



**February 2011**  
**S-Parameter Simulation**

© **Agilent Technologies, Inc. 2000-2011**

5301 Stevens Creek Blvd., Santa Clara, CA 95052 USA

No part of this documentation may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

**Acknowledgments**

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Mentor products and processes are registered trademarks of Mentor Graphics Corporation. \* Calibre is a trademark of Mentor Graphics Corporation in the US and other countries. "Microsoft®, Windows®, MS Windows®, Windows NT®, Windows 2000® and Windows Internet Explorer® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Oracle and Java and registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC. FLEXIm is a trademark of Globetrotter Software, Incorporated. Layout Boolean Engine by Klaas Holwerda, v1.7 <http://www.xs4all.nl/~kholwerd/bool.html> . FreeType Project, Copyright (c) 1996-1999 by David Turner, Robert Wilhelm, and Werner Lemberg. QuestAgent search engine (c) 2000-2002, JObjects. Motif is a trademark of the Open Software Foundation. Netscape is a trademark of Netscape Communications Corporation. Netscape Portable Runtime (NSPR), Copyright (c) 1998-2003 The Mozilla Organization. A copy of the Mozilla Public License is at <http://www.mozilla.org/MPL/> . FFTW, The Fastest Fourier Transform in the West, Copyright (c) 1997-1999 Massachusetts Institute of Technology. All rights reserved.

The following third-party libraries are used by the NlogN Momentum solver:

"This program includes Metis 4.0, Copyright © 1998, Regents of the University of Minnesota", <http://www.cs.umn.edu/~metis> , METIS was written by George Karypis (karypis@cs.umn.edu).

Intel@ Math Kernel Library, <http://www.intel.com/software/products/mkl>

SuperLU\_MT version 2.0 - Copyright © 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy). All rights reserved. SuperLU Disclaimer: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7-zip - 7-Zip Copyright: Copyright (C) 1999-2009 Igor Pavlov. Licenses for files are: 7z.dll: GNU LGPL + unRAR restriction, All other files: GNU LGPL. 7-zip License: This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. unRAR copyright: The decompression engine for RAR archives was developed using source code of unRAR program. All copyrights to original unRAR code are owned by Alexander Roshal. unRAR License: The unRAR sources cannot be used to re-create the RAR compression algorithm, which is proprietary. Distribution of modified unRAR sources in separate form or as a part of other software is permitted, provided that it is clearly stated in the documentation and source comments that the code may not be used to develop a RAR (WinRAR) compatible archiver. 7-zip Availability: <http://www.7-zip.org/>

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. AMD Availability: <http://www.cise.ufl.edu/research/sparse/amd>

UMFPACK 5.0.2 - UMFPACK Notice: The UMFPACK code was modified. Used by permission. UMFPACK Copyright: UMFPACK Copyright © 1995-2006 by Timothy A. Davis. All Rights Reserved. UMFPACK License: Your use or distribution of UMFPACK or any modified version of UMFPACK implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License

as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. UMFPACK Availability: <http://www.cise.ufl.edu/research/sparse/umfpack> UMFPACK (including versions 2.2.1 and earlier, in FORTRAN) is available at <http://www.cise.ufl.edu/research/sparse> . MA38 is available in the Harwell Subroutine Library. This version of UMFPACK includes a modified form of COLAMD Version 2.0, originally released on Jan. 31, 2000, also available at <http://www.cise.ufl.edu/research/sparse> . COLAMD V2.0 is also incorporated as a built-in function in MATLAB version 6.1, by The MathWorks, Inc. <http://www.mathworks.com> . COLAMD V1.0 appears as a column-preordering in SuperLU (SuperLU is available at <http://www.netlib.org> ). UMFPACK v4.0 is a built-in routine in MATLAB 6.5. UMFPACK v4.3 is a built-in routine in MATLAB 7.1.

Qt Version 4.6.3 - Qt Notice: The Qt code was modified. Used by permission. Qt copyright: Qt Version 4.6.3, Copyright (c) 2010 by Nokia Corporation. All Rights Reserved. Qt License: Your use or distribution of Qt or any modified version of Qt implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. Qt Availability: <http://www.qtsoftware.com/downloads> Patches Applied to Qt can be found in the installation at: `$HPEESOF_DIR/prod/licenses/thirdparty/qt/patches`. You may also contact Brian Buchanan at Agilent Inc. at [brian\\_buchanan@agilent.com](mailto:brian_buchanan@agilent.com) for more information.

The HiSIM\_HV source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code, is owned by Hiroshima University and/or STARC.

**Errata** The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

**Warranty** The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this documentation and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

**Technology Licenses** The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at <http://systemc.org/>. This software is redistributed by Agilent. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions or title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Agilent only.

**Restricted Rights Legend** U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

About S-Parameter Simulation . . . . .	7
Performing an S-Parameter Simulation . . . . .	8
Examples of S-Parameter Simulation . . . . .	9
Simulating an Amplifier . . . . .	9
Calculating Group Delay . . . . .	10
Simulating Linear Noise . . . . .	11
Analyzing a Frequency Translating Network . . . . .	12
Eliminating Unwanted Effects . . . . .	12
S-Parameter Simulation Description . . . . .	14
S-Parameter Definitions . . . . .	14
Group Delay . . . . .	15
S-Parameter Frequency Conversion . . . . .	16
S-Parameter Simulation Noise Analysis . . . . .	18
Noise Figure . . . . .	18
Calculating Noise Figure . . . . .	18
Noisy 2-Port Parameters . . . . .	19
S-Parameters from Various Input-Output Modes . . . . .	21
Improving S-Parameter Simulation Speed . . . . .	22
Saving Network Parameters into Other Formats . . . . .	23
S-Parameter Simulation Parameters . . . . .	24
Setting Frequency Sweep . . . . .	24
Defining Simulation Parameters . . . . .	25
Defining Noise Parameters . . . . .	28

# About S-Parameter Simulation

Using the S-parameter simulation controller (S-Parameters) from the Simulation-S\_Param palette enables you to:

- Obtain the scattering parameters (S-parameters) of a component, circuit, or subnetwork and convert those parameters to Y- or Z-parameters.
- Plot, for example, the variations in swept-frequency S-parameters with respect to another changing variable.
- Simulate group delay or linear noise.
- Simulate the effects of frequency conversion on small-signal S-parameters in a circuit employing a mixer. (This is also known as analyzing a frequency-translating network.)

The Simulation-S\_Param palette also contains components for general simulation options and sweeps, as well as a variety of measurement components for calculating relevant measurements. See the following topics for details on S-parameter simulation:

- *Performing an S-Parameter Simulation* (cktsimsp) has the minimum setup requirements for an S-parameter simulation.
- *Examples of S-Parameter Simulation* (cktsimsp) give detailed setups for running a basic S-parameter simulation on an amplifier, as well as examples for calculating group delay, linear noise, and frequency translation.
- *S-Parameter Simulation Description* (cktsimsp) is a brief description of the S-parameter simulator and some of its methods, such as group delay and frequency conversion.
- *S-Parameter Simulation Noise Analysis* (cktsimsp) gives some of the equations and techniques that are the basis of noise calculations.
- *S-Parameters from Various Input-Output Modes* (cktsimsp) describes the features available in ADS to simulate S-parameters for designs that use various input and output modes.
- *Improving S-Parameter Simulation Speed* (cktsimsp) describes how the LinearCollapse component can help increase S-parameter simulation speed in ADS.
- *S-Parameter Simulation Parameters* (cktsimsp) provides details about the parameters available in the S-parameters controller in ADS.



# Performing an S-Parameter Simulation

Start by creating your design, then add current probes and identify the nodes from which you want to collect data.

For a successful analysis, be sure to:

- Apply ports to all inputs and outputs. Use P\_1Tone or P\_nTone power sources to drive inputs. Terminate all other ports using port-impedance terminations (Term). Verify impedance. The Term component is found on the Simulation-S\_Param palette. Power sources are on the Sources-Freq Domain palette.
- Check the Num field for each port. The S-parameter port numbers are derived from these fields. For a 2-port circuit, you would want the input labeled as Num=1 and the output as Num=2.
- Add the S-parameter component to the schematic and double-click to edit it. For a basic simulation, fill in the fields under the Frequency tab:
  - Select the Sweep type, single point, logarithmic, or linear. For a linear or logarithmic sweep, elect to define the sweep with start/stop or center/span values.
- To calculate admittance or impedance parameters, enable the options under the Parameters tab.
- Group delay calculations can be enabled from the Parameters tab.
- You can enable frequency conversion, which is useful when analyzing circuits with standard (not user-defined) behavioral mixer models. This option is under the Parameters tab.
- To calculate noise, select the Noise tab and enable *Calculate noise* . You select a node for noise calculations from the Edit list, then click Add. Use the Mode list to sort the noise contributed by individual noise sources by name or value.

For details about each field, click *Help* from the dialog box.

For more detailed descriptions of simulation setups, refer to *Examples of S-Parameter Simulation* (cktsimsp).



# Examples of S-Parameter Simulation

This section contains examples for:

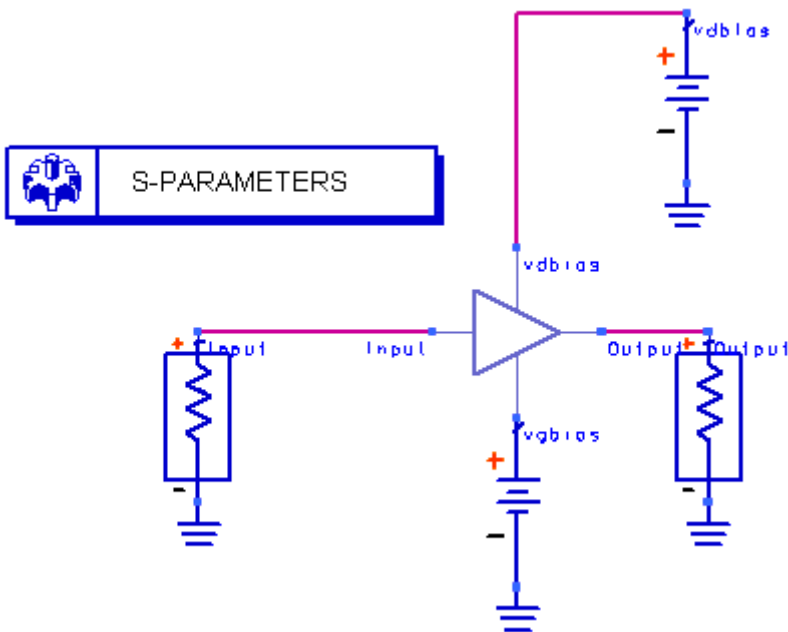
- [Simulating an Amplifier](#)
- [Calculating Group Delay](#)
- [Simulating Linear Noise](#)
- [Analyzing a Frequency Translating Network](#)
- [Eliminating Unwanted Effects](#)

These examples give detailed descriptions for setting up and running S-parameter simulations.

## Simulating an Amplifier

The following figure illustrates an example setup for performing a basic S-parameter simulation of an amplifier.

**Note** This design, *SP1*, is in the *Examples* directory under *Tutorial/SimModels\_wrk*. The results are in *SP1.dds*.



Example setup for a basic S-parameter simulation

To perform a basic S-parameter simulation:

1. From the **Simulation-S\_Param** palette, select a **Term** component for each port of the component or circuit to be simulated. You can edit the impedances as required,

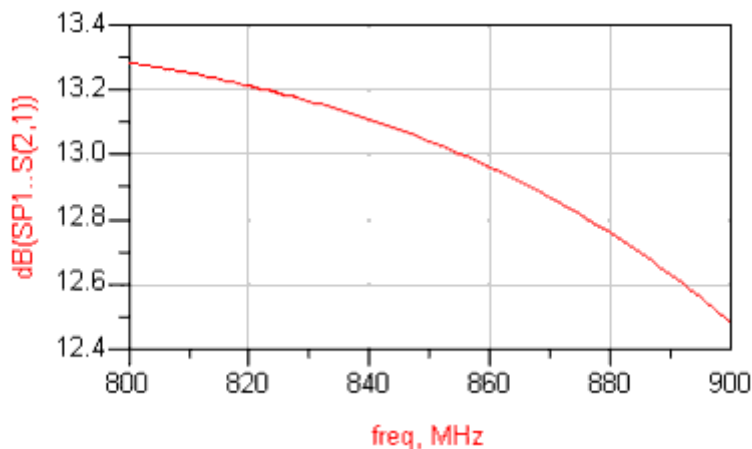
although the default value of 50 ohms is generally sufficient. Ensure that the terminations are properly connected to the component or circuit under test.

2. Ensure that the number of the input Term component is set to Num = **1**, and that of the output Term component to Num = **2**.

**Note**

By default, the Term component provides a noise contribution (Noise = yes), but is inactive unless noise contributions are requested. Also, ensure that the number of each Term component (as defined by the component's Num parameter) is appropriate to the location of the component in the circuit, to ensure that the S-parameter data is meaningful.

3. From the **Simulation-S\_Param** palette, select **SP**. Place this component on the schematic and select the **Frequency** tab. Ensure that *Start/Stop* is selected, then set the following parameters:
  - Sweep Type = **Linear**
  - Start = **800 MHz**
  - Stop = **900 MHz**
  - Step-size = **1 MHz**
4. To obtain S-parameters, select the **Parameters** tab and ensure that *S-parameters* is selected. For a description of the options on the Parameters tab, click **Help**. To obtain Y- (admittance) and Z- (impedance) parameters, select the corresponding options.
5. Click **OK** to accept changes and close the dialog box.
6. **Simulate**. When the simulation is finished, plot S(2,1) in the Data Display. The following is a plot of the gain ( $S_{21}$ ) versus frequency.



## Calculating Group Delay

By measuring the transit time, with respect to frequency, of a signal through the device under test, group delay is a useful measure of phase distortion in components such as amplifiers and filters.

To calculate group delay, you enable the *Group delay* option and, if desired, set the group delay aperture. These options are under the Parameters tab. The results appear in the Data Display variables list under *delay*.

For more information, refer to *Group Delay* (cktsimsp).

To calculate group delay:

1. Proceed as in [Simulating an Amplifier](#), setting frequencies and sweep parameters as needed.
2. Edit the S-Parameters component, select the **Parameters** tab, and enable **Group delay**.
3. Group delay aperture is an option that is found on network analyzers and behaves similarly here. The simulator sets the frequency aperture to 0.01% of the current frequency. To override the default frequency aperture, enable **Group delay aperture** and edit the value as needed.
4. Click **OK** to accept changes and close the dialog box.
5. **Simulate**. When the simulation is finished, plot the group delay data items, identified by the prefix *delay*. This is the absolute group delay, in seconds.



#### Hint

If the group delay data appears noisy, increase the value in the *Group delay aperture* field. If the results appear inaccurate, decrease the value. Generally, adjusting this value by a factor of 10 (in the appropriate direction) improves noisy or inaccurate results.

For an example of group delay data, see *Obtaining Group Delay Data* (cktsim).

## Simulating Linear Noise

Options for simulating linear noise are available from the *Noise* tab of the S-Parameters simulation component. For more information about how noise is calculated, refer to *S-Parameter Simulation Noise Analysis* (cktsimsp).

To simulate linear noise:

1. Proceed as in [Simulating an Amplifier](#), setting frequencies and sweep parameters as needed.
2. Edit the S-parameter Simulation component and select the **Noise** tab. Then select the **Calculate noise** option.
3. In the Edit field, enter the names of the nodes at which you want noise data to be reported.



#### Note

It is not necessary to name nodes if only noise figure is desired.

4. Use the Mode popup menu to sort the noise contributors (nodes) that are reported.
5. Either accept the default values for *Dynamic range to display* and *Bandwidth*, or edit these as required. The defaults are generally sufficient.
6. Click **OK** to accept changes and close the dialog box.
7. **Simulate**. When the simulation is finished, plot the noise data items. These are noise figure, identified as *nf[ port\_number ]*, and the equivalent input noise temperature, identified as *te[ port\_number ]*.

## Adjusting Noise Temperature

The IEEE definition of noise figure states that it should be measured at the standard noise temperature of 290 K (16.85°C). Advanced Design System uses this definition and value of the standard noise temperature in its calculation of noise figure. For a passive circuit, if the simulation temperature is not equal to this value, the noise figure will not be the same as the loss in decibels. The simulation temperature defaults to 25°C. It can be changed by adding an Options item to the cell and changing the simulation temperature there to 16.85°C.

## Analyzing a Frequency Translating Network

To simulate the effects of frequency translation (also known as frequency conversion) in circuits employing mixers, the S-parameter simulator uses the same algorithm as the AC simulation component. This option causes the simulator to consider the frequency not only of the input fundamental, but also the frequency of the resulting translations. A simple model is used to calculate the reference frequencies at each node.

Selecting the *Calculate noise* option (under the *Noise* tab) will result in frequency conversion data for nonlinear noise.

For more conversion information, refer to *S-Parameter Frequency Conversion* (cktsimsp).

To analyze a frequency translating network:

1. Proceed as in [Simulating an Amplifier](#).
2. Insert passive ports at locations where you want to obtain S-parameters.
3. Set frequencies and sweep parameters as needed.
4. Use a large-signal voltage or current source, such as V\_1Tone or I\_1Tone as the driving signal that causes the frequency translation (not a large-signal port source, such as a P\_1Tone).
5. Select the **Parameters** tab, then select **Enable AC frequency conversion**.
6. In the field labeled *S-parameter freq. conv. port*, enter **1**.



### Note

The frequency conversion port must be the number of the input port.

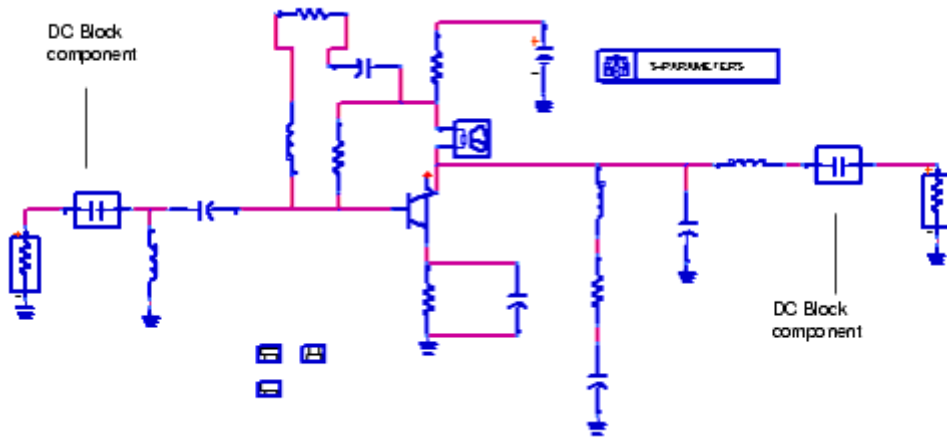
7. To calculate frequency conversion data for nonlinear noise, select the **Noise** tab and enable **Calculate noise**.

## Eliminating Unwanted Effects

It is sometimes helpful to reduce the contribution of other components in an analysis of a circuit involving, for example, amplifiers. The DC\_Block component functions as an open

during the DC part of the simulation (which is conducted automatically), while the DC\_Feed component functions as an open during the S-parameter simulation. This eliminates the loss that would otherwise be experienced with the Term and the bias resistors in the circuit. The following figure illustrates the use of the DC\_Block component in an example circuit.

**Note**  
This design, *Amp\_wBothMatches*, is in the *Examples* directory under *MW\_Ckts/LNA\_wrk*.



#### DC\_Block and DC\_Feed components in a circuit

To eliminate the effects of port and bias resistances:

From the **Lumped Elements** palette, select the **DC\_Block** and **DC\_Feed** components (as appropriate) and place them in the circuit as follows:

- Place the DC\_Block component (represented by a capacitor) between the ports and the device under test.
- Place the DC\_Feed components (represented by an inductor, not shown in this design) between the pins of the device under test and any bias resistors.

# S-Parameter Simulation Description

S-parameters are used to define the signal-wave response of an n-port electrical element at a given frequency. Detailed discussions of S-parameters can be found in standard textbooks on electrical circuit theory.

## Note

You may find it helpful to review the publication *S-Parameter Techniques for Faster, More Accurate Network Design* (AN 95-1), 5952-1130, available at <http://www.agilent.com/find/eesof-an95-1>.

S-parameter simulation is a type of small-signal AC simulation. It is most commonly used to characterize a passive RF component and establish the small-signal characteristics of a device at a specific bias and temperature.

If the circuit contains any nonlinear devices, a DC simulation is performed first. Following the DC bias simulation, the simulator linearizes all nonlinear devices about their bias points. A linearized model captures the small incremental changes of current due to small incremental changes of voltage. These are derivatives of the transistor model equations, which are evaluated at the DC bias point. Nonlinear resistors and current sources are replaced by linear resistors whose values are set by the small signal conductance  $dI/dV$ . Current sources that depend on voltages other than the voltage across the source are replaced by linear dependent current sources  $dI_1/dV_2$ . Nonlinear capacitors are replaced by linear capacitors of value  $dQ/dV$ .

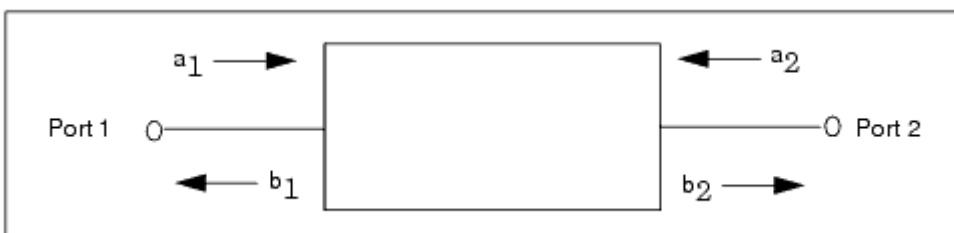
The linear circuit that results is analyzed as a multiport device. Each port is excited in sequence, a linear small-signal simulation is performed, and the response is measured at all ports in the circuit. That response is then converted into S-parameter data, which are in turn sent to the dataset. S-parameter simulation normally considers only the source frequency in a noise analysis. Use the *Enable AC Frequency Conversion* option if you also want to consider the frequency from a mixer's upper or lower sideband.

## Note

By selecting the appropriate option under the *Parameters* tab, it is possible to convert S-parameter data to Y- and Z-parameters. The S-parameter results are retained.

## S-Parameter Definitions

The following is a representation of a signal wave in a two-port electrical-element.



where

$a_1$  is the wave into port 1

$b_1$  is the wave out of port 1

$a_2$  is the wave into port 2

$b_2$  is the wave out of port 2

The S-parameters for this conventional element are defined in standard microwave textbooks as follows:

$$b_1 = a_1 s_{11} + a_2 s_{12}$$

$$b_2 = a_1 s_{21} + a_2 s_{22}$$

where

$s_{11}$  is the port-1 reflection coefficient:  $s_{11} = b_1/a_1$ ;  $a_2 = 0$

$s_{22}$  is the port-2 reflection coefficient:  $s_{22} = b_2/a_2$ ;  $a_1 = 0$

$s_{21}$  is the forward transmission coefficient:  $s_{21} = b_2/a_1$ ;  $a_2 = 0$

$s_{12}$  is the reverse transmission coefficient:  $s_{12} = b_1/a_2$ ;  $a_1 = 0$

These equations can be solved for  $b_1$  and  $a_1$  in terms of  $a_2$  and  $b_2$  to yield the transmission (T) parameters as follows:

$$b_1 = a_2 t_{11} + b_2 t_{12}$$

$$a_1 = a_2 t_{21} + b_2 t_{22}$$

The T-parameters are related to the S-parameters as follows:

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} = \begin{bmatrix} s_{12} - s_{11} & s_{22}/s_{21} & s_{11}/s_{21} \\ -s_{22}/s_{21} & 1/s_{21} \end{bmatrix}$$

S-parameters are defined with respect to a reference impedance that is typically 50 ohms. For 50-ohm S-parameters-with the 2-port element terminated with 50 ohms at each port - the  $s_{21}$  parameter represents the voltage gain of the element from port 1 to port 2.

## Group Delay

Group delay is a useful measure of phase distortion in components such as amplifiers and



filters. It measures the transit time, with respect to frequency, of a signal through the device under test.

The simulator calculates group delay by performing a finite difference of the phase response to obtain  $d\Phi/d\omega$ .

For more details on how the ADS S-parameter controller's group delay algorithm can be used with measured data, see the following support document:

[ADS Group Delay Calculation On Measured Data](#)

The simulator sets the frequency aperture to 0.01% of the current frequency. You can override this value by modifying the value in the *Group delay aperture* field, under the *Parameters* tab. This function is similar to that found on network analyzers, like the Agilent 8710. Refer also to these functions:

- delay
- dev\_lin\_phase
- diff
- phasedeg
- phaserad
- ripple (and GpDelRip measurement equation)
- unwrap
- volt\_gain

Descriptions are in *Measurement Expressions* (expmeas).

Group delay results are considered with respect to the input and output ports only. Results of group delay calculations include  $delay(2,1)$  and  $delay(1,2)$ , which can be viewed in the Data Display. These are absolute group delay, in seconds. For additional results data, add the measurement equations *dev\_lin\_phase* and *GpDelRip* to the schematic. Calculations from these equations will also be available in the Data Display.

## S-Parameter Frequency Conversion

S-parameter simulation normally allows only one frequency to be considered in a noise analysis - that of the source. This can be a disadvantage in obtaining simulation results for circuits involving mixers, which are inherently frequency-translating devices involving multiple frequencies. (See *Harmonic Balance for Mixers* (cktsimhb).)

As an aid in the simulation of frequencies involving mixer subnetworks, the option *Enable AC Frequency Conversion* (under the *Parameters* tab) causes the simulator to consider not only the frequency of the source but also that of one of the mixer's sidebands (which are defined by the user). Only the upper or lower sideband is considered, not both. The frequency-conversion results will appear in the dataset as for any nodes or probes placed to capture voltage or current data.

The S-parameter simulator uses the same conversion algorithm used by the AC Simulation component. For more information on this algorithm, see *Enabling Frequency Conversion* (cktsimac).


# S-Parameter Simulation Noise Analysis

During an S-parameter simulation, you can calculate these noise characteristics:

- Noise figure
- Noisy 2-port parameters

Each are discussed in the sections that follow.

For networks with more than two ports, the noise figure can be measured between two user-specified ports using Input Port and Output Port; the other ports are treated as resistors for the noise simulation.

 **Hint**  
To aid in noise figure measurements, use the noise circle measurement component, *NsCircle* or the function *ns\_circle*. See *ns\_circle()* (*expmeas*).

## Noise Figure

The parameter *nf(k)* in the dataset is the noise figure at output port *k*. You can view results using the Data Display. The noise figure for each port will appear in the variables list as *nf(1)*, *nf(2)*, and so on. *nf* displays noise figures for all ports.

When noise figure is calculated at a port, the other ports in the network are terminated in their respective impedances.

For a 2-port circuit, the noise figure is the signal-to-noise ratio at the input, divided by the signal-to-noise ratio at the output. It has units of dB. For a circuit with more than two ports, the noise figure is the ratio of the total noise at the output port to the transmitted input noise. The total noise is the transmitted input noise plus the noise contributed by the network. The transmitted input noise represents the portion of the incident thermal noise ( $kTB$ , where  $k$ =Boltzmann's constant,  $1.380658 \times 10^{-23}$  J/K,  $T=290$  K,  $B=1$  Hz) which passes through the system.

## Calculating Noise Figure

The common definition of noise factor is signal to noise at the input divided by the signal to noise at the output or

$$F = \frac{SNR_i}{SNR_o}$$

This definition describes the way that noise is computed in Advanced Design System.

The noise of the network without the ports is computed and denoted by

$$v_n^2$$

For more information on how the network noise is calculated, see *Linear Noise Simulation Description* (cktsimac).

The port noise is computed separately, and the noise figure equation is written as:

$$F = \frac{\frac{v_n^2}{R_L} + k T_o |S_{21}|^2}{k T_o |S_{21}|^2}$$

If there are multiple ports, the noise figure at output port  $k$  is generalized as:

$$F = \frac{\frac{v_n^2}{R_k} + \sum_{j \neq k}^N k T_o |S_{kj}|^2}{\sum_{j \neq k} k T_o |S_{kj}|^2}$$

Note that the output port noise is never included in the summation of noise sources. This definition makes it possible to generalize the noise figure calculation to the case of  $n$ -port networks, and in the limiting case of  $N=2$ , the calculation agrees with the classic 2-port definition.

For a circuit with more than two ports, the noise figure to all ports is calculated by default. For large circuits with many ports, this can be slow, especially when there is typically only one noise figure that is needed or meaningful. If the noise input port and noise output port are identified in the S-parameter simulation setup, only the one noise figure from the input port to the output port will be computed. In addition, the noisy two-port parameters ( $NF_{\min}$ ,  $R_N$ , and  $S_{\text{opt}}$ ) will be computed for these two ports. If either or both of the noise input port and noise output port are not specified, all of the noise figures will be calculated.

## Noisy 2-Port Parameters

$NF_{\min}$ ,  $R_n$  and  $S_{\text{opt}}$  are the noisy two-port parameters. They describe the noise properties

of a two-port and how the noise changes with respect to the source impedance. They describe circles of constant noise figure on the Smith chart.

$NF_{min}$  is the minimum noise figure that the circuit can produce, when the source has the optimum reflection coefficient  $S_{opt}$ .

$R_n$  is the noise resistance and controls how fast the noise increases as the source reflection coefficient changes from  $S_{opt}$ .

$$F = F_{min} + \frac{4r_n |\Gamma_{src} - \Gamma_{opt}|^2}{(1 - |\Gamma_{src}|^2) |1 + \Gamma_{opt}|^2}$$

where

$\Gamma_{opt} = S_{opt}$ , and  $\Gamma_{src}$  is the reflection coefficient of the source.

Reference: G Gonzalez, *Microwave Transistor Amplifiers*, Prentice-Hall, 1984, p 142.

# S-Parameters from Various Input-Output Modes

S-parameter results can be simulated for designs using various combinations of input and output modes. These modes include differential, common, and single-ended. Advanced Design System offers the following component and examples supporting various techniques for working with S-parameters:

- *SP\_Diff* is an instrument control component and is available in the *Simulation-Instruments* component library. See *SP\_Diff (Differential-Mode S-Parameters)* (cktsiminst).
- The example workspace *C:/Agilent/ADS2011\_01/designguides/projects/Wireline* contains the following designs and datasets that demonstrate simulations and mode-conversion equations. See the documentation for this workspace in *Examples > Wireline > Wireline Applications* then scroll to "Signal Integrity Simulations".
  - Common Mode Impedance Simulation is demonstrated in *ckt\_common\_imp\_ML\_thick\_metal*.
  - Differential Impedance Simulation is demonstrated in *ckt\_diff\_imp\_ML\_thick\_metal*.
  - Differential and Common Mode S-Parameter Basics is demonstrated in *mixed\_mode\_basics.dds*.

# Improving S-Parameter Simulation Speed

In ADS, you can increase simulation speed by including a LinearCollapse component in circuits where a Parameter Sweep controller is driving an S-parameter simulation.



LinearCollapse  
 LinearNetworkCollapse  
 NetworkRepresentation=S-Parameters

Library: [Simulation-S\\_Param](#) > [LinearNet](#)

## Parameters

Setup Dialog Name	Parameter Name	Description
NetworkRepresentation	NetworkRepresentation	Choose S-Parameters (default) or Y-Parameters to characterize the collapsed linear network.

The LinearCollapse component groups and collapses the linear devices that are not being swept. The collapsed network containing the linear devices is characterized at the frequency points of interest and the equivalent network parameter is used in its place.

The increase in simulation speed is due to the fact that the characterization of the collapsed network happens only once and the data is reused for the rest of the simulation. When a ParamSweep controller is driving an S-parameter analysis, it is common in a large network that most of the network is not being altered by the parameter sweep. By using a LinearCollapse component, instead of solving the linear network at each sweep point, the simulator solves the linear network once, then the simulator reuses the characterized data in place of the linear network.

When adding a LinearCollapse component:

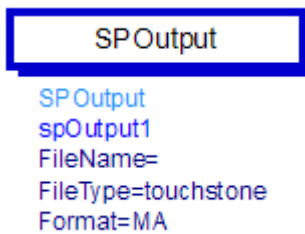
- Place the LinearCollapse component inside each sub-circuit that can be collapsed. You can place only one LinearCollapse component in a sub-circuit.
- Once the LinearCollapse components are placed inside the sub-circuits of interest, then continue with a normal S-parameter simulation.



# Saving Network Parameters into Other Formats

In ADS, you can save the network parameters produced by an S-parameter simulation into Touchstone, CITIfile or gmdif format. Simply place the *SPOutput* component in the schematic, specify the format and the file name to use for the data file, and the simulator will save the network parameters into that file.

The Touchstone file format does not support multidimensional data. Multidimensional data are those produced as a result of running sweeps. In this case, each sweep point is saved using a different file. Each file name is prepended with an index indicating the sweep number.



**Library: Simulation-S\_Param > SPOutput**

## Parameters

Setup Dialog Name	Parameter Name	Description
FileName	FileName	This is the name used for the file that will contain the network parameter data. Unless a complete path is specified, the file will be saved in the <i>data</i> directory.
FileType	FileType	This is the file format used to save the data. Allowed formats are: <ul style="list-style-type: none"> <li>• touchstone</li> <li>• citifile</li> <li>• gmdif</li> </ul>
Format	Format	This is the data format. The available formats are: <ul style="list-style-type: none"> <li>• MA (Magnitude Angle)</li> <li>• DB (Decibel)</li> </ul>

# S-Parameter Simulation Parameters

ADS provides access to S-parameters simulation parameters enabling you to define aspects of the simulation listed in the following table:

Tab Name	Description	For details, see...
Frequency	Sweep type and associated characteristics.	<a href="#">Setting Frequency Sweep</a>
Parameters	Provides options to set the following: <ul style="list-style-type: none"> <li>• Calculation of S-, Y-, or Z-parameters or group delay</li> <li>• Frequency conversion</li> <li>• Status levels for summary information</li> <li>• Device operating point information level</li> </ul>	<a href="#">Defining Simulation Parameters</a>
Noise	Parameters related to linear noise calculation (including port noise).	<a href="#">Defining Noise Parameters</a>
Output	Selectively save simulation data to a dataset.	For details, see <i>Selectively Saving and Controlling Simulation Data</i> (cktsim).
Display	Control the visibility of simulation parameters on the schematic.	For details, see <i>Displaying Simulation Parameters on the Schematic</i> (cktsim).

## Note

In ADS, the S-Parameters controller and the S-Parameter Test Lab controller both use the Scattering-Parameter Simulation dialog to set up a simulation. Use the following parameter information when using the setup dialog. For additional information about the S-Parameter Test Lab controller, see *S-Parameter Test Labs and Sequencer* (cktsim).

## Setting Frequency Sweep

Setting up the sweep portion of the simulation consists of two basic parts:

- Selecting the sweep type and setting the associated characteristics
- Optionally, specifying a sweep plan

## Note

In ADS, the S-Parameters controller and the S-Parameter Test Lab controller both use the Scattering-Parameter Simulation dialog to set up a simulation. Use the following parameter information when using the setup dialog. For additional information about the S-Parameter Test Lab controller, see *S-Parameter Test Labs and Sequencer* (cktsim).

To shorten simulation time in any parameter sweep, select a start point as close as possible to the convergence point and vary the parameter gradually. This yields better estimates for the next simulation, and achieves convergence more rapidly than if the parameter were changed abruptly. The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

### S-Parameters Simulation Frequency Sweep Parameters

Setup Dialog Name	Parameter Name	Description
Frequency		
Sweep Type-The sweep type and parameters (SweepVar="freq").		
Single point	Freq	Enables simulation at a single frequency point. Specify the desired value in the Frequency field.
Linear		Enables sweeping a range of values based on a linear increment. Click Start/Stop to set start and stop values for the sweep, or Center/Span to set the center value and a span of the sweep.
Log		Enables sweeping a range of values based on a logarithmic increment. Click Start/Stop to set start and stop values for the sweep, or Center/Span to set the center value and a span of the sweep.
Start/Stop Start, Stop, Step-size, Pts./decade, Num. of pts.	Start Stop Step Dec Lin	Select the Start/Stop option to sweep based on start, stop, step-size or pts./decade, and number of points. Linear sweep uses Step-size; Log sweep uses Pts./decade. <ul style="list-style-type: none"> <li>- Start-the start point of a sweep</li> <li>- Stop-the stop point of a sweep</li> <li>- Step-size-the increments at which the sweep is conducted</li> <li>- Pts./decade-number of points per decade</li> <li>- Num. of pts.-the number of points over which sweep is conducted</li> </ul>
Center/Span Center, Span, Step-size, Pts./decade, Num. of pts.	Center Span Step Dec Lin	Select the Center/Span option to sweep based on center and span, step-size or pts./decade, and number of points. Linear sweep uses Step-size; Log sweep uses Pts./decade. <ul style="list-style-type: none"> <li>- Center-the center point of a sweep</li> <li>- Span-the span of a sweep</li> <li>- Step-size-the increments at which the sweep is conducted</li> <li>- Pts./decade-number of points per decade</li> <li>- Num. of pts.-the number of points over which sweep is conducted</li> </ul>
Note: Changes to any of the Start, Stop, etc. fields causes the remaining fields to be recalculated automatically.		
Use sweep plan	SweepPlan	Enables use of an existing sweep plan component (SweepPlan). Select this option and enter the name of the plan or select it from the drop-down list.

## Defining Simulation Parameters

Defining the simulation parameters consists of the following basic parts:

- Calculation of S-, Y-, or Z-parameters, or group delay.
- Enabling the frequency conversion.
- Specifying the desired level of detail in the simulation status summary.
- Specifying the amount of device operating-point information to save.



**Note**

In ADS, the S-Parameters controller and the S-Parameter Test Lab controller both use the Scattering-Parameter Simulation dialog to set up a simulation. Use the following parameter information when using the setup dialog. For additional information about the S-Parameter Test Lab controller, see *S-Parameter Test Labs and Sequencer* (cktsim).

The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

**S-Parameters Simulation Parameters**

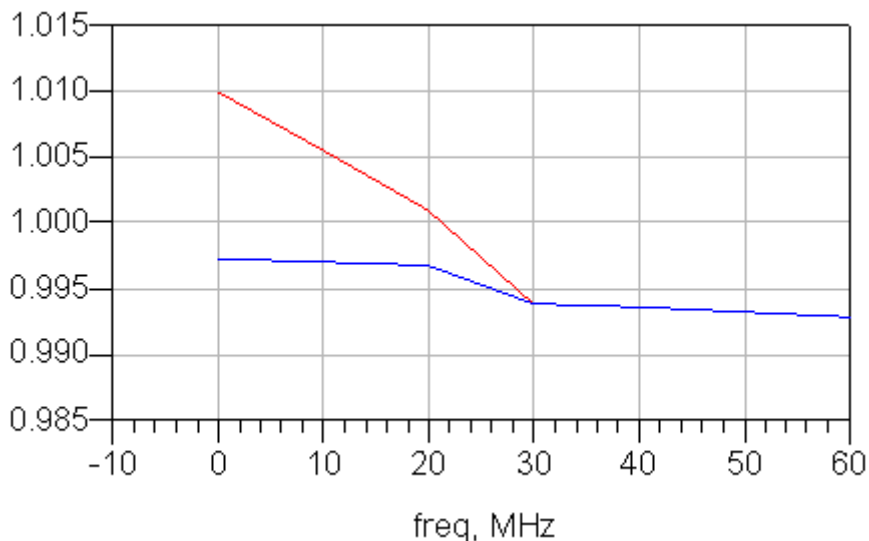
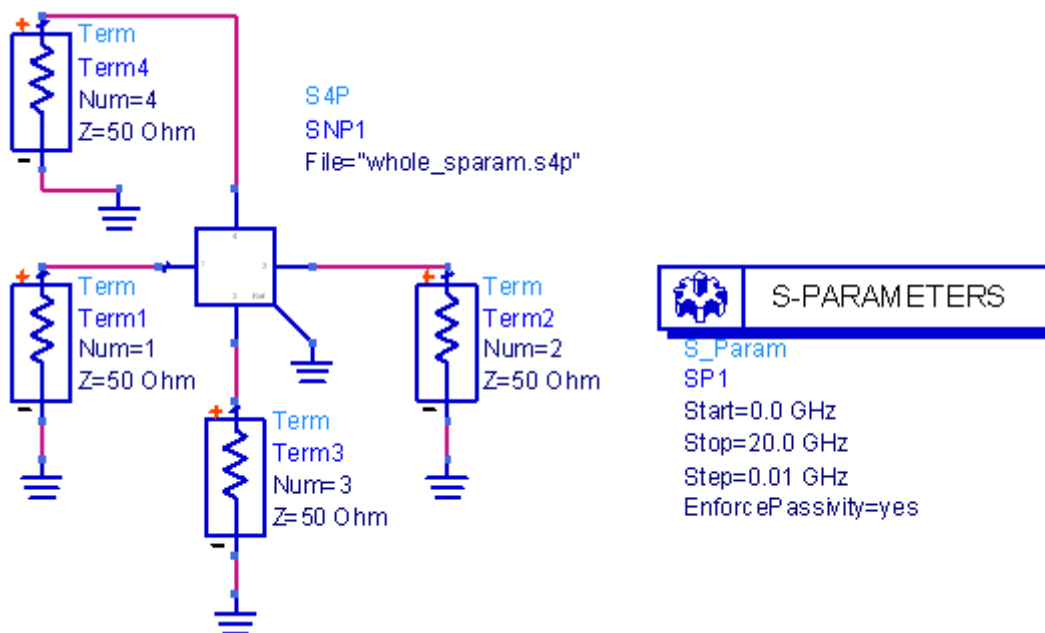
Setup Dialog Name	Parameter Name	Description
Calculate		
S-parameters	CalcS	Causes S-parameters to be calculated.
Enforce Passivity	EnforcePassivity	Select to enforce passivity in S-parameter calculation. Default=no (unselected). When selected, S-parameter passivity will be enforced. This option only applies to S-parameter calculation and is not active for Y- and Z-parameters, even if they are selected. Please see the section "Passivity Correction in S-parameter Simulation" below for more information on using this feature.
Y-parameters	CalcY	Converts the results of an S-parameter simulation to Y-parameters. The S-parameters are also output.
Z-parameters	CalcZ	Converts the results of an S-parameter simulation to Z-parameters. The S-parameters are also output.
Group delay Group delay aperture	CalcGroupDelay	Causes a group delay simulation to be calculated from the S-parameter data. Enter a value for <i>Group delay aperture</i> which is the frequency aperture (as a percentage of actual frequency) over which $d\Phi/d\omega$ is calculated.
Frequency Conversion		
Enable AC frequency conversion	FreqConversion	Select this option to enable AC frequency conversion. For <i>S-parameter frequency conv. port</i> (FreqConversionPort) add a numeric value to enable S-parameter frequency.
Levels		Enables you to set the level of detail in the simulation status report.
Status level	StatusLevel	Prints information about the simulation in the Status/Summary part of the Message Window. - 0 reports little or no information, depending on the simulation engine. - 1 and 2 yield more detail. - Use 3 and 4 sparingly since they increase process size and simulation times considerably. The type of information printed may include the sum of the current errors at each circuit node, whether convergence is achieved, resource usage, and where the dataset is saved. The amount and type of information depends on the status level value and the type of simulation.
Device operating point level	DevOpPtLevel	Enables you to save all the device operating-point information to the dataset. If this simulation performs more than one S-parameter analysis (from multiple S-parameter controllers), the device operating point data for all S-parameter analyses will be saved, not just the last one. Default setting is None.
None	=None	No information is saved.
Brief	=Brief	Saves device currents, power, and some linearized device parameters.
Detailed	=Detailed	Saves the operating point values which include the device's currents, power, voltages, and linearized device parameters.

## Passivity Correction in S-parameter Simulation

When EnforcePassivity=yes, passivity will be enforced following S-parameters calculation. The S-parameters saved to the dataset will be passive. Passivity is enforced at each frequency sweep point. The method works in this order:

1. Calculating the largest eigenvalue of the S-parameter matrix at each frequency point.
2. Scaling the S-parameter matrix at each offending frequency point such that the largest eigenvalue equals  $1 - \epsilon_{ps}$ , where  $\epsilon_{ps}$  is set to  $1e-6$ .

The following example demonstrates how a non-passive 4-port S-parameter file, in Touchstone format, can be made passive using S-parameter simulation. The original S-parameter data is imported to an S4P block and an S-parameter simulation is performed with `EnforcePassivity=yes`. The plot shows the resultant  $S(2,1)$  (in blue) in comparison with the original non-passive  $S(2,1)$  (in red). The difference shows the effect of passivity correction. Note that the resulting dataset contains passive S-parameters, but that the original Touchstone data remains unchanged. If needed, the Data File tool can be used to convert the dataset to a new, passive Touchstone file. For information about using the Data File tool, see *Working with Data Files* (cktsim) in the *Using Circuit Simulators* (cktsim) documentation.



## Defining Noise Parameters

Defining the noise parameters consists of the following basic parts:

- Enabling noise calculation.
- Specifying the nodes to use for noise parameter calculation.
- Specifying the noise contributors and the threshold for noise contribution.
- Optionally, specifying the bandwidth over which the noise simulation is performed.



**Note**

In ADS, the S-Parameters controller and the S-Parameter Test Lab controller both use the Scattering-Parameter Simulation dialog to set up a simulation. Use the following parameter information when using the setup dialog. For additional information about the S-Parameter Test Lab controller, see *S-Parameter Test Labs and Sequencer* (cktsim).

The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

### S-Parameters Simulation Noise Parameters



Setup Dialog Name	Parameter Name	Description
Calculate noise	CalcNoise	Enables linear noise simulation. This option must be selected for the remaining noise options to be available.
Noise input port	NoiseInputPort	The number of the source port at which noise is injected. This is commonly the input port. Although any valid port number can be used, the input port number is frequently defined as Num=1.
Noise output port	NoiseOutputPort	The number of the Term component at which noise is retrieved. This is commonly the output port. Although any valid port number can be used, the output port number is frequently defined as Num=2.
<p>Noise contributors-Use this area to sort the noise contributors list and to select a threshold below which noise contributors will not be reported. A list shows how each component contributes to noise at a specific node. The noise contributor data are always in units of V/sqrt(Hz) for noise voltages, and A/sqrt(Hz) for noise currents; they do not scale with noise bandwidth.</p>		
Mode	SortNoise	Provides options for sorting noise contributors by value or name.
Off	=Off	Causes no individual noise contributors to be selected. The result is simply a value for total noise at the output.
Sort by value	=Sort by value	Sorts individual noise contributors, from largest to smallest, that exceed a user-defined threshold (see below). The subcomponents of the nonlinear devices that generate noise (such as Rb, Rc, Re, Ib, and Ic in a BJT) are listed separately, as well as the total noise from the device.
Sort by name	=Sort by name	Causes individual noise contributors to be identified and sorts them alphabetically. The subcomponents of the nonlinear devices that generate noise (such as Rb, Rc, Re, Ib, and Ic in a BJT) are listed separately, as well as the total noise from the device.
Sort by value with no device details	=Sort by value with no device details	Sorts individual noise contributors, from largest to smallest, that exceed a user-defined threshold (see below). Unlike <i>Sort by value</i> , only the total noise from nonlinear devices is listed without any subcomponent details.
Sort by name with no device details	=Sort by name with no device details	Causes individual noise contributors to be identified and sorts them alphabetically. Unlike <i>Sort by name</i> , only the total noise from nonlinear devices is listed without subcomponent details.
Dynamic range to display	NoiseThresh	A threshold below the total noise, in dB, that determines what noise contributors are reported. All noise contributors less than this threshold will be reported. For example, assuming that the total noise voltage is 10 nV, a setting of 40 dB (a good typical value) ensures that all noise contributors up to 40 dB below 10 nV (that is, up to 0.1 nV) are reported. The default of 0 dB causes all noise contributors to be reported. This parameter is only used with <i>Sort by value</i> and <i>Sort by value with no device details</i> .
Bandwidth	BandwidthForNoise	The bandwidth over which the noise simulation is performed. 1 Hz is the recommended bandwidth for measurements of spectral noise power. The noise contributor data do not scale with noise bandwidth.