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

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

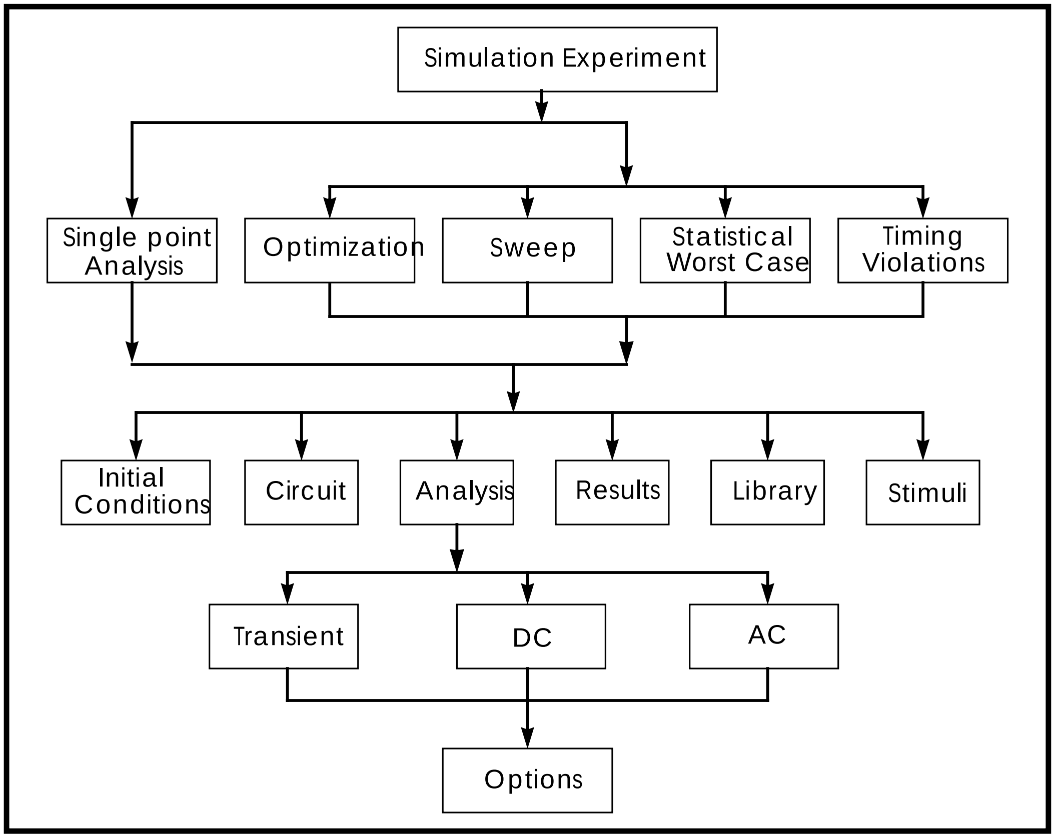

Figure 3-1: Simulation Program Structure

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.

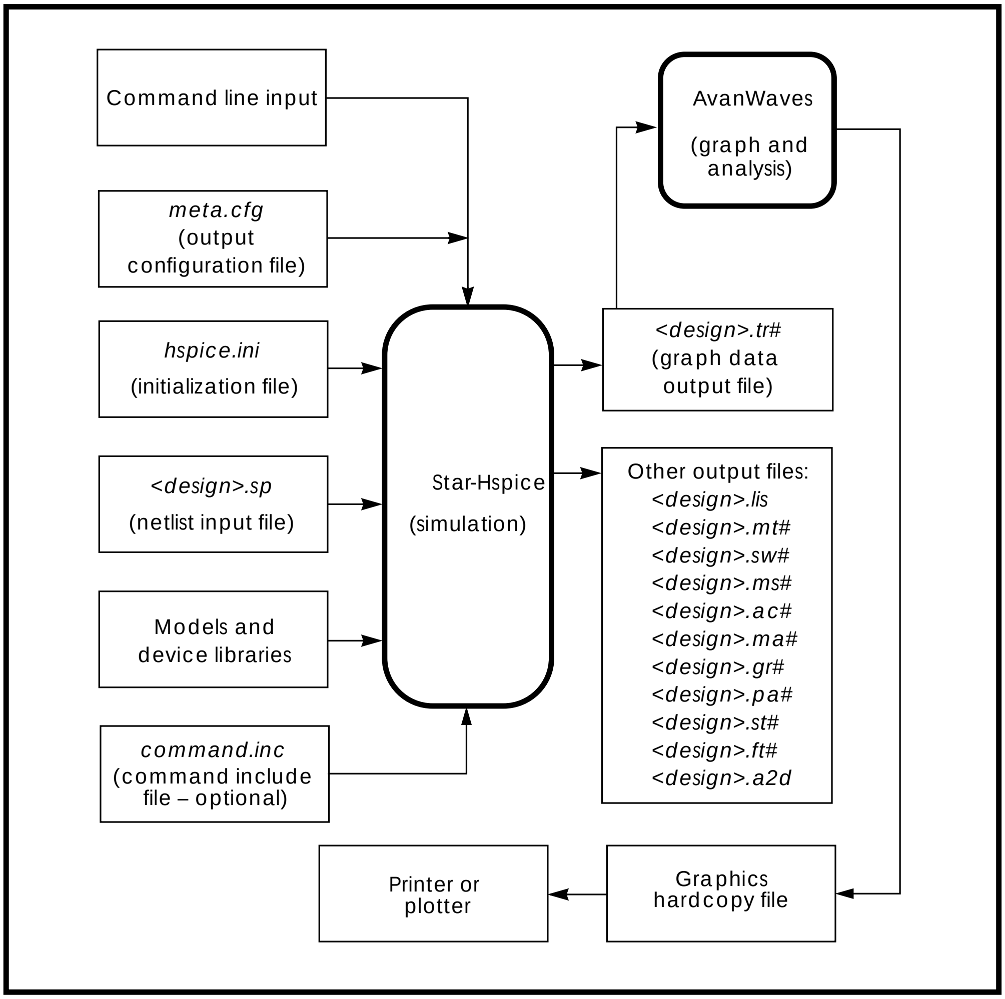

Figure 3-2: Overview of Star-Hspice Data Flow

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.

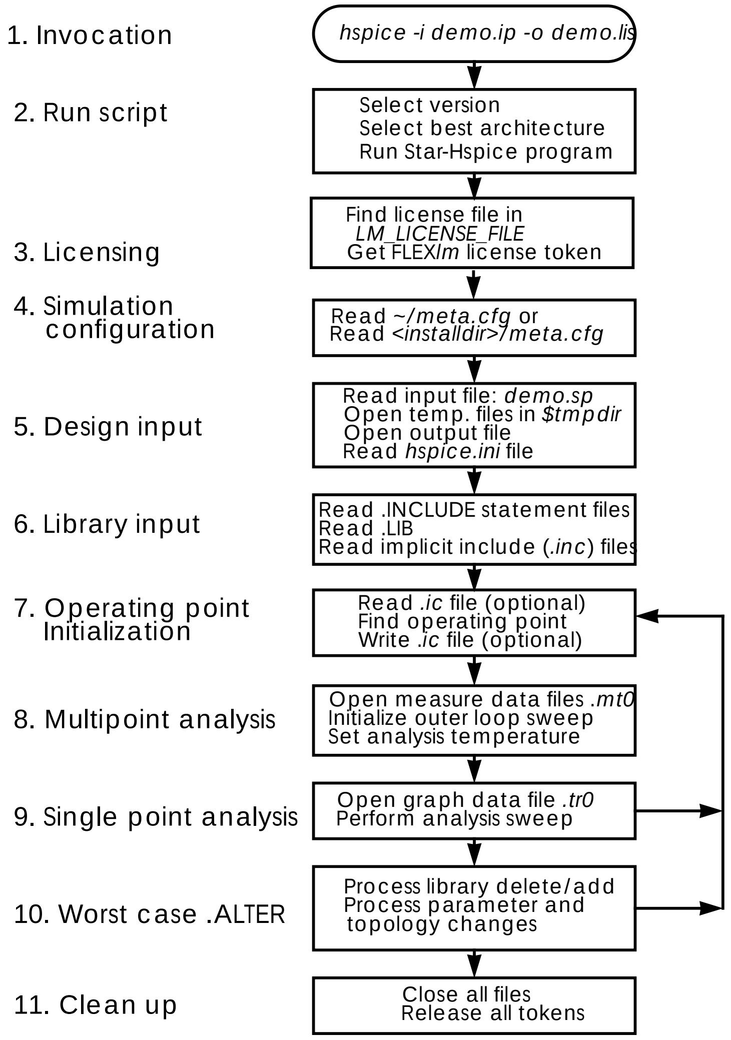

Figure 3-3: Star-Hspice Simulation Process

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: ❍ sun4, sol4 (SparcStation, Ultra) ❍ pa (HP 700/800/9000) ❍ alpha (DEC ALPHA) ❍ rs (IBM RS6000) ❍ sgi (SGI) ❍ cray (Cray) ❍ i386 (Windows 95/NT)
-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

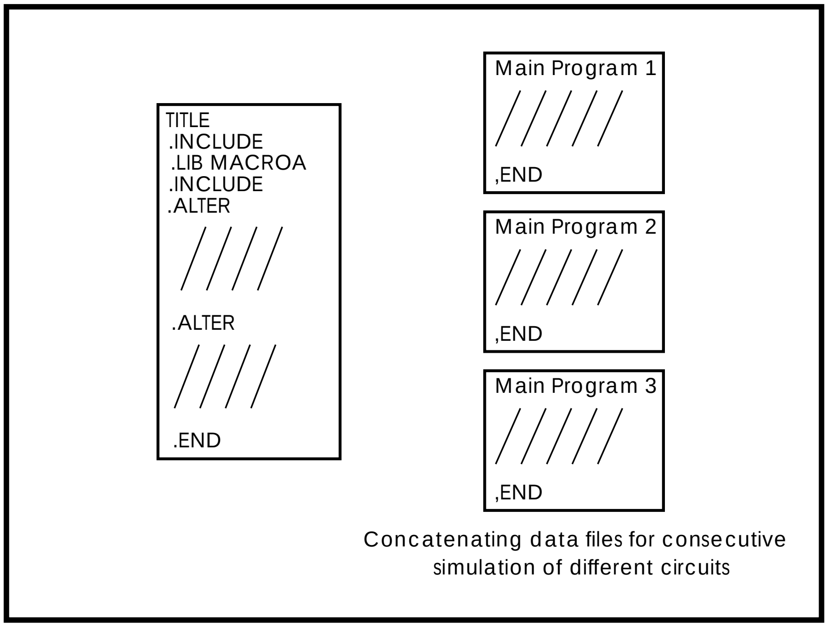

Figure 3-4: Input Netlist File Structures

.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:.

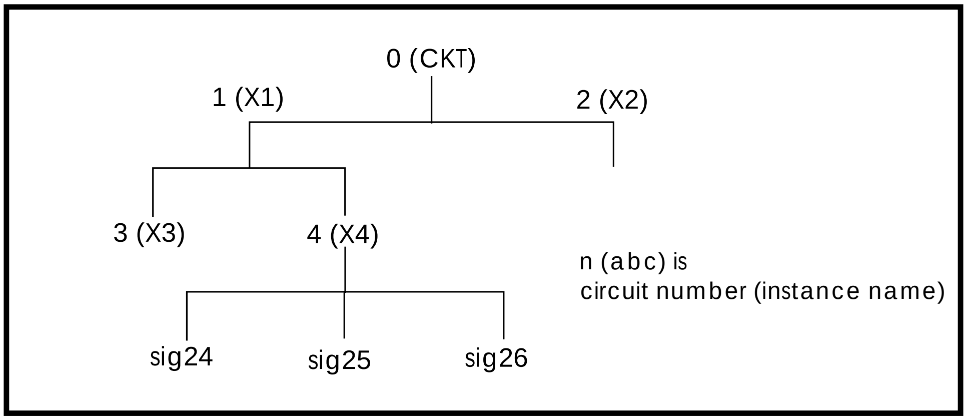

Figure 3-5: Subcircuit Calling Tree with Circuit Numbers and Instance Names

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.

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.

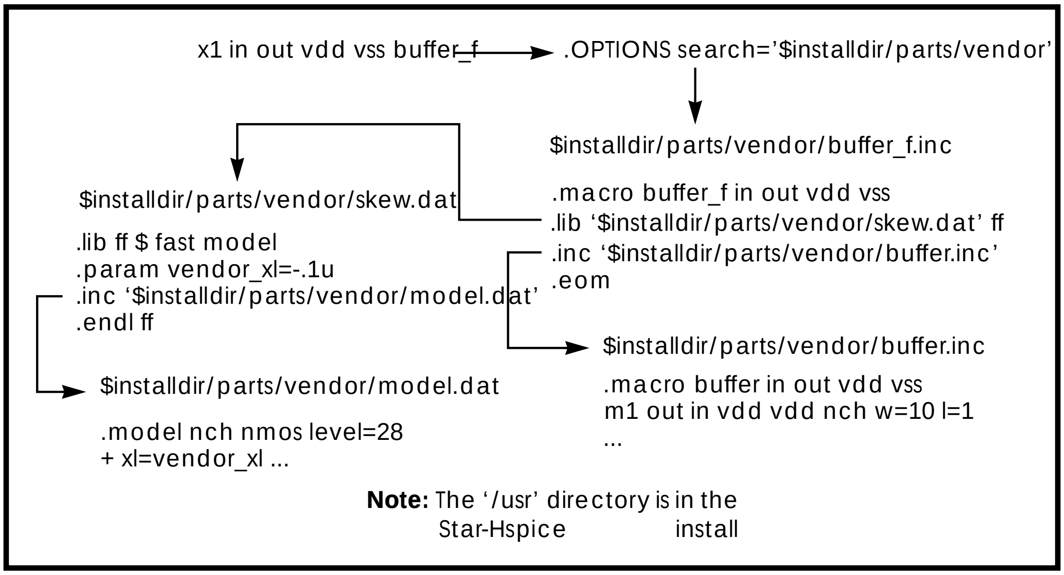
x1 in out vdd vss buffer_f
.OPTIONS search='$installdir/parts/vendor'
$installdir/parts/vendor/buffer_f.inc
.macro buffer_f in out vdd vss
.lib '$installdir/parts/vendor/skew.dat' ff
.inc '$installdir/parts/vendor/buffer.inc'
.eom
$installdir/parts/vendor/skew.dat
.lib ff $ fast model
.param vendor_xl=-.1u
.inc '$installdir/parts/vendor/model.dat'
.endl ff
$installdir/parts/vendor/buffer.inc
.macro buffer in out vdd vss
m1 out in vdd vdd nch w=10 l=1
...
$installdir/parts/vendor/model.dat
.model nch nmos level=28
+ xl=vendor_xl ...
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†	10.00u†	10.00u†
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††	500.00m††	750.00m††
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

† Set resmin to 0.01 ohm for backward compatibility with 93A or earlier.
†† 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†
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.