Advanced Design System 2002

Circuit Components
Introduction and Simulation Components

February 2002
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 © 2002, Agilent Technologies. All Rights Reserved.
Contents

1 Introduction
   BinModel (Bin Model for Automatic Model Selection) ......................................................... 1-2
   DataAccessComponent (Data Access Component) ............................................................... 1-6
   FORMAT A, B, C, D, E (Drawing Formats) ......................................................................... 1-12
   Ground (Ground Component) ......................................................................................... 1-13
   NetlistInclude (Netlist File Include Component) ............................................................. 1-14
   Port (Port Component) .................................................................................................... 1-17
   Term (Port Impedance for S-parameters) ......................................................................... 1-18
   VAR (Variables and Equations Component) ..................................................................... 1-19
   Multiplicity......................................................................................................................... 1-25
   Components from Product Migration ........................................................................... 1-27
      Component Categories .................................................................................................. 1-27

2 Lumped Components
   C (Capacitor) ................................................................................................................ 2-2
   C_Model (Capacitor Model) .............................................................................................. 2-4
   C_Conn (Capacitor (Connector Artwork)) ........................................................................ 2-5
   C_Pad1 (Capacitor (Pad Artwork)) .................................................................................. 2-6
   C_Space (Capacitor (Space Artwork)) ............................................................................. 2-7
   CAPP2_Conn (Chip Capacitor (Connector Artwork)) ...................................................... 2-8
   CAPP2_Pad1 (Chip Capacitor (Pad Artwork)) ............................................................... 2-10
   CAPP2_Space (Chip Capacitor (Space Artwork)) ............................................................ 2-12
   CAPQ (Capacitor with Q) .............................................................................................. 2-14
   CQ_Conn (Capacitor with Q (Connector Artwork)) ......................................................... 2-16
   CQ_Pad1 (Capacitor with Q (Pad Artwork)) ................................................................. 2-18
   CQ_Space (Capacitor with Q (Space Artwork)) .............................................................. 2-20
   DC_Block (DC Block) .................................................................................................... 2-22
   DC_Feed (DC Feed) ....................................................................................................... 2-23
   DICAP (Dielectric Laboratories Di-cap Capacitor) ......................................................... 2-24
   DILABMLC (Dielectric Laboratories Multi-Layer Chip Capacitor) .................................. 2-25
   INDQ (Inductor with Q) ............................................................................................... 2-27
   InDQ2 ............................................................................................................................ 2-29
   L (Inductor) .................................................................................................................. 2-31
   L_Conn (Inductor (Connector Artwork)) ......................................................................... 2-33
   L_Model (Inductor Model) ............................................................................................. 2-34
   L_Pad1 (Inductor (Pad Artwork)) .................................................................................. 2-35
   L_Space (Inductor (Space Artwork)) .............................................................................. 2-36
   LQ_Conn (Inductor with Q (Connector Artwork)) .......................................................... 2-37
   LQ_Pad1 (Inductor with Q (Pad Artwork)) .................................................................... 2-39
   LQ_Space (Inductor with Q (Space Artwork)) ............................................................... 2-41
<table>
<thead>
<tr>
<th>Component Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutual (Mutual Inductor)</td>
<td>2-43</td>
</tr>
<tr>
<td>PLC (Parallel Inductor-Capacitor)</td>
<td>2-45</td>
</tr>
<tr>
<td>PLCQ (Parallel Inductor-Capacitor with Q)</td>
<td>2-46</td>
</tr>
<tr>
<td>PRC (Parallel Resistor-Capacitor)</td>
<td>2-48</td>
</tr>
<tr>
<td>PRL (Parallel Resistor-Inductor)</td>
<td>2-49</td>
</tr>
<tr>
<td>PRLC (Parallel Resistor-Inductor-Capacitor)</td>
<td>2-50</td>
</tr>
<tr>
<td>R (Resistor)</td>
<td>2-51</td>
</tr>
<tr>
<td>R_Model (Resistor Model)</td>
<td>2-53</td>
</tr>
<tr>
<td>R_Conn (Resistor (Connector Artwork))</td>
<td>2-55</td>
</tr>
<tr>
<td>R_Pad1 (Resistor (Pad Artwork))</td>
<td>2-56</td>
</tr>
<tr>
<td>R_Space (Resistor (Space Artwork))</td>
<td>2-57</td>
</tr>
<tr>
<td>Short (Short)</td>
<td>2-58</td>
</tr>
<tr>
<td>SLC (Series Inductor-Capacitor)</td>
<td>2-59</td>
</tr>
<tr>
<td>SLCQ (Series Inductor-Capacitor with Q)</td>
<td>2-60</td>
</tr>
<tr>
<td>SMT_Pad (SMT Bond Pad)</td>
<td>2-62</td>
</tr>
<tr>
<td>SRC (Series Resistor-Capacitor)</td>
<td>2-64</td>
</tr>
<tr>
<td>SRL (Series Resistor-Inductor)</td>
<td>2-65</td>
</tr>
<tr>
<td>SRLC (Series Resistor-Inductor-Capacitor)</td>
<td>2-66</td>
</tr>
<tr>
<td>TF (Transformer)</td>
<td>2-67</td>
</tr>
<tr>
<td>TF3 (3-Port Transformer)</td>
<td>2-68</td>
</tr>
</tbody>
</table>

3 Miscellaneous Circuit Components

<table>
<thead>
<tr>
<th>Component Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPP2 (Chip Capacitor)</td>
<td>3-2</td>
</tr>
<tr>
<td>CIND (Ideal Torroidal Inductor)</td>
<td>3-4</td>
</tr>
<tr>
<td>RIND (Rectangular Inductor)</td>
<td>3-5</td>
</tr>
<tr>
<td>XFERP (Physical Transformer)</td>
<td>3-7</td>
</tr>
<tr>
<td>XFERRUTH (Ruthroff Transformer)</td>
<td>3-9</td>
</tr>
<tr>
<td>XFERTAP (Tapped Secondary Ideal Transformer)</td>
<td>3-11</td>
</tr>
</tbody>
</table>

4 Probe Components

<table>
<thead>
<tr>
<th>Component Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter (Counter Component)</td>
<td>4-2</td>
</tr>
<tr>
<td>I_Probe (Current Probe)</td>
<td>4-3</td>
</tr>
<tr>
<td>OscPort (Grounded Oscillator Port)</td>
<td>4-4</td>
</tr>
<tr>
<td>OscPort2 (Differential Oscillator Port)</td>
<td>4-7</td>
</tr>
<tr>
<td>OscTest (Grounded Oscillator Test)</td>
<td>4-14</td>
</tr>
<tr>
<td>TimeDelta (Time Delta Component)</td>
<td>4-15</td>
</tr>
<tr>
<td>TimeFrq (Time Frequency Component)</td>
<td>4-17</td>
</tr>
<tr>
<td>TimePeriod (Time Period Component)</td>
<td>4-18</td>
</tr>
<tr>
<td>TimeStamp (Time Stamp Component)</td>
<td>4-19</td>
</tr>
<tr>
<td>WaveformStats (WaveformStats Component)</td>
<td>4-20</td>
</tr>
</tbody>
</table>

5 Linear Data File Components

<table>
<thead>
<tr>
<th>Component Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deembed1 (1-Port De-Embed Data File)</td>
<td>5-2</td>
</tr>
</tbody>
</table>
Deembed2 (2-Port De-Embed File) ................................................................. 5-4
S1P (1-Port S-Parameter File) ................................................................. 5-6
S2P (2-Port S-Parameter File) ................................................................. 5-8
S2P_Conn (2-Port S-Parameter File; connector artwork) ......................... 5-10
S2P_PAD3 (2-Port S-Parameter File; pad artwork) ................................... 5-12
S2PMDIF (Multi-Dimensional 2-Port S-parameter File) ....................... 5-14
S2P_Spac (2-Port S-Parameter File) ........................................................ 5-16
S3P (3-Port S-Parameter File) ................................................................. 5-18
S4P (4-Port S-Parameter File) ................................................................. 5-20
S5P to S9P (5-Port to 9-Port S-Parameter File) ..................................... 5-22
S10P to S20P (10-Port to 20-Port S-Parameter File) ............................ 5-24
S21P to S99P (21-Port to 99-Port S-Parameter File) ............................ 5-26

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) ...... 6-8
  Z1P_Eqn to Z6P_Eqn (1- to 6-Port Z-parameters, Equation-Based) ...... 6-10

Index
Chapter 1: Introduction

The Advanced Design System Circuit Components catalog provides component information. Chapters in this catalog generally reflect library names and components are arranged alphabetically within each chapter.

Chapter 1 provides information for these common items:

• BinModel for automatic model selection
• Ground, Port, and Term
• DataAccessComponent to specify file-based parameters
• Drawing formats (design sheets)
• Netlist File Include Component to locate an externally referenced file
• VAR to define variables and equations
• The multiplicity feature, used in several components and devices to scale components or entire sub-circuits containing many components and sub-circuits.
• List of older versions of components from previous program versions that can be accessed for your convenience.
Introduction

BinModel (Bin Model for Automatic Model Selection)

Symbol

Parameters

N/A

Notes/Equations/References

1. This feature is available for use with the BJT, Diode, GaAs, JFET, and MOS models and is found 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

---

1-2  BinModel (Bin Model for Automatic Model Selection)
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 Bin Model box 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 BJTM1 instance in the schematic, the $B_f$ parameter was set to 100, and in BJTM2 it was set to 50.

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.
Introduction

7. The design was simulated, then the command *Simulate > Annotate DC Box* was selected. In the Schematic window, the value 100μA 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 50.0μA 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>100.0μA</td>
</tr>
<tr>
<td>35.000</td>
<td>50.0μA</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><strong>Model</strong></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>1_Probe1.i</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.00</td>
<td>100.0µA</td>
</tr>
<tr>
<td>35.00</td>
<td>50.0µA</td>
</tr>
<tr>
<td>45.00</td>
<td>25.0µA</td>
</tr>
<tr>
<td>55.00</td>
<td>10.0µA</td>
</tr>
</tbody>
</table>

12. The 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.** To delete a model from the table, select it, then choose this button.
- **Delete Param.** To delete a parameter from the table, select it, then choose this button.
Introduction

**DataAccessComponent (Data Access Component)**

**Symbol**

![DAC Symbol]

**Parameters**

File = filename
Type = file type
Block = block name
InterpMode = interpolation mode:

- **Index Lookup**: Specifies that iVal1, iVal2, etc. (described below) represent the integer indices (beginning with 0) of the independent variables in the data file. Real iVal values are truncated first, for index lookup.

- **Value Lookup**: 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.

- **Ceiling Value Lookup**: For a real independent variable, accesses the nearest point in the data file not less than the specified value.

- **Floor Value Lookup**: For a real independent variable, accesses the nearest point in the data file not greater than the specified value.

- **Linear, Cubic, Cubic Spline**: Specifies the interpolation mode in each dimension, except for splines, where only the innermost variable is spline-interpolated.

- **Value**: 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: {“linear”(0), “spline”(1), “cubic”(2), “index_lookup”(3), “value_lookup”(4), “ceiling_value_lookup”(5), “floor_value_lookup”(6)}. 

---

1-6  DataAccessComponent (Data Access Component)
InterpDom = interpolation domain:

- Rectangular: Interpolates real and imaginary parts separately. Recommended for immitances.
- DB: Interpolates in dB and angle format.

iVar1 = independent variable name or cardinality
iVal1 = independent variable value or index
iVar2 ... iVar10 = independent variable name or cardinality
iVal2 ... iVal10 = independent variable value or index

Notes/Equations/References

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. (See “Example 1” on page 1-9)

   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. (See “Example 2” on page 1-10)

   You can also sweep over several BJT models using two DAC components. (See “Example 3” on page 1-11)

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
Introduction

"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). For information on data file formats, refer to Chapter 20 in the Circuit Simulation manual. The files displayed in the Browser represent all files found based on the search paths specified by the DATA_FILES configuration variable.

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 (for example, Vgs) or is an integer representing the cardinality or nesting order of the independent variable (1 being innermost). 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 the at symbol (@) must be used to suppress quotes when using a variable, for example, @freq1, where freq1 is a variable declared in a VAR item.

Note that when the DataAccessComponent is referring to a time-domain MDIF file which has a .tim extension, the iVarx 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.

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.
7. Each \( iVal \) is normally a real or integer value of the independent variable to bracket or search for in the file. If InterpMode is 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 resistor R1 is stepped through all of the values under the “R” column in the file, “r.mdf”.

The contents of the r.mdf file:

<table>
<thead>
<tr>
<th>INDEX</th>
<th>R</th>
<th>Rsh</th>
<th>Length</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>1.0</td>
<td>2.0</td>
<td>3.0</td>
</tr>
<tr>
<td>2</td>
<td>101</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>51</td>
<td>9</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>123</td>
<td>22</td>
<td>11</td>
</tr>
</tbody>
</table>

END
Example 2

In this example, the resistor model RM1 accesses the parameters R, Rsh, Length and Width from the discrete file \textit{r.mdf}.

\begin{verbatim}
% INDEX  R    Rsh   Length  Width
1        100  1.0   2.0     3.0
2        101    4     5       6
3         51    9     8       7
4         10  123    22      11
END
\end{verbatim}
Example 3

This example shows 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.

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

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

Drawing

Parameters

None

Notes/Equations/References

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

\[ \Downarrow \]

Parameters

None

Notes/Equations/References

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

NetlistInclude (Netlist File Include Component)

Symbol

Parameters

IncludePath = Space-delimited search path for included files
IncludeFiles = List of files to include
UsePreprocessor = “yes” to use an #include directive, or “no” to copy the full text of the file (default is “yes”)

Range of Usage

N/A

Notes/Equations/References

1. The Netlist File Include component is provided as a means for the ADS simulator to utilize an external file. Examples of previous versions of built-in netlist include components are spiceInclude, geminiInclude and idfInclude. These special components 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 only 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 is used to locate 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":

---

1-14   NetlistInclude (Netlist File Include Component)
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:

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

Similarly, with the "SelectR" 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:

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 default value, then the netlister generates a set of preprocessor directives such as:

#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:
**Introduction**

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. Take caution when placing a NetlistInclude component in a subcircuit. If an included file contains a subcircuit definition, then 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.
Port (Port Component)

Symbol

Parameters

Num = port number (value type integer)

layer = (for Layout option) layer to which port is mapped

Range of Usage

Unlimited

Notes/Equations/References

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

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

Range of Usage
N/A

Notes/Equations/References
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 + JX. The reactance is ignored for dc simulations.
VAR (Variables and Equations Component)

Symbol

Parameters

X = name of variable or equation

Notes/Equations/References

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 (see note 3) and functions (see note 4).
Introduction

- **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 DAC data item placed in your currently active design. For more information on the use of DAC data files, refer to “DataAccessComponent (Data Access Component)” on page 1-6.

Name is the name of the file you are referencing, as identified in the DAC item.

Data Access Component Instance is the instance name of the particular DAC item that you are referencing.

Dependent Parameter Name is the name of a DAC 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 listed in Figure 1-1.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>2.718 282 ...</td>
</tr>
<tr>
<td>ln(10)</td>
<td>2.302 585 ...</td>
</tr>
<tr>
<td>c0</td>
<td>2.997 924 58 e+08 m/s</td>
</tr>
<tr>
<td>e0</td>
<td>8.854 188 ... e-12 F/m</td>
</tr>
<tr>
<td>u0</td>
<td>1.256 637 ... e-06 H/m</td>
</tr>
<tr>
<td>boltzmann</td>
<td>1.380 658 e-23 J/K</td>
</tr>
<tr>
<td>qelectron</td>
<td>1.602 177 33 e-19 C</td>
</tr>
<tr>
<td>planck</td>
<td>6.626 075 5 e-34 J*s</td>
</tr>
<tr>
<td>pi</td>
<td>3.141 593 ...</td>
</tr>
</tbody>
</table>

Figure 1-1.
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, $R=\sin\theta$. 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 4, Simulator Expression Reference in the Expressions, Measurements, and Simulation Data Processing manual for a full listing 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, $4 \times (x + .1)$ 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, $z = x + jy$.

- **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:
** Introduction

- ** exponentiation
- ^ exponentiation
- * multiplication
- / 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

\[
C10 * (1 + .005) \text{ evaluates differently than } \\
C10 * 1 + .005
\]

- ** 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,

\[
X = 1 \\
Y = \text{if} \ ( X>0) \ \text{then} \ ( \cos( pi/8)) \ \text{else} \ ( \sin( pi/8)) \ \text{endif}
\]

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.
Multiplicity

The aim of the multiplicity feature is to provide a convenient method to scale components or entire sub-circuits containing many components and sub-circuits. Given a component with some value M of Multiplicity, the simulator treats this component as if there were M such components all connected in parallel. Sub-circuits including within sub-circuits will be appropriately scaled. M does not have to be an integer.

The parameter is available from the component level as an additional parameter (_M) for several components and devices. For example, in the Gummel-Poon BJT, it is added at the instance level, as shown here. Note that the default visibility setting is off for this parameter.
Introduction

For sub-circuits, the parameter must be 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**, as shown here.

Most components already have the parameter. 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.
Components from Product Migration

The following lists of components include those from older products, such as Series IV or MDS that can still be placed in ADS designs. There is no documentation currently available for these components, but they are identified here for your convenience.

None of these are accessible from the component library browser or palettes, but can be placed in the Schematic window by typing the exact name (as shown) into the Component History box above the drawing area, then pressing Enter and moving your cursor to the drawing area.

Component Categories

The following categories of components are included:

- Series IV Components
- Spectral Sources
- Wideband Modems
- Other MDS Components

Series IV Components

- GAIN
- PULSE_TRAIN

Spectral Sources

- GMSK_SOURCE
- PIQPSK_SOURCE
- QAM16_SOURCE
- QPSK_SOURCE

Wideband Modems

- AM_DemodBroad
- AM_ModBroad
- FM_DemodBroad
- FM_ModBroad
Introduction

- IQ_ModBroad
- QAM_ModBroad
- QPSK_ModBroad
- PM_DemodBroad
- PM_ModBroad

Other MDS Components

- CPWTL_MDS
- GCPWTL_MDS
- CPWCTL_MDS
- ACPW_MDS
- ACPWTL_MDS
- CPWTLFG_MDS
- MSACTL_MDS
- MS3CTL_MDS
- MS4CTL_MDS
- MS5CTL_MDS
- MSABND_MDS
- MSBEND_MDS
- MSOBND_MDS
- MSCRNR_MDS
- MSTRL2_MDS
- MSCTL_MDS
- MSCROSS_MDS
- MSRBND_MDS
- MSGAP_MDS
- MSAGAP_MDS
V AR (Variables and Equations Component) 1-29

• 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
• MLACRN1
• MLCRN1
• MLRADIAL1
• MLSLANTED1
• MLCROSSOVER1
• SLTL_MDS
• SLOC_MDS
• SLCTL_MDS
• SL3CTL_MDS
Introduction

- 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
- 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
Introduction
Chapter 2: Lumped Components
Lumped Components

**C (Capacitor)**

**Symbol**

![Symbol](image)

**Parameters**

- $C$ = capacitance, in farads
- Temp = temperature, in °C
- Tnom = nominal temperature, in °C
- TC1 = linear temperature coefficient, in 1/°C
- TC2 = quadratic temperature coefficient, in 1/°C²
- wBV = breakdown voltage warning, in volts
- InitCond = initial condition voltages for transient analysis, in volts; when specified, the check-box “Use user-specified initial conditions” in the Convergence tab of the Simulation-Transient control item must be enabled for this condition to take effect.

- Model = name of a capacitor model to use
- Width = physical width for use with a model
- Length = physical length for use with a model
- _M = number of capacitors in parallel (default: 1)

**Range of Usage**

N/A

**Notes/Equations/References**

1. The capacitor value can be made a function of temperature by setting $T_{nom}$ and either $TC_1$ or $TC_2$ or both. $T_{nom}$ specifies the nominal temperature at which $C$ is given. $T_{nom}$ defaults to 25°C. If Temp≠$T_{nom}$, then the simulated capacitance value is given by:

   $$C' = C \times [1 + TC_1 (Temp - T_{nom}) + TC_2 (Temp - T_{nom})^2]$$

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 computed as:

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

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*Scale*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 Simulation-Transient controller for the parameter setting to take effect.
Lumped Components

**C_Model (Capacitor Model)**

**Symbol**

![Symbol](image)

**Parameters**

- \( C = \) default fixed capacitance, in farads (default: 1 pF)
- \( C_j = \) junction capacitance, in farads/meter\(^2\)
- \( C_{jsw} = \) sidewall or periphery capacitance, in farads/meter
- **Length** = default length, in specified units
- **Width** = default width, in specified units
- **Narrow** = length and width narrowing due to etching, in specified units
- **T_{nom}** = nominal temperature, in °C
- **TC1** = linear temperature coefficient, in 1/°C
- **TC2** = quadratic temperature coefficient, in 1/°C\(^2\)
- **wBV** = breakdown voltage (warning), in volts
- **AllParams** = DataAccessComponent-based parameters
- **Scale** = capacitance scaling factor (default:1)

**Range of Usage**

N/A

**Notes/Equations/References**

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.
C_Conn (Capacitor (Connector Artwork))

Symbol

Parameters

C = capacitance, in farads

Range of Usage

N/A

Notes/Equations/References

1. This component is a single connection in layout. For example, it can be used to represent parasitics.
Lumped Components

C_Pad1 (Capacitor (Pad Artwork))

Symbol

\[ \text{Symbol} \]

Parameters

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

Range of Usage

N/A

Notes/Equations/References

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

\[ \text{Diagram} \]
C_Space (Capacitor (Space Artwork))

Symbol

Parameters

C = capacitance, in farads
L1 = (for Layout option) pin-to-pin distance, in specified units

Range of Usage

N/A

Notes/Equations/References

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

**CAPP2_Conn (Chip Capacitor (Connector Artwork))**

**Symbol**

```
\[ \begin{array}{c}
  \downarrow \\
\end{array} \begin{array}{c}
  \downarrow \\
\end{array} \]
```

**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/References**

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

   \[ Q(f) = Q(F_{Q}) \times \left(\frac{FreqQ}{f}\right)^{Exp} \]

   where $f$ is the simulation frequency and $Q(F_{Q})$ 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.


Equivalent Circuit
Lumped Components

CAPP2_Pad1 (Chip Capacitor (Pad Artwork))

Symbol

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 = (for Layout option) width of pad, in specified units
S = (for Layout option) spacing, in specified units
L1 = (for Layout option) pin-to-pin distance, in specified units

Range of Usage

C, Q, FreqQ, FreqRes ≥ 0

Notes/Equations/References

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

   $Q(f) = Q(FreqQ) \times \left(\frac{FreqQ}{f}\right)^{Exp}$

   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

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
- **L1**: (for Layout option) pin-to-pin distance, in specified units

Range of Usage

- C, Q, FreqQ, FR ≥ 0

Notes/Equations/References

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
   \[ Q(f) = Q(FreqQ) \times \left(\frac{FreqQ}{f}\right)^{Exp} \]
   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 is represented as a connected gap in layout—into which a custom artwork object can be inserted.


Equivalent Circuit
Lumped Components

**CAPQ (Capacitor with Q)**

**Symbol**

![Symbol](image)

**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 see notes):
  - 1 is proportional to freq
  - 2 is proportional to sqrt(freq)
  - 3 is constant

**Range of Usage**

F ≥ 0

**Notes/Equations/References**

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 \frac{\sqrt{f/F}}{F} )</td>
<td>( G(f) = G(F) \times \frac{\sqrt{f/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**

![Equivalent Circuit Diagram]
Lumped Components

**CQ_Conn (Capacitor with Q (Connector Artwork))**

**Symbol**

![Symbol](image)

**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 see notes):
  1. proportional to freq
  2. proportional to sqrt(freq)
  3. constant

**Range of Usage**

F ≥ 0

**Notes/Equations/References**

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) \times \frac{f}{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.

---

2-16  CQ_Conn (Capacitor with Q (Connector Artwork))
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

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 see notes):

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

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

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

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

Range of Usage

F ≥ 0

Notes/Equations/References

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

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{1}{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</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

- $f = \text{simulation frequency}$
- $F = \text{reference frequency}$
- $G = \text{conductance of capacitor}$

**Equivalent Circuit**
Lumped Components

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

**Symbol**

![CQ_Space Symbol](image)

**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 see notes):
  1. proportional to freq
  2. proportional to $\sqrt{freq}$
  3. constant
- $L_1 =$ (for Layout option) pin-to-pin distance, in specified units

**Range of Usage**

$F \geq 0$

**Notes/Equations/References**

1. $Q = \frac{B}{G} = \frac{2\pi \cdot f \cdot C}{G}$

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

---

2-20  CQ_Space (Capacitor with Q (Space Artwork))
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**
Lumped Components

**DC Block (DC Block)**

**Symbol**

```
1
- - -
2
```

**Parameters**

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

**Range of Usage**

N/A

**Notes/Equations/References**

1. The C and L parameters are used for transient simulation only because open for DC_Block is non-causal for transient simulation.
**DC_Feed (DC Feed)**

**Symbol**

```
1
\|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-- |--``` 

**Parameters**

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

**Range of Usage**

N/A

**Notes/Equations/References**

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:

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

$Er =$ dielectric constant

$\tan\delta L =$ dielectric loss tangent value at 1 MHz

$RO =$ series resistance at 1 GHz, in ohms

Range of Usage

N/A

Notes/Equations/References

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 $cond2$; the bottom metal on layer $cond$; and, the capacitor dielectric on layer $diel$. 

2-24 DICAP (Dielectric Laboratories Di-cap Capacitor)
DILABMLC (Dielectric Laboratories Multi-Layer Chip Capacitor)

Symbol

Illustration:

Parameters
- $CO =$ nominal capacitance, in farads
- $\text{TanDel} =$ 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

Range of Usage
N/A

Notes/Equations/References

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.
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 wave-length.

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

Parameters
L = inductance, in henries
Q = quality factor
F = frequency at which Q is given, in hertz
Mode = loss mode for this device; options (also see notes):
 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 ≥ 0

Notes/Equations/References

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

where:

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>( Q(f) = Q(F) \times \frac{f}{F} )</td>
<td>( R(f) = R(F) )</td>
</tr>
<tr>
<td>proportional to sqrt (freq)</td>
<td>( Q(f) = Q(F) \times \sqrt{\frac{f}{F}} )</td>
<td>( R(f) = R(F) \times \sqrt{\frac{f}{F}} )</td>
</tr>
<tr>
<td>constant</td>
<td>( Q(f) = Q(F) )</td>
<td>( R(f) = R(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
- \( R \) = resistance of inductor
Lumped Components

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

**Equivalent Circuit**

![Equivalent Circuit Diagram](image)
InDQ2
Symbol

Parameters
L = inductance, in specified unit
Q = quality factor
F = frequency at which Q is given, in specific unit
Mode = loss mode of the device.

Options of mode are
Mode=1: Q being proportional to freq
Mode=2: Q being proportional to sqrt(freq)
Mode=3: Q being constant
Mode=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
Since the impedance of the device is Z = R + j*2*pi*freq*L, the Quality factor is given in this respect as:

\[ Q = \frac{\text{imag}(Z)}{\text{real}(Z)} = \frac{2\pi \text{freq} \cdot L}{R} \]

<table>
<thead>
<tr>
<th>Mode Setting</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportional to freq</td>
<td>( Q(f) = Q(F) \cdot f/F )</td>
<td>( R(f) = R(F) )</td>
</tr>
<tr>
<td>proportional to sqrt (freq)</td>
<td>( Q(f) = Q(F) \cdot \sqrt{f/F} )</td>
<td>( R(f) = R(F) \cdot \sqrt{f/F} )</td>
</tr>
<tr>
<td>constant</td>
<td>( Q(f) = Q(F) )</td>
<td>( R(f) = R(F) / f/F )</td>
</tr>
<tr>
<td>proportional to sqrt (freq), constant L</td>
<td>( Q(f) = Q(F) \cdot \sqrt{f/F} )</td>
<td>( R(f) = R(F) \cdot \sqrt{f/F} ) and L is constant</td>
</tr>
</tbody>
</table>
Lumped Components

where $f =$ simulation frequency

$F =$ reference frequency

$R =$ resistance of the inductor

Notes/Equations/References

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

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

Equivalent Circuit

-----VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV

$Z = R_{dc} + R \cdot j \cdot 2 \cdot \pi \cdot \text{freq} \cdot L$
**L (Inductor)**

**Symbol**

\[ \text{\[ \begin{array}{c}
\text{1} \\
\hline
\text{2}
\end{array} \] } \]

**Parameters**

- **L** = inductance, in henries
- **R** = series resistance, in ohms
- **Temp** = temperature, in °C
- **Tnom** = nominal temperature, in °C
- **TC1** = linear temperature coefficient, in 1/°C
- **TC2** = quadratic temperature coefficient, in 1/°C^2

**InitCond** = transient analysis initial condition current, in amperes; when specified, the check-box “Use user-specified initial conditions” in the Convergence tab of the Simulation-Transient control item must be enabled for this condition to take effect.

- **_M** = number of inductors in parallel (default: 1)

**Range of Usage**

N/A

**Notes/Equations/References**

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:

   \[ \langle i^2 \rangle = 4kT/R \]

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.
Lumped Components

5. \( M \) 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 \( R \times \text{Scale}/M \).

6. When InitCond is explicitly specified, the check-box *Use user-specified initial conditions* must be turned on in the Convergence tab of the Simulation-Transient controller for the parameter setting to take effect.
**L.Conn (Inductor (Connector Artwork))**

**Symbol**

![Symbol Image]

**Parameters**

$L = $ inductance, in henries

**Range of Usage**

N/A

**Notes/Equations/References**

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

Parameters

L = default inductance, in henries (default: 0)
R = default series resistance, in Ohms (default: 0)
Tnom = nominal temperature, in °C
TC1 = linear temperature coefficient, in 1/°C
TC2 = quadratic temperature coefficient, in 1/°C^2
Scale = capacitance scaling factor (default: 1)
All Params = Data Access Component (DAC) based parameters

Range of Usage

N/A

Notes/Equations/References

1. This model supplies values for an inductor L. This allows some common inductor values to be specified in a model.
**L_Pad1 (Inductor (Pad Artwork))**

**Symbol**

![Symbol Image]

**Parameters**

- **L** = inductance, in henries
- **W** = (for Layout option) width of pad, in specified units
- **S** = (for Layout option) spacing, in specified unit
- **L1** = nominal temperature, in °C

**Range of Usage**

N/A

**Notes/Equations/References**

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

![Diagram Image]
Lumped Components

**L_Space (Inductor (Space Artwork))**

**Symbol**

![Symbol](image)

**Parameters**

- \( L \) = inductance, in henries
- \( L_1 \) = nominal temperature, in °C

**Range of Usage**

N/A

**Notes/Equations/References**

1. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.
LQ_Conn (Inductor with Q (Connector Artwork))

Symbol

Parameters
L = inductance, in henries
Q = quality factor
F = reference frequency for Q

Mode = frequency dependence mode of Q; options (also see 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/References

1. $Q = \frac{2\pi f L}{R}$

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
Lumped Components

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

\[
\begin{array}{c}
\text{R} \\
\text{L} \\
\end{array}
\]
LQ_Pad1 (Inductor with Q (Pad Artwork))

Symbol

Parameters

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

Range of Usage

\(F \geq 0\)

Notes/Equations/References

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{\text{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>
Lumped Components

<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

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:

![Equivalent Circuit](image)

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

Symbol

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

Range of Usage
F ≥ 0

Notes/Equations/References

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
Lumped Components

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

\[
\begin{align*}
\text{Equivalent Circuit} & \\
\end{align*}
\]
**Mutual (Mutual Inductor)**

**Symbol**

![Mutual Inductor Symbol](image)

**Illustration:**

![Illustration of Mutual Inductor](image)

**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/References**

1. Either $K$ or $M$ is to be specified, not both. If both are specified, $M$ overrides $K$.

2. In the fields Inductor1 and Inductor2 enter the component names of any two inductors whose mutual inductance is given as $M$. For example, the entries Inductor1= CMP4 and Inductor2 = CMP16 result in simulations that use the value $M$ as mutual inductance between the inductors that appear on the schematic as CMP4 and CMP16. Use several mutual inductor components to define other mutual inductances.

3. The ends of the inductors that are in phase are identified by a small open circle on the schematic symbol for the inductors.
Lumped Components

4. The mutual inductor component can be placed anywhere on the schematic, and there is no limit to the number of mutual inductances that can be specified. It has no effect on auto-layout.
PLC (Parallel Inductor-Capacitor)

Symbol

![Parallel Inductor-Capacitor Symbol]

Parameters

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

Range of Usage

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

Notes/Equations/References

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

Equivalent Circuit

![Equivalent Circuit Diagram]
Lumped Components

**PLCQ (Parallel Inductor-Capacitor with Q)**

**Symbol**

![PLCQ Symbol]

**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 see 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 see notes): proportional to freq; proportional to sqrt(freq); constant; (value type: enumerated)
- Rdc = resistance for modes 2 and 3

**Range of Usage**
Use for high Q circuits, rather than individual components in parallel.

**Notes/Equations/References**

1. \[ Ql = \frac{2\pi f_s L}{R} \] (for inductors)
2. \[ Qc = \frac{2\pi f_s C}{G} \] (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>$Q_l(F_l) \times f_s/F_l$</td>
<td>proportional to freq</td>
<td>$Q_c(F_c) \times f_s/F_c$</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>
<td>proportional to sqrt (freq)</td>
</tr>
<tr>
<td>constant</td>
<td>$Q_l(F_l)$</td>
<td>constant</td>
<td>$Q_c(F_c)$</td>
</tr>
</tbody>
</table>

where

$R =$ resistance of inductor
$G =$ conductance of capacitor
$f_s =$ simulation frequency
$F_c, F_l =$ specified Q frequencies

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

**PRC (Parallel Resistor-Capacitor)**

**Symbol**

![Symbol Image]

**Parameters**

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

**Range of Usage**

N/A

**Notes/Equations/References**

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

**Equivalent Circuit**

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

Symbol

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

Range of Usage
N/A

Notes/Equations/References
1. This component has no default artwork associated with it.

Equivalent Circuit
Lumped Components

**PRLC (Parallel Resistor-Inductor-Capacitor)**

**Symbol**

![PRLC Symbol]

**Parameters**

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

**Range of Usage**

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

**Notes/Equations/References**

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

**Equivalent Circuit**

![Equivalent Circuit]
**R (Resistor)**

**Symbol**

![Resistor Symbol]

**Parameters**

- **R** = resistance, in ohms (default: 50 ohms)
- **Temp** = temperature, in °C
- **Tnom** = nominal temperature, in °C
- **TC1** = linear temperature coefficient, in 1/°C
- **TC2** = quadratic temperature coefficient, in 1/°C²
- **Noise** = resistor thermal noise option: YES=enable; NO=disable
- **wPmax** = maximum power dissipation (warning), in watts
- **wImax** = maximum current (warning), in amperes
- **Model** = name of a resistor model to use
- **Width** = physical width for use with a model
- **Length** = physical length for use with a model
- **_M** = number of resistors in parallel (default: 1)

**Range of Usage**

N/A

**Notes/Equations/References**

1. The resistor 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 R is given. Tnom defaults to 25°C. If Temp≠Tnom, then the simulated resistance value is given by:

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

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

3. The resistor generates thermal noise:

   \[ \langle i^2 \rangle = 4kT/R \]
Lumped Components

Noise generation can be disabled by setting Noise=NO.

4. \( w_{P\text{max}} \) and \( w_{I\text{max}} \) 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 computed as:
   
   \[
   R' = R + \text{Rsh} \times (\text{Length} - 2 \times \text{Narrow}) / (\text{Width} - 2 \times \text{Narrow})
   \]

7. \( _M \) is used to represent the number of resistors in parallel and defaults to 1. \( M \) 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 \( R \times \text{Scale} / M \).
**R_Model (Resistor Model)**

**Symbol**

```
[Resistor symbol]
```

**Parameters**

- **R** = default fixed resistance, in ohms (default: 50 ohms)
- **Rsh** = sheet resistance, in ohms/square
- **Length** = default length, in specified units
- **Width** = default width, in specified units
- **Narrow** = length and width narrowing due to etching, in specified units
- **Tnom** = nominal temperature, in °C
- **TC1** = linear temperature coefficient, in 1/°C
- **TC2** = quadratic temperature coefficient, in 1/°C²
- **wPmax** = maximum power dissipation (warning), in watts
- **wImax** = maximum current (warning), in amperes/meter
- **AllParams** = DataAccessComponent-based parameters
- **scale** = resistance scaling factor (default: 1)

**Range of Usage**

N/A

**Notes/Equations/References**

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, wImax is the current limit in amperes/meter:

   \[ w\text{Imax}' = w\text{Imax} \times (\text{Width}-2 \times \text{Narrow}) \]

   If the physical parameters Rsh, Width and Length are not specified, wImax is the current limit in amperes.
Lumped Components

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

**Symbol**

\[ \text{Symbol} \]

**Parameters**

R = resistance, in ohms

**Range of Usage**

N/A

**Notes/Equations/References**

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 a single connection in layout. For example, it can be used to represent parasitics.
Lumped Components

**R_Pad1 (Resistor (Pad Artwork))**

**Symbol**

![Symbol Diagram]

**Parameters**

- **R** = resistance, in ohms
- **W** = (for Layout option) width of pad, in specified units
- **S** = (for Layout option) spacing, in specified units
- **L1** = (for Layout option) pin-to-pin distance, in specified units

**Range of Usage**

N/A

**Notes/Equations/References**

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:

![Artwork Diagram]
R_Space (Resistor (Space Artwork))

Symbol

Parameters
R = resistance, in ohms
L1 = (for Layout option) pin-to-pin distance, in specified units

Range of Usage
N/A

Notes/Equations/References

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 _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**

![](image)

**Parameters**

- **Mode** = mode: short, dc block, dc feed (value type: integer)
- **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)

**Range of Usage**

N/A

**Notes/Equations/References**

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.
SLC (Series Inductor-Capacitor)

Symbol

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

Range of Usage
Use when modeling high Q circuits as opposed to two individual components.

Notes/Equations/References
1. This component has no default artwork associated with it.
Lumped Components

SLCQ (Series Inductor-Capacitor with Q)

Symbol

Parameters
L = inductance, in henries
Ql = quality factor of inductor
Fl = frequency at which Q is defined, in hertz
ModL = frequency dependence mode of inductor Q; options (also see 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 see notes):
   1 is proportional to freq
   2 is proportional to sqrt(freq)
   3 is constant
Rdc = resistance for modes 2 and 3

Range of Usage
Use when modeling high Q circuits rather than individual components in series.

Notes/Equations/References

1. \[ Ql = \frac{2\pi f_s L}{R} \] (for inductors)
2. \[ Qc = \frac{2\pi f_s C}{G} \] (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>$Q_l(F_l) \times \frac{f_s}{F_l}$</td>
<td>proportional to freq</td>
<td>$Q_c(F_c) \times \frac{f_s}{F_c}$</td>
</tr>
<tr>
<td>proportional to sqrt(freq)</td>
<td>$Q_l(F_l) \times \sqrt{\frac{f_s}{F_l}}$</td>
<td>proportional to sqrt(freq)</td>
<td>$Q_c(F_c) \times \sqrt{\frac{f_s}{F_c}}$</td>
</tr>
<tr>
<td>constant</td>
<td>$Q_l(F_l)$</td>
<td>constant</td>
<td>$Q_c(F_c)$</td>
</tr>
</tbody>
</table>

where

- \( R \) = resistance of inductor
- \( G \) = conductance of capacitor
- \( f_s \) = simulation frequency
- \( F_c, F_l \) = specified Q frequencies

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

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

**Equivalent Circuit**

![Equivalent Circuit Diagram]
Lumped Components

SMT_Pad (SMT Bond Pad)

Symbol

Illustration

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/References

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 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.

2-62 SMT_Pad (SMT Bond Pad)
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**

\[
\begin{array}{c}
\text{R} \\
\text{C}
\end{array}
\]

**Parameters**

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

**Range of Usage**

N/A

**Notes/Equations/References**

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

Symbol

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

Range of Usage
N/A

Notes/Equations/References
1. This component has no default artwork associated with it.
Lumped Components

**SRLC (Series Resistor-Inductor-Capacitor)**

**Symbol**

\[ \begin{array}{c}
\text{R}\text{C}\text{L} \\
\rightarrow
\end{array} \]

**Parameters**

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

**Range of Usage**

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

**Notes/Equations/References**

1. This component has no default artwork associated with it.
TF (Transformer)

Symbol

![Transformer Symbol]

Parameters

T = turns ratio T1/T2

Range of Usage

N/A

Notes/Equations/References

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. Parasitic inductances of the primary and secondary are not modeled. To do this, use the component for mutual inductance (M). The ends that are in phase are identified by a small open circle on the schematic symbol.

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

4. This component passes DC.
Lumped Components

TF3 (3-Port Transformer)

Symbol

Parameters
T1 = turn 1
T2 = turn 2

Range of Usage
N/A

Notes/Equations/References
1. The turns ratio $T$ is the ratio of turns in the secondary to turns in the primary:

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

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
CAPP2 (Chip Capacitor)

Symbol

Parameters

C = capacitance, in farads
TanD = dielectric loss tangent
Q = quality factor
Freq = reference frequency for Q, in hertz
FreqRes = resonance frequency, in hertz
Exp = exponent for frequency dependence of Q

Range of Usage

C, Q, FreqQ, FreqRes ≥ 0

Notes/Equation/References

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 has no default artwork associated with it.


CIND (Ideal Torroidal Inductor)

Symbol

Illustration

Parameters

N = number of units

AL = inductance index

Range of Usage

N, AL > 0

Notes/Equation/References

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

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

Symbol

Illustration

Parameters

N = number of turns (need not be an integer)
L1 = length of second outermost segment, in specified units
L2 = 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)
FR = resonant frequency, in hertz
Temp = physical temperature, in °C

Range of Usage

N must be such that all segments fit given L1, L2, W, and S.

Notes/Equation/References

1. For time-domain analysis, an impulse response obtained from the frequency-domain analytical model is used.
Miscellaneous Circuit Components

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

**XFERP (Physical Transformer)**

**Symbol**

![Symbol Image]

**Parameters**

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

**Range of Usage**

$0 < K < 1$

**Notes/Equation/References**

1. **Primary leakage:**
   
   $$L_1 = L_p \left( \frac{1}{K} - 1 \right)$$

2. **Secondary leakage:**
   
   $$L_2 = \frac{L_1}{N^2}$$

2. This component has no default artwork associated with it.
Miscellaneous Circuit Components

**Equivalent Circuit**

![Equivalent Circuit Diagram]

---

3-8  XFERP (Physical Transformer)
**XFERRUTH (Ruthroff Transformer)**

**Symbol**

![Diagram of XFERRUTH Transformer](image)

**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/Equation/References**

1. Inductance: \( L = N^2 \times AL \)
2. This component has no default artwork associated with it.
5. Ruthroff, C. L. “Some Broadband Transformers,” *Proceedings of the IRE*, Vol. 47, No. 8, August 1959, pp. 1337-1342 (Fig. 3, page 1338, in particular).
Equivalent Circuit
**XFERTAP (Tapped Secondary Ideal Transformer)**

**Symbol**

![Symbol](image)

**Parameters**

- $N_{12} = \text{turns ratio, } N_1/N_2$
- $N_{13} = \text{turns ratio, } N_1/N_3$
- $L_1 = \text{primary winding inductance, in henries}$
- $K = \text{coupling coefficient}$

**Range of Usage**

$0 < K < 1$

**Notes/Equation/References**

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

**Counter (Counter Component)**

**Symbol**

![Counter Symbol](image)

**Parameters**

- **Direction** = direction one
- **Thresh** = threshold one, in volts

**Range of Usage**

N/A

**Notes/Equation/References**

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

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)

Range of Usage

N/A

Notes/Equation/References

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. This component has no default artwork associated with it.
OscPort (Grounded Oscillator Port)

Symbol

Parameters

V = initial guess at fundamental voltage
Z = initial value for $Z_0$, in ohms
NumOctaves = number of octaves to search
Steps = number of steps per search octave
FundIndex = fundamental number for oscillator
MaxLoopGainStep = maximum arc length continuation step size during loop-gain search

Range of Usage

N/A

Notes/Equation/References

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 $\text{Freq}/2$ to $\text{Freq} \times 2$. 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 no fundamental voltage V is 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. If it is not known, don’t specify it. An inaccurate value increases the simulation time and might prevent convergence.

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...
Probe Components

 impedance, some gain controlling value, or another factor. In short, anything that will make the oscillator more linear, yet still let it oscillate.
**OscPort2 (Differential Oscillator Port)**

**Symbol**

![Symbol Image]

**Parameters**

- **Mode** = oscillator mode: automatic, 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)
- **NumOctaves** = number of octaves to search (automatic mode only)
- **Steps** = number of steps per search octave (automatic mode only)
- **FundIndex** = fundamental number for oscillator (automatic mode only)
- **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)

**Range of Usage**

N/A

**Notes/Equation/References**

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 \( Freq \times 2 \). **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 no fundamental voltage **V** is 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.
 Probe Components

If it is not known, don’t specify it. An inaccurate value increases the simulation time and might prevent convergence.

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.

The equivalent circuit for OscPort2 in automatic mode is shown next.

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...
Probe Components

mode. The analysis computes 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 1 + 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 1 + 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 here.

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 computes 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 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 here.

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

\[ S_{OTHER} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \]
Probe Components

**OscTest (Grounded Oscillator Test)**

**Symbol**

![Symbol Image]

**Parameters**

- **Port Number** = port number
- **Z** = initial value for $Z_0$, in ohms
- **Start** = start frequency, in hertz
- **Stop** = stop frequency, in hertz
- **Points** = number of frequency points

**Range of Usage**

N/A

**Notes/Equation/References**

1. 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 $1 + j0$ 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 $1 + j0$ 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.
**TimeDelta (Time Delta Component)**

**Symbol**

![TimeDelta Symbol]

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

**Range of Usage**
N/A

**Notes/Equation/References**

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,
Probe Components

various fall times, pulse widths, etc. The output voltage can be used for other behavioral models, for optimization, or for output to presentations.
**TimeFq (Time Frequency Component)**

**Symbol**

```
1  \[\text{trig}\]  2
```

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

**Range of Usage**
N/A

**Notes/Equation/References**
None
**TimePeriod (Time Period Component)**

**Symbol**

![TimePeriod Symbol]

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

**Range of Usage**
N/A

**Notes/Equation/References**

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 a reasonable value which might otherwise be less than the simulator’s absolute convergence criteria.
**TimeStamp (Time Stamp Component)**

**Symbol**

\[ 
\begin{array}{c}
\text{1} \\
\text{2}
\end{array}
\]

**Parameters**

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

**Range of Usage**

N/A

**Notes/Equation/References**

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

Parameters
None

Range of Usage
N/A

Notes/Equation/References
1. This behavioral model can be used to measure the statistics of the baseband component of the input voltage. The three inputs all have infinite input impedance. All six outputs are ideal voltage sources with zero output impedance.

2. If the enable is low during a reset, then the accumulators are reset to 0. If the enable is high, then N is set to one, and Sum is set equal to the input.

3. 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.

4. 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.

5. This model operates in transient and envelope time domain analysis modes.
Chapter 5: Linear Data File Components
Linear Data File Components

Deembed1 (1-Port De-Embed Data File)

Symbol

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

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

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

Temp = physical temperature, in °C

Range of Usage

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

Notes/Equations/References

1. For details on data file format, refer to the Circuit Simulation manual.
2. One of the Deembed1 data file applications is to negate the 1-port subcircuit by using this data file.
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: transfer to (mag, angle) before interpolation.
   DB: transfer to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
4. If the S-parameters belong to a passive DUT, Temp is used to compute noise performance. If the S-parameters belong to an active DUT, Temp is ignored and the component is assumed to be noiseless.
5. 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.

6. This component has no default artwork associated with it.
Deembed2 (2-Port De-Embed File)

Symbol

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, CITIfile

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

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

Temp = physical temperature, in °C

Range of Usage

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

Notes/Equations/References

1. For details on data file format, refer to the Circuit Simulation manual.

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

3. 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.

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: transfer to (mag, angle) before interpolation.
   - DB: transfer to (dB, angle) before interpolation
   - Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
5. If the file contains noise parameters, the noise performance of this component is calculated using the noise parameters and Temp is ignored. If the noise parameters are not specified and the S-parameters belong to a passive component, noise performance is calculated based on S-parameters and Temp. If noise parameters are not specified and the S-parameters belong to an active component, Temp is ignored and the component is assumed to be noiseless.

6. 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.

7. This component has no default artwork associated with it.
### S1P (1-Port S-Parameter File)

**Symbol**

![Symbol](image)

**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, CITIfile
- **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
- **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

**Range of Usage**

N/A

**Notes/Equations/References**

1. For details on data file format, refer to the *Circuit Simulation* manual.
2. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:

---

**5-6  S1P (1-Port S-Parameter File)**
Rectangular: transform to (real, imag) before interpolation
Polar: transfer to (mag, angle) before interpolation.
DB: transfer to (dB, angle) before interpolation

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

3. Regarding the use of 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. If the S-parameters belong to a passive DUT, Temp is used to compute noise performance. If the S-parameters belong to an active DUT, Temp is ignored and the component is assumed to be noiseless.

5. 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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.

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

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

**S2P (2-Port S-Parameter File)**

**Symbol**

![Symbol Diagram]

**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
- **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
- **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

**Range of Usage**

N/A

**Notes/Equations/References**

1. For details on data file formats, refer to the *Circuit Simulation* 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 as the common terminal.
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**: transfer to (mag, angle) before interpolation.
- **DB**: transfer to (dB, angle) before interpolation
- **Data Based**: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

4. Regarding the use of 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 file contains noise parameters, the noise performance of this component is calculated using the noise parameters and Temp is ignored. If the noise parameters are not specified and the S-parameters belong to a passive component, noise performance is calculated based on S-parameters and Temp. If noise parameters are not specified and the S-parameters belong to an active component, Temp is ignored and the component is assumed to be noiseless.

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.
S2P_Conn (2-Port S-Parameter File; connector artwork)

Symbol

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/References

1. For details on data file formats, refer to the Circuit Simulation 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 as the common terminal.

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

   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transfer to (mag, angle) before interpolation.
   - DB: transfer to (dB, angle) before interpolation.

Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
4. Regarding the use of 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 file contains noise parameters, the noise performance of this component is calculated using the noise parameters and Temp is ignored. If the noise parameters are not specified and the S-parameters belong to a passive component, noise performance is calculated based on S-parameters and Temp. If noise parameters are not specified and the S-parameters belong to an active component, Temp is ignored and the component is assumed to be noiseless.

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.
Linear Data File Components

S2P_PAD3 (2-Port S-Parameter File; pad artwork)

Symbol

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 = (for Layout Option) width of line at pins 1 and 2, in length units

W2 = (for Layout Option) width of line at pin 3, in length units

S = (for Layout Option) spacing (length from pin 1 to pin 2, in length units

L1 = (for Layout Option) length from pin 1 to pin 2, in length units

L2 = (for 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/References

1. For details on data file formats, refer to the Circuit Simulation 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 as the common terminal.

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

---

5-12  S2P_PAD3 (2-Port S-Parameter File; pad artwork)
Rectangular: transform to (real, imag) before interpolation
Polar: transfer to (mag, angle) before interpolation
DB: transfer to (dB, angle) before interpolation

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

4. Regarding the use of 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 file contains noise parameters, the noise performance of this component is calculated using the noise parameters and Temp is ignored. If the noise parameters are not specified and the S-parameters belong to a passive component, noise performance is calculated based on S-parameters and Temp. If noise parameters are not specified and the S-parameters belong to an active component, Temp is ignored and the component is assumed to be noiseless.

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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.

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

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

**S2PMDF (Multi-Dimensional 2-Port S-parameter File)**

**Symbol**

![Symbol Image]

**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: S2PMDF, Touchstone, Dataset, CITIfile

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

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

**Range of Usage**

N/A

**Notes/Equations/References**

1. For details on data file format, refer to the *Circuit Simulation* manual.

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

---

5-14 S2PMDF (Multi-Dimensional 2-Port S-parameter File)
Rectangular: transform to (real, imag) before interpolation
Polar: transfer to (mag, angle) before interpolation.
DB: transfer to (dB, angle) before interpolation.

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

3. Regarding the use of 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. Noise parameters (optimum source reflection coefficient, minimum noise figure in dB and effective source resistance) can be present in the file. If the file contains noise parameters, the noise performance of this component is calculated using the noise parameters and Temp is ignored. If the noise parameters are not specified and the S-parameters belong to a passive component, noise performance is calculated based on S-parameters and Temp. If noise parameters are not specified and the S-parameters belong to an active component, Temp is ignored and the component is assumed to be noiseless.

Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous). Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning). If ImpMode, If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.

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

6. 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.

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

S2P_Spac (2-Port S-Parameter File)

Symbol

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/References

1. For details on data file formats, refer to the Circuit Simulation 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 as the common terminal.

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

4. Regarding the use of 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 file contains noise parameters, the noise performance of this component is calculated using the noise parameters and Temp is ignored. If the noise parameters are not specified and the S-parameters belong to a passive component, noise performance is calculated based on S-parameters and Temp. If noise parameters are not specified and the S-parameters belong to an active component, Temp is ignored and the component is assumed to be noiseless.

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.
S3P (3-Port S-Parameter File)

Symbol

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

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

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

Range of Usage

N/A

Notes/Equations/References

1. For details on data file formats, refer to the `Circuit Simulation` manual.
2. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:

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

3. Regarding the use of 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. If the S-parameters belong to a passive DUT, Temp is used to compute the noise performance of the S3P component; if the S-parameters belong to an active DUT, Temp is ignored and the component is assumed to be noiseless.

5. 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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.

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

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

S4P (4-Port S-Parameter File)

Symbol

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, CITfile

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

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

Range of Usage

N/A

5-20  S4P (4-Port S-Parameter File)
Notes/Equations/References

1. For details on data file formats, refer to the Circuit Simulation manual.

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

3. Regarding the use of 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. 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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.

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

6. 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**

![Diagram of S5P to S9P components](image)  

**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
- **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
- **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

**Range of Usage**

N/A

**Notes/Equations/References**
1. The number of terminals increases sequentially from 5 to 9, and is equal to the number of ports of the component.

2. For details on data file formats, refer to the Circuit Simulation manual.

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

5. Regarding the use of 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. 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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.

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

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

S10P to S20P (10-Port to 20-Port S-Parameter File)

Symbol

![Symbol Diagram]

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

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

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
Range of Usage

N/A

Notes/Equations/References

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.

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

6. Regarding the use of 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. 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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.

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

9. 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**

![Symbol Diagram]

**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
- **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
- **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

**Range of Usage**

N/A

**Notes/Equations/References**

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. To support up to 99-port networks, the port number of a Port component can go up to 99.

3. The SMAT, YMAT, ZMAT, and NMAT measurements are allowed for up to 99-port networks. However, single measurements such as SIJ, GD, and VSWR are only applicable for up to 20-port networks.

4. These components primarily support electromagnetic simulation results of circuits with a large number of ports, such as antenna feed networks.

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

6. Regarding the use of 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.

   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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.

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

8. This component has no default artwork associated with it.
Chapter 6: Equation-Based Linear Components
**Chain (2-Port User-Defined Linear Chain)**

**Symbol**

![Chain Symbol](image)

**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)
- **ImpMax Freq** = 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**

N/A

**Notes/Equations/References**

1. Port polarity is indicated by a minus sign (−) and a plus sign (+) on each port. Chain parameters are useful when cascading a number of networks. Any chain parameter that is not defined initially is set to a default value of zero and cannot be modified later.

2. 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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.
Equation-Based Linear Components

Hybrid (2-Port User-Defined Linear Hybrid)

Symbol

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

Range of Usage

N/A

Notes/Equations/References

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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.
S1P_Eqn to S6P_Eqn (1- to 6-Port S-parameters, Equation-Based)

Symbol

Parameters

- \( S[i, j] \) = S-parameter in real and imaginary form
- \( Z[i] \) = port i reference impedance, in ohms
- \( \text{Recip} \) = port is reciprocal: NO, YES
- \( \text{NFmin} \) = minimum noise figure, in dB
- \( R_n \) = noise resistance, in ohms
- \( \text{Sopt} \) = optimum noise match
- \( \text{Temp} \) = device noise temperature, in °C
- \( \text{ImpNoncausalLength} \) = non-causal function impulse response order (value type: integer)
- \( \text{ImpMode} \) = convolution mode (value type: integer)
- \( \text{ImpMaxFreq} \) = 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
Range of Usage

$1 \leq i, j \leq \text{port number}$

$S_{ij}$ can be made dependent on frequency by using the global variable $freq$. For example, you can use a brick wall lowpass filter by using

$S_{21} = \text{if}(freq < 1 \text{ GHz}), \text{then} 1 \text{ else } 0.$

Notes/Equations/References

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

   If no value is entered for $S_{[i,j]}$, it is set to a default value of zero (0, 0) and cannot be modified later. To enter a value for $S_{[i,j]}$, use the syntax $a+j$ or $\text{complex}(a,b)$. 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.

2. If $NF_{\text{min}}$, $S_{\text{opt}}$, and $R_n$ are used to characterize noise, the following relation must be satisfied for a realistic model.

   $$\frac{R_n}{Z_0} \geq \frac{T_{\text{min}}(1+\frac{S_{\text{opt}}}{S_{11}})}{T^4 \left[1 - \frac{S_{11}^2}{S_{\text{opt}}^2}\right]^2}$$

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

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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.
Equation-Based Linear Components

**Y1P_Eqn to Y6P_Eqn (1- to 6-Port Y-parameters, Equation-Based)**

**Symbol**

![Port Diagram]

**Parameters**

- \( Y_{i,j} \) = Y-parameter magnitude and phase
- 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**

\[ 1 \leq i, j \leq \text{port number} \]

**Notes/Equations/References**

1. Port polarity is indicated by a minus sign (–) and a plus sign (+) on each port.
   
   If no value is entered for \( Y_{i,j} \), it is set to a default value of zero (0S, 0S) and cannot be modified later. To enter a value, use the syntax \( a+j \times b \) or complex(a,b).

---

6-8 Y1P_Eqn to Y6P_Eqn (1- to 6-Port Y-parameters, Equation-Based)
If the port parameter 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. No state currents are generated or available.

2. 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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.
Equation-Based Linear Components

**Z1P_Eqn to Z6P_Eqn (1- to 6-Port Z-parameters, Equation-Based)**

**Symbol**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z[i,j]</td>
<td>Z-parameter magnitude and phase</td>
</tr>
<tr>
<td>C[i]</td>
<td>port 1 controlling current (see Notes/Equations/References)</td>
</tr>
<tr>
<td>Recip</td>
<td>port is reciprocal: NO, YES</td>
</tr>
<tr>
<td>ImpNoncausalLength</td>
<td>non-causal function impulse response order (value type: integer)</td>
</tr>
<tr>
<td>ImpMode</td>
<td>convolution mode (value type: integer)</td>
</tr>
<tr>
<td>ImpMaxFreq</td>
<td>maximum frequency to which device is evaluated, in hertz</td>
</tr>
<tr>
<td>ImpDeltaFreq</td>
<td>sample spacing in frequency, in hertz</td>
</tr>
<tr>
<td>ImpMaxOrder</td>
<td>maximum impulse response order (value type: integer)</td>
</tr>
<tr>
<td>ImpWindow</td>
<td>smoothing window (value type: integer)</td>
</tr>
<tr>
<td>ImpRelTol</td>
<td>relative impulse response truncation factor</td>
</tr>
<tr>
<td>ImpAbsTol</td>
<td>absolute impulse response truncation factor</td>
</tr>
</tbody>
</table>

**Range of Usage**

1 ≤ i, j ≤ port number

**Notes/Equations/References**

1. Port polarity is indicated by a minus sign (−) and a plus sign (+) on each port. The port can be made reciprocal by setting REC=1 (yes). By declaring the device to be reciprocal, Z[i,j] is always forced to be equal to Z[j,i]. Only one of the two can be defined.

---

6-10 Z1P_Eqn to Z6P_Eqn (1- to 6-Port Z-parameters, Equation-Based)
If no value is entered for $Z[i,j]$, it is set to a default value of zero and cannot be modified later. To enter a value for $Z[i,j]$, use the syntax $a+j \times b$ or complex($a,b$).

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 Z-port.

2. 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. Refer to Chapter 33, Simulation - Transient for default values on these parameters and a more detailed description of their use.

3. 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.
Equation-Based Linear Components

6-12  Z1P_Eqn to Z6P_Eqn (1- to 6-Port Z-parameters, Equation-Based)
Index

B
Bin model, 1-2

D
Data Access Component, 1-6
data file components, linear, 5-1

E
equation-based linear components, 6-1

L
lumped components, 2-1

M
M_ parameter, 1-25
multiplicity feature, 1-25

P
probe components, 4-1

V
Var, 1-19
variables and equations, 1-19