HSPICE® Quick Start Guide

Version R-2020.12, December 2020

SYNOPSYS®

Copyright and Proprietary Information Notice

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Free and Open-Source Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

www.synopsys.com

Feedback

Contents

About This Manual

This guide provides information to quickly get you started with using the Synopsys HSPICE® tool for simulation and analysis of your circuit designs.

It provides an overview of the HSPICE simulation flow and information on the installation, invocation, and documentation of the HSPICE tool. It also provides inks to the detailed documentation for the analyses, elements, and models supported by the HSPICE tool.

This preface includes the following sections:

- New in This Release *(on page 5)*
- Related Products, Publications, and Trademarks *(on page 5)*
- Conventions *(on page 6)*
- Customer Support *(on page 7)*

New in This Release

Information about new features, enhancements, and changes, known limitations, and resolved Synopsys Technical Action Requests (STARs) is available in the HSPICE Release Notes on the SolvNetPlus site.

Related Products, Publications, and Trademarks

For additional information about the HSPICE simulator, see the documentation on the Synopsys SolvNetPlus support site at the following address:

https://solvnetplus.synopsys.com

You might also want to see the documentation for the following related Synopsys products:

Cadence® Virtuoso® Analog Design Environment
Synopsys Custom WaveView™
Synopsys SolvNetPlus support site

Conventions

The following conventions are used in Synopsys documentation.

Convention	Description
`Courier`	Indicates syntax, such as `write_file`.
`Courier italic`	Indicates a user-defined value in syntax, such as `write_file` *`design_list`*
`Courier bold`	Indicates user input—text you type verbatim—in examples, such as `prompt>` **`write_file top`**
`Purple`	- Within an example, indicates information of special interest. - Within a command-syntax section, indicates a default, such as `include_enclosing =` **`true`** `| false`
[ ]	Denotes optional arguments in syntax, such as `write_file [-format` *`fmt`*`]`
...	Indicates that arguments can be repeated as many times as needed, such as *`pin1 pin2 ... pinN`*.
|	Indicates a choice among alternatives, such as `low | medium | high`
\	Indicates a continuation of a command line.
/	Indicates levels of directory structure.
Bold	Indicates a graphical user interface (GUI) element that has an action associated with it.
Edit > Copy	Indicates a path to a menu command, such as opening the **Edit** menu and choosing **Copy**.
Ctrl+C	Indicates a keyboard combination, such as holding down the Ctrl key and pressing C.

Customer Support

Customer support is available through SolvNetPlus.

Accessing SolvNetPlus

The SolvNetPlus site includes a knowledge base of technical articles and answers to frequently asked questions about Synopsys tools. The SolvNetPlus site also gives you access to a wide range of Synopsys online services including software downloads, documentation, and technical support.

To access the SolvNetPlus site, go to the following address:

https://solvnetplus.synopsys.com

If prompted, enter your user name and password. If you do not have a Synopsys user name and password, follow the instructions to sign up for an account.

If you need help using the SolvNetPlus site, click REGISTRATION HELP in the top-right menu bar.

Contacting Customer Support

To contact Customer Support, go to https://solvnetplus.synopsys.com.

1

HSPICE Overview

Provides an overview of the HSPICE simulation flow, analysis types, element statements, and supported semiconductor device models. Describes the HSPICE documentation set and how to access the documentation.

This chapter discusses the following topics:

- Simulation Flow *(on page 8)*
- Analysis Types *(on page 11)*
- Elements *(on page 14)*
- Models *(on page 15)*
- Demonstration Files *(on page 18)*
- Documentation *(on page 19)*
- HSPICE Application Notes *(on page 22)*

Simulation Flow

Figure 1 *(on page 9)* on page 9 illustrates the HSPICE simulation flow.

In an HSPICE simulation, the following sequence of events occur:

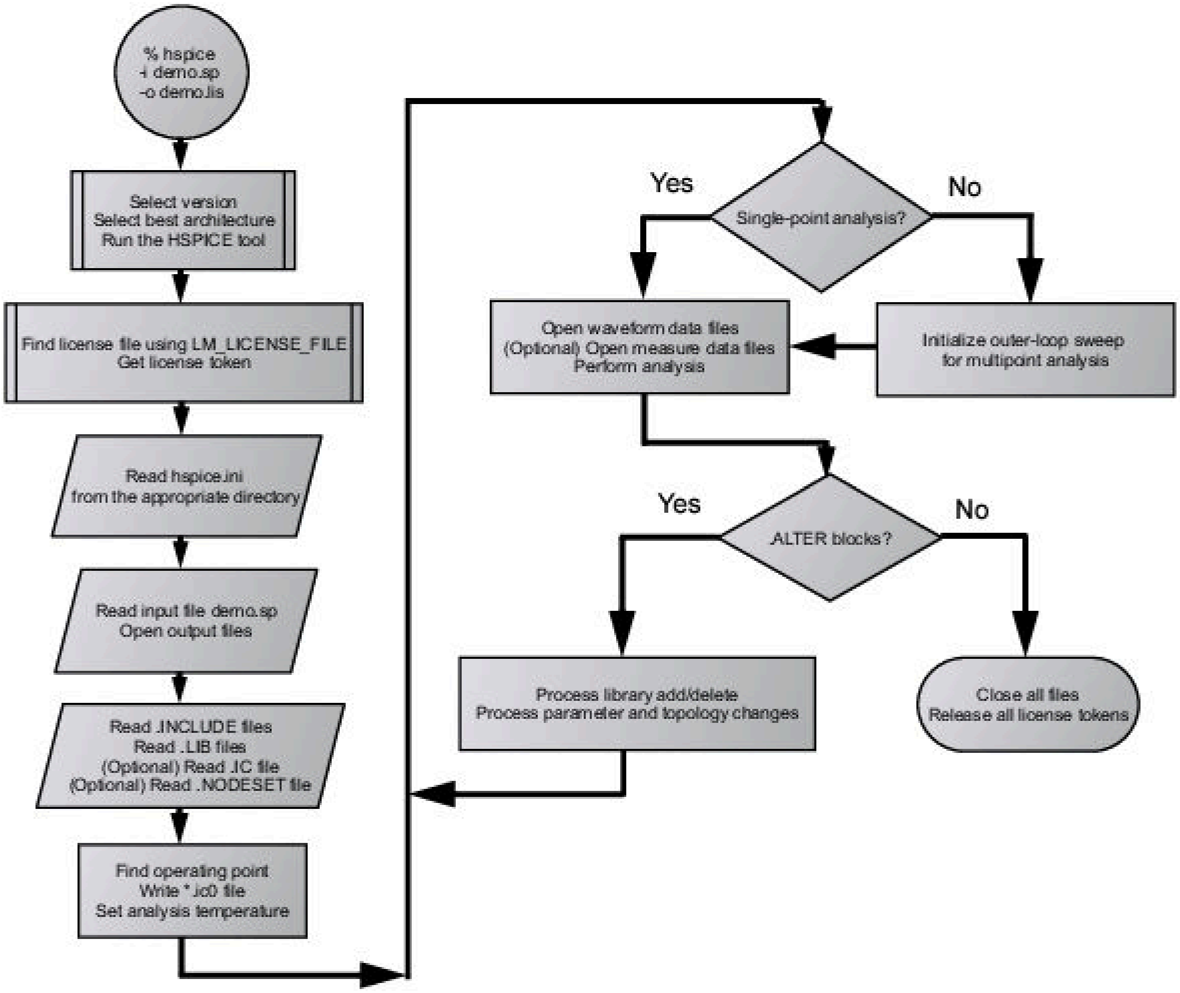
- Invocation:

  You invoke the HSPICE tool by entering the following command, for example, at the shell prompt:

  ```
  % hspice -i demo.sp -o demo.lis &
  ```

  This command invokes the HSPICE tool on the input netlist file `demo.sp` and directs the output listing to the file `demo.lis`. The `&` character at the end of the command invokes the HSPICE tool in the background, so that you can continue to use the command window and the keyboard while the HSPICE tool runs.

Figure 1 HSPICE Simulation Flow



- Script execution:

  The `hspice` command starts the HSPICE executable from the appropriate architecture (machine type) directory. The UNIX `run` script launches an HSPICE simulation. This procedure establishes the environment for the HSPICE executable. The script prompts for information such as the platform that you are using and the HSPICE version to run.

(The available versions of the HSPICE tool are pre-determined during the installation of the tool.)

- Licensing:

  The HSPICE tool supports the FLEXlm licensing management system. When you use FLEXlm licensing, the HSPICE tool reads the `SNPSLMD_LICENSE_FILE` or the `LM_LICENSE_FILE` environment variable to find the available license server. If the HSPICE tool cannot authorize the access, the job terminates at this point and the HSPICE tool prints an error message in the output listing file.

- Simulation configuration:

  The HSPICE tool reads the appropriate `hspice.ini` file. The search order for the configuration file is:

  1. The current working directory.
  2. The $HOME directory of the user.
  3. The HSPICE installation directory.

- Design input:

  The HSPICE tool opens the input netlist file `demo.sp`. If this file does not exist, a `no input data` error message appears in the output listing file.

  The HSPICE tool then opens the output listing file `demo.lis` for writing. If you do not own the current directory, the tool terminates with a file open error.

- Library input:

  The HSPICE tool reads the files, if any, that you specified in the `.INCLUDE` and `.LIB` statements.

- Operating point initialization:

  The HSPICE tool reads initial conditions, if any, that you specified in the `.IC` and `.NODESET` commands, finds an operating point (that you can save with a `.SAVE` command), and writes any operating point information that you requested.

- Analysis:

  The HSPICE tool performs the specified single or multipoint sweep of the design and produces a set of output files.

- `.ALTER` commands:

  Using `.ALTER` commands, you can vary the simulation conditions and repeat the specified single or multipoint analysis. You can activate multiprocessing while running `.ALTER` cases by entering `hspice -dp` on the command line.

- Suspending a simulation:

  To suspend a simulation job, press Ctrl+Z. The Load Sharing Facility (LSF) then frees up the HSPICE license for another simulation job. To resume the suspended job, on the same terminal, type either `fg` or `bg`. This command accesses an HSPICE license and continues the suspended simulation.

- Normal termination:

  After you complete the simulation, the HSPICE tool closes the opened files and releases the license tokens used for the simulation.

Analysis Types

The HSPICE tool can simulate electrical circuits in the time and frequency domains. The HSPICE tool provides analysis capabilities that support the design of advanced analog and high-speed circuits. See the following table for the analysis types supported by the HSPICE tool. See also Chapter 3, Running an Analysis *(on page 25)* in this guide.

Table 1 HSPICE Analysis Types

Analysis	Description
.AC	AC sweep analysis
.ACMATCH	Analyzes the effects of local variations on the AC response of a circuit.
.ACPHASENOISE	AC phase noise analysis
.BIASCHK	Dynamic bias checking
.DC	DC sweep analysis
.DCMATCH	Analyze the effects of local variations on the DC characteristics of a circuit.
.DCSENS	DC sensitivity analysis using variation definitions specified using variation blocks.
.DISTO	Distortion analysis
.ENV	Envelope analysis
.ENVFFT	Envelope FFT analysis
.ENVOSC	Envelope analysis for oscillators

Table 1 HSPICE Analysis Types (Continued)

Analysis	Description
.FFT	Fast Fourier transform (FFT) analysis
.FOUR	Fourier analysis
.HB	Harmonic balance analysis
.HBAC	Multi-tone harmonic balance AC analysis
.HBLIN	Frequency translation S-parameter extraction
.HBLSP	Large-signal S-parameter analysis
.HBNOISE	Multi-tone harmonic balance noise analysis
.HBOSC	Harmonic balance analysis for oscillators
.HBXF	Harmonic balance transfer function analysis
.LIN	Linear network analysis
.LSTB	Loop stability analysis
.MOSRA	MOSFET reliability analysis
.NOISE	Small-signal noise analysis
.OP	Operating point analysis
.PHASENOISE	Phase noise analysis
.PTDNOISE	Periodic time-domain noise analysis
.PZ	Pole/zero analysis
.SAMPLE	Data sampling noise analysis
.SENS	Determines DC small-signal sensitivities of output variables for circuit parameters.
.SN	Shooting Newton analysis
.SNAC	Shooting Newton AC analysis
.SNFT	Shooting Newton discrete Fourier transform analysis
.SNNOISE	Shooting Newton noise analysis
.SNOSC	Shooting Newton analysis for oscillators

Feedback

Table 1 HSPICE Analysis Types (Continued)

Analysis	Description
.SNXF	Shooting Newton transfer function analysis
.STATEYE	Statistical eye diagram analysis
.TF	Calculates small-signal values for transfer functions for both DC and AC simulations.
.TRAN	Transient analysis
.TRANNOISE	Transient noise analysis

See also Chapter 3, Running an Analysis *(on page 25)* in this guide.

Elements

HSPICE element statements describe the netlists of devices and sources. Each HSPICE element begins with a unique letter that identifies its device type. See the following table. See also Chapter 4, Working With Elements *(on page 40)* in this guide.

Table 2 HSPICE Elements

Element	Description
B	IBIS Buffer
C	Capacitor
D	Diode
E	Voltage Dependent Voltage Source
F	Current Dependent Current Source
G	Voltage Dependent Current Source
H	Current Dependent Voltage Source
I	Independent Current Source
J	JFET or MESFET
K	Mutual Inductor
L	Inductor
M	MOSFET
P	Port
Q	BJT Transistor
R	Resistor
S	S Parameter
T	Ideal Transmission Line
U	Lossy Transmission Line
V	Independent Voltage Source
W	Lossy Transmission Line (preferred)

Table 2 HSPICE Elements (Continued)

Element	Description
X	Subcircuit

See also Chapter 4, Working With Elements *(on page 40)* in this guide.

Models

The HSPICE tool supports many common semiconductor device models. See the following table. See also Chapter 5, Working With Models *(on page 42)* in this guide.

Table 3 Device Models Supported by the HSPICE Tool

Device Model	Description
Capacitors	
CMC Varactor	Level=7 CMC Varactor Model
Resistors	
CMC R2 Resistor	Level=2 CMC R2 resistor model
CMC R3 Resistor	Level=5 CMC R3 resistor model
Diodes	
Non-Geometric Junction Diode	Level=1 non-geometric junction diode model
Fowler-Nordheim Diode	Level=2 Fowler-Nordhiem diode model
Geometric Junction Diode	Level=3 geometric junction diode model
JUNCAP1 Diode	Level=4 junction capacitance diode model
JUNCAP2 Diode	Level=6 junction capacitance diode model
Philips D500 Diode	Level=5 Philips D500 advanced diode model
CMC Diode	Level=7 CMC diode model

Table 3 Device Models Supported by the HSPICE Tool (Continued)

Device Model	Description
JFETs and MESFETs	
Level 1 JFET	Level=1 JFET model
Level 2 JFET	Level=2 JFET model
Level 3 MESFET	Level=3 MESFET model
TriQuint MESFET	Level=7 TriQuint MESFET model
Materka MESFET	Level=8 Materka MESFET
BJTs	
Level 1 BJT	Level=1 BJT model
Level 2 BJT	Level=2 BJT model
VBIC	Level=4 VBIC Model
Philips Bipolar MEXTRAM Level 503	Level=6 Philips Bipolar Model
Philips Bipolar MEXTRAM Level 504	Level=6 Philips Bipolar Model
HICUM	Level=8 HICUM Model
VBIC99	Level=9 VBIC99 Model
Philips MODELLA Bipolar	Level=10 Philips MODELLA Bipolar Model
UCSD HBT	Level=11 UCSD HBT Model
HICUM/L0	Level=13 HICUM/L0 Model
MOSFETs	
Schichman-Hodges	Level=1Schichman-Hodges Model
MOS2 Grove-Frohman (SPICE 2G)	Level=2 MOS2 Grove-Frohman Model
MOS3 Empirical (SPICE 2G)	Level=3 MOS3 Empirical Model
Grove-Frohman Level 2	Level=4 Grove-Frohman Model
AMI-ASPEC	Level=5 AMI-ASPEC Model
Lattin-Jenkins-Grove (ASPEC)	Level=6 Lattin-Jenkins-Grove (ASPEC) Model

Table 3 Device Models Supported by the HSPICE Tool (Continued)

Device Model	Description
Lattin-Jenkins-Grove (SPICE)	Level=7 Lattin-Jenkins-Grove (SPICE) Model
Advanced Level 2 MOS2 Grove-Frohman (SPICE 2G)	Level=8 Advanced LEVEL 2: MOS2 Grove-Frohman Model (SPICE 2G)
BSIM	Level=13 BSIM Model
SOSFET	Level=27 SOSFET Model
BSIM Derived	Level=28 BSIM Derived Model
Cypress Depletion	Level=38 Cypress Depletion Model
BSIM2	Level=39 BSIM2 Model
HP a-Si TFT	Level=40 HP a-Si TFT Model
BSIM3 Version 2 MOSFET	Level=47 BSIM3 Version 2 MOSFET Model
BSIM3 Version 3 MOSFET	Level=49 and Level=53 BSIM3 Version 3 MOSFET Model
Philips MOS9	Level=50 Philips MOS9 Model
BSIM4 MOSFET	Level=54 BSIM4 MOSFET Model
EPFL-EKV MOSFET	Level=55 EPFL-EKV MOSFET Model
BSIM3 SOI	Level=57 BSIM3 SOI Model
UFSOI MOSFET	Level=58 UFSOI MOSFET Model
BSIM3 SOI FD	Level=59BSIM3 SOI FD Model
BSIM3 SOI DD	Level=60 BSIM3 SOI DD Model
RPI a-Si TFT	Level=61 RPI a-Si TFT Model
RPI Poli-Si TFT	Level=62 RPI Poli-Si TFT Model
Philips MOS11	Level=63 Philips MOS11 Model
STARC HiSIM MOSFET	Level=64 STARC HiSIM MOSFET Model
SSIMSOI	Level=65 SSIMSOI Model

Table 3 Device Models Supported by the HSPICE Tool (Continued)

Device Model	Description
HSPICE HVMOS	Level=66 HSPICE HVMOS Model
STARC HiSIM2 MOSFET	Level=68 STARC HiSIM2 MOSFET Model
PSP	Level=69 PSP Model
BSIMSOI4	Level=70 BSIMSOI4 Model
Level 71 TFT	Level=71 TFT Model
BSIM-CMG MOSFET	Level=72 BSIM-CMG MOSFET Model
HiSIM_HV	Level=73 HiSIM_HV Model
Level=74 MOS Model 20	Level=74 MOS Model 20 Model
LETI-UTSOI MOSFET	Level=76 LETI-UTSOI MOSFET Model
Level 77 BSIM6 MOSFET	Level=77 BSIM6 MOSFET Model
BSIM-IMG	Level=78 BSIM-IMG Model
EKV3	Level=79 EKV3 Model

See also Chapter 5, Working With Models *(on page 42)* in this guide.

Demonstration Files

The HSPICE tool is shipped with as many as 350 demonstration netlist files that illustrate the use of HSPICE commands and control options, for various applications.

You can find the demonstration files at:

- For HSPICE: `$installdir/demo/hspice/`
- For HSPICE advanced analyses: `$installdir/demo/hspice/rf_examples/`

Where *`$installdir`* is the directory where the HSPICE tool is installed on your network or machine.

For a detailed listing of the HSPICE demonstration file categories and descriptions of the netlists, see:

- For HSPICE examples: Listing of Demonstration Input Files in the *HSPICE User Guide: Demonstration Netlists*.
- For HSPICE advanced analyses: Advanced Analog Demonstration Input Files in the *HSPICE User Guide: Advanced Analog Simulation and Analysis*.

Documentation

The documentation set for the HSPICE tool is available as PDF manuals and online help.

The following sections discuss these topics:

- The HSPICE Documentation Set *(on page 19)*
- Finding HSPICE Documentation *(on page 20)*
- Accessing HSPICE Documentation *(on page 20)*
- Launching Context-Sensitive Help *(on page 21)*

The HSPICE Documentation Set

The HSPICE documentation set, both PDF and online help, includes the following manuals:

Manual	Description
HSPICE® User Guide: Demonstration Netlists	Provides detailed information on locating and running the HSPICE demonstration netlists. See also Topics in This Document in the *HSPICE User Guide: Demonstration Netlists*.
hspice_qsg.ditamap	This guide provides information to quickly get you started with using the HSPICE tool for simulation and analysis of your circuit designs. See also .
HSPICE® User Guide: Basic Simulation and Analysis	Describes how to use HSPICE to simulate and analyze your circuit designs, and includes simulation applications. This is the main HSPICE user guide. See also Topics in This Document in the *HSPICE User Guide: Basic Simulation and Analysis*.

Manual	Description
HSPICE® User Guide: Elements	Describes the syntax for the basic elements of a circuit netlist in HSPICE, descriptions of each of the elements keywords, and examples of common usage for each element. See also Topics in This Document in the *HSPICE User Guide: Elements*.
HSPICE® User Guide: Signal Integrity Modeling and Analysis	Describes how to use HSPICE® to achieve and maintain signal integrity in your chip design. See also Topics in This Document in the *HSPICE User Guide: Signal Integrity Modeling and Analysis*.
HSPICE® User Guide: Advanced Analog Simulation and Analysis	Describes how to use special set of analysis and design capabilities added to HSPICE to support RF and high-speed circuit design. See also Topics in This Document in the *HSPICE User Guide: Advanced Analog Simulation and Analysis*.
HSPICE® User Guide: Advanced Variability Analysis	Describes the principles of HSMC and Sigma Amplification simulations and provides information on how to perform an HSMC and Sigma Amplification simulation in HSPICE. See also Topics in This Document in the *HSPICE User Guide: Advanced Variability Analysis*.
HSPICE® Reference Manual: Commands and Control Options	Describes the individual HSPICE commands you can use to simulate and analyze your circuit designs. See also Topics in This Document in the *HSPICE Reference Manual: Commands and Control Options*.
HSPICE® Reference Manual: Device Models	Describes standard models you can use when simulating your circuit designs in HSPICE, including passive devices, diodes, JFET and MESFET devices, and BJT devices. See also Topics in This Document in the *HSPICE Reference Manual: Device Models*.
HSPICE® Reference Manual: MOSFET Models	Describes available MOSFET models you can use when simulating your circuit designs in HSPICE. See also Topics in This Document in the *HSPICE Reference Manual: MOSFET Models*.
HSPICE® Integration to Cadence® Virtuoso® Analog Design Environment User Guide	Describes use of the HSPICE simulator integration to the Cadence tool. See also Topics in This Document in the *HSPICE Integration to Cadence Virtuoso Analog Design Environment*.

Finding HSPICE Documentation

You can find the HSPICE documentation files, the online help files and the PDFs, in your HSPICE installation. The documentation files are in the `$installdir/docs_help` directory, where `$installdir` is the directory where the HSPICE tool is installed on your network or machine.

You will find the PDFs in the `$installdir/docs_help/ni` directories.

Accessing HSPICE Documentation

You can view the latest version of the HSPICE online help or download the documentation in PDF format from the SolvNetPlus online support site at HSPICE Documentation on SolvNet.

To view and print the HSPICE documentation in PDF, Synopsys recommends that you install Adobe Reader on your machine. For best results, use Adobe Reader version 6 or later.

You can access the HSPICE documentation in PDF format from the command line using the following command:

```
% hspice -doc
```

To launch the searchable, HTML browser-based HSPICE online help system, enter the following command on the command line:

```
% hspice -help
```

You must have a HTML browser installed on your machine to launch and view the online help.

Launching Context-Sensitive Help

The HSPICE online help is context sensitive. That is, you can directly access and view information about HSPICE commands, options, keywords, and other common help topics in the online help.

- For information about a command, use:

  ```
  % hspice -help command
  ```

  For example:

  ```
  % hspice -help .AC
  ```

- For information about a control option, use:

  ```
  % hspice -help .OPTION_option
  ```

  For example:

  ```
  % hspice -help .OPTION_DELMAX
  ```

- For information about a demo file, use:

  ```
  % hspice -help demo_file_name
  ```

For example:

```
% hspice -help clockbuf4sn.sp
```

- For information about a topic, use:

```
% hspice -help keyword
```

For example:

```
% hspice -help bsim3v3
```

For the list of keywords for online help topics, see Viewing Online Help Topics from the Command-Line in the *HSPICE Reference Manual: Commands and Control Options*.

HSPICE Application Notes

To obtain the HSPICE application notes, go to the SolvNetPlus online support site at https://solvnetplus.synopsys.com or the HSPICE web site at https://www.synopsys.com/verification/ams-verification/hspice.html.

2

Customizing Your Environment

Presents an overview of HSPICE installation, licensing, and environment settings.

To obtain the latest HSPICE installation notes, go to http://www.synopsys.com/install. For detailed licensing setup and troubleshooting assistance, see the *Synopsys Licensing QuickStart Guide* at http://www.synopsys.com/licensing.

This chapter discusses the following topics:

- Installing the HSPICE Tool—An Overview *(on page 23)*
- Setting Up the Licensing Environment *(on page 24)*

Installing the HSPICE Tool—An Overview

The HSPICE tool is available for the Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server, and Windows platforms. For detailed information on the supported compute platforms, operating systems, and windowing environments for your HSPICE release, see *Synopsys® HSPICE® Installation Notes*.

Installing the HSPICE Tool on UNIX

The HSPICE tool uses the Synopsys Installer, which allows you to install the product using either a text script or a graphical user interface (GUI). For information about downloading the Synopsys Installer, see the Synopsys Installer section at http://www.synopsys.com/install.

To install the HSPICE tool, follow the procedures described in the *Synopsys Installation Guide* downloaded from this site.

For detailed information, see *Synopsys® HSPICE® Installation Notes*.

Installing the HSPICE Tool on Windows

For installation on Windows, download the appropriate HSPICE release, from the SolvNetPlus Download Center, to a temporary directory. Double-click the downloaded `HSPICE.exe` file to start the installation process. For detailed information, see *Synopsys® HSPICE® Installation Notes*.

Setting Up the Licensing Environment

The HSPICE tool requires you to set the `LM_LICENSE_FILE` environment variable. This variable specifies the location of the `license.dat` license file. Set the `LM_LICENSE_FILE` environment variable to *`port@hostname`* to point to a license file on a server.

- If you are using the C shell, add the following line to the `.cshrc` file:

  ```
  setenv LM_LICENSE_FILE port@hostname
  ```

- If you are using the Bash or Bourne shell, add these lines to the `.bashrc` or the `.profile` file:

  ```
  LM_LICENSE_FILE=port@hostname
  export LM_LICENSE_FILE
  ```

The *`port`* and *`hostname`* variables correspond to the TCP port and the license server host name specified in the SERVER line of the Synopsys license file.

Each license file can contain licenses for many packages from multiple vendors. You can specify multiple license files by separating each entry. For Linux platforms, use a colon (:) as the separator. For Windows platforms, use a semicolon (;) as the separator.

For detailed information on setting up the user environment for HSPICE licensing, see *Synopsys® HSPICE® Installation Notes*.

Setting Up License Queuing

The optional `META_QUEUE` environment variable is a useful feature that you can use to control the HSPICE tool to wait for an available license. It is particularly useful in environments where the tool runs sequentially from batch files and a license checkout failure could result in the loss of important data.

Setting the `META_QUEUE` environment variable to 1 enables queuing of HSPICE licenses:

```
setenv META_QUEUE 1
```

3

Running an Analysis

Contains information on running an analysis in the HSPICE tool. Introduces the HSPICE netlist structure and the analysis outputs. Presents a few simulation examples.

This chapter discusses the following topics:

- Running the HSPICE Tool *(on page 25)*
- Netlist Structure *(on page 26)*
- Output Files *(on page 30)*
- Simulation Examples *(on page 31)*

Running the HSPICE Tool

Use the following syntax to run the HSPICE tool:

```
hspice [-dpredict]
       [-i path/input_file]
       [-o path/output_file]
       [-n number]
       [-gz]
       [-d]
       [-C path/input_file]
       [-CC path/input_file]
       [-I]
       [-K]
       [-L command_file]
       [-include_first ins_filename]
       [-include_last app_filename]
       [-S]
       [-case 0|1]
       [-datamining -i datamining.cfg [-o outname]]
       [-dp [num]
            [-dpconfig dp_configuration_file]
            [-dplocation NFS|TMP]
```

```
          [-merge]
          [-dpgui]
          [-dpmode alter|sweep]
          [-dpincremental original_path]
          [-dscale dp_num]]
     [-mp process_count]
     [-mt thread_count]
     [-hpp]
     [-meas measure_file]
     [-mrasim [0|1|2|3]]
     [-top subcktname]
     [-restore checkpoint_file]
     [-hdl file_name]
     [-hdlpath pathname]
     [-vamodel name]
     [-vamodel name2...]
     [-sae]
     [-search path1[:path2:path3...]]
     [-alter_select  num | num1:num2 | alter_name]
     [-help]
     [-doc]
     [-h]
     [-v]
     [-t | -time | -stop  time_value]
     [-wavefmt | -format fsdb|wdf|psf|tr0]
```

For a detailed description of the `hspice` command syntax and arguments, see hspice in the *HSPICE® Reference Manual: Commands and Control Options*. For examples on starting and running the HSPICE tool, see Starting HSPICE — Examples.

For multiple processing, multithreading, distributed processing, and HSPICE Precision Parallel features, see Chapter 4, Distributed Processing, Multithreading, and HSPICE Precision Parallel in *HSPICE® User Guide: Basic Simulation and Analysis*.

For the analysis types supported by the HSPICE tool, see Analysis Types *(on page 11)* on page 11 in this guide.

Netlist Structure

The following sections discuss these topics:

- HSPICE Netlists *(on page 26)*
- Sections in a Netlist File *(on page 27)*
- Examples *(on page 28)*

HSPICE Netlists

HSPICE netlists can be generated from text editors or from netlisters that generate circuits from schematics. The HSPICE tool can accept either hierarchical or flat netlists.

The process of creating a schematic involves the following steps:

1. Symbol creation with a symbol editor.
2. Circuit encapsulation.
3. Property creation.
4. Symbol placement.
5. Symbol property definition.
6. Wire routing and definition.

Sections in a Netlist File

The following table lists and describes the sections in an HSPICE netlist:

Table 4 Sections of the Input Netlist File

Section	Command/Control Option Used	Description
Title	.TITLE	The first line in the netlist is the title of the input netlist file. The `.TITLE` command is optional.
Setup	.OPTION	Sets conditions for simulation.
	.IC or .NODESET	Initial values in circuit and subcircuit.
	.PARAM	Sets parameter values in the netlist.
	.GLOBAL	Sets the node name globally in the netlist.
Sources	Sources and digital inputs	Sets input stimuli (I or V element).
Netlist	.SUBCKT, .ENDS, or .MACRO, .EOM	Circuit elements for simulation. Subcircuit or macro definitions.
Analysis	.DC, .TRAN, .AC, and so on.	Statements to perform analyses.

Table 4 Sections of the Input Netlist File (Continued)

Section	Command/Control Option Used	Description
	.SAVE and .LOAD	Saves and loads operating point information. (HSPICE only)
	.DATA	Creates a table for data-driven analysis.
	.TEMP	Sets temperature analysis.
Library, model and file Inclusion	.INCLUDE	General include files.
	.MODEL	Element model descriptions.
	.LIB	Model library.
	.MALIAS	Assigns an alias to a diode, BJT, JFET, or MOSFET.
	.OPTION SEARCH	Search path for libraries and included files. (HSPICE only)
Output	.PRINT, .PROBE	Statements to output variables.
	.MEASURE	Statement to evaluate and report user-defined functions of a circuit.
Alter blocks	.ALTER	Sequence for inline case analysis.
	.DEL LIB	Removes previous library selection.
End of netlist	.END	End of netlist (required).

Examples

Table 5 Netlist Command Examples

Item	Example
Title	The first line is always the title of the netlist.
Comment characters	`*`: Comment character for a line
	`$`: Comment character used after a command
Options	`.option post`

Table 5 Netlist Command Examples (Continued)

Item	Example
	`.option runlvl=5`
Print/Probe/Analysis	`.print tran v(d) i(rl)`
	`.probe tran v(g)`
	`.tran 0.1n 10n`
Initial conditions	`.ic v(b) = 0 $ input state`

Sources	`Vg g 0 pulse 0 1 0 0.1n 0.1n 1n 2n`
	`* example of a voltage source`

Circuit description	`MN d g gnd n nmos`
	`RL vdd d 1K`

Models	`.model n nmos level = 49`
	`+ vto = 1 tox = 7n`
	`* '+' continuation character`

End	`.end $ terminates the simulation`

Output Files

Table 6 *(on page 30)* lists the various types of output files produced by the HSPICE tool.

For information on the output files produced when performing advanced analog analyses, see HSPICE Advanced Analog Output Files in *HSPICE® User Guide: Advanced Analog Simulation and Analysis*.

Table 6 HSPICE Output Files and Their Extensions

Output File Type	**Extension**
AC analysis measurement results	`.ma#`[1]
AC analysis results (from `.POST` statement)	`.ac#`
Monte Carlo results	`.mc#`
Data-mining results	`.mpp0`
DC analysis measurement results	`.ms#`
DC analysis results (from `.POST` statement)	`.sw#`
FFT analysis graph data (from `.FFT` statement)	`.ft#`
Operating point information (from `.OPTION OPFILE` statement)	`.dp#`
Operating point node voltages (initial conditions)	`.ic#`
Output listing	`.lis` or user-specified
Output status	`.st#`
Output tables (from `.DCMATCH OUTVAR` statement)	`.dm#`
Pole-zero analysis results	`.pz#`
StatEye analysis measurement results	`.mste#`
StatEye analysis initial transient measurement results	`.mtNp{f}#`[2]
StatEye analysis initial transient results	`.trNp{f}#`[3]

[1] *The hash character (#) can be either a sweep number or a hardcopy file number. For .ac#, .dp#, .dm#, .ic#, .st#, .sw#, and .tr# files, # can be a number from 0 through 9999.*

[2] *N is the incident port index, f is the edge index when edge is greater than 1, and # is the alter index.*

[3] *N is the incident port index, f is the edge index when edge is greater than 1, and # is the alter index.*

Table 6 HSPICE Output Files and Their Extensions (Continued)

Output File Type	Extension
StatEye analysis time-based results (from `.POST` statement)	`.stet#`
StatEye analysis voltage based results (from `.POST` statement)	`.stev#`
Subcircuit cross-listing	`.pa#`
Transient analysis measurement results	`.mt#`
Transient analysis results (from `.POST` statement)	`.tr#`
Waveform viewing files from `.OPTION WDF` argument for use with Synopsys WaveView/SX tools	`*_wdf.tr#`, `*_wdf.sw#`, or `*_wdf.ac#`

Simulation Examples

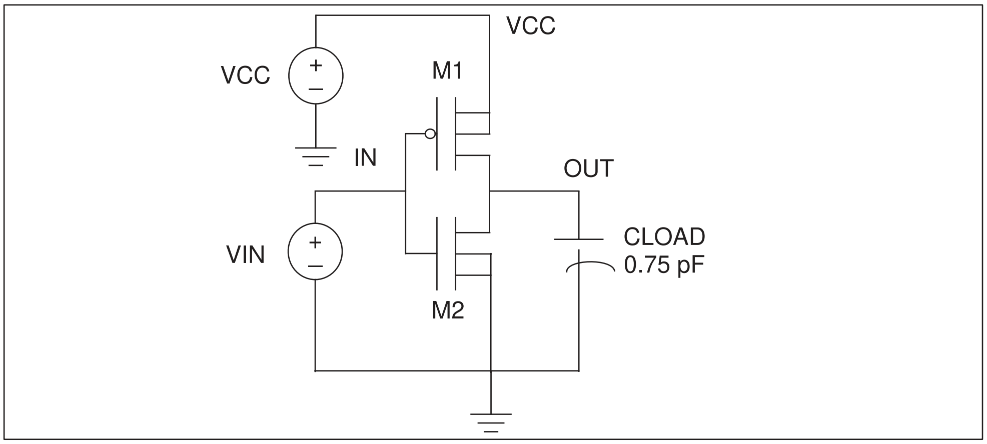
The following examples illustrate the basic HSPICE simulation types: DC, AC, and transient analysis:

- DC Analysis of an Inverter *(on page 32)*
- AC Analysis of an RC Network *(on page 33)*
- Transient Analysis of an RC Network *(on page 36)*
- Transient Analysis of an Inverter *(on page 38)*

DC Analysis of an Inverter

You can analyze the DC behavior of the simple MOS inverter shown in the following figure:

Figure 2 MOS Inverter Circuit



To analyze the DC behavior:

1. Type the following netlist data into a file named `quickDC.sp`:

```
Inverter Circuit
.OPTION POST
.DC VIN 0 5 0.1
.PRINT DC 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
```

   You can find the complete netlist for this example in `$installdir/demo/hspice/apps/quickDC.sp`.

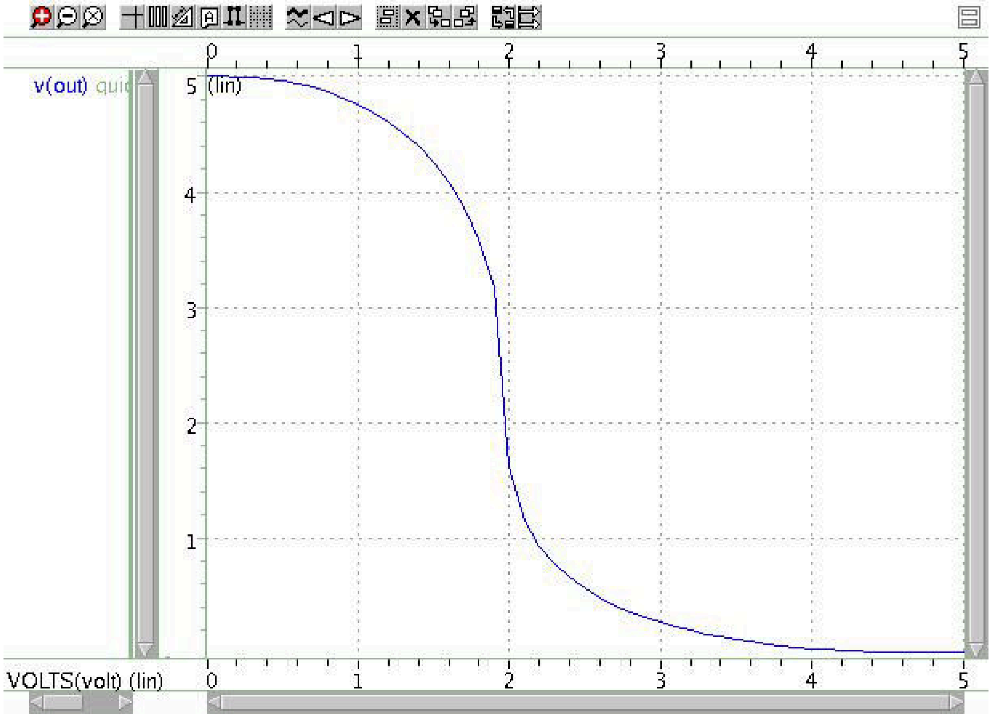
2. Run the HSPICE analysis by typing the following command:

```
hspice -i quickDC.sp -o quickDC.lis
```

3. Use WaveView to examine the voltage waveform at the inverter OUT node.

Figure 3 *(on page 33)* on page 33 shows the waveforms.

Figure 3 *Voltage at Inverter Node v(out)*

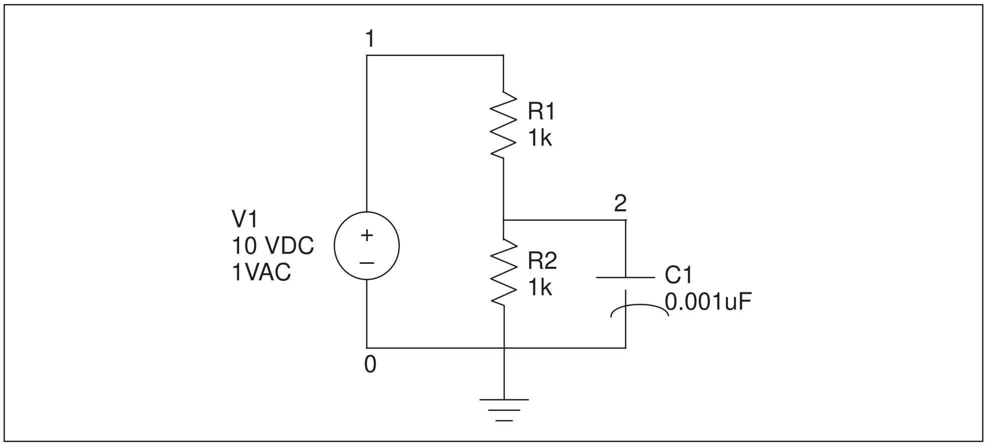


AC Analysis of an RC Network

Figure 4 *(on page 34)* on page 34 shows a simple RC network with a DC and an AC source applied. The circuit consists of:

- Two resistors, R1 and R2.
- Capacitor C1.
- Voltage source V1.
- Node 1 is the connection between the source positive terminal and R1.
- Node 2 is where R1, R2, and C1 are connected.
- HSPICE ground is always node 0.

Figure 4 *RC Network Circuit*



The netlist for this RC network is based on the demonstration netlist `quickAC.sp`, which is available in the directory `$installdir/demo/hspice/apps`:

```
A SIMPLE AC RUN
.OPTION 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
```

To perform an AC analysis for an RC network circuit:

1. Type the above netlist into a file named `quickAC.sp`.
2. To run an HSPICE analysis, type:

   ```
   hspice quickAC.sp > quickAC.lis
   ```

   When the run finishes, the HSPICE tool displays:

   ```
   >info:     ***** hspice job concluded
   ```

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

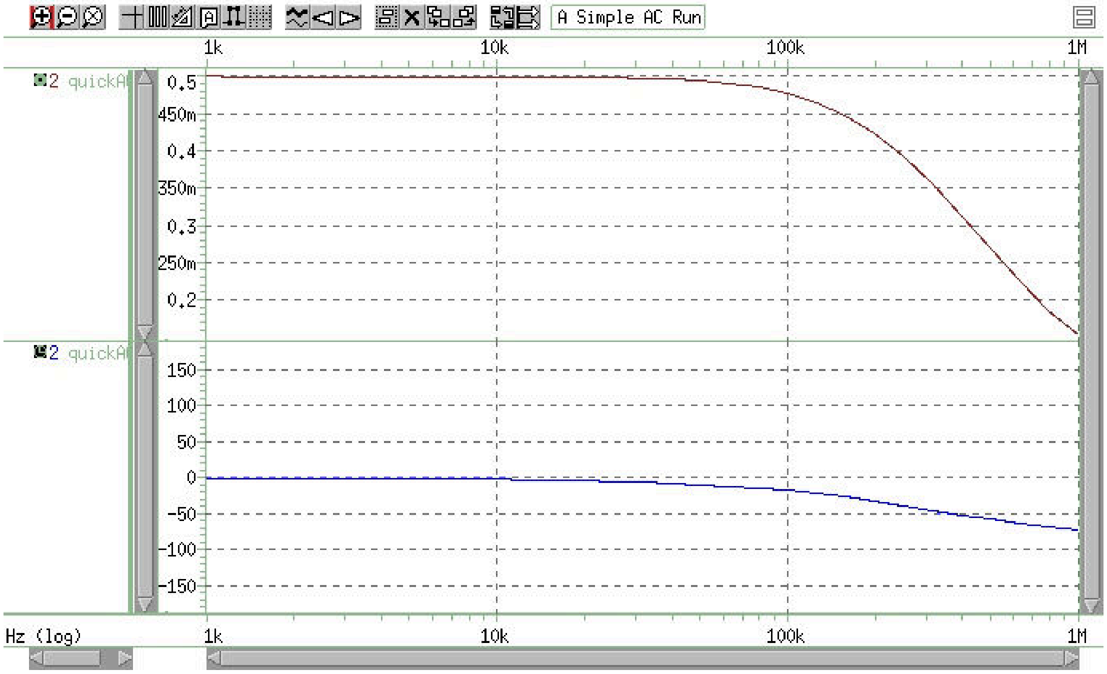
Your run directory includes the following new files:

- `quickAC.ac0`
- `quickAC.ic0`
- `quickAC.lis`
- `quickAC.st0`

3. Use a text editor to view the `.lis` and `.st0` files to examine the simulation results and status.
4. Run WaveView.
5. From the menu bar, select **File** > **Import** > **Waveform File**.
6. Select the `quickAC.ac0` file from the Open: Waveform Files window.
7. Display the voltage at node 2 by using a log scale on the x-axis.

Figure 5 *(on page 35)* shows the waveform that the HSPICE tool produces if you sweep the response of node 2, as you vary the frequency of the input from 1 kHz to 1 MHz.

Figure 5 RC Network Node 2 Frequency Response



As you sweep the input from 1 kHz to 1 MHz, the `quickAC.lis` file displays:

- Input netlist.
- Details about the elements and topology.
- Operating point information.
- Table of requested data.

The `quickAC.ic0` file contains information about the DC operating point conditions. The `quickAC.st0` file contains information about the simulation run status.

To use the operating point conditions for subsequent simulation runs, execute the `.LOAD` statement.

Transient Analysis of an RC Network

To run a transient analysis of an RC network with a pulse source, a DC source, and an AC source:

1. Type the following netlist into a file named `quickTRAN.sp`:

   ```
   A SIMPLE TRANSIENT RUN
   .OPTION 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
   ```

   This example uses the demonstration netlist `quickTRAN.sp`, which is available in the directory `$`*`installdir`*`/demo/hspice/apps`.

   Note:

   > The V1 source specification includes a pulse source. For the syntax of pulse sources and other types of sources, see Sources and Stimuli in the *HSPICE User Guide: Elements*.

2. To run HSPICE, type the following:

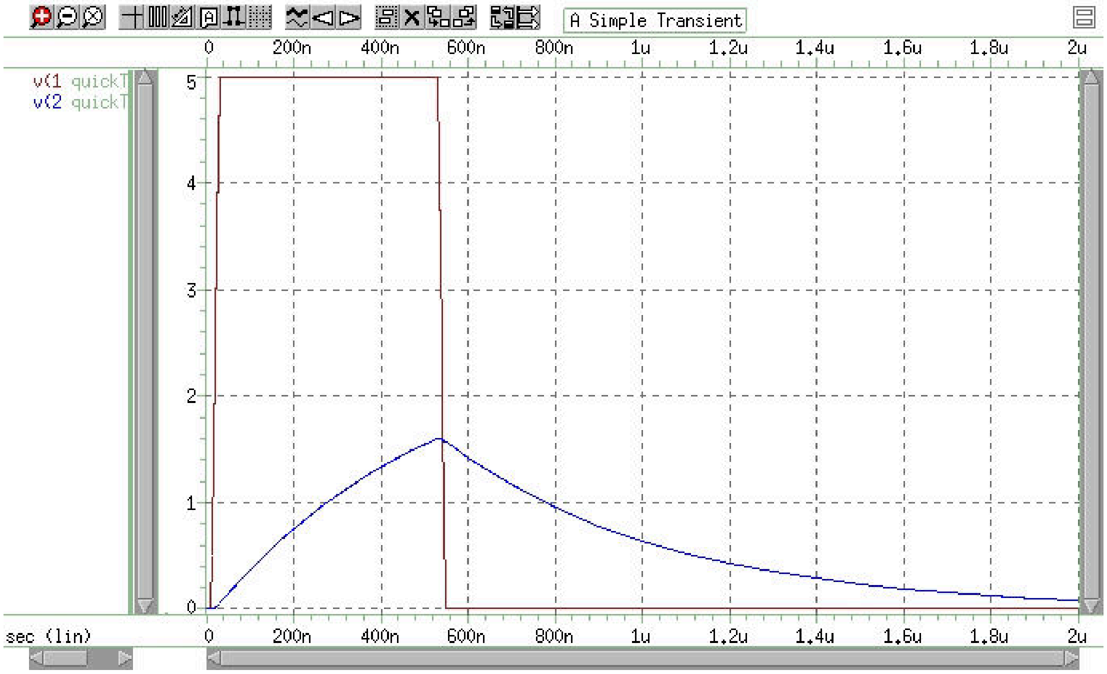
   ```
   hspice quickTRAN.sp > quickTRAN.lis
   ```

3. To examine the simulation results and status, use a text editor and view the `.lis` and `.st0` files.

4. Run WaveView and open the `.sp` file.
5. From the menu bar, select **File** > **Import Waveform** > **File**.
6. Select the `quickTRAN.tr0` file from the Open: Waveform Files window.
7. Display the voltage at nodes 1 and 2 on the x-axis.

Figure 6 *(on page 37)* shows the waveforms.

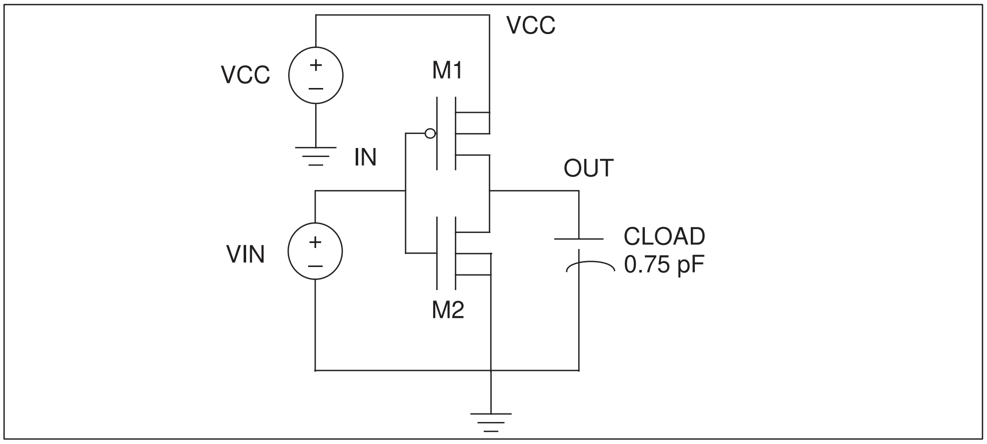
Figure 6 Voltages at RC Network Circuit Node 1 and Node 2



Transient Analysis of an Inverter

You can analyze the transient behavior of the simple MOS inverter shown in Figure 7 *(on page 38)*.

Figure 7 MOS Inverter Circuit



To analyze this behavior:

1. Type the following netlist data into a file named `quickINV.sp`:

```
Inverter Circuit
.OPTION 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
```

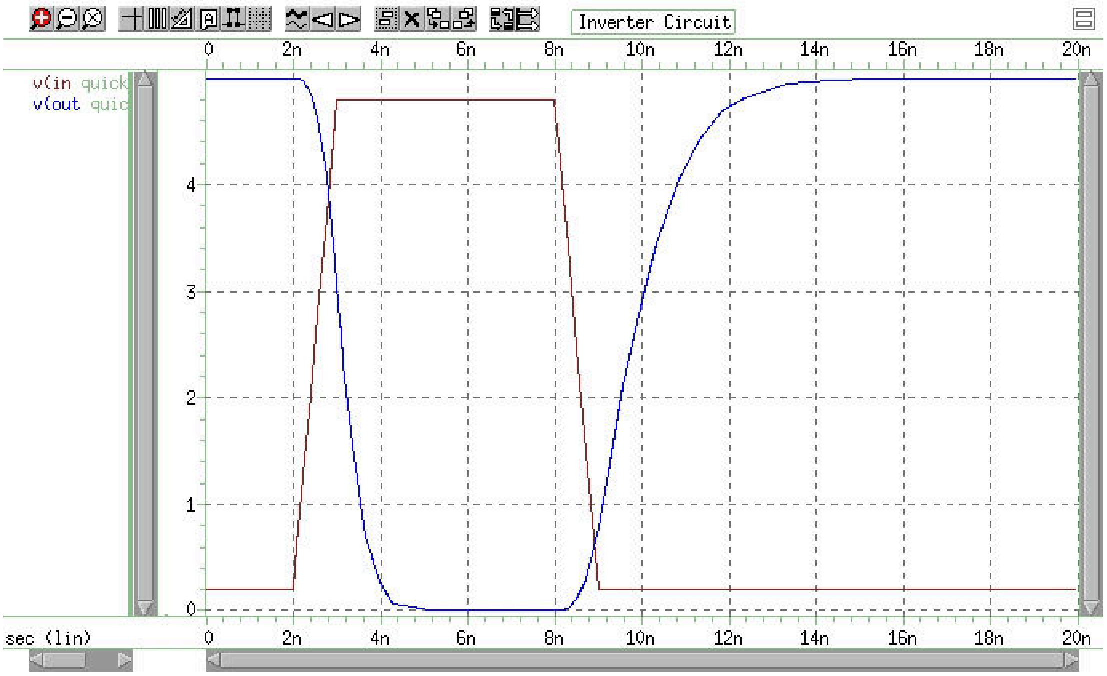
   You can find the complete netlist for this example in the directory `$installdir/demo/hspice/apps/quickINV.sp`.

2. To run HSPICE, type:

   ```
   hspice quickINV.sp > quickINV.lis
   ```

3. Use WaveView to examine the voltage waveforms, at the inverter IN and OUT nodes. Figure 8 *(on page 39)* on page 39 shows the waveforms.

Figure 8 Voltage at MOS Inverter Nodes v(in) and v(out)



4

Working With Elements

Presents an overview of elements in the HSPICE tool.

This chapter discusses the following topics:

Introduction to Elements

Element statements in the HSPICE tool describe the devices and sources in the netlist. Nodes are used to connect elements to one another. Nodes can be defined using either numbers or names.

Element statements specify the following characteristics of elements:

- The type of device.
- The nodes to the connected device.
- The operating electrical characteristics of the device.

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

For detailed descriptions of element statements for each of the supported elements, see *HSPICE® User Guide: Elements*. See also Elements *(on page 14)* on page 14 in this guide.

About Element Instances

The names of element instances begin with the element key letter (see the following table), except in subcircuits where instance names begin with X. Instance names can be up to 1024 characters long.

Table 7 Element Identifiers

Key Letter (First Character)	Element	Example Line
B	IBIS buffer	`b_io_0 nd_pu0 nd_pd0 nd_out nd_in0 nd_en0 nd_outofin0 nd_pc0 nd_gc0`
C	Capacitor	`Cbypass 1 0 10pf`
D	Diode	`D7 3 9 D1`
E	Voltage-controlled voltage source	`Ea 1 2 3 4 K`
F	Current-controlled current source	`Fsub n1 n2 vin 2.0`
G	Voltage-controlled current source	`G12 4 0 3 0 10`
H	Current-controlled voltage source	`H3 4 5 Vout 2.0`
I	Current source	`I A 2 6 1e-6`
J	JFET or MESFET	`J1 7 2 3 GAASFET`
K	Linear mutual inductor (general form)	`K1 L1 L2 1`
L	Linear inductor	`LX a b 1e-9`
M	MOS transistor	`M834 1 2 3 4 N1`
P	Port	`P1 in gnd port=1 z0=50`
Q	Bipolar junction transistor	`Q5 3 6 7 8 pnp1`
R	Resistor	`R10 21 10 1000`
S	S-parameter	`S1 nd1 nd2 s_model2`
V	Voltage source	`V1 8 0 5`
T, U, W	Transmission line	`W1 in1 0 out1 0 N=1 L=1`
X	Subcircuit call	`X1 2 4 17 31 MULTI WN=100 LN=5`

5

Working With Models

Presents an overview of the usage of standard device models in the HSPICE tool.

This chapter discusses the following topics:

Introduction to Models

Every device model is a template defining various versions of each supported element type used in a netlist formatted for use by the HSPICE tool. Individual elements in your netlist can refer to these standard models for their basic definitions. When you use these models, you can quickly and efficiently create a netlist and simulate your circuit design.

Within your netlist, each element that refers to a model is known as an instance of that model. When your netlist refers to predefined device models, you reduce both the time required to create and simulate a netlist and the risk of errors, as compared to completely defining each element in your netlist.

See also Models *(on page 15)* on page 15 in this guide.

Selecting Models

To specify a device in your netlist, use both an element and a model statement. The element statement uses the model name of the simulation device to reference the model statement. The following example uses the name PCH to refer to a MOSFET model. The example uses a PMOS model type to describe a P-channel MOSFET transistor:

```
M3 3 2 1 0 PCH L=1u W=1u
.MODEL PCH PMOS VERSION = 3.2 tnom=27.0 tox=1.00000E-08
```

```
+ xj=1.00000E-07 lint=8.195860E-08 wint=-1.821562E-07 vth0=-.86094574
+ vsat=60362.05
```

You can specify parameters in both element and model statements. If you specify the same parameter in both an element and a model, the element parameter (local to the specific instance of the model) always overrides the model parameter (global default for all instances of the model, if you do not define the parameter locally). The model statement specifies the type of device, for example, a MOSFET; the device type might be N-channel or P-channel.

Models can be selected from model libraries using the `.LIB` command. The following example calls the model library file `mosfet.lib`, which contains the PCH model used in the netlist. The model corner used in the netlist is the *tt* corner:

```
.LIB '../models/mosfet.lib' tt
M3 3 2 1 0 PCH L=1u W=1u
```

A

Measurement System, Units, and Numbers

Describes the measurement system used in the HSPICE tool.

This appendix covers the following topics:

- About the HSPICE Measurement System *(on page 44)*

About the HSPICE Measurement System

The HSPICE tool uses the MKS (meter, kilogram, and second) measurement system, unless otherwise stated. The tool expects length and width in units of meters. However, the HSPICE tool will directly support the unit *mil* (=0.001inch or 25.4e-06 meters) as input.

Caution:

Be careful when mixing units of measurements; otherwise, the HSPICE tool may not produce the results you expect.

You can enter numbers as an integer, a floating point, a floating point with an integer exponent, or an integer or a floating point with one of the scale factors listed in the following table.

Table 8 Scale Factors

Scale Factor	Prefix	Symbol	Multiplying Factor
T	tera	T	1e+12
G	giga	G	1e+9
ME, MEG, X, or Z	mega	M, ME, X, or Z	1e+6
K	kilo	k	1e+3
MI or MIL	n/a	MI or MIL	25.4e-6

Table 8 Scale Factors (Continued)

Scale Factor	Prefix	Symbol	Multiplying Factor
U	micro	µ	1e-6
N	nano	n	1e-9
P	pico	p	1e-12
F	femto	f	1e-15
A	atto	a	1e-18
DB	DB	db	$10^{(value/20)}$
MIN	MIN	min	60
HR	HR	hr	3600
DAY	DAY	day	86400
YR	YR	yr	31536000

Note:

- Scale factor A is not a scale factor in a character string that contains *amps*. For example, the HSPICE tool interprets the string *20amps* as 20 amperes (of current), not as *20e-18 amps*.
- The scale factor M indicates either the suffix *milli* or *mega*. The HSPICE tool uses MEG or X to represent the suffix *mega* and M to represent the suffix *milli*. That is, 1m = 1e-3 (*milli*) and 1meg = 1x = 1e6 (*mega*)

B

Best Practices

Lists a few best practices for HSPICE netlists, netlist topologies, and analysis.

This appendix covers the following topics:

Netlists

Best practices for generating input netlists for the HSPICE tool include:

- Use either a schematic netlister or a text editor to generate the input netlist and library input files for the HSPICE tool.
- Each netlist line (logical record) cannot exceed 1024 characters. Use the `+` line continuation character to break up lines longer than 1024 characters, to avoid generating an error.
- An input file name can be up to 1024 characters long for all platforms, except Windows platforms which have a limitation of only up to 256 characters.
- The HSPICE tool has a limitation on the number of characters in a path name plus file name, of up to 1024 characters (except Windows platforms which have a limitation of only up to 256 characters). For example:

  ```
  hspice -i path_name/input_file -o out_file
  ```

- When specifying a path and file name using `-i` or `-o`, the length must be 1024 characters or fewer, on all platforms. If the working directory path is greater than 1024 characters, the HSPICE tool aborts with an error message.

- Model names in a netlist must begin with a letter. If you enter a model name with a leading integer, the HSPICE tool issues a parsing error.
- Statements in the input netlist file can be in any order, except the first line which is the netlist title line. In the HSPICE tool, the last `.ALTER` submodule must appear at the end of the file and before the `.END` statement. If you do not place an `.END` statement at the end of the input netlist file, the HSPICE tool issues an error message.
- To indicate the ground node, use either the number 0, or the name GND, or GND!, or GROUND. The ground node is global.
- The HSPICE tool ignores differences between uppercase and lowercase characters in input lines, except in quoted filenames and after the `.INC` and `.LIB` commands. Use the `-case` command-line option if case sensitivity is required.
- Lines can be continued using the `+` character at the beginning of the continued line. For example:

```
V1 1 0 DC=0
+ PULSE (0 1 0 1n 1n 5n 10n
```

- Use a double backslash (`\\`) at the end of the line to continue the line on to the next line, when the continuation is inside a string. Whitespace is optional to precede the string continuation. For example:

```
g1noise 1 2 noise='sqrt(4*1.3806266e23*\\
(TEMPER+273.15)*0.001)'
```

Netlist Topologies

Best practices for constructing netlist topologies for the HSPICE tool include:

- When constructing a netlist, avoid the following topologies because they will cause the HSPICE tool to abort:
  - Voltage loops (that is, voltage sources in parallel with no other elements).
  - An ideal voltage source in a closed inductor loop.
  - Stacked current sources (that is, current sources in series).
  - An ideal current source in a closed capacitor loop.

- The HSPICE tool terminates floating power supply nodes with a 1 MegΩ resistor and a warning message.
- 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).

Analysis

Best practices for analysis statements for the HSPICE tool include:

- For the HSPICE tool to create waveform files, include `.option POST` or `.option WDB` in the input netlist file.
- If you are performing multiple analyses, to avoid warning messages from the HSPICE tool, set the analysis type in all `.PROBE` or `.PRINT` statements.
- Be careful while adding analysis statements in `.ALTER` blocks. The added analysis statement does not replace the analysis statement previously defined in the top level. Instead, the HSPICE tool executes the added analysis command in each `.ALTER` run, in addition to the analysis statement in the top level. This means that, the HSPICE tool will output more analysis result files than expected.
- When using temperature analysis with `.TEMP` and `.ALTER` blocks, it is recommended that the `.TEMP` value be parameterized and only one `.TEMP` statement at the top level be used. You can then change the temperature by changing the value of the parameter assigned to the `.TEMP` command.
- By default, the HSPICE tool uses the last parameter or option found in the netlist. Avoid multiple definitions of the same parameter or option.

C

Abbreviations and Acronyms

Lists the abbreviations and acronyms used in HSPICE documentation.

This appendix covers the following topics:

- List of Abbreviations and Acronyms *(on page 49)*

List of Abbreviations and Acronyms

Table 9 Abbreviations and Acronyms

Abbreviation/Acronym	Expansion
BER	Bit error rate
CFL	Compiled Function Library
CMI	Common Model Interface
DFT	Discrete Fourier transform
DP	Distributed processing
FFT	Fast Fourier transform
HPP	HSPICE Precision Parallel
HSPUI	HSPICE graphical user interface
LIN	Linear network
LSF	Load Sharing Facility
LSTB	Loop stability
MOSRA	MOSFET reliability analysis

Table 9 Abbreviations and Acronyms (Continued)

Abbreviation/Acronym	**Expansion**
MT	Multithreading
SI	Signal integrity
SOA	Safe Operating Area
SPUTIL	S-parameter Utility
StatEye	Statistical eye diagram