Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms that apply to this software product is available upon request from your Agilent Technologies representative.

Restricted Rights Legend

Use, duplication or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Agilent Technologies
395 Page Mill Road
Palo Alto, CA 94304 U.S.A.

Copyright © 1998-2004, Agilent Technologies. All Rights Reserved.

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries.

Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® 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.

Java™ is a U.S. trademark of Sun Microsystems, Inc.
Contents

1 Circuit Components
   Introduction ................................................................. 1-1
   BinModel (Bin Model for Automatic Model Selection) .............. 1-2
   FORMAT A, B, C, D, E Drawing Formats ................................ 1-6
   Ground (Ground Component) ............................................ 1-7
   Port (Port Component) .................................................... 1-8
   Term (Port Impedance for S-parameters) .............................. 1-9
   Multiplicity Parameter _M ................................................ 1-10
   Series IV or MDS Product Migration Components ................. 1-11

2 Lumped Components
   C (Capacitor) ............................................................... 2-2
   C_Model (Capacitor Model) .............................................. 2-4
   C_Conn (Capacitor (Connector Artwork)) ............................ 2-6
   C_dxdy (Capacitor (Delta X - Delta Y)) ............................ 2-7
   C_Pad1 (Capacitor (Pad Artwork)) ................................... 2-8
   C_Space (Capacitor (Space Artwork)) ............................... 2-9
   CAPP2_Conn (Chip Capacitor (Connector Artwork)) ............... 2-10
   CAPP2_Pad1 (Chip Capacitor (Pad Artwork)) ...................... 2-12
   CAPP2_Space (Chip Capacitor (Space Artwork)) ................... 2-14
   CAPP (Capacitor with Q) ................................................. 2-16
   CQ_Conn (Capacitor with Q (Connector Artwork)) ................. 2-18
   CQ_Pad1 (Capacitor with Q (Pad Artwork)) ....................... 2-20
   CQ_Space (Capacitor with Q (Space Artwork)) .................... 2-22
   DC_Block (DC Block) ..................................................... 2-24
   DC_Feed (DC Feed) ....................................................... 2-25
   DICAP (Dielectric Laboratories Di-cap Capacitor) ................. 2-26
   DILABMLC (Dielectric Laboratories Multi-Layer Chip Capacitor) 2-27
   INDQ (Inductor with Q) .................................................. 2-29
   InDQ2 (Inductor with Q) ................................................ 2-31
   L (Inductor) ................................................................. 2-33
   L_Conn (Inductor (Connector Artwork)) ............................. 2-35
   L_Model (Inductor Model) .............................................. 2-36
   L_Pad1 (Inductor (Pad Artwork)) .................................... 2-38
   L_Space (Inductor (Space Artwork)) ................................ 2-39
   LQ_Conn (Inductor with Q (Connector Artwork)) ................. 2-40
   LQ_Pad1 (Inductor with Q (Pad Artwork)) ......................... 2-42
   LQ_Space (Inductor with Q (Space Artwork)) ..................... 2-44
   Mutual (Mutual Inductor) .............................................. 2-46
   PLC (Parallel Inductor-Capacitor) ................................... 2-48
PLCQ (Parallel Inductor-Capacitor with Q) ............................................................... 2-49
PRC (Parallel Resistor-Capacitor) ............................................................................. 2-51
PRL (Parallel Resistor-Inductor) .............................................................................. 2-52
PRLC (Parallel Resistor-Inductor-Capacitor) .......................................................... 2-53
R (Resistor) .............................................................................................................. 2-54
R_Model (Resistor Model)........................................................................................ 2-56
R_Conn (Resistor (Connector Artwork)) .................................................................. 2-60
R_dxdy (Resistor (Delta X - Delta Y))....................................................................... 2-61
R_Pad1 (Resistor (Pad Artwork)) ............................................................................. 2-62
R_Space (Resistor (Space Artwork)) ....................................................................... 2-63
Short (Short)............................................................................................................. 2-64
SLC (Series Inductor-Capacitor) .............................................................................. 2-65
SLCQ (Series Inductor-Capacitor with Q) ............................................................... 2-66
SMT_Pad (SMT Bond Pad) ...................................................................................... 2-68
SRC (Series Resistor-Capacitor).............................................................................. 2-70
SRL (Series Resistor-Inductor)................................................................................. 2-71
SRLC (Series Resistor-Inductor-Capacitor) ............................................................. 2-72
TF (Transformer)....................................................................................................... 2-73
TF3 (3-Port Transformer) ........................................................................................ 2-74

3 Miscellaneous Circuit Components
CAPP2 (Chip Capacitor)........................................................................................... 3-2
CIND (Ideal Torroidal Inductor)................................................................................. 3-4
RIND (Rectangular Inductor) .................................................................................... 3-5
XFERP (Physical Transformer)................................................................................. 3-7
XFERRUTH (Ruthroff Transformer).......................................................................... 3-9
XFERTAP (Tapped Secondary Ideal Transformer).................................................... 3-11

4 Probe Components
Counter (Counter Component) ................................................................................. 4-2
I_Probe (Current Probe) ........................................................................................... 4-3
OscPort (Grounded Oscillator Port).......................................................................... 4-4
OscPort2 (Differential Oscillator Port)...................................................................... 4-7
OscTest (Grounded Oscillator Test).......................................................................... 4-14
SProbe (SProbe Component).................................................................................... 4-17
SProbePair (SProbePair Component)........................................................................ 4-19
TimeDelta (Time Delta Component)......................................................................... 4-21
TimeFrq (Time Frequency Component).................................................................. 4-23
TimePeriod (Time Period Component).................................................................... 4-24
TimeStamp (Time Stamp Component)................................................................... 4-25
WaveformStats (WaveformStats Component)....................................................... 4-26

5 Linear Data File Components
Deembedded 1 (1-Port De-Empbed Data File) .................................................. 5-18
Deembedded 2 (2-Port De-Embed Data File)...................................................... 5-20
Netlist Include (Netlist File Include Component)........................................... 5-23
S1P (1-Port S-parameter File) .......................................................................... 5-26
S2P (2-Port S-parameter File) .......................................................................... 5-28
S2P_Conn (2-Port S-parameter File; connector artwork)................................... 5-31
S2P_Pad3 (2-Port S-parameter File; pad artwork)............................................. 5-33
S2PMDF (Multi-Dimensional 2-Port S-parameter File) .................................... 5-35
S2P_Spac (2-Port S-parameter File) .................................................................. 5-38
S3P (3-Port S-parameter File) .......................................................................... 5-40
S4P (4-Port S-parameter File) .......................................................................... 5-43
S5P to S9P (5-Port to 9-Port S-parameter File) ............................................... 5-46
S10P to S20P (10-Port to 20-Port S-parameter File) ....................................... 5-49
S21P to S99P (21-Port to 99-Port S-parameter File) ........................................ 5-52
SnP component (n>99) .................................................................................. 5-55
VAR (Variables and Equations Component)................................................... 5-59

6 Equation-Based Linear Components
Chain (2-Port User-Defined Linear Chain) ...................................................... 6-2
Hybrid (2-Port User-Defined Linear Hybrid) .................................................... 6-4
S1P_Eqn to S6P_Eqn (1- to 6-Port S-parameters, Equation-Based) ............... 6-6
Y1P_Eqn to Y6P_Eqn (1- to 6-Port Y-parameters, Equation-Based) ............... 6-9
Z1P_Eqn to Z6P_Eqn (1- to 6-Port Z-parameters, Equation-Based) ............... 6-11

Index
Chapter 1: Circuit Components

Introduction

The Advanced Design System Circuit Components catalog provides component information. Chapters in this book are organized by component types; components are arranged alphabetically within each chapter.

Chapter 1 provides information for these common items:

- BinModel component for automatic model selection
- Ground, Port, and Term components
- Drawing formats (design sheets)
- Multiplicity parameter _M to scale components or entire sub-circuits containing multiple components and sub-circuits
- VAR component to define variables and equations
- Series IV and MDS components that can be used in ADS
BinModel (Bin Model for Automatic Model Selection)

Symbol

Available in ADS and RFDE
Supported via model include file in RFDE

Parameters
None

Notes/Equations

1. This feature is available for use with the BJT, Diode, GaAs, JFET, and MOS models and is provided in the library for each respective model.

2. BinModel allows you to sweep a parameter (usually a geometry, such as gate length), then enable the simulator to automatically select between different model cards. If a circuit contains nonlinear devices for circuit simulation, each device should be associated with one device model through schematic or netlist editing. However, modern processes require multiple models for different device sizes to improve simulation accuracy. For example, as illustrated here, a model (Model 2) that is accurate for a 4u channel length MOSFET is not necessarily a good model (Model 1) for a 1u channel length. If mixed analog and digital circuits are combined in a single part, multiple models are the easiest way to create high accuracy over a wide range of device sizes.
3. Depending on device size, one of the multiple models should be selected for a device at simulation time. If device size needs to be varied over a certain range, manual model change for each new device size would be very cumbersome. The model binning feature automatically searches for a model with the size range that covers the device size and uses this model in simulation.

4. Following is a generalized example of the use of Bin Model.

The BinModel window appears when you click the BinModel instance placed in a design in the Schematic window. In this example, the value Area was typed into the Param box of the dialog box, as shown here, and two BJT devices instances from the same schematic design were entered in the tabular listing, with desired minimum and maximum values for Area also identified.

5. In the corresponding BJ TM1 instance in the schematic, the Bf parameter was set to 100, and in BJ TM2 it was set to 50.
Circuit Components

6. In the device model placed in the schematic (for example, BJT NPN), the first bin model to be used for simulation was identified (Model = BinModel1) and the AREA parameter was set to 25.

7. The design was simulated, then the command Simulate > Annotate DC Box was selected. In the Schematic window, the value 100uA appeared near the device symbol in the schematic.

8. The process was repeated for the BJTM2 model, with Model = BinModel2, and the AREA parameter set to 35. The design was simulated, then the command Simulate > Annotate DC Box was selected. In the Schematic window, the value 50.0uA appeared near the device symbol in the schematic. The data display window was opened, with a List chart chosen, and I_Probe1.i measurement selected, allowing us to compare the results of the bin models associated with the separate simulations of BinModel1 and BinModel2.

<table>
<thead>
<tr>
<th>Bin</th>
<th>I-Probe1.i</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.000</td>
<td>100.0uA</td>
</tr>
<tr>
<td>35.000</td>
<td>50.0uA</td>
</tr>
</tbody>
</table>

9. Two more BJT models were added to the schematic, with Bf parameter set to 25 and 10, respectively. We allowed the third and fourth models to be selected for a device with Area from 40 to 50 and 50 to 60.

<table>
<thead>
<tr>
<th>BinModel Parameter Table Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Model</td>
</tr>
<tr>
<td>BJTM1</td>
</tr>
<tr>
<td>BJTM2</td>
</tr>
<tr>
<td>BJTM3</td>
</tr>
<tr>
<td>BJTM4</td>
</tr>
</tbody>
</table>

10. The circuit was simulated to perform parameter sweep over Area from 25 to 55 with steps of 10.
11. The four results were then compared in the data display window.

<table>
<thead>
<tr>
<th>Bin</th>
<th>I_Probe1.i</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.000</td>
<td>100.0uA</td>
</tr>
<tr>
<td>35.000</td>
<td>50.0uA</td>
</tr>
<tr>
<td>45.00</td>
<td>25.0uA</td>
</tr>
<tr>
<td>55.00</td>
<td>10.0uA</td>
</tr>
</tbody>
</table>

12. Buttons beneath the table function as follows:

- **Add Model** adds additional rows to the Model column for specification of more models
- **Add Param** adds additional entry boxes to the Param field for specification of more parameters
- **Delete Model** deletes a selected model
- **Delete Param** deletes a selected parameter
Circuit Components

FORMAT A, B, C, D, E Drawing Formats

Drawing

Available in ADS

Notes/Equations

1. The Drawing Formats library provides popular sheet sizes (in inches): A (8.5×11), B (11×17), C (17×22), D (22×34), E (34×44).

2. Turn on Drawing Format filter through Options > Preferences > Select. You can then move or delete the drawing sheet. Turn off the filter when not needed.
Ground (Ground Component)

Symbol

▲

Available in ADS and RFDE

Parameters

None

Notes/Equations

1. When you place a ground, position the pin directly on the end of the pin or wire to which you are connecting.
Circuit Components

**Port (Port Component)**

**Symbol**

![Port Symbol]

**Available in**  
ADS

**Parameters**

- **Num** = port number (value type: integer)
- **layer** = (ADS Layout option) layer to which port is mapped

**Notes/Equations**

1. Port is the standard port component offered and used to define networks.
2. When you place a port, position the square directly on the end of the pin or wire to which you are connecting.
3. The number of ports of a network is the same as the number of Port components connected in it.
4. The port number is supplied by the program and automatically incremented each time a Port is placed. However, you can override the program-supplied value by typing in an integer value of your choice. Port numbers must be consecutive; and, port numbers of multiple Port components cannot be the same.
**Term (Port Impedance for S-parameters)**

**Symbol**

![Symbol](image)

**Available in** ADS and RFDE

**Parameters**

- **Num** = port number (value type: integer)
- **Z** = reference impedance, in ohms
- **Noise** = enable/disable port thermal noise: YES, NO (for AC or harmonic balance analysis only; not for S-parameter analysis)
- **Vdc** = open circuit dc voltage, in volts

**Notes/Equations**

1. Term is used in ac and S-parameter simulations. For S-parameter simulations it is used to define the impedance and location of the ports. When not in use, it is treated as an impedance with the value R + J X. The reactance is ignored for dc simulations.
Circuit Components

Multiplicity Parameter \( _M \)

The multiplicity feature provides a way to scale components or entire sub-circuits containing many components and sub-circuits. Given a component with a multiplicity value \( M \), the simulator treats this component as if there were \( M \) such components all connected in parallel. Sub-circuits within sub-circuits will be appropriately scaled.

The \( _M \) parameter is available at the component level as shown here. (For components that don't explicitly have a Multiplicity parameter, the same functionality can be achieved by placing the component in a sub-circuit and using the sub-circuit's Multiplicity parameter, as described next.)

For sub-circuits, the parameter is enabled by selecting **File > Design Parameters** from the Schematic window. In the dialog box, select the **Parameters** tab. To add the Multiplicity parameter, choose **Add Multiplicity Factor \(_M\)**.
Series IV or MDS Product Migration Components

Older Series IV or MDS components can still be placed in ADS designs. While they are not accessible from the component library, they can be placed in a Schematic window by entering the exact component name in the Component History field above the design area, pressing Enter, and moving the cursor into the design area. Documentation is not provided for these components.

- **Series IV Components**
  - GAIN
  - PULSE_TRAM

- **Spectral Sources**
  - GMSK_SOURCE
  - PIQPSK_SOURCE
  - QAM16_SOURCE
  - QPSK_SOURCE

- **Wideband Modems**
  - AM_DemodBroad
  - AM_ModBroad
  - FM_DemodBroad
  - FM_ModBroad
  - IQ_ModBroad
  - QAM_ModBroad
  - QPSK_ModBroad
  - PM_DemodBroad
  - PM_ModBroad

- **MDS Components**
  - CPWTL_MDS
  - GCPWTL_MDS
  - CPWCTL_MDS
Circuit Components

• ACPW_MDS
• ACPWTL_MDS
• CPWTLF_MDS
• MSACTL_MDS
• MS3CTL_MDS, MS4CTL_MDS, MS5CTL_MDS
• MSABND_MDS
• MSBEND_MDS
• MSOBND_MDS
• MSCRNR_MDS
• MSTRNL2_MDS
• MSCTL_MDS
• MSCROSS_MDS
• MSRBND_MDS
• MSGAP_MDS
• MSAGAP_MDS
• MSIDCF_MDS
• MSIDC_MDS
• MSLANGE_MDS
• MSTL_MDS
• MSOC_MDS
• MSSPLC_MDS
• MSSPLS_MDS
• MSSPLR_MDS
• MSSTEP_MDS
• MSRTL_MDS
• MSSLIT_MDS
• MSTAPER_MDS
• MSTEE_MDS
• TFC_MDS
• MSWRAP_MDS
• TFR_MDS
• MSVIA_MDS
• MSSVIA_MDS
• MLACRNR1
• MLCRNR1
• MLRADIAL1
• MLSLANTED1
• MLCROSSOVER1
• SLTL_MDS
• SLOC_MDS
• SLCTL_MDS
• SL3CTL_MDS, SL4CTL_MDS, SL5CTL_MDS
• SLUCTL_MDS
• SLGAP_MDS
• SLSTEP_MDS
• SLTEE_MDS
• SLOBND_MDS
• SLCNR_MDS
• SLRBND_MDS
• SLABND_MDS
• SLUTL_MDS
• SSTL_MDS
• SSCTL_MDS
• SS3CTL_MDS, SS4CTL_MDS, SS5CTL_MDS
Circuit Components

- SSSPLC_MDS
- SSPLS_MDS
- SSPLR_MDS
- SSLANGE_MDS
- SSTFR_MDS
- BRCTL_MDS
- BR0CTL_MDS, BR3CTL_MDS, BR4CTL_MDS
- CTL_MDS
- COAX_MDS
- DRC_MDS
- TL_MDS
- TLOC_MDS
- RWGTL_MDS
- FINLINE_MDS
- ETAPER_MDS
- SLOTTL_MDS
- RIBBONG_MDS
- RIBBONS_MDS
- WIREG_MDS
- WIRES_MDS
Chapter 2: Lumped Components
Lumped Components

C (Capacitor)

Symbol

Available in ADS and RFDE

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>capacitance</td>
<td>fF, pF, nF, uF, mF</td>
<td>1.0 pF</td>
</tr>
<tr>
<td>Temp</td>
<td>temperature</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>Trise</td>
<td>temperature rise over ambient</td>
<td>°C</td>
<td>0</td>
</tr>
<tr>
<td>Tnom</td>
<td>nominal temperature</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>TC1</td>
<td>linear temperature coefficient</td>
<td>1/°C</td>
<td></td>
</tr>
<tr>
<td>TC2</td>
<td>quadratic temperature coefficient</td>
<td>1/°C^2</td>
<td></td>
</tr>
<tr>
<td>wBV</td>
<td>breakdown voltage warning</td>
<td>fV, pV, nV, uV, mV, V</td>
<td></td>
</tr>
<tr>
<td>InitCond</td>
<td>initial condition voltages for transient analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td>name of a capacitor model to use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>physical width for use with a model</td>
<td>um, mm, cm, meter, mil, in</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>physical length for use with a model</td>
<td>um, mm, cm, meter, mil, in</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>number of capacitors in parallel</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Notes/Equations

1. The capacitor value can be made a function of temperature by setting Tnom and either TC1 or TC2 or both. Tnom specifies the nominal temperature at which C is given. Tnom defaults to 25°C. If Temp≠Tnom, then the simulated capacitance value is given by:

\[
C' = C \times [1 + TC1(T_{\text{Temp}} - T_{\text{nom}}) + TC2(T_{\text{Temp}} - T_{\text{nom}})^2] 
\]

2. If Temp is not explicitly specified, it defaults to the global temperature specified in the options item.

3. wBV is used by the overload alert feature. It sets a limit on the maximum voltage across the capacitor. If this limit is specified, the simulator will issue a
warning the first time it is exceeded during a dc, harmonic balance or transient simulation. Simulation results are not affected by this parameter.

4. If a model name is given, then values that are not specified on the capacitor instance are taken from the model values. Typical values that can be defaulted are capacitance, length and width, nominal temperature, temperature coefficients, and overload alert parameters.

If a model is used, the capacitance value to be simulated (before temperature scaling is applied) is calculated as:

\[
C' = C - C_j \times (\text{Length} - 2 \times \text{Narrow}) \times -(\text{Width} - 2 \times \text{Narrow}) \\
+ C_{\text{Jsw}} \times 2 \times (\text{Length} + \text{Width} - 4 \times \text{Narrow})
\]

5. \( M \) is used to represent the number of capacitors in parallel and defaults to 1. If a capacitor model is used, an optional scaling parameter Scale can also be defined on the model; it defaults to 1. The effective capacitance that will be simulated is \( C \times \text{Scale} \times M \).

6. When InitCond is explicitly specified, the check-box Use user-specified initial conditions must be turned on in the Convergence tab of the Tran transient simulation controller for the parameter setting to take effect.
**C_Model (Capacitor Model)**

**Symbol**

![Capacitor symbol]

**Available in** ADS and RFDE  
Supported via model include file in RFDE

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>default fixed capacitance</td>
<td>fF, pF, nF, uF, mF</td>
<td>1.0 pF</td>
</tr>
<tr>
<td>Cj</td>
<td>junction capacitance</td>
<td>farads/meter²</td>
<td></td>
</tr>
<tr>
<td>Cjsw</td>
<td>sidewall or periphery capacitance</td>
<td>farads/meter</td>
<td></td>
</tr>
<tr>
<td>Length (L)</td>
<td>default length</td>
<td>um, mm, cm, meter, mil, in</td>
<td></td>
</tr>
<tr>
<td>Width (W)</td>
<td>default width</td>
<td>um, mm, cm, meter, mil, in</td>
<td></td>
</tr>
<tr>
<td>Narrow (Etch)</td>
<td>length and width narrowing due to etching</td>
<td>um, mm, cm, meter, mil, in</td>
<td></td>
</tr>
<tr>
<td>Tnom</td>
<td>nominal temperature</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>Trise</td>
<td>temperature rise over ambient</td>
<td>°C</td>
<td>0</td>
</tr>
<tr>
<td>TC1</td>
<td>linear temperature coefficient</td>
<td>1/°C</td>
<td></td>
</tr>
<tr>
<td>TC2</td>
<td>quadratic temperature coefficient</td>
<td>1/°C²</td>
<td></td>
</tr>
<tr>
<td>wBV</td>
<td>breakdown voltage (warning)</td>
<td>fV, pV, nV, uV, mV, V</td>
<td></td>
</tr>
<tr>
<td>Scale (Scalec)</td>
<td>capacitance scaling factor</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Coeffs</td>
<td>nonlinear capacitor polynomial coefficients</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AllParams</td>
<td>DataAccessComponent-based parameters</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Netlist Format**

Model statements for the ADS circuit simulator may be stored in an external file. This is typically done with foundry model kits. For more information on how to set up and use foundry model kits, refer to the Design Kit Development manual.

```
model modelname C_Model [parm=value] *
```

The model statement starts with the required keyword model. It is followed by the model name that will be used by capacitor components to refer to the model. The third parameter indicates the type of model; for this model it is C_Model. The rest of the
model contains pairs of model parameters and values, separated by an equal sign. The name of the model parameter must appear exactly as shown in the parameters table—these names are case sensitive. Some model parameters have aliases, which are listed in parentheses after the main parameter name; these are parameter names that can be used instead of the primary parameter name. Model parameters may appear in any order in the model statement. Model parameters that are not specified take the default value indicated in the parameters table. For more information about the ADS circuit simulator netlist format, refer to “ADS Simulator Input Syntax” in the Circuit Simulation manual.

Example:

```plaintext
model mimCap C_Model \
  Cje=1e-3 Cjsw=1e-10 Tc1=-1e-3 
  Coeffs=list(1,2)
```

Notes/Equations

For RFDE Users Information about this model must be provided in a model file; refer to the Netlist Format section.

1. This model supplies values for a capacitor C. This allows physically-based capacitors to be modeled based on length and width.

2. Use AllParams with a.DataAccessComponent to specify file-based parameters (refer to DataAccessComponent). Note that model parameters that are explicitly specified take precedence over those specified via AllParams.

3. The capacitor value can be made a nonlinear function of the applied voltage \( V \) by specifying the polynomial coefficients list (\( \text{Coeffs} = \text{list}(c_1, c_2, c_3, \ldots) \)). The capacitance value \( C(V) \) is then given by:

\[
C(V) = \frac{\text{d}Q(V)}{\text{d}V} = C(1 + c_1 \times V + c_2 \times V^2 + \ldots)
\]

where \( C \) is the capacitance of the instance, and \( c_k \) is the \( k \)-th entry in the \( \text{Coeffs} \) list. If \( C \) for the instance is not given, \( C \) for the model will be used.

The charge as a function of the applied voltage is:

\[
Q(V) = C \times V \times \left(1 + \left(\frac{1}{2}\right) \times c_1 \times V + \left(\frac{1}{3}\right) \times c_2 \times V^2 + \ldots\right)
\]
Lumped Components

**C_Conn (Capacitor (Connector Artwork))**

**Symbol**

²

**Available in** ADS

**Parameters**

C = capacitance, in farads

**Notes/Equations**

1. This component is a single connection in layout. For example, it can be used to represent parasitics.
C_dxxy (Capacitor (Delta X - Delta Y))

Symbol

Available in ADS

Parameters

C = capacitance, in farads
dx = delta X, in specified units
dy = delta Y, in specified units
Temp = temperature, in °C

Notes/Equations

1. This component shifts the next artwork in X/Y direction during layout in design synchronization from schematic to layout.
Lumped Components

C_Pad1 (Capacitor (Pad Artwork))

Symbol

Available in ADS

Parameters

C = capacitance, in farads
W = width of pad, in specified units
S = spacing, in specified units
L1 = (ADS Layout option) pin-to-pin distance, in specified units

Notes/Equations

1. This component’s artwork is composed of two rectangular pads with pins on the outer edges, as shown:
C_Space (Capacitor (Space Artwork))

Symbol

Available in ADS

Parameters

C = capacitance, in farads

L1 = (ADS Layout option) pin-to-pin distance, in specified units

Notes/Equations

1. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.
CAPP2_Conn (Chip Capacitor (Connector Artwork))

Symbol

Available in ADS

Parameters
C = capacitance, in farads
TanD = dielectric loss tangent
Q = quality factor
FreqQ = resistance frequency for Q, in hertz
FreqRes = resistance frequency, in hertz
Exp = exponent for frequency dependence of Q

Range of Usage
C, Q, FreqQ, FreqRes ≥ 0

Notes/Equations
1. The series resistance Rs is determined by the Q and the parallel resistance Rp is determined by TanD.
   The frequency-dependence of Q is given by
   \[ Q(f) = Q(F_{FQ}) \times (F_{FreqQ}/f)^{Exp} \]
   where f is the simulation frequency and Q(FreqQ) is the specified value of Q at FreqQ.
2. If Q or FreqQ are set to 0, Q is assumed to be infinite.
3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component has no default artwork associated with it.

References
Equivalent Circuit
Lumped Components

CAPP2_Pad1 (Chip Capacitor (Pad Artwork))
Symbol

Available in ADS
Parameters
C = capacitance, in farads
TanD = dielectric loss tangent
Q = quality factor
FreqQ = resistance frequency for Q, in hertz
FreqRes = resistance frequency, in hertz
Exp = exponent for frequency dependence of Q
W = (ADS Layout option) width of pad, in specified units
S = (ADS Layout option) spacing, in specified units
L1 = (ADS Layout option) pin-to-pin distance, in specified units

Range of Usage
C, Q, FreqQ, FreqRes ≥ 0

Notes/Equations
1. The series resistance Rs is determined by the Q and the parallel resistance Rp is determined by TanD.
   The frequency-dependence of Q is given by
   \[ Q(f) = Q(FreqQ) \times \left(\frac{FreqQ}{f}\right)^{Exp} \]
   where f is the simulation frequency and Q(FreqQ) is the specified value of Q at FreqQ.
2. If Q or FreqQ are set to 0, Q is assumed to be infinite.
3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component’s artwork is composed of two rectangular pads with pins on the outer edges.


Equivalent Circuit
Lumped Components

**CAPP2_Space (Chip Capacitor (Space Artwork))**

**Symbol**

```
   i
   1
   2
```

**Available in ADS**

**Parameters**

- C = capacitance, in farads
- TanD = dielectric loss tangent
- Q = quality factor
- FreQ = resistance frequency for Q, in hertz
- FreqRes = resistance frequency, in hertz
- Exp = exponent for frequency dependence of Q
- L1 = (ADS Layout option) pin-to-pin distance, in specified units

**Range of Usage**

\[ C, Q, FreqQ, FreqRes \geq 0 \]

**Notes/Equations**

1. The series resistance \( R_s \) is determined by the Q and the parallel resistance \( R_p \) is determined by TanD.
   
   The frequency-dependence of Q is given by
   
   \[ Q(f) = Q(FreqQ) \times (FreqQ/f)^{Exp} \]
   
   where \( f \) is the simulation frequency and \( Q(FreqQ) \) is the specified value of Q at \( FreqQ \).

2. If Q or FreqQ are set to 0, Q is assumed to be infinite.

3. For time-domain analysis, the frequency-domain analytical model is used.

4. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.

**References**


Equivalent Circuit
Lumped Components

CAPQ (Capacitor with Q)

Symbol

Available in ADS and RFDE

Parameters

C = capacitance, in farads
Q = quality factor
F = frequency at which Q is defined, in hertz
Mode = frequency dependence mode of Q; options (also refer to notes):

1 is proportional to freq
2 is proportional to sqrt(freq)
3 is constant

Range of Usage

F \geq 0

Notes/Equations

1. \[ Q = \frac{B}{G} = \frac{2\pi f C}{G} \]

where:

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>( Q(f) = Q(F) \times \frac{f}{F} )</td>
<td>( G(f) = G(F) )</td>
</tr>
<tr>
<td>proportional to sqrt(freq)</td>
<td>( Q(f) = Q(F) \times \sqrt{\frac{f}{F}} )</td>
<td>( G(f) = G(F) \times \sqrt{\frac{f}{F}} )</td>
</tr>
<tr>
<td>constant</td>
<td>( Q(f) = Q(F) )</td>
<td>( G(f) = G(F) \times \frac{f}{F} )</td>
</tr>
</tbody>
</table>

If F is set to zero, then Q is assumed to be infinite.

where

\( f = \) simulation frequency
\( F = \) reference frequency
\( G = \) conductance of capacitor
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component has no default artwork associated with it.

Equivalent Circuit
Lumped Components

CQ_Conn (Capacitor with Q (Connector Artwork))

Symbol

Available in ADS

Parameters

C = capacitance, in farads
Q = quality factor
F = frequency at which Q is defined, in hertz
Mode = frequency dependence mode of Q; options (also refer to notes):

1. proportional to freq
2. proportional to sqrt(freq)
3. constant

Range of Usage

F ≥ 0

Notes/Equations

1. \[ Q = \frac{B}{G} = \frac{2\pi f C}{G} \]

where:

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>( Q(f) = Q(F) \times f/F )</td>
<td>( G(f) = G(F) )</td>
</tr>
<tr>
<td>proportional to sqrt(freq)</td>
<td>( Q(f) = Q(F) \times \sqrt{f/F} )</td>
<td>( G(f) = G(F) \times \sqrt{f/F} )</td>
</tr>
<tr>
<td>constant</td>
<td>( Q(f) = Q(F) )</td>
<td>( G(f) = G(F) \times f/F )</td>
</tr>
</tbody>
</table>

If \( F \) is set to zero, then \( Q \) is assumed to be infinite.

where

f = simulation frequency
F = reference frequency
G = conductance of capacitor
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component is a single connection in layout. For example, it can be used to represent parasitics.

Equivalent Circuit

![Equivalent Circuit Diagram]
Lumped Components

CQ_Pad1 (Capacitor with Q (Pad Artwork))

Symbol

Available in  ADS

Parameters
C = capacitance, in farads
Q = quality factor
F = frequency at which Q is defined, in hertz
Mode = frequency dependence mode of Q; options (also refer to notes):
1 is proportional to freq
2 is proportional to sqrt(freq)
3 is constant
W = (ADS Layout option) width of pad, in specified units
S = (ADS Layout option) spacing, in specified units
L1 = (ADS Layout option) pin-to-pin distance, in specified units

Range of Usage
F ≥ 0

Notes/Equations

1. \( Q = \frac{B}{G} = \frac{2\pi f C}{G} \)

where:

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>( Q(f) = Q(F) \times \frac{f}{F} )</td>
<td>( G(f) = G(F) )</td>
</tr>
<tr>
<td>proportional to sqrt (freq)</td>
<td>( Q(f) = Q(F) \times \sqrt{\frac{f}{F}} )</td>
<td>( G(f) = G(F) \times \sqrt{\frac{f}{F}} )</td>
</tr>
</tbody>
</table>
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component’s artwork is composed of two rectangular pads with pins on the outer edges, as shown:

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q(f) = Q(F)</th>
<th>G(f) = G(F) × f/F</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If F is set to zero, then Q is assumed to be infinite.

where
- \( f \) = simulation frequency
- \( F \) = reference frequency
- \( G \) = conductance of capacitor

**Equivalent Circuit**

```
   1  G  2
    |
    |
    |
```

CQ_Pad1 (Capacitor with Q (Pad Artwork))  2-21
Lumped Components

**CQ_Space (Capacitor with Q (Space Artwork))**

**Symbol**

![Symbol Image]

**Available in ADS**

**Parameters**
- C = capacitance, in farads
- Q = quality factor
- F = frequency at which Q is defined, in hertz
- Mode = frequency dependence mode of Q; options (also refer to notes):
  - 1 is proportional to freq
  - 2 is proportional to sqrt(freq)
  - 3 is constant
- L1 = (ADS Layout option) pin-to-pin distance, in specified units

**Range of Usage**

\[ F \geq 0 \]

**Notes/Equations**

1. \[ Q = \frac{B}{G} = \frac{2\pi f C}{G} \]

where:

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>[ Q(f) = Q(F) \times \frac{f}{F} ]</td>
<td>[ G(f) = G(F) ]</td>
</tr>
<tr>
<td>proportional to sqrt(freq)</td>
<td>[ Q(f) = Q(F) \times \sqrt{\frac{f}{F}} ]</td>
<td>[ G(f) = G(F) \times \sqrt{\frac{f}{F}} ]</td>
</tr>
<tr>
<td>constant</td>
<td>[ Q(f) = Q(F) ]</td>
<td>[ G(f) = G(F) \times \frac{f}{F} ]</td>
</tr>
</tbody>
</table>

If \( F \) is set to zero, then \( Q \) is assumed to be infinite.

where

- \( f = \) simulation frequency
- \( F = \) reference frequency
- \( G = \) conductance of capacitor
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.

**Equivalent Circuit**

![Equivalent Circuit Diagram]
Lumped Components

**DC_Block (DC Block)**

Symbol

Available in  ADS and RFDE

**Parameters**

Mode = integer values denote functionality:
-1: blocks AC but feeds DC
0: feeds both AC and DC
1: blocks DC but feeds AC (default setting)
(refer to DC_Feed and Short components for other default settings)

C = dc block capacitance, in farads
L = dc feed inductance, in henries
Gain = current gain
wImax = maximum current (warning), in amperes (value type: real)

**Notes/Equations**

1. The C and L parameters are used for transient simulation only because open for DC_Block is non-causal for Transient simulation.
   The dc block in Transient is not an infinite C; it defaults to 1 \( \mu \)f.
   Reasonable C and L values (especially for Transient and Circuit Envelope simulation) are strongly recommended wherever possible.
DC_Feed (DC Feed)

Symbol

Available in ADS and RFDE

Parameters

Mode = integer values denote functionality:
-1: blocks AC but feeds DC (default setting)
0: feeds both AC and DC
1: blocks DC but feeds AC
(refer to DC_Block and Short components for other default settings)

C = dc block capacitance, in farads
L = dc feed inductance, in henries
Gain = current gain
wImax = maximum current (warning), in amperes (value type: real)

Notes/Equations

1. The C and L parameters are used for transient simulation only because short for DC_Feed is non-causal for transient simulation.
Lumped Components

DICAP (Dielectric Laboratories Di-cap Capacitor)

Symbol

Illustration

Available in ADS

Parameters

\( W = \) width of metal plates and dielectric, in specified units
\( L = \) length of metal plates and dielectric, in specified units
\( T = \) thickness of dielectric, in specified units
\( E_r = \) dielectric constant
\( \tan \delta L = \) dielectric loss tangent value at 1 MHz
\( R_O = \) series resistance at 1 GHz, in ohms

Notes/Equations

1. This is the Di-cap capacitor model by Dielectric Laboratories Incorporated; for the parameter values, please contact Dielectric Laboratories.

2. DICAP is a single-layer capacitor that behaves as lossy parallel plate transmission lines. Pin 1 is on the bottom metal plate; pin 2 is on the top metal plate. The connection (such as Wire or Ribbon) from the top metal plate (pin 2) to the connecting transmission line is not included in the model—the user must connect it separately.

3. For time-domain analysis, the frequency-domain analytical model is used.

4. In the layout, the top metal will be drawn on layer \textit{cond2}; the bottom metal on layer \textit{cond}; and, the capacitor dielectric on layer \textit{diel}.
DILABMLC (Dielectric Laboratories Multi-Layer Chip Capacitor)

Symbol

Illustration

Available in ADS

Parameters

CO = nominal capacitance, in farads

TanDel = dielectric loss tangent value at 1 MHz

RO = bulk resistivity of termination at 1 MHz, in ohms

Rt = termination loss resistance at 1 MHz

Re = electrode loss resistance per electrode at 1 GHz

Mount = mounting orientation: flat or edge

Notes/Equations

1. This is the multi-layer chip capacitor model by Dielectric Laboratories Incorporated; for the parameter values, please contact Dielectric Laboratories.

2. DILABMLC behaves as an open-ended transmission line. Pins 1 and 2 are at the edges of the capacitor’s solder leads. The connections (such as Wire or Ribbon) from the solder leads to the connecting transmission line are not included in the model—the user must connect them separately.

3. For transient analysis, the DILABMLC is modelled as a series RLC equivalent circuit.
Lumped Components

4. For convolution analysis, the frequency domain analytical model is used.

5. Attention should be given on the mounting orientation of the DILABMLC capacitor (whether it is flat or edge-mounted). The orientation of the capacitor relative to the gap in the microstrip affects the sequence of resonances.

   When the internal electrodes are parallel to the plane of the microstrip (flat mounted) parallel resonances occur when the equivalent line length is either an even or odd multiple of a half wavelength.

   When the internal electrodes are normal to the substrate (edge mounted), resonances occur only when the multiple is even. This suppression of odd-ordered resonances is the result of exciting the equivalent line at its center rather than at one end. Consequently, resonance occurs at higher frequencies when edge mounted.
INDQ (Inductor with Q)

**Symbol**

Available in ADS and RFDE

**Parameters**

L = inductance, in henries
Q = quality factor
F = frequency at which Q is given, in hertz
Mode = loss mode for this device:
1 is proportional to freq
2 is proportional to sqrt(freq)
3 is constant
Rdc = resistance for modes 2 and 3

**Range of Usage**

F \geq 0

**Notes/Equations**

1. Rdc is the value of R at dc; that is, 0 Hz. It can be thought of as the starting value for R. In Mode 1, the value of Rdc is ignored; in modes 2 and 3 it is used in the calculations.

2. The equivalent circuit for the INDQ component can be thought of as a complex impedance with the value

\[ Z = r_1 + j \times x_1 \]

where \( r_1 \) and \( x_1 \) are calculated using the following equations:

- freq = simulation frequency
- F = reference frequency
- Rdc = dc resistance (used in modes 2 and 3 only)

- For Mode 1:
  - \( \omega = 2 \times \pi \times \text{freq} \)
  - \( w_q = 2 \times \pi \times F \)
Lumped Components

\[ r_1 = w_q \times \frac{L}{Q} \]
\[ x_1 = \omega \times L \]

• For Mode 2:
  \[ \omega = 2 \times \pi \times \text{freq} \]
  \[ w_q = 2 \times \pi \times F \]
  \[ r_{t1} = w_q \times \frac{L}{Q} - R_{dc} \]
  \[ q_{t1} = w_q \times \frac{L}{r_{t1}} \]
  \[ r_{ac} = \sqrt{\omega \times w_q \times \frac{L}{q_{t1}}} \]
  \[ r_1 = R_{dc} + r_{ac} \]
  \[ x_1 = r_{ac} + \omega \times L \times (1-1/q_{t1}) \]

• For Mode 3:
  \[ \omega = 2 \times \pi \times \text{freq} \]
  \[ w_q = 2 \times \pi \times F \]
  \[ r_{q1} = w_q \times \frac{L}{Q} \]
  \[ r_{q2} = \sqrt{r_{q1} \times r_{q1} - R_{dc} \times R_{dc}} \]
  \[ q_{t} = w_q \times \frac{L}{r_{q2}} \]
  \[ r_{ac} = \omega \times L/q_{t} \]
  \[ r_1 = \sqrt{R_{dc} \times R_{dc} + r_{ac} \times r_{ac}} \]
  \[ x_1 = \omega \times L_{indq} \]

Thus using \( r_1 \) and \( x_1 \) from the above calculations, we can say the final equivalent circuit is a Resistor and Inductor in series, where the resistance value is set by \( r_1 \) and the inductance value would be \( x_1/(2 \times \pi \times \text{freq}) \)

3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component has no default artwork associated with it.

Equivalent Circuit

\[ \text{Equivalent Circuit} \]

\[ 1 \quad R \quad L \quad 2 \]
**InDQ2 (Inductor with Q)**

**Symbol**

![Inductor symbol]

Available in **ADS and RFDE**

**Parameters**

- **L** = inductance
- **Q** = quality factor
- **F** = frequency at which Q is given, in Hertz
- **Mode** = loss mode for this device
  - 1: Q being proportional to freq
  - 2: Q being proportional to sqrt (freq)
  - 3: Q being constant
  - 4: Q being proportional to sqrt (freq), with constant L
- **Rdc** = series constant resistance associated with the device, for Mode=2, 3, and 4 only

**Range of Usage**

\[ F \geq 0 \]

**Notes/Equations**

1. **Rdc** is the value of R at dc; that is, 0 Hz. It can be thought of as the starting value for R. In Mode 1, the value of Rdc is ignored; in modes 2, 3, and 4 it is used in the calculations.

2. The equivalent circuit for the INDQ2 component can be thought of as a complex impedance with the value

\[ Z = r_1 + j \times x_1 \]

where \( r_1 \) and \( x_1 \) are calculated using the following equations:

- For Mode 1:
  \[ \omega = 2 \times \pi \times \text{freq} \]
  \[ w_q = 2 \times \pi \times F \]
Lumped Components

\[ r_1 = w_q \times \frac{L}{Q} \]
\[ x_1 = \omega \times L \]

- For Mode 2:
  \[ \omega = 2 \times \pi \times \text{freq} \]
  \[ w_q = 2 \times \pi \times F \]
  \[ rt_1 = w_q \times \frac{L}{Q} - R_{dc} \]
  \[ qt_1 = w_q \times \frac{L}{rt_1} \]
  \[ rac = \sqrt{\omega \times w_q \times \frac{L}{qt_1}} \]
  \[ r_1 = R_{dc} + rac \]
  \[ x_1 = rac + \omega \times L \times (1 - \frac{1}{qt_1}) \]

- For Mode 3:
  \[ \omega = 2 \times \pi \times \text{freq} \]
  \[ w_q = 2 \times \pi \times F \]
  \[ rq_1 = w_q \times \frac{L}{Q} \]
  \[ rq_2 = \sqrt{rq_1 \times rq_1 - R_{dc} \times R_{dc}} \]
  \[ qt = w_q \times \frac{L}{rq_2} \]
  \[ rac = \omega \times \frac{L}{qt} \]
  \[ r_1 = \sqrt{R_{dc} \times R_{dc} + rac \times rac} \]
  \[ x_1 = \omega \times L_{indq} \]

- For Mode 4:
  \[ \omega = 2 \times \pi \times \text{freq} \]
  \[ w_q = 2 \times \pi \times F \]
  \[ rt_1 = w_q \times \frac{L}{Q} - R_{dc} \]
  \[ qt_1 = w_q \times \frac{L}{rt_1} \]
  \[ rac = \sqrt{\omega \times w_q \times \frac{L}{qt_1}} \]
  \[ r_1 = R_{dc} + rac \]
  \[ x_1 = \omega \times L \]

Thus using \( r_1 \) and \( x_1 \) from the above calculations, we can say the final equivalent circuit is a Resistor and Inductor in series, where the resistance value is set by \( r_1 \) and the inductance value would be \( x_1/(2 \times \pi \times \text{freq}) \).

3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component has no default artwork associated with it.
L (Inductor)

Symbol

Available in ADS and RFDE

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>inductance</td>
<td>fH, pH, nH, uH, mH</td>
<td>1.0 nH</td>
</tr>
<tr>
<td>R</td>
<td>series resistance</td>
<td>mOhm, Ohm, kOhm, MOhm, GOhm</td>
<td></td>
</tr>
<tr>
<td>Temp</td>
<td>nominal temperature</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>Trise</td>
<td>temperature rise over ambient</td>
<td>°C</td>
<td>0</td>
</tr>
<tr>
<td>Tnom</td>
<td>nominal temperature</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>TC1</td>
<td>linear temperature coefficient</td>
<td>1/°C</td>
<td></td>
</tr>
<tr>
<td>TC2</td>
<td>quadratic temperature coefficient</td>
<td>1/°C²</td>
<td></td>
</tr>
<tr>
<td>InitCond</td>
<td>transient analysis initial condition current</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>noise generation option: yes=1, no=0</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td>model instance name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_M</td>
<td>number of inductors in parallel</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Notes/Equations

1. The inductor value can be made a function of temperature by setting Tnom and either TC1 or TC2 or both. Tnom specifies the nominal temperature at which L is given. Tnom defaults to 25°C. If Temp≠Tnom, then the simulated inductance value is given by:

\[ L' = L \times [1 + TC1 (Temp - Tnom) + TC2 (Temp - Tnom)^2] \]

The resistance, if specified, is not temperature scaled.

2. If Temp is not explicitly specified, it defaults to the global temperature specified in the options item.

3. If the series resistance is specified, it always generates thermal noise:

\[ <i^2> = 4kT/R. \]
Lumped Components

4. If a model name is given, then values that are not specified on the inductor instance are taken from the model values. Typical values that can be defaulted are the inductance, series resistance, nominal temperature and temperature coefficients.

5. When InitCond is explicitly specified, the check-box Use user-specified initial conditions must be turned on in the Convergence tab of the Tran transient simulation controller for the parameter setting to take effect.

6. _M is used to represent the number of inductors in parallel and defaults to 1. M cannot be zero. If an inductor model is used, an optional scaling parameter Scale can also be defined on the model; it defaults to 1. The effective inductance that will be simulated is \( L \times \text{Scale}/M \); the effective resistance is \( R \times \text{Scale}/M \).

7. Table 2-1 lists the DC operating point parameters that can be sent to the dataset.

Table 2-1. DC Operating Point Information

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Current</td>
<td>A</td>
</tr>
</tbody>
</table>
L_Conn (Inductor (Connector Artwork))
Symbol

Available in ADS
Parameters
L = inductance, in henries
Notes/Equations
1. This component is a single connection in layout. For example, it can be used to represent parasitics.
Lumped Components

**L_Model (Inductor Model)**

**Symbol**

**Available in** ADS and RFDE

Supported via model include file in RFDE

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>default inductance</td>
<td>H</td>
<td>1.0 nH</td>
</tr>
<tr>
<td>R</td>
<td>series resistance</td>
<td>ohms</td>
<td></td>
</tr>
<tr>
<td>Tnom</td>
<td>nominal temperature</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>Trise</td>
<td>temperature rise over ambient</td>
<td>°C</td>
<td>0</td>
</tr>
<tr>
<td>TC1</td>
<td>linear temperature coefficient</td>
<td>1/°C</td>
<td></td>
</tr>
<tr>
<td>TC2</td>
<td>quadratic temperature coefficient</td>
<td>1/°C²</td>
<td></td>
</tr>
<tr>
<td>Scale (Scalei)</td>
<td>scaling factor</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Kf</td>
<td>flicker noise coefficient</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Af</td>
<td>flicker noise exponent</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Coeffs</td>
<td>nonlinear inductor polynomial coefficients</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All Params</td>
<td>Data Access Component (DAC) based parameters</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Netlist Format**

Model statements for the ADS circuit simulator may be stored in an external file. This is typically done with foundry model kits. For more information on how to set up and use foundry model kits, refer to the Design Kit Development manual.

```
model modelname L_Model [parm=value]*
```

The model statement starts with the required keyword model. It is followed by the model name that will be used by inductor components to refer to the model. The third parameter indicates the type of model; for this model it is L_Model. The rest of the model contains pairs of model parameters and values, separated by an equal sign. The name of the model parameter must appear exactly as shown in the parameters table—these names are case sensitive. Some model parameters have aliases, which are listed in parentheses after the main parameter name; these are parameter names that can be used instead of the primary parameter name. Model parameters may
appear in any order in the model statement. Model parameters that are not specified take the default value indicated in the parameters table. For more information about the ADS circuit simulator netlist format, refer to “ADS Simulator Input Syntax” in the Circuit Simulation manual.

Example:

```
model bondWire L_Model \
Tc1=20e-6
Coeffs=list(1,2)
```

Notes/Equations

1. This model supplies values for an inductor L. This allows some common inductor values to be specified in a model.

2. Kf and Af add flicker noise using the equation:

   \[ <i^2> = Kf \times I_d^2 / f \]

3. The inductor value can be made a nonlinear function of the inductor current I by specifying the polynomial coefficients list (Coeffs = list (c1, c2, c3, ...)). The inductance value L(I) is then given by:

   \[ L(I) = \frac{dF_{\text{flux}}(I)}{dI} = L(1 + c1 \times I + c2 \times I^2 + ...) \]

   where L is the inductance of the instance, and ck is the k-th entry in the Coeffs list. If L for the instance is not given, L for the model will be used.

   The branch flux as a function of the inductor current is:

   \[ F_{\text{flux}}(I) = L \times I \times \left( I + \left(1 \times \frac{1}{2}\right) \times c1 \times I + \left(1 \times \frac{1}{3}\right) \times c2 \times I^2 + ... \right) \]
Lumped Components

**L_Pad1 (Inductor (Pad Artwork))**

**Symbol**

```
1 ---- 2
```

**Available in ADS**

**Parameters**

- \( L \) = inductance, in henries
- \( W \) = (ADS Layout option) width of pad, in specified units
- \( S \) = (ADS Layout option) spacing, in specified unit
- \( L_1 \) = nominal temperature, in °C

**Notes/Equations**

1. This component’s artwork is composed of two rectangular pads with pins on the outer edges, as shown:

```
\[ \text{Diagram of two rectangular pads with pins on the outer edges.} \]
```
L_Space (Inductor (Space Artwork))

Symbol

Available in ADS

Parameters

L = inductance, in henries
L₁ = nominal temperature, in °C

Notes/Equations

1. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.
Lumped Components

LQ_Conn (Inductor with Q (Connector Artwork))

Symbol

Available in ADS

Parameters

L = inductance, in henries
Q = quality factor
F = reference frequency for Q
Mode = frequency dependence mode of Q; options (also refer to notes):
  1 is proportional to freq
  2 is proportional to sqrt(freq)
  3 is constant

Temp = temperature, in °C

Range of Usage

F ≥ 0

Notes/Equations

1. \[ Q = \frac{2\pi fL}{R} \]

   where:

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>( Q(f) = Q(F) \times \frac{f}{F} )</td>
<td>( G(f) = G(F) )</td>
</tr>
<tr>
<td>proportional to sqrt(freq)</td>
<td>( Q(f) = Q(F) \times \sqrt{\frac{f}{F}} )</td>
<td>( G(f) = G(F) \times \sqrt{\frac{f}{F}} )</td>
</tr>
<tr>
<td>constant</td>
<td>( Q(f) = Q(F) )</td>
<td>( G(f) = G(F) \times \frac{f}{F} )</td>
</tr>
</tbody>
</table>

If \( F \) is set to zero, then \( Q \) is assumed to be infinite.

where

\( f = \) simulation frequency
\( F = \) reference frequency
\( G = \) conductance of capacitor
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component is a single connection in layout. For example, it can be used to represent parasitics.

Equivalent Circuit

![Equivalent Circuit Diagram]
Lumped Components

LQ_Pad1 (Inductor with Q (Pad Artwork))

Symbol

Available in ADS

Parameters

L = inductance, in henries
Q = quality factor
F = reference frequency for Q
Mode = loss mode for this device; options (also refer to notes):
1 is proportional to freq
2 is proportional to sqrt(freq)
3 is constant
Temp = temperature, in °C
W = (ADS Layout option) width of pad, in specified units
S = (ADS Layout option) spacing, in specified units
L1 = (ADS Layout option) pin-to-pin distance, in specified units

Range of Usage

F ≥ 0

Notes/Equations

1. \( Q = \frac{2\pi fl}{R} \)

where:

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>( Q(f) = Q(F) \times f/F )</td>
<td>( G(f) = G(F) )</td>
</tr>
<tr>
<td>proportional to sqrt (freq)</td>
<td>( Q(f) = Q(F) \times \sqrt{f/F} )</td>
<td>( G(f) = G(F) \times \sqrt{f/F} )</td>
</tr>
</tbody>
</table>
2. For time-domain analysis, the frequency-domain analytical model is used.

3. This component’s artwork is composed of two rectangular pads with pins on the outer edges, as shown:

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>$Q$</th>
<th>$G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td>$Q(f) = Q(F)$</td>
<td>$G(f) = G(F) \times f / F$</td>
</tr>
</tbody>
</table>

If $F$ is set to zero, then $Q$ is assumed to be infinite.

where

- $f$ = simulation frequency
- $F$ = reference frequency
- $G$ = conductance of capacitor

Equivalent Circuit
Lumped Components

**LQ_Space (Inductor with Q (Space Artwork))**

**Symbol**

![Inductor Symbol]

**Available in** ADS

**Parameters**

- **L** = inductance, in henries
- **Q** = quality factor
- **F** = reference frequency for Q
- **Mode** = loss mode for this device; options (also refer to notes):
  - 1 is proportional to freq
  - 2 is proportional to sqrt(freq)
  - 3 is constant
- **Temp** = temperature, in °C
- **L1** = (ADS Layout option) pin-to-pin distance, in specified units

**Range of Usage**

\[ F \geq 0 \]

**Notes/Equations**

1. \( Q = \frac{2\pi fL}{R} \)

where:

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>( Q(f) = Q(F) \times \frac{f}{F} )</td>
<td>( G(f) = G(F) )</td>
</tr>
<tr>
<td>proportional to sqrt(freq)</td>
<td>( Q(f) = Q(F) \times \sqrt{\frac{f}{F}} )</td>
<td>( G(f) = G(F) \times \sqrt{\frac{f}{F}} )</td>
</tr>
</tbody>
</table>
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.

**Equivalent Circuit**

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td>Q(f) = Q(F)</td>
<td>G(f) = G(F) × f/F</td>
</tr>
</tbody>
</table>

If F is set to zero, then Q is assumed to be infinite.

where

f = simulation frequency
F = reference frequency
G = conductance of capacitor
Lumped Components

**Mutual (Mutual Inductor)**

**Symbol**

![Mutual Inductor Symbol](image)

**Illustration**

![Mutual Inductor Illustration](image)

**Available in** ADS and RFDE

**Parameters**

- $K =$ mutual inductor coupling coefficient
- $M =$ mutual inductance, in henries
- Inductor1 ID of inductor 1 (value type: string)
- Inductor2 ID of inductor 2 (value type: string)

**Range of Usage**

$-1.0 \leq K \leq 1.0$

**Notes/Equations**

1. Specify $K$ or $M$; if both are specified, $M$ overrides $K$.
2. For Inductor1 and Inductor2, enter the component names of any two inductors whose mutual inductance is given as $M$. For example, setting Inductor1 = L4 and Inductor2 = L16 result in simulations that use the value $M$ as mutual inductance between the inductors that appear on the schematic as L4 and L16. Use several mutual inductor components to define other mutual inductances; there is no limit to the number of mutual inductances that can be specified.

**Note** To edit string parameters on a schematic, highlight the parameter and enter a value enclosed with double quote symbols.

2-46  Mutual (Mutual Inductor)
3. The ends of the inductors that are in-phase are identified by a small open circle on the schematic symbol for the inductors.

4. Mutual inductor components can be placed anywhere on the schematic; they do not effect auto-layout.
Lumped Components

**PLC (Parallel Inductor-Capacitor)**

**Symbol**

![Symbol of PLC](image)

**Available in** ADS and RFDE

**Parameters**

$L =$ inductance, in henries

$C =$ capacitance in farads

**Notes/Equations**

1. Use for high Q circuits rather than individual components in parallel.
2. This component has no default artwork associated with it.

**Equivalent Circuit**

![Equivalent Circuit](image)
PLCQ (Parallel Inductor-Capacitor with Q)

Symbol

Available in ADS and RFDE

Parameters

L = capacitance, in farads
Ql = quality factor of inductor
Fl = frequency at which Q is defined, in hertz
Mode = frequency dependence mode of inductor Q; options (also refer to notes):
   1 is proportional to freq
   2 is proportional to sqrt(freq)
   3 is constant
C = capacitance, in farads
Qc = quality factor of capacitor
Fc = frequency at which capacitor Q is given, in hertz
ModC = frequency dependence mode of capacitor Q; options (also refer to notes):
   proportional to freq; proportional to sqrt(freq); constant; (value type: enumerated)
Rdc = resistance for modes 2 and 3

Notes/Equations

1. Use for high Q circuits, rather than individual components in parallel.

2. Ql = \( \frac{2\pi f_s L}{R} \) (for inductors)
   \( Q_c = \frac{2\pi f_s C}{G} \) (for capacitors)

where

<table>
<thead>
<tr>
<th>ModL Setting</th>
<th>Ql</th>
<th>ModC Setting</th>
<th>Qc</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>Ql(F) \times f_s/Fl</td>
<td>proportional to freq</td>
<td>Qc(Fc) \times f_s/Fc</td>
</tr>
<tr>
<td>proportional to sqrt (freq)</td>
<td>Q(f) = Q(F) \times \sqrt{f/F}</td>
<td>proportional to sqrt (freq)</td>
<td>G(f) = G(F) \times \sqrt{f/F}</td>
</tr>
</tbody>
</table>

PLCQ (Parallel Inductor-Capacitor with Q) 2-49
Lumped Components

<table>
<thead>
<tr>
<th>ModL Setting</th>
<th>Ql</th>
<th>ModC Setting Ql</th>
<th>Qc</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td>Ql(Fl)</td>
<td>constant</td>
<td>Qc(Fc)</td>
</tr>
</tbody>
</table>

where
- \( R \) = resistance of inductor
- \( G \) = conductance of capacitor
- \( f_s \) = simulation frequency
- \( F_C, F_I \) = specified Q frequencies

3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component has no default artwork associated with it.

**Equivalent Circuit**
PRC (Parallel Resistor-Capacitor)

Symbol

Available in ADS and RFDE

Parameters

R = resistance, in ohms
C = capacitance, in farads

Notes/Equations

1. This component has no default artwork associated with it.

Equivalent Circuit
Lumped Components

**PRL (Parallel Resistor-Inductor)**

**Symbol**

![Symbol Image]

**Available in** ADS and RFDE

**Parameters**

- \( R \) = resistance, in ohms
- \( C \) = capacitance, in henries

**Notes/Equations**

1. This component has no default artwork associated with it.

**Equivalent Circuit**

![Equivalent Circuit Image]
PRLC (Parallel Resistor-Inductor-Capacitor)

Symbol

Available in ADS and RFDE

Parameters
R = resistance, in ohms
L = inductance, in henries
C = capacitance, in farads

Notes/Equations
1. Use with high Q circuits, rather than individual components in parallel.
2. This component has no default artwork associated with it.

Equivalent Circuit
Lumped Components

**R (Resistor)**

**Symbol**

![Resistor Symbol]

**Available in** ADS and RFDE

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>resistance</td>
<td>mOhm, Ohm, kOhm, MOhm, GOhm</td>
<td>50 Ohm</td>
</tr>
<tr>
<td>Temp</td>
<td>temperature</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>Trise</td>
<td>temperature rise above ambient</td>
<td>°C</td>
<td>0</td>
</tr>
<tr>
<td>Tnom</td>
<td>nominal temperature</td>
<td>°C</td>
<td>25</td>
</tr>
<tr>
<td>TC1</td>
<td>linear temperature coefficient</td>
<td>1/°C</td>
<td></td>
</tr>
<tr>
<td>TC2</td>
<td>quadratic temperature coefficient</td>
<td>1/°C²</td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>resistor thermal noise option: yes=enable; no=disable</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>wPmax</td>
<td>maximum power dissipation (warning)</td>
<td>pW, nW, uW, mW, W, kW, dbm</td>
<td></td>
</tr>
<tr>
<td>wImax</td>
<td>maximum current (warning)</td>
<td>fA, pA, nA, uA, mA, A</td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td>name of a resistor model to use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>physical width for use with a model</td>
<td>um, mm, cm, meter, mil, in</td>
<td>model</td>
</tr>
<tr>
<td>Length</td>
<td>physical length for use with a model</td>
<td>um, mm, cm, meter, mil, in</td>
<td>model</td>
</tr>
<tr>
<td>_M</td>
<td>number of resistors in parallel</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. The resistor value can be made a function of temperature by setting Tnom and TC1 or TC2 or both. Tnom specifies the nominal temperature at which R is given. Tnom defaults to 25°C. If Temp≠Tnom, then the simulated resistance value is given by:

   \[ R' = R \times [1 + TC1 (\text{Temp} - \text{Tnom}) + TC2 (\text{Temp} - \text{Tnom})^2] \]

2. If Temp is not explicitly specified, it defaults to the global temperature specified in the options item.

3. The resistor generates thermal noise:
\[ \langle \sigma^2 \rangle = 4kT/R \]

Noise generation can be disabled by setting Noise=no.

4. \( wP_{\text{max}} \) and \( wI_{\text{max}} \) are used by the overload alert feature. They set limits on the maximum instantaneous power dissipated by the resistor and maximum current through the resistor. If these limits are specified, the simulator will issue a warning the first time they are exceeded during a dc, harmonic balance or transient simulation. Simulation results are not affected by this parameter.

5. For a transient simulation, the resistance can vary with time. The resistance value should be assigned an expression that is a function of the reserved variable time, which is the simulation time in seconds.

6. If a model name is given, then values that are not specified on the resistor instance are taken from the model values. Typical values that can be defaulted are resistance, length and width, nominal temperature, temperature coefficients, and overload alert parameters.

If a model is used, the resistance value to be simulated (before temperature scaling is applied) is calculated as:

\[ R' = R + R_{\text{sh}} \times \frac{(\text{Length} - 2 \times \text{Narrow} - 2 \times \text{Dw})}{(\text{Width} - 2 \times \text{Narrow} - 2 \times \text{Dl})} \]

7. \( _{\text{M}} \) is used to represent the number of resistors in parallel and defaults to 1. \( _{\text{M}} \) cannot be zero. If a resistor model is used, an optional scaling parameter Scale can also be defined on the model; it defaults to 1. The effective resistance that will be simulated is \( R \times \text{Scale/M} \).

8. Table 2-2 lists the DC operating point parameters that can be sent to the dataset.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Current</td>
<td>A</td>
</tr>
</tbody>
</table>
Lumped Components

**R_Model (Resistor Model)**

**Symbol**

Available in ADS and RFDE

Supported via model include file in RFDE

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>resistance</td>
<td>mOhm, Ohm, kOhm, MOhm, GOhm</td>
<td>50 Ohm</td>
</tr>
<tr>
<td>Rsh</td>
<td>sheet resistance</td>
<td>ohms/square</td>
<td></td>
</tr>
<tr>
<td>Length (L)</td>
<td>default length</td>
<td>um, mm, cm, meter, mil, in</td>
<td></td>
</tr>
<tr>
<td>Width (W)</td>
<td>default width</td>
<td>um, mm, cm, meter, mil, in</td>
<td></td>
</tr>
<tr>
<td>Narrow</td>
<td>length and width narrowing due to etching</td>
<td>um, mm, cm, meter, mil, in</td>
<td></td>
</tr>
<tr>
<td>Tnom</td>
<td>nominal temperature</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>Trise</td>
<td>temperature rise over ambient</td>
<td>°C</td>
<td>0</td>
</tr>
<tr>
<td>TC1</td>
<td>linear temperature coefficient</td>
<td>1/°C</td>
<td></td>
</tr>
<tr>
<td>TC2</td>
<td>quadratic temperature coefficient</td>
<td>1/°C²</td>
<td></td>
</tr>
<tr>
<td>wPmax</td>
<td>maximum power dissipation (warning)</td>
<td>pW, nW, uW, mW, W, kW, dbm</td>
<td></td>
</tr>
<tr>
<td>wImax</td>
<td>maximum current (warning)</td>
<td>amperes or amperes/meter</td>
<td></td>
</tr>
<tr>
<td>Scale (Scaler)</td>
<td>resistance scaling factor</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>AllParams</td>
<td>DataAccessComponent-based parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dw (Etch)</td>
<td>width narrowing due to etching</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>Di (Etch)</td>
<td>length narrowing due to etching</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>Kf</td>
<td>flicker noise coefficient</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Af</td>
<td>flicker noise exponent</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Wdexp</td>
<td>flicker noise W exponent</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Ldexp</td>
<td>flicker noise L exponent</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Weexp</td>
<td>flicker noise Weff exponent</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Leexp</td>
<td>flicker noise Leff exponent</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Unit</td>
<td>default</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>Fexp</td>
<td>flicker noise frequency exponent</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Coeffs</td>
<td>nonlinear resistor polynomial coefficients</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lumped Components

**Netlist Format**

Model statements for the ADS circuit simulator may be stored in an external file. This is typically done with foundry model kits. For more information on how to set up and use foundry model kits, refer to the Design Kit Development manual.

```
model modelname R_Model [parm=value]*
```

The model statement starts with the required keyword model. It is followed by the modelname that will be used by resistor components to refer to the model. The third parameter indicates the type of model; for this model it is `R_Model`. The rest of the model contains pairs of model parameters and values, separated by an equal sign. The name of the model parameter must appear exactly as shown in the parameters table; these names are case sensitive. Some model parameters have aliases, which are listed in parentheses after the main parameter name; these are parameter names that can be used instead of the primary parameter name. Model parameters may appear in any order in the model statement. Model parameters that are not specified take the default value indicated in the parameters table. For more information about the ADS circuit simulator netlist format, refer to “ADS Simulator Input Syntax” in the Circuit Simulation manual.

Example:

```
model  polyRes R_Model \n   Rsh=100 Etch=2.5e-8 TC1=50e-6
   Coeffs=list(1,2)
```

**Notes/Equations**

For RFDE Users  Information about this model must be provided in a model file; refer to the Netlist Format section.

1. `R_Model` supplies model parameters for use with a resistor R. This allows physically-based resistors to be modeled based on length and width.

2. When the physical parameters Rsh, Width and Length are specified, Wl max is the current limit in amperes/meter:

   \[
   \text{Wl max}^* = \text{Wl max} \times (\text{Width} - 2 \times \text{Narrow} - 2 \times \text{Dw})
   \]

   If the physical parameters Rsh, Width and Length are not specified, Wl max is the current limit in amperes.
3. Use AllParams with a DataAccessComponent to specify file-based parameters (refer to DataAccessComponent). Note that model parameters that are explicitly specified take precedence over those specified via AllParams.

4. Flicker noise is added using the following equation:

\[ \langle \delta^2 \rangle = \frac{K_f I_{DC} \sqrt{f}}{W_{W} d_{W} \sqrt{L_{E} d_{L} \sqrt{W_{W} e_{W} L_{W} e_{L} L_{W} e_{L}}} \right. \]

5. The resistor value can be made a nonlinear function of the applied voltage \( V \) by specifying the polynomial coefficients list (\( \text{Coeffs} = \text{list}(c_1, c_2, c_3, \ldots) \)). The resistance value \( R(V) \) is then given by:

\[ R(V) = \frac{dV}{dI} = \frac{R}{(1 + c_1 \times V + c_2 \times V^2 + \ldots)} \]

where \( R \) is the resistance of the instance, and \( c_k \) is the \( k \)-th entry in the \( \text{Coeffs} \) list.

The branch current as a function of the applied voltage is:

\[ I(V) = \left( \frac{V}{R} \right) \times \left( 1 + \left( \frac{1}{2} \right) \times c_1 \times V + \left( \frac{1}{3} \right) \times c_2 \times V^2 + \ldots \right) \]
Lumped Components

**R_Conn (Resistor (Connector Artwork))**

**Symbol**

```
\text{Symbol}
```

**Available in** ADS

**Parameters**

\[ R = \text{resistance, in ohms} \]

**Notes/Equations**

1. For time-domain analysis, the resistance can vary with time. The resistance value should be an equation whose value is calculated from the reserved variable \texttt{_time}.

2. This component is a single connection in layout. For example, it can be used to represent parasitics.
R_dxdy (Resistor (Delta X - Delta Y))

Symbol

Available in  ADS

Parameters

R = resistance, in ohms
dx = delta X, in specified units
dy = delta Y, in specified units
Temp = temperature, in °C

Notes/Equations

1. This component shifts the next artwork in X/Y direction during layout in design synchronization from schematic to layout.
Lumped Components

R_Pad1 (Resistor (Pad Artwork))

Symbol

Available in ADS

Parameters

R = resistance, in ohms

W = (ADS Layout option) width of pad, in specified units

S = (ADS Layout option) spacing, in specified units

L1 = (ADS Layout option) pin-to-pin distance, in specified units

Notes/Equations

1. For transient and convolution analyses, resistance can vary with time. The resistance value should be an equation whose value is calculated from the reserved variable _time.

2. This component’s artwork is composed of two rectangular pads with pins on the outer edges, as shown:

![Diagram of R_Pad1](image.png)
**R_Space (Resistor (Space Artwork))**

**Symbol**

![Resistor Symbol](image)

**Available in** ADS

**Parameters**

- \( R \) = resistance, in ohms
- \( L_1 \) = (ADS Layout option) pin-to-pin distance, in specified units

**Notes/Equations**

1. For time-domain analysis, the resistance can vary with time. The resistance value should be an equation whose value is calculated from the reserved variable \(_\text{time}\).
2. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.
Lumped Components

Short (Short)

Symbol

Available in ADS and RFDE

Parameters

Mode = integer values denote functionality:
-1: blocks AC but feeds DC
0: feeds both AC and DC (default setting)
1: blocks DC but feeds AC
(refer to DC_Block and DC_Feed components for other default settings)

C = dc block capacitance, in farads
L = dc feed inductance, in henries
Gain = current gain
SaveCurrent = save branch current (default: no)
wImax = maximum current warning, in amperes (value type: real)

Notes/Equations

1. This component behaves like a current probe. It can be used to measure the current anywhere in the circuit.
2. The variable name for the current is label.i, where label is the label of this component.
3. Table 2-3 lists the DC operating point parameters that can be sent to the dataset.

Table 2-3. DC Operating Point Information

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Current</td>
<td>A</td>
</tr>
</tbody>
</table>

2-64 Short (Short)
SLC (Series Inductor-Capacitor)

Symbol

Available in ADS and RFDE

Parameters
L = inductance, in henries
C = capacitance, in farads

Notes/Equations
1. Use when modeling high Q circuits as opposed to two individual components
2. This component has no default artwork associated with it.
Lumped Components

**SLCQ (Series Inductor-Capacitor with Q)**

**Symbol**

![Symbol of SLCQ](image)

**Available in** ADS and RFDE

**Parameters**

- $L =$ inductance, in henries
- $Q_l =$ quality factor of inductor
- $F_l =$ frequency at which $Q$ is defined, in hertz
- $\text{Mod}_L =$ frequency dependence mode of inductor $Q$; options (also refer to notes):
  1. is proportional to freq
  2. is proportional to $\sqrt{\text{freq}}$
  3. is constant
- $C =$ capacitance, in farads
- $Q_c =$ quality factor of capacitor
- $F_c =$ frequency at which capacitor $Q$ is given, in hertz
- $\text{Mod}_C =$ frequency dependence mode of capacitor $Q$; options (also refer to notes):
  1. is proportional to freq
  2. is proportional to $\sqrt{\text{freq}}$
  3. is constant
- $R_{dc} =$ resistance for modes 2 and 3

**Notes/Equations**

1. Use when modeling high $Q$ circuits rather than individual components in series.

2. $Q_l = \frac{2\pi f_s L}{R}$ (for inductors) \hspace{1cm} $Q_c = \frac{2\pi f_s C}{G}$ (for capacitors)
where

<table>
<thead>
<tr>
<th>ModL Setting</th>
<th>Q₁</th>
<th>ModC Setting</th>
<th>Qₖ</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>Q₁(F₁)×fₛ/F₁</td>
<td>proportional to freq</td>
<td>Qₖ(Fₖ)×fₛ/Fₖ</td>
</tr>
<tr>
<td>proportional to sqrt(freq)</td>
<td>Q₁(F₁)×√fₛ/F₁</td>
<td>proportional to sqrt(freq)</td>
<td>Qₖ(Fₖ)×√fₛ/Fₖ</td>
</tr>
<tr>
<td>constant</td>
<td>Q₁(F₁)</td>
<td>constant</td>
<td>Qₖ(Fₖ)</td>
</tr>
</tbody>
</table>

where

R = resistance of inductor
G = conductance of capacitor
fₛ = simulation frequency
Fₖ, F₁ = specified Q frequencies

3. For time-domain analysis, the frequency-domain analytical model is used.

4. This component has no default artwork associated with it.

Equivalent Circuit

![Equivalent Circuit Diagram]
Lumped Components

**SMT_Pad (SMT Bond Pad)**

*Symbol*

```
SMT_Pad
```

*Illustration*

![SMT Pad Illustration](image)

*Available in ADS*

*Parameters*

- **W** = width of pad, in specified units
- **L** = length of pad, in specified units
- **PadLayer** = layer of pad: default, cond, cond2, resi, diel, diel2, hole, bond, symbol, text, leads, packages; (value type: enumerated)
- **SMO** = solder mask overlap, in specified units
- **SM_Layer** = solder mask layer: solder_mask, hole, bond, symbol, text, leads, packages, ports, bound, silk_screen, silk_screen2, case_dimensions; (value type: enumerated)
- **PO** = pad offset from connection pin, in specified units

*Range of Usage*

- **W** ≥ 0
- **L** ≥ 0

*Notes/Equations*

1. This component is required for layout of SMT library parts.
2. For any library item to which this component is attached, the PO parameter specifies the offset of the bond pad center from the position of pin connections designated for that item's package artwork. An offset of 0 centers the pad.

---

2-68 SMT_Pad (SMT Bond Pad)
around the location of the pins. A positive value shifts the pad away from the package body; a negative value shifts the pad toward the package body.

3. A positive value for SMO increases the area of the solder mask; a negative value decreases it.
Lumped Components

**SRC (Series Resistor-Capacitor)**

**Symbol**

![Symbol](symbol.png)

**Available in** ADS and RFDE

**Parameters**

- \( R \) = inductance, in ohms
- \( C \) = capacitance, in farads

**Notes/Equations**

1. This component has no default artwork associated with it.
SRL (Series Resistor-Inductor)

Symbol

Available in ADS and RFDE

Parameters

R = resistance, in ohms
C = inductance, in henries

Notes/Equations

1. This component has no default artwork associated with it.
Lumped Components

SRLC (Series Resistor-Inductor-Capacitor)

Symbol

\[ \text{Symbol} \]

Available in ADS and RFDE

Parameters

- \( R \) = resistance, in ohms
- \( L \) = inductance, in henries
- \( C \) = capacitance, in farads

Notes/Equations

1. Use for high Q circuits, rather than individual components in parallel.
2. This component has no default artwork associated with it.
TF (Transformer)

Symbol

Available in ADS and RFDE

Parameters

T = turns ratio T1/T2

Notes/Equations

1. The turns ratio T is the ratio of turns in the primary to turns in the secondary (T:1).

   A turns ratio less than 1 describes a transformer in which there are more turns in the secondary than in the primary.

2. The TF component is a Hybrid component with the parameters H12=T and H21=-T. For more information on Hybrid components, refer to "Hybrid (2-Port User-Defined Linear Hybrid)" on page 6-4.

3. Parasitic inductances of the primary and secondary are not modeled. To do this, use the mutual inductance component “Mutual” located on the Lumped-Components palette. The ends that are in phase are identified by a small open circle on the schematic symbol.

4. Because this is an ideal transformer, the impedance transformation is the same at DC as it is at nonzero frequencies.

5. This component passes DC.
Lumped Components

**TF3 (3-Port Transformer)**

**Symbol**

Available in ADS and RFDE

**Parameters**

T1 = turn 1
T2 = turn 2

**Notes/Equations**

1. The turns ratio $T$ is the ratio of turns in the secondary to turns in the primary:

   $$ T = \frac{T_{\text{primary}}}{T_{\text{secondary}}} $$

2. A turns ratio less than 1 describes a transformer in which there are more turns in the secondary than in the primary. Parasitic inductances of the primary and secondary are not modeled; to do this, use the component for mutual inductance (M).

3. The ends that are in phase are identified by a small open circle on the schematic symbol.

4. DC voltages are also converted.
Chapter 3: Miscellaneous Circuit Components
Miscellaneous Circuit Components

**CAPP2 (Chip Capacitor)**

**Symbol**

![Symbol](image)

**Available in** ADS and RFDE

**Parameters**
- $C =$ capacitance, in farads
- $\text{TanD} =$ dielectric loss tangent
- $Q =$ quality factor
- $\text{Freq} =$ reference frequency for $Q$, in hertz
- $\text{FreqRes} =$ resonance frequency, in hertz
- $\text{Exp} =$ exponent for frequency dependence of $Q$

**Range of Usage**

$C$, $Q$, $\text{FreqQ}$, $\text{FreqRes} \geq 0$

**Notes/Equations**

1. The series resistance $R_s$ is determined by the $Q$ and the parallel resistance $R_p$ is determined by $\text{TanD}$.

   The frequency-dependence of $Q$ is given by

   $$Q(f) = Q(\text{FreqQ}) \times (\text{FreqQ}/f)^\text{Exp}$$

   where $f$ is the simulation frequency and $Q(\text{FreqQ})$ is the specified value of $Q$ at $\text{FreqQ}$.

2. If $Q$ or $\text{FreqQ}$ are set to 0, $Q$ is assumed to be infinite.

3. For time-domain analysis, the frequency-domain analytical model is used.

4. This component has no default artwork associated with it.

**References**


Equivalent Circuit
CIND (Ideal Toroidal Inductor)

Symbol

Illustration

Available in ADS and RFDE

Parameters

N = number of units
AL = inductance index

Range of Usage

N, AL > 0

Notes/Equations

1. The inductance is given by
   \[ L = N^2 \times AL \]

2. This component has no default artwork associated with it.
RIND (Rectangular Inductor)

Symbol

![Illustration of RIND symbol]

Available in ADS and RFDE

Parameters

- \( N \) = number of turns (need not be an integer)
- \( L_1 \) = length of second outermost segment, in specified units
- \( L_2 \) = length of outermost segment, in specified units
- \( W \) = conductor width, in specified units
- \( S \) = conductor spacing, in specified units
- \( T \) = conductor thickness, in specified units
- \( \rho \) = conductor resistivity (relative to copper)
- \( F_R \) = resonant frequency, in hertz
- \( \text{Temp} \) = physical temperature, in °C

Range of Usage

\( N \) must be such that all segments fit given \( L_1, L_2, W, \) and \( S \).

Notes/Equations
Miscellaneous Circuit Components

1. For time-domain analysis, an impulse response obtained from the frequency-domain analytical model is used.
2. This component has no default artwork associated with it.

References

XFERP (Physical Transformer)

Symbol

Available in ADS and RFDE

Parameters

- \( N \) = turns ratio \( N_1/N_2 \)
- \( L_p \) = magnetizing inductance, in henries
- \( R_c \) = core loss resistance, in ohms
- \( K \) = coefficient of coupling
- \( R_1 \) = primary loss resistance, in ohms
- \( R_2 \) = secondary loss resistance, in ohms
- \( C_1 \) = primary capacitance, in farads
- \( C_2 \) = secondary capacitance, in farads
- \( C \) = interwinding capacitance, in farads

Range of Usage

\( 0 < K < 1 \)

Notes/Equations

1. Primary leakage:

\[
L_1 = L_p \left( \frac{1}{K} - 1 \right)
\]

Secondary leakage:

\[
L_2 = \frac{L_1}{N^2}
\]

2. This component has no default artwork associated with it.
Equivalent Circuit
**XFERRUTH (Ruthroff Transformer)**

**Symbol**

![Symbol Image](image)

**Available in** ADS and RFDE

**Parameters**

- \( N \) = number of turns
- \( AL \) = inductance index, in henries
- \( Z \) = characteristic impedance of transmission line, in ohms
- \( E \) = electrical length of transmission line, in degrees
- \( F \) = reference frequency for electrical length, in hertz

**Range of Usage**

- \( N > 0 \)
- \( AL > 0 \)

**Notes/Equations**

1. Inductance: \( L = N^2 \times AL \)
2. This component has no default artwork associated with it.

**References**


Equivalent Circuit

3-10 XFERRUTH (Ruthroff Transformer)
XFERTAP (Tapped Secondary Ideal Transformer)

Symbol

Available in ADS and RFDE

Parameters

N12 = turns ratio, N1/N2
N13 = turns ratio, N1/N3
L1 = primary winding inductance, in henries
K = coupling coefficient

Range of Usage

0 < K < 1

Notes/Equations

1. This component has no default artwork associated with it.
Miscellaneous Circuit Components
Chapter 4: Probe Components
Counter Components

**Counter (Counter Component)**

**Symbol**

```
\[ \begin{array}{c} \text{Trig} \\
\text{Count} \\
\end{array} \]
```

**Available in** ADS

**Parameters**

Direction = direction one

Thresh = threshold one, in volts

**Notes/Equations**

1. This time counter model generates an output voltage equal to the number of times that the user-specified trigger has occurred. The trigger point is defined by setting a threshold voltage and a slope. The slope can be specified as either rising or falling by setting the direction parameter to a 1 or −1. A direction parameter value of 0 is used if a trigger for either slope is desired.

2. Only the baseband component of the input voltages is used to generate the trigger, so the model may be used in either envelope or transient time domain analysis modes. Linear interpolation is used to estimate the actual trigger crossing time to a finer resolution than the simulation time step.

3. The input impedance is infinite. The output impedance is 1 ohm. The open circuit output voltage is equal to \( n \), the number of triggers that have occurred up to the present simulation time. This count does not change until a trigger event occurs, and is held constant until another event occurs.
I_Probe (Current Probe)

Symbol

Available in ADS

Parameters

Mode = type of mode: short, dc block, dc feed (value type: integer)
C = dc block capacitance (transient only)
L = dc feed inductance (transient only)
Gain = current gain
SaveCurrent = save branch current (default: yes)
wImax = maximum current warning (value type: real)

Notes/Equations

1. The positive current flow direction is assumed to be from pin 1 to pin 2.
2. To measure a branch current, an ammeter must be connected in that branch before performing the analysis.
3. Table 4-1 lists the DC operating point parameters that can be sent to the dataset.

Table 4-1. DC Operating Point Information

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Current</td>
<td>A</td>
</tr>
</tbody>
</table>

4. The current sampled by I_Probe will have the following name in the dataset: <instance name>.i; for example: I_Probe1.i.
5. This component has no default artwork associated with it.
Probe Components

**OscPort (Grounded Oscillator Port)**

**Symbol**

![Symbol of OscPort](image)

**Available in** ADS and RFDE

**Parameters**

- **V** = initial guess at fundamental voltage
- **Z** = initial value for Z₀, in ohms (default: 1.1 ohms)
- **NumOctaves** = number of octaves to search (default: 2)
- **Steps** = number of steps per search octave (default: 10)
- **FundIndex** = fundamental number for oscillator (default: 1)
- **Harm** = harmonic or fundamental for oscillator (default: 1)
- **MaxLoopGainStep** = maximum arc length continuation step size during loop-gain search

**Notes/Equations**

1. This is a special device used for an oscillator analysis. Do not use more than one oscillator port in a circuit.

2. **NumOctaves** specifies the total number of octaves over which the oscillator search is done. Half of the octaves are below the initial frequency and half are above. For example, if **NumOctaves** is 2, then the frequency search goes from \( F_{\text{freq}}/2 \) to \( F_{\text{freq}} \times 2 \). **Steps** sets the number of frequency points per octave that are used in the search. For a high-Q oscillator, a large number of steps might be required.

3. If fundamental voltage **V** is not specified, the simulator first performs a small-signal AC analysis to determine the actual frequency and oscillation voltage.

   If **V** is specified, it represents an initial guess at the fundamental oscillator voltage at the point where the OscPort is inserted. The initial guess for **V** should be as close to the actual value as possible. An inaccurate value increases the simulation time and might prevent convergence. If it is not known, don't specify it.
4. Provided the circuit produces at least one complex conjugate pole pair in the right-half-plane over the frequency range tested, the analysis will determine the oscillation waveform and amplitude. Proper probe placement and impedance can reduce the analysis time significantly and help ensure accurate oscillator analysis results. To reduce the probability of a failed analysis, place the probe and set the initial impedance in a manner consistent with the following guidelines:

**Feedback Oscillators (such as Colpitts)**

- Insert probe at a point in the feedback loop where the signal is contained to a single path.
- Point the arrow of the probe in the direction of positive gain around the loop.
- Insert probe at a point in the feedback loop where source impedance is much smaller than load impedance (at least a factor of 10; a factor of 100 or more is preferable).
- Point the arrow of the probe at the high impedance (load) node.
- Set the initial probe impedance \( Z_0 \) to a value approximately half-way between the source and load impedances presented by the circuit at the point of insertion.

To minimize the analysis time, set the probe impedance to a factor of 10 below the load impedance, and a factor of 10 above the source impedance (provided the source and load impedances are sufficiently far apart). Doing this effectively reduces to zero the dependence of the small signal loop gain on \( Z_0 \).

**Negative Resistance Oscillators**

- Insert probe between a negative and positive impedance in the circuit. There should be no other signal paths between these two parts of the circuit. Typically, the probe is inserted between the resonator and the effective negative resistance.
- You can point the arrow of the probe at either the negative impedance node or the positive impedance node.
- Set the initial probe impedance to any reasonable value. To minimize the analysis time, it should be at least a factor of two higher or lower than the magnitude of the passive load impedance.
The frequency is specified on the harmonic balance analysis component. The value for Z is chosen based on impedance levels in the circuit and the degree of non-linearity in the circuit. Do not use either 1 or 0 for Z as this will cause convergence problems.

If the oscillator analysis fails, and this test indicates that the circuit should oscillate, the failure may be due to the fact that the circuit is too nonlinear. This problem can sometimes be solved by trying different impedance values of OscPort (determined by the Z attribute). Lower impedance values usually seem to work better. Also try reversing the OscPort direction.

Another approach is to try to get the oscillator to oscillate at some nicer parameter value and then to sweep the parameter value to the desired value. The parameter may be bias, self-bias impedance, some gain controlling value, or another factor. In short, anything that will make the oscillator more linear, yet still let it oscillate.

FundIndex is used for selecting which fundamental tone is considered the unknown frequency during oscillator analysis.

The FundIndex default is 1, which means that Freq[1] on the Harmonic Balance controller is the unknown frequency. This should be changed only if a larger multi-tone system is simulated, such as an oscillator and mixer. In this case, the user may want Freq[1] to be a known driven source and Freq[2] to be the unknown frequency used by the oscillator; for this, set FundIndex=2. For the best harmonic balance solution, the frequency that causes the most nonlinearity should be Freq[1].

Harm is used to make a circuit oscillator on a harmonic of the fundamental frequency rather than directly on the fundamental. For example, the circuit may consist of a 2GHz oscillator followed by a divide-by-two circuit. In this case, the harmonic balance analysis would be set up with Freq[1]=1 GHz, and OscPort2 would have Harm=2. (Note that successful simulation of an oscillator and divider will most likely require that transient-assisted harmonic balance be used.)
OscPort2 (Differential Oscillator Port)

Symbol

Available in ADS and RFDE

Parameters

Mode = oscillator mode: automatic (default), small signal loop gain, or large signal loop gain

V = initial guess at fundamental voltage (automatic mode only)

Z = initial value for Z₀, in ohms (all modes) (default: 1.1 ohms)

NumOctaves = number of octaves to search (automatic mode only) (default: 2)

Steps = number of steps per search octave (automatic mode only) (default: 10)

FundIndex = fundamental number for oscillator (automatic mode only) (default: 1)

Harm = harmonic or fundamental for oscillator (default: 1)

MaxLoopGainStep = maximum arc length continuation step size during loop-gain search (automatic mode only)

FreqPlan = sweep plan for frequency (small and large signal loop gain modes only)

VinjPlan = sweep plan for injected loop voltage (large signal loop gain mode only)

Notes/Equations

1. This is a special device used for an oscillator analysis. Do not use more than one oscillator test element (OscTest, OscPort, OscPort2) in a circuit.

2. NumOctaves specifies the total number of octaves over which the oscillator search is done. Half of the octaves are below the initial frequency and half are above. For example, if NumOctaves is 2, then the frequency search goes from Freq/2 to Freq × 2. Steps sets the number of frequency points per octave that are used in the search. For a high-Q oscillator, a large number of steps might be required.

3. If a fundamental voltage V is not specified, the simulator first performs a small-signal AC analysis to determine the actual frequency and oscillation voltage.
If $V$ is specified, it represents an initial guess at the fundamental oscillator voltage at the point where the OscPort is inserted. The initial guess for $V$ should be as close to the actual value as possible. An inaccurate value increases the simulation time and might prevent convergence. If it is not known, don't specify it.

4. This device can operate in one of three different modes. In automatic mode, it is similar to the OscPort device, and is used with the harmonic balance oscillator analysis to determine the oscillator frequency, large signal solution and optionally phase noise. In small signal loop gain mode, it is similar to the OscTest device, and is used to perform a small signal analysis of the oscillator loop gain versus frequency. In large signal loop gain mode, it is used to simulate the large signal nonlinear loop gain of the oscillator versus frequency and injected loop voltage.

5. This device can be used for both single-ended and differential oscillator topologies. For single-ended oscillators, the negative pins of this element should be grounded. For differential oscillators, it should be connected differentially into the oscillator loop.

6. Provided the circuit produces at least one complex conjugate pole pair in the right-half-plane over the frequency range tested, the analysis will determine the oscillation waveform and amplitude. Proper probe placement and impedance can reduce the analysis time significantly and help ensure accurate oscillator analysis results. To reduce the probability of a failed analysis, place the probe and set the initial impedance in a manner consistent with the following guidelines:

**Feedback Oscillators (such as Colpitts)**

- Insert probe at a point in the feedback loop where the signal is contained to a single path.
- Point the arrow of the probe in the direction of positive gain around the loop.
- Insert probe at a point in the feedback loop where source impedance is much smaller than load impedance (at least a factor of 10; a factor of 100 or more is preferable).
- Point the arrow of the probe at the high impedance (load) node.
- Set the initial probe impedance ($Z_0$) to a value approximately half-way between the source and load impedances presented by the circuit at the point of insertion.
To minimize the analysis time, set the probe impedance to a factor of 10 below the load impedance, and a factor of 10 above the source impedance (provided the source and load impedances are sufficiently far apart). Doing this effectively reduces to zero the dependence of the small signal loop gain on $Z_0$.

**Negative Resistance Oscillators**

- Insert probe between a negative and positive impedance in the circuit. There should be no other signal paths between these two parts of the circuit. Typically, the probe is inserted between the resonator and the effective negative resistance.
- You can point the arrow of the probe at either the negative impedance node or the positive impedance node.
- Set the initial probe impedance to any reasonable value. To minimize the analysis time, it should be at least a factor of two higher or lower than the magnitude of the passive load impedance.

The frequency is specified on the harmonic balance analysis component. The value for $Z$ is chosen based on impedance levels in the circuit and the degree of non-linearity in the circuit. Do not use either 1 or 0 for $Z$ as this will cause convergence problems.

If the oscillator analysis fails, and this test indicates that the circuit should oscillate, the failure may be due to the fact that the circuit is too nonlinear. This problem can sometimes be solved by trying different impedance values of OscPort (determined by the Z attribute). Lower impedance values usually seem to work better. Also try reversing the OscPort direction.

Another approach is to try to get the oscillator to oscillate at some nicer parameter value and then to sweep the parameter value to the desired value. The parameter may be bias, self-bias impedance, some gain controlling value, or another factor. In short, anything that will make the oscillator more linear, yet still let it oscillate.

FundIndex is used for selecting which fundamental tone is considered the unknown frequency during oscillator analysis.

The FundIndex default is 1, which means that Freq[1] on the Harmonic Balance controller is the unknown frequency. This should be changed only if a larger multi-tone system is simulated, such as an oscillator and mixer. In this case, the user may want Freq[1] to be a known driven source and Freq[2] to be the unknown frequency used by the oscillator; for this, set FundIndex=2. For
the best harmonic balance solution, the frequency that causes the most nonlinearity should be Freq[1].

Harm is used to make a circuit oscillator on a harmonic of the fundamental frequency rather than directly on the fundamental. For example, the circuit may consist of a 2GHz oscillator followed by a divide-by-two circuit. In this case, the harmonic balance analysis would be set up with Freq[1]=1 GHz, and OscPort2 would have Harm=2. (Note that successful simulation of an oscillator and divider will most likely require that transient-assisted harmonic balance be used.)

The equivalent circuit for OscPort2 in automatic mode is shown.

7. The small-signal loop gain mode is used to examine the small signal linear behavior of the oscillator feedback loop. In this mode, the OscPort2 element behaves as an analysis controller. Any simulation controllers should be disabled before using the OscPort2 in this mode. The analysis calculates and places in the dataset a complex value called LoopGain which is the small signal loopgain of the oscillator.

The range of frequencies over which to analyze loop gain should be specified with a SweepPlan item. The name of this SweepPlan should then be assigned to the parameter FreqPlan.

This device is used to evaluate the ability of a closed-loop system to produce one or more complex conjugate pole pairs in the right-half-plane (RHP) of a pole/zero diagram. This device measures the open-loop gain and phase of the closed-loop system. These results must be plotted on a polar graph (Nyquist diagram) to properly interpret them.
The number of clockwise encirclements of the \( l + j0 \) point indicates the number of RHP poles that were produced due to the feedback. The total number of RHP poles is the sum of the number of clockwise encirclements plus the number of RHP poles present in the individual networks that comprise the closed-loop system.

An important aspect of this last point is that traditional feedback or negative resistance topology systems may be unstable even though the \( l + j0 \) point is not encircled in a Nyquist diagram. For example, in a negative resistance topology circuit, if the reference impedance of the OscTest device is set equal to the passive load impedance, the measured loop gain is zero. The circuit will oscillate, however, because the negative resistance one-port generates an RHP pole prior to being configured with the remaining part of the system.

The equivalent circuit for OscPort2 in small signal loop gain mode is shown.

8. The large-signal loop gain mode is used to examine the behavior of the oscillator feedback loop as a function of frequency and injected voltage. It can be used to observe the compression of loop gain as the loop voltage is increased. In this mode, the OscPort2 element behaves as an analysis controller. Any simulation controllers should be disabled before using the OscPort2 in this mode. The analysis calculates and places in the dataset a complex value called LoopGain.
which is the large signal loop gain of the oscillator. The circuit will sustain oscillation at the point at which the magnitude of LoopGain equals one and the phase of LoopGain equals zero.

The range of frequencies over which to analyze loop gain should be specified with a SweepPlan item. The name of this SweepPlan should then be assigned to the parameter FreqPlan.

The range of voltages over which to analyze loop gain should be specified with a SweepPlan item. The name of this SweepPlan should then be assigned to the parameter VinjPlan. Initially the sweep should be done with a logarithmic sweep to determine where the oscillator loop goes into compression. Once this range is estimated, a linear sweep can be done to zero in on north injected voltage that causes oscillation to be sustained.

A useful way to interpret results from this analysis is to plot the phase of LoopGain against LoopGain in decibels. Lines of constant frequency will be plotted with values at each voltage value. The circuit will oscillate at the frequency and voltage associated with the (0,0) point on the graph.

The equivalent circuit for OscPort2 in large signal loop gain mode is shown.
In both small signal and large signal loop gain modes, this element injects a test signal into an oscillator circuit for stimulating oscillations. The specialized directional coupler has zero electrical length and is invisible to normal circuit simulation. It injects a fundamental frequency test signal, blocks the fundamental frequency flow in the feedback path, monitors the signal returned by the feedback path and calculates the loop gain.

The directional coupler in loop gain mode (both small and large signal) is designed to allow the injection of a test signal from port 3 to port 2 as the loop input and to pass the loop output from port 1 to port 4. It does this only at the signal frequency: the AC frequency for small signal loop gain and the fundamental tone for large signal loop gain. All other frequencies, including DC, are coupled from port 1 to port 2.

The scattering matrices follow.

\[
S_{\text{FUNDAMENTAL}} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 
\end{bmatrix}
\]

\[
S_{\text{OTHER}} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 
\end{bmatrix}
\]
OscTest (Grounded Oscillator Test)

Symbol

Available in ADS and RFDE

Parameters

Port_Number = port number
Z = port impedance, in ohms
Start = start frequency, in hertz
Stop = stop frequency, in hertz
Points = number of frequency points

Notes/Equations

1. This component performs an S-parameter analysis to evaluate the closed loop, small signal gain of a potential oscillator. It contains an analysis controller and sweeps the frequency from Start to Stop. S(1,1) is the loop gain.

2. This device is used to evaluate the ability of a closed-loop system to produce one or more complex conjugate pole pairs in the right-half-plane (RHP) of a pole/zero diagram. This device measures the open-loop gain and phase of the closed-loop system. These results must be plotted on a polar graph (Nyquist diagram) to properly interpret them.

The number of clockwise encirclements of the l+j0 point indicates the number of RHP poles that were produced due to the feedback. The total number of RHP poles is the sum of the number of clockwise encirclements plus the number of RHP poles present in the individual networks that comprise the closed-loop system.

An important aspect of this last point is that traditional feedback or negative resistance topology systems may be unstable even though the l+j0 point is not encircled in a Nyquist diagram. For example, in a negative resistance topology circuit, if the reference impedance of the OscTest device is set equal to the passive load impedance, the measured loop gain is zero. The circuit will oscillate, however, because the negative resistance one-port generates an RHP pole prior to being configured with the remaining part of the system.
3. Another way of looking for potential oscillations is to look for the point(s) where the magnitude of the loop gain is greater than 1, phase is 0, and the phase is decreasing with increasing frequency.

4. If Port is set to 1 and $S_{11}$ never goes outside the unity circle, the circuit will not oscillate and the frequency search will fail. The circuit must be redesigned, or it will be entered incorrectly. For the circuit to oscillate, the simulated loop gain must be greater than unity (1) when the phase is 0. If Port is set to 2, $S_{22}$ is the $S$-parameter to test; if Port is set to 3, $S_{33}$ is the $S$-parameter to test, and so on.

If the oscillator analysis fails, and this test indicates that the circuit should oscillate, the failure may be because the circuit is too nonlinear. This problem can be solved by trying different impedance values for OscTest (determined by the Z attribute). Lower impedance values usually work better, presumably because most nonlinearities are voltage controlled instead of current controlled. Reversing the OscTest direction should also be tried. The component could be inserted in the wrong direction; or, as occurs with some reflection oscillator cases, the solution may converge with the oscillator inserted in one direction and not the other.

Another alternative is to try to get the oscillator to oscillate at some nicer parameter value and then sweep the parameter value to the desired value. The parameter may be bias, self-bias impedance, some gain controlling value, or another factor; in short, anything that will make the oscillator more linear, yet still oscillate.
5. The equivalent circuit of OscTest is shown.
SProbe (SProbe Component)

Symbol

Available in ADS

Parameters
- Z0 = port impedance
- FLowStart = start frequency (low)
- FLowStop = stop frequency (low)
- FLowStep = step frequency (low)
- FHighStart = start frequency (high)
- FHighStop = stop frequency (high)
- FHighStep = step frequency (high)

Notes/Equations
1. The SProbe component is used to determine small-signal impedances or reflection coefficients looking both directions (Z1 and Z2). These impedances may be used to determine whether the conditions for oscillation are satisfied at any simulated frequency.
2. This component can be inserted anywhere into a circuit without loading it.
3. There are schematic and data display templates that use this component.
4. Do not combine the SProbe with other simulations that require Term components, including sources that have built-in Terms (e.g., P_1Tone).
5. The equivalent circuit of SProbe is shown.

This SProbe determines small signal impedances to reflection coefficients loss both at 50Ω and 40Ω. These impedances may be used to set the effect of coupling, the modulation and variation at any simulated frequency. There are schematic and data slightly complicated with one of the components.
SProbePair (SProbePair Component)

Symbol

Available in ADS

Parameters

- \( Z_0 \) = port impedance
- \( F_{\text{LowStart}} \) = start frequency (low)
- \( F_{\text{LowStop}} \) = stop frequency (low)
- \( F_{\text{LowStep}} \) = step frequency (low)
- \( F_{\text{HighStart}} \) = start frequency (high)
- \( F_{\text{HighStop}} \) = stop frequency (high)
- \( F_{\text{HighStep}} \) = step frequency (high)

Notes/Equations

1. The SProbePair component is used to determine small-signal impedances or reflection coefficients looking both directions at the input and output planes of a device or circuit. These impedances may be used to determine whether the conditions for oscillation are satisfied at any simulated frequency.

2. This component can be inserted anywhere into a circuit without loading it.

3. Do not combine the SProbePair with other simulations that require Term components, including sources that have built-in Terms (e.g., P_1Tone).

4. SProbePair is used in the following example:

\$HPEESOF_DIR/examples/MW_Ckts/MMIC_Amp_prj/TwoStgAmpInZ_TB
5. The equivalent circuit of SProbePair is shown.
**TimeDelta (Time Delta Component)**

**Symbol**

![Diagram of TimeDelta Component]

**Available in ADS**

**Parameters**

- Direction1 = direction one
- Direction2 = direction two
- Thresh1 = threshold one, in volts
- Thresh2 = threshold two, in volts
- Scale = scale factor

**Notes/Equations**

1. TimeDelta generates an output voltage proportional to the time difference between two trigger points on two different baseband input voltage waveforms. The trigger points are user-defined by setting a threshold voltage and a slope. The slope can be specified as either rising or falling by setting the direction parameter to a 1 or -1. A direction parameter value of 0 is used if a trigger for either slope is desired.

2. Only the baseband component of the input voltages are used to generate the triggers, so this model can be used in either envelope or transient time domain analysis modes. Linear interpolation is used to estimate the actual trigger crossing time to a finer resolution than the simulation time step.

3. The input impedances are infinite. The output impedance is 1 ohm. The open circuit output voltage is equal to the time difference between the trigger2 and trigger1 events, multiplied by the scaling factor. The output does not change until a trigger2 event occurs and is held constant until another trigger2 event occurs. The scaling factor is used so that the output voltages can be set to reasonable values (i.e., not nanovolts) which would often be less than the simulator's absolute convergence criteria.

4. Several example measurements possible with this model might be the input to output propagation delay of a circuit, the -40 to +20 dBm rise time of a demodulated RF pulse, various fall times, pulse widths, etc. The output voltage
Probe Components

...can be used for other behavioral models, for optimization, or for output to presentations.
TimeFrc (Time Frequency Component)

Symbol

Available in ADS

Parameters
Direction = direction one
Thresh = threshold one, in volts
Scale = scale factor
Probe Components

**TimePeriod (Time Period Component)**

**Symbol**

![Symbol Image]

**Available in** ADS

**Parameters**

- Direction = direction one
- Thresh = threshold one, in volts
- Scale = scale factor

**Notes/Equations**

1. This time period model generates an output voltage proportional to the time between two consecutive triggers. The trigger point is defined by setting a threshold voltage and a slope. The slope can be specified as either rising or falling by setting the direction parameter to a 1 or −1. A direction parameter value of 0 is used if a trigger for either slope is desired.

2. Only the baseband component of the input voltages is used to generate the trigger, so this model may be used in either envelope or transient time domain analysis modes. Linear interpolation is used to estimate the actual trigger crossing time to a significantly higher resolution than the simulation time step.

3. The input impedance is infinite. The output impedance is 1 ohm. The open circuit output voltage is equal to the time difference between the last two trigger events multiplied by the scaling factor. The output does not change until a trigger event occurs and is held constant until another event occurs. The scaling factor is used so that the output voltage can be set to reasonable value which might otherwise be less than the simulator’s absolute convergence criteria.
TimeStamp (Time Stamp Component)

Symbol

Available in ADS

Parameters

Direction = direction one

Thresh = threshold one, in volts

Scale = scale factor

Notes/Equations

1. TimeStamp generates an output voltage proportional to the time that the last user-defined trigger occurred. The trigger point is defined by setting a threshold voltage and a slope. The slope can be specified as either rising or falling by setting the direction parameter to a 1 or –1. A direction parameter value of 0 is used if a trigger for either slope is desired.

2. Only the baseband component of the input voltages is used to generate the trigger, so the model may be used in either envelope or transient time analysis modes. Linear interpolation is used to estimate the actual trigger crossing time to a significantly higher resolution than the simulation time step domain.

3. The input impedance is infinite. The output impedance is 1 ohm. The open circuit output voltage is equal to the time of the last trigger event multiplied by the scaling factor. The output does not change until a trigger event occurs and is held constant until another event occurs. The scaling factor is used so that the output voltage can be set to a reasonable value which might otherwise be less than the simulator's absolute convergence criteria.
Probe Components

WaveformStats (WaveformStats Component)

Symbol

Available in ADS

Parameters
None

Notes/Equations

1. This behavioral model can be used to measure the statistics of the baseband component of the input voltage. The inputs all have infinite input impedance; all outputs are ideal voltage sources with zero output impedance.

2. It calculates the running statistics of the signal since the last time reset went to 0; the reset signal should be high (1) for normal operation. The enable signal must be high (1) for normal operation; it can be put in hold mode temporarily by bringing the enable signal to 0. The calculated running statistics are mean, standard deviation, minimum, maximum, sum, and number of samples of the input signal.

3. If the enable is low during a reset, the accumulators are reset to 0; if the enable is high, then N is set to 1, and Sum is set equal to the input.

4. In addition to making gated, statistical measurements for use in optimizations or presentations, you can use this device to model circuits such as ideal integrate-and-dump circuits or peak detector circuits.

5. To measure the statistics of an RF carrier in circuit envelope mode, the correct demodulator must first be used to create a baseband voltage that can then be used as an input to this device.

6. This model operates in transient and envelope time domain analysis modes.

7. The schematic example shows how this component works and a plot of the signals after simulating it.
Probe Components
Chapter 5: Linear Data File Components
Linear Data File Components

**DataAccessComponent (Data Access Component)**

**Symbol**

![DAC symbol]

**Available in ADS and RFDE**

The DataAccessComponent contains the following parameters types:

- **File** – File name, file type and block name if there is more than one block of data in the file.
- **Independent Variable** – pairs of independent variable names and values.
- **Interpolation** – parameters related to interpolation setup.
- **Display (ADS)** – Control the visibility of component parameters on the Schematic. For details, refer to the topic “Displaying Simulation Parameters on the Schematic” in the chapter, Simulation Basics, in the Using Circuit Simulators manual.
Setting the File Parameters (ADS)

Use the File tab to specify file related parameters.

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>filename</td>
</tr>
<tr>
<td>Type</td>
<td>file type</td>
</tr>
<tr>
<td>Block</td>
<td>block name</td>
</tr>
</tbody>
</table>
Linear Data File Components

**File Name**
The file name can be defined in either Data filename entry mode or File Based entry mode. When Data filename entry mode is selected, file name can be given as a string which represents explicit file name or assigned through a variable as @variable_name.

- Select Edit... to display the file or modify its contents.
- Select Copy Template... to copy a data file of a specific format for use as a template.
- Select Data files list... to select a data file in the current project.

**File Type**
Provides a drop down list of available file types.
Block
Using Block Name is optional. If you do not specify a block name, the block of data contained in the dataset file is used.

DataSet Viewer
The View Dataset button is only enabled when an explicit file name is given and the file type is dataset. If a file name is given with a relative path, `<cur_prj>/data/` is pre-appended to the file name by ADS. Click View Dataset to open a new window displaying the data blocks contained in the file and information about the independent and dependent variables in the selected block. ADS automatically sets up Block and iVar*(independent variable names) if a block is selected and OK is chosen in the dataset viewing window.
Linear Data File Components

![Dataset Viewer6 interface with file selection and variable information]

- **Independent Variables**
  - Index: Integer

- **Dependent Variables**
  - HL: Real

---

5-6  DataAccessComponent (Data Access Component)
When no data block is specified, the Dataset Viewer defaults to the first data block in the file. If there is more than one block of data, you can click another block to select it and view its variable information.
Linear Data File Components

Setting Independent Variable Parameters (ADS)

Use the Independent Variable tab to specify independent variable names and values.

Parameters

<table>
<thead>
<tr>
<th>iVar1, ..., iVar10</th>
<th>independent variable name or cardinality (1: outermost)</th>
</tr>
</thead>
<tbody>
<tr>
<td>iVal1, ..., iVal10</td>
<td>independent variable value or index (0: first/starting index)</td>
</tr>
</tbody>
</table>

The spread sheet on the left hand side of this tab lists the name and value of the independent variable. The spread sheet itself is not editable. The input fields are on right hand side of the tab.

To add the name and value of an independent variable:

1. Type the variable name in Variable Name input field.

5-8  DataAccessComponent (Data Access Component)
2. Give the variable value in Variable Value “Value” input field.

3. Click Add to insert the pairs to the left hand side spread sheet.

To remove an independent variable setup:

1. Select the row to be deleted from the spread sheet.

2. Click Cut to remove the selected row.

To edit an independent variable setup:

1. Select the row to be edited from the spread sheet.

2. Modify the name or value on the right hand side input fields.
Linear Data File Components

Setting Interpolation Parameters (ADS)

Use the Interpolation tab to specify setup related parameters.
Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InterpMode</td>
<td>interpolation mode:</td>
</tr>
<tr>
<td></td>
<td><strong>Index Lookup</strong> Specifies that ( n ) represents the integer indices (beginning with 0) of the independent variables in the data file. Real ( n ) values are truncated first for index lookup.</td>
</tr>
<tr>
<td></td>
<td><strong>Value Lookup</strong> For real/integer independent variable, accesses the point in the data file closest to the specified value. If midway, the average of the bracketing points is used.</td>
</tr>
<tr>
<td></td>
<td><strong>Ceiling Value Lookup</strong> For a real independent variable, accesses the nearest point in the data file not less than the specified value.</td>
</tr>
<tr>
<td></td>
<td><strong>Floor Value Lookup</strong> For a real independent variable, accesses the nearest point in the data file not greater than the specified value.</td>
</tr>
<tr>
<td></td>
<td><strong>Linear, Cubic, Cubic Spline</strong> Specifies the interpolation mode in each dimension (except for splines, where only the innermost variable is spline-interpolated).</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong> This is provided if the interpolation mode is variable or unknown, for example, as a passed parameter of a subnetwork. The resulting value should be a string (or integer) from the following set: {&quot;linear&quot;(0), &quot;spline&quot;(1), &quot;cubic&quot;(2), &quot;index_lookup&quot;(3), &quot;value_lookup&quot;(4), &quot;ceiling_value_lookup&quot;(5), &quot;floor_value_lookup&quot;(6)}.</td>
</tr>
<tr>
<td>InterpDom</td>
<td>interpolation domain:</td>
</tr>
<tr>
<td></td>
<td><strong>Rectangular</strong> Interpolates real and imaginary parts separately; recommended for emittances.</td>
</tr>
<tr>
<td></td>
<td><strong>Polar</strong> (arc interpolation) Interpolates magnitude and angle separately; recommended for S-parameters.</td>
</tr>
<tr>
<td></td>
<td><strong>DB</strong> Interpolates in dB and angle format.</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong> This is provided if the interpolation domain is a variable or unknown; for example as a passed parameter of a subnetwork. The resulting value should be a string (or integer) from the following set: {&quot;ri&quot; (0), &quot;ma&quot; (1), &quot;db&quot; (3)}.</td>
</tr>
<tr>
<td>ExtrapMode</td>
<td>extrapolation mode:</td>
</tr>
<tr>
<td></td>
<td><strong>Interpolation Mode</strong>: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.</td>
</tr>
<tr>
<td></td>
<td><strong>Constant Extrapolation</strong>: when extrapolation occurs, no interpolation is performed. The value of the nearest data point is returned.</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This component can be used to extract/interpolate multidimensional dependent variables as a function of up to 10 independent variables. By setting the DAC File parameter to the desired filename, and setting the parameter of the component of interest to point to the DAC (by Instance ID), the data in the specified file can be accessed. (Refer to "Example 1" on page 5-13)
You can quickly set all parameters (with matching names) of a device model by setting the model’s `AllParams` parameter to the DAC’s Instance ID, which in turn, references the data file. Parameter names in a data file that are not device model parameters are ignored. A device model parameter value that is explicitly specified will override the value set by an `AllParams` association. (Refer to “Example 2” on page 5-14)

You can also sweep over several BJT models using two DAC components. (Refer to “Example 3” on page 5-15)

S-parameter data can be read directly from a Touchstone file using a DAC. (Refer to “Example 4” on page 5-16).

Ptolemy simulation examples using DAC are located in the Controllers_Pri project; to access the example project from the ADS Main window, click on File > Example Project > PtolemyDocExamples > Controllers_Pri, then open the Read_DAC_DSCR.dsn and Read_DAC_MDIF.dsn designs.

2. For a complex dependent variable, the two parts (real/imag, mag/degree, or dB/degree) are interpolated separately. For arc-like data (for example S-parameters vs. frequency), it may be more appropriate to interpolate in the mag/degree domain.

3. This component is actually a special subnetwork whose expressions can be used outside. In particular, one of these expressions is `_TREE` (the multi-dimensional table). The following example shows using this expression with the `get_max_points` function.

   Example:  
   ```
   get_max_points(DAC1._TREE, "freq")
   ```

   where:

   - `DAC1._TREE` represents the Instance ID of the DAC
   - "freq" represents the name of the independent variable

   It returns the maximum # of points (over all sweeps of that variable) of the independent variable (for discrete files with implicit row #, use 1 for the second argument)

   4. The `Type` parameter specifies the format of the disk file, which includes Touchstone, CITIfile, several MDIF types, SPW and binary datasets (possibly from a previous simulation or via instrument server).
The files displayed in the Browser represent all files found based on the search paths specified by the DATA_FILES configuration variable.

For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

5. The Block name specifies which table to use when the file contains two or more multidimensional tables, (e.g., “ACDATA”, “NDATA” in an MDIF file, “HB1.HB”, “HB1.HB_NOISE” in a harmonic balance analysis dataset). A unique prefix is sufficient; it can also be the sequence number (starting with 1) of the table, for example, 1 for an “ACDATA” table and 2 for “NDATA”. Note that the “at” symbol (@) should be used to suppress quotes when using a variable to identify a table as the independent variable for making DAC parameter assignments.

6. Each iVar is either the name of an independent variable in the file (e.g., Vgs) or is an integer representing the cardinality or nesting order of the independent variable (1 being outermost). A cardinal value must be used when an independent variable is implicit; for example, row index in discrete files is the innermost independent variable. Note that @ must be used to suppress quotes when using a variable, for example, @freq1, where freq1 is a variable declared in a VAR item.

When the DataAccessComponent refers to a time-domain MDIF file which has a .tim extension, the iVar parameter must be set to time and the reference to the dependent parameter must be set to voltage, independent of the names of the columns in the .tim file.

A string iVar parameter is searched in a case-preferential manner, i.e., it is searched in a case-sensitive manner, failing that, it is searched again in a case-insensitive manner.

7. Each iVal is normally a real or integer value of the independent variable to bracket or search for in the file. If InterpMode = Index Lookup (which must be the case for implicit variables), this value is the integer index, starting from 0. For example, the row value for a discrete file block runs from 0 to #rows – 1.

8. For all value lookup modes, a tolerance of 0.01% is used. A warning message is issued when extrapolation occurs.

Example 1

In this example, the resistance of R1 is stepped through all values under the R column in the “r.mdf” file
Example 2

In this example, resistor model RM1 accesses the $R$, $R_{sh}$, \textit{Length} and \textit{Width} parameters from the discrete "r.mdf" file.
Example 3

This example illustrates how a pair of DACs can be used to sweep over several BJT models. The first DAC, STEER, retrieves a model filename from a discrete file bfqtm.txt, and the second DAC, DAC_BJT, retrieves the model data.
Linear Data File Components

**Note** An assignment of the type: $R1=\text{file}\{\text{DAC1, "Rnom"}\}$, is equivalent to the expression $R1=\text{dep\_data(DAC1\_DAC, "Rnom")}$.

**Example 4**

This example illustrates reading S-parameter data from a Touchstone file using the DataAccessComponent.

---

5-16  DataAccessComponent (Data Access Component)
DataAccessComponent (Data Access Component) 5-17

# hz S ma R 50
! 2 Port Network Data from SP1.SP block

1e+009 0.85 -32 0.53 58 0.53 58 0.85 -32
2e+009 0.80 -35 0.57 55 0.57 55 0.82 -35
3e+009 0.72 -37 0.61 53 0.6 53 0.8 -37
Deembed1 (1-Port De-Embed Data File)

Symbol

Available in ADS and RFDE

Parameters
File = name of .s1p file containing 1-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s1p and the default directory is <prj>/data where <prj> is your current project directory.

Type = file type: Touchstone, Dataset, CITIfile

Block = (for Type=Dataset) name of S-parameter data block

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup

InterpDom = interpolation domain: Data Based, Rectangular, Polar, DB

ExtrapMode = extrapolation mode: Interpolation Mode, Constant Extrapolation

Temp = physical temperature, in °C

Range of Usage
Within the frequency range of the S-, Y-, or Z-parameter file

Notes/Equations
1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

2. One of the Deembed1 data file applications is to negate the 1-port subcircuit by using this data file.

3. In the following example, the BJT emitter's parasitics leads are de-embedded to obtain just the chip BJT. This ideal short and open behavior is not guaranteed if the deembed circuit has one or more frequency bands where a stop behavior is observed.
4. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.

5. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated.
   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation
   - DB: transform to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

6. ExtrapMode specifies the extrapolation mode.
   - Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   - Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.

7. This component does not generate any noise.

8. For time-domain analysis, the impulse response used for transient will be noncausal. This model should not be used for transient or circuit envelope analysis.

9. This component has no default artwork associated with it.
Linear Data File Components

Deembed2 (2-Port De-Embed File)

Symbol

Available in ADS and RFDE

Parameters

File = name of .s2p file containing 2-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s2p and the default directory is <prj>/data where <prj> is your current project directory.

Type = file type: Touchstone, Dataset, CITI file

Block = (for Type=Dataset) name of S-parameter data block

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup

InterpDom = interpolation domain: Data Based, Rectangular, Polar, DB

ExtrapMode = extrapolation mode: Interpolation Mode, Constant Extrapolation

Temp = physical temperature, in °C

Range of Usage

Within the frequency range of the S-, G-, H-, Y-, or Z-parameter file

Notes/Equations

1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

2. One of the Deembed2 data file applications is to negate the 2-port subcircuit by using this data file.

3. When this component is connected in series with the sub-circuit being negated, and the hookup is back-to-back as shown in the following illustration, the result is a short. When this component is connected in parallel, with the subcircuit being negated, the result is an open. This ideal short and open behavior is not guaranteed if the deembed circuit has one or more frequency bands where a stop behavior is observed.
Another example is to de-embed the parasitics of the leads in a transistor. In the following illustration the base and collector parasitics are de-embedded to give just the chip BJT.

4. The 2-port S-parameters are assumed to be measured with pin 1 as the input, pin 2 as the output, and pin 3 as the common terminal.

5. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.

6. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation
   - DB: transform to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

7. ExtrapMode specifies the extrapolation mode.
   - Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
Linear Data File Components

Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.

8. This component does not generate any noise.

9. For time-domain analysis, the impulse response used for transient will be noncausal. This model should not be used for transient or circuit envelope analysis.

10. This component has no default artwork associated with it.
**NetlistInclude (Netlist File Include Component)**

**Symbol**

![Netlist Include Symbol]

**Available in** ADS

**Parameters**

- **IncludePath** = Space-delimited search path for included files
- **IncludeFiles** = List of files to include
- **UsePreprocessor** = yes (default) to use an `#include` directive; no to copy the full text of the file

**Notes/Equations**

1. The NetlistInclude component provides a mechanism for the ADS simulator to use an external file.

   Previous versions of built-in netlist include components (spiceInclude, geminiInclude and idfInclude) can be placed on a schematic by typing their names into the Component History field in the schematic window; for these components, you must manually enter the name of the included file. Beginning with ADS 2002, NetlistInclude is the recommended mechanism for including external files, though the deprecated components may continue to work.

2. The **IncludePath** parameter is a space-delimited search path that locates included files. Using the Browse button to select values for IncludeFiles will automatically add to IncludePath as needed. Note that, in directory names, path prefixes such as the dot (.), dot-dot (..), tilde (~), and dollar sign ($) all have their usual UNIX interpretation. The forward slash (/) should be used as the directory delimiter, even on Windows.

3. The **IncludeFiles** parameter enables you to build a list of netlist files that you want to include in the simulation. Use the Add button to include more than one file with a single NetlistInclude component.

   Each included file may have an optional Section designator. This enables you to include only a portion of a file, provided that the file has been set up properly. Establishing sections within a file requires bracketing the sections using `#ifdef` `<section>` and `#endif` directives. As an example, this file defines a subcircuit and two sections, SelectR and SelectC:
Linear Data File Components

```c
#define RCsub ( in out )
#ifdef SelectR
R:R1 in out R=50 Ohm
#endif
#ifdef SelectC
C:C1 in out C=1.0 pF
#endif
end RCsub
```

If this file is included with the SelectR section designated, the simulator will read the file as:

```c
define RCsub ( in out )
R:R1 in out R=50 Ohm
end RCsub
```

Similarly, with the SelectC section designated, the simulator will see only the capacitor. By using the Add button to add the file twice, both sections can be specified, and the simulator will read the file as:

```c
define RCsub ( in out )
R:R1 in out R=50 Ohm
C:C1 in out C=1.0 pF
end RCsub
```

4. The UsePreprocessor parameter selects the exact mechanism by which the listed files are included. If UsePreprocessor is set to yes, the netlister generates a set of preprocessor directives such as:

```c
#ifndef inc__users_default_default_prj_models_resistor_lib
#define inc__users_default_default_prj_models_resistor_lib
#include "users/default/default_prj/models/resistor.lib"
#endif
```

As the simulator reads the netlist, it will also read the referenced file (/users/default/default_prj/models/resistor.lib in this example). This may cause a problem for remote simulations, since the simulation machine may not be able to find that file at the same path. In this case UsePreprocessor should be set to no, which instructs the netlister to copy the file in its entirety into the netlist. This option will work for both local and remote simulations, but it may be noticeably slower. The speed difference is directly related to the size of the included files.

The ifndef, define, and endif lines are used to guard against attempts to include the same file more than once.

5. Example component parameters:
IncludeFiles[1]="functions.def"
IncludeFiles[2]="resistor.lib Nominal"
IncludePath="C:/ADS/my_prj ./misc"
UsePreprocessor=yes

A NetlistInclude component with these parameters would generate the following netlist output:

```c
#ifndef inc_C__ADS_my_prj_misc_functions_def
#define inc_C__ADS_my_prj_misc_functions_def
#include "C:\ADS\my_prj\misc\functions.def"
#endif
#define Nominal
#ifndef inc_C__ADS_my_prj_resistor_lib
#define inc_C__ADS_my_prj_resistor_lib
#include "C:\ADS\my_prj\resistor.lib"
#endif
#undef Nominal
```

6. Use caution when placing a NetlistInclude component in a subcircuit. If an included file contains a subcircuit definition, the simulator will find one subcircuit definition inside another, and will stop after reporting a syntax error. Included files containing subcircuit definitions must be referenced from a top-level design.
Linear Data File Components

S1P (1-Port S-parameter File)

Symbol

Available in ADS and RFDE

Parameters

File = name of data file containing 1-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s1p and the default directory is <prj>/data where <prj> is your current project directory.

Type = file type: Touchstone, Dataset, CITI file

Block = (for Type=Dataset) name of S-parameter data block

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup

InterpDom = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB

ExtrapMode = extrapolation mode: Interpolation Mode, Constant Extrapolation

Temp = physical temperature, in °C

ImpNoncausalLength = non-causal function impulse response order (value type: integer)

ImpMode = convolution mode (value type: integer)

ImpMaxFreq = maximum frequency to which device is evaluated, in hertz

ImpDeltaFreq = sample spacing in frequency, in hertz

ImpMaxOrder = maximum impulse response order (value type: integer)

ImpWindow = smoothing window (value type: integer)

ImpRelTol = relative impulse response truncation factor

ImpAbsTol = absolute impulse response truncation factor

Notes/Equations

1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.
2. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.

3. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

4. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation
   - DB: transform to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

5. ExtrapMode specifies the extrapolation mode.
   - Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   - Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.

6. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.

7. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous). Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   - If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

8. Ref pin 2 is the common terminal; it is normally grounded, but can be used in non-grounded mode.

9. For time-domain analysis, the frequency-domain S-parameters are used.

10. This component has no default artwork associated with it.
Linear Data File Components

S2P (2-Port S-parameter File)
Symbol

Available in ADS and RFDE

Parameters

File = name of data file containing 2-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s2p and the default directory is <prj>/data where <prj> is your current project directory.

Type = file type: Touchstone, Dataset, CITIfile

Block = (for Type=Dataset) name of S-parameter data block

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup

InterpDom = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB

ExtrapMode = extrapolation mode: Interpolation Mode, Constant Extrapolation

Temp = physical temperature, in °C

ImpNoncausalLength = non-causal function impulse response order (value type: integer)

ImpMode = convolution mode (value type: integer)

ImpMaxFreq = maximum frequency to which device is evaluated, in hertz

ImpDeltaFreq = sample spacing in frequency, in hertz

ImpMaxOrder = maximum impulse response order (value type: integer)

ImpWindow = smoothing window (value type: integer)

ImpRelTol = relative impulse response truncation factor

ImpAbsTol = absolute impulse response truncation factor

Notes/Equations
1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

2. The 2-port S-parameters are assumed to be measured with pin 1 as the input, pin 2 as the output, and pin 3 (Ref) as the common terminal.

   The Ref node (common terminal) is normally grounded; but can also be used in non-grounded mode under special circumstances. For example, an Inductor in series with ref port before ground (that is, other end of inductor grounded) can be used in a BJT model S-parameter file to convert the amplifier to an oscillator.

3. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.

4. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

5. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation.
   - DB: transform to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

6. ExtrapMode specifies the extrapolation mode.
   - Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   - Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.

7. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the file contains the noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn), these parameters are used to calculate the devices noise performance, independent of Temp. If the S-parameters describe a
Linear Data File Components

passive device, then Temp and Twiss’s theorem are used to calculate its noise
performance. If the S-parameters describe an active device, no noise is
generated.

8. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).

   If these values are not specified, they default to the corresponding global
   parameter values specified by the transient analysis controller item.

9. For time-domain analysis, the frequency-domain S-parameters are used.

10. This component has no default artwork associated with it.
S2P_Conn (2-Port S-parameter File; connector artwork)

Symbol

Available in ADS

Parameters

File = name of data file containing 2-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s2p and the default directory is <prj>/data where <prj> is your current project directory.

Type = file type: Touchstone, Dataset, CITIfile, Value

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup, Ceiling Value Lookup, Floor Value Lookup, Value

InterpDom = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB

Temp = physical temperature, in °C

Side = top or bottom

Range of Usage

S-, G-, Y-, or Z-parameters

Notes/Equations

1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

2. The 2-port S-parameters are assumed to be measured with pin 1 as the input, pin 2 as the output, and pin 3 (Ref) as the common terminal.

   The Ref node (common terminal) is normally grounded; but can also be used in non-grounded mode under special circumstances. For example, an Inductor in series with ref port before ground (that is, other end of inductor grounded) can be used in a BJT model S-parameter file to convert the amplifier to an oscillator.
Linear Data File Components

3. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation.
   - DB: transform to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

4. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

5. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the file contains the noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn), these parameters are used to calculate the devices noise performance, independent of Temp. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.

6. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous). Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).

   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

7. For transient analysis, pins 1 and 2 are shorted together.

8. For convolution analysis, the frequency-domain S-parameters are used.
S2P_Pad3 (2-Port S-parameter File; pad artwork)

Symbol

Available in ADS

Parameters

File = name of data file containing 2-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s2p and the default directory is <prj>/data where <prj> is your current project directory.

Type = file type: Touchstone, Dataset, CITIfile, Value

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup, Ceiling Value Lookup, Floor Value Lookup, Value

InterpDom = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB

Temp = physical temperature, in °C

W1 = (ADS Layout option) width of line at pins 1 and 2, in length units

W2 = (ADS Layout option) width of line at pin 3, in length units

S = (ADS Layout option) spacing (length from pin 1 to pin 2, in length units)

L1 = (ADS Layout option) length from pin 1 to pin 2, in length units

L2 = (ADS Layout option) length between pin 3 to pins 1 and 2, in length units

Side = top, bottom

Range of Usage

S-, G-, H-, Y-, or Z-parameters

Notes/Equations

1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.
Linear Data File Components

2. The 2-port S-parameters are assumed to be measured with pin 1 as the input, pin 2 as the output, and pin 3 (Ref) as the common terminal.

The Ref node (common terminal) is normally grounded; but can also be used in non-grounded mode under special circumstances. For example, an Inductor in series with ref port before ground (that is, other end of inductor grounded) can be used in a BJT model S-parameter file to convert the amplifier to an oscillator.

3. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation
   - DB: transform to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

4. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

5. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the file contains the noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn), these parameters are used to calculate the devices noise performance, independent of Temp. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.

6. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   - Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   - If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

7. For transient analysis, pins 1 and 2 are shorted together.

8. For convolution analysis, the frequency-domain S-parameters are used.
S2PMDIF (Multi-Dimensional 2-Port S-parameter File)

Symbol

Available in ADS and RFDE

Parameters

File = name of MDIF file containing 2-port S-, G-, H-, Y-, or Z-parameters for this component, with optional noise parameters. The file extension and directory path are optional. Default extension is .s2p and the default directory is <prj>/data where <prj> is your current project directory.

Type = file type: S2PMDIF, Touchstone, Dataset, CITIfiile

Block = (for Type=Dataset) name of S-parameter data block

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup

InterpDom = interpolation domain: Data Based, Rectangular, Polar, DB

ExtrapMode = extrapolation mode: Interpolation Mode, Constant Extrapolation

Temp = physical temperature, in °C

ImpNoncausalLength = non-causal function impulse response order (value type: integer)

ImpMode = convolution mode (value type: integer)

ImpMaxFreq = maximum frequency to which device is evaluated, in hertz

ImpDeltaFreq = sample spacing in frequency, in hertz

ImpMaxOrder = maximum impulse response order (value type: integer)

ImpWindow = smoothing window (value type: integer)

ImpRelTol = relative impulse response truncation factor

ImpAbsTol = absolute impulse response truncation factor

Note For more information on interpolation refer to “Setting Interpolation Parameters (ADS)” on page 5-10
Notes/Equations

1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

2. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.

3. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

4. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation
   - DB: transform to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

   Note that no matter in which format the data is given, it is stored in Real/imag. If the InterpDom is DB, the data is transformed to DB/angle before the interpolation is performed.

5. ExtrapMode specifies the extrapolation mode.
   - Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   - Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.

6. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the file contains the noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn), these parameters are used to calculate the devices noise performance, independent of Temp. If the S-parameters describe a passive device, then Temp and Twiss’s theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.
7. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   If these values are not specified, they default to the corresponding global
   parameter values specified by the transient analysis controller item.

8. For time-domain analysis, the frequency-domain S-parameters are used.

9. Note that a string iVar parameter is searched in a case-preferential manner, i.e.,
   it is searched in a case-sensitive manner, failing that, it is searched again in a
   case-insensitive manner.

10. This component has no default artwork associated with it.
Linear Data File Components

**S2P_Spac (2-Port S-parameter File)**

Symbol

---

**Available in** ADS

**Parameters**

File = name of data file containing 2-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s2p and the default directory is `<prj>/data` where `<prj>` is your current project directory.

Type = file type: Touchstone, Dataset, CITIfile, Value

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup, Ceiling Value Lookup, Floor Value Lookup, Value

InterpDom = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB

Temp = physical temperature, in °C

L = length (meter, mil, in, ft)

Side = top, bottom

**Range of Usage**

S-, G-, H-, Y-, or Z-parameters

**Notes/Equations**

1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

2. The 2-port S-parameters are assumed to be measured with pin 1 as the input, pin 2 as the output, and pin 3 (Ref) as the common terminal.

   The Ref node (common terminal) is normally grounded; but can also be used in non-grounded mode under special circumstances. For example, an Inductor in series with ref port before ground (that is, other end of inductor grounded) can be used in a BJT model S-parameter file to convert the amplifier to an oscillator.
3. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation.
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

4. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

5. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the file contains the noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn), these parameters are used to calculate the devices noise performance, independent of Temp. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.

6. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

7. For time-domain analysis, the frequency-domain S-parameters are used.

8. This component has no default artwork associated with it.

9. For transient analysis, pins 1 and 2 are shorted together.

10. For convolution analysis, the frequency-domain S-parameters are used.

11. This component is represented as a connected gap in layout -- into which a custom artwork object can be inserted.
Linear Data File Components

**S3P (3-Port S-parameter File)**

**Symbol**

![Symbol Diagram]

**Available in ADS and RFDE**

**Parameters**

- **File** = name of data file containing 3-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is `.s3p` and the default directory is `<prj>/data` where `<prj>` is your current project directory.
- **Type** = file type: Touchstone, Dataset, CITIfile
- **Block** = (for Type=Dataset) name of S-parameter data block
- **InterpMode** = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup
- **InterpDom** = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB
- **ExtrapMode** = extrapolation mode: Interpolation Mode, Constant Extrapolation
- **Temp** = physical temperature, in °C
- **ImpNoncausalLength** = non-causal function impulse response order (value type: integer)
- **ImpMode** = convolution mode (value type: integer)
- **ImpMaxFreq** = maximum frequency to which device is evaluated, in hertz
- **ImpDeltaFreq** = sample spacing in frequency, in hertz
- **ImpMaxOrder** = maximum impulse response order (value type: integer)
- **ImpWindow** = smoothing window (value type: integer)
- **ImpRelTol** = relative impulse response truncation factor
- **ImpAbsTol** = absolute impulse response truncation factor

**Notes/Equations**
1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

2. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.

3. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

4. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation
   - DB: transform to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

5. ExtrapMode specifies the extrapolation mode.
   - Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   - Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.

6. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.

7. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   - Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   - If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

8. Ref pin 4 is the common terminal; it is normally grounded, but can be used in non-grounded mode.

9. For time-domain analysis, the frequency-domain S-parameters are used.
Linear Data File Components

10. This component has no default artwork associated with it.
S4P (4-Port S-parameter File)

Symbol

Available in  
ADS and RFDE

Parameters

File = name of data file containing 4-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s4p and the default directory is <prj>/data where <prj> is your current project directory.

Type = file type: Touchstone, Dataset, CITIfile

Block = (for Type=Dataset) name of S-parameter data block

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup

InterpDom = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB

ExtrapMode = extrapolation mode: Interpolation Mode, Constant Extrapolation

Temp = physical temperature, in °C

ImpNoncausalLength = non-causal function impulse response order (value type: integer)

ImpMode = convolution mode (value type: integer)

ImpMaxFreq = maximum frequency to which device is evaluated, in hertz

ImpDeltaFreq = sample spacing in frequency, in hertz

ImpMaxOrder = maximum impulse response order (value type: integer)

ImpWindow = smoothing window (value type: integer)

ImpRelTol = relative impulse response truncation factor

ImpAbsTol = absolute impulse response truncation factor
Linear Data File Components

Notes/Equations

1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

2. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.

3. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

4. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation
   - DB: transform to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

5. ExtrapMode specifies the extrapolation mode.
   - Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   - Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.

6. If the component temperature Temp is less than \(-273\)°C, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss’s theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.

7. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous). Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   - If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

8. Ref pin 5 is the common terminal; it is normally grounded, but can be used in non-grounded mode.
9. For time-domain analysis, the frequency-domain S-parameters are used.
10. This component has no default artwork associated with it.
Linear Data File Components

**S5P to S9P (5-Port to 9-Port S-parameter File)**

**Symbol**

![Symbol Diagram]

**Available in ADS and RFDE**

**Parameters**

File = name of data file containing #port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s#p and the default directory is `<prj>/data` where `<prj>` is your current project directory.

Type = file type: Touchstone, Dataset, CITIfile

Block = (for Type=Dataset) name of S-parameter data block

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup

InterpDom = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB

ExtrapMode = extrapolation mode: Interpolation Mode, Constant Extrapolation

Temp = physical temperature, in °C

ImpNoncausalLength = non-causal function impulse response order (value type: integer)

ImpMode = convolution mode (value type: integer)

ImpMaxFreq = maximum frequency to which device is evaluated, in hertz

ImpDeltaFreq = sample spacing in frequency, in hertz

ImpMaxOrder = maximum impulse response order (value type: integer)

ImpWindow = smoothing window (value type: integer)

ImpRelTol = relative impulse response truncation factor
ImpAbsTol = absolute impulse response truncation factor

Notes/Equations

1. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

2. The number of terminals increases sequentially from 5 to 9, and is equal to the number of ports of the component.
   Ref is the common terminal; it is normally grounded, but can be used in non-grounded mode.

3. If no extension is supplied with the file name, then a default value of “.s(#)p” is used, where (#) is the number of ports of the component.

4. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.

5. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

6. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation.
   DB: transform to (dB, angle) before interpolation.
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

7. ExtrapMode specifies the extrapolation mode.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.

8. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the S-parameters describe a passive device, then
Linear Data File Components

Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.

9. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

10. For time-domain analysis, the frequency-domain S-parameters are used.
11. This component has no default artwork associated with it.
S10P to S20P (10-Port to 20-Port S-parameter File)

Symbol

Available in ADS and RFDE

Parameters

File = name of data file containing #-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s#p and the default directory is <prj>/data where <prj> is your current project directory.

Type = file type: Touchstone, Dataset, CITI file

Block = (for Type=Dataset) name of S-parameter data block

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup

InterpDom = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB

ExtrapMode = extrapolation mode: Interpolation Mode, Constant Extrapolation

Temp = physical temperature, in °C

ImpNoncausalLength = non-causal function impulse response order (value type: integer)

ImpMode = convolution mode (value type: integer)

ImpMaxFreq = maximum frequency to which device is evaluated, in hertz

ImpDeltaFreq = sample spacing in frequency, in hertz

ImpMaxOrder = maximum impulse response order (value type: integer)

ImpWindow = smoothing window (value type: integer)
Linear Data File Components

**ImpRelTol** = relative impulse response truncation factor
**ImpAbsTol** = absolute impulse response truncation factor

**Notes/Equations**

1. The items S11P through S99P cannot be selected from the component palette. They are accessed by typing the appropriate name (such as S12P or S98P) into the right entry box (directly above the viewing area), pressing Enter, then moving the cursor to the viewing area to place the item.

2. The number of terminals increases sequentially from 10 to 20, and is equal to the number of ports of the component.

   Ref is the common terminal; it is normally grounded, but can be used in non-grounded mode.

3. For information on data file formats, refer to “Working with Data Files” in the Using Circuit Simulators manual.

4. If no extension is supplied with the File name, then a default value of .s(#)p is used, where (#) is the number of ports of the component.

5. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.

6. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

7. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:

   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation
   - DB: transform to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

8. ExtrapMode specifies the extrapolation mode.

   **Interpolation Mode**: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.

9. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.

10. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous). Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).

    If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

11. For time-domain analysis, the frequency-domain S-parameters are used.

12. This component has no default artwork associated with it.
Linear Data File Components

S21P to S99P (21-Port to 99-Port S-parameter File)

Symbol

Available in ADS

Parameters

File = name of data file containing #-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .s#p and the default directory is `<prj>/data` where `<prj>` is your current project directory.

Type = file type: Touchstone, Dataset, CITIfile

Block = (for Type=Dataset) name of S-parameter data block

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup

InterpDom = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB

ExtrapMode = extrapolation mode: Interpolation Mode, Constant Extrapolation

Temp = physical temperature, in °C

ImpNoncausalLength = non-causal function impulse response order (value type: integer)

ImpMode = convolution mode (value type: integer)
ImpMaxFreq = maximum frequency to which device is evaluated, in hertz
ImpDeltaFreq = sample spacing in frequency, in hertz
ImpMaxOrder = maximum impulse response order (value type: integer)
ImpWindow = smoothing window (value type: integer)
ImpRelTol = relative impulse response truncation factor
ImpAbsTol = absolute impulse response truncation factor

Notes/Equations
1. S11P through S99P support up to 99-port networks. They cannot be selected from the component palette; they are accessed by typing the appropriate name (such as S11P or S99P) into the field above the viewing area, pressing Enter, then moving the cursor to the viewing area to place the item.
2. Ref is the common terminal; it is normally grounded, but can be used in non-grounded mode.
3. The S, Y, Z, and N matrix measurements are allowed for up to 99-port networks. In addition, single measurements are applicable:
   • SIJ, for example (S(29,28))
   • VSWR, for example (S(29,29))
4. These components primarily support electromagnetic simulation results of circuits with a large number of ports, such as antenna feed networks.
5. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
6. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
7. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
Linear Data File Components

Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

8. ExtrapMode specifies the extrapolation mode.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.

9. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.

10. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

11. For time-domain analysis, the frequency-domain S-parameters are used.

12. This component has no default artwork associated with it.
SnP component (n>99)

Symbol

Available in ADS

Parameters

File = name of data file containing #port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is .snp and the default directory is <prj>/data where <prj> is your current project directory.

Type = file type: Touchstone, Dataset, CITIfile

Block = (for Type=Dataset) name of S-parameter data block

InterpMode = interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup

InterpDom = interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB

ExtrapMode = extrapolation mode: Interpolation Mode, Constant Extrapolation

Temp = physical temperature, in °C

ImpNoncausalLength = non-causal function impulse response order (value type: integer)

ImpMode = convolution mode (value type: integer)
Linear Data File Components

ImpMaxFreq = maximum frequency to which device is evaluated, in hertz
ImpDeltaFreq = sample spacing in frequency, in hertz
ImpMaxOrder = maximum impulse response order (value type: integer)
ImpWindow = smoothing window (value type: integer)
ImpRelTol = relative impulse response truncation factor
ImpAbsTol = absolute impulse response truncation factor

Notes/Equations

1. SnP component \( (n>99) \) is used to create SnP components with a port number larger than 99. They cannot be selected from the component palette; they are accessed by typing the appropriate name (such as S123P) into the field above the viewing area and press Enter, then moving the cursor to the viewing area to place the item.

2. SnP\( (n>99) \) component is not a built-in component. You must create this component by following the procedure given in the Example section below.

3. Required Files
   The following files are required when creating an SnP component:
   - \(<\text{cur\_prj}\>\text{S*P} \) (netlist definition)
   - \(<\text{cur\_Prj}\>\text{networks/SYM_S*P.dsn} \) (symbol definition)
   - \(<\text{cur\_prj}\>\text{networks/S*p.ael} \) (parameter definition)

4. Tools used to create an SnP component
   - \text{makeSnP} \hspace{1em} A Ksh script that creates the netlist definition \text{S*P} and the parameter definition \text{S*P.ael}.
     Usage: \text{makeSnP \ numPort}
   - \text{generate_snp_symbol} \hspace{1em} This is an AEL function that generates the symbol definition \text{SYM_S*P.dsn}. This must be invoked from ADS.
     Usage: \text{generate_snp_symbol(numPort)}

Example
Follow these steps to create an SnP component:

1. Set \text{HPEESOF\_DIR} from a terminal window and set \$\text{HPEESOF\_DIR} to \$\text{PATH}
2. Run the command `makeSnP <123>` from a terminal window. This creates the netlist definition (e.g., S123P) and the parameter definition (e.g., S123P.ael) in the current directory.

3. Copy the netlist definition to `<cur_prj>` directory.

4. Copy the parameter definition to `<cur_prj>/networks` directory.

5. Open the project in ADS.

6. From the Main window select **Tools > Command Line** and type:
   
   ```
   generate_snp_symbol(123)
   ```

   where:
   
   123 is the name of the SnP component.

   ![Command Line Window](image)

   This generates the SYM_S123P.dsn file in `<cur_prj>/network` and loads the device into ADS.

7. Type the name of the component (e.g., S123P) in the schematic component history list to place it into the Schematic window.

If the component symbol (e.g., SYM_123.dsn) already exists, or if it was created before performing steps 1 through 3, you must load the SnP component definition into ADS manually.

Once all the required files are in place, do one of the following:
Linear Data File Components

- Reopen the project (ADS will load all designs that are defined in the `<cur_prj>/network` directory.
- Load the device definition manually:
  From the Main window select **Tools > Command Line** and type:

```bash
load( strcat( getcwd(),"/networks/S123P"))
```

where:
- `S123P` is the name of the SnP component.
VAR (Variables and Equations Component)

Symbol

Available in ADS

Parameters

$X =$ name of variable or equation

Notes/Equations

1. A schematic can include any number of VAR items. A VAR item can define multiple variables or equations.

   All variables and equations have the form LHS=RHS, where LHS is the name of the variable or equation to the left of the equality symbol = ; RHS is the value or expression to the right of the equality symbol. Variable and equation names (LHS) must begin with a letter and cannot exceed 32 characters. Names cannot begin with an underscore (_) unless it is one of the program-reserved variables explained later. Names are case sensitive; for example, $X$ and $x$ are different names.

2. Variable or Equation Entry Mode

   • Name=Value. Equations are defined when Variable or Equation Entry Mode is set to Name=Value and multiple variables and equations can be entered into the field provided. Equation values (RHS) must be an expression that equates to a numeric or a string value. An equation numeric value can be complex and the complex operator $j$ is recognized; for example, $z=x+j*y$, where $x$ and $y$ can be real or complex numbers or functions. The equation value can use built-in constants (refer to note 3) and functions (refer to note 4).
Linear Data File Components

- **Standard.** Variables are defined when the Variable or Equation Entry Mode is set to Standard, a single variable can be entered into the fields provided. Variable Value must be a numeric value (2.567, for example) or a string value enclosed in double-quote symbols. For example, the string value for a precision type of parameter can be defined as 2.14 for Signal Processing, or “MSUB1” for Circuit. Variable values can also be defined as a nominal value with associated optimization range.

Note that expression X has a numeric value; expression Y uses a predefined constant; expression Z uses a predefined function.
• **File Based.** To use variable or equation data from a file, reference a DataAccessComponent placed in your currently active design. For more information on the use of DAC data files, refer to DataAccessComponent.

**Name** is the name of the file you are referencing, as identified in the DataAccessComponent.

**Data Access Component Instance** is the instance name of the particular DataAccessComponent that you are referencing.

**Dependent Parameter Name** is the name of a DataAccessComponent parameter for which you want to include data.
3. Pre-defined Built-in Constants

The pre-defined built-in constants available for use in an equation are here.

- \( e \approx 2.718 \) 282 ...
- \( \ln(10) \approx 2.302 \) 585 ...
- \( c_0 \approx 2.997 \) 924 58 e+08 m/s speed of light
- \( e_0 \approx 8.854 \) 188 ... e-12 F/m vacuum permittivity
- \( u_0 \approx 1.256 \) 637 ... e-06 H/m vacuum permeability
- \( \text{boltzmann} \approx 1.380 \) 658 e-23 J/K Boltzmann's constant
- \( q_{\text{electron}} \approx 1.602 \) 177 33 e-19 C charge of an electron
- \( \text{planck} \approx 6.626 \) 075 5 e-34 J*s Planck's constant
- \( \pi \approx 3.141 \) 593 ...

4. Simulator Expressions (VarEqn functions)

Known as Simulator Expressions or sometimes as VarEqn functions. These expressions or functions can be entered into the program by means of the VarEqn component or used in place of a parameter for any component; for
example in a resistor, $R=\sin 5$. These functions are evaluated at the start of simulation. If a term is undefined at the start of simulation, such as $R=S_{11}$, where the results of $S_{11}$ will not be known until the simulation is complete, an error will be returned.

Function arguments have the following meaning.

- $x, y$ are complex
- $r, r_0, r_1, r_x, r_y, lower\_bound, upper\_bound$ are real
- $s, s_1, s_2$ are strings

In general, the functions return a complex number, unless it is a string operator as noted. Refer to Chapter 1, in the Simulator Expressions manual for a complete list of simulator expressions. A function that returns a real value effectively has a zero value imaginary term.

5. Equation Editor Syntax

Mathematical expressions entered equations can include the following items.

- **Blank spaces** Blank spaces within an expression are ignored; they can be used to improve readability. For example, $4 * (x + .1)$ evaluates the same as $4*(x+.1)$

- **Numerical constants** Real numbers such as 12.68, exponential notation numbers such as 1e6 or 25.1e3, pi can be used, and complex numbers can be defined. For example, $z = x + j*y$.

- **LHS form assignment** The LHS assignment takes the form of integer, double, complex or string dependent on what form is associated with the RHS. For example, $X=4, Y=4.0, W=1.0+j*3.0, Z=4^*$ associates the form of integer, double, complex and string to $X, Y, W$ and $Z$, respectively. The LHS form is important when subsequently used in following expressions.

- **Mathematical operators** Standard operators are available:
  - **exponentiation**
  - $^*$ exponentiation
  - multiplication
Linear Data File Components

/ division
+ addition
− subtraction

In evaluating an expression, operator precedence is: ** ^ * / + − . Operators at the same level (for example */ ) are evaluated left to right. Any number of parentheses pairs can be used to modify an expression in the usual way. For example

C10 * (1 + .005) evaluates differently than
C10 *1 + .005

• Parameters of a parametric subnetwork  Any formal parameters that are passed into a parametric subnetwork can be included in equations defined in that subnetwork. These parameters are defined for a schematic design using the File > Design/ Parameters menu selection.

• Use of if...then...else...endif statements  An equation can use a conditional statement: if ( conditional expression) then ( expression1) else (expression2) endif. For example,

  X =1
  Y =if ( X>0) then ( cos( pi/8) ) else ( sin( pi/8) ) endif

  The conditional expression can be a simple or complex numeric conditional expression with arguments separated by the standard symbols:

  < > <= >= != && ||

  Each expression can be any valid numeric expression. The entire if...then...else...endif expression must be on one line.
Chapter 6: Equation-Based Linear Components
Chain (2-Port User-Defined Linear Chain)

Symbol

Available in ADS

Parameters

A = reverse voltage gain (v1/v2 with i2=0)
B = reverse transresistance, in ohms (v1/i2 with v2=0)
C = reverse transconductance, in Siemens (i1/v2 with i2=0)
D = reverse current gain (i1/i2 with v2=0)
ImpNoncausalLength = non-causal function impulse response order (value type: integer)
ImpMode = convolution mode (value type: integer)
ImpMaxFreq = maximum frequency to which device is evaluated, in hertz
ImpDeltaFreq = sample spacing in frequency, in hertz
ImpMaxOrder = maximum impulse response order (value type: integer)
ImpWindow = smoothing window (value type: integer)
ImpRelTol = relative impulse response truncation factor
ImpAbsTol = absolute impulse response truncation factor

Notes/Equations

1. Port polarity is indicated by a minus (-) and a plus sign (+) on each port. Chain parameters are used when cascading a number of networks.
2. Any chain parameter that is not defined initially is set to a default value of zero and cannot be modified later. Any chain parameter that is defined initially, even if it is set to zero, can be modified and swept. It can also be swept indirectly by sweeping a variable that it depends on. State current is available for port 2.
3. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous). Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

4. Parameters A, B, C, and D can be made dependent on frequency by using the global variable freq.
Equation-Based Linear Components

Hybrid (2-Port User-Defined Linear Hybrid)

Symbol

Available in ADS

Parameters

- \( H_{11} \) = input impedance (\( v_{1}/i_{1} \) with \( v_{2}=0 \))
- \( H_{12} \) = reverse voltage gain (\( v_{1}/v_{2} \) with \( i_{1}=0 \))
- \( H_{21} \) = forward current gain (\( i_{2}/i_{1} \) with \( v_{2}=0 \))
- \( H_{22} \) = output conductance (\( i_{2}/v_{2} \) with \( i_{1}=0 \))
- \( \text{ImpNoncausalLength} \) = non-causal function impulse response order (value type: integer)
- \( \text{ImpMode} \) = convolution mode (value type: integer)
- \( \text{ImpMax Freq} \) = maximum frequency to which device is evaluated, in hertz
- \( \text{ImpDeltaFreq} \) = sample spacing in frequency, in hertz
- \( \text{ImpMaxOrder} \) = maximum impulse response order (value type: integer)
- \( \text{ImpWindow} \) = smoothing window (value type: integer)
- \( \text{ImpRelTol} \) = relative impulse response truncation factor
- \( \text{ImpAbsTol} \) = absolute impulse response truncation factor

Notes/Equations

1. Port polarity is indicated by a minus sign (-) and a plus sign (+) on each port.
2. Any \( H \)-parameter that is not defined initially is set to a default value of 0 and cannot be modified later. Any \( H \)-parameter that is defined initially, even if it is set to 0, can be modified and swept. It can also be swept indirectly, by sweeping a variable that it depends on. State current is available for port 1.
3. Allowed values for \( \text{ImpMode} \) are 1 (Discrete) and 2 (PWL Continuous). Allowed values for \( \text{ImpWindow} \) are 0 (Rectangle) and 1 (Hanning). If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.
4. Hij can be made dependent on frequency by using the global variable freq.
Equation-Based Linear Components

**S1P_Eqn to S6P_Eqn (1- to 6-Port S-parameters, Equation-Based)**

**Symbol**

![Diagram showing S1P_Eqn and S6P_Eqn symbols]

**Available in ADS**

**Parameters**

- $S[i,j] = S$-parameter in real and imaginary format
- $Z[i] = $port $i$ reference impedance, in ohms
- Recip = port is reciprocal: NO, YES
- NFmin = minimum noise figure, in dB
- Rn = noise resistance, in ohms
- Sopt = optimum noise match
- Temp = device noise temperature, in °C
- ImpNoncausalLength = non-causal function impulse response order (value type: integer)
- ImpMode = convolution mode (value type: integer)
- ImpMaxFreq = maximum frequency to which device is evaluated, in hertz
- ImpDeltaFreq = sample spacing in frequency, in hertz
- ImpMaxOrder = maximum impulse response order (value type: integer)
- ImpWindow = smoothing window (value type: integer)
- ImpRelTol = relative impulse response truncation factor
- ImpAbsTol = absolute impulse response truncation factor
Range of Usage

1 ≤ i, j ≤ port number

Notes/Equations

1. To enter a value for S[i,j], use the syntax a+j*b or complex(a,b). Expressions can be used as an entry, and the first syntax is in fact a special case with j being a reserved symbol representing complex(0,1). Thus, the first syntax can be further simplified if either real or imaginary part is zero. For example, you can enter just S[1,1]=3 instead of S[1,1]=3+j*0. Also, subtraction, negative numbers and parentheses can be used as desired. The following four entries

S[1,1]=0+j*2, S[1,1]=j*2, S[1,1]=0-j*(-2) S[1,1]=-j*(-2) are all supported and represent the same value. Similarly, the entries S[1,1]=j*1 and S[1,1]=1.0 represent the same value.

2. If a value is not entered for S[i,j], it is set to a zero default value (0, 0) and cannot be modified later. If S[i,j] is initially defined (even as zero), it can be modified and swept. It can also be swept indirectly, by sweeping a variable that it depends on. State currents are available for the port.

3. Port polarity is indicated by a minus (-) and a plus sign (+) on each port. The port can be made reciprocal by setting Recip=YES. By declaring the device to be reciprocal, S[i,j] is always forced to equal S[j,i]. Only one of the two can be defined.

4. If NFmin, Sopt, and Rn are used to characterize noise in S2P_Eqn, the following relation must be satisfied for a realistic model.

\[
\frac{Rn}{Zo} \geq \frac{T(\text{Fmin} - 1)(1 + \text{Sopt})^2}{8} \left( \frac{1 - |S_{11}|^2}{|1 - \text{Sopt} S_{11}|^2} \right)
\]

A warning message will be issued if Rn does not meet this criterion. If the noise parameters attempt to describe a system that requires negative noise (due to Rn being too small), the negative part of the noise will be set to zero and a warning message will be issued.

5. If the component temperature Temp is < -273°C, the component does not generate any noise. For S2P_Eqn only, if noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn) are specified, these parameters are used to calculate the device's noise performance, independent of Temp. If the S-parameters
Equation-Based Linear Components

describe a passive device, Temp and Twiss’s theorem are used to calculate noise performance; if the S-parameters describe an active device, (i.e., the S-parameters are not passive), non-real noise is generated resulting in meaningless noise data. Further, if the network is not passive, a warning is issued to the Simulation Status window.

6. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous). Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning). If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

7. $S[i,j]$ can be made dependent on frequency by using the global variable freq. For example, you can use a brick wall lowpass filter by using $S21=\text{if}(\text{freq} < 1 \text{ GHz})$, then 1 else 0.
**Y1P_Eqn to Y6P_Eqn (1- to 6-Port Y-parameters, Equation-Based)**

**Symbol**

![Diagram of Y1P_Eqn and Y6P_Eqn symbols]

**Available in ADS**

**Parameters**

- $Y_{i,j}$ = $Y$-parameter in real and imaginary format
- `Recip` = port is reciprocal: NO, YES
- `ImpNoncausalLength` = non-causal function impulse response order (value type: integer)
- `ImpMode` = convolution mode (value type: integer)
- `ImpMaxFreq` = maximum frequency to which device is evaluated, in hertz
- `ImpDeltaFreq` = sample spacing in frequency, in hertz
- `ImpMaxOrder` = maximum impulse response order (value type: integer)
- `ImpWindow` = smoothing window (value type: integer)
- `ImpRelTol` = relative impulse response truncation factor
- `ImpAbsTol` = absolute impulse response truncation factor

**Range of Usage**

$1 \leq i, j \leq$ port number
Notes/Equations

1. To enter a value for $Y[i,j]$, use the syntax $a+j*b$ or complex $(a,b)$. Expressions can be used as an entry, and the first syntax is in fact a special case with $j$ being a reserved symbol representing $\text{complex}(0,1)$. Thus, the first syntax can be further simplified if either real or imaginary part is zero. For example, you can enter just $Y[1,1]=3$ instead of $Y[1,1]=3+j*0$. Also, subtraction, negative numbers and parentheses can be used as desired. The following four entries $Y[1,1]=0+j*2$, $Y[1,1]=j*2$, $Y[1,1]=0-j*(-2)$ $Y[1,1]=-j*(-2)$ are all supported and represent the same value. Similarly, the entries $Y[1,1]=j^*j$ and $Y[1,1]=1.0$ represent the same value.

2. If a value is not entered for $Y[i,j]$, it is set to a zero default value $(0, 0)$ and cannot be modified later. If $Y[i,j]$ is initially defined (even as zero), it can be modified and swept. It can also be swept indirectly, by sweeping a variable that it depends on. State currents are available for the port.

3. Port polarity is indicated by a minus (-) and a plus sign (+) on each port. The port can be made reciprocal by setting Recip=YES. By declaring the device to be reciprocal, $Y[i,j]$ is always forced to equal $Y[j,i]$. Only one of the two can be defined.

4. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).

   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).

   If ImpMode, ImpMaxFreq, or ImpMaxOrder are not specified, they default to the global ImpMode specified by the transient analysis controller item.

5. $Y[i,j]$ can be made dependent on frequency by using the global variable freq.
Z1P_Eqn to Z6P_Eqn (1- to 6-Port Z-parameters, Equation-Based)

Symbol

Available in ADS

Parameters

\( Z[i,j] \) = Z-parameter in real and imaginary format

\( C[i] \) = port 1 controlling current (refer to Notes)

Recip = port is reciprocal: NO, YES

ImpNoncausalLength = non-causal function impulse response order (value type: integer)

ImpMode = convolution mode (value type: integer)

ImpMaxFreq = maximum frequency to which device is evaluated, in hertz

ImpDeltaFreq = sample spacing in frequency, in hertz

ImpMaxOrder = maximum impulse response order (value type: integer)

ImpWindow = smoothing window (value type: integer)

ImpRelTol = relative impulse response truncation factor

ImpAbsTol = absolute impulse response truncation factor

Range of Usage

\( 1 \leq i, j \leq \) port number
Notes/Equations

1. To enter a value for $Z[i,j]$, use the syntax $a+j*b$ or complex $(a,b)$. Expressions can be used as an entry, and the first syntax is in fact a special case with $j$ being a reserved symbol representing complex$(0,1)$. Thus, the first syntax can be further simplified if either real or imaginary part is zero. For example, you can enter just $Z[1,1]=3$ instead of $Z[1,1]=3+j*0$. Also, subtraction, negative numbers and parentheses can be used as desired. The following four entries $Z[1,1]=0+j*2$, $Z[1,1]=j*2$, $Z[1,1]=0-j*(-2)$ $Z[1,1]=j*(-2)$ are all supported and represent the same value. Similarly, the entries $Z[1,1]=j^*$ and $Z[1,1]=1.0$ represent the same value.

2. If a value is not entered for $Z[i,j]$, it is set to a zero default value $(0, 0)$ and cannot be modified later. If $Z[i,j]$ is initially defined (even as zero), it can be modified and swept. It can also be swept indirectly, by sweeping a variable that it depends on. State currents are available for the port.

3. Port polarity is indicated by a minus (-) and a plus sign (+) on each port. The port can be made reciprocal by setting Recip=YES. By declaring the device to be reciprocal, $Z[i,j]]$ is always forced to equal $Z[j,i]$. Only one of the two can be defined.

4. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous). Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).

   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

5. The $C[i]$ parameter can be used to model the mutual coupling between ZnP_Eqn and other components in the circuit. For example, $Z1P_Eqn_A$ is used to model a one-port block and $Z1P_Eqn_B$ is used to model another one-port block. $C[1]$ can be used to model the mutual coupling between $Z1P_Eqn_A$ and $Z1P_Eqn_B$.

6. $Z[i,j]$ can be made dependent on frequency by using the global variable freq.
Index

B
BinModel, 1-2

C
C, 2-2
C_Conn, 2-6
C_dxdy, 2-7
C_Model, 2-4
C_Pad1, 2-8
C_Space, 2-9
CAPP2, 3-2
CAPP2_Conn, 2-10
CAPP2_Pad1, 2-12
CAPP2_Space, 2-14
CAPQ, 2-16
Chain, 6-2
CIND, 3-4
Counter, 4-2
CQ_Conn, 2-18
CQ_Pad1, 2-20
CQ_Space, 2-22

D
DataAccessComponent, 5-2
Dataset Viewer, 5-5
DC_Block, 2-24
DC_Feed, 2-25
Deembed1, 5-18
Deembed2, 5-20
DICAP, 2-26
DILABMLC, 2-27

F
FORMAT A, B, C, D, E, 1-6

G
Ground, 1-7

H
Hybrid, 6-4

I
I_Probe, 4-3
INDQ, 2-29
InDQ2, 2-31

L
L, 2-33
L_Conn, 2-35
L_Model, 2-36
L_Pad1, 2-38
L_Space, 2-39
LQ_Conn, 2-40
LQ_Pad1, 2-42
LQ_Space, 2-44

M
Mutual, 2-46

N
NetlistInclude, 5-23

O
OscPort, 4-4
OscPort2, 4-7
OscTest, 4-14

P
PLC, 2-48
PLCQ, 2-49
Port, 1-8
PRC, 2-51
PRL, 2-52
PRLC, 2-53

R
R, 2-54
R_Conn, 2-60
R_dxdy, 2-61
R_Model, 2-56
R_Pad1, 2-62
R_Space, 2-63
RIND, 3-5

S
S10P to S20P, 5-49
S1P, 5-26
S1P_Eqn to S6P_Eqn, 6-6
S21P to S99P, 5-52
S2P, 5-28
S2P_Conn, 5-31
S2P_Pad3, 5-33
S2P_Spac, 5-38
S2PMDIF, 5-35
S3P, 5-40
S4P, 5-43
S5P to S9P, 5-46
Short, 2-64
SLC, 2-65
SLCQ, 2-66
SMT_Pad, 2-68
SnP component, 5-55
SProbe, 4-17
SProbePair, 4-19
SRC, 2-70
SRL, 2-71
SRLC, 2-72

T
Term, 1-9
TF, 2-73
TF3, 2-74
TimeDelta, 4-21
TimeFrg, 4-23
TimePeriod, 4-24
TimeStamp, 4-25

V
VAR, 5-59

W
WaveformStats, 4-26

X
XFERP, 3-7
XFERRUTH, 3-9
XFERTAP, 3-11

Y
Y1P_Eqn to Y6P_Eqn, 6-9

Z
Z1P_Eqn to Z6P_Eqn, 6-11