaio -f IbisTest
```

To capture the results in a log file, use the following command:

```
metaio -f IbisTest > IbisTest.log
```

## The Results

The following is a log file that is produced by the example run:

```
0 : #                   Meta I/O
0 : #
0 : #    Copyright 1998 Avant!, Inc.
0 : #            All Rights Reserved
0 : #
1 : #
2 : # IbisTest
3 : #
4 :
5 : SetConvFunc             i2h;
6 : SetConvType             typ;
```

```
 7 :
 8 : SetIbisDescripDir        ./lib/IbisLib;
 9 : SetIbisTemplateDir       ./lib/Templates;
10 : SetStar-HspiceDir            ./lib/Star-Hspice;
11 :
12 : SetDefTemplate               d2a.inc;
13 : Requirement            I/O  d3a.inc;
14 :
15 : SetFileMask            *;
16 : SetInFileSuffix        .ibs;
17 : SetOutFileSuffix       .inc;
18 :
19 : SelectFiles;
20 : Iterate;
   : i2hConvert = ./lib/IbisLib/ibistest.ibs

Checking ibistest.ibs for IBIS 2.1 <Beta> Compatibility...

WARNING (line  108) - Pulldown Maximum data is non-monotonic
WARNING (line  109) - Pulldown Typical data is non-monotonic
WARNING (line  109) - Pulldown Minimum data is non-monotonic
WARNING (line  152) - Pullup Maximum data is non-monotonic
WARNING (line  155) - Pullup Typical data is non-monotonic
WARNING (line  155) - Pullup Minimum data is non-monotonic
WARNING (line  279) - Pulldown Maximum data is non-monotonic
WARNING (line  280) - Pulldown Typical data is non-monotonic
WARNING (line  281) - Pulldown Minimum data is non-monotonic

IBIS Structure Okay

*** Warning : Line  20 : Can't get IBIS Package Models for
some reason!

Errors  : 0
Warnings: 9

File Passed

21 : UnSelectFiles;
22 :
23 :
```

# Using the Meta I/O Commands

This section is a Meta I/O command reference. All commands, their arguments, and syntax are described.

## General Form and Conventions

The general form of a Meta I/O command is:

```
<CommandName> [<Arg1>[<Arg2> ...]];
```

Every command must be terminated by a semicolon (;).

Command names are case insensitive – they can be entered in any combination of upper- and lowercase letters. Arguments are also case insensitive, except for file names and directory names.

You can enter comments by preceding the comment text by a pound sign (#). A comment can start at the beginning of a line or following the semicolon, as in the following examples:

```
#
# This is a comment
#
SetIbisDescripDir ./lib/Ibis; # So is this
```

### Wild Card Characters

Meta I/O supports the use of wild cards for file selection. The allowable wild cards are the asterisk and the character range.

#### The Asterisk

Use the asterisk (*) to match zero or more of any characters.

The wild card

```
buf*
```

matches

```
buf
```

```
buff
buffer
buf1
```

### The Range

Use the Range wild card to match a single ASCII character . The range wild card is delimited by the square brackets and the hyphen character.

The syntax for the range wild card is:

```
'[' <Char> '-' <Char> ']'
```

A range wild card

```
buf[0-2]
```

matches

```
buf0
buf1
buf2
```

but does not match

```
buff
buf6
```

## Directory Management Commands

### SetIbisDescripDir <Path>;

Sets the directory that Meta I/O refers to when accessing IBIS description files for conversion.

The <Path> parameter is a string.

### GetIbisDescripDir;

Returns a string describing the directory where Meta I/O accesses IBIS description files.

**SetIbisTemplateDir <Path>;**

Sets the directory that Meta I/O refers to when accessing IBIS template files for use in conversion.

The <Path> parameter is a string.

**GetIbisTemplateDir;**

Returns a string describing the directory where Meta I/O accesses IBIS template files.

**SetStar-HspiceDir <Path>;**

Sets the directory that Meta I/O refers to when producing Star-Hspice results files during conversion.

The <Path> parameter is a string.

The actual path is suffixed with the current value of the function *GetConvType*.

**GetStar-HspiceDir;**

Returns a string describing the directory where Meta I/O writes Star-Hspice files during conversion.

The returned path does not include the current value of the function *GetConvType*.

**SetVerDir <Path>;**

Sets the directory that Meta I/O refers to when running Star-Hspice verifications after conversion.

The <Path> parameter is a string.

**GetVerDir;**

Returns a string describing the directory where Meta I/O stores temporary files.

**SetTmpDir <Path>;**

Sets the directory that Meta I/O refers to when running Star-Hspice verifications.

The <Path> parameter is a string.

**GetTmpDir;**

Returns a string that describing the directory where Meta I/O stores temporary files.

## File Management Commands

**SetFileMask <WildCard>;**

Sets the file selection mask (a regular expression) for files in the current directory that are subject to the current conversion function.

The parameter <WildCard> for example, might be a string of the form "74ls*", which matches all files starting with the literal "74ls" that contain zero or more characters.

**GetFileMask;**

Returns a string representing the current value of file selection mask.

**SetInFileSuffix <WildCard>;**

Sets the input file suffix mask regular expression thatselects files in the current input directory subject to the current conversion function.

The parameter <WildCard>, for example, might be a string of the form ".ibs", which would match all files with the literal ".ibs" suffix that contain one or more characters up to the literal ".".

**GetInFileSuffix;**

Returns a string that represents the current value of the input file suffix mask.

**SetOutFileSuffix <String>;**

Sets the output file suffix mask (a regular expression) used to create files in the current output directory subject to the current conversion function.

The parameter <String>, for example, might be a string of the form ".inc", which matches all files with the literal ".inc" suffix that contain one or more characters up to the literal ".".

**GetOutFileSuffix;**

Returns a string that represents the current value of the output file suffix mask.

**SelectFiles;**

Selects the files in the current conversion directory based on the values of the:

- applicable directory
- SetFileMask
- SetInFileSuffix strings

The function returns the number of files in the selection.

### *Example*
```
SetIbisDescDir/lib/dir1;
SetFileMask 74ls*;
SetInFileSuffix.ibs;
SelectFiles;
```

with the following files in /lib/dir1:
```
7400.ibs
7402.ibs
74ls00.ibs
74ls02.ibs
74ls04.ibs
83333.ibs
83334.ibs
```

sets the current selection list to contain

```
74ls00.ibs
74ls02.ibs
74ls04.ibs
```

To append the new filenames to the selection list, change the wild cards and reissue a SelectFiles command.

### **UnSelectFiles;**

Clears the current selection list.

## Conversion Management Commands

### **SetConvFunction <ConvFunc>;**

This command sets the conversion function to one of the predefined conversions: Spice to IBIS, IBIS to Star-Hspice, Star-Hspice Encryption, and Verification.

**Table 23-1: – SetConvFunc Conversion Types**

| Conversion Function | Token |
|---|---|
| Spice to IBIS | s2i |
| IBIS to Star-Hspice | i2h |
| Encryption | Enc |
| Verification | Ver |

### **GetConvFunction;**

Returns a string that indicates the current conversion function.

**SetConvType <ConvType>;**

Use this command to select between the three corners of the IBIS description. These corners are defined as Minimum, Typical, and Maximum and represent different values (scalars and vectors) within the IBIS description.

Example:

```
...
SetConvType Min;
Iterate;
...
SetConvType Typ;
Iterate;
...
```

For each of the files that match the selection criteria, this example produces minimum and typical models.

In addition to the selection of the corner, this function should also provide the directory paths outlined in Figure 23-2:. The conversion type string is placed in the output directory path:

```
<OutDir>/<ConvType>/<Compname>/<Buffer><OutFileSuffix>
```

### Table 23-2: SetConvType Conversion Types

| Conversion Type | Token |
|-----------------|-------|
| All | All |
| Minimum | Min |
| Typical | Typ |
| Maximum | Max |

**GetConvType;**

Returns a string that indicates the current conversion type.

**Iterate;**

Successively applies the conversion function to each file in the selection list.

### Direct Iteration

The Direct Iteration feature allows code within the compiled body of the Meta I/O API to be directly executed. Some functions that exist only within Meta I/O and not as stand-alone programs require successive application to the input files.

This method is primarily used for the IBIS to Star-Hspice behavioral conversion. Because the IBIS Golden Parser is built into Meta I/O directly, an iteration through this conversion function causes Meta I/O to read and parse the IBIS Description, map the relevant pieces into the Star-Hspice behavioral template, and emit the final model into the specified directory tree.

### Process Fork

This is the standard process fork found in most operating systems. The primary method is through the system() call.

The appropriate parameters are passed with each call to system().

Execution of the Spice to IBIS and encryption functions is handled by this method.

## Verification Management Commands

To ensure accurate conversion of transistor level to behavioral level buffers, Meta I/O provides an automated verification method which casts each type of circuit into a user-defined test harness before Star-Hspice execution.

The test harness provides identical stimuli to each buffer and measures the final outcome. The results can be viewed with MetaWaves or by other user validation means.

The test harnesses are stored in the same directory as the Star-Hspice transistor level modes and have a different suffix to distinguish them from the buffer models.

Name matching is used to connect the test harness with the components under test.

## System Management Commands

### ExitMetaIoApi <ReturnValue>;

Cleans up and returns the integer <ReturnValue> to calling environment.

*Note:   ExitMetaIoApi is not implemented in revision 1.0.*

```
* A simple test harness

.INCLUDE 'Voltages'
.INCLUDE 'Stimili'
.INCLUDE 'Temp'
.OPTIONS Search= '/lib/Star-HspiceLib/typ/comp'

* IBIS Component Comp1
*
x1 ..... b1
x2 ..... b2
x3 ..... b3

.TRAN ....
.OPTIONS Post
.END
```

_Avant!_

# Chapter 24

# Performing Pole/Zero Analysis

Pole/zero analysis is a useful method for studying the behavior of linear, time-invariant networks, and may be applied to the design of analog circuits, such as amplifiers and filters. It may be used for determining the stability of a design, and it may also be used to calculate the poles and zeroes for specification in a POLE statement as "Using Pole/Zero Analysis" on page 24-3 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 24-3.

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" at the end of this chapter 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) |

### Examples

```
.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 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, 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:

$$\text{If } |X_{imag}| \le RITOL \cdot |X_{real}|, \text{ then } X_{imag} = 0$$

$$\text{If } |X_{real}| \le RITOL \cdot |X_{imag}|, \text{ then } X_{real} = 0$$

*(X0R,X0I)*              the three complex starting trial points in the Muller
*(x1R,X1I)*              algorithm for pole/zero analysis. Defaults:
*(X2R,X21)*                      X0R=-1.23456e6    X0I=0.0
                                 X1R=1.23456e5     X1I=0.0
                                 X2R=+1.23456e6    X21=0.0
                         These initial points are multiplied by FSCAL.

## Pole/Zero Analysis Examples

### Pole/Zero 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
```
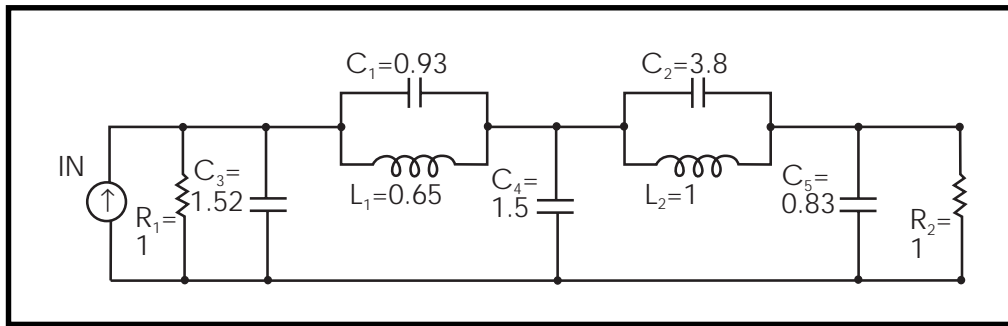


**Figure 24-1:  Low-Pass Prototype Filter**

Table 24-1: 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 24-1:  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 24-1: 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 | | | |
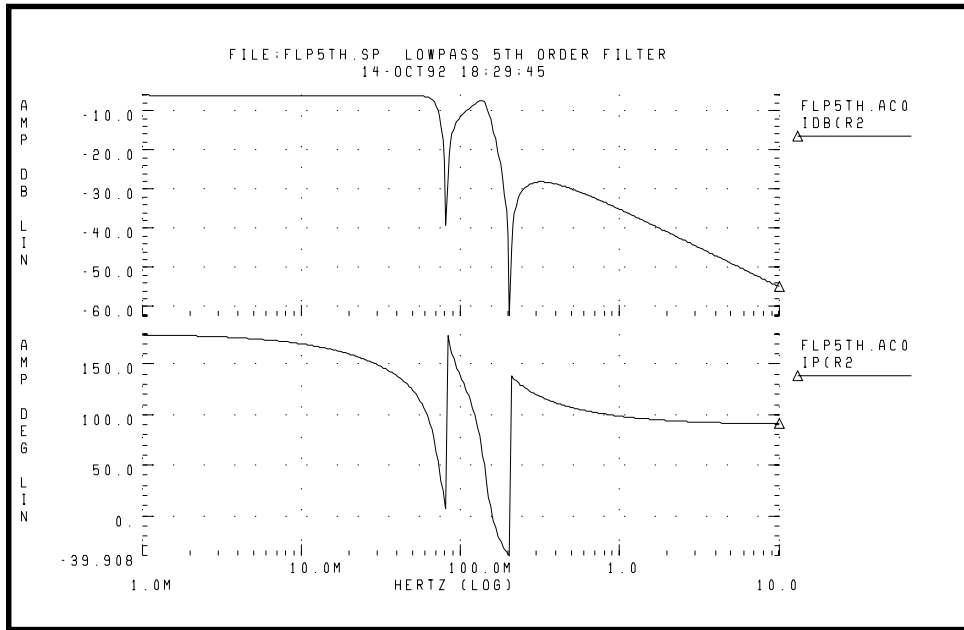


**Figure 24-2: Fifth-Order Low-Pass Filter Response**

## Pole/Zero 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 24-2: 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
```
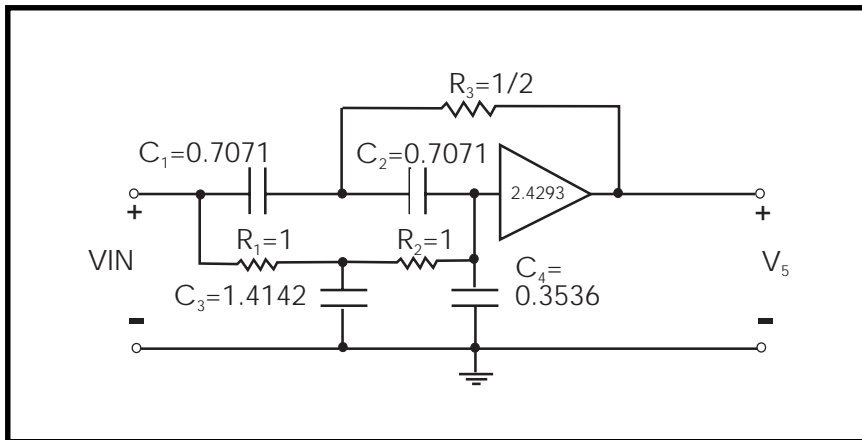


**Figure 24-3: Design Example for Kerwin's Circuit**

## Table 24-2: 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 | | | |

## Pole/Zero 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 24-3:.

### 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
```
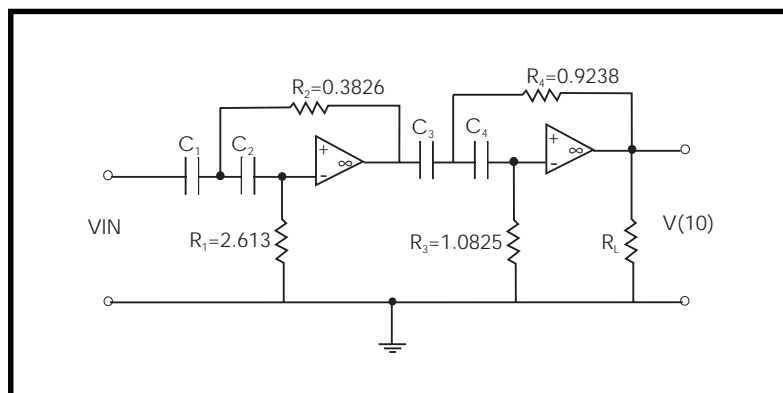


**Figure 24-4:  Fourth-Order High-Pass Butterworth Filter**

### Table 24-3: 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 | | | |

## Pole/Zero 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 24-4:.

### 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 24-5: CMOS Differential Amplifier**

## Table 24-4: 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 | | | |

### Pole/Zero Example 5 – Simple Amplifier

The following is an HSPICE input file for pole/zero analysis of an equivalent circuit of a simple amplifier with RS =RPI=RL=1000 ohms, gm=0.04 mho, CMU=1.0e-11 farad, and CPI =1.0e-9 farad[1]. The file can be found in *$installdir/demo/hspice/apps/ampg.sp*. The analysis results are shown in Table 24-5:.

### 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
```
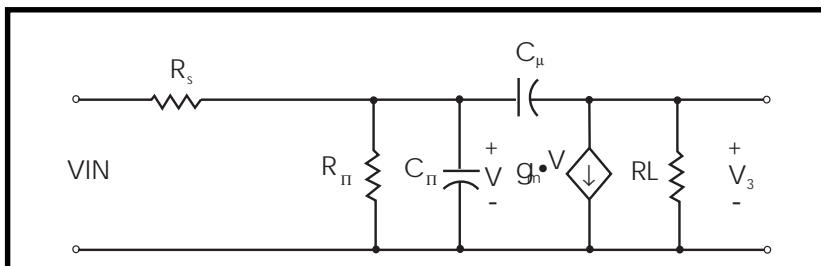


**Figure 24-6: Simple Amplifier**

| 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 | | | |

**Table 24-5:  Pole/Zero Analysis Results for Amplifier**

## Pole/Zero 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 24-6:.

### 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
```
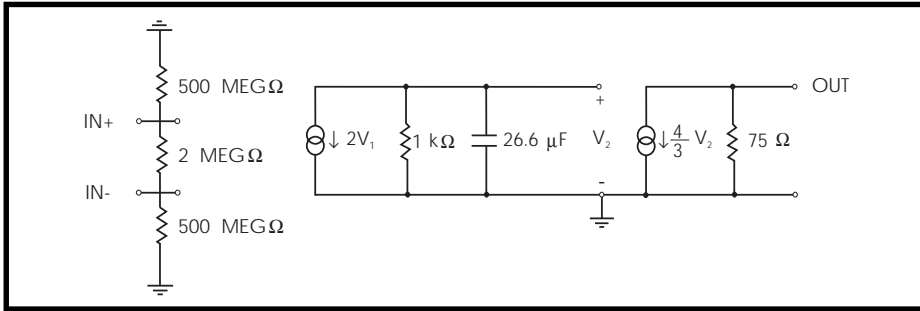
**Figure 24-7: Linear Model of the 741C Op-Amp**
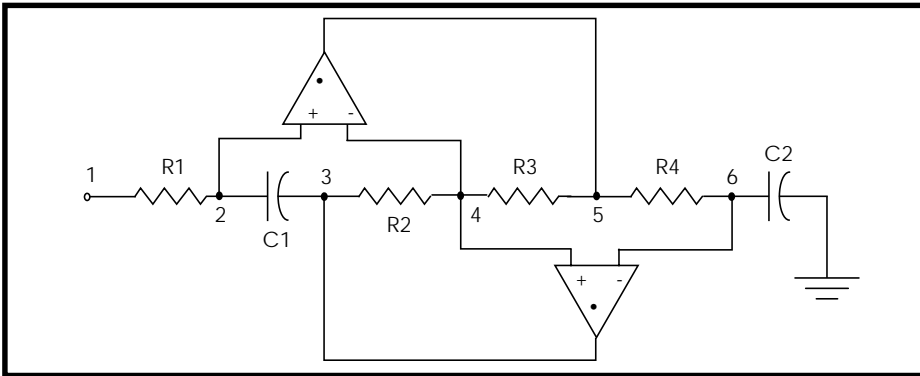


**Figure 24-8: The FDNR Subcircuit**
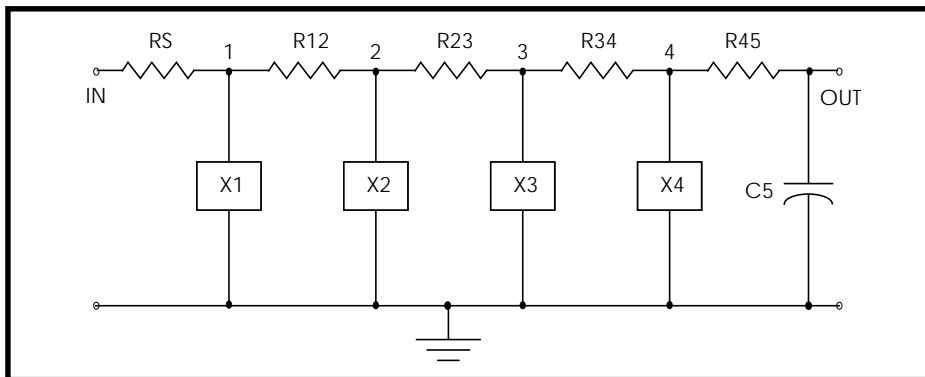


**Figure 24-9: Active Realization of the Low-Pass Filter**

## Table 24-6: 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 |
| 1.637164e+00 | 2.981593e+04 | 2.605627e-01 | 4.745353e+03 |
| 1.637164e+00 | -2.981593e+04 | 2.605627e-01 | -4.745353e+03 |

## Table 24-6: Pole/Zero Analysis Results for the Active Low-Pass Filter

| | | | |
|---|---|---|---|
| 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 | | | |
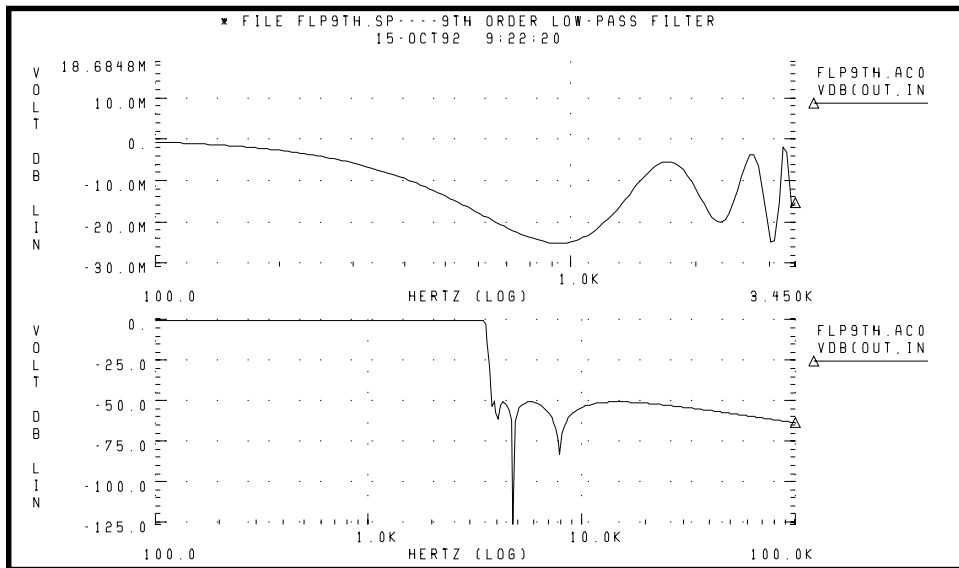


**Figure 24-10:  9th Order Low-Pass Filter Response**

The top graph in Figure 24-10: 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.

# Chapter 25

# 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, through a second order interpolation, obtains waveform samples based on the user-specified number of points. This accounts for the high degree of accuracy of the FFT results in Star-Hspice. Moreover, by using one of the windowing functions, you can 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
- Example 1 – AM Modulation
- Example 2 – 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. Figure 25-1: shows the windows available in Star-Hspice. Table 25-1: lists the common performance parameters for FFT windows available in Star-Hspice.
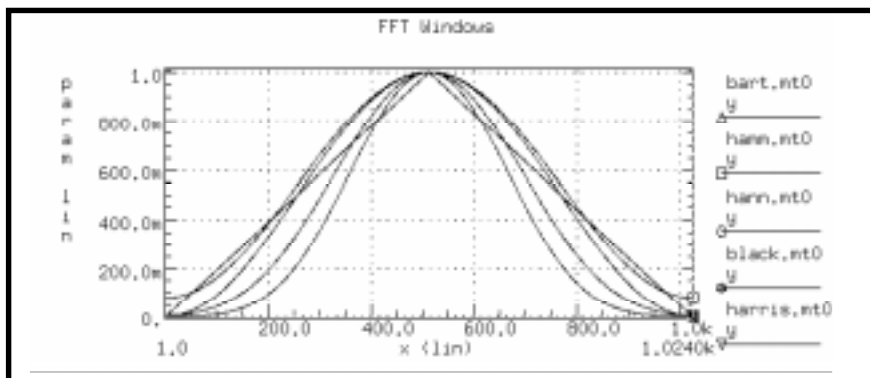


**Figure 25-1: FFT Windows**

## Table 25-1:  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,<br>$0 \le n < NP^{\dagger}$ | -13 | -6 | 0.89 | 3.92 |
| Bartlett | W($n$)=2$n$/(NP–1),<br>$0 \le n \le$ (NP/2)–1<br>W($n$)=2–2$n$/(NP–1),<br>NP/2 $\le n <$ NP | -27 | -12 | 1.28 | 3.07 |
| Hanning | W($n$)=0.5–0.5[cos(2π$n$/(NP–1))],<br>$0 \le n <$ NP | -32 | -18 | 1.44 | 3.18 |
| Hamming | W($n$)=0.54–0.46[cos(2π$n$/(NP–1))],<br>$0 \le n <$ NP | -43 | -6 | 1.30 | 3.10 |
| Blackman | W($n$)=0.42323<br>–0.49755[cos(2π$n$/(NP–1))]<br>+0.07922cos[cos(4π$n$/(NP–1))],<br>$0 \le n <$ NP | -58 | -18 | 1.68 | 3.47 |
| Blackman-Harris | W($n$)=0.35875<br>–0.48829[cos(2π$n$/(NP–1))]<br>+0.14128[cos(4π$n$/(NP–1))]<br>–0.01168[cos(6π$n$/(NP–1))],<br>$0 \le n <$ NP | -92 | -6 | 1.90 | 3.85 |
| Gaussian<br>a=2.5<br>a=3.0<br>a=3.5 | W(n)=exp[-0.5a2(NP/2-1-n)2/(NP)2],<br>$0 \le n \le$ (NP/2)-1<br>W(n)=exp[-0.5a2(n-NP/2)2/(NP)2],<br>NP/2 $\le n <$ NP | -42<br>-55<br>-69 | -6<br>-6<br>-6 | 1.33<br>1.55<br>1.79 | 3.14<br>3.40<br>3.73 |
| Kaiser-Bessel<br>a=2.0<br>a=2.5<br>a=3.0<br>a=3.5 | W(n)=I0(x2)/I0(x1)<br>x1=pa<br>x2=x1*sqrt[1-(2(NP/2-1-n)/NP)2],<br>$0 \le n \le$ (NP/2)-1<br>x2=x1*sqrt[1-(2(n-NP/2)/NP)2],<br>NP/2 $\le n <$ NP<br>I0 is the zero-order modified Bessel function | -46<br>-57<br>-69<br>-82 | -6<br>-6<br>-6<br>-6 | 1.43<br>1.57<br>1.71<br>0.89 | 3.20<br>3.38<br>3.56<br>3.74 |

†NP is the number of points used for the FFT analysis.

The most important parameters in Table 25-1: 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 25-2 and 25-3.
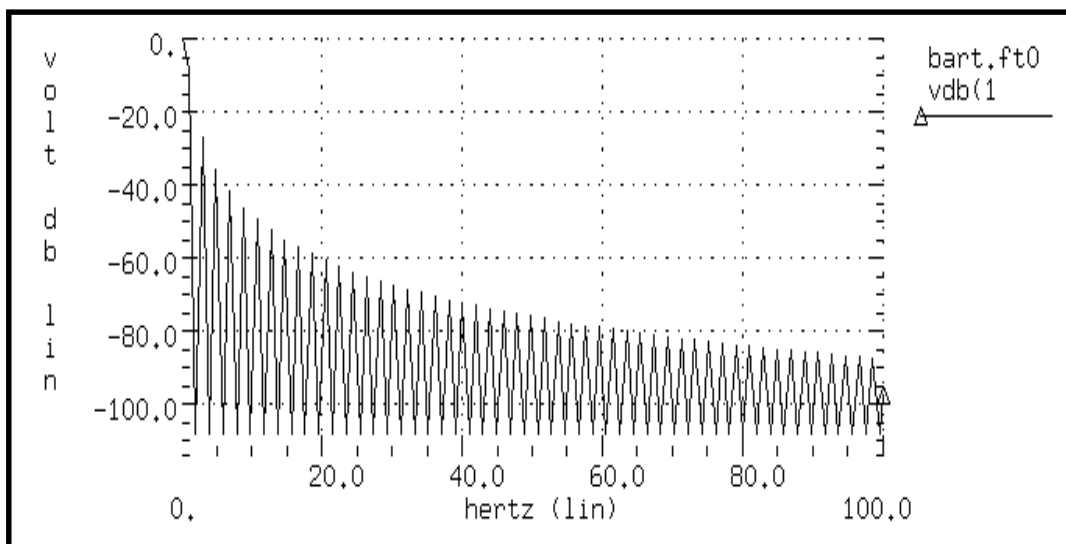


**Figure 25-2: Bartlett Window Characteristics**
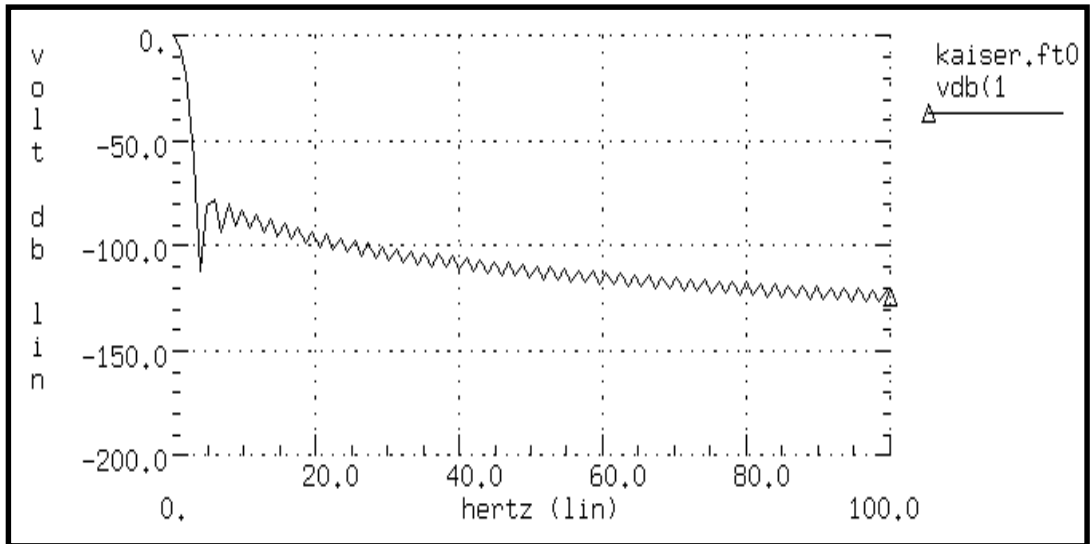
**Figure 25-3: Kaiser-Bessel Window Characteristics, ALFA=3.0**

# Using the .FFT Statement

The general form of the .FFT statement is shown below. The parameters are described in Table 25-2:.

```
.FFT <output_var> <START=value> <STOP=value> <NP=value>
<FORMAT=keyword> <WINDOW=keyword> <ALFA=value> <FREQ=value>
<FMIN=value> <FMAX=value>
```

### Table 25-2:  .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:<br><br>NORM= normalized magnitude<br>UNORM=unnormalized magnitude |

# Table 25-2:  .FFT Statement Parameters

| Parameter | Default | Description |
| --- | --- | --- |
| WINDOW | RECT | specifies the window type to be used:<br><br>RECT=simple rectangular truncation window<br>BART=Bartlett (triangular) window<br>HANN=Hanning window<br>HAMM=Hamming window<br>BLACK=Blackman window<br>HARRIS=Blackman-Harris window<br>GAUSS=Gaussian window<br>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<br><br>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.<br><br>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. |

### Syntax Examples

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 '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 25-3: describes the output of the Star-Hspice FFT analysis.

### Table 25-3:  .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 MetaWaves. The magnitude in dB and the phase in degrees are available for display.

**Notes:**

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

$$\text{frequency increment} = 1.0/(STOP - START)$$

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.

# Example 1 – AM Modulation

This example input listing on the following page shows a 1 kHz carrier (FC) that is modulated by a 100 Hz signal (FM). The voltage at node 1, which is an AM signal, can be described by

$$v(1) = sa \cdot (offset + \sin(\omega_m(Time - td))) \cdot \sin(\omega_c(Time - td))$$

The preceding equation can be expanded as follows.

$$v(1) = (sa \cdot offset \cdot \sin(\omega_c(Time - td)) + 0.5 \cdot sa \cdot \cos((\omega_c - \omega_m)(Time - td)))$$

$$- 0.5 \cdot sa \cdot \cos((\omega_c + \omega_m)(Time - td))$$

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
```

*Example 1 – AM Modulation*                    *Performing FFT Spectrum Analysis*

## 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 25-4 and 25-5 display the results. Figure 25-4: shows the time domain curve of node 1. Figure 25-5: 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.
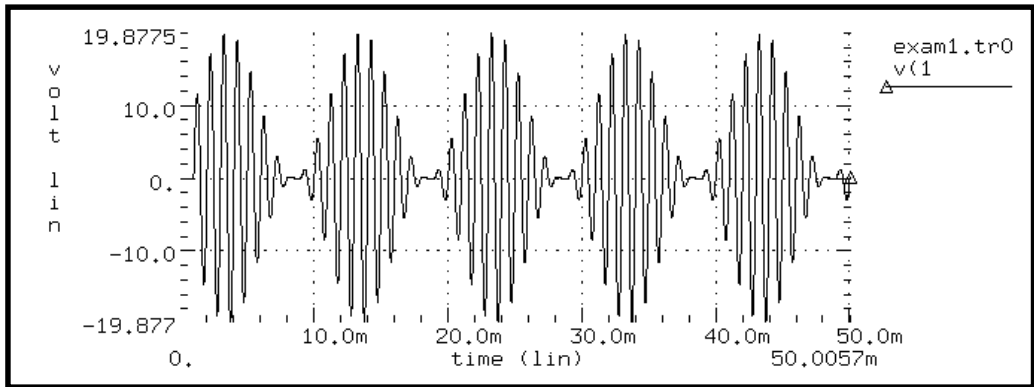
**Figure 25-4: AM Modulation**
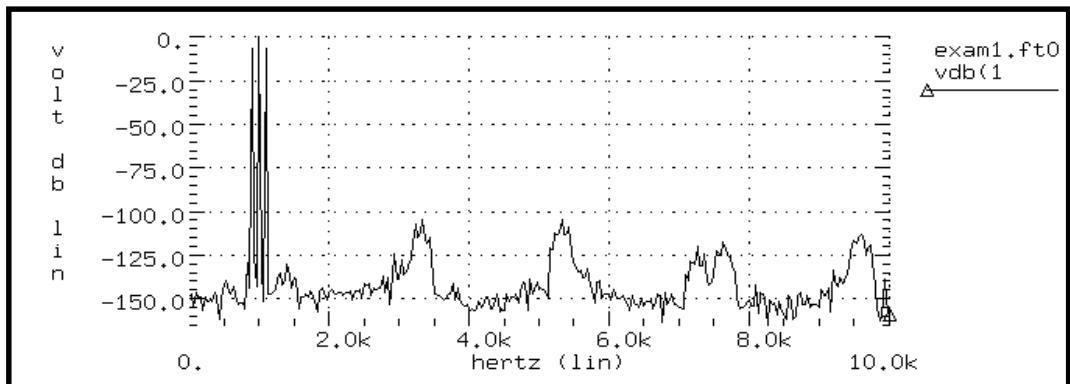


**Figure 25-5: AM Modulation Spectrum**

# Example 2 – 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 25-6 through 25-14. 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 25-9: is normalized.
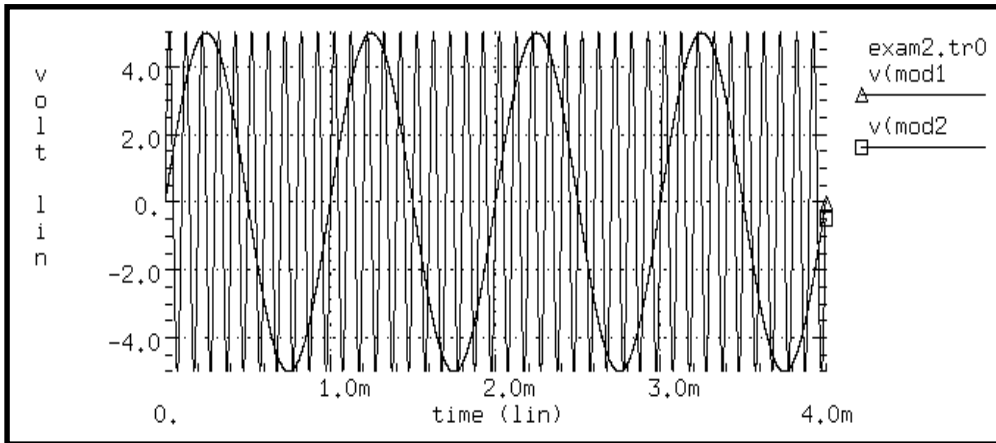
**Figure 25-6: Modulating and Modulated Signals**
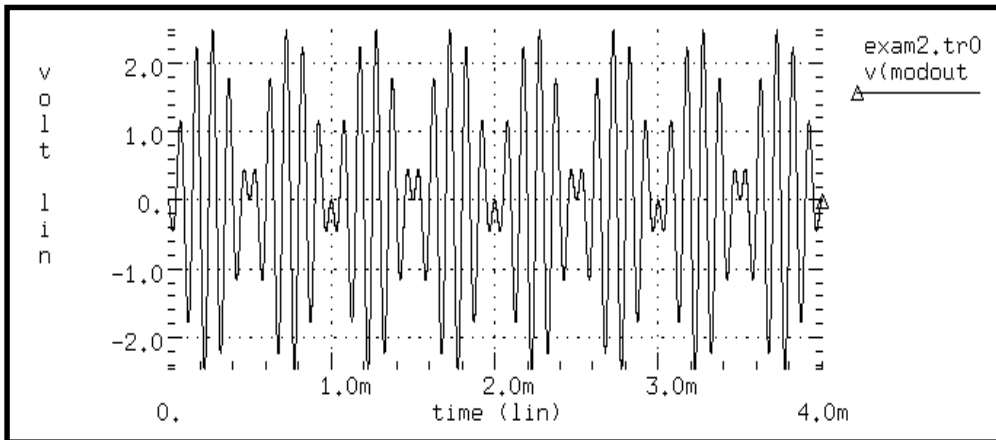


**Figure 25-7: Modulated Signal**
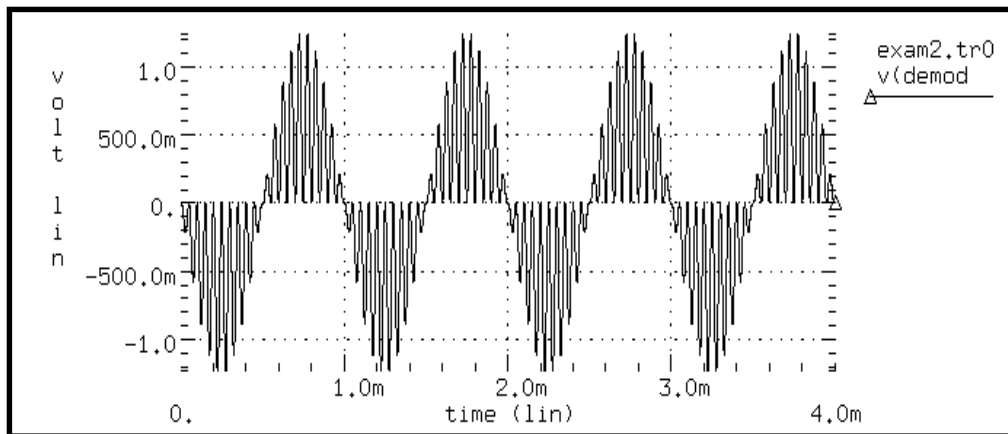
**Figure 25-8: Demodulated Signal**
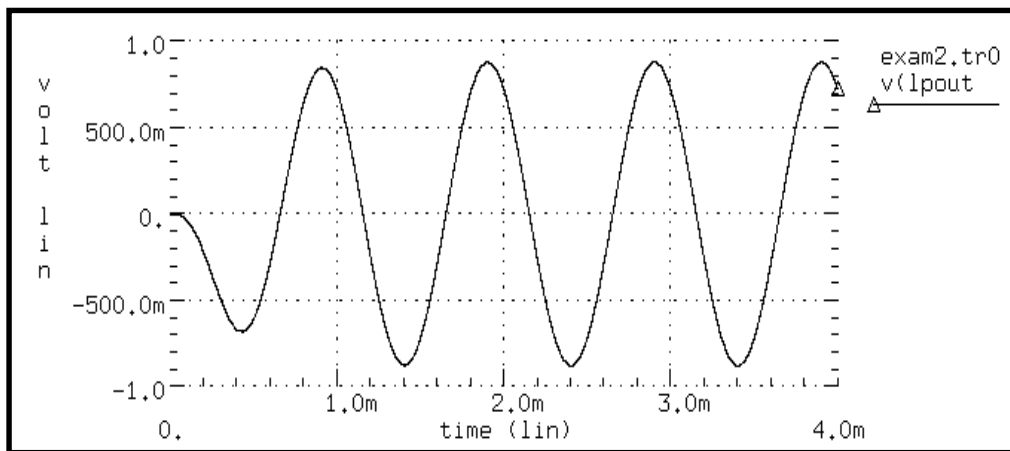


**Figure 25-9: Filtered Output Signal**
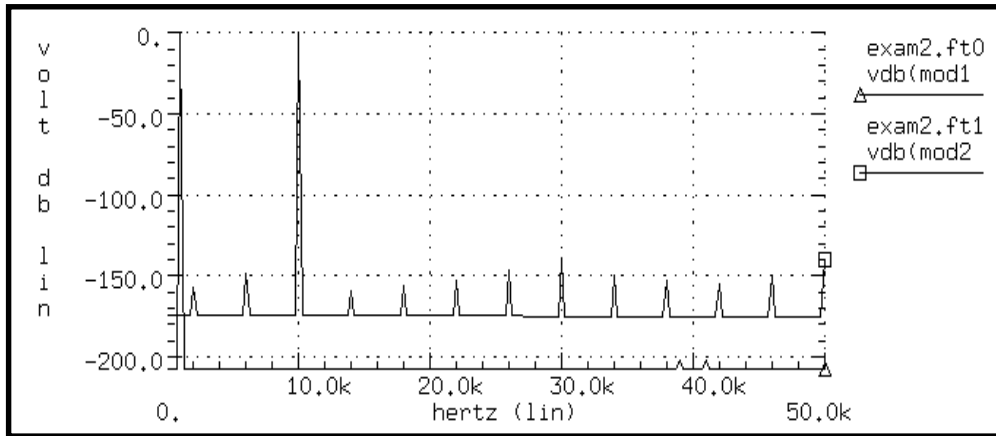
**Figure 25-10: Modulating and Modulated Signal Spectrum**
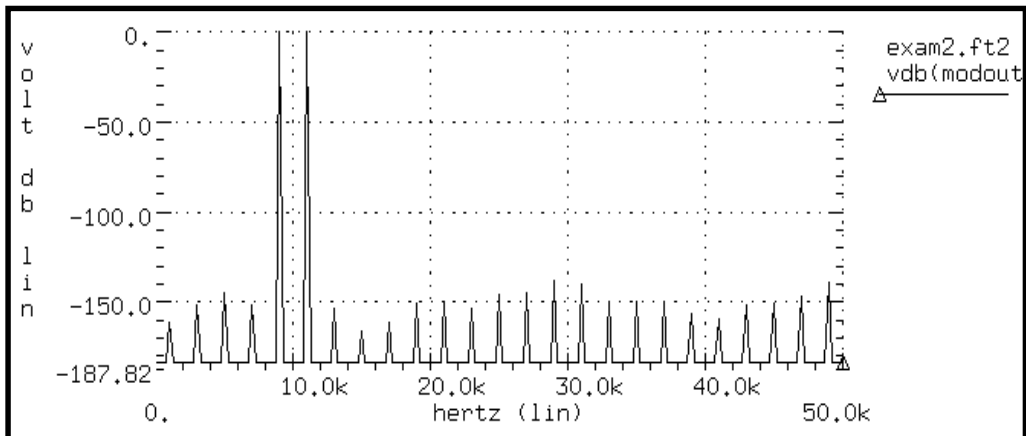


**Figure 25-11: Modulated Signal Spectrum**
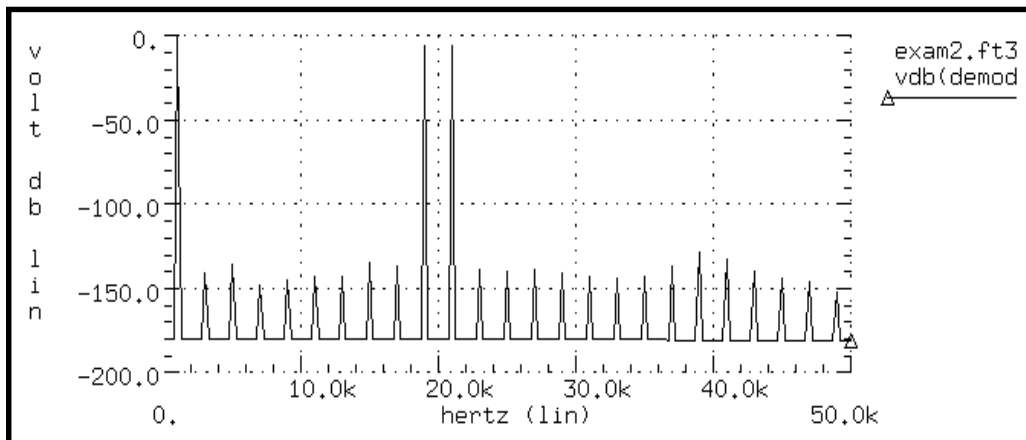
**Figure 25-12: Demodulated Signal Spectrum**
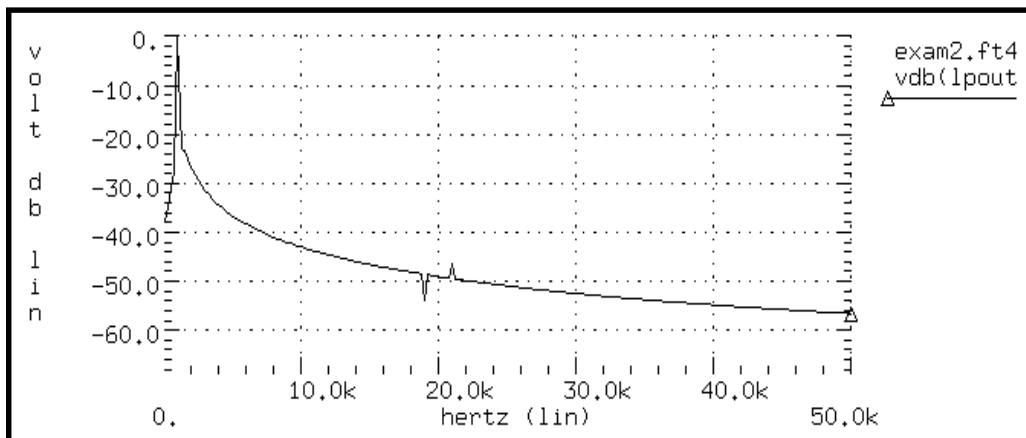


**Figure 25-13: Filtered Output Signal (no window)**
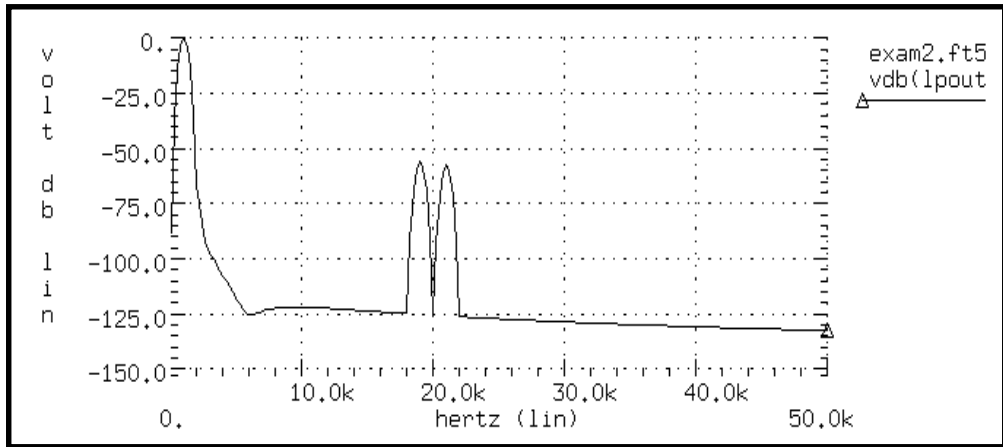
**Figure 25-14: Filtered Output Signal (Blackman-Harris window)**

## Example 3 – 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 25-15:, the spectra of the output for all of the FFT window types are shown in Figures 25-16 through 25-22. Without windowing, the weak signal is essentially undetectable due to spectral leakage.
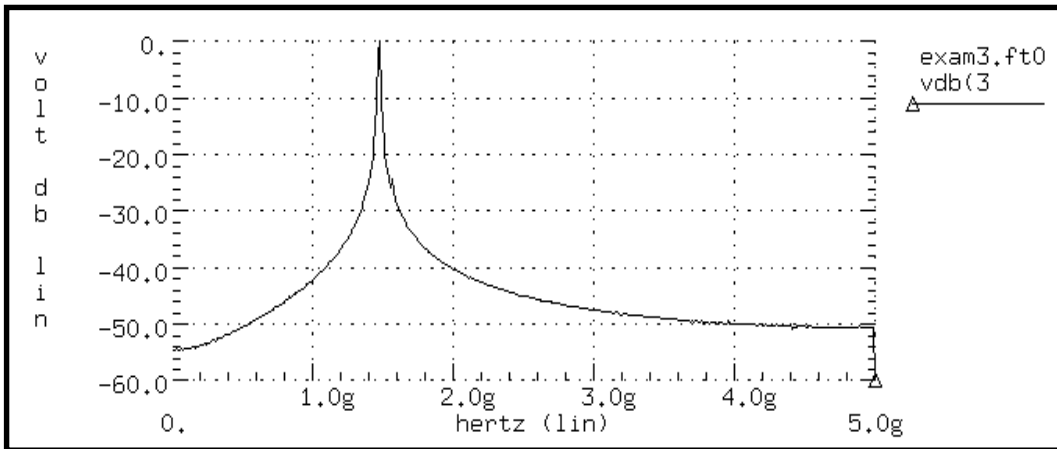


**Figure 25-15: Mixer Output Spectrum, Rectangular Window**

In the Bartlett window in Figure 25-16:, 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 25-21 and 25-22) 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.
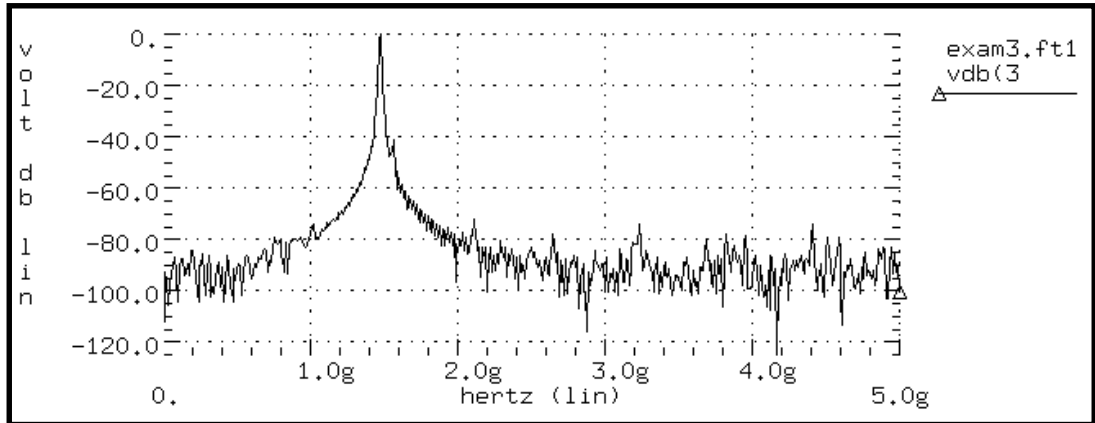


**Figure 25-16: Mixer Output Spectrum, Bartlett Window**
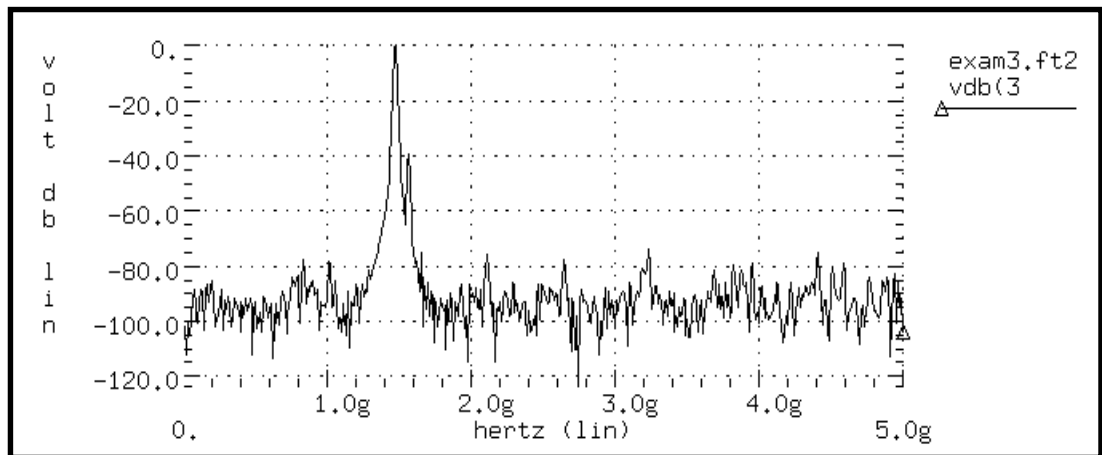


**Figure 25-17: Mixer Output Spectrum, Hanning Window**
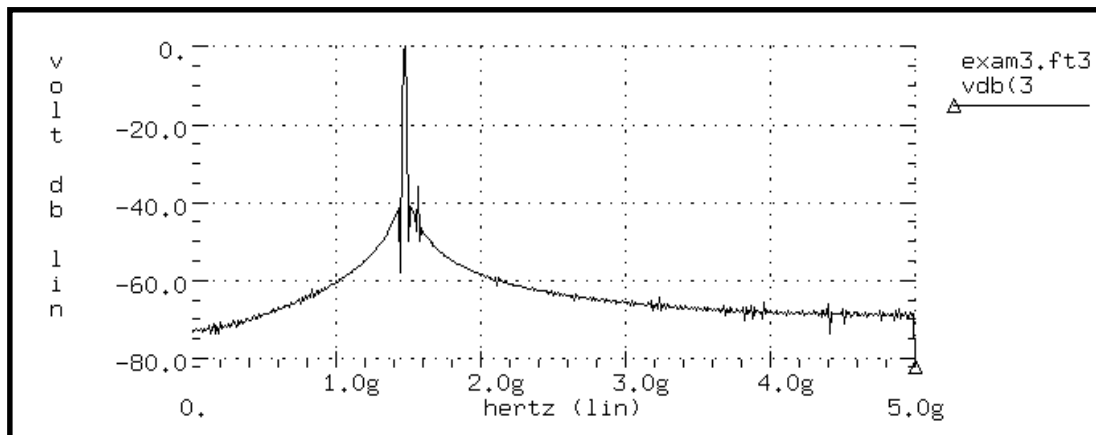
**Figure 25-18: Mixer Output Spectrum, Hamming Window**
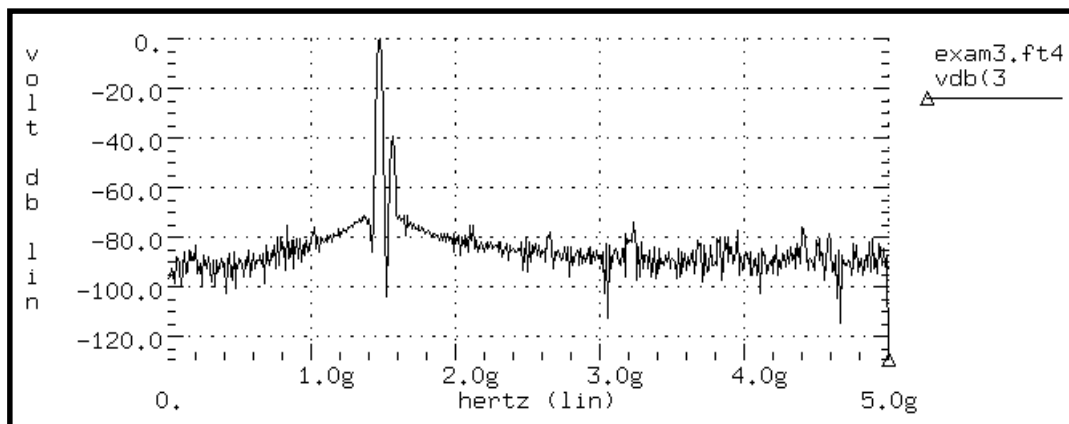


**Figure 25-19: Mixer Output Spectrum, Blackman Window**
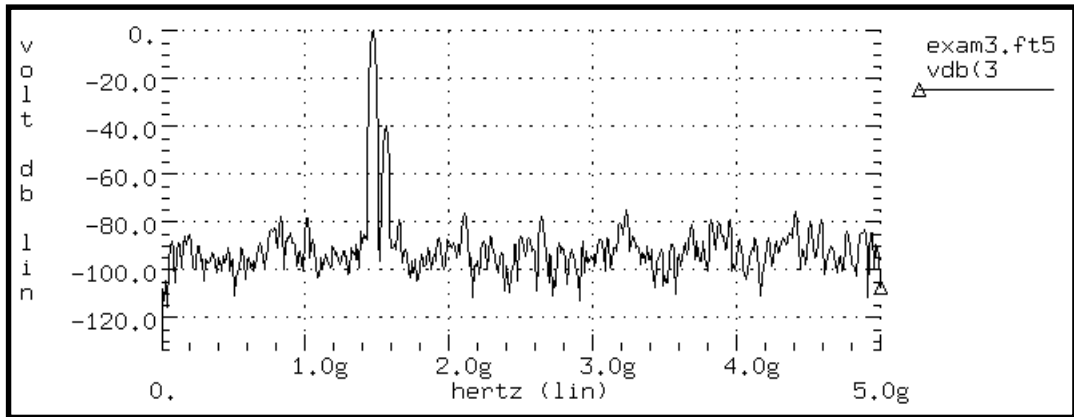
**Figure 25-20: Mixer Output Spectrum, Blackman-Harris Window**
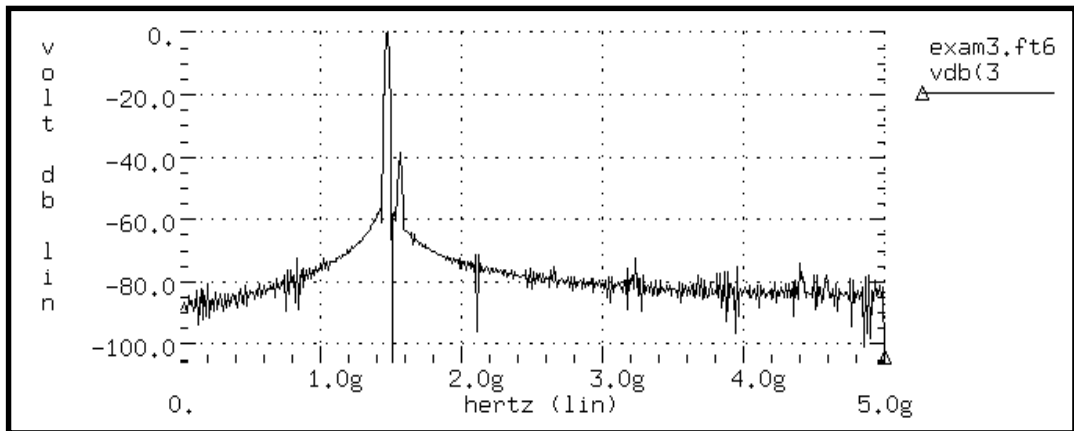


**Figure 25-21: Mixer Output Spectrum, Gaussian Window**
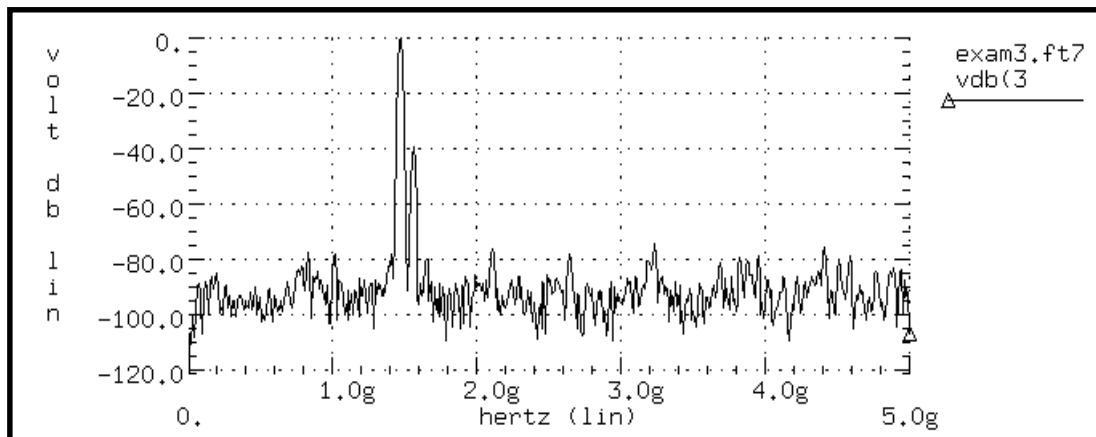
**Figure 25-22: Mixer Output Spectrum, Kaiser-Bessel Window**

## 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.

# Chapter 26

# 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
- Modeling with Laplace and Pole-Zero
- 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 "Using Pole/Zero Analysis" on page 24-3 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 transforms

■ Pole/zero modeling

■ 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 $\Delta$ is the time interval or time resolution.

For the frequency response table form (FREQ) of the LAPLACE function, Star-Hspice's performance-enhanced algorithm is used to convert H(f) to h(t). This algorithm requires N to be a power of 2. The frequency point $f_n$ is determined by

$$f_n \;=\; \frac{n}{N \cdot \Delta} \; , \;\;\; n = 0, 1, 2, ..., N\text{-}1$$

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 $\Delta$ 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 $\Delta$ can be determined by

$$\Delta = \frac{1}{2 \cdot f_c}$$

Notice the frequency resolution

$$\Delta f = f_1 = \frac{1}{N \cdot \Delta}$$

is inversely proportional to the maximum time ($N \cdot \Delta$) over which h(t) is evaluated. Therefore, the transient analysis accuracy also depends on the frequency resolution or the number of points (N). You can specify the frequency resolution DELF and maximum frequency MAXF in the G or E element statement. N is calculated by $2 \cdot$MAXF/DELF. Then, N is modified to be a power of 2. The effective DELF is determined by $2 \cdot$MAXF/N to reflect the changes in N.

## Laplace Transform – LAPLACE Function

The syntax is:

Transconductance H(s):

    Gxxx $n_+$ $n_-$ LAPLACE $in_+$ $in_-$   $k_0$, $k_1$, ..., $k_n$ / $d_0$, $d_1$, ..., $d_m$

    + <SCALE=val> <TC1=val> <TC2=val> <M=val>

Voltage Gain H(s):

    Exxx $n_+$ $n_-$ LAPLACE $in_+$ $in_-$   $k_0$, $k_1$, ..., $k_n$ / $d_0$, $d_1$, ..., $d_m$

    <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.

### Examples

```
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

$$H(s) = \frac{s^3}{1 + 2s + 2s^2 + s^3}$$

## Laplace Transform – Pole-Zero Function

General Forms

Transconductance H(s):

Gxxx $n_+$ $n_-$ POLE $in_+$ $in_-$ *a* $\alpha_{z1}$, $f_{z1}$, ..., $\alpha_{zn}$, $f_{zn}$ / *b*, $\alpha_{p1}$, $f_{p1}$, ..., $\alpha_{pm}$, $f_{pm}$
+ <SCALE=val> <TC1=val> <TC2=val> <M=val>

Voltage Gain H(s):

Exxx $n_+$ $n_-$ POLE $in_+$ $in_-$ *a* $\alpha_{z1}$, $f_{z1}$, ..., $\alpha_{zn}$, $f_{zn}$ / *b*, $\alpha_{p1}$, $f_{p1}$, ..., $\alpha_{pm}$, $f_{pm}$
+ <SCALE=val> <TC1=val> <TC2=val>

H(s) in terms of poles and zeros is defined by

$$H(s) = \frac{a \cdot (s + \alpha_{z1} - j2\pi f_{z1})\ldots(s + \alpha_{zn} - j2\pi f_{zn})(s + \alpha_{zn} + j2\pi f_{zn})}{b \cdot (s + \alpha_{p1} - j2\pi f_{p1})\ldots(s + \alpha_{pm} - j2\pi f_{pm})(s + \alpha_{pm} + j2\pi f_{pm})}$$

Notice the complex poles or zeros are in conjugate pairs. In the element description, only one of them is specified, and the program includes the conjugate. The a, b, $\alpha$, and f values can be parameterized.

### Examples

```
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

$$H(s) = \frac{1.0}{1.0 \cdot (s + 1)(s + 0.5 + j2\pi \cdot 0.1379)(s + 0.5 - (j2\pi \cdot 0.1379))}$$

## Laplace Transform- Frequency Response Table

The syntax is:

Transconductance H(s):

    Gxxx  n+ n- FREQ in+ in-   $f_1$, $a_1$, $\phi_1$, ..., $f_i$, $a_i$, $\phi_1$

    + <DELF=val> <MAXF=val> <SCALE=val> <TC1=val>

    + <TC2=val> <M=val> <LEVEL=val>

Voltage Gain H(s):

    Exxx n+ n- FREQ in+ in-   $f_1$, $a_1$, $\phi_1$, ..., $f_i$, $a_i$, $\phi_1$

    + <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 $\phi_1$ 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 $\Delta f$. The inverse of DELF is the time window over which h(t) is calculated from H(s). The smaller DELF is, the more accurate the transient analysis, and the longer the CPU time. The number of points, N, used in the conversion of H(s) to h(t) is N=2·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: |

$$\text{A}LEeff \ = \ SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta$$

*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.
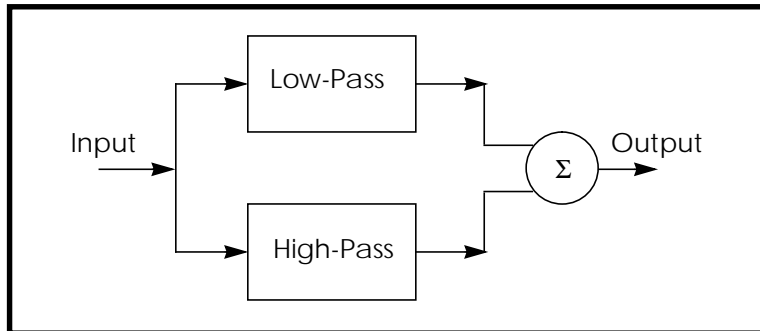


**Figure 26-1: Band-Reject Filter**

### Example Band-Reject Filter

```
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
```
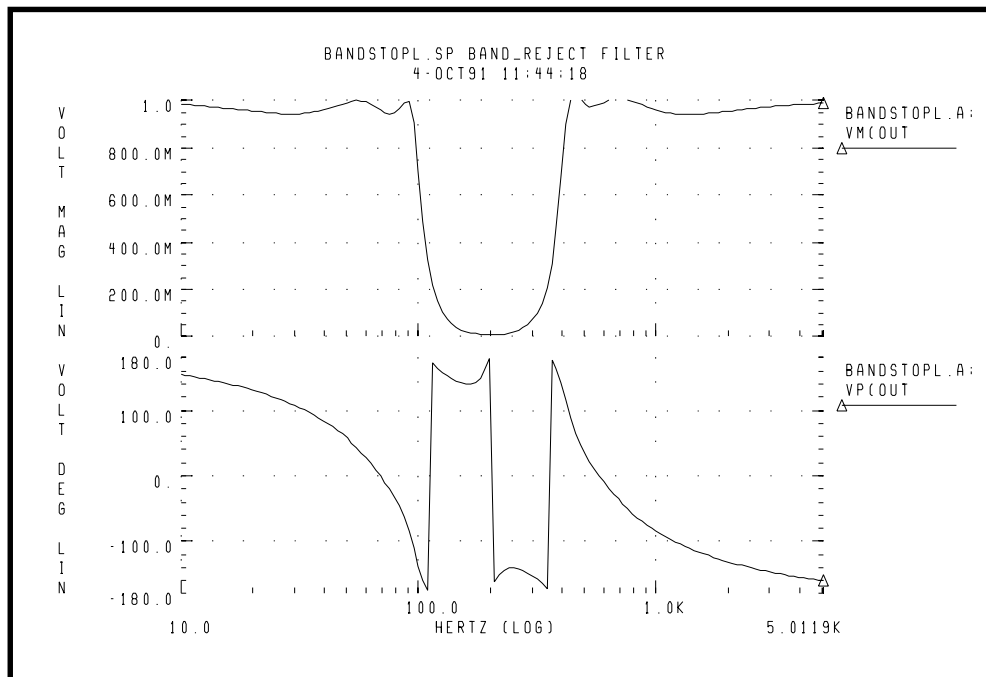


**Figure 26-2: Frequency Response of the Band-Reject Filter**
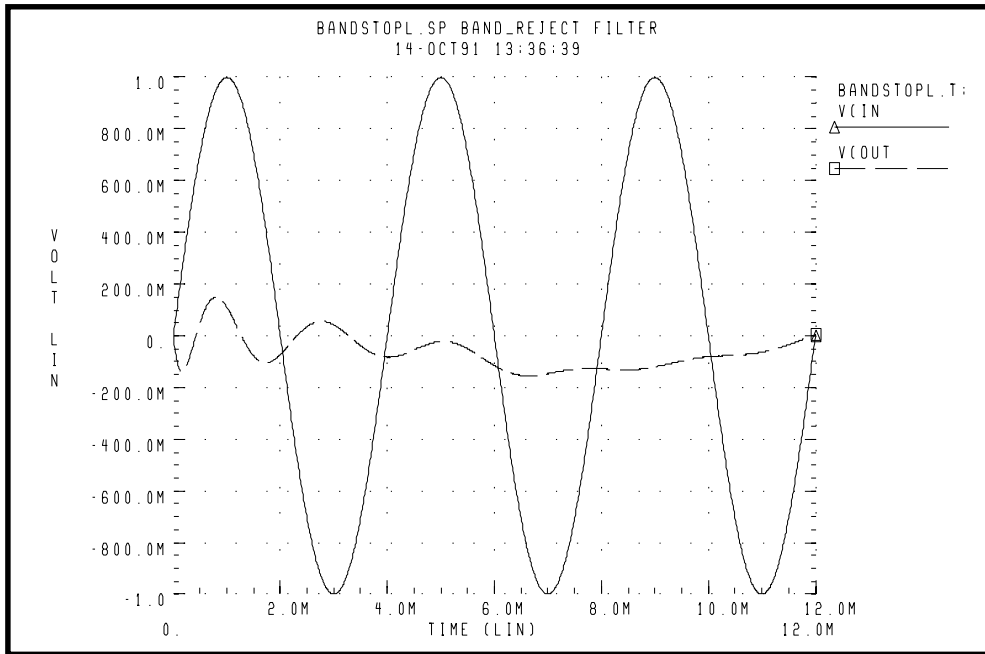
**Figure 26-3: Transient Response of the Band-Reject Filter to a 250 Hz Sine Wave**

## 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.
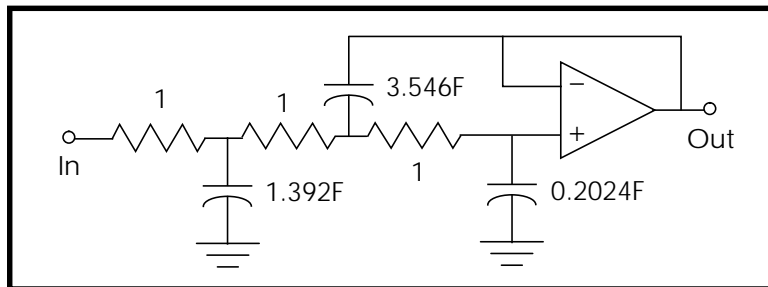


**Figure 26-4: Third-Order Active Low-Pass Filter**

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 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
```
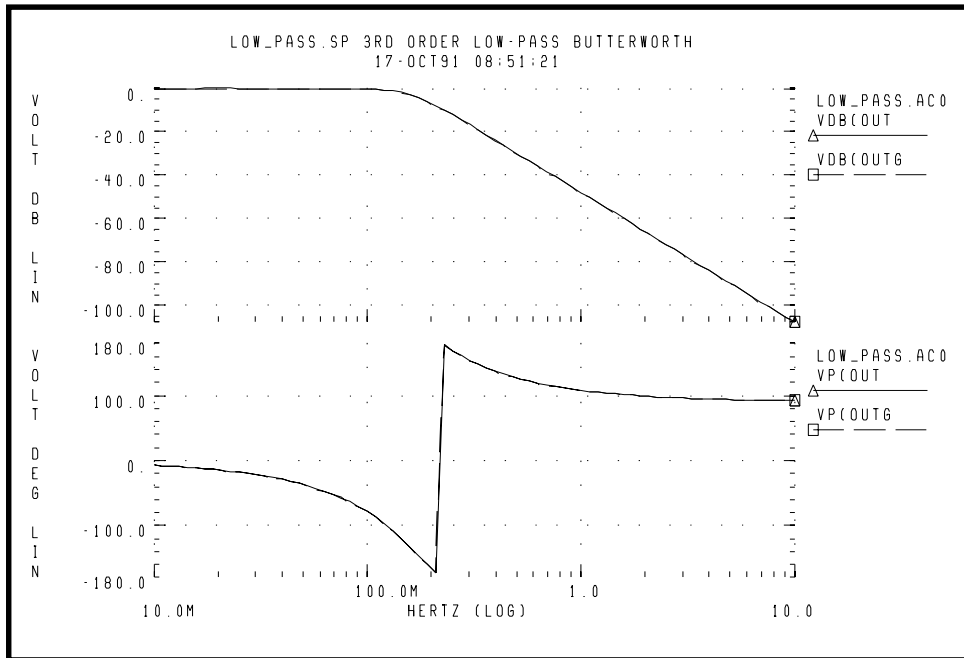
**Figure 26-5: Frequency Response of Circuit and Functional Element**
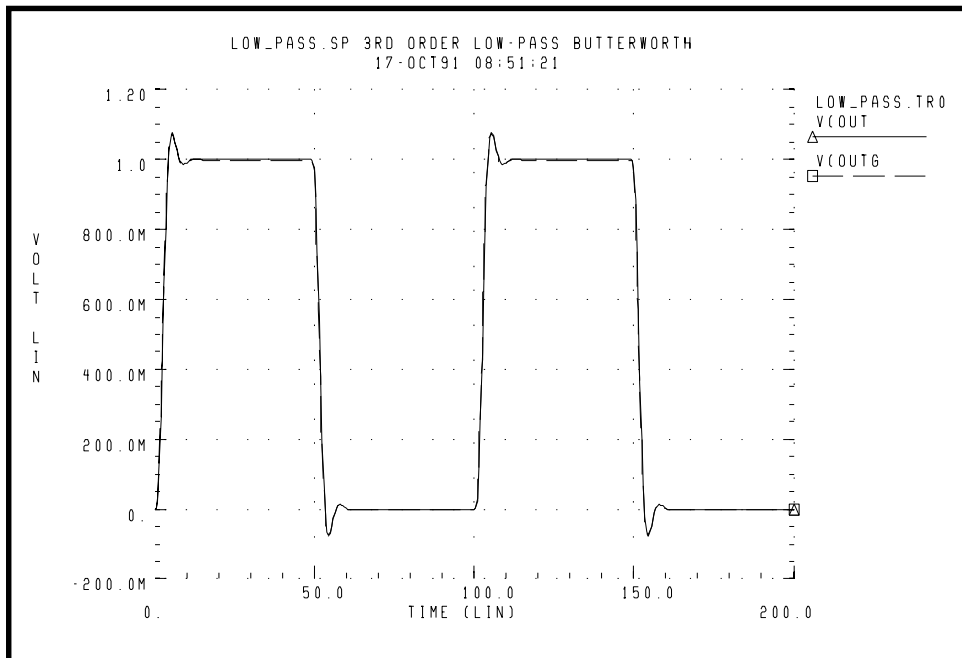
**Figure 26-6: Transient Response of Circuit and Functional Element to a Pulse**

# Modeling with Laplace and Pole-Zero

## The 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 -4 for the general forms and 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 *j2πf*, *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 26-1 and 26-2 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 26-1: 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 26-1: 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 26-2: 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 26-2: Laplace Transforms for Common Operations

| f(t) | $L\{f(t)\} = F(s)$ |
|------|--------------------|
| $\dfrac{d^2}{dt^2}f(t)$ | $s^2 F(s) - sf(0) - \dfrac{d}{dt}f(0-\ )$ |
| $\dfrac{d^n}{dt^n}f(t)$ | $s^n F(s) - s^{n-1}f(0) - s^{n-2}\dfrac{d}{dt}f(0)$ |
| $\displaystyle\int_{-\infty}^{t} f(t)dt$ | $\dfrac{F(s)}{s} + \dfrac{f^{-1}(0)}{s}$ |
| $f(t-a)u(t-a), a > 0$ <br><br> (*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 26-7:, convert the circuit to its equivalent *s*-domain circuit and solve for $v_o / v_g$.
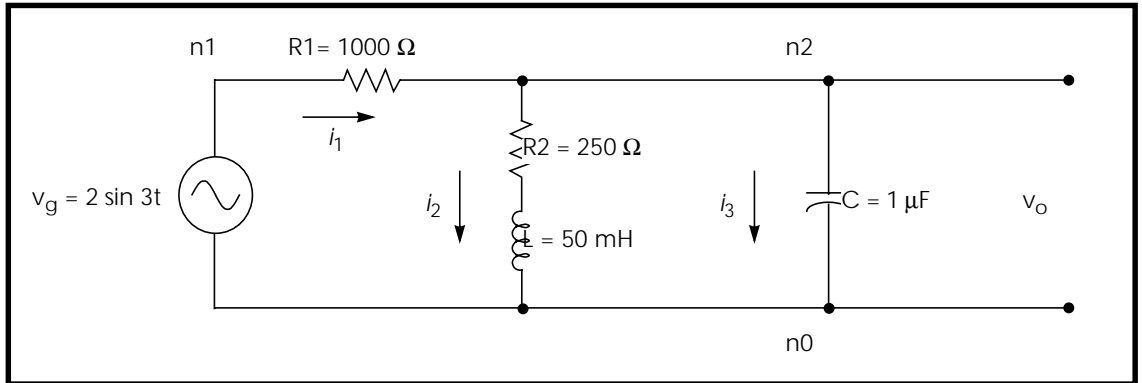


**Figure 26-7: LAPLACE Example 1 Circuit**

Use transforms from Table 26-2: 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 26-1:

$$L\{2\sin 3t\} = 2 \cdot \frac{3}{s^2 + 3^2} = \frac{6}{s^2 + 9}$$

Figure 26-8: displays the *s*-domain equivalent circuit.
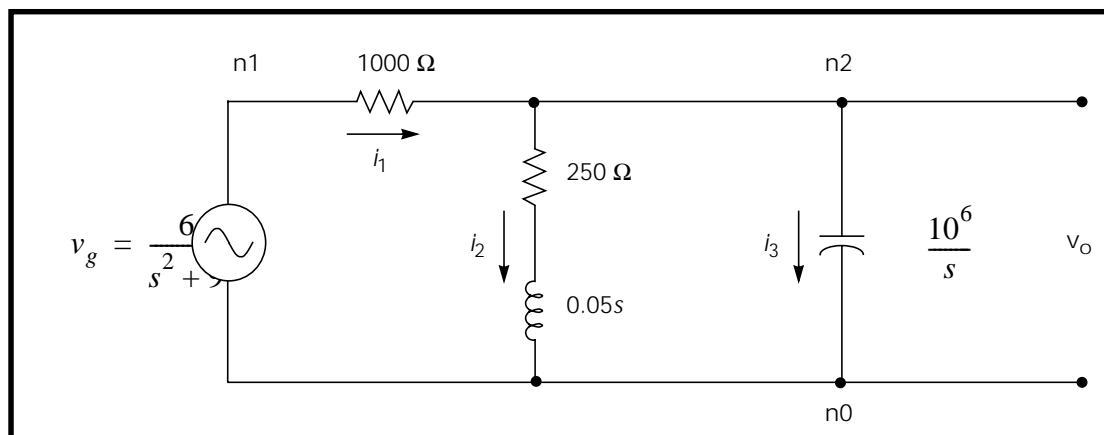


**Figure 26-8: S-Domain Equivalent of the LAPLACE Example 1 Circuit**

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

$k_0 = 5 \times 10^6$　　　　$k_1 = 1000$

$d_0 = 25 \times 10^6$　　　　$d_1 = 6000$　　　　$d_2 = 1$

Using these coefficients, a Star-Hspice Laplace modeling function call for the voltage gain transfer function of the circuit in Figure 26-7: 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:* $V_{\text{out}} = ksV_{\text{in}}$

*G element:* $I_{\text{out}} = ksV_{\text{in}}$

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:

$$E \text{ Element: } V_{out} = \frac{k}{s} V_{in}$$

$$G \text{ Element: } I_{out} = \frac{k}{s} V_{in}$$

In the time domain:

*E Element:* $v_{out} = k \int V_{in} dt$

*G Element:* $i_{out} = k \int V_{in} dt$

For an integrator, the voltage gain transfer function is:

$$H(s) = \frac{V_{out}}{V_{in}} = \frac{k}{s}$$

In the general form of the transfer function,

$$H(s) = \frac{k_0 + k_1 s + \ldots + k_n s^n}{d_0 + d_1 s + \ldots + d_m s^m}$$

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.

### The 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 "Using Pole/Zero Analysis" on page 24-3.

There are two forms of the Star-Hspice LAPLACE function call, one for transconductance and one for voltage gain transfer functions. See "Using G and E Elements" on page -4 for the general forms and list of optional parameters.

To use the POLE pole/zero modeling function, find the $a$, $b$, $f$, and $\alpha$ coefficients of the transfer function. The transfer function is the *s*-domain (frequency domain) ratio of the output of a single-source circuit to the input, with initial conditions set to zero.

## General Form of the Transfer Function

The general expanded form of the pole/zero transfer function *H(s)* is:

$$H(s) = \frac{a(s + \alpha_{z1} + j2\pi f_{z1})(s + \alpha_{z1} - j2\pi f_{z1})\dots(s + \alpha_{zn} + j2\pi f_{zn})(s + \alpha_{zn} - j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})(s + \alpha_{p1} - j2\pi f_{p1})\dots(s + \alpha_{pm} + j2\pi f_{pm})(s + \alpha_{pm} - j2\pi f_{pm})}$$

*(1)*

The a, b, $\alpha$, and f values can be parameterized.

## Examples

```
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 an Star-Hspice pole/zero circuit description for an element, you need to know the element's transfer function *H(s)* in terms of the *a*, *b*, *f*, and $\alpha$ coefficients. Use the values of these coefficients in POLE function calls in the Star-Hspice circuit description.

First, however, simplify the transfer function, as described in the next section.

## Star-Hspice Reduced Form of the Transfer Function

Complex poles and zeros occur in conjugate pairs (a set of complex numbers differ only in the signs of their imaginary parts):

$$(s + \alpha_{pm} + j2\pi f_{pm})(s + \alpha_{pm} - j2\pi f_{pm}), \text{ for poles}$$

and

$$(s + \alpha_{zn} + j2\pi f_{zn})(s + \alpha_{zn} - j2\pi f_{zn}), \text{ for zeros}$$

To write the transfer function in Star-Hspice pole/zero format, supply coefficients for one term of each conjugate pair and Star-Hspice provides the coefficients for the other term. If you omit the negative complex roots, the result is the reduced form of the transfer function, *Reduced{H(s)}*. Find the reduced form by collecting all the general form terms with negative complex roots:

$$H(s) = \frac{a(s + \alpha_{z1} + j2\pi f_{z1})\dots(s + \alpha_{zn} + j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})\dots(s + \alpha_{pm} + j2\pi f_{pm})} \cdot \frac{a(s + \alpha_{z1} - j2\pi f_{z1})\dots(s + \alpha_{zn} -}{b(s + \alpha_{p1} - j2\pi f_{p1})\dots(s + \alpha_{pm} -}$$

*(1)*

Then discard the right-hand term, which contains all the terms with negative roots. What remains is the reduced form:

$$Reduced\{H(s)\} = \frac{a(s + \alpha_{z1} + j2\pi f_{z1})\dots(s + \alpha_{zn} + j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})\dots(s + \alpha_{pm} + j2\pi f_{pm})} \text{ (2)}$$

For this function find the *a*, *b*, *f*, and $\alpha$ coefficients to use in an Star-Hspice POLE function for a voltage gain transfer function. The following examples show how to determine the coefficients and write POLE function calls for a high-pass filter and a low-pass filter.

### POLE Example 1 – Highpass Filter

For a high-pass filter with a given transconductance transfer function, such as

$$H(s) = \frac{s}{(s + 0.001)}$$

Find the $a$, $b$, $\alpha$, and $f$ coefficients necessary to write the transfer function in the general form (1) shown previously, so that you can clearly see the conjugate pairs of complex roots. You only need to supply one of each conjugate pair of roots in the Laplace function call. Star-Hspice automatically inserts the other root.

To get the function into a form more similar to the general form of the transfer function, rewrite the given transconductance transfer function as

$$H(s) = \frac{1.0(s + 0.0)}{1.0(s + 0.001)}$$

Since this function has no negative imaginary parts, it is already in the Star-Hspice reduced form (2) shown previously. Now you can identify the $a$, $b$, $f$, and $\alpha$ coefficients so that the transfer function $H(s)$ matches the reduced form. This matching process obtains the following values:

*n = 1, m = 1,*

*a = 1.0     $\alpha_{z1}$ = 0.0          $f_{z1}$ = 0.0*

*b = 1.0     $\alpha_{p1}$ = 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 $a$, $b$, $\alpha$, and $f$ coefficients to write the transfer function in the general form, so that you can identify the complex roots with negative imaginary parts.

To separate the reduced form, *Reduced*{*H*(*s*)}, from the terms with negative imaginary parts, rewrite the given voltage gain transfer function as

$$H(s) = \frac{1.0}{1.0(s + 1.0 + j2\pi \cdot 0.0)(s + 0.5 + j2\pi \cdot 0.15)} \cdot \frac{1.0}{(s + 0.5 - j2\pi \cdot 0.15)}$$

$$= Reduced\{H(s)\} \cdot \frac{1.0}{(s + 0.5 - j2\pi \cdot 0.15)}$$

So

$$Reduced\{H(s)\} = \frac{1.0}{1.0(s + 1.0)(s + 0.5 + j2\pi \cdot 0.15)}$$

or

$$\frac{a(s + \alpha_{z1} + j2\pi f_{z1})\ldots(s + \alpha_{zn} + j2\pi f_{zn})}{b(s + \alpha_{p1} + j2\pi f_{p1})\ldots(s + \alpha_{pm} + j2\pi f_{pm})} = \frac{1.0}{1.0(s + 1.0 + j2\pi \cdot 0.0)(s + 0.5 + j2\pi}$$

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.
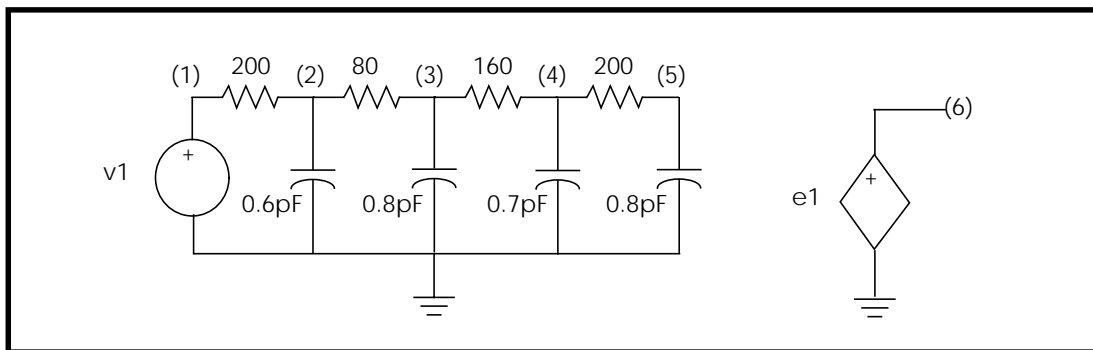


**Figure 26-9: Circuits for an RC Line**

### 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 26-9: can be closely approximated by a single pole response, as shown in Figure 26-10:.
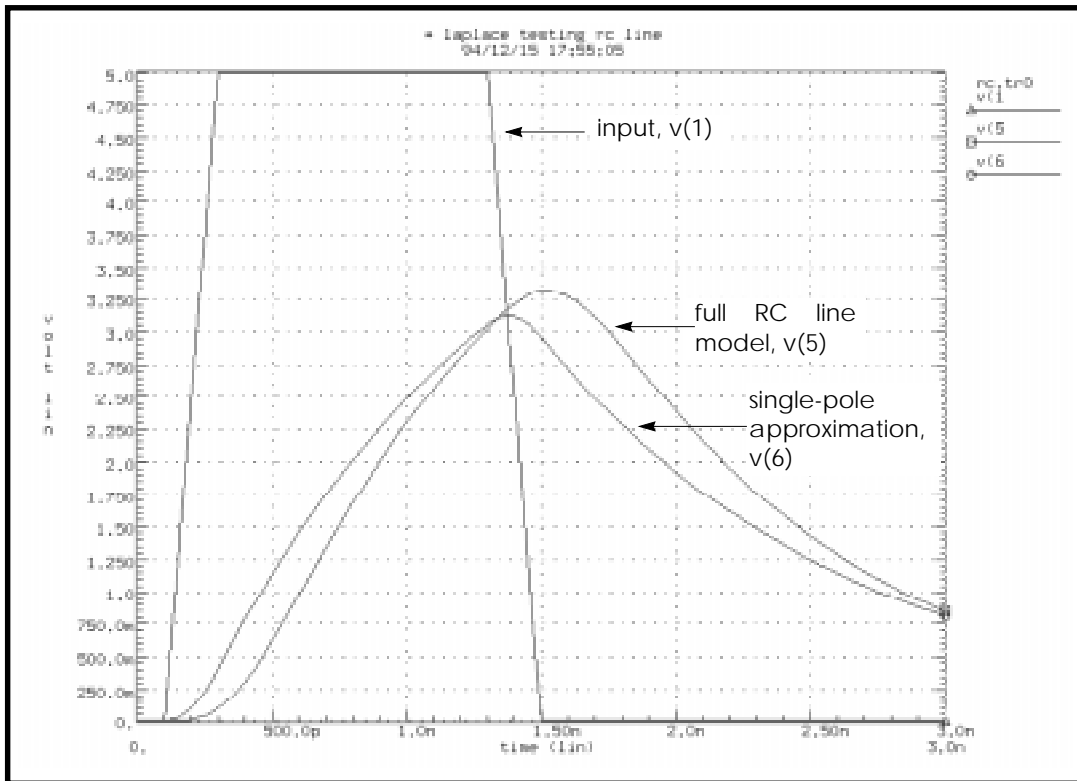
**Figure 26-10: Transient Response of the RC Line and Single-Pole Approximation**

Notice in Figure 26-10: 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 26-11: 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].
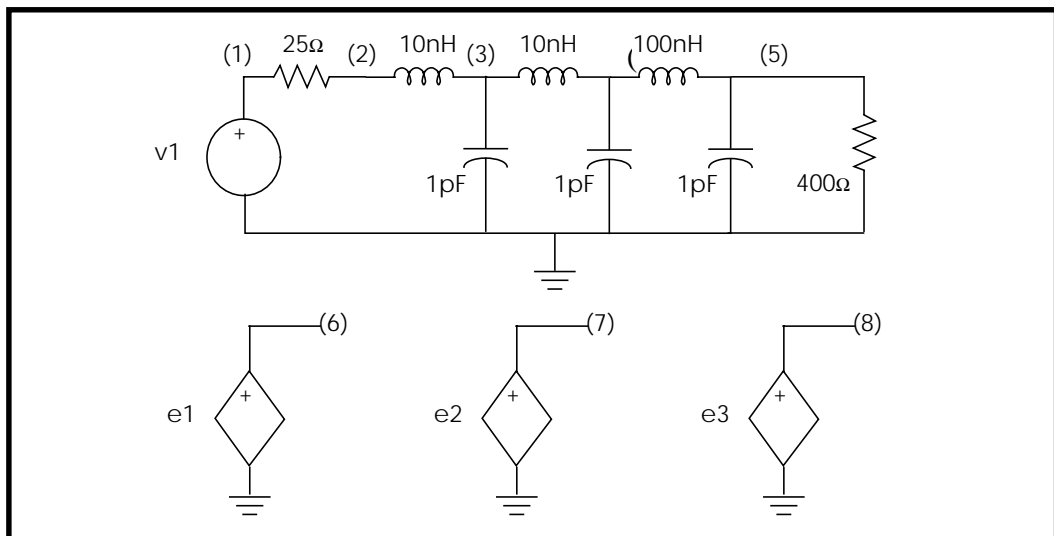


**Figure 26-11: Circuits for a Low-Loss Line**

### 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 26-12: 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.
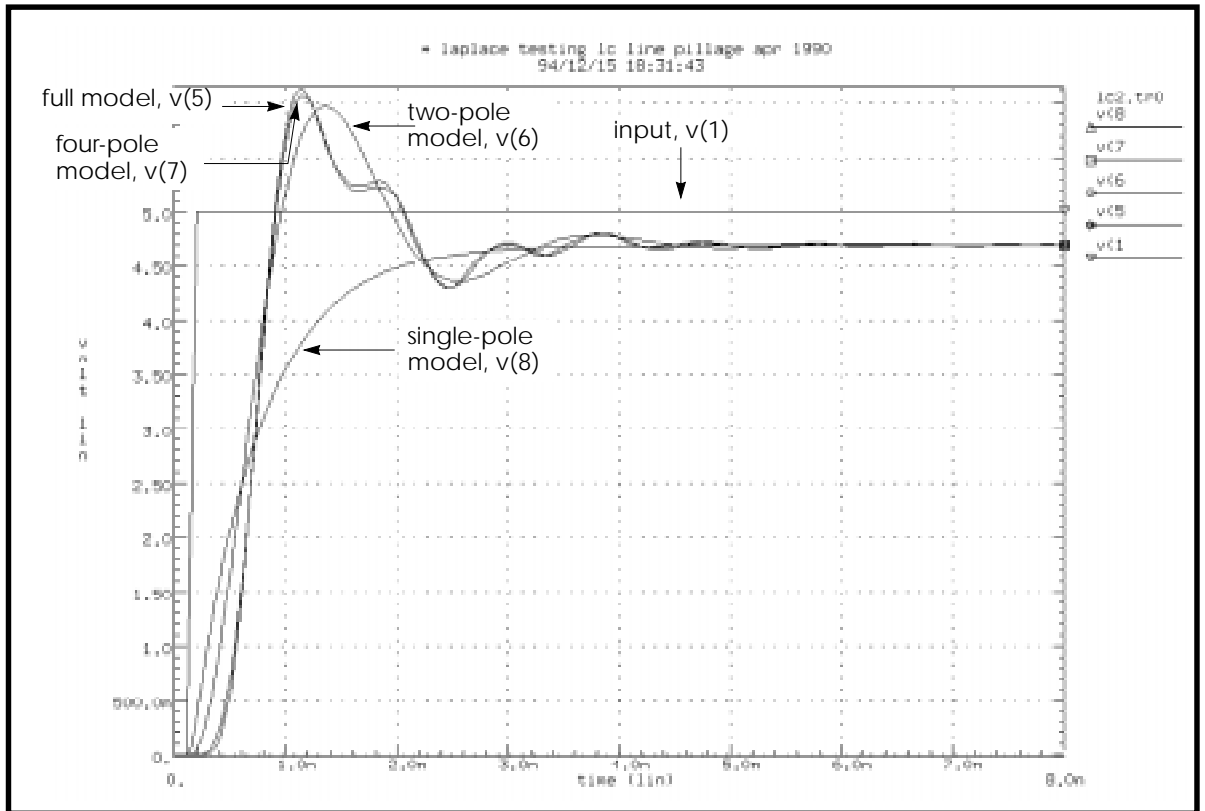
**Figure 26-12: – Transient Response of the Low-Loss Line**

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 an 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 26-13:, compare the response of the line using a Y-parameter model and a single-pole model.
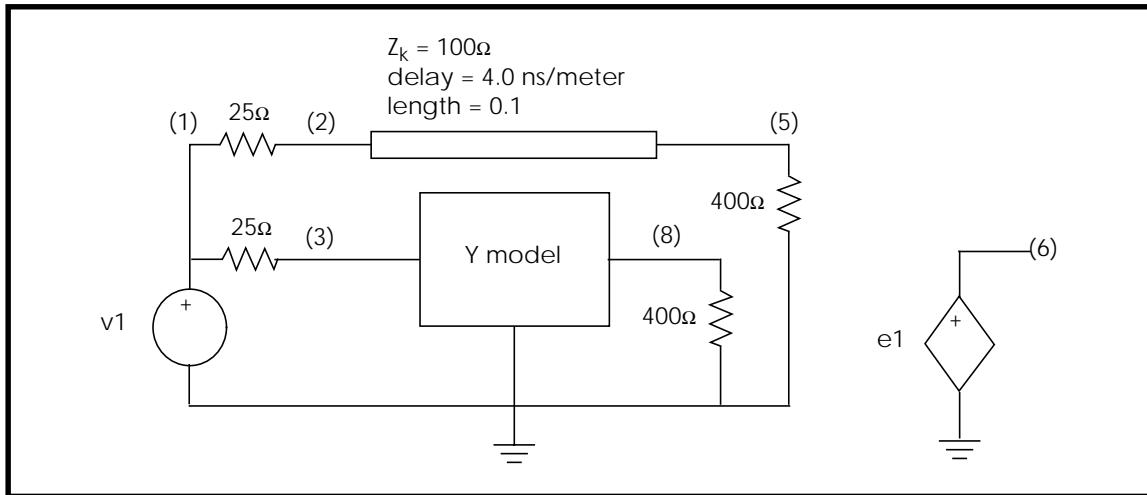


**Figure 26-13: Line and Y-Parameter Modeling**

The voltage and current definitions for a Y-parameter model are shown in Figure 26-14:.
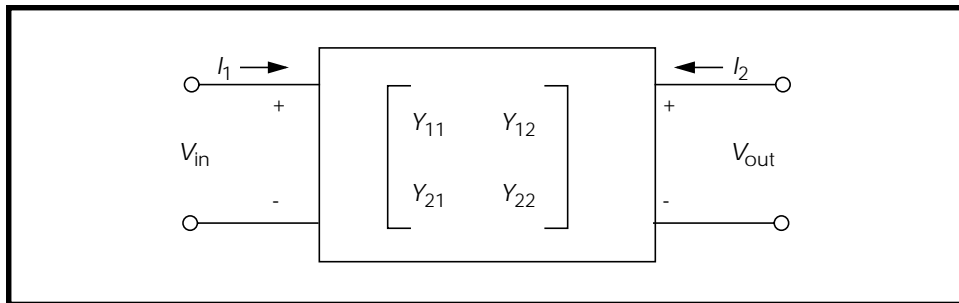


**Figure 26-14: – Y Matrix for the Two-Port Network**

The general network in Figure 26-14: 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 26-15:. Note that the circuit is essentially composed of G elements.
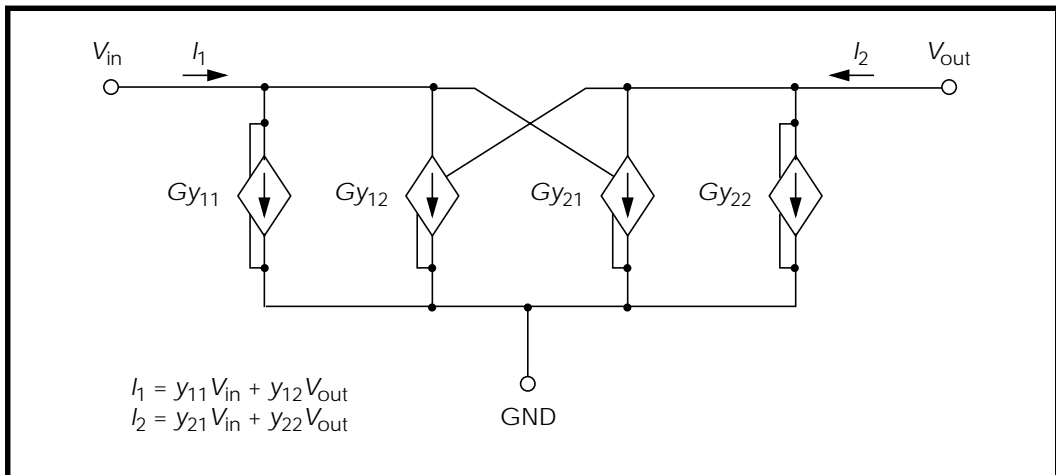


**Figure 26-15: – Schematic for the Y-Parameter Network**

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.

$$Y = \frac{1}{Z_o} \cdot \begin{bmatrix} \coth(p) & -\mathrm{csch}(p) \\ -\mathrm{csch}(p) & \coth(p) \end{bmatrix},$$

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:

$$coth(p) \rightarrow \frac{\left(1 + \frac{2}{5} \cdot p^2\right)}{\left(p + \frac{1}{15} \cdot p^3\right)}$$

$$csch(p) \rightarrow \frac{\left(1 - \frac{1}{20} \cdot p^2\right)}{\left(p + \frac{7}{60} \cdot p^3\right)}$$

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 $Z_O$ ($Z_o = \sqrt{L/C}$) 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 26-16: 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.
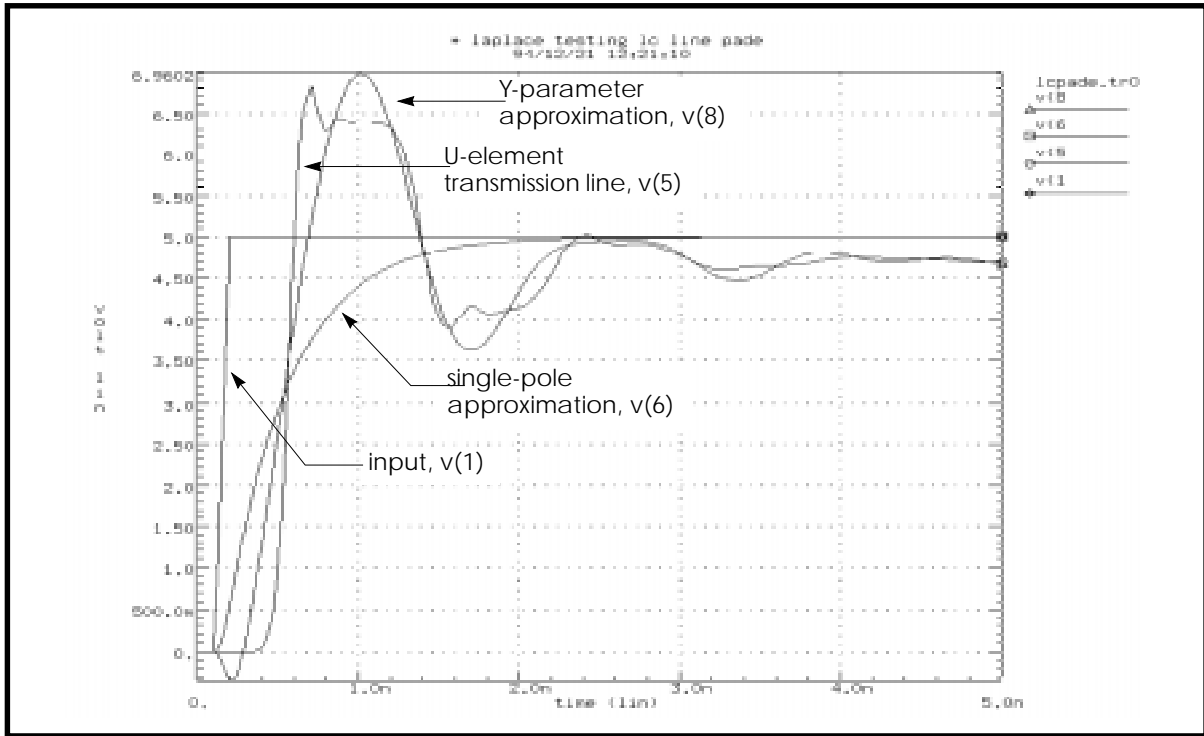
**Figure 26-16: – Transient Response of the Y-Parameter Line Model**

## 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" on page 26-42. For larger circuits, the computation time saving can be much higher.

The input listings for each model type are shown below. Figures 26-17 and 26-18 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**
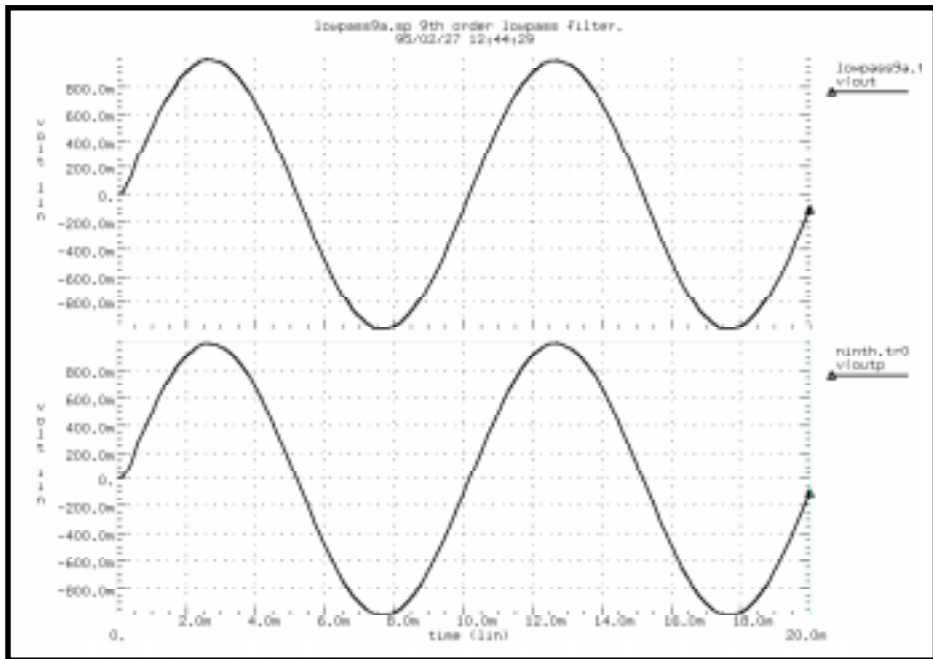


**Figure 26-17: Transient Responses of the Circuit and Pole/Zero Models**
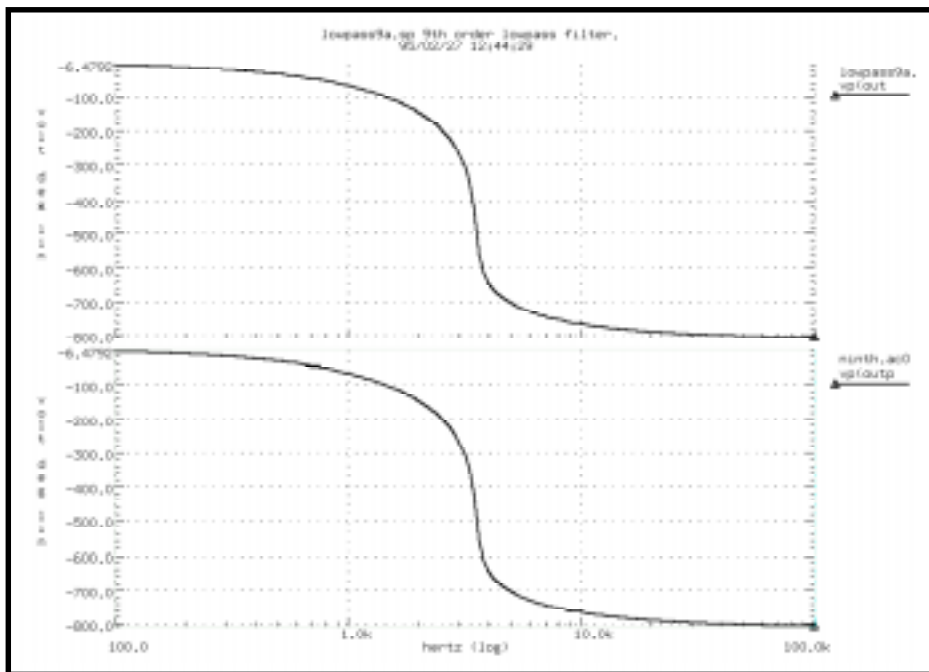
**Figure 26-18: AC Analysis Responses of the Circuit and Pole/Zero Models**

# Modeling Switched Capacitor Filters

This section describes how to create a model

## 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.
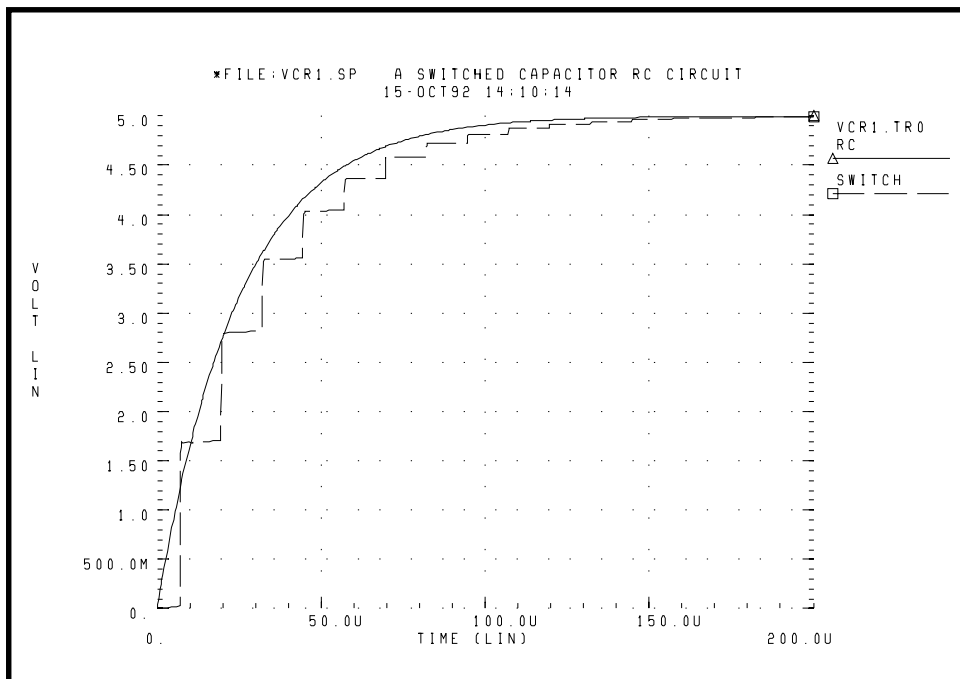
**Figure 26-19: VCR1.SP Switched Capacitor RC Circuit**

## 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 - Fifth Order

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].
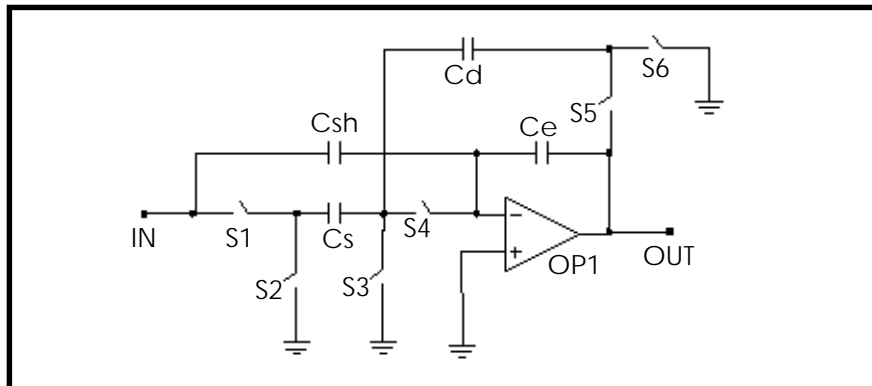


**Figure 26-20: Linear Section**
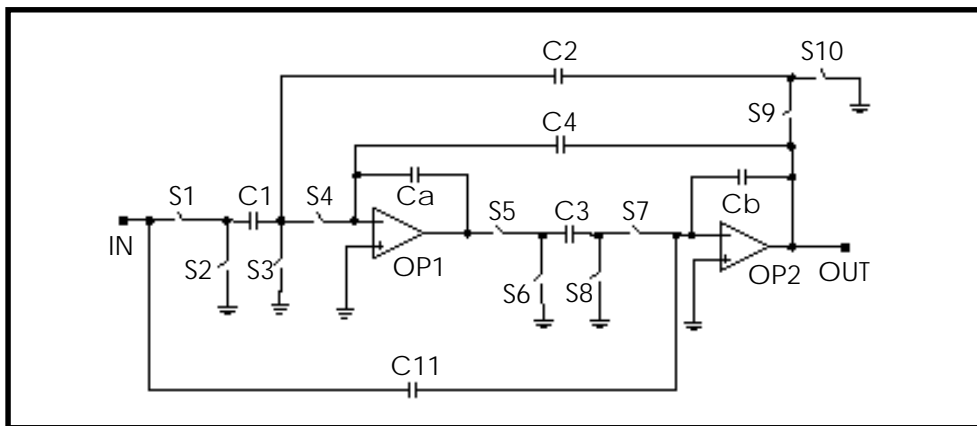
**Figure 26-21: High_Q Biquad Section**
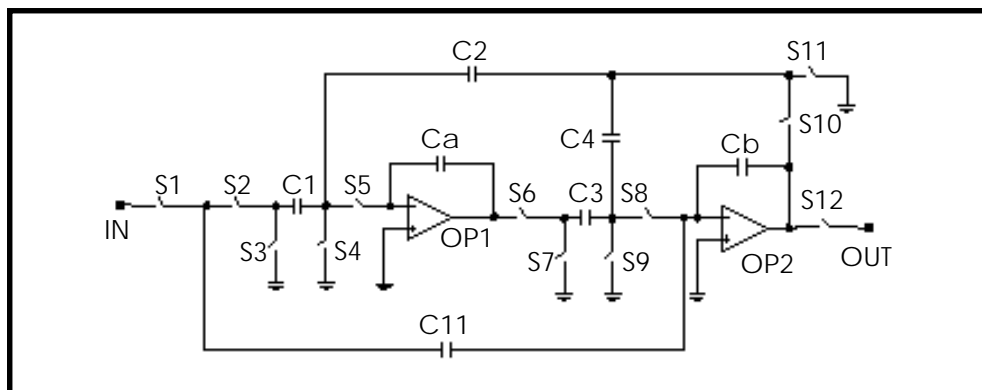


**Figure 26-22: Low_Q Biquad Section**

## Star-Hspice Input File for 5th Order 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
```
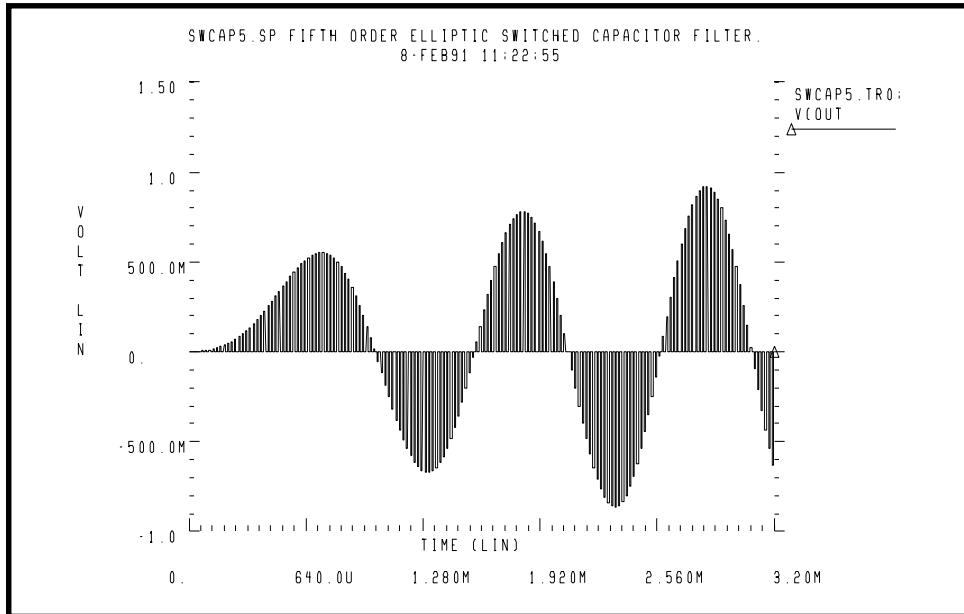


**Figure 26-23: Response to 1 kHz Sinusoidal Input**
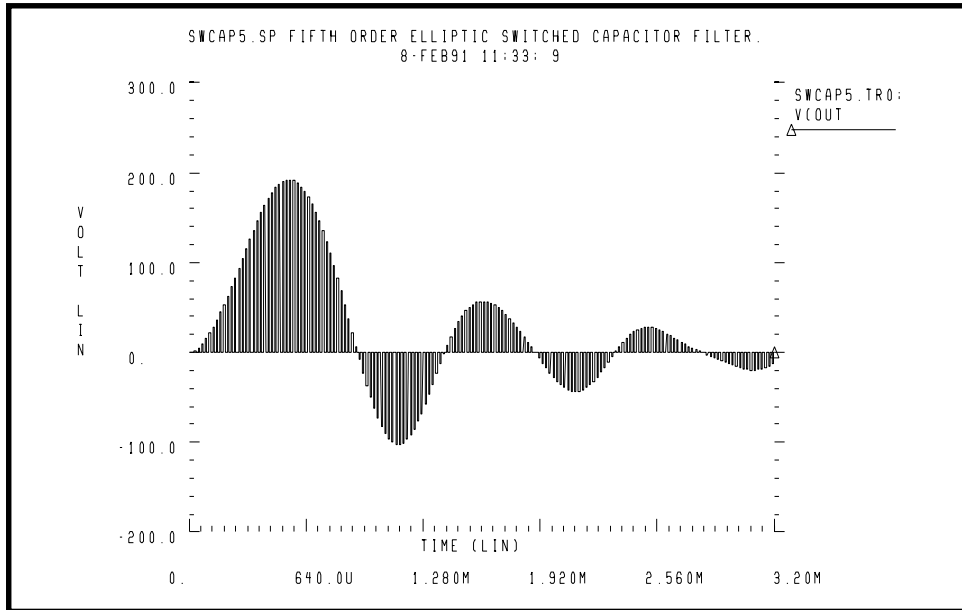
**Figure 26-24: Response to 2 kHz Sinusoidal Input**

## 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 27

# 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 Chapter 11, "Optimizing Performance".

The following topics are covered in this chapter:

# 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 27-1: 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.
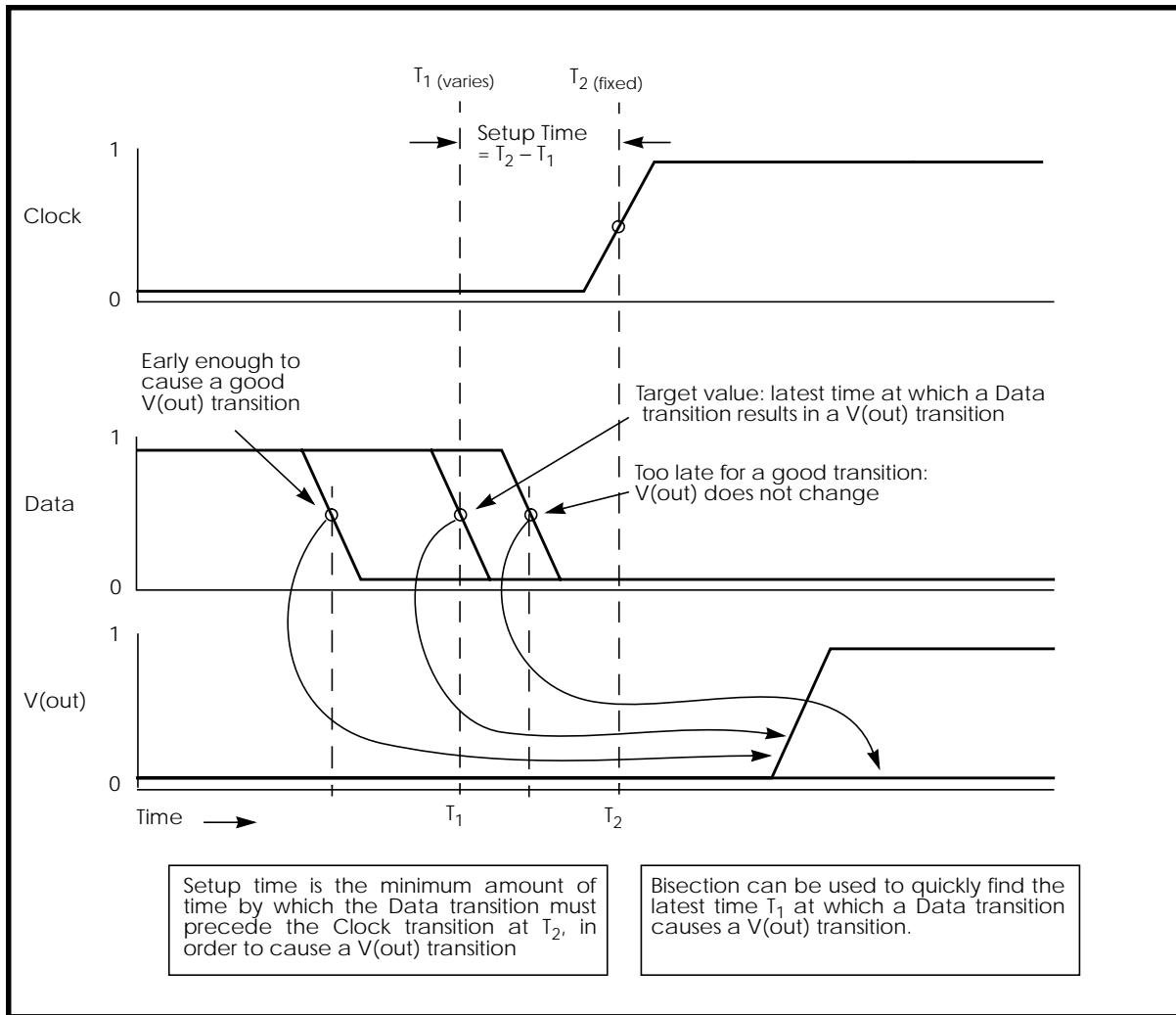
**Figure 27-1: Determining Setup Time with Bisection Violation Analysis**

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 which 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" on page -6). 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 27-1:) 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  |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 27-2: 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 27-1. 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 $\geq$ 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 27-2:).

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 Chapter 11, "Optimizing Performance" 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>
```

# Example 1 – 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 27-2 and Figure 27-3 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
```

Example 1 – Setup Time Analysis                    Timing Analysis Using Bisection

```
*
.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
```

## Results

The top plot in Figure 27-3: shows the relationship between the clock and data pulses that determine the setup time. The bottom plot shows the output transition.

*Example 1 – Setup Time Analysis*                    *Timing Analysis Using Bisection*
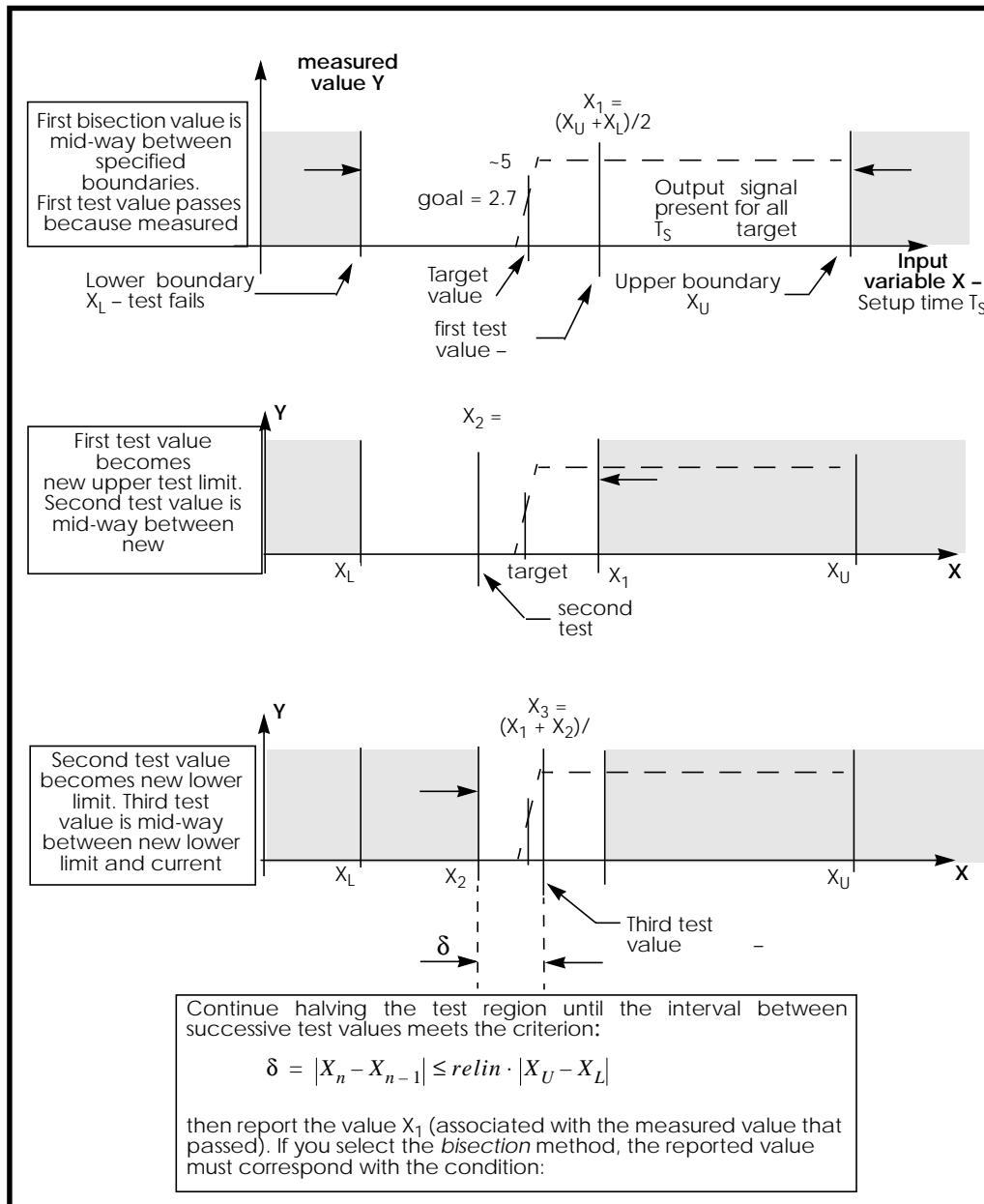


**Figure 27-2: – Bisection Example for Three Iterations**
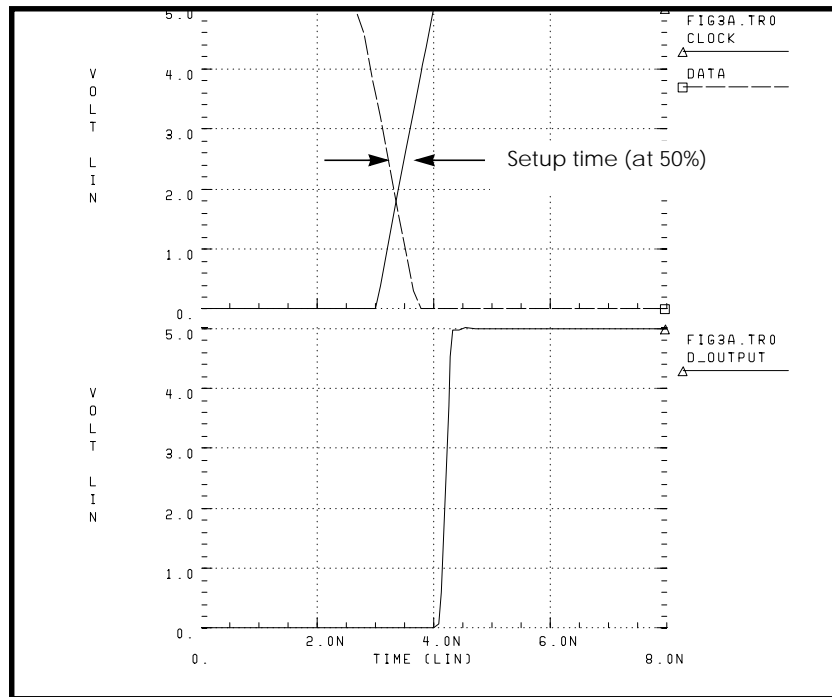
**Figure 27-3: – Transition at Minimum Setup Time**

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 27-4: shows examples of early and late data transitions, as well as the transition at the minimum setup time. The bottom plot shows how the

*Example 1 – Setup Time Analysis*                    *Timing Analysis Using Bisection*

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
```
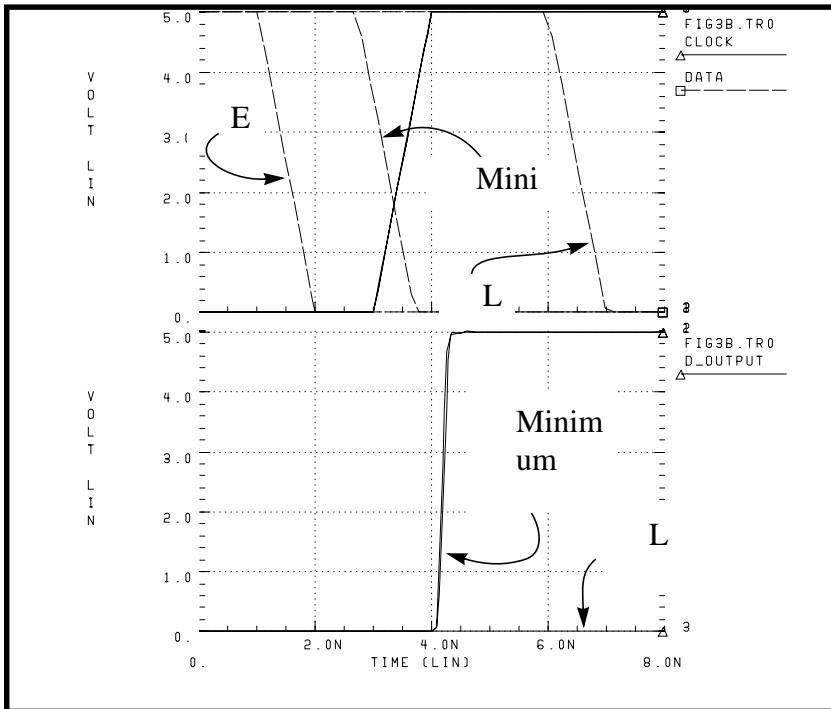


**Figure 27-4: Early, Minimum, and Late Setup and Hold Times**

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
```

# Example 2 – 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 27-5:.

## Input listing

```
File: $installdir/demo/bisect/inv_a.sp
$ Inv_a.sp testing bisectional search, cload=10p & 20p
*
* Parameters
.PARAM Cloud   =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 27-5: shows the results of the pass/fail search for two different capacitive loads.
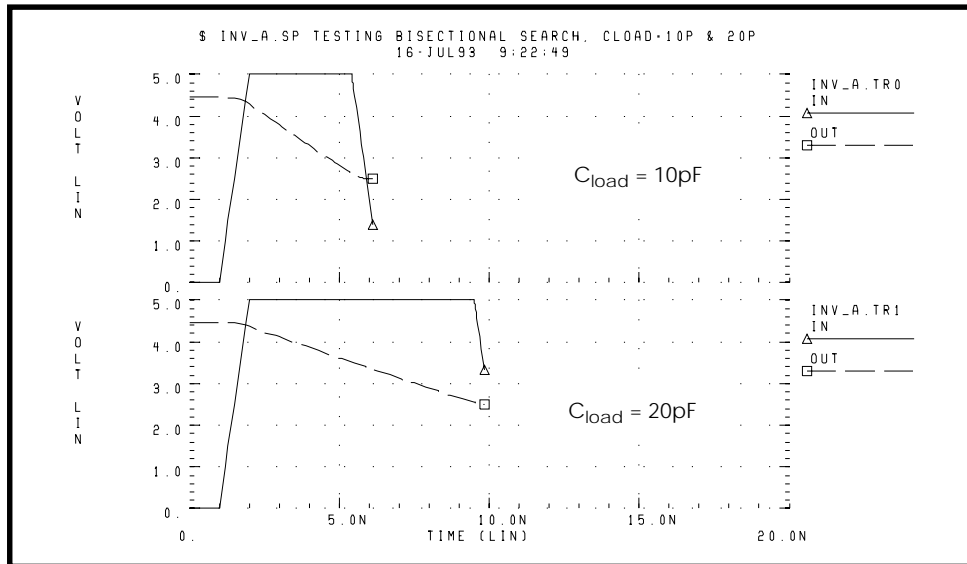


**Figure 27-5: Results of Bisectional Pass/Fail Search**

*Example 2 – Minimum Pulse Width Analysis*                    *Timing Analysis Using Bisection*

*Avant!*

# Chapter 28

# 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 Chapter 2, "Getting Started." 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 28-1: shows the organization of subcircuits and models in libraries used in this example.
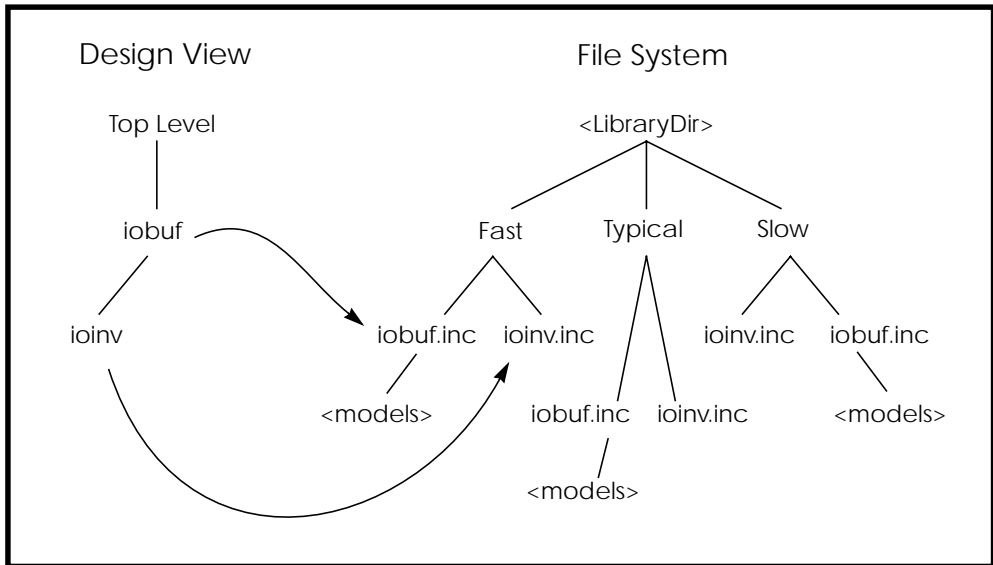
**Figure 28-1: Encrypted Library Structure**

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 22-9 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!*

# Chapter 29

# Running Demos

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 theTemperature 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 *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.
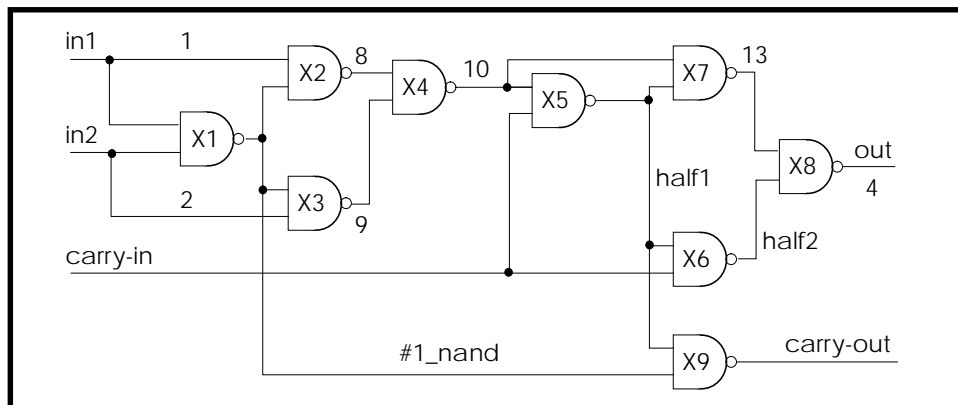


**Figure 29-1: – One-bit Adder Subcircuit**
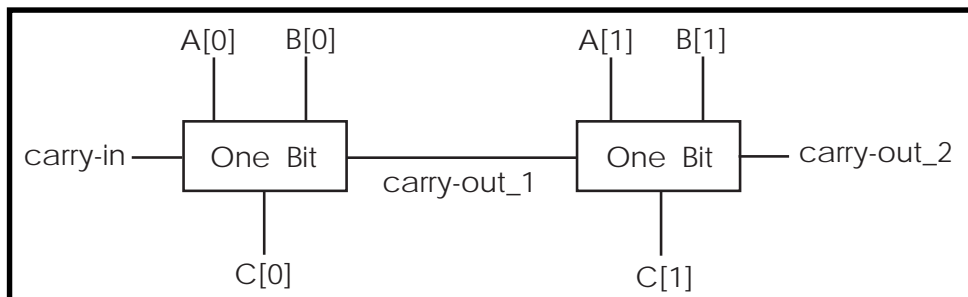


**Figure 29-2: – Two-bit Adder Circuit**
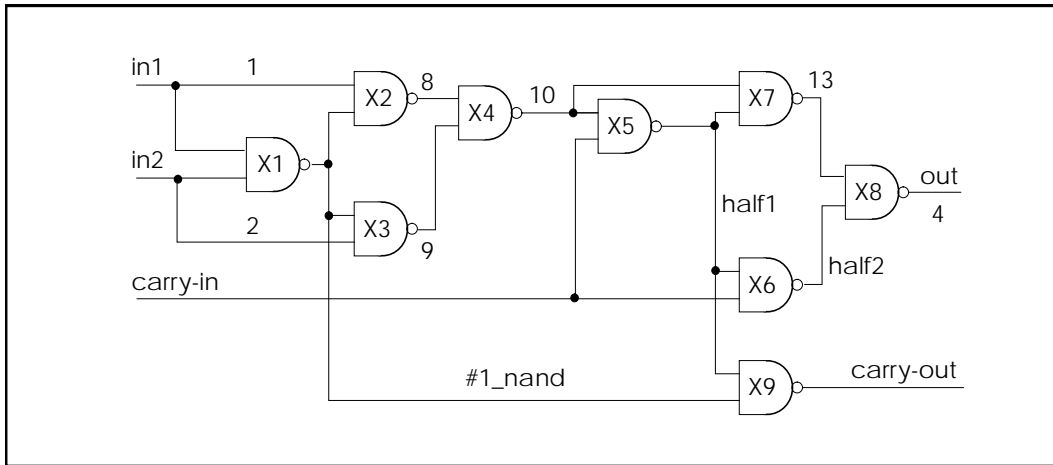
**Figure 29-3: – 1-bit NAND Gate Binary Adder**

## 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, *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) |
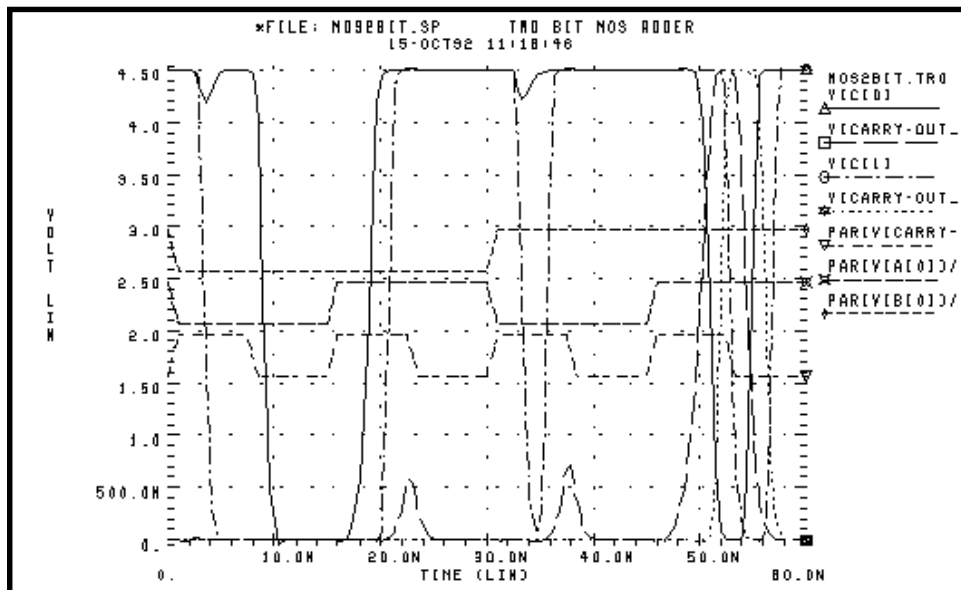
**Figure 29-4: – MOS IDS Plot**
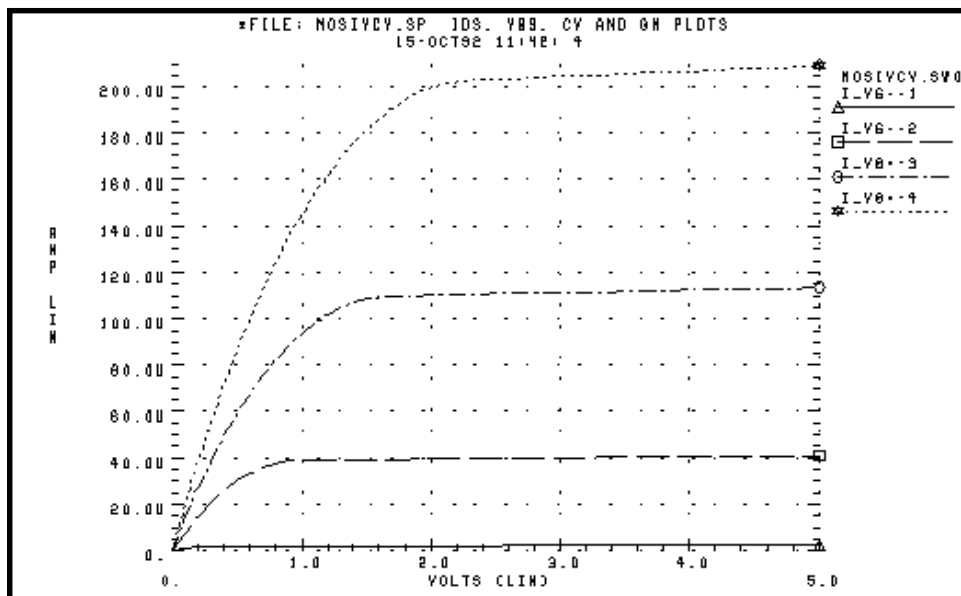


**Figure 29-5: – MOS VGS Plot**
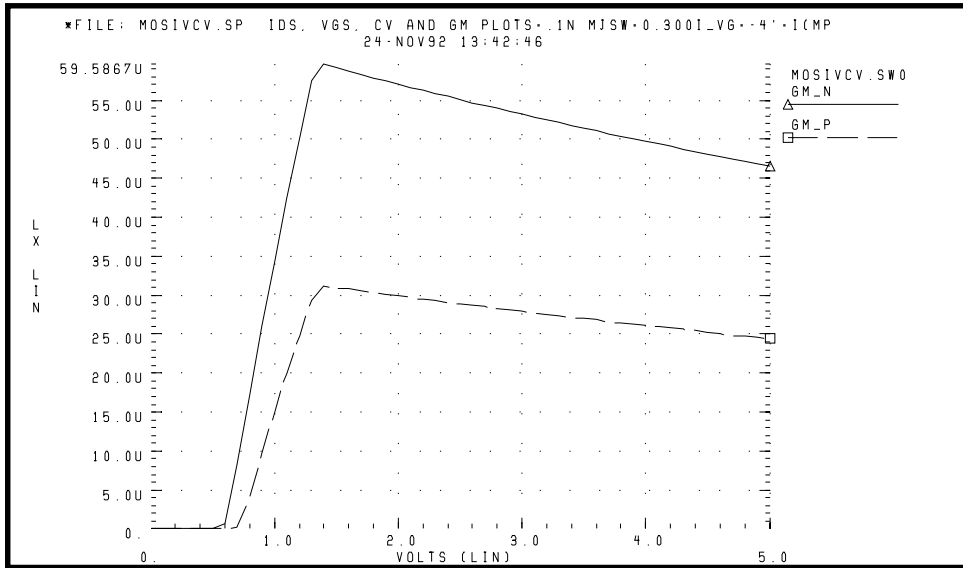
**Figure 29-6: MOS GM Plot**
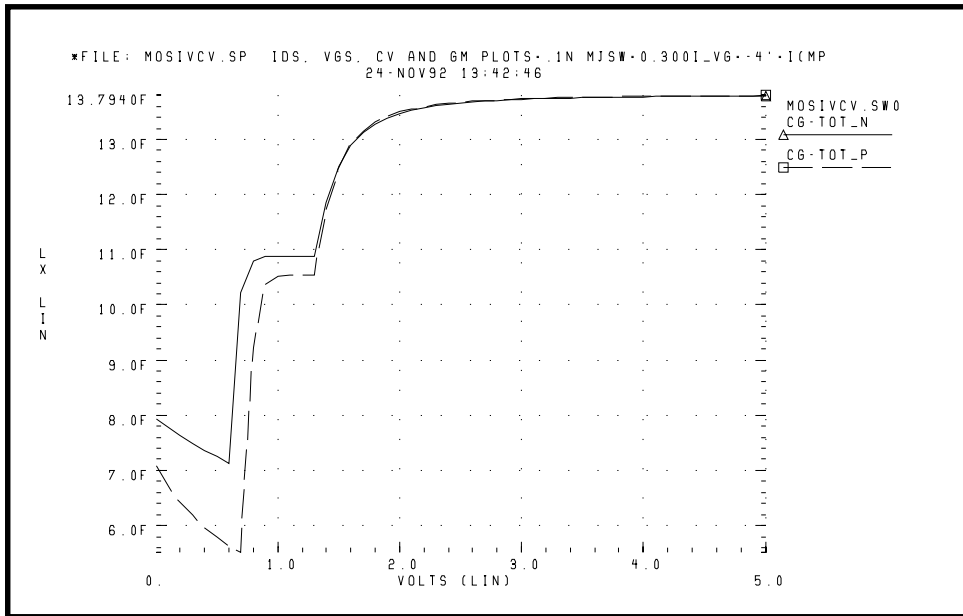


**Figure 29-7: MOS C-V Plot**

## 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 *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
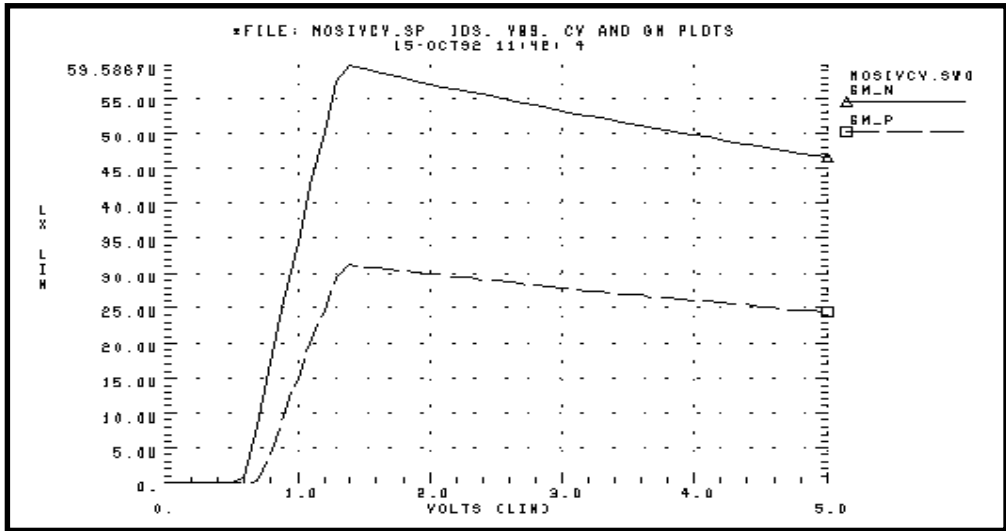
**Figure 29-8: – Noise Bounce**
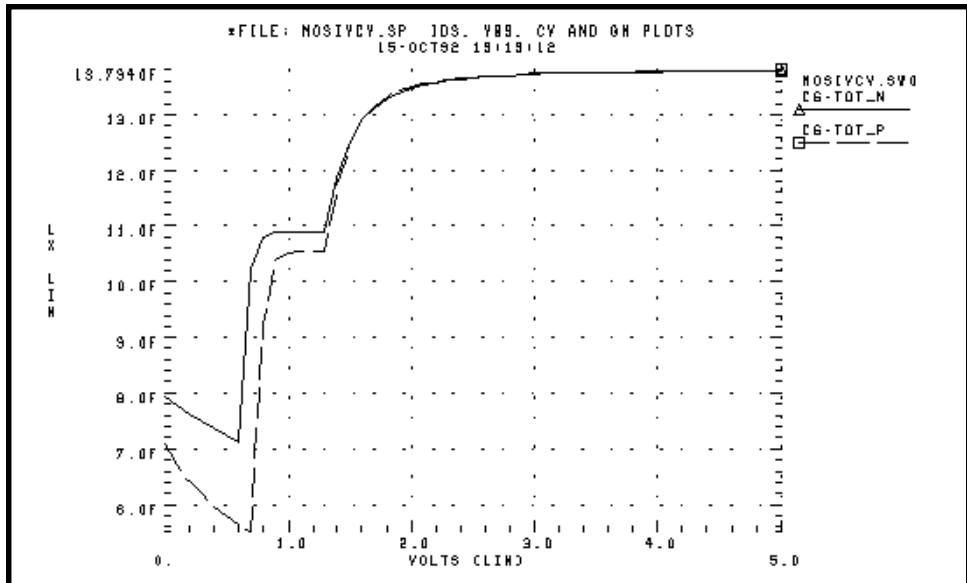


**Figure 29-9: Asic1.sp Demo Local Supply Voltage**
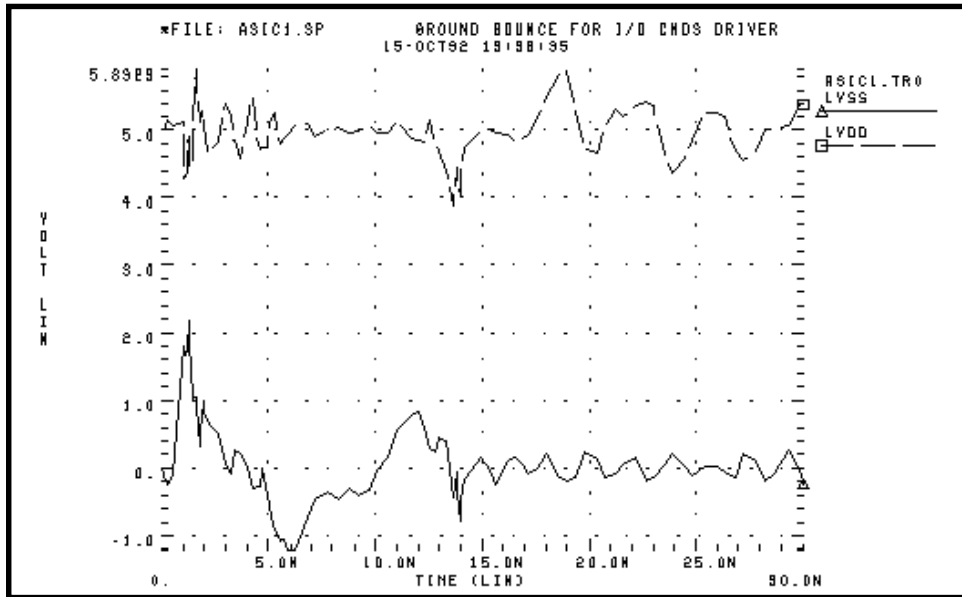
**Figure 29-10: Asic1.sp Demo Local Supply Current**
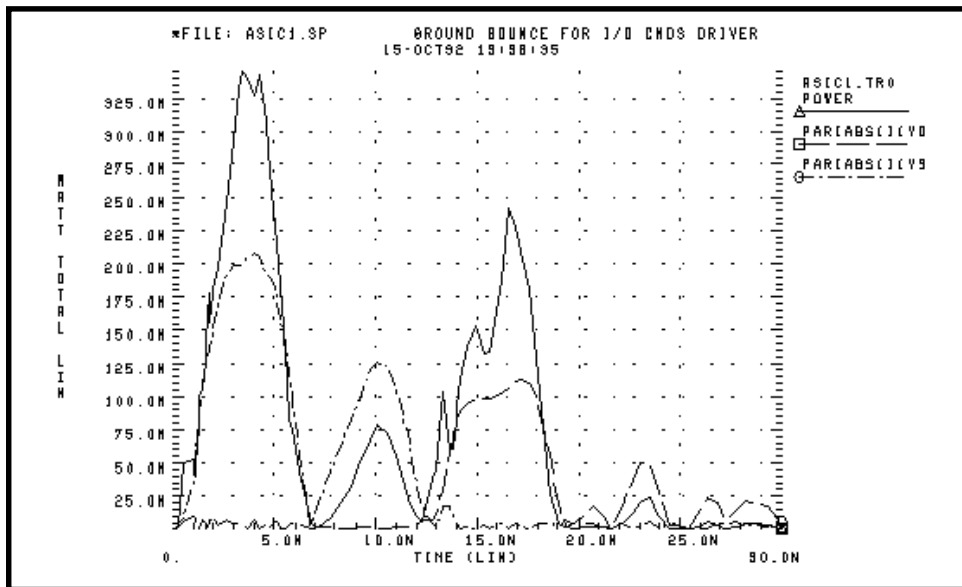


**Figure 29-11: Asic1.sp Demo Input and Output Signals**

## 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 theTemperature 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 |
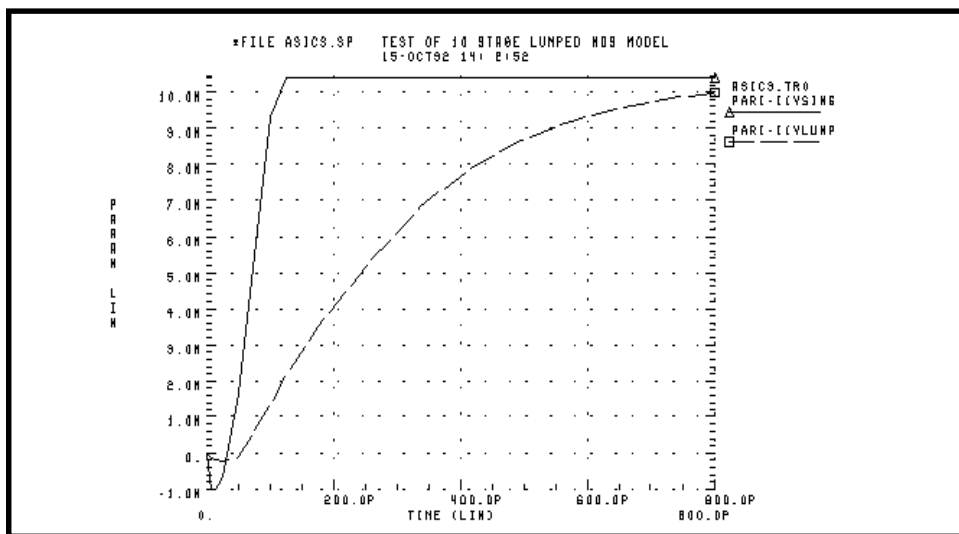


**Figure 29-12: – T2N2222 Optimization**

### 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

$$Rpoly = \frac{W}{L} \cdot 40$$

$$poly = \frac{Esio \cdot nsi}{tox} \cdot L \cdot W$$

$$Rpoly = \frac{100}{5} \cdot 40 = 800, \quad Co = \frac{3.9 \cdot 8.86}{800} \cdot 100 \cdot 5 = 215\ fF$$

$$RC = 138\ ps$$

For a transistor with

> channel width = 100 microns
>
> channel length = 1.2 microns
>
> gate oxide thickness = 250 Angstroms

$$Rpoly = \frac{channel\ width}{channel\ length} \cdot 40$$

$$Co \ = \ \frac{3.9 \cdot 8.86}{Tox} \cdot channel \ \ width \cdot channel \ \ length$$

*RC = 546 ps*

You can model the RC delay introduced in modern CMOS technologies by using a nine-stage ladder model.
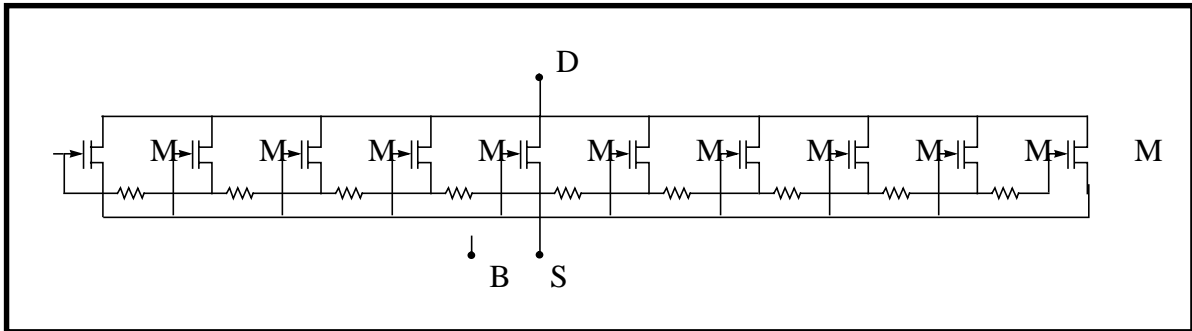


**Figure 29-13: Nine-stage Ladder Model**

In this example, the nine-stage ladder model was entered into an 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 9-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
```
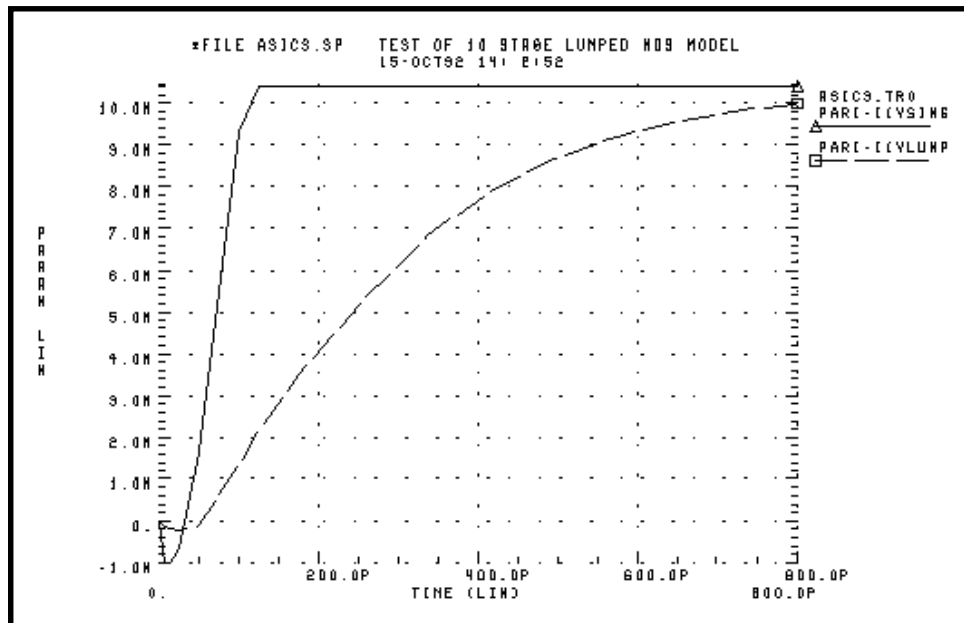


**Figure 29-14: – Asic3 Single versus Lumped Model**

# 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 |

# Index

# N

narrow width effect 17-14

National Semiconductor model 16-5

natural frequency 25-2

natural log function 10-6

n-channel
    JFET models 3-65
    MOSFET models 3-65

ND model parameter 22-17

NDIM 5-20

negative conductance, logging 3-49

nested library calls 3-63

.NET statement 9-19

netlists 3-29
    encrypting 29-1
    flat 3-29

networks
    analysis 9-19
    filenames 3-43
    output 9-20

NEWTOL option 6-18

NL
    element parameter 22-11
    model parameter 22-26

NLAY model parameter 22-19, 22-26

NODE option 3-51
    suppressing listing 3-67

nodes
    connection requirements 3-31
    cross-reference table 3-51
    floating supply 3-31
    global versus local 3-25
    internal 3-31

MOSFET substrate 3-31

names 3-29, 3-31, 3-33, 23-6, 15
    ground node 3-31
    period in 3-30
    subcircuits 3-30, 3-31
    zeros in 3-33

numbers 3-29, 3-31

phase or magnitude difference 9-9

printing 3-51

shorted 6-33

terminators 3-31

voltages, encrypting 29-1

NODESET keyword 6-11

.NODESET statement 6-3, 6-15, 25-3
    balancing input nodes 23-83
    DC operating point initialization 6-7
    from .SAVE 6-11

node-to-element list 6-19

NOELCK option 3-52

noise
    calculations 9-17
    coupled line noise 20-26
    equations
        BJT 14-45
        JFET and MESFET 15-39
    folding 9-18
    input 4-15
    JFET and MESFET
        equations 15-39
        summary printout 15-41
    MOSFET 16-104
    numerical 7-8
    output 4-15, 9-17
    parameters
        BJT 14-26

## S

# U

# W