Agilent Technologies

# Introduction to Circuit Components

**May 2003**

# Notice

## 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.

## Acknowledgments

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

Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation.

Pentium® is a U.S. registered trademark of Intel Corporation.

PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated.

UNIX® is a registered trademark of the Open Group.

# Contents

## 1 Introduction

## 2 Lumped Components

**3 Miscellaneous Circuit Components**

**4 Probe Components**

**5 Linear Data File Components**

**6 Equation-Based Linear Components**

**Index**

# Chapter 1: Introduction

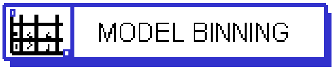
The Advanced Design System Circuit Components catalog provides component information. Chapters in this book are organized by library; components are arranged alphabetically within each chapter.

Chapter 1 provides information for these common items:

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

# BinModel (Bin Model for Automatic Model Selection)

## Symbol



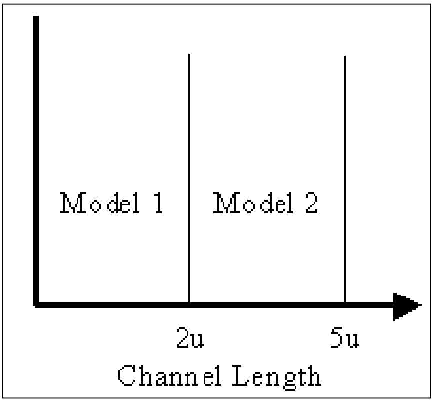
## Parameters

None

## Notes/Equations

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

2. BinModel allows you to sweep a parameter (usually a geometry, such as gate length), then enable the simulator to automatically select between different model cards. If a circuit contains nonlinear devices for circuit simulation, each device should be associated with one device model through schematic or netlist editing. However, modern processes require multiple models for different device sizes to improve simulation accuracy. For example, as illustrated here, a model (Model 2) that is accurate for a 4u channel length MOSFET is not necessarily a good model (Model 1) for a 1u channel length. If mixed analog and digital circuits are combined in a single part, multiple models are the easiest way to create high accuracy over a wide range of device sizes.

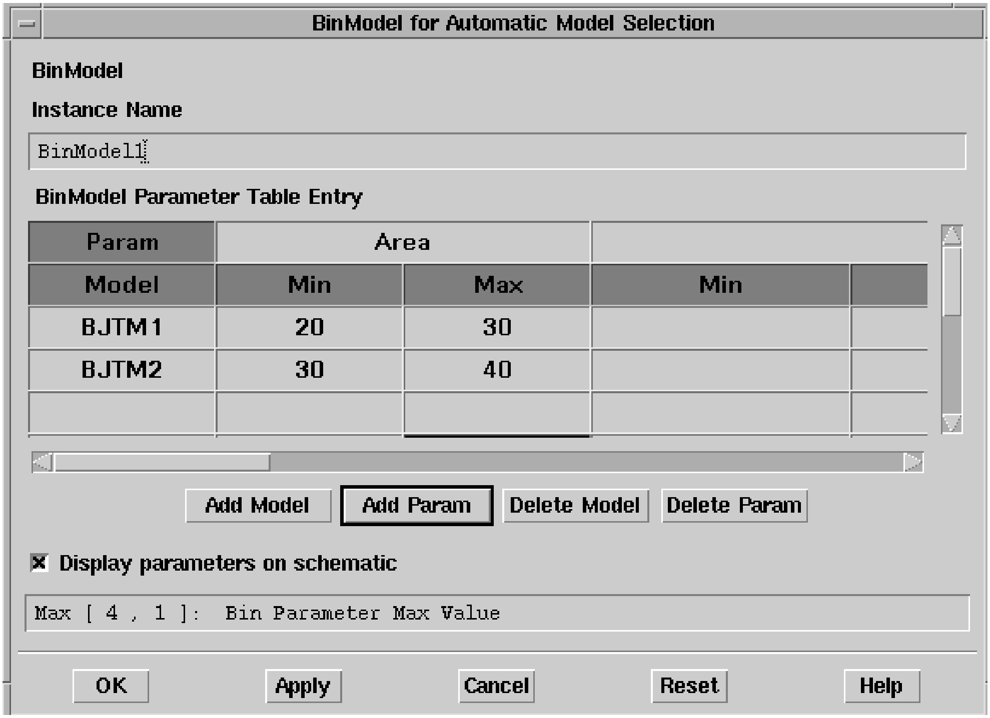


3. Depending on device size, one of the multiple models should be selected for a device at simulation time. If device size needs to be varied over a certain range, manual model change for each new device size would be very cumbersome. The

model binning feature automatically searches for a model with the size range that covers the device size and uses this model in simulation.

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

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



Instance Name: BinModel1

| Param | Area | | | |
|---|---|---|---|---|
| Model | Min | Max | Min | |
| BJTM1 | 20 | 30 | | |
| BJTM2 | 30 | 40 | | |
| | | | | |

☒ Display parameters on schematic

Max [ 4 , 1 ]:  Bin Parameter Max Value

5. In the corresponding BJTM1 instance in the schematic, the *Bf* parameter was set to 100, and in BJTM2 it was set to 50.

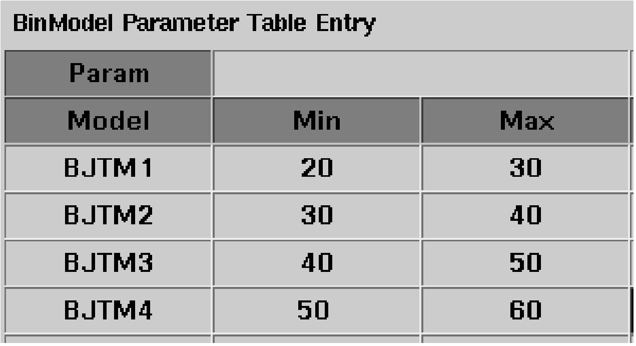
6. In the device model placed in the schematic (for example, BJT_NPN), the first bin model to be used for simulation was identified (*Model = BinModel1*) and the AREA parameter was set to 25.

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

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

| **Bin** | **I-Probe1.i** |
|---|---|
| 25.000 | 100.0uA |
| 35.000 | 50.0uA |

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.



BinModel Parameter Table Entry

| Param | | |
|---|---|---|
| Model | Min | Max |
| BJTM1 | 20 | 30 |
| BJTM2 | 30 | 40 |
| BJTM3 | 40 | 50 |
| BJTM4 | 50 | 60 |

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.

| Bin | I_Probe1.i |
|---|---|
| 25.000 | 100.0uA |
| 35.000 | 50.0uA |
| 45.00 | 25.0uA |
| 55.00 | 10.0uA |

12. Buttons beneath the table function as follows:

**Add Model** adds additional rows to the Model column for specification of more models

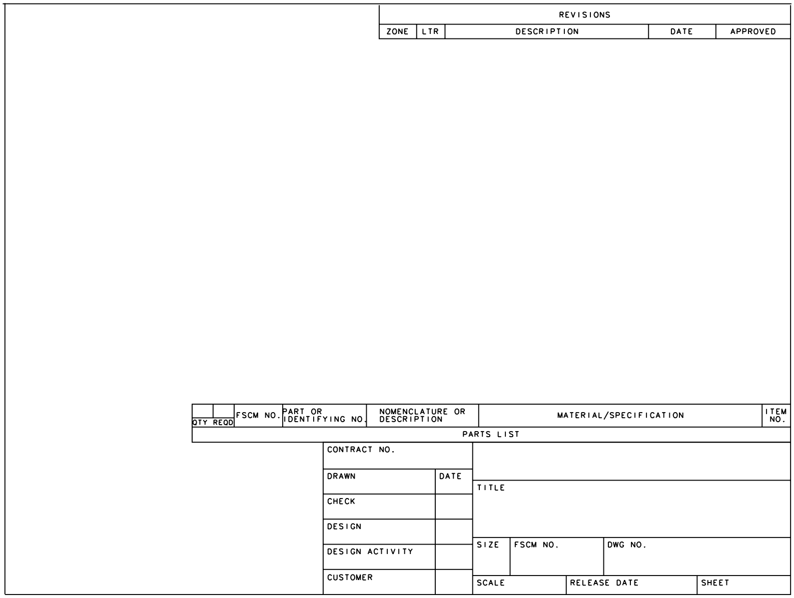
**Add Param** adds additional entry boxes to the Param field for specification of more parameters

**Delete Model** deletes a selected model

**Delete Param** deletes a selected parameter

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

## Drawing



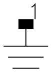
FORMATA

## Notes/Equations

1. The Drawing Formats library provides popular sheet sizes (in inches): A (8.5×11), B (11×17), C (17×22), D (22×34), E (34×44).
2. Turn on Drawing Format filter through *Options* > *Preferences* > *Select*. You can then move or delete the drawing sheet. Turn off the filter when not needed.

# Ground (Ground Component)

## Symbol



## Parameters

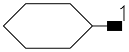
None

## Notes/Equations

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

# Port (Port Component)

## Symbol



## Parameters

Num = port number (value type: integer)

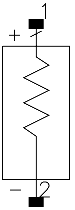
layer = (ADS Layout option) layer to which port is mapped

## Notes/Equations

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

# Term (Port Impedance for S-parameters)

## Symbol



## Parameters

Num = port number (value type: integer)

Z = reference impedance, in ohms

Noise = enable/disable port thermal noise: YES, NO (for AC or harmonic balance analysis only; not for S-parameter analysis)

Vdc = open circuit dc voltage, in volts

## Notes/Equations

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

# VAR (Variables and Equations Component)

## Symbol

## Parameters

X = name of variable or equation

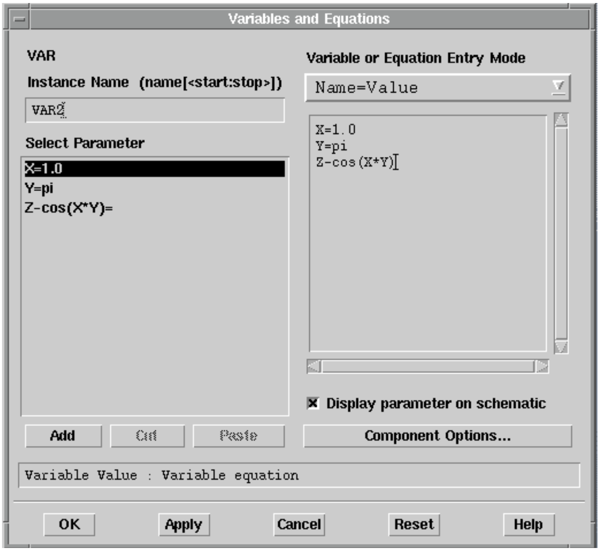
## Notes/Equations

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

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

2. **Variable or Equation Entry Mode**

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



Instance Name (name[<start:stop>]): VAR2

Variable or Equation Entry Mode: Name=Value

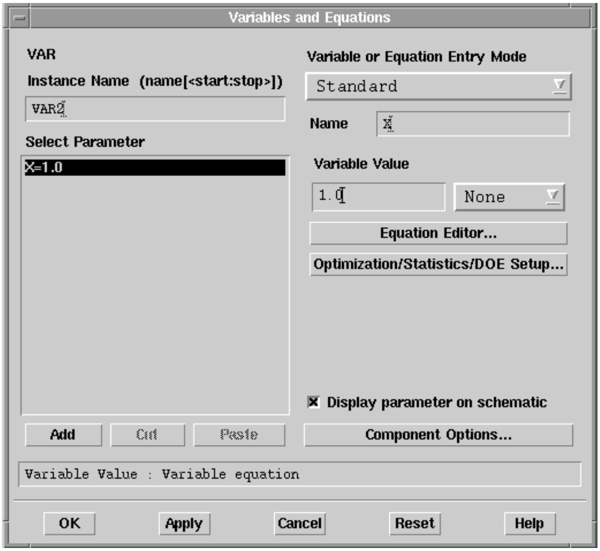
```
X=1.0
Y=pi
Z-cos(X*Y)
```

☒ Display parameter on schematic

Variable Value : Variable equation

Note that
expression X has a numeric value;
expression Y uses a predefined constant;
expression Z uses a predefined function.

- **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.



Variables and Equations

VAR

Instance Name (name[<start:stop>]) VAR2

Select Parameter

X=1.0

Variable or Equation Entry Mode Standard

Name X

Variable Value 1.0 None

☒ Display parameter on schematic

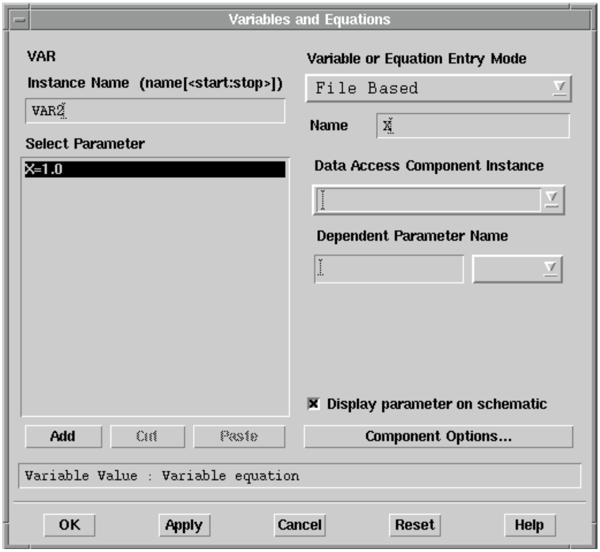
Variable Value : Variable equation

- **File Based.** To use variable or equation data from a file, reference a DataAccessComponent placed in your currently active design. For more information on the use of DAC data files, refer to DataAccessComponent.

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

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

  **Dependent Parameter Name** is the name of a DataAccessComponent parameter for which you want to include data.



Instance Name (name[<start:stop>])

VAR2

X=1.0

Variable or Equation Entry Mode

File Based

Name X

Data Access Component Instance

Dependent Parameter Name

☒ Display parameter on schematic

Variable Value : Variable equation

### 3. Pre-defined Built-in Constants

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

| | | |
|---|---|---|
| `e` | `= 2.718 282 ...` | `e` |
| `ln10` | `= 2.302 585 ...` | `ln(10)` |
| `c0` | `= 2.997 924 58 e+08 m/s` | `speed of light` |
| `e0` | `= 8.854 188 ... e-12 F/m` | `vacuum permittivity (1/(u0*c0*c0)` |
| `u0` | `= 1.256 637 ... e-06 H/m` | `vacuum permeability (4*pi*1e-7)` |
| `boltzmann` | `= 1.380 658 e-23 J/K` | `Boltzmann's constant` |
| `qelectron` | `= 1.602 177 33 e-19 C` | `charge of an electron` |
| `planck` | `= 6.626 075 5 e-34 J*s` | `Planck's constant` |
| `pi` | `= 3.141 593 ...` | `pi` |

### 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 resister, R=sin5. 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, r0, r1, rx, ry, lower_bound, upper_bound` are real
- `s, s1, s2` 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 * (x + .1) evaluates the same as
  4*(x+.1)
- **Numerical constants** Real numbers such as 12.68, exponential notation numbers such as 1e6 or 25.1e3, pi can be used, and complex numbers can be defined. For example, z = x + j*y.
- **LHS form assignment** The LHS assignment takes the form of integer, double, complex or string dependent on what form is associated with the RHS. For example,
  X=4, Y=4.0, W=1.0+j*3.0, Z="4"
  associates the form of integer, double, complex and string to X, Y, W and Z, respectively. The LHS form is important when subsequently used in following expressions.

- **Mathematical operators** Standard operators are available:

  ** exponentiation
  ^exponentiation
  * multiplication
  / division
  + addition
  – subtraction

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

  C10 * (1 + .005) evaluates differently than

  C10 *1 + .005

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

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

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

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

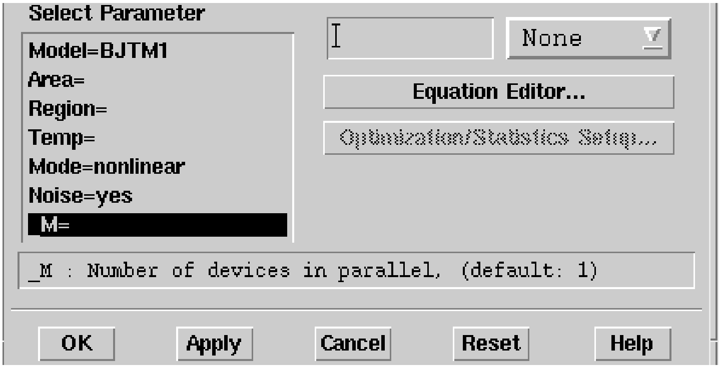
  < > <= >= = != && ||

  Each expression can be any valid numeric expression. The entire if...then...else...endif expression must be on one line.

# Multiplicity Parameter _M

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

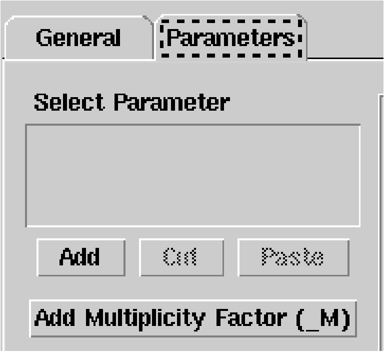
The _M parameter is available at the component level as shown here. (For components that don't explicitly have a Multiplicity parameter, the same functionality can be achieved by placing the component in a sub-circuit and using the sub-circuit's Multiplicity parameter, as described next.)



Model=BJTM1
Area=
Region=
Temp=
Mode=nonlinear
Noise=yes
_M=

```
_M : Number of devices in parallel, (default: 1)
```

For sub-circuits, the parameter is enabled by selecting **File** > **Design Parameters** from the Schematic window. In the dialog box, select the **Parameters** tab. To add the Multiplicity parameter, choose **Add Multiplicity Factor_M**.



# Series IV or MDS Product Migration Components

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

- **Series IV Components**
  - GAIN
  - PULSE_TRAIN
- **Spectral Sources**
  - GMSK_SOURCE
  - PIQPSK_SOURCE
  - QAM16_SOURCE
  - QPSK_SOURCE
- **Wideband Modems**
  - AM_DemodBroad
  - AM_ModBroad
  - FM_DemodBroad
  - FM_ModBroad
  - IQ_ModBroad
  - QAM_ModBroad
  - QPSK_ModBroad
  - PM_DemodBroad
  - PM_ModBroad
- **MDS Components**
  - CPWTL_MDS
  - GCPWTL_MDS
  - CPWCTL_MDS

- 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
- MSIDCF_MDS
- MSIDC_MDS
- MSLANGE_MDS
- MSTL_MDS
- MSOC_MDS
- MSSPLC_MDS
- MSSPLS_MDS
- MSSPLR_MDS
- MSSTEP_MDS
- MSRTL_MDS
- MSSLIT_MDS
- MSTAPER_MDS

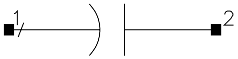
- MSTEE_MDS
- TFC_MDS
- MSWRAP_MDS
- TFR_MDS
- MSVIA_MDS
- MSSVIA_MDS
- MLACRNR1
- MLCRNR1
- MLRADIAL1
- MLSLANTED1
- MLCROSSOVER1
- SLTL_MDS
- SLOC_MDS
- SLCTL_MDS
- SL3CTL_MDS, SL4CTL_MDS, SL5CTL_MDS
- SLUCTL_MDS
- SLGAP_MDS
- SLSTEP_MDS
- SLTEE_MDS
- SLOBND_MDS
- SLCNR_MDS
- SLRBND_MDS
- SLABND_MDS
- SLUTL_MDS
- SSTL_MDS
- SSCTL_MDS
- SS3CTL_MDS, SS4CTL_MDS, SS5CTL_MDS

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

# Chapter 2: Lumped Components

# C (Capacitor)

## Symbol



## Parameters

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

## Notes/Equations

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

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

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

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

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

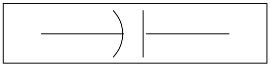
   $$C' = C - Cj \times (Length - 2 \times Narrow) \times -(Width - 2 \times Narrow)$$
   $$+ Cjsw \times 2 \times (Length + Width - 4 \times Narrow)$$

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 Tran transient simulation controller for the parameter setting to take effect.

# C_Model (Capacitor Model)

## Symbol



## Parameters

| Name | Description | Unit | Default |
|---|---|---|---|
| C | default fixed capacitance | fF, pF, nF, uF, mF | 1.0 pF |
| Cj | junction capacitance | farads/meter$^2$ | |
| Cjsw | sidewall or periphery capacitance | farads/meter | |
| Length (L) | default length | um, mm, cm, meter, mil, in | |
| Width (W) | default width | um, mm, cm, meter, mil, in | |
| Narrow (Etch) | length and width narrowing due to etching | um, mm, cm, meter, mil, in | |
| Tnom | nominal temperature | °C | |
| Trise | temperature rise over ambient | °C | 0 |
| TC1 | linear temperature coefficient | 1/°C | |
| TC2 | quadratic temperature coefficient | 1/°C$^2$ | |
| wBV | breakdown voltage (warning) | fV, pV, nV, uV, mV, V | |
| Scale (Scalec) | capacitance scaling factor | | 1 |
| Coeffs | nonlinear capacitor polynomial coefficients | | |
| AllParams | DataAccessComponent-based parameters | | |

## Netlist Format

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

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

The model statement starts with the required keyword *model*. It is followed by the *modelname* that will be used by capacitor components to refer to the model. The third parameter indicates the type of model; for this model it is *C_Model*. The rest of the model contains pairs of model parameters and values, separated by an equal sign. The name of the model parameter must appear exactly as shown in the parameters table-these names are case sensitive. Some model parameters have aliases, which are listed in parentheses after the main parameter name; these are parameter names that can be used instead of the primary parameter name. Model parameters may appear in any order in the model statement. Model parameters that are not specified take the default value indicated in the parameters table. For more information about the ADS circuit simulator netlist format, refer to Chapter 8, *ADS Simulator Input Syntax* in the *Circuit Simulation* book.

Example:

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

## Notes/Equations

---

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

---

1. This model supplies values for a capacitor C. This allows physically-based capacitors to be modeled based on length and width.
2. Use AllParams with a DataAccessComponent to specify file-based parameters (refer to DataAccessComponent). Note that model parameters that are explicitly specified take precedence over those specified via AllParams.
3. The capacitor value can be made a nonlinear function of the applied voltage V by specifying the polynomial coefficients list (Coeffs = list(c1, c2, c3, ...) ). The capacitance value $C(V)$ is then given by:

   $$C(V) = \frac{dQ(V)}{dV} = C(1 + c1 \times V + c2 \times V^2 + \ldots)$$

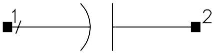
   where $C$ is the capacitance of the instance, and $ck$ is the $k$-th entry in the Coeffs list.

   The charge as a function of the applied voltage is:

   $$Q(V) = C \times V \times \left(1 + \left(\frac{1}{2}\right) \times c1 \times V + \left(\frac{1}{3}\right) \times c2 \times V^2 + \ldots\right)$$

# C_Conn (Capacitor (Connector Artwork))

## Symbol



## Parameters

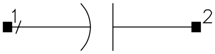
C = capacitance, in farads

## Notes/Equations

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

# C_dxdy (Capacitor (Delta X - Delta Y))

## Symbol



## Parameters

C = capacitance, in farads

dx = delta X, in specified units

dy = delta Y, in specified units

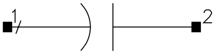
Temp = temperature, in °C

## Notes/Equations

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

# C_Pad1 (Capacitor (Pad Artwork))

## Symbol



## Parameters

C = capacitance, in farads

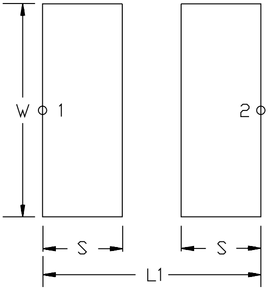
W = width of pad, in specified units

S = spacing, in specified units

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

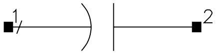
## Notes/Equations

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



# C_Space (Capacitor (Space Artwork))

## Symbol



## Parameters

C = capacitance, in farads

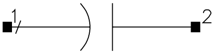
L1 = (ADS Layout option) pin-to-pin distance, in specified units

## Notes/Equations

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

# CAPP2_Conn (Chip Capacitor (Connector Artwork))

## Symbol



## Parameters

C = capacitance, in farads

TanD = dielectric loss tangent

Q = quality factor

FreQ = resistance frequency for Q, in hertz

FreqRes = resistance frequency, in hertz

Exp = exponent for frequency dependence of Q

## Range of Usage

C, Q, FreqQ, FreqRes $\geq 0$

## Notes/Equations

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

   The frequency-dependence of Q is given by

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

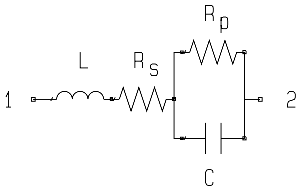
   where f is the simulation frequency and Q(FreqQ) is the specified value of Q at FreqQ.
2. If Q or FreqQ are set to 0, Q is assumed to be infinite.
3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component has no default artwork associated with it.

## References

[1] C. Bowick, *RF Circuit Design*, Howard Sams & Co., 1987.

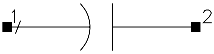
[2] *The RF Capacitor Handbook*, American Technical Ceramics Corp., September 1983.

## Equivalent Circuit



## CAPP2_Pad1 (Chip Capacitor (Pad Artwork))

### Symbol



### Parameters

C = capacitance, in farads

TanD = dielectric loss tangent

Q = quality factor

FreQ = resistance frequency for Q, in hertz

FreqRes = resistance frequency, in hertz

Exp = exponent for frequency dependence of Q

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

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

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

### Range of Usage

C, Q, FreqQ, FreqRes ≥ 0

### Notes/Equations

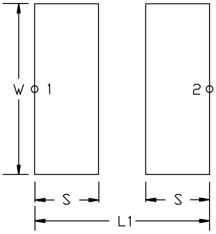
1. The series resistance $R_s$ is determined by the Q and the parallel resistance $R_p$ is determined by TanD.

   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.

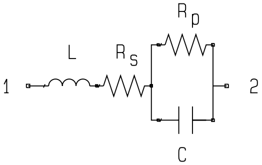
4. This component's artwork is composed of two rectangular pads with pins on the outer edges.



5. Bowick, Cris. *RF Circuit Design*, Howard Sams & Co., 1987.

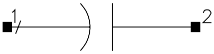
6. *The RF Capacitor Handbook*, American Technical Ceramics Corp., September 1983.

## Equivalent Circuit



# CAPP2_Space (Chip Capacitor (Space Artwork))

## Symbol



## Parameters

C = capacitance, in farads

TanD = dielectric loss tangent

Q = quality factor

FreQ = resistance frequency for Q, in hertz

FreqRes = resistance frequency, in hertz

Exp = exponent for frequency dependence of Q

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

## Range of Usage

C, Q, FreqQ, FR ≥ 0

## Notes/Equations

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

   The frequency-dependence of Q is given by

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

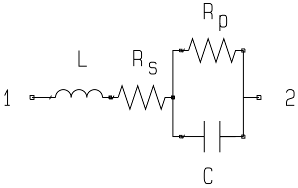
   where f is the simulation frequency and Q(FreqQ) is the specified value of Q at FreqQ.
2. If Q or FreqQ are set to 0, Q is assumed to be infinite.
3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.

## References

[1] C. Bowick, *RF Circuit Design*, Howard Sams & Co., 1987.

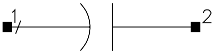
[2] *The RF Capacitor Handbook*, American Technical Ceramics Corp., September 1983.

## Equivalent Circuit



# CAPQ (Capacitor with Q)

## 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 refer to notes):

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

## Range of Usage

$F \geq 0$

## Notes/Equations

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

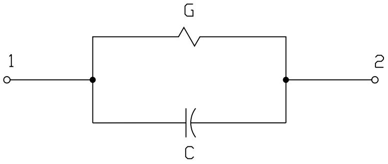
where:

| Mode Setting | Q | G |
|---|---|---|
| proportional to freq | $Q(f) = Q(F) \times f/F$ | G(f) = G(F) |
| proportional to sqrt (freq) | $Q(f) = Q(F) \times \sqrt{f/F}$ | $G(f) = G(F) \times \sqrt{f/F}$ |
| constant | Q(f) = Q(F) | $G(f) = G(F) \times f/F$ |
| If F is set to zero, then Q is assumed to be infinite.<br>where<br>f = simulation frequency<br>F = reference frequency<br>G = conductance of capacitor | | |

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

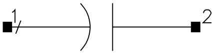
3. This component has no default artwork associated with it.

## Equivalent Circuit



# CQ_Conn (Capacitor with Q (Connector 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 refer to notes):

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

## Range of Usage

$F \geq 0$

## Notes/Equations

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

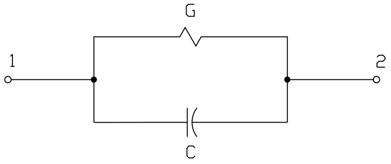
where:

| Mode Setting | Q | G |
|---|---|---|
| proportional to freq | $Q(f) = Q(F) \times f/F$ | G(f) = G(F) |
| proportional to sqrt (freq) | $Q(f) = Q(F) \times \sqrt{f/F}$ | $G(f) = G(F) \times \sqrt{f/F}$ |
| constant | Q(f) = Q(F) | $G(f) = G(F) \times f/F$ |
| If F is set to zero, then Q is assumed to be infinite.<br>where<br>f = simulation frequency<br>F = reference frequency<br>G = conductance of capacitor | | |

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

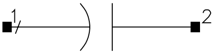
3. This component is a single connection in layout. For example, it can be used to represent parasitics.

## Equivalent Circuit



# 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 refer to notes):

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

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

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

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

## Range of Usage

$F \geq 0$

## Notes/Equations

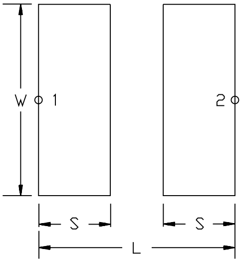
1. $Q = \frac{B}{G} = \frac{2\pi f C}{G}$

where:

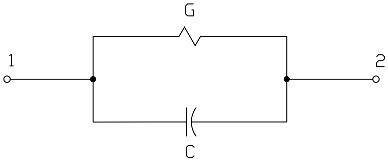
| Mode Setting | Q | G |
|---|---|---|
| proportional to freq | $Q(f) = Q(F) \times f/F$ | G(f) = G(F) |
| proportional to sqrt (freq) | $Q(f) = Q(F) \times \sqrt{f/F}$ | $G(f) = G(F) \times \sqrt{f/F}$ |

| **Mode Setting** | **Q** | **G** |
|---|---|---|
| constant | Q(f) = Q(F) | $G(f) = G(F) \times f / F$ |
| If F is set to zero, then Q is assumed to be infinite.<br>where<br>f = simulation frequency<br>F = reference frequency<br>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:

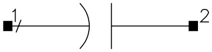


## Equivalent Circuit



# CQ_Space (Capacitor with Q (Space 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 refer to notes):

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

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

## Range of Usage

$F \geq 0$

## Notes/Equations

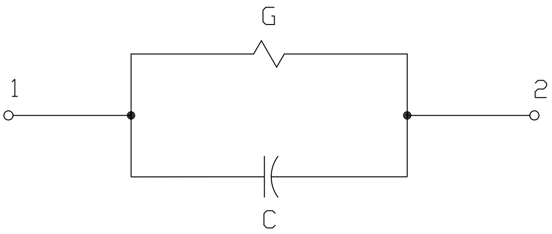
1. $Q = \frac{B}{G} = \frac{2\pi f C}{G}$

where:

| **Mode Setting** | **Q** | **G** |
|---|---|---|
| proportional to freq | $Q(f) = Q(F) \times f/F$ | $G(f) = G(F)$ |
| proportional to sqrt (freq) | $Q(f) = Q(F) \times \sqrt{f/F}$ | $G(f) = G(F) \times \sqrt{f/F}$ |
| constant | $Q(f) = Q(F)$ | $G(f) = G(F) \times f/F$ |
| If F is set to zero, then Q is assumed to be infinite.<br>where<br>f = simulation frequency<br>F = reference frequency<br>G = conductance of capacitor | | |

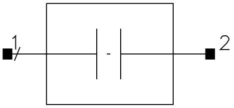
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.

## Equivalent Circuit



# DC_Block (DC Block)

## Symbol



## 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

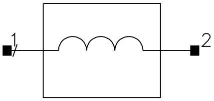
wImax = maximum current (warning), in amperes (value type: real)

## Notes/Equations

1. The C and L parameters are used for transient simulation only because open for DC_Block is non-causal for transient simulation.

# DC_Feed (DC Feed)

## Symbol



## 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

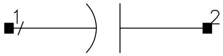
wImax = maximum current (warning), in amperes (value type: real)

## Notes/Equations

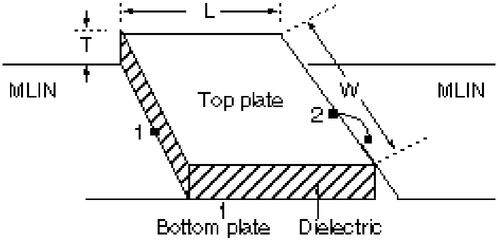
1. The C and L parameters are used for transient simulation only because short for DC_Feed is non-causal for transient simulation.

## 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

TanDeL = dielectric loss tangent value at 1 MHz

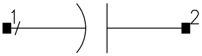
RO = series resistance at 1 GHz, in ohms

### Notes/Equations

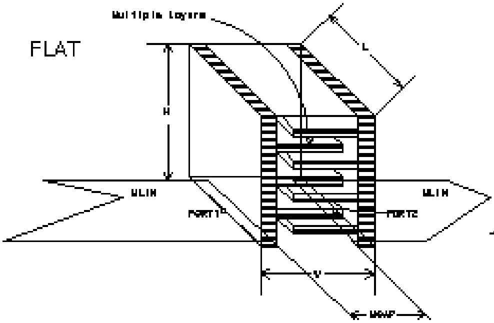
1. This is the Di-cap capacitor model by Dielectric Laboratories Incorporated; for the parameter values, please contact Dielectric Laboratories.
2. DICAP is a single-layer capacitor that behaves as lossy parallel plate transmission lines. Pin 1 is on the bottom metal plate; pin 2 is on the top metal plate. The connection (such as Wire or Ribbon) from the top metal plate (pin 2) to the connecting transmission line is not included in the model—the user must connect it separately.
3. For time-domain analysis, the frequency-domain analytical model is used.
4. In the layout, the top metal will be drawn on layer *cond2*; the bottom metal on layer *cond*; and, the capacitor dielectric on layer *diel*.

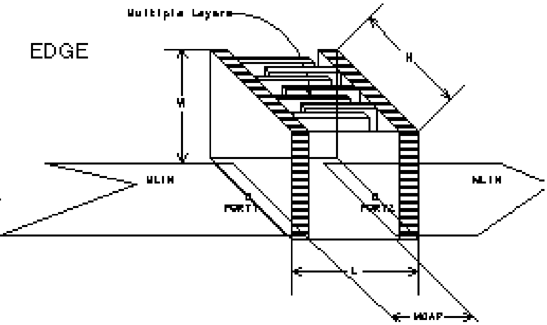
# DILABMLC (Dielectric Laboratories Multi-Layer Chip Capacitor)

## Symbol



## Illustration





## Parameters

CO = nominal capacitance, in farads

TanDel = dielectric loss tangent value at 1 MHz

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

Rt = termination loss resistance at 1 MHz

Re = electrode loss resistance per electrode at 1 GHz

Mount = mounting orientation: *flat* or *edge*

## Notes/Equations

1. This is the multi-layer chip capacitor model by Dielectric Laboratories Incorporated; for the parameter values, please contact Dielectric Laboratories.
2. DILABMLC behaves as an open-ended transmission line. Pins 1 and 2 are at the edges of the capacitor's solder leads. The connections (such as Wire or Ribbon) from the solder leads to the connecting transmission line are not included in the model—the user must connect them separately.
3. For transient analysis, the DILABMLC is modelled as a series RLC equivalent circuit.
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 it 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:

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

Rdc = resistance for modes 2 and 3

## Range of Usage

$F \geq 0$

## Notes/Equations

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

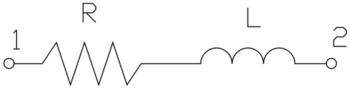
where:

<table>
<tr><th>Mode Setting</th><th>Q</th><th>R</th></tr>
<tr><td>proportional to freq</td><td>$Q(f) = Q(F) \times f/F$</td><td>$R(f) = R(F)$</td></tr>
<tr><td>proportional to sqrt (freq)</td><td>$Q(f) = Q(F) \times \sqrt{f/F}$</td><td>$R(f) = R(F) \times \sqrt{f/F}$</td></tr>
<tr><td>constant</td><td>$Q(f) = Q(F)$</td><td>$R(f) = R(F) \times f/F$</td></tr>
<tr><td colspan="3">If F is set to zero, then Q is assumed to be infinite.<br>where<br>f = simulation frequency<br>F = reference frequency<br>R = resistance of inductor</td></tr>
</table>

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

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

## Equivalent Circuit



# InDQ2 (Inductor with Q)

## Symbol

## Parameters

L = inductance

Q = quality factor

F = frequency at which Q is given, in Hertz

Mode = loss mode for this device

1: Q being proportional to freq
2: Q being proportional to sqrt (freq)
3: Q being constant
4: Q being proportional to sqrt (freq), with constant L

Rdc = series constant resistance associated with the device, for Mode=2, 3, and 4 only

## Range of Usage

Since the impedance of the device is $Z= R+j \times 2 \times \pi \times freq \times L$, the Quality factor is given in this respect as:

$$\begin{aligned} Q &= imag(Z)/ real(Z) \\ &= 2 \times \pi \times freq \times L/R \end{aligned}$$

| Mode Setting | Q | R |
|---|---|---|
| proportional to freq | Q(f) = Q(F) × f/F | R(f) = R(F) |
| proportional to sqrt (freq) | Q(f) = Q(F) × sqrt(f/F) | R(f) = R(F) × sqrt(f/F) |
| constant | Q(f) = Q(F) | R(f) = R(F) × f/F |
| proportional to sqrt (freq), constant L | Q(f) = Q(F) × sqrt(f/F) | R(f) = R(F) × sqrt(f/F)<br>and L is constant |
| where<br>f = simulation frequency<br>F = reference frequency<br>R = resistance of the inductor | | |

## Notes/Equations

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

# L (Inductor)

## Symbol

## Parameters

| Name | Description | Unit | Default |
|---|---|---|---|
| L | inductance | fH, pH, nH, uH, mH | 1.0 nH |
| R | series resistance | mOhm, Ohm, kOhm, MOhm, GOhm | |
| Temp | nominal temperature | °C | |
| Trise | temperature rise over ambient | ˚C | 0 |
| Tnom | nominal temperature | °C | |
| TC1 | linear temperature coefficient | 1/°C | |
| TC2 | quadratic temperature coefficient | 1/°C$^2$ | |
| InitCond | transient analysis initial condition current | | |
| Noise | noise generation option: yes=1, no=0 | | yes |
| Model | model instance name | | |
| _M | number of inductors in parallel | | 1 |

## Notes/Equations

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

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

   The resistance, if specified, is not temperature scaled.

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

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

   $$<i^2> = 4kT/R.$$

4. If a model name is given, then values that are not specified on the inductor instance are taken from the model values. Typical values that can be defaulted are the inductance, series resistance, nominal temperature and temperature coefficients.
5. When InitCond is explicitly specified, the check-box *Use user-specified initial conditions* must be turned on in the *Convergence* tab of the Tran transient simulation controller for the parameter setting to take effect.
6. _M is used to represent the number of inductors in parallel and defaults to 1. M cannot be zero. If an inductor model is used, an optional scaling parameter Scale can also be defined on the model; it defaults to 1. The effective inductance that will be simulated is L × Scale/M; the effective resistance is R × Scale/M.
7. Table 2-1 lists the DC operating point parameters that can be sent to the dataset.

Table 2-1. DC Operating Point Information

| **Name** | **Description** | **Units** |
|---|---|---|
| I | Current | A |

# L_Conn (Inductor (Connector Artwork))

## Symbol

## Parameters

L = inductance, in henries

## Notes/Equations

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

# L_Model (Inductor Model)

## Symbol

## Parameters

| Name | Description | Unit | Default |
|---|---|---|---|
| L | default inductance | H | 1.0 nH |
| R | series resistance | ohms | |
| Tnom | nominal temperature | °C | |
| Trise | temperature rise over ambient | °C | 0 |
| TC1 | linear temperature coefficient | 1/°C | |
| TC2 | quadratic temperature coefficient | 1/°C$^2$ | |
| Scale (Scalei) | scaling factor | | 1 |
| Kf | flicker noise coefficient | | 0 |
| Af | flicker noise exponent | | 0 |
| Coeffs | nonlinear inductor polynomial coefficients | | |
| All Params | Data Access Component (DAC) based parameters | | |

## Netlist Format

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

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

The model statement starts with the required keyword *model*. It is followed by the *modelname* that will be used by inductor components to refer to the model. The third parameter indicates the type of model; for this model it is *L_Model*. The rest of the model contains pairs of model parameters and values, separated by an equal sign. The name of the model parameter must appear exactly as shown in the parameters table-these names are case sensitive. Some model parameters have aliases, which are listed in parentheses after the main parameter name; these are parameter names that can be used instead of the primary parameter name. Model parameters may appear in any order in the model statement. Model parameters that are not specified take the default value indicated in the parameters table. For more information about the ADS circuit simulator netlist format, refer to Chapter 8, *ADS Simulator Input Syntax* in the *Circuit Simulation* book.

Example:

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

## Notes/Equations

---

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

---

1. This model supplies values for an inductor L. This allows some common inductor values to be specified in a model.
2. Kf and Af add flicker noise using the equation:

   $$<i^2> \; = Kf \times I_{dc}^{Af} / f$$

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

   $$L(I) \; = \; \frac{dFlux(I)}{dl} = \; L(1 + c1 \times I + c2 \times I^2 + \ldots)$$

   where $L$ is the inductance of the instance, and $ck$ is the $k$-th entry in the Coeffs list.

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

   $$Flux(I) \; = \; L \times I \times \left(1 + \left(\frac{1}{2}\right) \times c1 \times I + \left(\frac{1}{3}\right) \times c2 \times I^2 + \ldots\right)$$

# L_Pad1 (Inductor (Pad Artwork))

## Symbol

## Parameters

L = inductance, in henries

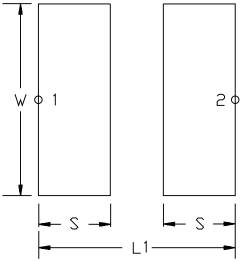
W = (ADS Layout option) width of pad, in specified units

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

L1 = nominal temperature, in °C

## Notes/Equations

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



# L_Space (Inductor (Space Artwork))

## Symbol

## Parameters

L = inductance, in henries

L1 = nominal temperature, in °C

## Notes/Equations

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

# 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 refer to notes):

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

Temp = temperature, in °C

## Range of Usage

$F \geq 0$

## Notes/Equations

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

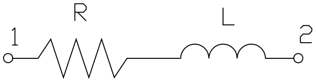
where:

| Mode Setting | Q | G |
|---|---|---|
| proportional to freq | $Q(f) = Q(F) \times f/F$ | G(f) = G(F) |
| proportional to sqrt (freq) | $Q(f) = Q(F) \times \sqrt{f/F}$ | $G(f) = G(F) \times \sqrt{f/F}$ |
| constant | Q(f) = Q(F) | $G(f) = G(F) \times f/F$ |
| If F is set to zero, then Q is assumed to be infinite.<br>where<br>f = simulation frequency<br>F = reference frequency<br>G = conductance of capacitor | | |

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

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

## Equivalent Circuit



# 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 refer to notes):

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

Temp = temperature, in °C

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

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

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

## Range of Usage

$F \geq 0$

## Notes/Equations

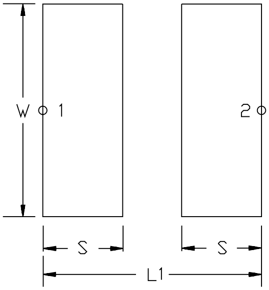
1. $Q = \frac{2\pi f L}{R}$

   where:

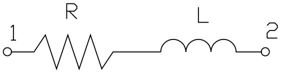
| Mode Setting | Q | G |
|---|---|---|
| proportional to freq | $Q(f) = Q(F) \times f/F$ | G(f) = G(F) |
| proportional to sqrt (freq) | $Q(f) = Q(F) \times \sqrt{f/F}$ | $G(f) = G(F) \times \sqrt{f/F}$ |

| Mode Setting | Q | G |
|---|---|---|
| constant | Q(f) = Q(F) | $G(f) = G(F) \times f/F$ |
| If F is set to zero, then Q is assumed to be infinite.<br>where<br>f = simulation frequency<br>F = reference frequency<br>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



# 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 refer to notes):

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

Temp = temperature, in °C

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

## Range of Usage

$F \geq 0$

## Notes/Equations

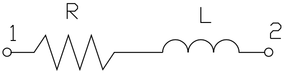
1. $Q = \frac{2\pi f L}{R}$

where:

| Mode Setting | Q | G |
|---|---|---|
| proportional to freq | $Q(f) = Q(F) \times f/F$ | $G(f) = G(F)$ |
| proportional to sqrt (freq) | $Q(f) = Q(F) \times \sqrt{f/F}$ | $G(f) = G(F) \times \sqrt{f/F}$ |
| constant | $Q(f) = Q(F)$ | $G(f) = G(F) \times f/F$ |
| If F is set to zero, then Q is assumed to be infinite.<br>where<br>f = simulation frequency<br>F = reference frequency<br>G = conductance of capacitor | | |

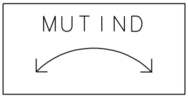
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.

## Equivalent Circuit

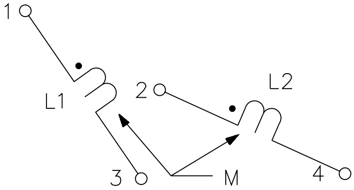


# Mutual (Mutual Inductor)

## Symbol



## Illustration



## 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

$-1.0 \le K \le 1.0$

## Notes/Equations

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

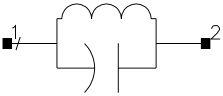
**Note** To edit *string* parameters on a schematic, highlight the parameter and enter a value enclosed with double quote symbols.

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

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

# PLC (Parallel Inductor-Capacitor)

## Symbol



## Parameters

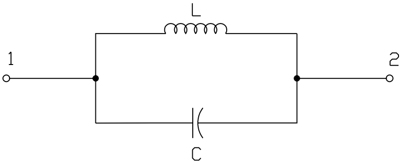
L = inductance, in henries

C = capacitance in farads

## Notes/Equations

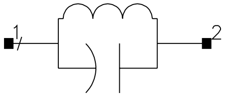
1. Use for high Q circuits rather than individual components in parallel.
2. This component has no default artwork associated with it.

## Equivalent Circuit



# PLCQ (Parallel Inductor-Capacitor with Q)

## 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 refer to notes):

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

C = capacitance, in farads

Qc = quality factor of capacitor

Fc = frequency at which capacitor Q is given, in hertz

ModC = frequency dependence mode of capacitor Q; options (also refer to notes): proportional to freq; proportional to sqrt(freq); constant; (value type: enumerated)

Rdc = resistance for modes 2 and 3

## Notes/Equations

1. Use for high Q circuits, rather than individual components in parallel.
2. $Ql = \frac{2\pi f_S L}{R}$ (for inductors) $Qc = \frac{2\pi f_S C}{G}$ (for capacitors)

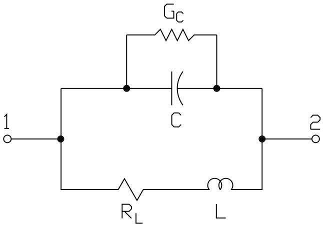
where

| ModL Setting | Ql | ModC Setting Ql | Qc |
|---|---|---|---|
| proportional to freq | $Ql(Fl) \times f_S / Fl$ | proportional to freq | $Qc(Fc) \times f_S / Fc$ |
| proportional to sqrt (freq) | $Q(f) = Q(F) \times \sqrt{f / F}$ | $G(f) = G(F) \times \sqrt{f / F}$ | proportional to sqrt (freq) |

| **ModL Setting** | **Ql** | **ModC Setting Ql** | **Qc** |
|---|---|---|---|
| constant | Ql(Fl) | constant | Qc(Fc) |
| where<br>R = resistance of inductor<br>G= conductance of capacitor<br>$f_s$ = simulation frequency<br>Fc, Fl = specified Q frequencies | | | |

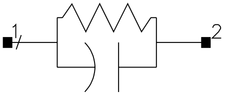
3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component has no default artwork associated with it.

## Equivalent Circuit



# PRC (Parallel Resistor-Capacitor)

## Symbol



## Parameters

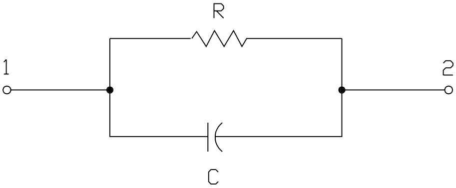
R = resistance, in ohms

C = capacitance, in farads

## Notes/Equations

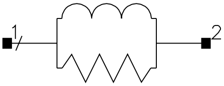
1. This component has no default artwork associated with it.

## Equivalent Circuit



# PRL (Parallel Resistor-Inductor)

## Symbol



## Parameters

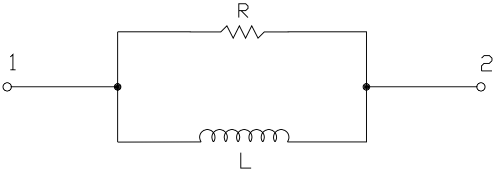
R = resistance, in ohms

C = capacitance, in henries

## Notes/Equations

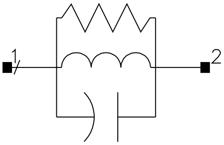
1. This component has no default artwork associated with it.

## Equivalent Circuit



# PRLC (Parallel Resistor-Inductor-Capacitor)

## Symbol



## Parameters

R = resistance, in ohms

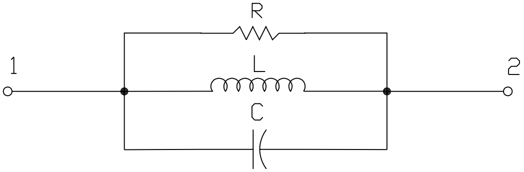
L = inductance, in henries

C = capacitance, in farads

## Notes/Equations

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

## Equivalent Circuit



# R (Resistor)

## Symbol

## Parameters

| Name | Description | Unit | Default |
|---|---|---|---|
| R | resistance | mOhm, Ohm, kOhm, MOhm, GOhm | 50 Ohm |
| Temp | temperature | °C | |
| Trise | temperature rise above ambient | °C | 0 |
| Tnom | nominal temperature | °C | 25 |
| TC1 | linear temperature coefficient | 1/°C | |
| TC2 | quadratic temperature coefficient | $1/°C^2$ | |
| Noise | resistor thermal noise option: yes=enable; no=disable | | yes |
| wPmax | maximum power dissipation (warning) | pW, nW, uW, mW, W, kW, dbm | |
| wImax | maximum current (warning) | fA, pA, nA, uA, mA, A | |
| Model | name of a resistor model to use | | |
| Width | physical width for use with a model | um, mm, cm, meter, mil, in | model |
| Length | physical length for use with a model | um, mm, cm, meter, mil, in | model |
| _M | number of resistors in parallel | | 1 |

## Notes/Equations

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

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

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

3. The resistor generates thermal noise:

   $$\langle i^2 \rangle = 4kT/R$$

Noise generation can be disabled by setting Noise=no.

4. wPmax and wImax are used by the overload alert feature. They set limits on the maximum instantaneous power dissipated by the resistor and maximum current through the resistor. If these limits are specified, the simulator will issue a warning the first time they are exceeded during a dc, harmonic balance or transient simulation. Simulation results are not affected by this parameter.
5. For a transient simulation, the resistance can vary with time. The resistance value should be assigned an expression that is a function of the reserved variable time, which is the simulation time in seconds.
6. If a model name is given, then values that are not specified on the resistor instance are taken from the model values. Typical values that can be defaulted are resistance, length and width, nominal temperature, temperature coefficients, and overload alert parameters.

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

   $$R' = R + Rsh \times \frac{(Length - 2 \times Narrow - 2 \times Dw)}{(Width - 2 \times Narrow - 2 \times Dl)}$$

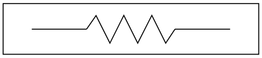
7. _M is used to represent the number of resistors in parallel and defaults to 1. M cannot be zero. If a resistor model is used, an optional scaling parameter Scale can also be defined on the model; it defaults to 1. The effective resistance that will be simulated is R × Scale/M.
8. Table 2-2 lists the DC operating point parameters that can be sent to the dataset.

Table 2-2. DC Operating Point Information

| **Name** | **Description** | **Units** |
|---|---|---|
| I | Current | A |

# R_Model (Resistor Model)

## Symbol



## Parameters

| Name | Description | Unit | default |
|---|---|---|---|
| R | resistance | mOhm, Ohm, kOhm, MOhm, GOhm | 50 Ohm |
| Rsh | sheet resistance | ohms/square | |
| Length (L) | default length | um, mm, cm, meter, mil, in | |
| Width (W) | default width | um, mm, cm, meter, mil, in | |
| Narrow | length and width narrowing due to etching | um, mm, cm, meter, mil, in | |
| Tnom | nominal temperature | °C | |
| Trise | temperature rise over ambient | ˚C | 0 |
| TC1 | linear temperature coefficient | 1/°C | |
| TC2 | quadratic temperature coefficient | $1/°C^2$ | |
| wPmax | maximum power dissipation (warning) | pW, nW, uW, mW, W, kW, dbm | |
| wImax | maximum current (warning) | amperes or amperes/meter | |
| Scale (Scaler) | resistance scaling factor | | 1 |
| AllParams | DataAccessComponent-based parameters | | |
| Dw (Etch) | width narrowing due to etching | in specified units | |
| Di (Etchl) | length narrowing due to etching | in specified units | |
| Kf | flicker noise coefficient | | 0 |
| Af | flicker noise exponent | | 0 |
| Coeffs | nonlinear resistor polynomial coefficients | | |

## Netlist Format

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

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

The model statement starts with the required keyword *model*. It is followed by the *modelname* that will be used by resistor components to refer to the model. The third parameter indicates the type of model; for this model it is *R_Model*. The rest of the model contains pairs of model parameters and values, separated by an equal sign. The name of the model parameter must appear exactly as shown in the parameters table-these names are case sensitive. Some model parameters have aliases, which are listed in parentheses after the main parameter name; these are parameter names that can be used instead of the primary parameter name. Model parameters may appear in any order in the model statement. Model parameters that are not specified take the default value indicated in the parameters table. For more information about the ADS circuit simulator netlist format, refer to Chapter 8, *ADS Simulator Input Syntax* in the *Circuit Simulation* book.

Example:

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

## Notes/Equations

---

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

---

1. R_Model supplies model parameters for use with a resistor R. This allows physically-based resistors to be modeled based on length and width.
2. When the physical parameters Rsh, Width and Length are specified, wImax is the current limit in amperes/meter:

   $$wImax' = wImax \times (Width - 2 \times Narrow - 2 \times Dw)$$

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

3. Use AllParams with a DataAccessComponent to specify file-based parameters (refer to DataAccessComponent). Note that model parameters that are explicitly specified take precedence over those specified via AllParams.
4. Kf and Af add flicker noise using the equation:

   $$< i^2 > \; = Kf \times I_{dc}^{Af} / f$$

5. The resistor value can be made a nonlinear function of the applied voltage V by specifying the polynomial coefficients list (Coeffs = list(c1, c2, c3, ...) ). The resistance value $R(V)$ is then given by:

   $$R(V) \; = \; \frac{dV}{dI} = \frac{R}{(1 + c1 \times V + c2 \times V^2 + \ldots)}$$

   where $R$ is the resistance of the instance, and $ck$ is the $k$-th entry in the Coeffs list.

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

   $$I(V) \; = \; \left(\frac{V}{R}\right) \times \left(1 + \left(\frac{1}{2}\right) \times c1 \times V + \left(\frac{1}{3}\right) \times c2 \times V^2 + \ldots\right)$$

# R_Conn (Resistor (Connector Artwork))

## Symbol

## Parameters

R = resistance, in ohms

## Notes/Equations

1. For time-domain analysis, the resistance can vary with time. The resistance value should be an equation whose value is calculated from the reserved variable *_time*.
2. This component is a single connection in layout. For example, it can be used to represent parasitics.

# R_dxdy (Resistor (Delta X - Delta Y))

## Symbol

## Parameters

R = resistance, in ohms

dx = delta X, in specified units

dy = delta Y, in specified units

Temp = temperature, in °C

## Notes/Equations

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

## R_Pad1 (Resistor (Pad Artwork))

### Symbol

### Parameters

R = resistance, in ohms

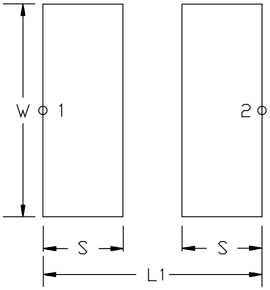
W = (ADS Layout option) width of pad, in specified units

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

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

### Notes/Equations

1. For transient and convolution analyses, resistance can vary with time. The resistance value should be an equation whose value is calculated from the reserved variable _*time*.
2. This component's artwork is composed of two rectangular pads with pins on the outer edges, as shown:



# R_Space (Resistor (Space Artwork))

## Symbol

## Parameters

R = resistance, in ohms

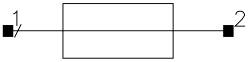
L1 = (ADS Layout option) pin-to-pin distance, in specified units

## Notes/Equations

1. For time-domain analysis, the resistance can vary with time. The resistance value should be an equation whose value is calculated from the reserved variable *_time*.
2. This component is represented as a connected gap in layout—into which a custom artwork object can be inserted.

## Short (Short)

### Symbol



### 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)

### Notes/Equations

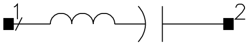
1. This component behaves like a current probe. It can be used to measure the current anywhere in the circuit.
2. The variable name for the current is *label.i*, where *label* is the label of this component.
3. Table 2-3 lists the DC operating point parameters that can be sent to the dataset.

Table 2-3. DC Operating Point Information

| **Name** | **Description** | **Units** |
|---|---|---|
| I | Current | A |

## SLC (Series Inductor-Capacitor)

### Symbol



### Parameters

L = inductance, in henries

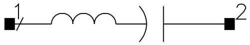
C = capacitance, in farads

### Notes/Equations

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

# 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 refer to notes):

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

C = capacitance, in farads

Qc = quality factor of capacitor

Fc = frequency at which capacitor Q is given, in hertz

ModC =frequency dependence mode of capacitor Q; options (also refer to notes):

1 is proportional to freq

2 is proportional to sqrt(freq)

3 is constant

Rdc = resistance for modes 2 and 3

## Notes/Equations

1. Use when modeling high Q circuits rather than individual components in series.
2. $Ql = \frac{2\pi f_s L}{R}$ (for inductors) $Qc = \frac{2\pi f_s C}{G}$ (for capacitors)

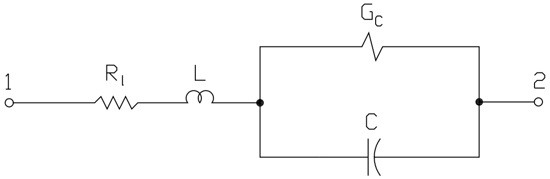
where

| ModL Setting | Ql | ModC Setting | Qc |
|---|---|---|---|
| *proportional to freq* | $Ql(Fl) \times f_s / Fl$ | *proportional to freq* | $Qc(Fc) \times f_s / Fc$ |
| *proportional to sqrt(freq)* | $Ql(Fl) \times \sqrt{f_s / Fl}$ | *proportional to sqrt(freq)* | $Qc(Fc) \times \sqrt{f_s / Fc}$ |
| *constant* | Ql(Fl) | *constant* | Qc(Fc) |
| where<br>R = resistance of inductor<br>G= conductance of capacitor<br>$f_s$ = simulation frequency<br>Fc, Fl = specified Q frequencies | | | |

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

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

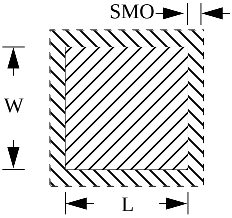
## Equivalent Circuit



# 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

$W \geq 0$
$L \geq 0$

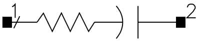
## Notes/Equations

1. This component is required for layout of SMT library parts.
2. For any library item to which this component is attached, the PO parameter specifies the offset of the bond pad center from the position of pin connections designated for that item’s package artwork. An offset of 0 centers the pad around the location of the pins. A positive value shifts the pad away from the package body; a negative value shifts the pad toward the package body.

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

# SRC (Series Resistor-Capacitor)

## Symbol



## Parameters

R = inductance, in ohms

C = capacitance, in farads

## Notes/Equations

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

# SRL (Series Resistor-Inductor)

## Symbol

## Parameters

R = resistance, in ohms

C = inductance, in henries

## Notes/Equations

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

## SRLC (Series Resistor-Inductor-Capacitor)

### Symbol

### Parameters

R = resistance, in ohms

L = inductance, in henries

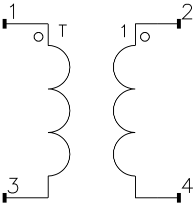
C = capacitance, in farads

### Notes/Equations

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

# TF (Transformer)

## Symbol



## Parameters

T = turns ratio T1/T2

## Notes/Equations

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

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

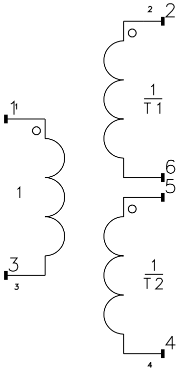
2. Parasitic inductances of the primary and secondary are not modeled. To do this, use the component for mutual inductance (M). The ends that are in phase are identified by a small open circle on the schematic symbol.

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.

# TF3 (3-Port Transformer)

## Symbol



## Parameters

T1 = turn 1

T2 = turn 2

## Notes/Equations

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

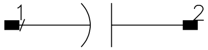
$$T = \frac{T_{primry}}{T_{secondary}}$$

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 $\geq 0$

## Notes/Equations

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

   The frequency-dependence of Q is given by

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

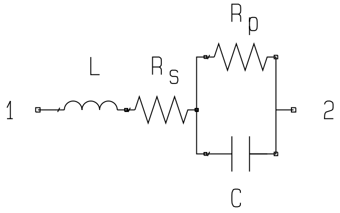
   where f is the simulation frequency and Q(FreqQ) is the specified value of Q at FreqQ.
2. If Q or FreqQ are set to 0, Q is assumed to be infinite.
3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component has no default artwork associated with it.

## References

[1] C. Bowick, *RF Circuit Design*, Howard Sams & Co., 1987.

[2] *The RF Capacitor Handbook*, American Technical Ceramics Corp., September 1983.

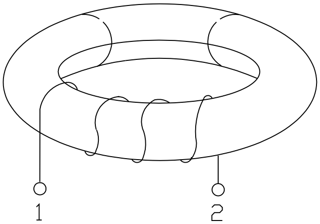
## Equivalent Circuit



# CIND (Ideal Torroidal Inductor)

## Symbol

## Illustration



## Parameters

N = number of units

AL = inductance index

## Range of Usage

N, AL > 0

## Notes/Equations

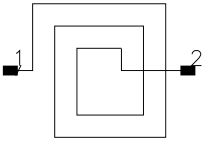
1. The inductance is given by

   $L = N^2 \times AL$

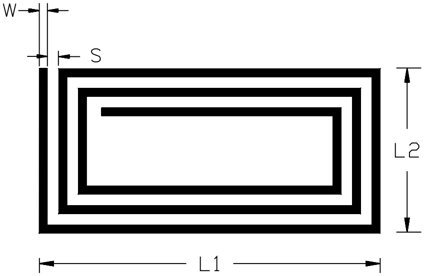
2. This component has no default artwork associated with it.

# RIND (Rectangular Inductor)

## Symbol



## Illustration



## 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/Equations

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

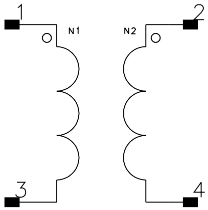
2. This component has no default artwork associated with it.

## References

[1] H. M. Greenhouse, “Design of Planar Rectangular Microelectronic Inductors,” *IEEE Transactions as Parts, Hybrids, and Packaging*, Vol. PHP-10, No. 2, June 1974, pp. 101-109.

# XFERP (Physical Transformer)

## Symbol



## Parameters

N = turns ratio N1/N2

Lp = magnetizing inductance, in henries

Rc = core loss resistance, in ohms

K = coefficient of coupling

R1 = primary loss resistance, in ohms

R2 = secondary loss resistance, in ohms

C1 = primary capacitance, in farads

C2 = secondary capacitance, in farads

C = interwinding capacitance, in farads

## Range of Usage

$0 < K < 1$

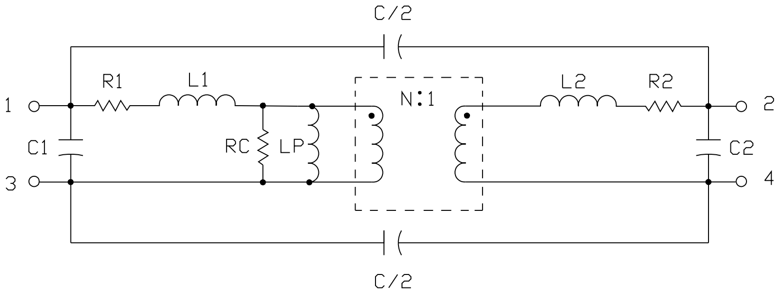
## Notes/Equations

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

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

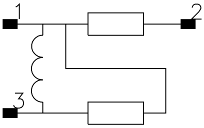
2. This component has no default artwork associated with it.

## Equivalent Circuit



# XFERRUTH (Ruthroff Transformer)

## Symbol



## Parameters

N = number of turns

AL = inductance index, in henries

Z = characteristic impedance of transmission line, in ohms

E = electrical length of transmission line, in degrees

F = reference frequency for electrical length, in hertz

## Range of Usage

$N > 0$
$AL > 0$

## Notes/Equations

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

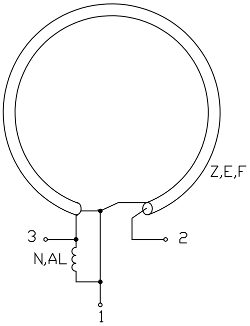
## References

[1] H. L. Krauss and C. W. Allen. “Designing torroidal transformers to optimize wideband performance,” *Electronics*, pp. 113-116, August 16, 1973.

[2] O. Pitzalis, Jr. and T. P. Couse. “Practical Design Information for Broadband Transmission Line Transformers,” *Proceedings of the IEEE*, Vol. 56, No. 4 (Proceedings Letters), April 1968, pp. 738-739.

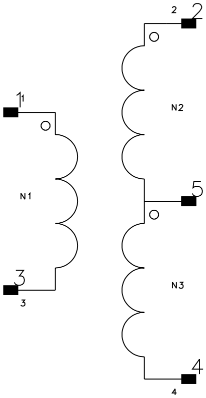
[3] C. L. Ruthroff. “Some Broadband Transformers,” *Proceedings of the IRE*, Vol. 47, No. 8, August 1959, pp. 1337-1342 (Fig. 3, page 1338, in particular).

## Equivalent Circuit



# XFERTAP (Tapped Secondary Ideal Transformer)

## Symbol



## Parameters

N12 = turns ratio, N1/N2

N13 = turns ratio, N1/N3

L1 = primary winding inductance, in henries

K = coupling coefficient

## Range of Usage

$0 < K < 1$

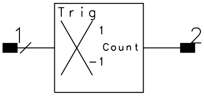
## Notes/Equations

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

# Chapter 4: Probe Components

# Counter (Counter Component)

## Symbol



## Parameters

Direction = direction one

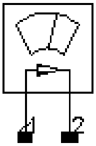
Thresh = threshold one, in volts

## Notes/Equations

1. This time counter model generates an output voltage equal to the number of times that the user-specified trigger has occurred. The trigger point is defined by setting a threshold voltage and a slope. The slope can be specified as either rising or falling by setting the direction parameter to a 1 or –1. A direction parameter value of 0 is used if a trigger for either slope is desired.
2. Only the baseband component of the input voltages is used to generate the trigger, so the model may be used in either envelope or transient time domain analysis modes. Linear interpolation is used to estimate the actual trigger crossing time to a finer resolution than the simulation time step.
3. The input impedance is infinite. The output impedance is 1 ohm. The open circuit output voltage is equal to $n$, the number of triggers that have occurred up 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)

## Notes/Equations

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

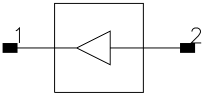
Table 4-1. DC Operating Point Information

| Name | Description | Units |
|---|---|---|
| I | Current | A |

4. 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 (default: 1.1 ohms)

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

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

FundIndex = fundamental number for oscillator (default: 1)

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

MaxLoopGainStep = maximum arc length continuation step size during loop-gain search

## Notes/Equations

1. This is a special device used for an oscillator analysis. Do not use more than one oscillator port in a circuit.
2. NumOctaves specifies the total number of octaves over which the oscillator search is done. Half of the octaves are below the initial frequency and half are above. For example, if NumOctaves is 2, then the frequency search goes from Freq/2 to Freq × 2. Steps sets the number of frequency points per octave that are used in the search. For a high-Q oscillator, a large number of steps might be required.
3. If fundamental voltage V is not specified, the simulator first performs a small-signal AC analysis to determine the actual frequency and oscillation voltage.

   If V is specified, it represents an initial guess at the fundamental oscillator voltage at the point where the OscPort is inserted. The initial guess for V should be as close to the actual value as possible. An inaccurate value increases the simulation time and might prevent convergence. If it is not known, don’t specify it.

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

   **Feedback Oscillators (such as Colpitts)**

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

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

   **Negative Resistance Oscillators**

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

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

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

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

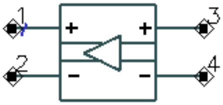
FundIndex is used for selecting which fundamental tone is considered the unknown frequency during oscillator analysis.

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

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

# OscPort2 (Differential Oscillator Port)

## Symbol



## Parameters

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

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

Z = initial value for $Z_0$, in ohms (all modes) (default: 1.1 ohms)

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

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

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

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

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

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

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

## Notes/Equations

1. This is a special device used for an oscillator analysis. Do not use more than one oscillator test element (OscTest, OscPort, OscPort2) in a circuit.
2. NumOctaves specifies the total number of octaves over which the oscillator search is done. Half of the octaves are below the initial frequency and half are above. For example, if NumOctaves is 2, then the frequency search goes from Freq/2 to Freq × 2. Steps sets the number of frequency points per octave that are used in the search. For a high-Q oscillator, a large number of steps might be required.
3. If a fundamental voltage V is not specified, the simulator first performs a small-signal AC analysis to determine the actual frequency and oscillation voltage.

If V is specified, it represents an initial guess at the fundamental oscillator voltage at the point where the OscPort is inserted. The initial guess for V should be as close to the actual value as possible. An inaccurate value increases the simulation time and might prevent convergence. If it is not known, don't specify it.

4. This device can operate in one of three different modes. In automatic mode, it is similar to the OscPort device, and is used with the harmonic balance oscillator analysis to determine the oscillator frequency, large signal solution and optionally phase noise. In small signal loop gain mode, it is similar to the OscTest device, and is used to perform a small signal analysis of the oscillator loop gain versus frequency. In large signal loop gain mode, it is used to simulate the large signal nonlinear loop gain of the oscillator versus frequency and injected loop voltage.
5. This device can be used for both single-ended and differential oscillator topologies. For single-ended oscillators, the negative pins of this element should be grounded. For differential oscillators, it should be connected differentially into the oscillator loop.
6. Provided the circuit produces at least one complex conjugate pole pair in the right-half-plane over the frequency range tested, the analysis will determine the oscillation waveform and amplitude. Proper probe placement and impedance can reduce the analysis time significantly and help ensure accurate oscillator analysis results. To reduce the probability of a failed analysis, place the probe and set the initial impedance in a manner consistent with the following guidelines:

**Feedback Oscillators (such as Colpitts)**

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

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

### Negative Resistance Oscillators

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

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

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

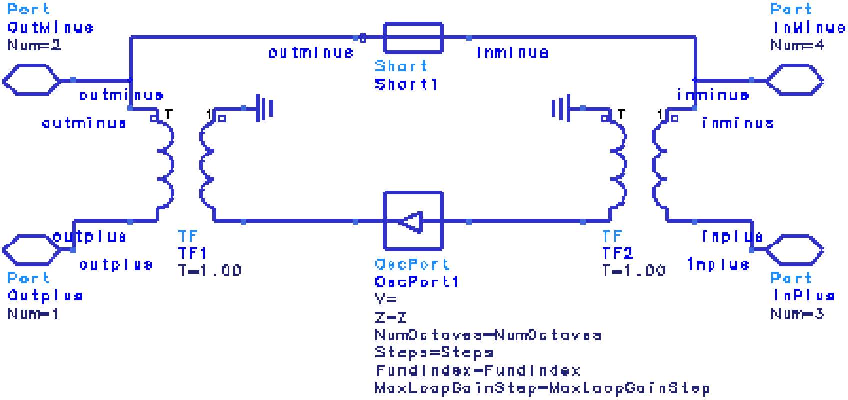
Another approach is to try to get the oscillator to oscillate at some nicer parameter value and then to sweep the parameter value to the desired value. The parameter may be bias, self-bias impedance, some gain controlling value, or another factor. In short, anything that will make the oscillator more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

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

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

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

The equivalent circuit for OscPort2 in automatic mode is shown 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 mode. The analysis calculates and places in the dataset a complex value called LoopGain which is the small signal loopgain of the oscillator.

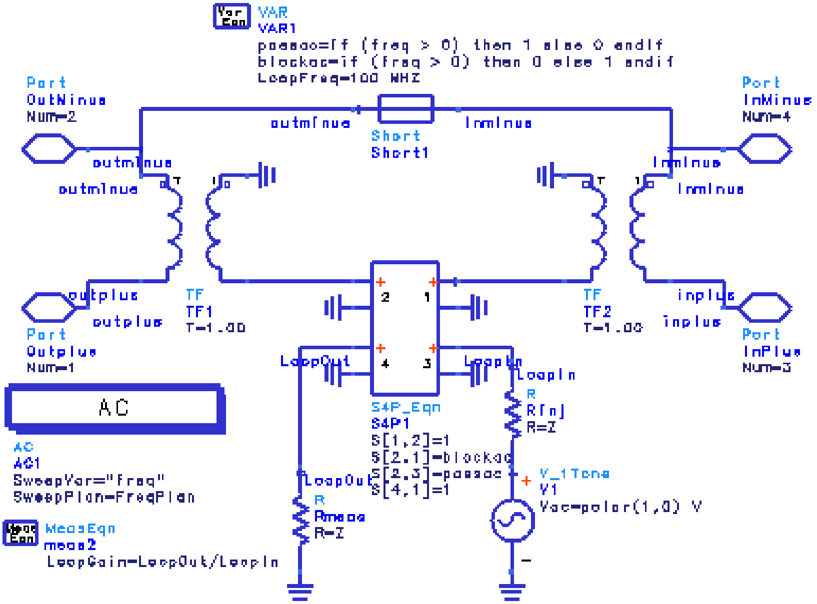
   The range of frequencies over which to analyze loop gain should be specified with a SweepPlan item. The name of this SweepPlan should then be assigned to the parameter FreqPlan.

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

The number of clockwise encirclements of the 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

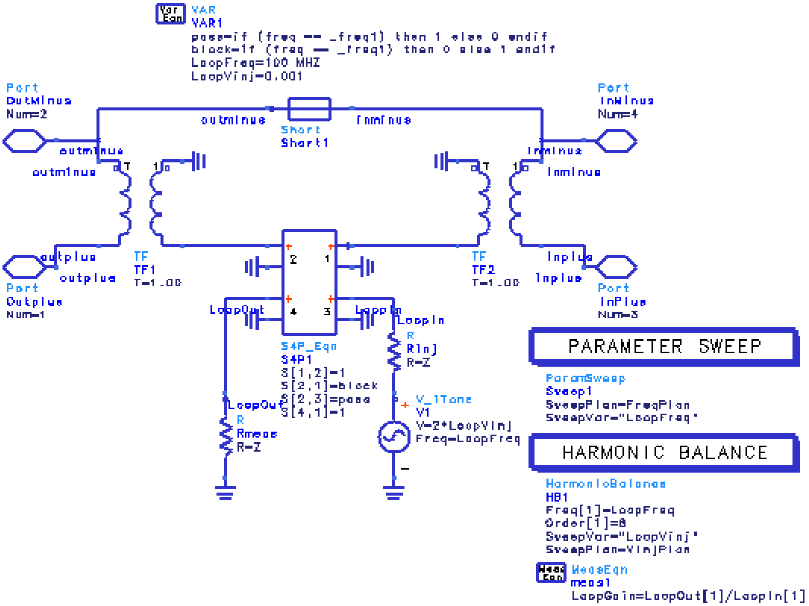
analysis calculates and places in the dataset a complex value called LoopGain which is the large signal loop gain of the oscillator. The circuit will sustain oscillation at the point at which the magnitude of LoopGain equals one and the phase of LoopGain equals zero.

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

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

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

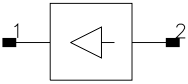
The scattering matrices follow.

$$S_{FUNDAMENTAL} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$S_{OTHER} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

# OscTest (Grounded Oscillator Test)

## Symbol



## Parameters

Port_Number = port number

Z = port impedance, in ohms

Start = start frequency, in hertz

Stop = stop frequency, in hertz

Points = number of frequency points

## Notes/Equations

1. This component performs an S-parameter analysis to evaluate the closed loop, small signal gain of a potential oscillator. It contains an analysis controller and sweeps the frequency from Start to Stop. S(1,1) is the loop gain.
2. This device is used to evaluate the ability of a closed-loop system to produce one or more complex conjugate pole pairs in the right-half-plane (RHP) of a pole/zero diagram. This device measures the open-loop gain and phase of the closed-loop system. These results must be plotted on a polar graph (Nyquist diagram) to properly interpret them.

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

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

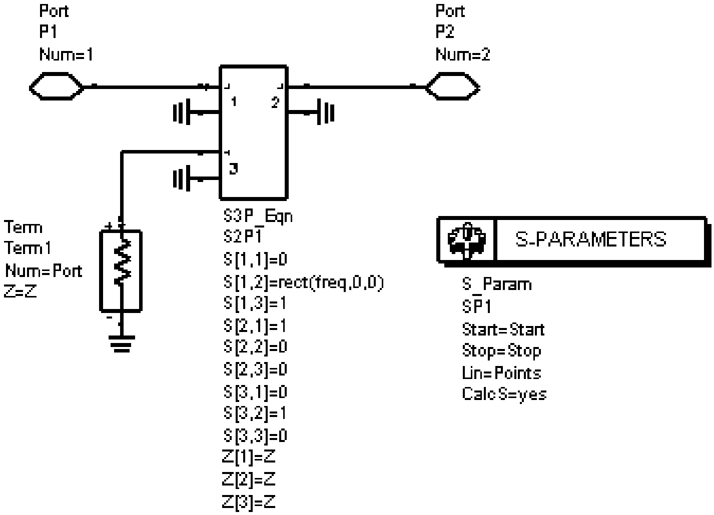
3. Another way of looking for potential oscillations is to look for the point(s) where the magnitude of the loop gain is greater than 1, phase is 0, and the phase is decreasing with increasing frequency.
4. If Port is set to 1 and $S_{11}$ never goes outside the unity circle, the circuit will not oscillate and the frequency search will fail. The circuit must be redesigned, or it will be entered incorrectly. For the circuit to oscillate, the simulated loop gain must be greater than unity (1) when the phase is 0. If Port is set to 2, $S_{22}$ is the S-parameter to test; if Port is set to 3, $S_{33}$ is the S-parameter to test, and so on.

   If the oscillator analysis fails, and this test indicates that the circuit should oscillate, the failure may be because the circuit is too nonlinear. This problem can be solved by trying different impedance values for OscTest (determined by the Z attribute). Lower impedance values usually work better, presumably because most nonlinearities are voltage controlled instead of current controlled.

   Reversing the OscTest direction should also be tried. The component could be inserted in the wrong direction; or, as occurs with some reflection oscillator cases, the solution may converge with the oscillator inserted in one direction and not the other.

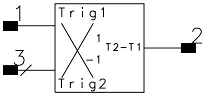
   Another alternative is to try to get the oscillator to oscillate at some nicer parameter value and then sweep the parameter value to the desired value. The parameter may be bias, self-bias impedance, some gain controlling value, or another factor; in short, anything that will make the oscillator more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

5. The equivalent circuit of OscTest is shown next.



# TimeDelta (Time Delta Component)

## Symbol



## Parameters

Direction1 = direction one

Direction2 = direction two

Thresh1 = threshold one, in volts

Thresh2 = threshold two, in volts

Scale = scale factor

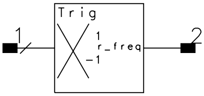
## Notes/Equations

1. TimeDelta generates an output voltage proportional to the time difference between two trigger points on two different baseband input voltage waveforms. The trigger points are user-defined by setting a threshold voltage and a slope. The slope can be specified as either rising or falling by setting the direction parameter to a 1 or –1. A direction parameter value of 0 is used if a trigger for either slope is desired.
2. Only the baseband component of the input voltages are used to generate the triggers, so this model can be used in either envelope or transient time domain analysis modes. Linear interpolation is used to estimate the actual trigger crossing time to a finer resolution than the simulation time step.
3. The input impedances are infinite. The output impedance is 1 ohm. The open circuit output voltage is equal to the time difference between the trigger2 and trigger1 events, multiplied by the scaling factor. The output does not change until a trigger2 event occurs and is held constant until another trigger2 event occurs. The scaling factor is used so that the output voltages can be set to reasonable values (i.e., not nanovolts) which would often be less than the simulator's absolute convergence criteria.
4. Several example measurements possible with this model might be the input to output propagation delay of a circuit, the -40 to +20 dBm rise time of a demodulated RF pulse, various fall times, pulse widths, etc. The output voltage

can be used for other behavioral models, for optimization, or for output to presentations.

# TimeFrq (Time Frequency Component)

## Symbol



## Parameters

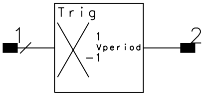
Direction = direction one

Thresh = threshold one, in volts

Scale = scale factor

# TimePeriod (Time Period Component)

## Symbol



## Parameters

Direction = direction one

Thresh = threshold one, in volts

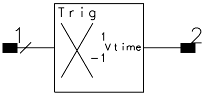
Scale = scale factor

## Notes/Equations

1. This time period model generates an output voltage proportional to the time between two consecutive triggers. The trigger point is defined by setting a threshold voltage and a slope. The slope can be specified as either rising or falling by setting the direction parameter to a 1 or –1. A direction parameter value of 0 is used if a trigger for either slope is desired.
2. Only the baseband component of the input voltages is used to generate the trigger, so this model may be used in either envelope or transient time domain analysis modes. Linear interpolation is used to estimate the actual trigger crossing time to a significantly higher resolution than the simulation time step.
3. The input impedance is infinite. The output impedance is 1 ohm. The open circuit output voltage is equal to the time difference between the last two trigger events multiplied by the scaling factor. The output does not change until a trigger event occurs and is held constant until another event occurs. The scaling factor is used so that the output voltage can be set to reasonable value which might otherwise be less than the simulator's absolute convergence criteria.

## TimeStamp (Time Stamp Component)

### Symbol



### Parameters

Direction = direction one

Thresh = threshold one, in volts

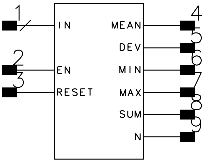
Scale = scale factor

### Notes/Equations

1. TimeStamp generates an output voltage proportional to the time that the last user-defined trigger occurred. The trigger point is defined by setting a threshold voltage and a slope. The slope can be specified as either rising or falling by setting the direction parameter to a 1 or –1. A direction parameter value of 0 is used if a trigger for either slope is desired.
2. Only the baseband component of the input voltages is used to generate the trigger, so the model may be used in either envelope or transient time analysis modes. Linear interpolation is used to estimate the actual trigger crossing time to a significantly higher resolution than the simulation time step domain.
3. The input impedance is infinite. The output impedance is 1 ohm. The open circuit output voltage is equal to the time of the last trigger event multiplied by the scaling factor. The output does not change until a trigger event occurs and is held constant until another event occurs. The scaling factor is used so that the output voltage can be set to a reasonable value which might otherwise be less than the simulator’s absolute convergence criteria.

# WaveformStats (WaveformStats Component)

## Symbol

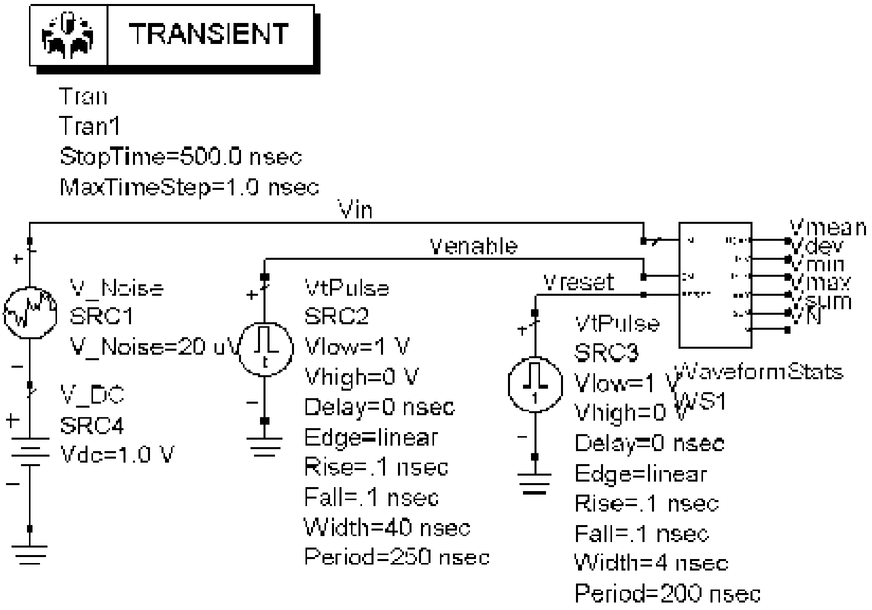


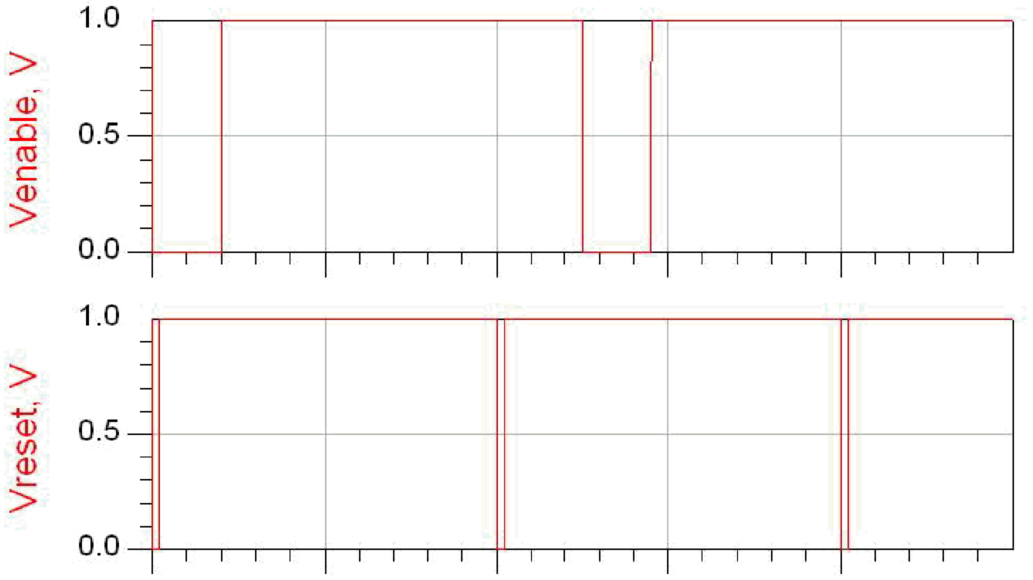
## Parameters

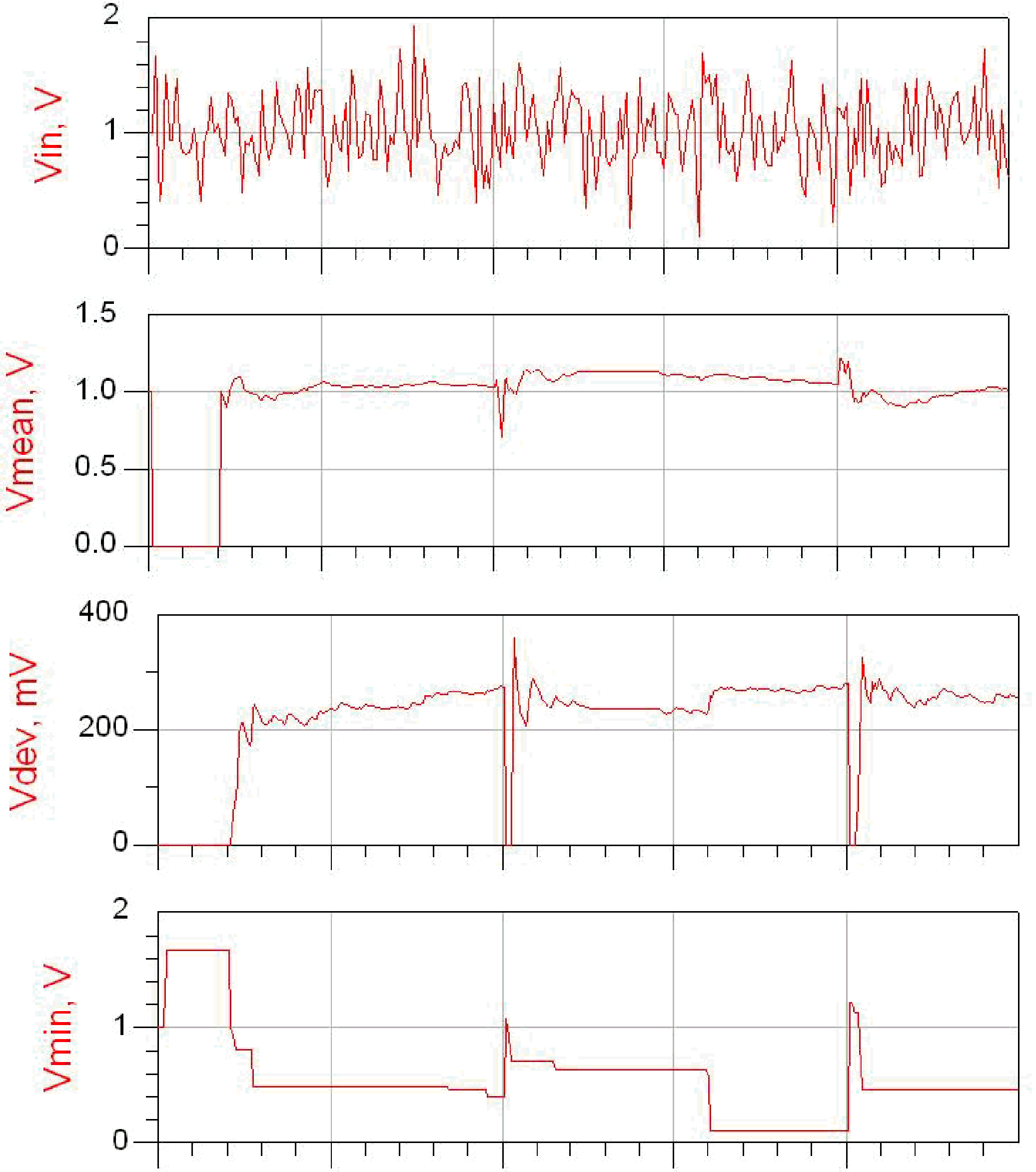
None

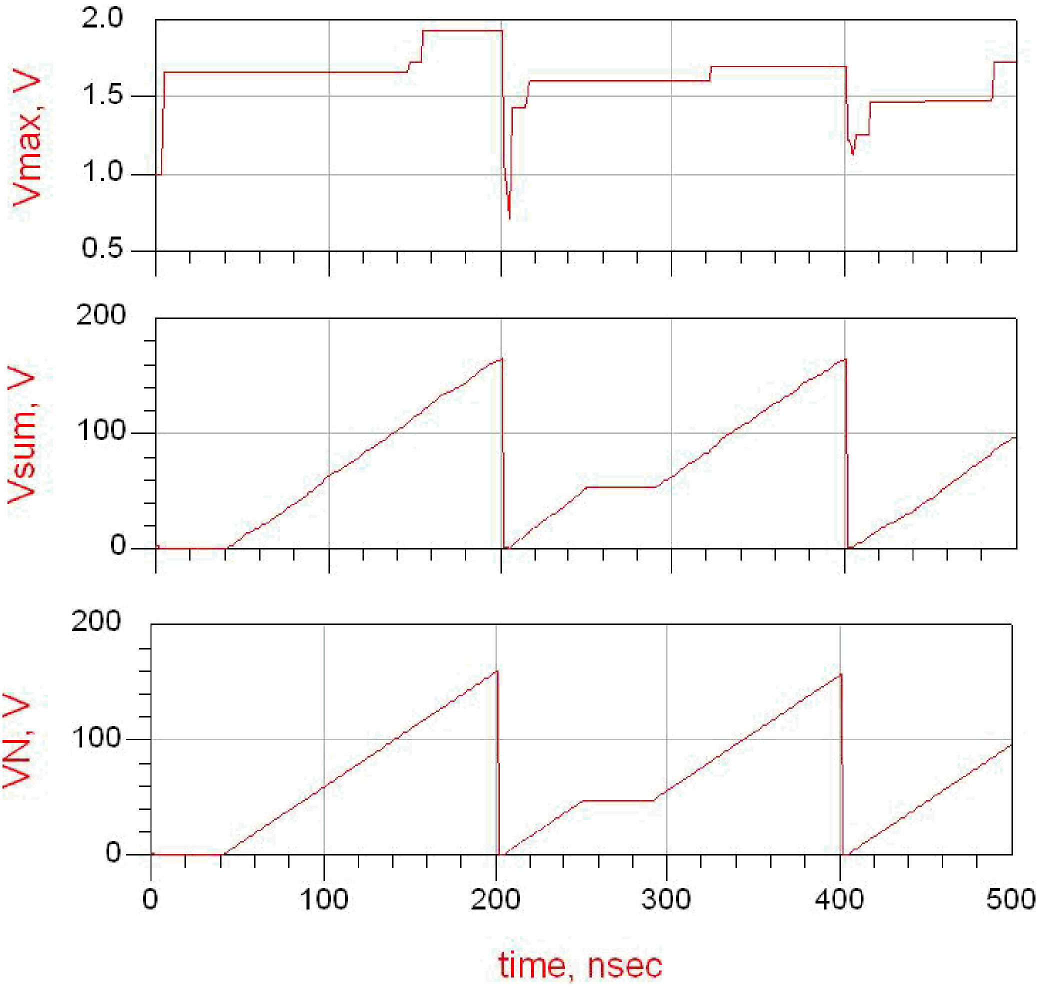
## Notes/Equations

1. This behavioral model can be used to measure the statistics of the baseband component of the input voltage. The inputs all have infinite input impedance; all outputs are ideal voltage sources with zero output impedance.
2. It calculates the running statistics of the signal since the last time reset went to 0; the reset signal should be high (1) for normal operation. The enable signal must be high (1) for normal operation; it can be put in hold mode temporarily by bringing the enable signal to 0. The calculated running statistics are mean, standard deviation, minimum, maximum, sum, and number of samples of the input signal.
3. If the enable is low during a reset, the accumulators are reset to 0; if the enable is high, then N is set to 1, and Sum is set equal to the input.
4. In addition to making gated, statistical measurements for use in optimizations or presentations, you can use this device to model circuits such as ideal integrate-and-dump circuits or peak detector circuits.
5. To measure the statistics of an RF carrier in circuit envelope mode, the correct demodulator must first be used to create a baseband voltage that can then be used as an input to this device.
6. This model operates in transient and envelope time domain analysis modes.
7. The schematic example shows how this component works and a plot of the signals after simulating it.


TRANSIENT
Tran
Tran1
StopTime=500.0 nsec
MaxTimeStep=1.0 nsec
Vin
Venable
Vreset
V_Noise
SRC1
V_Noise=20 uV
V_DC
SRC4
Vdc=1.0 V
VtPulse
SRC2
Vlow=1 V
Vhigh=0 V
Delay=0 nsec
Edge=linear
Rise=.1 nsec
Fall=.1 nsec
Width=40 nsec
Period=250 nsec
VtPulse
SRC3
Vlow=1 V
Vhigh=0 V
Delay=0 nsec
Edge=linear
Rise=.1 nsec
Fall=.1 nsec
Width=4 nsec
Period=200 nsec
WaveformStats
WS1
Vmean
Vdev
Vmin
Vmax
Vsum
VN


Venable, V
1.0
0.5
0.0
Vreset, V
1.0
0.5
0.0

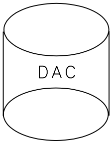

2
1
0
Vin, V
1.5
1.0
0.5
0.0
Vmean, V
400
200
0
Vdev, mV
2
1
0
Vmin, V


Vmax, V
2.0
1.5
1.0
0.5
Vsum, V
200
100
0
VN, V
200
100
0
0
100
200
300
400
500
time, nsec

# Chapter 5: Linear Data File Components

# DataAccessComponent (Data Access Component)

## Symbol



## Parameters

| Name | Description |
|---|---|
| File | filename |
| Type | file type |
| Block | block name |
| InterpMode | interpolation mode:<br>**Index Lookup** Specifies that iVal *n* represents the integer indices (beginning with 0) of the independent variables in the data file. Real iVal values are truncated first for index lookup.<br>**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.<br>**Ceiling Value Lookup** For a real independent variable, accesses the nearest point in the data file not less than the specified value.<br>**Floor Value Lookup** For a real independent variable, accesses the nearest point in the data file not greater than the specified value.<br>**Linear, Cubic, Cubic Spline** Specifies the interpolation mode in each dimension (except for splines, where only the innermost variable is spline-interpolated).<br>**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)}. |
| InterpDom | interpolation domain:<br>**Rectangular** Interpolates real and imaginary parts separately; recommended for emittances.<br>**Polar** (arc interpolation) Interpolates magnitude and angle separately; recommended for S-parameters.<br>**DB** Interpolates in dB and angle format.<br>**Value** This is provided if the interpolation domain is a variable or unknown; for example as a passed parameter of a subnetwork. The resulting value should be a string (or integer) from the following set: {“ri” (0), “ma” (1), “db” (3) }. |

| **Name** | **Description** |
|---|---|
| iVar1, ... , iVar10 | independent variable name or cardinality (1: outermost) |
| iVal1, ... , iVal10 | independent variable value or index (0: first/starting index) |

## Notes/Equations

1. This component can be used to extract/interpolate multidimensional dependent variables as a function of up to 10 independent variables. By setting the DAC *File* parameter to the desired filename, and setting the parameter of the component of interest to point to the DAC (by Instance ID), the data in the specified file can be accessed. (Refer to "Example 1" on page 5-5)

   You can quickly set all parameters (with matching names) of a device model by setting the model's *AllParams* parameter to the DAC's Instance ID, which in turn, references the data file. Parameter names in a data file that are not device model parameters are ignored. A device model parameter value that is explicitly specified will override the value set by an *AllParams* association. (Refer to "Example 2" on page 5-6)

   You can also sweep over several BJT models using two DAC components. (Refer to "Example 3" on page 5-7)

   S-parameter data can be read directly from a Touchstone file using a DAC. (Refer to "Example 4" on page 5-8).

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

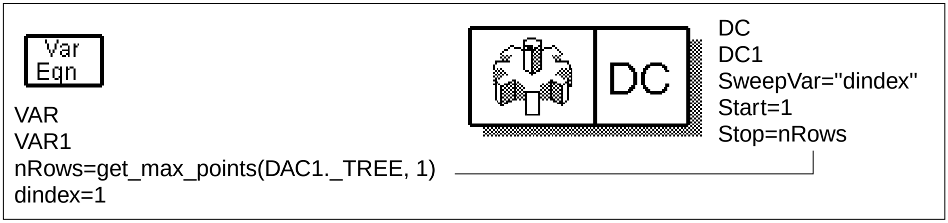
3. This component is actually a special subnetwork whose expressions can be used outside. In particular, one of these expressions is *_TREE* (the multi-dimensional table). The following example shows using this expression with the *get_max_points* function.

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

   where:

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

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



VAR
VAR1
nRows=get_max_points(DAC1._TREE, 1)
dindex=1

DC
DC1
SweepVar="dindex"
Start=1
Stop=nRows

4. The *Type* parameter specifies the format of the disk file, which includes Touchstone, CITIfile, several MDIF types, SPW and binary datasets (possibly from a previous simulation or via instrument server).

   The files displayed in the Browser represent all files found based on the search paths specified by the DATA_FILES configuration variable.

   For information on data file formats, refer to Chapter 7 *Working with Data Files* in the *Circuit Simulation* manual.

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

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

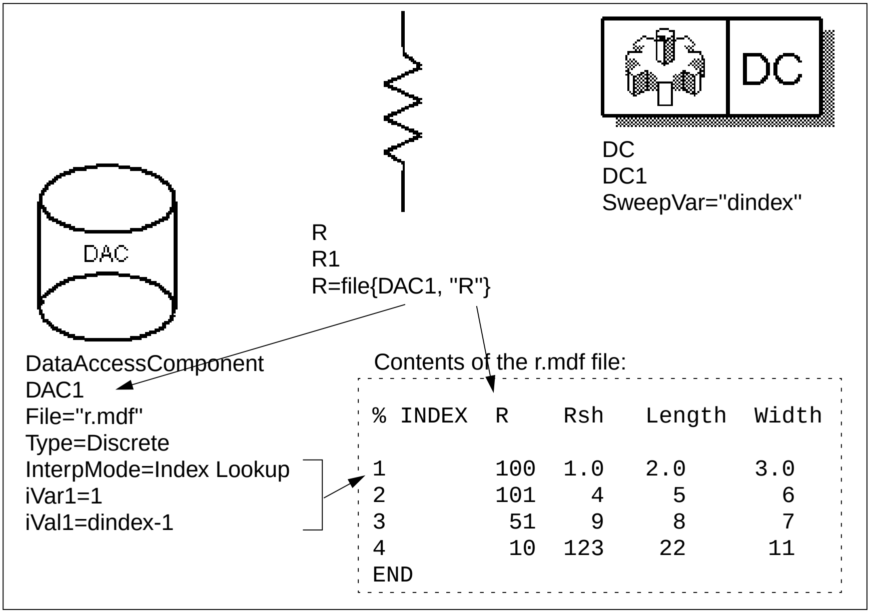
   When the DataAccessComponent refers to a time-domain MDIF file which has a *.tim* extension, the iVar*x* parameter must be set to *time* and the reference to the dependent parameter must be set to *voltage*, independent of the names of the columns in the *.tim* file.

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

7. Each *iVal* is normally a real or integer value of the independent variable to bracket or search for in the file. If InterpMode = Index Lookup (which must be the case for implicit variables), this value is the integer index, starting from 0. For example, the row value for a discrete file block runs from 0 to #rows – 1.
8. For all value lookup modes, a tolerance of 0.01% is used. A warning message is issued when extrapolation occurs.

### Example 1

In this example, the resistance of R1 is stepped through all values under the *R* column in the "r.mdf" file

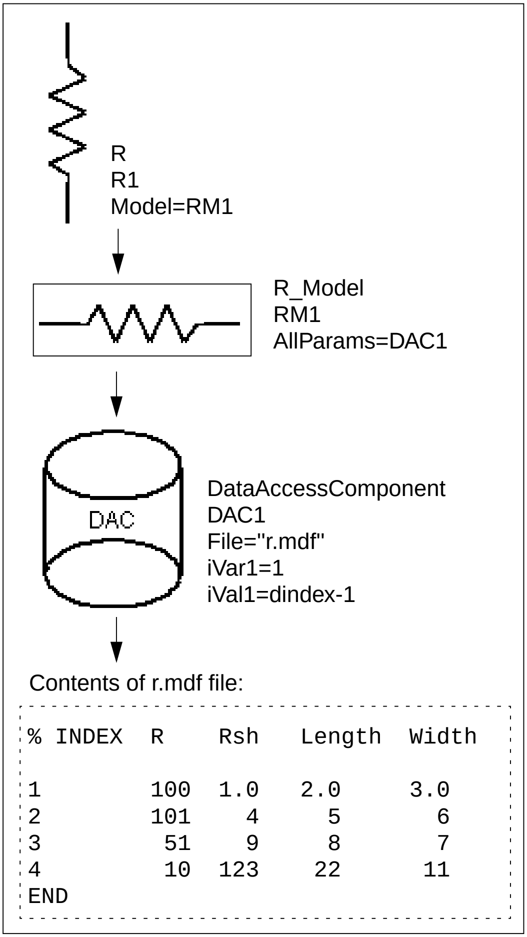


```
% INDEX  R    Rsh   Length  Width

1        100  1.0   2.0     3.0
2        101    4     5       6
3         51    9     8       7
4         10  123    22      11
END
```

### Example 2

In this example, resistor model RM1 accesses the *R*, *Rsh*, *Length* and *Width* parameters from the discrete "r.mdf" file.



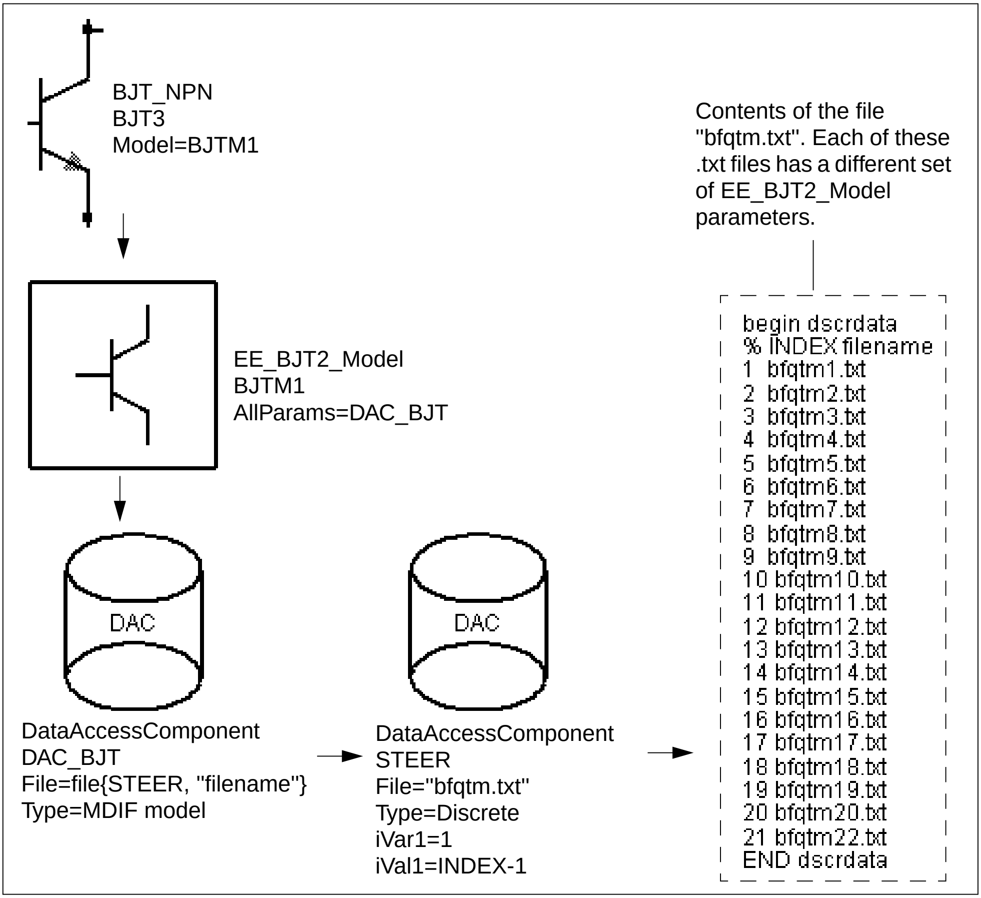
Contents of r.mdf file:

```
% 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
```

### Example 3

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



---

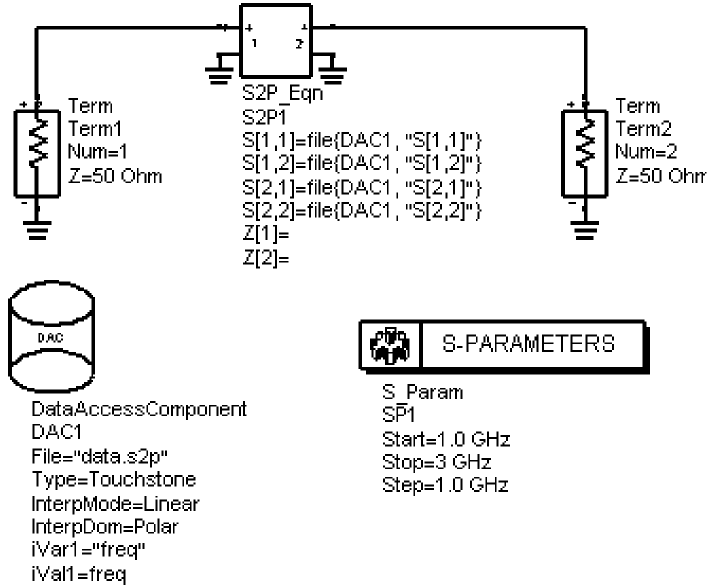
**Note** An assignment of the type: R1=file{DAC1, "Rnom"}, is equivalent to the expression `R1=dep_data(DAC1._DAC, "Rnom"`).

---

## Example 4

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

```
# hz S ma R 50
! 2 Port Network Data from SP1.SP block
    1e+009   0.85   -32   0.53   58  0.53  58  0.85  -32
    2e+009   0.80   -35   0.57   55  0.57  55  0.82  -35
    3e+009   0.72   -37   0.61   53  0.6   53  0.8   -37
```



Term
Term1
Num=1
Z=50 Ohm

S2P_Eqn
S2P1
S[1,1]=file{DAC1, "S[1,1]"}
S[1,2]=file{DAC1, "S[1,2]"}
S[2,1]=file{DAC1, "S[2,1]"}
S[2,2]=file{DAC1, "S[2,2]"}
Z[1]=
Z[2]=

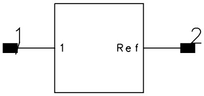
Term
Term2
Num=2
Z=50 Ohr

DataAccessComponent
DAC1
File="data.s2p"
Type=Touchstone
InterpMode=Linear
InterpDom=Polar
iVar1="freq"
iVal1=freq

S_Param
SP1
Start=1.0 GHz
Stop=3 GHz
Step=1.0 GHz

# 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

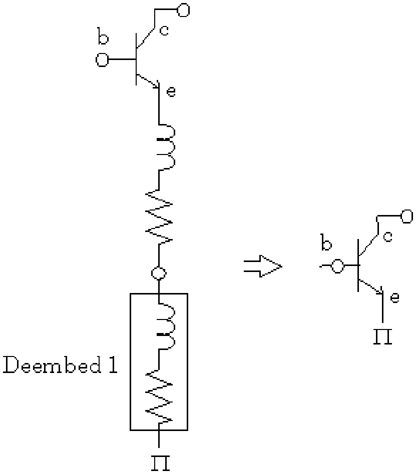
Temp = physical temperature, in °C

## Range of Usage

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

## Notes/Equations

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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. In the following example, the BJT emitter's parasitics leads are de-embedded to obtain just the chip BJT.



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.

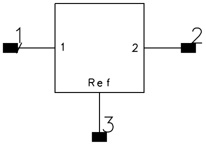
   DB: transfer to (dB, angle) before interpolation

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

5. This component does not generate any noise.
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.

# 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

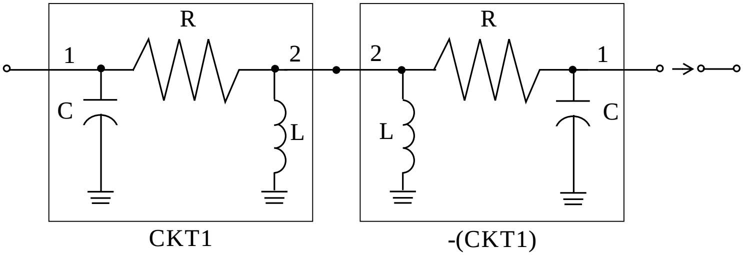
Temp = physical temperature, in °C

## Range of Usage

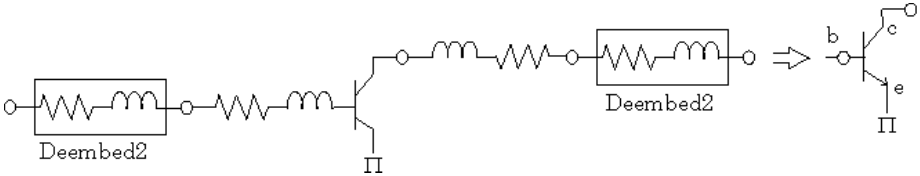
Within the frequency range of the S-, G-, H-, Y-, or Z-parameter file

## Notes/Equations

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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. When this component is connected in series with the sub-circuit being negated, and the hookup is back-to-back as shown in the following illustration, the result is a short. When this component is connected in parallel, with the subcircuit being negated, the result is an open.



Another example is to de-embed the parasitics of the leads in a transistor. In the following illustration the base and collector parasitics are de-embedded to give just the chip BJT.



4. The 2-port S-parameters are assumed to be measured with pin 1 as the input, pin 2 as the output, and pin 3 as the common terminal.
5. 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. This component does not generate any noise.
7. 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.
8. This component has no default artwork associated with it.

## NetlistInclude (Netlist File Include Component)

### Symbol

### Parameters

IncludePath = Space-delimited search path for included files

IncludeFiles = List of files to include

UsePreprocessor = *yes* (default) to use an *#include* directive; *no* to copy the full text of the file

### Notes/Equations

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

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

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

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

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

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

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

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

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

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

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

5. Example component parameters:

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

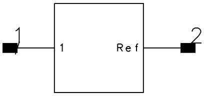
A NetlistInclude component with these parameters would generate the following netlist output:

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

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

# S1P (1-Port S-parameter File)

## Symbol



## 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

## Notes/Equations

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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

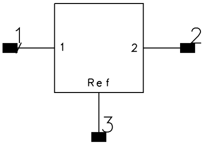
Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

3. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
4. If the component temperature Temp is less than -273ºC, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.
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.
6. Ref pin 2 is the common terminal; it is normally grounded, but can be used in non-grounded mode.
7. For time-domain analysis, the frequency-domain S-parameters are used.
8. This component has no default artwork associated with it.

## S2P (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

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

### Notes/Equations

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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 (Ref) as the common terminal.

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

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

   Rectangular: transform to (real, imag) before interpolation

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

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

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

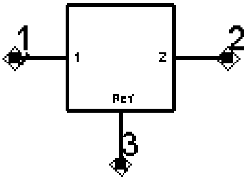
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

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

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

## 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

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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 (Ref) as the common terminal.

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

Rectangular: transform to (real, imag) before interpolation

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

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

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

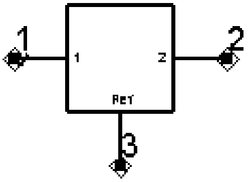
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

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

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

## S2P_Pad3 (2-Port S-parameter File; pad artwork)

### Symbol



### Parameters

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

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

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

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

Temp = physical temperature, in °C

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

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

S = (ADS Layout option) spacing (length from pin1 to pin 2, in length units

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

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

Side = top, bottom

### Range of Usage

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

### Notes/Equations

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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 (Ref) as the common terminal.

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

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

   Rectangular: transform to (real, imag) before interpolation

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

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

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

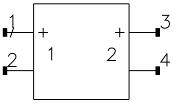
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

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

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

## S2PMDIF (Multi-Dimensional 2-Port S-parameter File)

### Symbol



### Parameters

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

Type = file type: S2PMDIF, Touchstone, Dataset, 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

### Notes/Equations

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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

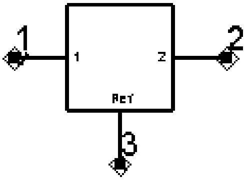
Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

3. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
4. If the component temperature Temp is less than -273ºC, then the component does not generate any noise. If the file contains the noisy two-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn), these parameters are used to calculate the devices noise performance, independent of Temp. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.
5. 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.
6. For time-domain analysis, the frequency-domain S-parameters are used.
7. 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.
8. This component has no default artwork associated with it.

## 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

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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 (Ref) as the common terminal.

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

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

   Rectangular: transform to (real, imag) before interpolation

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

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

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

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

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

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

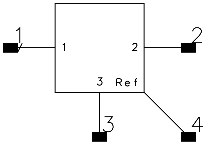
9. For transient analysis, pins 1 and 2 are shorted together.

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

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

## 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

### Notes/Equations

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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

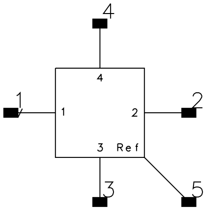
Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

3. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
4. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.
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.
6. Ref pin 4 is the common terminal; it is normally grounded, but can be used in non-grounded mode.
7. For time-domain analysis, the frequency-domain S-parameters are used.
8. This component has no default artwork associated with it.

# 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, 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

## Notes/Equations

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.

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

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.

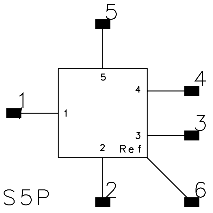
6. Ref pin 5 is the common terminal; it is normally grounded, but can be used in non-grounded mode.

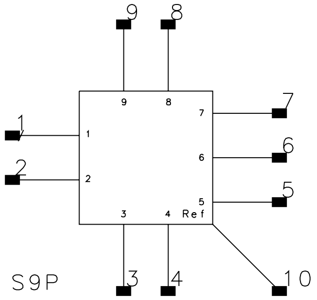
7. For time-domain analysis, the frequency-domain S-parameters are used.

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

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

## Symbol





## 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

## Notes/Equations

1. For information on data file formats, refer to Chapter 7 *Working with Data Files* in the *Circuit Simulation* manual.
2. The number of terminals increases sequentially from 5 to 9, and is equal to the number of ports of the component.

   Ref is the common terminal; it is normally grounded, but can be used in non-grounded mode.
3. If no extension is supplied with the file name, then a default value of ".s(#)p" is used, where (#) is the number of ports of the component.
4. 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.

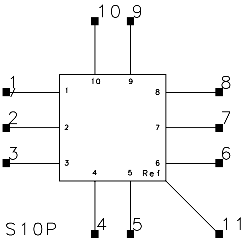
   DB: transfer to (dB, angle) before interpolation

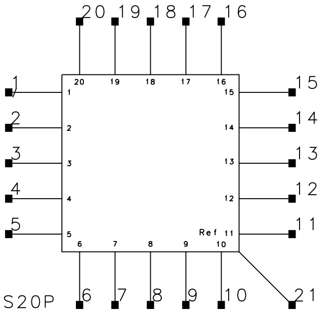
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
5. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
6. If the component temperature Temp is less than -273$^o$C, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.
7. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous). Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).

   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.
8. For time-domain analysis, the frequency-domain S-parameters are used.
9. This component has no default artwork associated with it.

## S10P to S20P (10-Port to 20-Port S-parameter File)

### Symbol





### 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

## Notes/Equations

1. The items S11P through S99P cannot be selected from the component palette. They are accessed by typing the appropriate name (such as S12P or S98P) into the right entry box (directly above the viewing area), pressing Enter, then moving the cursor to the viewing area to place the item.
2. The number of terminals increases sequentially from 10 to 20, and is equal to the number of ports of the component.

   Ref is the common terminal; it is normally grounded, but can be used in non-grounded mode.
3. For information on data file formats, refer to Chapter 7 *Working with Data Files* in 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

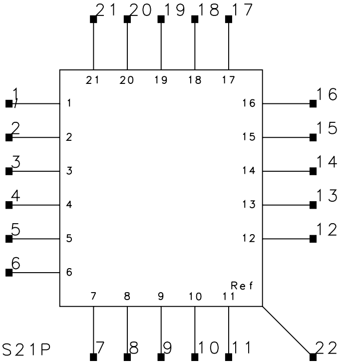
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
6. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
7. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.
8. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous). Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).

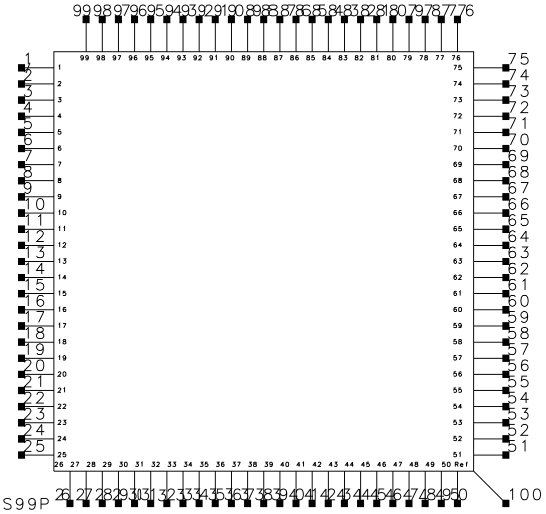
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

9. For time-domain analysis, the frequency-domain S-parameters are used.
10. This component has no default artwork associated with it.

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

## Symbol





## 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

## Notes/Equations

1. S11P through S99P support up to 99-port networks. They cannot be selected from the component palette; they are accessed by typing the appropriate name (such as S11P or S99P) into the field above the viewing area, pressing Enter, then moving the cursor to the viewing area to place the item.
2. Ref is the common terminal; it is normally grounded, but can be used in non-grounded mode.
3. The S, Y, Z, and N matrix measurements are allowed for up to 99-port networks. In addition, single measurements are applicable:
    - SIJ, for example (S(29,28))
    - VSWR, for example (S(29,29))
4. These components primarily support electromagnetic simulation results of circuits with a large number of ports, such as antenna feed networks.
5. 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. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
7. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.

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

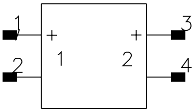
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.

9. For time-domain analysis, the frequency-domain S-parameters are used.
10. This component has no default artwork associated with it.

# Chapter 6: Equation-Based Linear Components

## Chain (2-Port User-Defined Linear Chain)

### Symbol



### 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

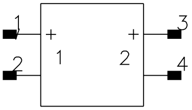
### Notes/Equations

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

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

# Hybrid (2-Port User-Defined Linear Hybrid)

## Symbol



## Parameters

H11 = input impedance (v1/i1 with v2=0)

H12 = reverse voltage gain (v1/v2 with i1=0)

H21 = forward current gain (i2/i1 with v2=0)

H22 = output conductance (i2/v2 with i1=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

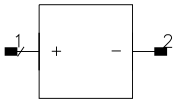
ImpAbsTol = absolute impulse response truncation factor

## Notes/Equations

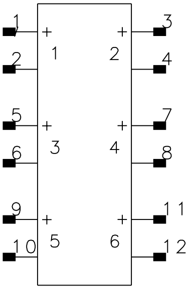
1. Port polarity is indicated by a minus sign (-) and a plus sign (+) on each port.
2. Any H-parameter that is not defined initially is set to a default value of 0 and cannot be modified later. Any H-parameter that is defined initially, even if it is set to 0, can be modified and swept. It can also be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

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

## Symbol



S1P_Eqn



S6P_Eqn

## Parameters

S[i, j] = S-parameter in real and imaginary forma

Z[i] = port i reference impedance, in ohms

Recip = port is reciprocal: NO, YES

NFmin = minimum noise figure, in dB

Rn = noise resistance, in ohms

Sopt = optimum noise match

Temp = device noise temperature, in °C

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

ImpMode = convolution mode (value type: integer)

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

$1 \leq i, j \leq$ port number

## Notes/Equations

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

   If a value is not entered for S[i,j], it is set to a zero default value (0, 0) and cannot be modified later. To enter a value for S[i,j], use the syntax a+j*b or 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 NFmin, Sopt, and Rn are used to characterize noise in S2P_Eqn, the following relation must be satisfied for a realistic model.

$$\frac{Rn}{Zo} \geq \frac{To(Fmin-1)\left|1+Sopt\right|^2}{T\ 4} \frac{(1-\left|S_{11}\right|^2)}{\left|1-Sopt\ S_{11}\right|^2}$$

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

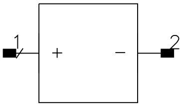
3. If the component temperature Temp is < -273°C, the component does not generate any noise. For S2P_Eqn only, if noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn) are specified, these parameters are used to calculate the device's noise performance, independent of Temp. If the S-parameters describe a passive device, Temp and Twiss's theorem are used to calculate noise performance; if the S-parameters describe an active device, no noise is generated.

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

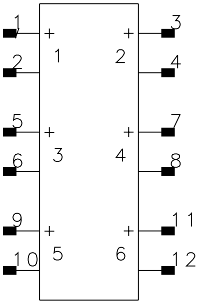
5. S[i,j] can be made dependent on frequency by using the global variable freq. For example, you can use a brick wall lowpass filter by using S21=if(freq<1 GHz), then 1 else 0.

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

### Symbol



Y1P_Eqn



Y6P_Eqn

### 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)

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

$1 \leq i, j \leq$ port number

### Notes/Equations

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*b or complex(a,b).

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).

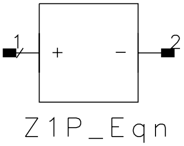
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).

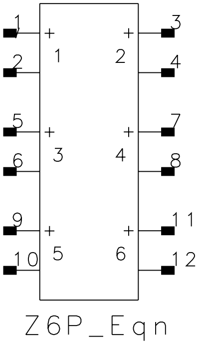
   If ImpMode, ImpMaxFreq, or ImpMaxOrder are not specified, they default to the global ImpMode specified by the transient analysis controller item.

3. Y[i,j] can be made dependent on frequency by using the global variable freq.

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

## Symbol





## Parameters

Z[i,j] = Z-parameter magnitude and phase

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

Recip = port is reciprocal: NO, YES

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

ImpMode = convolution mode (value type: integer)

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

$1 \leq i, j \leq$ port number

## Notes/Equations

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.

   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*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.

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.

4. Z[i,j] can be made dependent on frequency by using the global variable freq.