Agilent Technologies

Advanced Design System 2011.01

Feburary 2011
Introduction to Circuit Components

Acknowledgments

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

Circuit Components

Introduction

The Circuit Components catalog provides component information. Sections in this document are organized by component types; components are arranged alphabetically within each section.
This section provides information for these common items:

- *BinModel* (ccsim) component for automatic model selection
- *Ground (Ground Component)* (ccsim)
- *Term (Port Impedance for S-parameters)* (ccsim)
- *Drawing Formats* (ccsim) (design sheets)
- *Multiplicity parameter _M* (ccsim) to scale components or entire sub-circuits containing multiple components and sub-circuits
- *Series IV and MDS components* (ccsim) that can be used in ADS

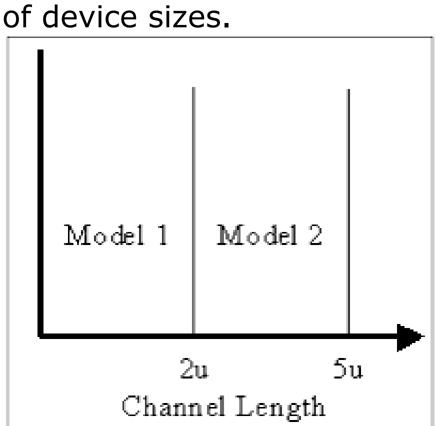
Bin Model (Bin Model for Automatic Model Selection)

Symbol

Parameters

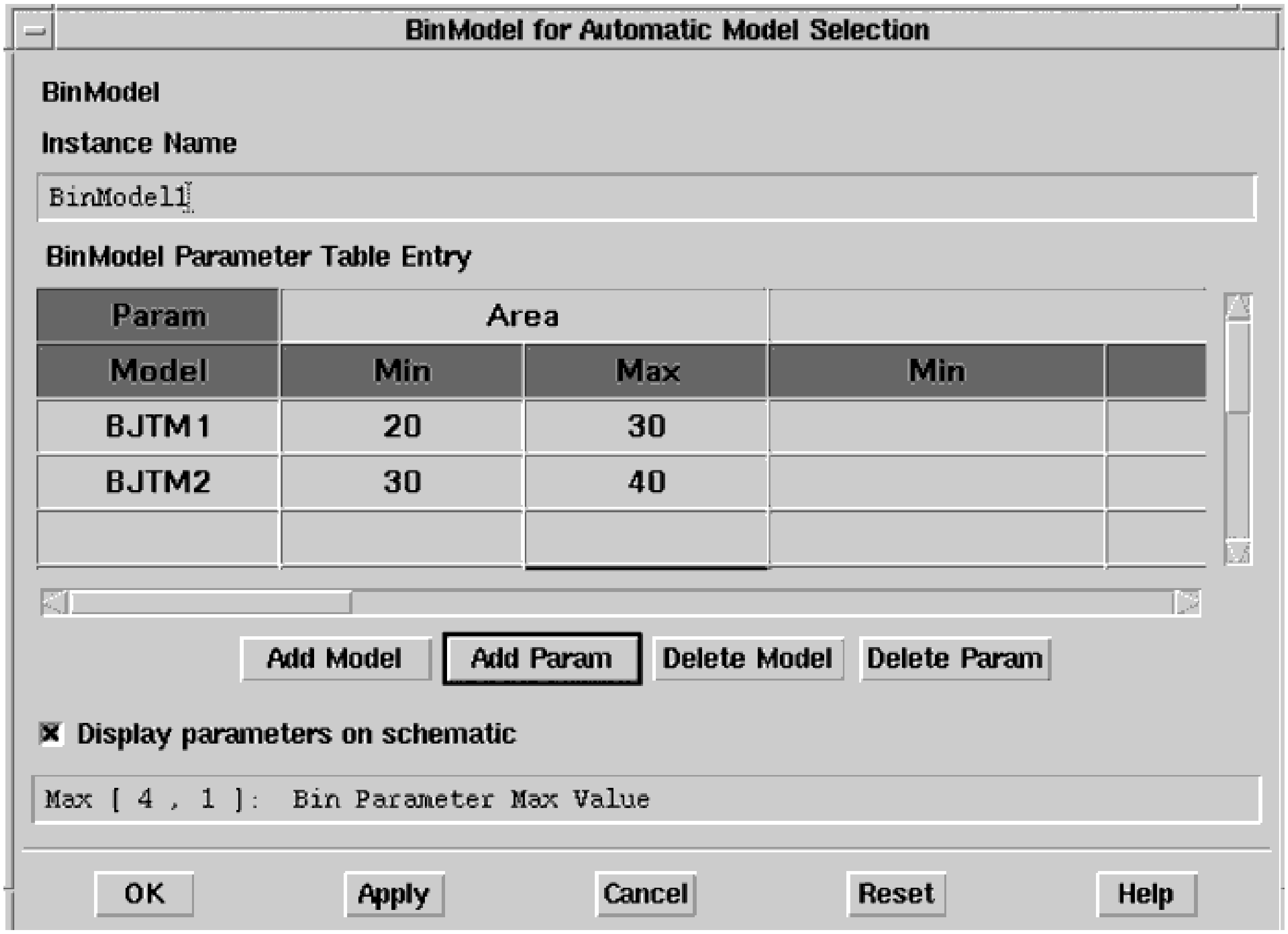
None

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.



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 Solution** 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 Solution** 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

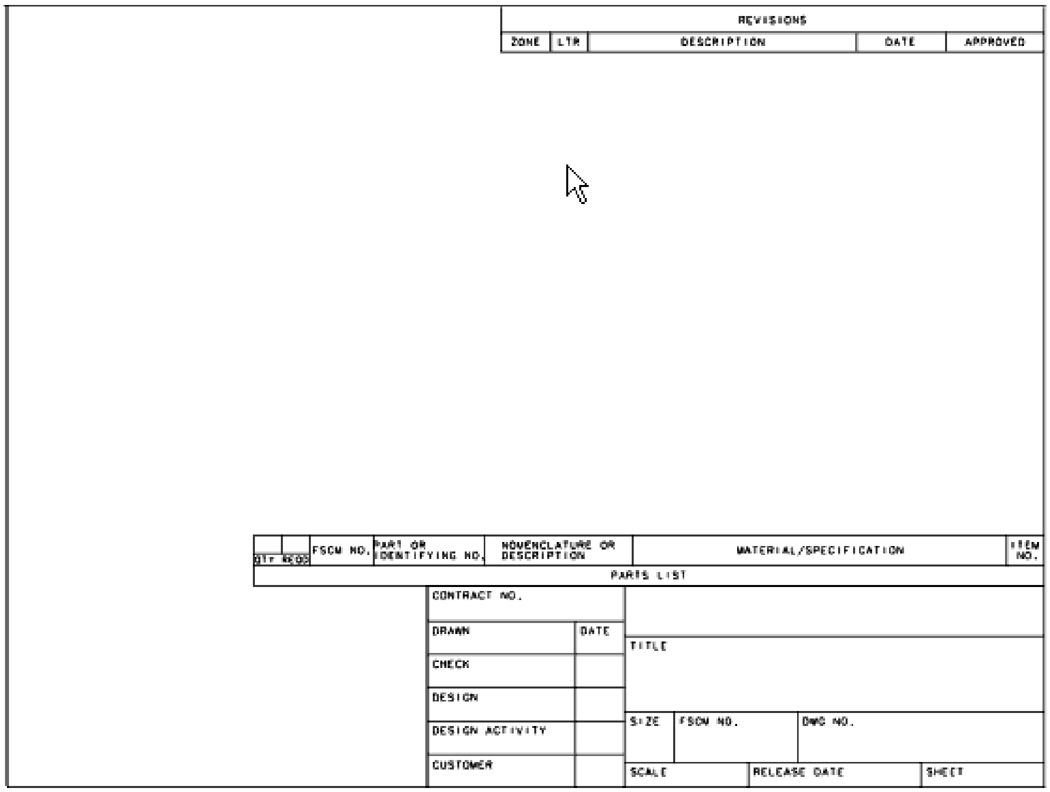
10. The circuit was simulated to perform parameter sweep over Area from 25 to 55 with steps of 10.
11. The four results were then compared in the data display window.

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



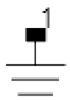
FORMATA

Notes/Equations

1. The Drawing Formats library provides popular sheet sizes (in inches): A (8.511), B (1117), C (1722), D (2234), E (3444).
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

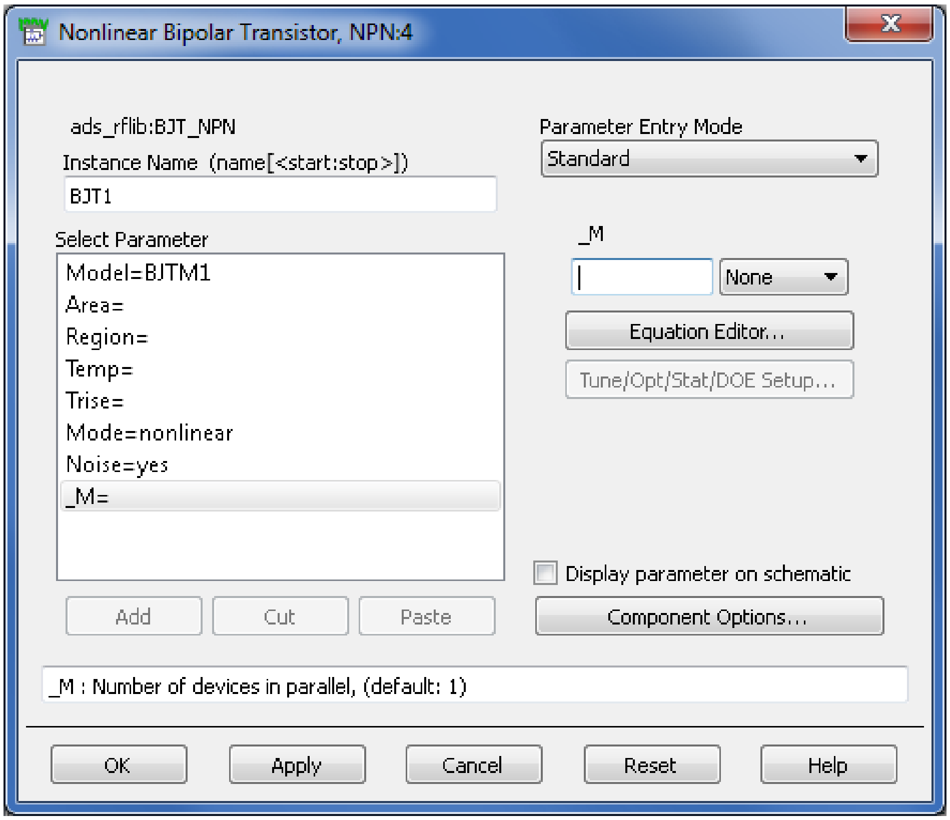
Notes/Equations

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

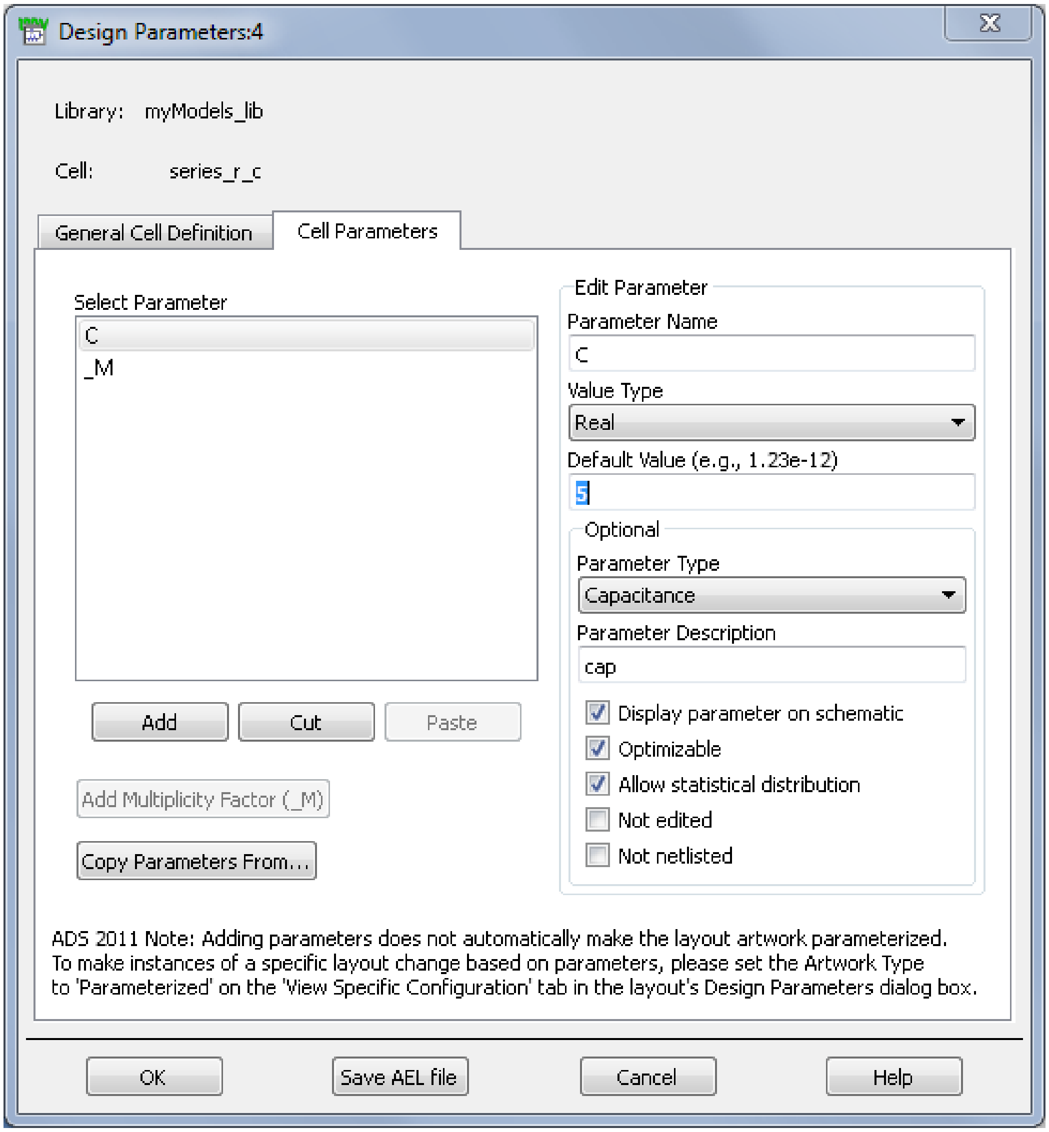
Multiplicity Parameter _M

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

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



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



Note
The *Add Multiplicity Factor_M* button is disabled if the *Multiplicity Factor* is in already in the parameter list. Otherwise, it is enabled.

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

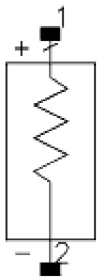
- CPWTL_MDS
- GCPWTL_MDS
- CPWCTL_MDS
- ACPW_MDS
- ACPWTL_MDS
- CPWTLFG_MDS
- MSACTL_MDS
- MS3CTL_MDS, MS4CTL_MDS, MS5CTL_MDS
- MSABND_MDS
- MSBEND_MDS
- MSOBND_MDS
- MSCRNR_MDS
- MSTRL2_MDS
- MSCTL_MDS
- MSCROSS_MDS
- MSRBND_MDS
- MSGAP_MDS
- MSAGAP_MDS

- 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

Term (Port Impedance for S-parameters)

Symbol



Parameters

Name	Description	Units	Default
Num	Port number	Integer	1
Z	Reference impedance, use 1+j*0 for complex	Ohm	50
Noise	Enable/disable port thermal noise: yes, no (for AC or harmonic balance analysis only; not for S-parameter analysis)	None	yes
Vdc	Open circuit DC voltage		None
Temp	Temperature	°C	None

Notes/Equations

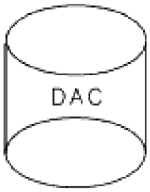
1. Term can be used in all simulations. For S-parameter simulations it is used to define the impedance and location of the ports. When not in use, it is treated as an impedance with the value R + JX. The reactance is ignored for dc simulations.

Data File Components

- *DataAccessComponent (Data Access Component)* (ccsim)
- *De_Embed and De_EmbedSnP (2-Port to 12-Port De-Embed Data File)* (ccsim)
- *Deembed1 (1-Port De-Embed Data File)* (ccsim)
- *Deembed2 (2-Port De-Embed File)* (ccsim)
- *Deembed4 (4-Port De-Embed Data File)* (ccsim)
- *Deembed6 (6-Port De-Embed Data File)* (ccsim)
- *Deembed8 (8-Port De-Embed Data File)* (ccsim)
- *Deembed12 (12-Port De-Embed Data File)* (ccsim)
- *NetlistInclude (Netlist File Include Component)* (ccsim)
- *S1P (1-Port S-parameter File)* (ccsim)
- *S2P (2-Port S-parameter File)* (ccsim)
- *S2P Conn (2-Port S-parameter File connector artwork)* (ccsim)
- *S2PMDIF (Multi-Dimensional 2-Port S-parameter File)* (ccsim)
- *S2P Pad3 (2-Port S-parameter File pad artwork)* (ccsim)
- *S2P Spac (2-Port S-parameter File)* (ccsim)
- *S3P (3-Port S-parameter File)* (ccsim)
- *S4P (4-Port S-parameter File)* (ccsim)
- *S5P to S9P (5-Port to 9-Port S-parameter File)* (ccsim)
- *S10P to S20P (10-Port to 20-Port S-parameter File)* (ccsim)
- *S21P to S99P (21-Port to 99-Port S-parameter File)* (ccsim)
- *SnP component (n99)* (ccsim)
- *SnP Diff component* (ccsim)
- *VAR (Variables and Equations Component)* (ccsim)
- *W_Element (Multi-Conductor Transmission Lines)* (ccsim)
- *XnP Components (X1P - X10P)* (ccsim)

DataAccessComponent (Data Access Component)

Symbol

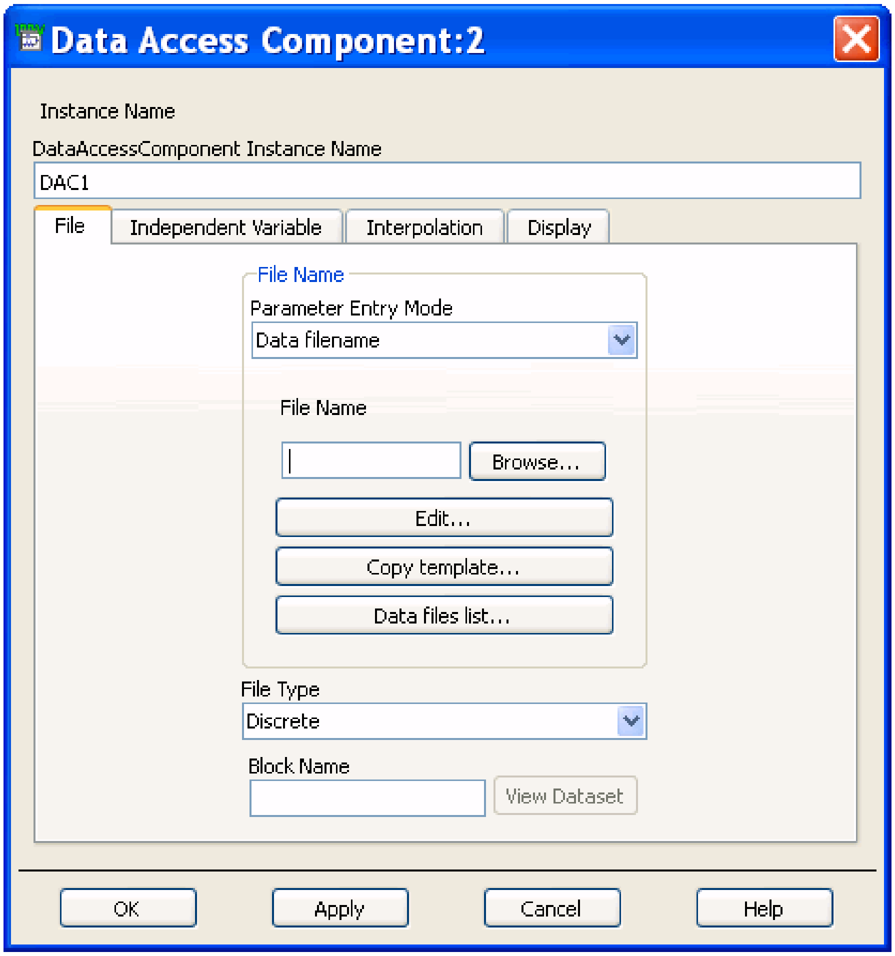


The DataAccessComponent contains the following parameters types:

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

Setting the File Parameters (ADS)

Use the *File* tab to specify file related parameters.

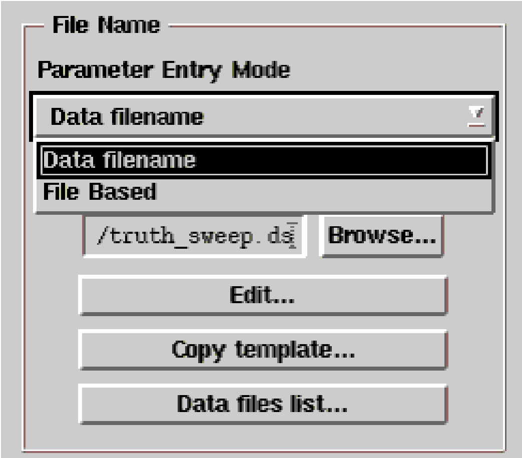


Parameters

Name	Description	Units	Default
File Name	File name	None	None
File Type	Discrete, model MDIF, S2PMDIF, Dataset, Touchstone, CITIfile, P2D MDIF, S2D MDIF, IMT MDIF, GCOMP MDIF, FIR MDIF, SPW, TIM MDIF, SPE MDIF, SDF, Generic MDIF, COD MDIF, LAS MDIF, Value	None	Discrete
Block Name	Block name	None	None

File Name

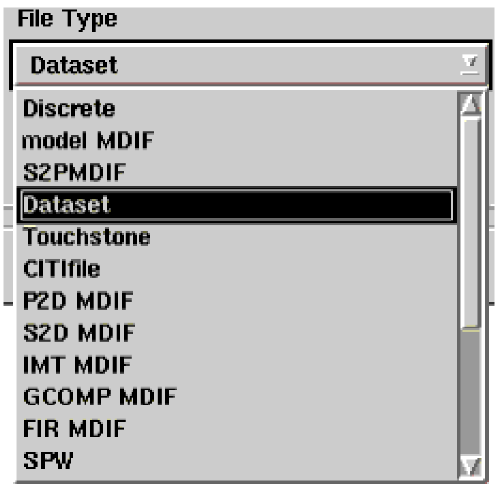
The file name can be defined in either *Data filename* entry mode or *File Based* entry mode. When *Data filename* entry mode is selected, file name can be given as a string which represents explicit file name or assigned through a variable as *@variable_name* .



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

File Type

Provides a drop down list of available file types.



Block

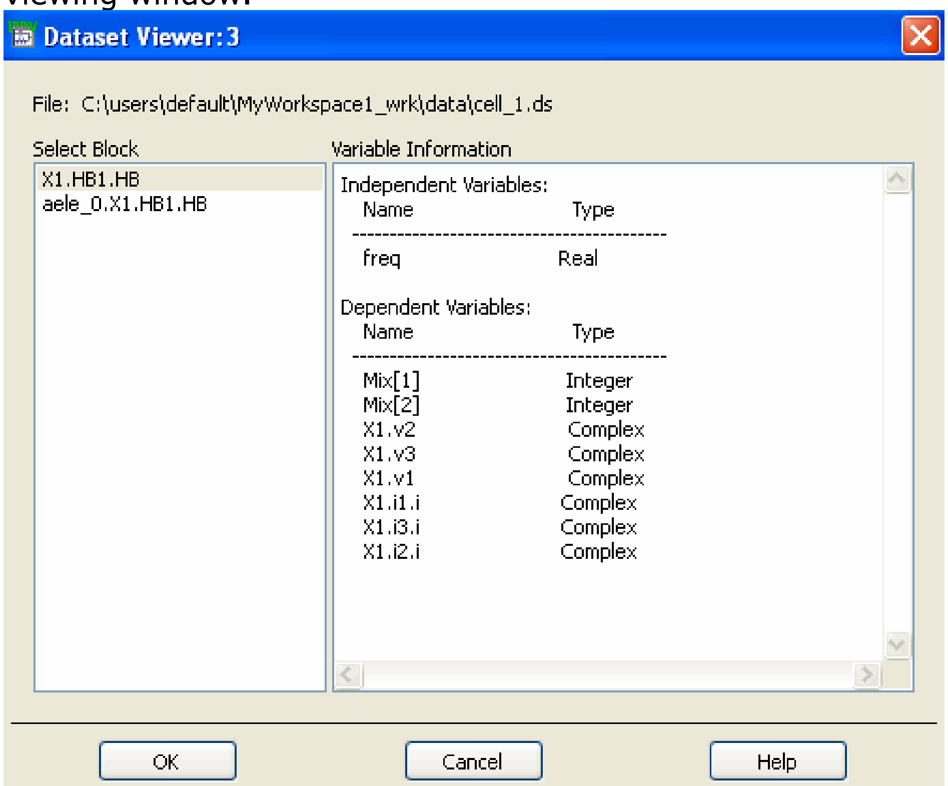
If you do not specify a block name, the block of data contained in the dataset file is used. The Block parameter is required when the generic MDIF file has multiple blocks.
Here is an example of a generic MDIF file with multiple blocks. A DAC which references this file should have Block set to "VB" or "IQ". Note that the outer independent variables Vc1 and Vb1 will only apply to block VB; they will have no relation to block IQ.

```
VAR Vc1(1) = 0.2
VAR Vb1(1) = 0.3
BEGIN VB
% Index_a(1) Vbase(1)
0 0.3
1 0.4
```

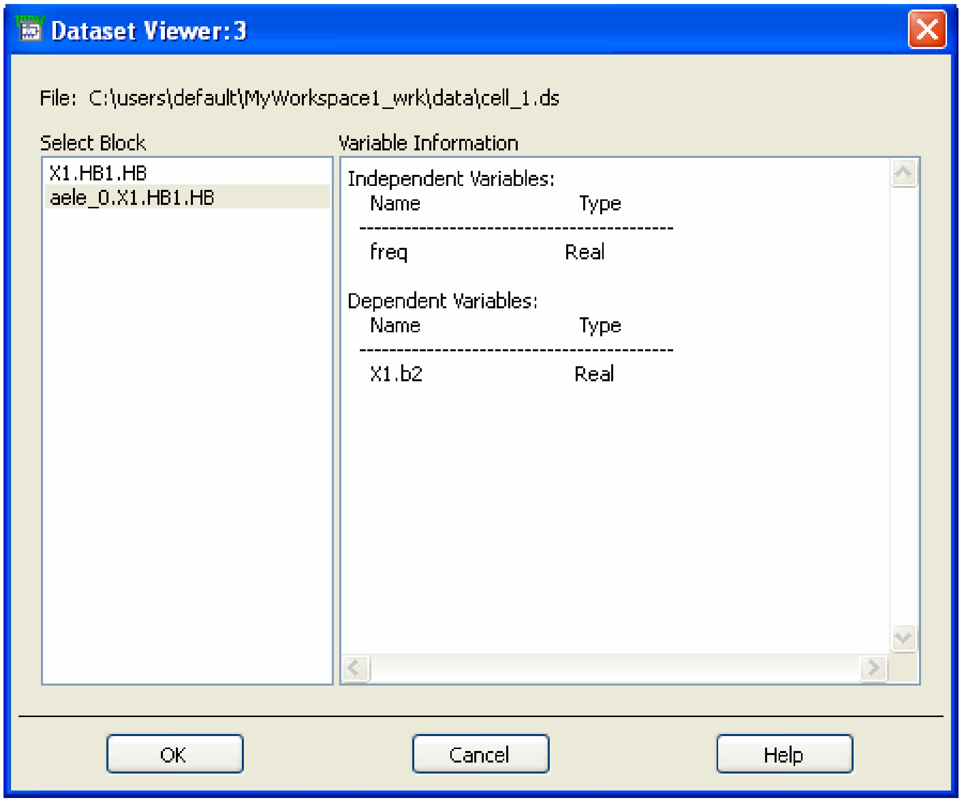
```
END
BEGIN IQ
% Index_b(1) Ib_A(1) Ic_A(1)
0 1.0E-005 0.002
1 2.0E-005 0.003
END
```

DataSet Viewer

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

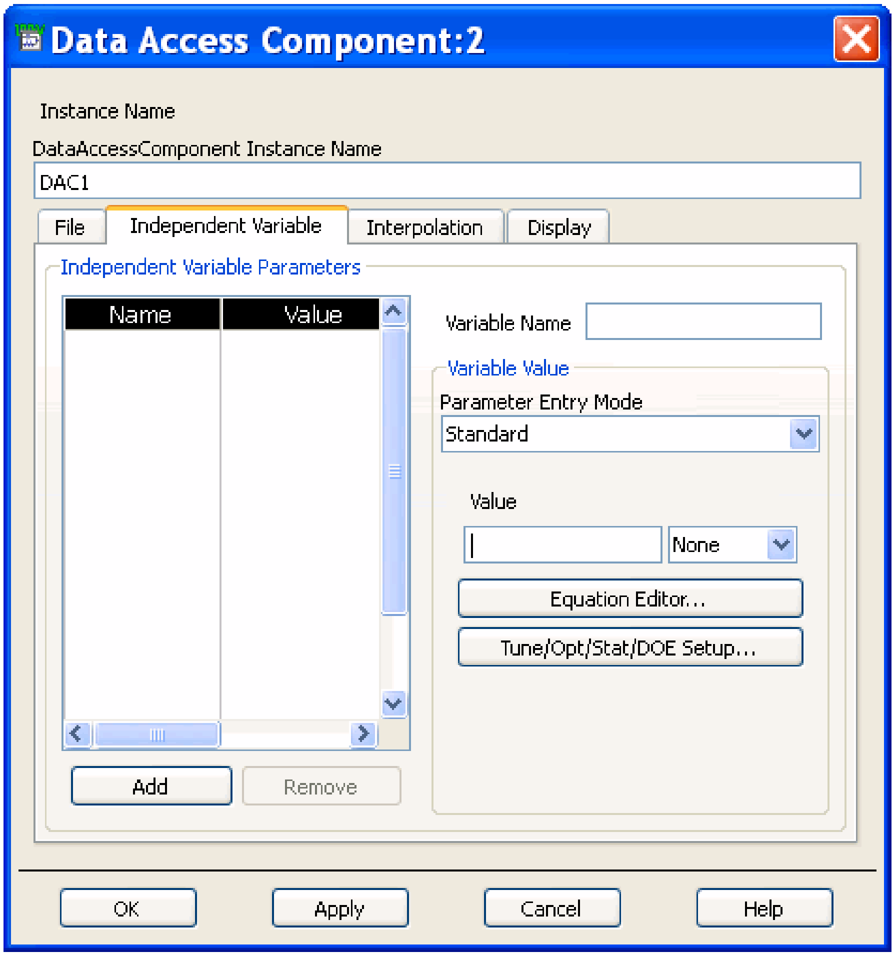


When no data block is specified, the Dataset Viewer defaults to the first data *block* in the file. If there is more than one block of data, you can click another *block* to select it and view its variable information.



Setting Independent Variable Parameters (ADS)

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



Parameters

Name	Description	Units	Default
iVar1, ... , iVar10	Independent variable name or cardinality (1: outermost)	None	None
iVal1, ... , iVal10	Independent variable value or index (0: first/starting index)	None	None

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

To add the name and value of an independent variable:

1. Type the variable name in *Variable Name* input field.
2. Give the variable value in *Variable Value* "Value" input field.
3. Click **Add** to insert the pairs to the left hand side spread sheet.

To remove an independent variable setup:

1. Select the row to be deleted from the spread sheet.
2. Click **Cut** to remove the selected row.

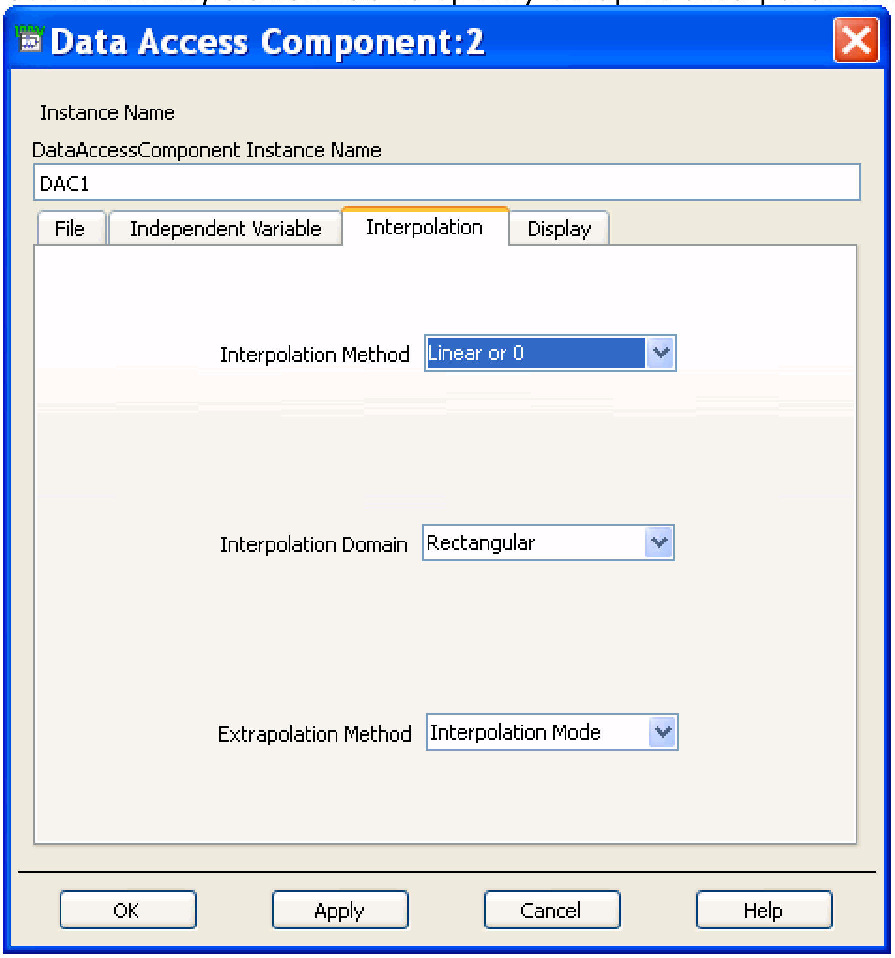
To edit an independent variable setup:

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

2. Modify the name or value on the right hand side input fields.

Setting Interpolation Parameters (ADS)

Use the *Interpolation* tab to specify setup related parameters.



Parameters

Name	Description	Units	Default
Interpolation Method (InterpMode)	Interpolation mode: **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. **Note** If you use the Index Lookup interpolation mode, you can access and/or interpolate the "indices" of the independent variable in the file. In this context the first data row of the file will have an index of 0, the second row will have an index of 1 and so on. **Value Lookup** For real/integer independent variable, accesses the point in the data file closest to the specified value. If midway, the average of the bracketing points is used. **Ceiling Value Lookup** For a real independent variable, accesses the nearest point in the data file not less than the specified value. **Floor Value Lookup** For a real independent variable, accesses the nearest point in the data file not greater than the specified value. **Linear, Cubic, Cubic Spline** Specifies the interpolation mode in each dimension (except for splines, where only the innermost variable is spline-interpolated). **Value** This is provided if the interpolation mode is variable or unknown, for example, as a passed parameter of a subnetwork. The resulting value should be a string (or integer) from the following set: `{"linear"(0), "spline"(1), "cubic"(2), "index_lookup"(3), "value_lookup"(4), "ceiling_value_lookup"(5), "floor_value_lookup"(6)}`	None	Index Lookup or 3
Interpolation Domain (InterpDom)	Interpolation domain: **Rectangular** Interpolates real and imaginary parts separately; recommended for emittances. **Polar** (arc interpolation) Interpolates magnitude and angle separately; recommended for S-parameters. **DB** Interpolates in dB and angle format. **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) }`	None	Rectangular
Extrapolation Method (ExtrapMode)	Extrapolation mode: **Interpolation Mode**: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation. **Constant Extrapolation**: when extrapolation occurs, no interpolation is performed. The value of the nearest data point is returned. An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.	None	Interpretation Mode

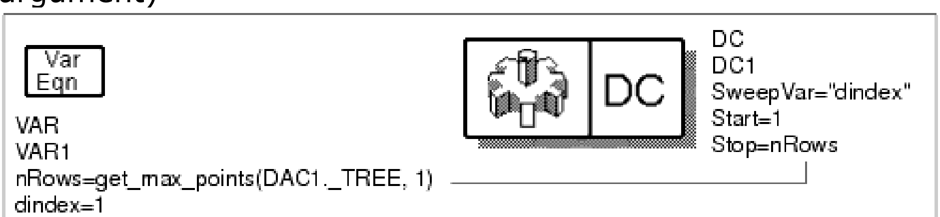
Notes/Equations

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)
    - 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)

- You can also sweep over several BJT models using two DAC components. (Refer to Example 3)
- S-parameter data can be read directly from a Touchstone file using a DAC. (Refer to Example 4).
- Discrete data can be read directly from a Discrete data type file using a DAC in an ADS Ptolemy simulation. (Refer to Example 5).
- Multi-dimensional data can be read directly from a Generalized Multi-dimensional data type file using a DAC in an ADS Ptolemy simulation. (Refer to Example 6).
- Ptolemy simulation examples using DAC are located in the *Controllers_wrk*; to access the example from the ADS Main window, click on **File** > **Open** > **Example** > *PtolemyDocExamples* > *Controllers_wrk* , then open the *Read_DAC_DSCR* and *Read_DAC_MDIF* designs.

2. For a complex dependent variable, the two parts (real/imag, mag/degree, or dB/degree) are interpolated separately. For arc-like data (for example S-parameters vs. frequency), it may be more appropriate to interpolate in the mag/degree domain.
3. This component is actually a special subnetwork whose expressions can be used outside. In particular, one of these expressions is *_TREE* (the multi-dimensional table). The following example shows using this expression with the *get_max_points* {{}} function.
   Example: `get_max_points(DAC1._TREE, "freq")`
   where:
   `DAC1._TREE` represents the Instance ID of the DAC
   {{"freq" }} represents the name of the independent variable
   It returns the maximum # of points (over all sweeps of that variable) of the independent variable (for discrete files with implicit row #, use 1 for the second argument)



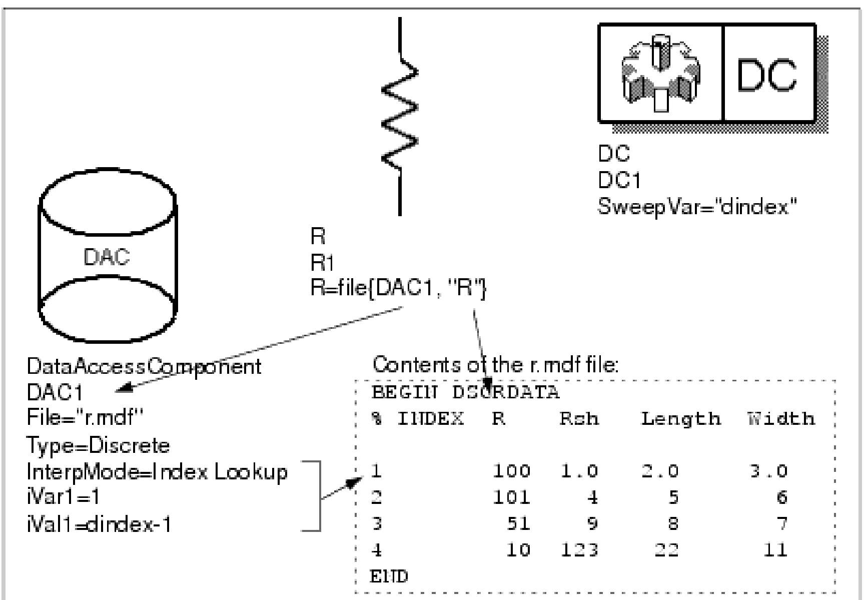
4. The *Type* parameter specifies the format of the disk file, which includes Touchstone, CITIfile, several MDIF types, SPW and binary datasets (possibly from a previous simulation or via instrument server).
   The files displayed in the Browser represent all files found based on the search paths specified by the DATA_FILES configuration variable.
   For information on data file formats, refer to *Working with Data Files* (cktsim).
5. The *Block* name specifies which table to use when the file contains two or more multidimensional tables, (e.g., "ACDATA", "NDATA" in an MDIF file, "HB1.HB", "HB1.HB_NOISE" in a harmonic balance analysis dataset). A unique prefix is sufficient; it can also be the sequence number (starting with 1) of the table, for example, 1 for an "ACDATA" table and 2 for "NDATA". Note that the " *at* " symbol (@) should be used to suppress quotes when using a variable to identify a table as the independent variable for making DAC parameter assignments.
6. Each *iVar* is either the name of an independent variable in the file (e.g., Vgs) or is an integer representing the cardinality or nesting order of the independent variable (1 being outermost). A cardinal value must be used when an independent variable is implicit; for example, row index in discrete files is the innermost independent variable. Note that @ must be used to suppress quotes when using a variable, for

example, *@freq1* , where *freq1* is a variable declared in a VAR item.
When the *DataAccessComponent* refers to a time-domain MDIF file which has a . *tim* extension, the iVar *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 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.
9. The *DataAccessComponent* doesn't work across hierarchies. When a DAC is used in a subnetwork, it must be located at the same level where it is referred, or a simulation will fail with path errors.

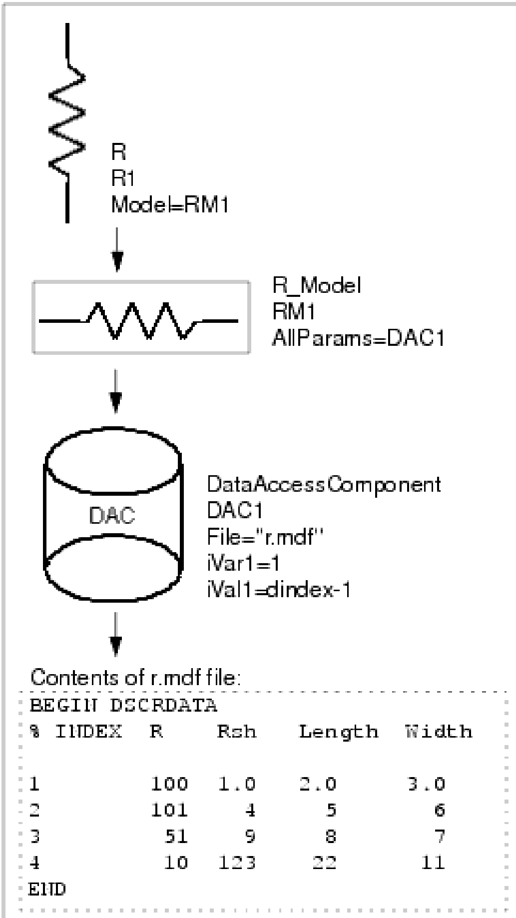
Example 1

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



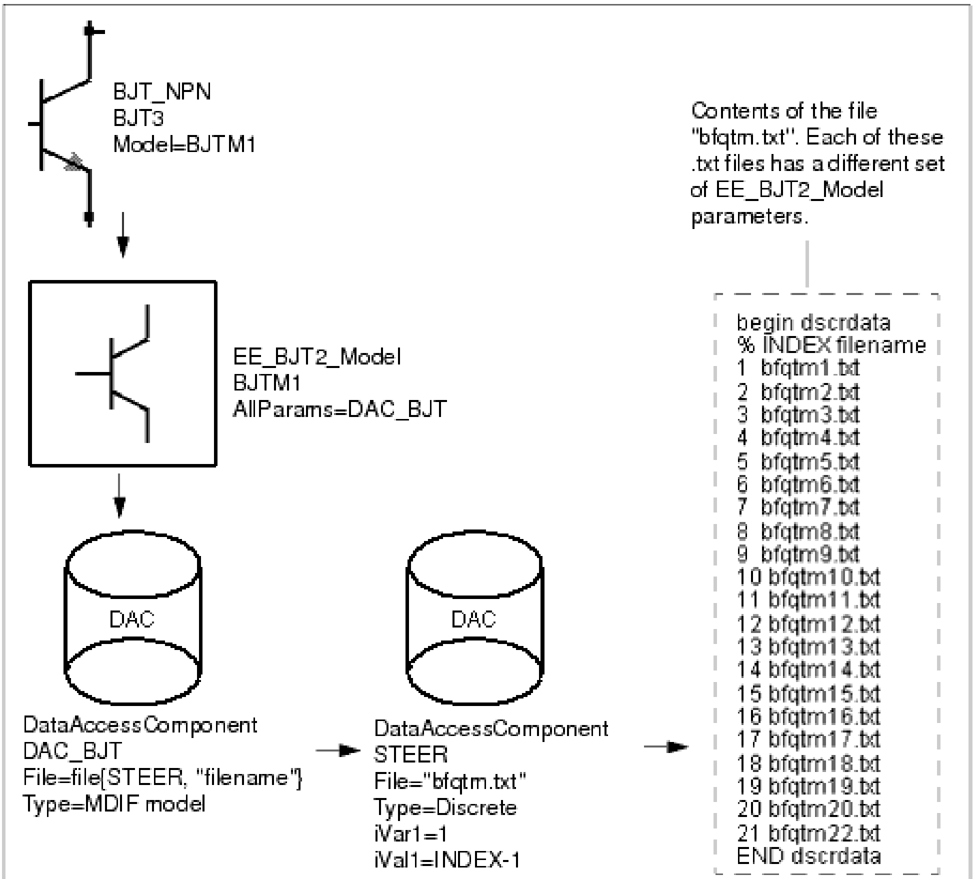
Example 2

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



Example 3

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

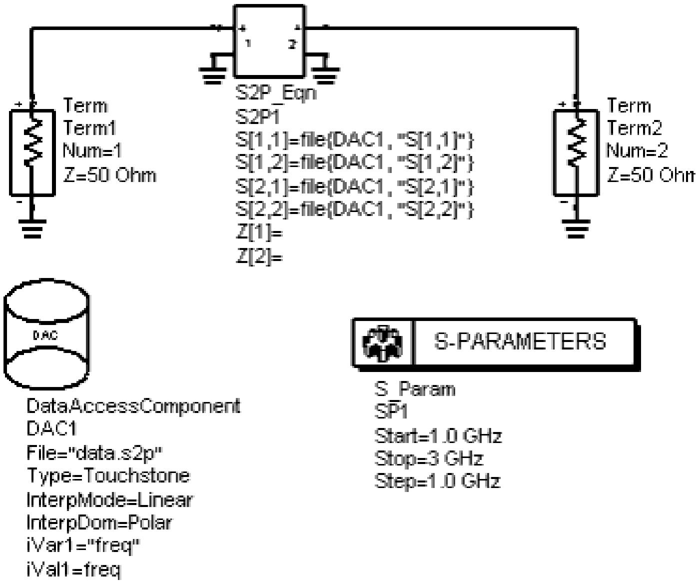


Note
An assignment of the type: R1=file{DAC1, "Rnom"}, is equivalent to the expression `R1=dep_data(DAC1._DAC, "Rnom")`.

Example 4

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

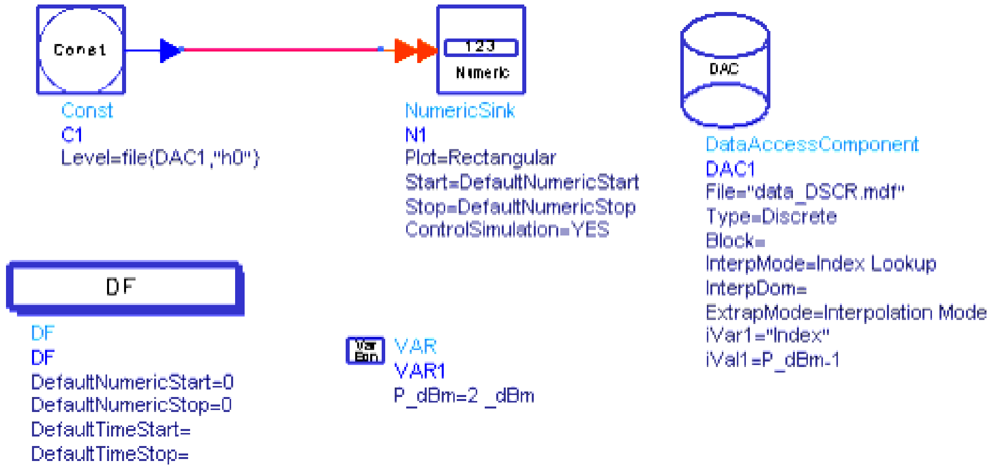
```
# hz S ma R 50
! 2 Port Network Data from SP1.SP block
    1e+009   0.85   -32   0.53   58   0.53   58   0.85   -32
    2e+009   0.80   -35   0.57   55   0.57   55   0.82   -35
    3e+009   0.72   -37   0.61   53   0.6    53   0.8    -37
```



Example 5

This example illustrates reading discrete data from the file data_DSCR.mdf using the DataAccessComponent (DAC) in an ADS Ptolemy design.

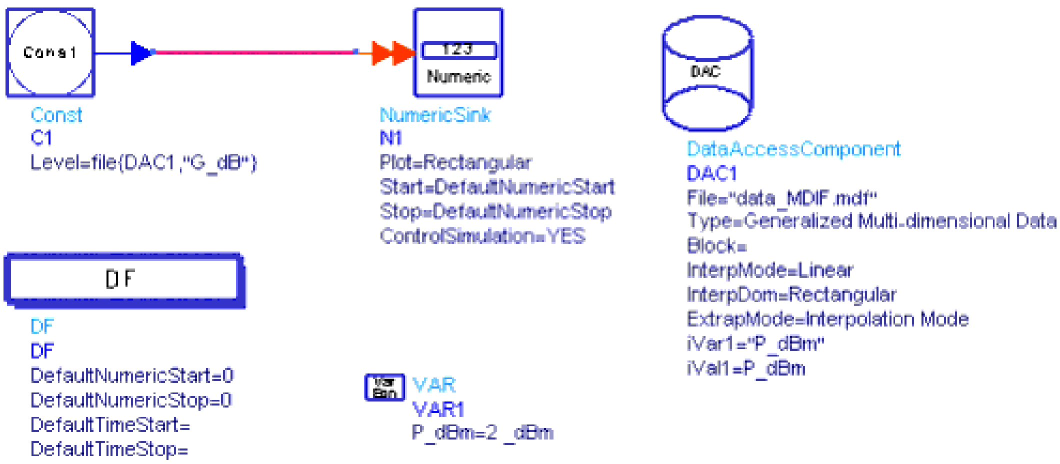
```
BEGIN DSCRDATA
% Index         G_dB            h0              h1
  1             0.1             0.11            0.111
  2             0.2             0.22            0.222
  3             0.3             0.33            0.333
  4             0.4             0.44            0.444
  5             0.5             0.55            0.555
  6             0.6             0.66            0.666
END DSCRDATA
```



Example 6

This example illustrates reading generalized multi-dimensional data from the file data_MDIF.mdf using the DataAccessComponent (DAC) in an ADS Ptolemy design.

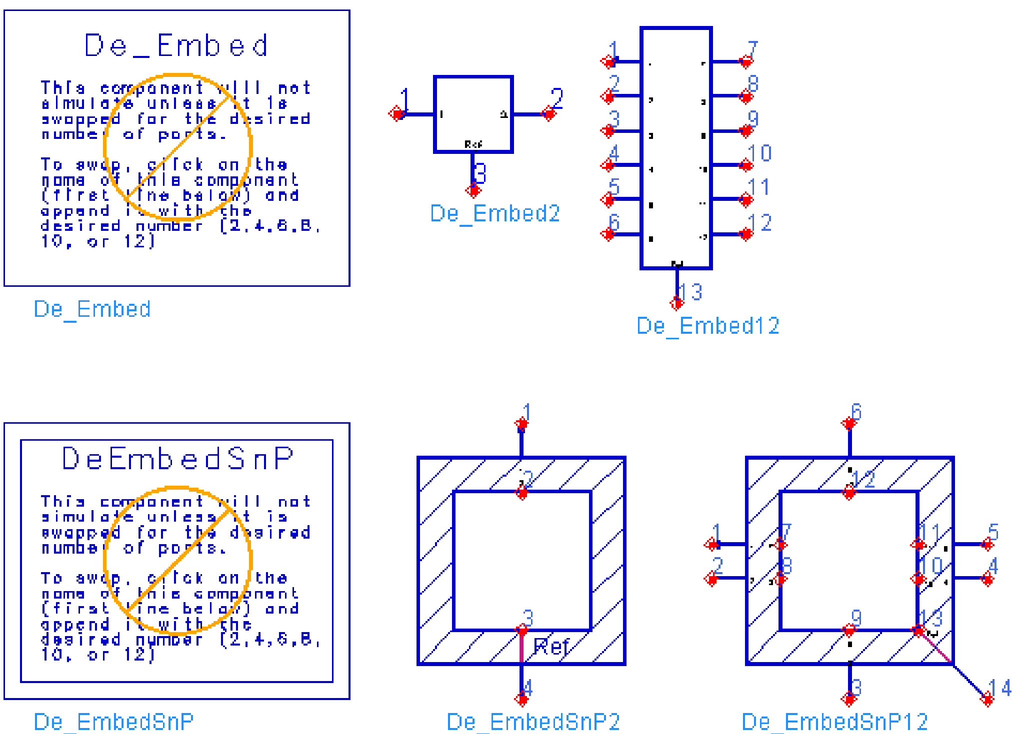
```
BEGIN Block1
% P_dBm(real)    G_dB(real)    h0(real)    h1(real)
  1              0.1           0.11        0.111
  2              0.2           0.22        0.222
  3              0.3           0.33        0.333
  4              0.4           0.44        0.444
  5              0.5           0.55        0.555
  6              0.6           0.66        0.666
END Block1
```



De_Embed/De_EmbedSnP (2-Port to 12-Port De-Embed Data File)

Symbols

Shown below are the generic De_Embed and De_EmbedSnP symbols, and specific symbols for 2-port and 12-port components. The easiest way to place a specific symbol in the Schematic window is to place the desired generic symbol then edit its name directly in the schematic. Append the desired number of ports to the generic component name: 2, 4, 6, 8, 10, 12. For example, change *De_Embed* to *De_Embed2* or *De_EmbedSnP* to *De_EmbedSnP2*. Preconfigured 2- and 4-port components are available on the palette for your convenience.



Parameters

Setup Dialog Name	Parameter Name	Description	Units	Default
File Name	File	Name of the data file containing S-, G-, H-, Y-, or Z-parameters for this component; the file extension and directory path are optional. Default directory is *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	Type	File type: Touchstone, Dataset, CITIfile, Value	None	Touchstone
Block Name	Block	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method	InterpMode	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	Linear or 0
Interpolation Domain	InterpDom	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method	ExtrapMode	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Pin-to-port mapping Type	PortMappingType	Standard (1), Custom (2)	None	Standard
Pin-to port mapping	PortMapping[n]	Integer array of fixture port mapping with respect to the reference (standard) pin arrangement; default is: 1,2,3,...	None	Standard
Display		Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, ExtrapMode, PortMappingType, PortMapping	None	File, PortMappingType

Range of Usage

1. The De_Embed/De_EmbedSnP components should be used within the frequency range of the S-, G-, H-, Y-, or Z-parameter data.
2. The De_Embed/DeEmbedSnP components are not intended for, and should not be used in transient simulations.
3. For proper de-embeding operation, the S-, G-, H-, Y-, or Z data must represent a bi-lateral (not necessarily reciprocal) behavior between the corresponding input and output ports at all simulation frequencies.

Notes

1. The data file, identified by the parameter File, must contain p-port data in the form of p×p S, Y, G, H or Z matrices. Only even numbers of ports p = 2n are allowed. In the case of S parameters, the use of different port reference impedances for different ports is supported. Also, complex reference impedances are supported provided that the definition of the S parameters is consistent with that of ADS (i.e., the power definition).
2. For information on data file formats, refer to *Working with Data Files* (cktsim), in the *Using Circuit Simulators* (cktsim) documentation.
3. Block is used only when Type=Dataset. Specify the name of an S-parameter data block if there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
4. InterpDom defines the domains in which two parts of a complex dependent variable are interpolated:
   - Rectangular: transform to (real, imag) before interpolation
   - Polar: transform to (mag, angle) before interpolation

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

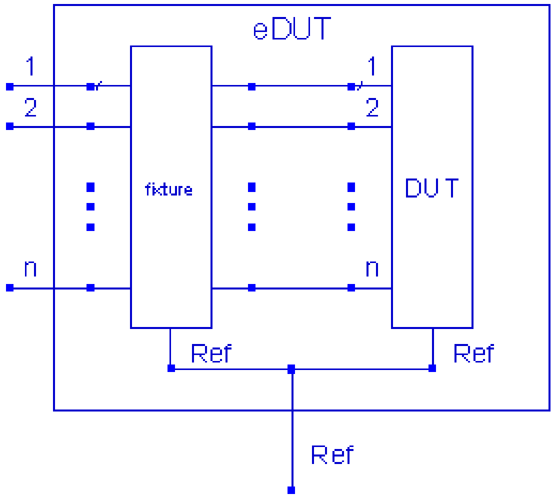
5. ExtrapMode specifies the extrapolation mode:
    - Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
    - Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
6. This component does not generate any noise.

Usage/Terminology

1. As shown at the top, there are two symbols for each number of ports p: De_Embed and De_EmbedSnP. De_Embed is a universal symbol with the input (the first n ports) on the left and with the output (the remaining n ports) on the right. This arrangement reflects the cascaded nature of the de-embeding process. De_EmbedSnP is specifically designed for the prevailing case of the embedded DUT being an SnP component. It enforces the cascaded connection by snapping the SnP symbol to the inside of the De_EmbedSnP symbol.
2. The available connection nodes include p (2n) pins and a reference pin *Ref*, and thus the total number of connection nodes is 2n + 1. The reference pin is numbered as 2n + 1 in the universal symbol. The SnP type symbols have two reference nodes, namely 2n + 1 and 2n + 2, which are short-circuited internally. Node 2n + 2 is intended for external connections.
3. The ports of the component are established by pairs of nodes with one of the first 2n pins as the "+" node and with the reference pin as the "-" node. The numbering of the ports of the De-Embed components follow that of the pin numbering: the k^{th} port is established between the pins *k* and *Ref*.
4. The first n ports are termed the input ports. The ports n + 1, ..., 2n are termed the output ports.
5. The connection (hook-up) of the De-Embed components to the embedded DUT (eDUT) is described in the section Connecting De_Embed Component to eDUT.
6. The section Fixture Pin/Port Mapping describes how to handle non-standard fixture data.

Embeded DUT (eDUT)

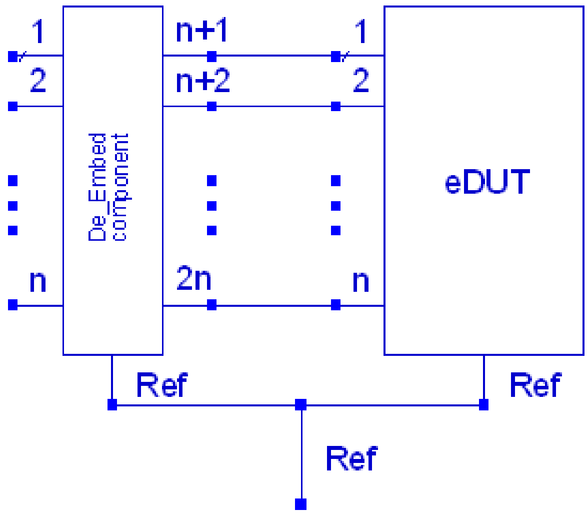
The purpose of the De_Embed components is to *undo* the effect of taking measurements at some externally available ports instead of the ports of interest: the ports of DUT (Device Under Test).



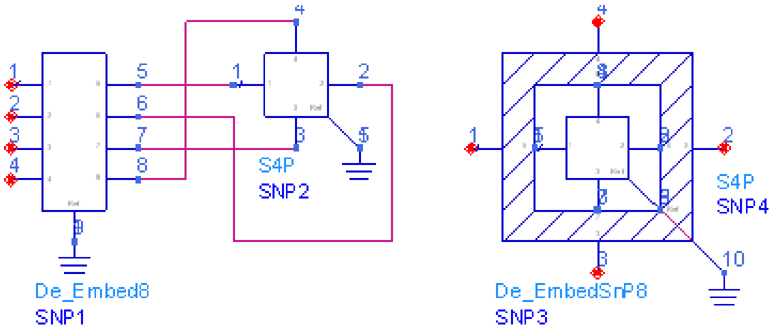
As shown in the diagram above, a measurement fixture is present between the points of actually taken measurements and DUT, and thus the measurement data is that of the embedded DUT (eDUT). The underlying assumption here is that the pins of eDUT correspond to the same numbered pins of DUT (the ports are established with respect to the Ref node and numbered consistently with the pins).

Connecting De_Embed Component to eDUT

There are two ways to connect the De_Embed component to eDUT. The first one is described here and the other in the next section. The first of the two connections follows an intuitive placement and wiring as shown in the following figure. The shape of the De_EmbedSnP components enforces the same pin-to-pin connection as for the universal component shown. The electrical behavior of eDUT is described by n-port parameters of the emebeded DUT, i.e., of DUT plus the fixture. The electrical behavior of the De_Embed component is described by 2n-port parameters of the fixture.

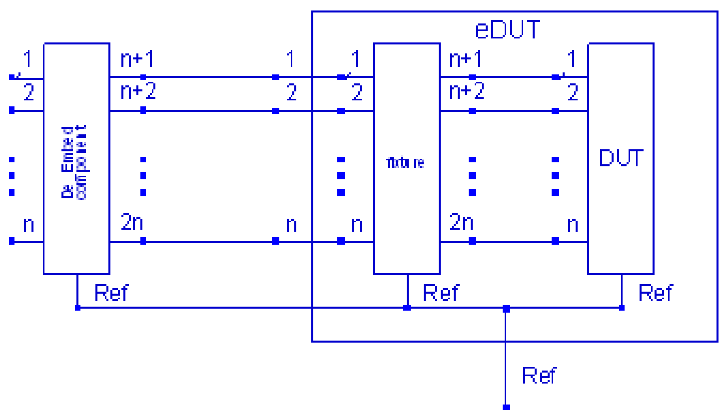


The underlying motivation in this connection is for the input pins of the De_Embed component to represent the same numbered pins of eDUT, and therefore the same numbered pins of DUT. Note that an additional requirement in this hook-up is that the output $(n+k)^{th}$ pin of the De_Embed component is to be connected to the k^{th} pin of eDUT. Examples of actual schematic connections are as follows.



By combining the diagrams of Embeded DUT and of the above connection of the De_Embed component to eDUT we have the following reference diagram.

Reference De-embeding Diagram



This diagram shows a reference connection of the fixture to DUT, resulting in a standard cascade connection of the De_Embed component and the fixture. Here, the term *standard cascade* refers to the fact that the input of one is connected to the output of the other. The cascade connection is the foundation of the de-embeding process as it is to provide *transparent* connections from the inputs of the De_Embed component to the corresponding inputs of DUT.

The simulation results of cascading the De_Embed component and the fixture can be affected by port ordering of the fixture. To facilitate the desired *transparency* of the De_Embed/fixture cascade the 2n-port data of the fixture (used by the De_Embed component) must have the port numbering consistent with the pin numbering of the fixture shown above. If it is not the case, please see the section Fixture Pin/Port Mapping.

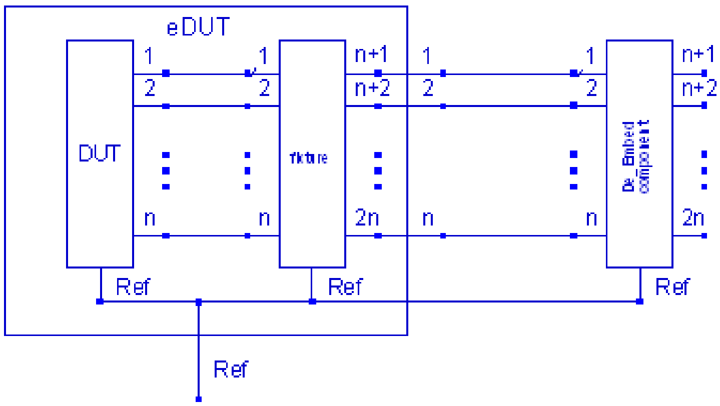
Alternative Connection at the Output Side

If you want to place the De_Embed component on the right-hand side of eDUT, one way is to follow the mirror images of the preceding diagrams. However, you may prefer to use the following, alternative connection.

This alternative hook-up, shown in the Alternative Reference De-embeding Diagram, also uses the standard cascade connection of the fixture and the De_Embed component, just in the opposite order. Here the output of the fixture is connected to the input of the De_Embed component.

In this hook-up the port associated with the $(n+k)^{th}$ pin of the De_Embed component represents the k^{th} port of DUT.

Alternative Reference De-embeding Diagram

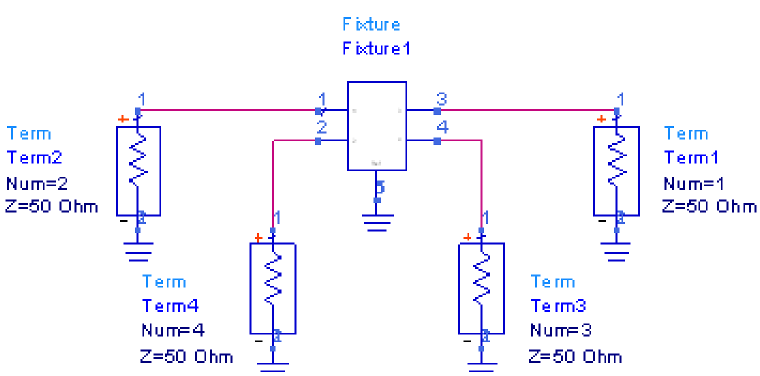


Again, to facilitate the desired *transparency* of the fixture/De_Embed cascade the 2n-port data of the fixture (used by the De_Embed component) must have the port numbering consistent with the pin numbering of the fixture shown above. If it is not the case, please refer to the section [Fixture Pin/Port Mapping].

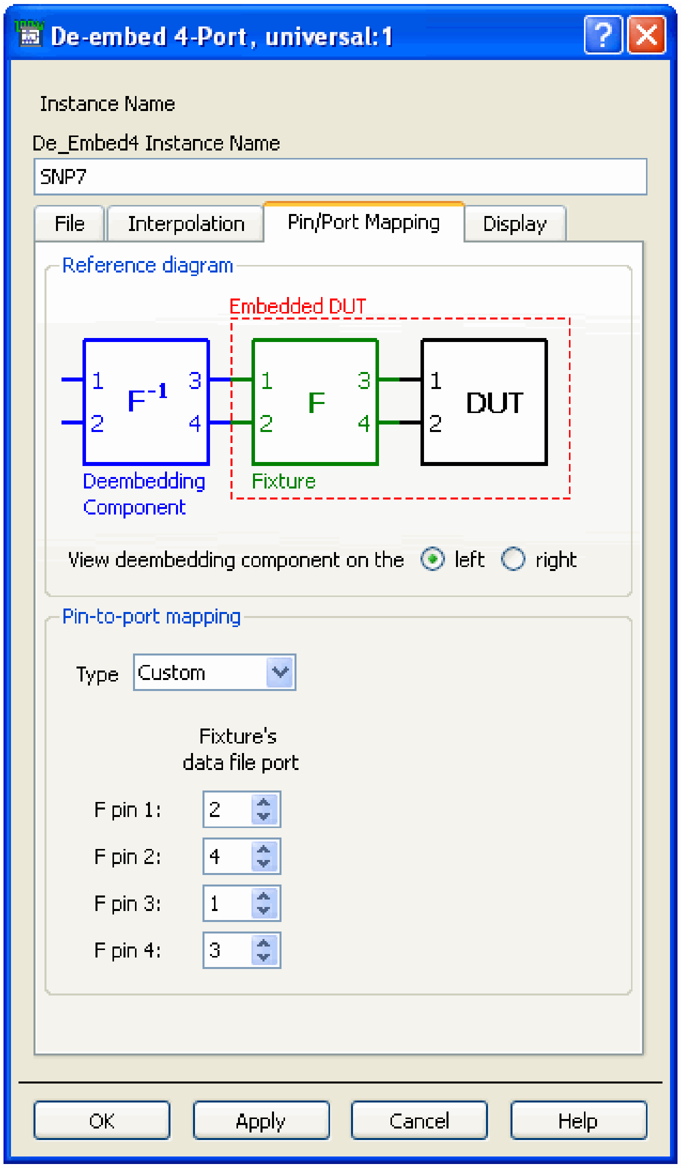
Fixture Pin/Port Mapping

The port mapping feature of the De_Embed component helps to accommodate the cases where the fixture 2n-port data (which is the input data for the De_Embed component) does not conform to the standard cascade configurations shown either in the [Reference De-embeding Diagram] or the [Alternative Reference De-embeding Diagram], respectively. If the fixture measurement setup is such that the port ordering is not consistent with the reference pin ordering then you can set *Pin-to-port mapping Type* to *Custom* and enter the actual port numbers for each of the 2n pins.

For example, for the following fixture measurement setup



the corresponding entries for *Pin-to-port mapping* in the UI are as follows.

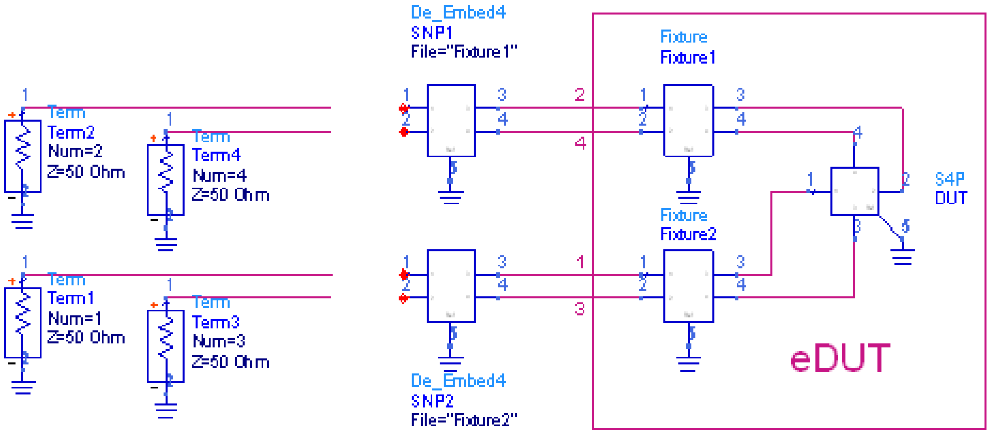


Duplicate entries are invalid and the entered port numbers must be between 1 and 2n.

Multiple Fixtures and Half-Fixtures

Depending on whether there exist couplings or not, the entire fixture may be split into two or more fixtures. Instead of combining the individual fixture data into one 2n×2n matrix, two or more De_Embed components may be used. Then the universal symbols must be used.

Note that the ports of DUT are split accordingly and thus the reference numbering of DUT pins for individual fixtures is no longer applicable. For example, one fixture may be connected to pins 1 and 3 of DUT, and another to pins 2 and 4, as shown in the following schematic.

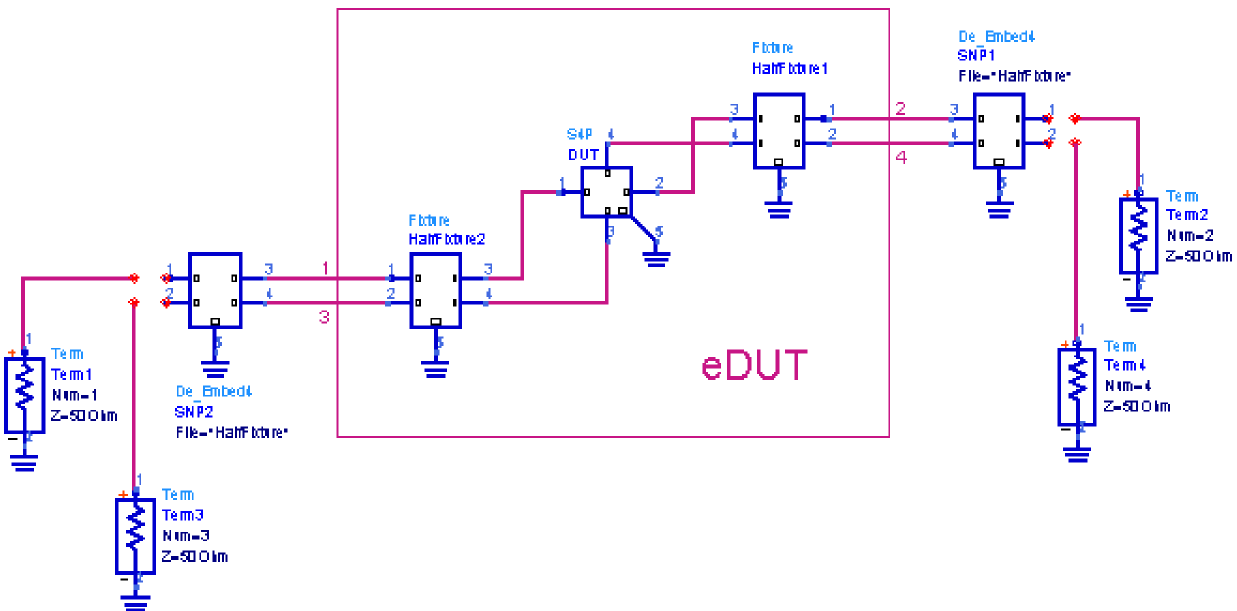


A proper setup involves

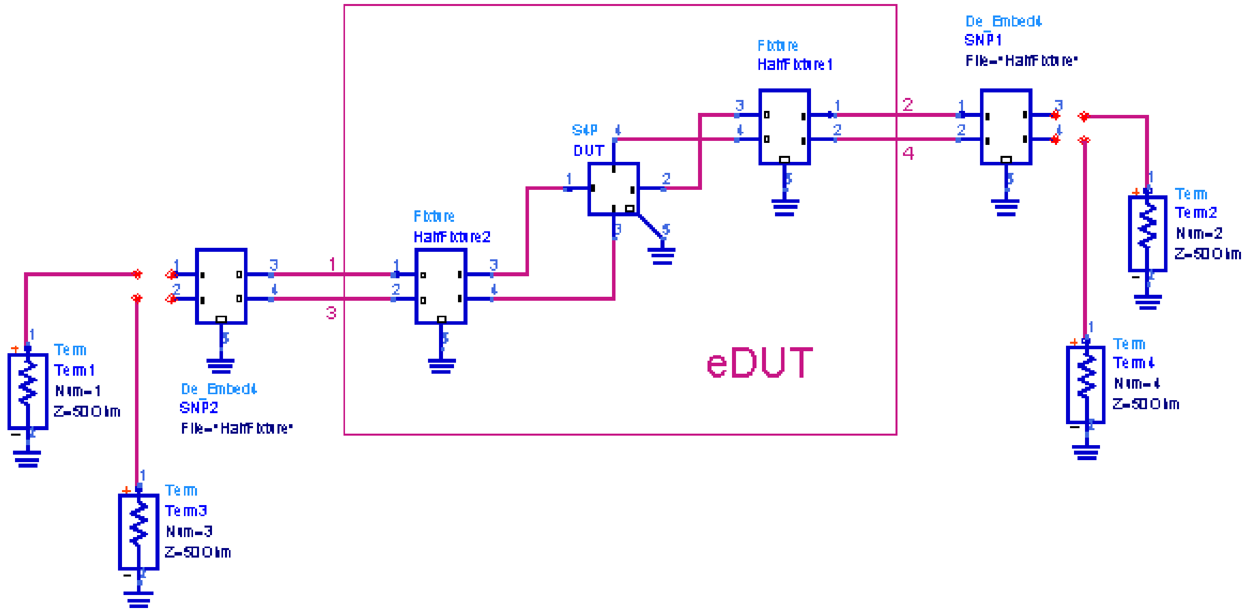
1. connecting the output of SNP1 to the pins 2 and 4 of eDUT,
2. connecting the output of SNP2 to the pins 1 and 3 of eDUT,
3. applying Fixture1 data to SNP1,
4. applying Fixture2 data to SNP2,
5. setting up appropriately Pin-to-port mappings.

If, in the diagram above, Fixture1 is identical to Fixture2 and the same file is used for both De_Embed components then the setting of Pin-to-port mapping on both De_Embed components will be identical. A case like that is frequently termed as a *half-fixture* setup.

If it is convenient to place the *output* De_Embed component to the right of eDUT you can use either the mirror image of the setup shown above, which is shown here,



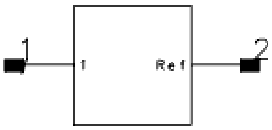
or the [Alternative Connection at the Output Side](#)



Since the two halves are symmetrical with respect to DUT the latter setup is inconsistent with the Alternative Reference De-embeding Diagram as far as the fixture is concerned. This can, however, be easily accommodated by pin-to-port mapping of the actual port numbering with respect to the reference diagram.

Deembed1 (1-Port De-Embed Data File)

Symbol



Parameters

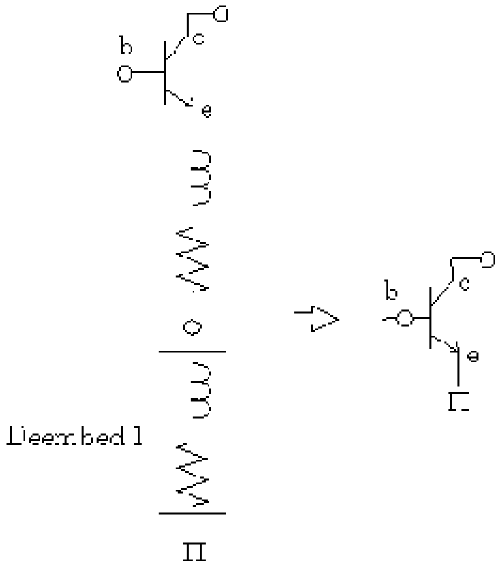
Name	Description	Units	Default
File Name	Name of data file containing 1-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is *.s1p* and the default directory is *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile, Value	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	Linear or 0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, ExtrapMode, Temp, UseLogFreq	None	File

Range of Usage

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

Notes/Equations

1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. One of the Deembed1 data file applications is to negate the 1-port subcircuit by using this data file.
3. In the following example, the BJT emitter's parasitics leads are de-embedded to obtain just the chip BJT. This ideal short and open behavior is not guaranteed if the deembed circuit has one or more frequency bands where a stop behavior is observed.

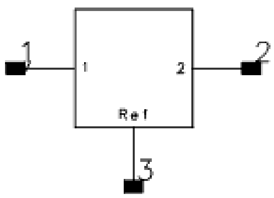


4. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
5. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated.
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
6. ExtrapMode specifies the extrapolation mode.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
7. This component does not generate any noise.
8. For time-domain analysis, the impulse response used for transient will be noncausal. This model should not be used for transient or circuit envelope analysis.
9. This component has no default artwork associated with it.

Deembed2 (2-Port De-Embed File)

ℹ	**Note** This component is **obsolete**, and has been replaced by the *De_Embed/De_EmbedSnP* (ccsim) component.

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *.s2p* and the default directory is *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile, Value	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	Linear or 0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	physical temperature	°C	27.0
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, ExtrapMode, Temp, UseLogFreq	None	File

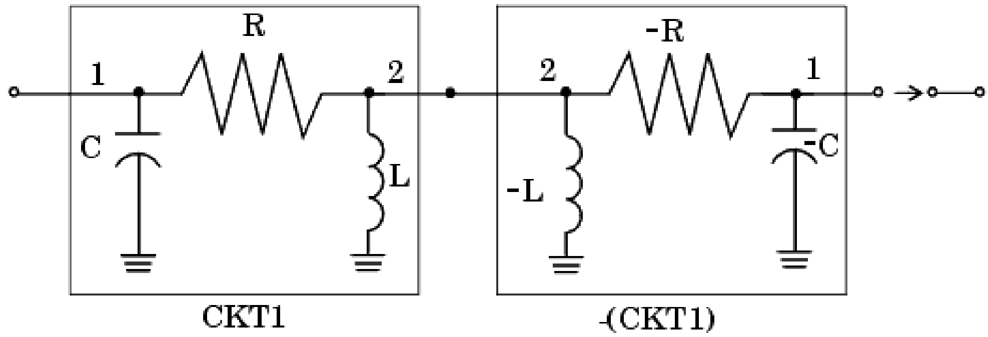
Range of Usage

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

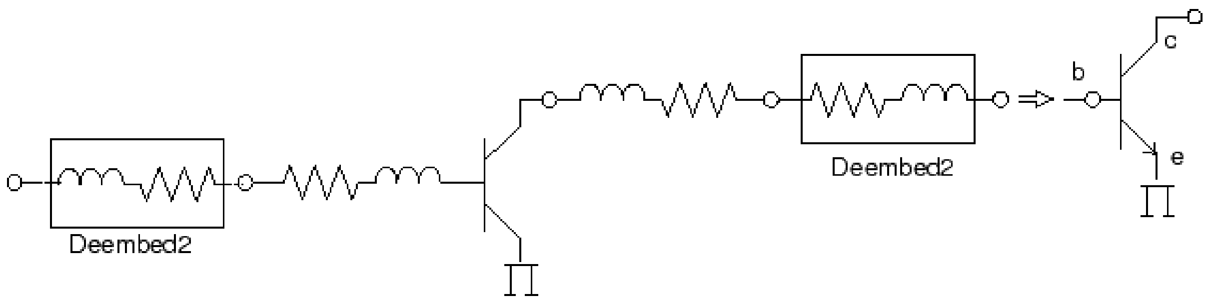
Notes/Equations

1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. One of the Deembed2 data file applications is to negate the 2-port subcircuit by using this data file.
3. When this component is connected in series with the sub-circuit being negated, and

the hookup is back-to-back as shown in the following illustration, the result is a short. When this component is connected in parallel, with the subcircuit being negated, the result is an open. This ideal short and open behavior is not guaranteed if the deembed circuit has one or more frequency bands where a stop behavior is observed.



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

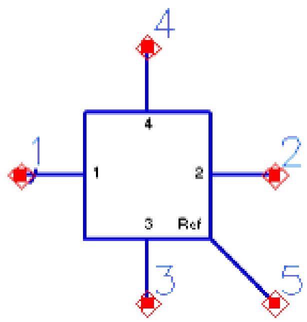


4. The 2-port S-parameters are assumed to be measured with pin 1 as the input, pin 2 as the output, and pin 3 as the common terminal.
5. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
6. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
7. ExtrapMode specifies the extrapolation mode.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
8. This component does not generate any noise.
9. For time-domain analysis, the impulse response used for transient will be noncausal. This model should not be used for transient or circuit envelope analysis.
10. This component has no default artwork associated with it.

Deembed4 (4-Port De-Embed Data File)

Note
This component is **obsolete**, and has been replaced by the *De_Embed/De_EmbedSnP* (ccsim) component.

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *.s4p* and the default directory is *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile, Value	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	Linear or 0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, ExtrapMode, Temp	None	File

Range of Usage

Within the frequency range of the S-parameter file.

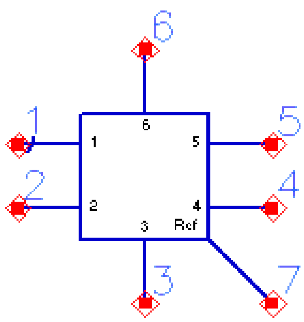
Notes/Equations

1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. One of the Deembed4 data file applications is to negate the 4-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, the result is a short. When this component is connected in parallel, with the subcircuit being negated, the result is an open. This ideal short and open behavior is not guaranteed if the deembed circuit has one or more frequency bands where a stop behavior is observed.
4. Block is used only when Type = Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
5. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
6. ExtrapMode specifies the extrapolation mode.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
7. This component does not generate any noise.
8. For time-domain analysis, the impulse response used for transient will be noncausal. This model should not be used for transient or circuit envelope analysis.
9. This component has no default artwork associated with it.

Deembed6 (6-Port De-Embed Data File)

Note
This component is **obsolete**, and has been replaced by the *De_Embed/De_EmbedSnP* (ccsim) component.

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *.s6p* and the default directory is *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile, Value	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	Linear or 0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, ExtrapMode, Temp	None	File

Range of Usage

Within the frequency range of the S-parameter file.

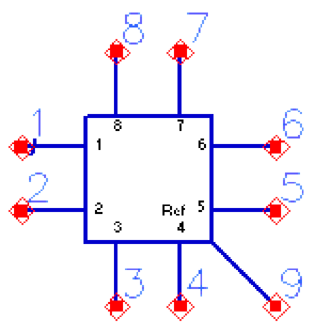
Notes/Equations

1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. One of the Deembed6 data file applications is to negate the 6-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, the result is a short. When this component is connected in parallel, with the subcircuit being negated, the result is an open. This ideal short and open behavior is not guaranteed if the deembed circuit has one or more frequency bands where a stop behavior is observed.
4. Block is used only when Type = Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
5. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
6. ExtrapMode specifies the extrapolation mode.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
7. This component does not generate any noise.
8. For time-domain analysis, the impulse response used for transient will be noncausal. This model should not be used for transient or circuit envelope analysis.
9. This component has no default artwork associated with it.

Deembed8 (8-Port De-Embed Data File)

Note
This component is **obsolete**, and has been replaced by the *De_Embed/De_EmbedSnP* (ccsim) component.

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *.s8p* and the default directory is *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile, Value	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	Linear or 0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, ExtrapMode, Temp	None	File

Range of Usage

Within the frequency range of the S-parameter file.

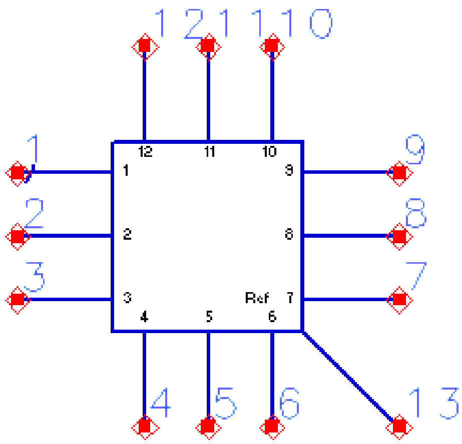
Notes/Equations

1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. One of the Deembed8 data file applications is to negate the 8-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, the result is a short. When this component is connected in parallel, with the subcircuit being negated, the result is an open. This ideal short and open behavior is not guaranteed if the deembed circuit has one or more frequency bands where a stop behavior is observed.
4. Block is used only when Type = Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
5. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
6. ExtrapMode specifies the extrapolation mode.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
7. This component does not generate any noise.
8. For time-domain analysis, the impulse response used for transient will be noncausal. This model should not be used for transient or circuit envelope analysis.
9. This component has no default artwork associated with it.

Deembed12 (12-Port De-Embed Data File)

Note
This component is **obsolete**, and has been replaced by the *De_Embed/De_EmbedSnP* (ccsim) component.

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile, Value	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	Linear or 0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, ExtrapMode, Temp	None	File

Range of Usage

Within the frequency range of the S-parameter file.

Notes/Equations

1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. One of the Deembed12 data file applications is to negate the 12-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, the result is a short. When this component is connected in parallel, with the subcircuit being negated, the result is an open. This ideal short and open behavior is not guaranteed if the deembed circuit has one or more frequency bands where a stop behavior is observed.
4. Block is used only when Type = Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
5. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
6. ExtrapMode specifies the extrapolation mode.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
7. This component does not generate any noise.
8. For time-domain analysis, the impulse response used for transient will be noncausal. This model should not be used for transient or circuit envelope analysis.

NetlistInclude (Netlist File Include Component)

Symbol

Parameters

Name	Description	Units	Default
IncludePath	Space-delimited search path for included files	None	None
IncludeFiles	List of files to include	None	None
UsePreprocessor	Use preprocessor: *yes* to use an *#include* directive; *no* to copy the full text of the file	None	yes

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 NetlistInclude component can directly read a Spectre file.
3. 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.
4. 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
```

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_wrk_models_resistor_lib
#define inc__users_default_default_wrk_models_resistor_lib
#include "/users/default/default_wrk/models/resistor.lib"
#endif
```

As the simulator reads the netlist, it will also read the referenced file (/users/default/default_wrk/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.
Example component parameters:

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

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

```
#ifndef inc_C__ADS_my_wrk_misc_functions_def
#define inc_C__ADS_my_wrk_misc_functions_def
#include "C:\ADS\my_wrk\misc\functions.def"
#endif
#define Nominal
#ifndef inc_C__ADS_my_wrk_resistor_lib
#define inc_C__ADS_my_wrk_resistor_lib
#include "C:\ADS\my_wrk\resistor.lib"
#endif
#undef Nominal
```

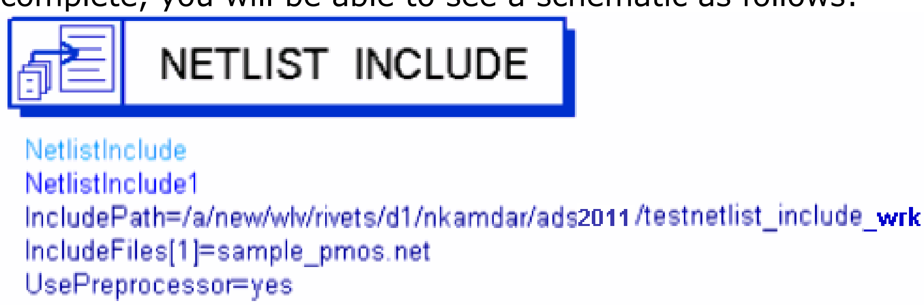
Use caution when placing a NetlistInclude component in a subcircuit. If an included file contains a subcircuit definition, the simulator will find one subcircuit definition inside another, and will stop after reporting a syntax error. Included files containing subcircuit definitions must be referenced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

Typically if you import a spice netlist file into an ADS netlist (or if you have an ADS netlist to begin with), you can end up using a NetlistInclude component. The following is an example of importing a spice file into an ADS netlist and then using the NetlistInclude component.

Start off with a simple HSPICE file with a PMOS transistor in it:

```
* HSPICE file
.model Qtest PMOS LEVEL=49
* Hspice model for PMOS transistor *
+TOX=5.50000E-09 XJ=1.5000001E-07
+NCH=8.5423000E+17 LLN=2.0000000 LWN=1.0000000
+WLN=1.0183233 WWN=0.4000000 LINT=1.9999999E-10
+LL=1.0225185E-21 LW=0.00 LWL=-2.3295743E-28
+WINT=3.025E-08 WL=-3.1805480E-15 WW=-1.6802396E-11
+WWL=0.00 MOBMOD=1 BINUNIT=2
+TNOM=25 DWG=-6.2803000E-09 DWB=6.9950080E-09
+VTH0=-0.5077000 K1=0.9734 K2=-0.1520450
+MJSW=.386 PBSW=.589 CLE=.6
+JS=1.46E-07 JSW=4E-12 CLC=.0000001
+CGDO=2.8116E-10 CGSO=2.8116E-10 CGBO=4.7E-11
+CAPMOD=2 NQSMOD=0 ELM=5
+XPART=1 CGSL=0 CGDL=0
+CKAPPA=.6 NOIMOD=2 NOIA=3.080481E+19
+NOIB=4.359458E+05 NOIC=-6.416658E-12 EF=1.059525
.end
```

Import this file into an open ADS workspace. From the ADS Main window select **File** > **Import**. Make sure **File Type** is set to *Netlist*, select *More Options* under **File Type** and choose *HSPICE* as Input Netlist Dialect and *ADS Netlist* as the Translated Output Format. For Import file name, select the above spice file and then click **OK**. Once the import is complete, you will be able to see a schematic as follows:

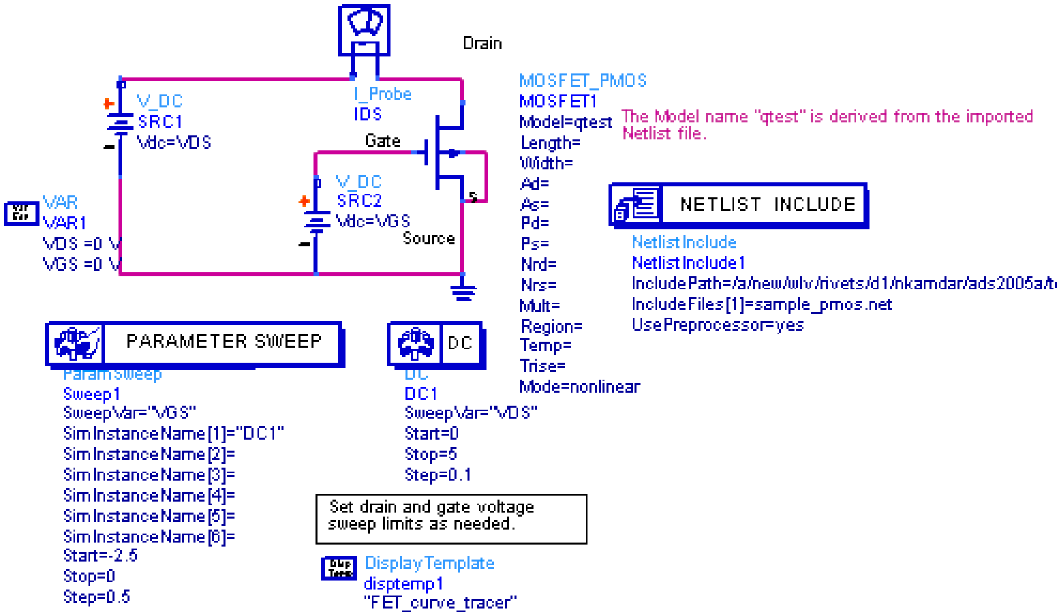


The Hspice file has been imported into an ADS netlist file called sample_pmos.net. You can view it by browsing in the directory. This is what it looks like:

```
; Translated with ADS Netlist Translator  290.400 Aug 6 2005
;Hspice file
model qtest MOSFET Tox=5.50000e-09 Xj=1.5000001e-07 Nch=8.5423000e+17 Lln=2.0000000 Lwn=1.0000000
\
Wln=1.0183233 Wwn=0.4000000 Lint=1.9999999e-10 Ll=1.0225185e-21 Lw=0.00 Lwl=-2.3295743e-28 \
Wint=3.025e-08 Wl=-3.1805480e-15 Ww=-1.6802396e-11 Wwl=0.00 Mobmod=1 Binunit=2 Tnom=25 \
Dwg=-6.2803000e-09 Dwb=6.9950080e-09 Vth0=-0.5077000 K1=0.9734 K2=-0.1520450 Mjsw=.386 \
Pbsw=.589 Cle=.6 Js=1.46e-07 Jsw=4e-12 Clc=.0000001 Cgdo=2.8116e-10 Cgso=2.8116e-10 \
Cgbo=4.7e-11 Capmod=2 Nqsmod=0 Elm=5 Xpart=1 Cgsl=0 Cgdl=0 Ckappa=.6 Noimod=2 Noia=3.080481e+19 \
Noib=4.359458e+05 Noic=-6.416658e-12 Ef=1.059525 PMOS=1 NMOS=0 Idsmod=8 Version=3.1 \
Cj=5.79e-4 Cjsw=0 Vbm=-3.0 U0=250 Uc=-4.56e-11 Delta=0.01 Uc1=-5.69e-11 Kt1=0.0 Em=4.1e7
```

```
; Hspice model for PMOS transistor *
```

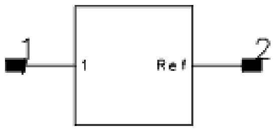
This Netlist include component now points to the file and represents the PMOS model qtest.
Here is a how you could set up a simulation for this PMOS model:



The Model=qtest parameter in the MOSFET_PMOS component actually points to the model in the netlist file which is represented by the NetlistInclude component.

S1P (1-Port S-parameter File)

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile	None	Touchstone
Block Name	(for Type=Dataset) name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
CheckPassivity	Check passivity or not. Yes=check; No=do not check	None	no
ImpNoncausalLength	Non-causal function impulse response order	Integer	None
ImpMode	Convolution mode (value type: integer)	Integer	None
ImpMaxFreq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order	Integer	None
ImpWindow	Smoothing window	Integer	None
ImpRelTol	Relative impulse response truncation factor	Integer	None
ImpAbsTol	Absolute impulse response truncation factor	Integer	None
EnforcePassivity	Sets passivity enforcement for S-parameters in transient/convolution simulations. If *Yes* is selected, passivity in this particular component is enforced. If *No* is selected, passivity is not enforced. If *Auto* is selected, the decision follows the setting of ImpEnforcePassivity in the Transient controller. This setting overwrites ImpEnforcePassivity in the Transient controller for only this component.	None	Auto
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, extrapMode, Temp, UseLogFreq, CheckPassivity, ImpNoncausalLength, ImpMode, ImpMaxFreq, ImpDeltaFreq, ImpMaxOrder, ImpWindow, ImpRelTol, ImpAbsTol	None	File

Notes/Equations

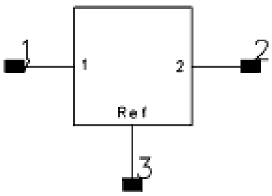
1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by

SNPC_COMPONENT_NAME.CMP1.

3. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
4. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
5. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
6. ExtrapMode specifies the extrapolation mode. An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
7. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the 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. Ref pin 2 is the common terminal; it is normally grounded, but can be used in non-grounded mode.
10. For time-domain analysis, the frequency-domain S-parameters are used.
11. This component has no default artwork associated with it.

S2P (2-Port S-parameter File)

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *<wrk>/data* where *<wrk>* is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
CheckPassivity	Check passivity or not. Yes=check; No=do not check	None	no
ImpNoncausalLength	Non-causal function impulse response order (value type: integer)	None	None
ImpMode	Convolution mode (value type: integer)	None	None
ImpMaxFreq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order (value type: integer)	None	None
ImpWindow	Smoothing window (value type: integer)	None	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None
EnforcePassivity	Sets passivity enforcement for S-parameters in transient/convolution simulations. If *Yes* is selected, passivity in this particular component is enforced. If *No* is selected, passivity is not enforced. If *Auto* is selected, the decision follows the setting of ImpEnforcePassivity in the Transient controller. This setting overwrites ImpEnforcePassivity in the Transient controller for only this component.	None	Auto
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, extrapMode, Temp, UseLogFreq, CheckPassivity, ImpNoncausalLength, ImpMode, ImpMaxFreq, ImpDeltaFreq, ImpMaxOrder, ImpWindow, ImpRelTol, ImpAbsTol	None	File

Notes/Equations

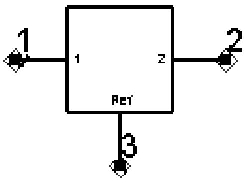
1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. The 2-port S-parameters are assumed to be measured with pin 1 as the input, pin 2 as the output, and pin 3 (Ref) as the common terminal.
   The Ref node (common terminal) is normally grounded; but can also be used 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. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.
4. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block

remains blank, the first S-parameter data block in the dataset file will be used.

5. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
6. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation.
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
7. ExtrapMode specifies the extrapolation mode. An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
8. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the file contains the noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn), these parameters are used to calculate the devices noise performance, independent of Temp. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.
9. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.
10. For time-domain analysis, the frequency-domain S-parameters are used.
11. This component has no default artwork associated with it.

S2P Conn (2-Port S-parameter File; connector artwork)

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset. CITIfile, Value	None	Touchstone
InterpMode	Interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup, Ceiling Value Lookup, Floor Value Lookup, Value	None	Linear
InterpDom	Interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB	None	Data Based
Temp	Physical temperature	°C	27.0

Range of Usage

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

Notes/Equations

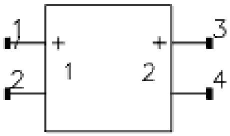
1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. The 2-port S-parameters are assumed to be measured with pin 1 as the input, pin 2 as the output, and pin 3 (Ref) as the common terminal.
   The Ref node (common terminal) is normally grounded; but can also be used 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. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.
4. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation.
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
5. 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°C, then the component does not generate any noise. If the file contains the noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn), these parameters are used to calculate the devices noise performance, independent of Temp. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.
7. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.
8. For transient analysis, pins 1 and 2 are shorted together.
9. For convolution analysis, the frequency-domain S-parameters are used.

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

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *<wrk>/data* where *< wrk >* is your current workspace directory.	None	None
File Type	File type: S2PMDIF, Touchstone, Dataset, CITIfile	None	S2PMDIF
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	Linear or 0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	
CheckPassivity	Check passivity or not: Yes=check; No=do not check	None	no
ImpNoncausalLength	Non-causal function impulse response order	Integer	None
ImpMode	convolution mode	Integer	None
ImpMaxFreq	maximum frequency to which device is evaluated	Hz	None
ImpDeltaFreq	sample spacing in frequency	Hz	None
ImpMaxOrder	Maximum impulse response order	Integer	None
ImpWindow	Smoothing window	Integer	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, extrapMode, Temp, UseLogFreq, CheckPassivity, ImpNoncausalLength, ImpMode, ImpMaxFreq, ImpDeltaFreq, ImpMaxOrder, ImpWindow, ImpRelTol, ImpAbsTol	None	File

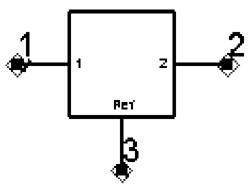
Note For more information on interpolation refer to *Setting Interpolation Parameters (ADS)* (ccsim)

Notes/Equations

1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.
3. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
4. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
5. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
   Note that no matter in which format the data is given, it is stored in Real/imag. If the interpDom is DB, the data is transformed to DB/angle before the interpolation is performed.
6. ExtrapMode specifies the extrapolation mode. An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
7. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the file contains the noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn), these parameters are used to calculate the devices noise performance, independent of Temp. If the S-parameters describe a 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. 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.
11. This component has no default artwork associated with it.

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

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile, Value	None	Touchstone
InterpMode	Interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup, Ceiling Value Lookup, Floor Value Lookup, Value	None	Linear
InterpDom	Interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB	None	Data Based
Temp	Physical temperature	°C	27.0
W1	(ADS Layout option) Width of line at pins 1 and 2	mil	5.0
W2	(ADS Layout option) Width of line at pin 3	mil	25.0
S	(ADS Layout option) Spacing (length from pin1 to pin 2	mil	10.0
L1	(ADS Layout option) Length from pin 1 to pin 2	mil	50.0
L2	(ADS Layout option) Length between pin 3 to pins 1 and 2	mil	50.0

Range of Usage

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

Notes/Equations

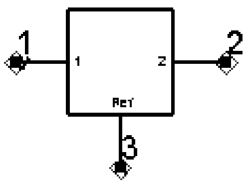
1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. The 2-port S-parameters are assumed to be measured with pin 1 as the input, pin 2 as the output, and pin 3 (Ref) as the common terminal.
   The Ref node (common terminal) is normally grounded; but can also be used 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. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.
4. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation.
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for

Y- and Z-parameters

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

S2P_Spac (2-Port S-parameter File)

Symbol



Parameters

Name	Description	Units	Default
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 *<wrk>/data* where <wrk> is your current workspace directory.	None	None
Type	File type: Touchstone, Dataset, CITIfile, Value	None	Touchstone
InterpMode	Interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup, Ceiling Value Lookup, Floor Value Lookup, Value	None	Linear
InterpDom	Interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB	None	Data Based
Temp	Physical temperature	°C	27.0
L	Length	mil	50.0

Range of Usage

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

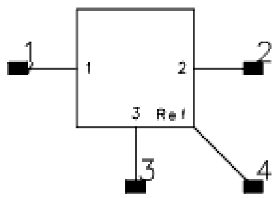
Notes/Equations

1. For information on data file formats, refer to *Working with Data Files* (cktsim).
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. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.
4. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation.
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
5. 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°C, then the component does not generate any noise. If the file contains the noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn), these parameters are used to calculate the devices noise performance, independent of Temp. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.
7. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.
8. For time-domain analysis, the frequency-domain S-parameters are used.
9. This component has no default artwork associated with it.
10. For transient analysis, pins 1 and 2 are shorted together.
11. For convolution analysis, the frequency-domain S-parameters are used.
12. 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

Name	**Description**	**Units**	**Default**
File Name	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 *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
CheckPassivity	Check passivity or not. Yes=check; No=do not check	None	no
ImpNoncausalLength	Non-causal function impulse response order (value type: integer)	None	None
ImpMode	Convolution mode (value type: integer)	None	None
ImpMaxFreq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order (value type: integer)	None	None
ImpWindow	Smoothing window (value type: integer)	None	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None
EnforcePassivity	Sets passivity enforcement for S-parameters in transient/convolution simulations. If *Yes* is selected, passivity in this particular component is enforced. If *No* is selected, passivity is not enforced. If *Auto* is selected, the decision follows the setting of ImpEnforcePassivity in the Transient controller. This setting overwrites ImpEnforcePassivity in the Transient controller for only this component.	None	Auto
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, extrapMode, Temp, UseLogFreq, CheckPassivity, ImpNoncausalLength, ImpMode, ImpMaxFreq, ImpDeltaFreq, ImpMaxOrder, ImpWindow, ImpRelTol, ImpAbsTol	None	File

Notes/Equations

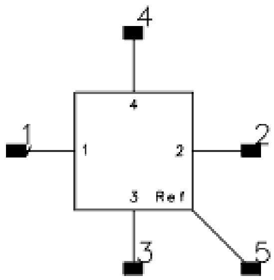
1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.
3. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
4. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
5. InterpDom defines the domains in that the two parts of a complex dependent

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

6. ExtrapMode specifies the extrapolation mode. An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
7. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the 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. Ref pin 4 is the common terminal; it is normally grounded, but can be used in non-grounded mode.
10. For time-domain analysis, the frequency-domain S-parameters are used.
11. This component has no default artwork associated with it.

S4P (4-Port S-parameter File)

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
CheckPassivity	Check passivity or not. Yes=check; No=do not check	None	no
ImpNoncausalLength	Non-causal function impulse response order (value type: integer)	None	None
ImpMode	Convolution mode (value type: integer)	None	None
ImpMaxFreq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order (value type: integer)	None	None
ImpWindow	Smoothing window (value type: integer)	None	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None
EnforcePassivity	Sets passivity enforcement for S-parameters in transient/convolution simulations. If *Yes* is selected, passivity in this particular component is enforced. If *No* is selected, passivity is not enforced. If *Auto* is selected, the decision follows the setting of ImpEnforcePassivity in the Transient controller. This setting overwrites ImpEnforcePassivity in the Transient controller for only this component.	None	Auto
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, extrapMode, Temp, UseLogFreq, CheckPassivity, ImpNoncausalLength, ImpMode, ImpMaxFreq, ImpDeltaFreq, ImpMaxOrder, ImpWindow, ImpRelTol, ImpAbsTol	None	File

Notes/Equations

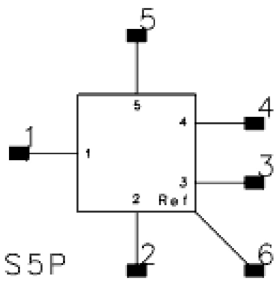
1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.
3. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
4. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
5. InterpDom defines the domains in that the two parts of a complex dependent

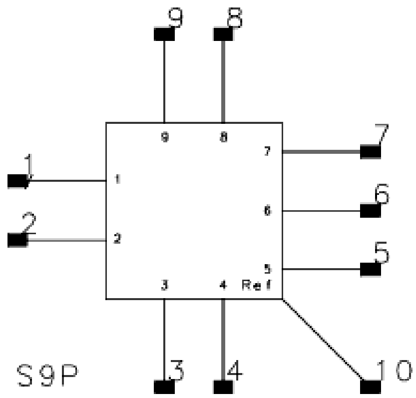
variable are interpolated:
Rectangular: transform to (real, imag) before interpolation
Polar: transform to (mag, angle) before interpolation
DB: transform to (dB, angle) before interpolation
Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters

6. ExtrapMode specifies the extrapolation mode. An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
7. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the 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. Ref pin 5 is the common terminal; it is normally grounded, but can be used in non-grounded mode.
10. For time-domain analysis, the frequency-domain S-parameters are used.
11. This component has no default artwork associated with it.

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

Symbol





Parameters

Name	Description	Units	Default
File Name	Name of data file containing 5 to 9-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is *.s<#>p* where # ranges from 5 to 9 and the default directory is *<wrk>/data* where *<wrk>* is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
CheckPassivity	Check passivity or not. Yes=check; No=do not check	None	no
ImpNoncausalLength	Non-causal function impulse response order (value type: integer)	None	None
ImpMode	Convolution mode (value type: integer)	None	None
ImpMaxFreq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order (value type: integer)	None	None
ImpWindow	Smoothing window (value type: integer)	None	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None
EnforcePassivity	Sets passivity enforcement for S-parameters in transient/convolution simulations. If *Yes* is selected, passivity in this particular component is enforced. If *No* is selected, passivity is not enforced. If *Auto* is selected, the decision follows the setting of ImpEnforcePassivity in the Transient controller. This setting overwrites ImpEnforcePassivity in the Transient controller for only this component.	None	Auto
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, extrapMode, Temp, UseLogFreq, CheckPassivity, ImpNoncausalLength, ImpMode, ImpMaxFreq, ImpDeltaFreq, ImpMaxOrder, ImpWindow, ImpRelTol, ImpAbsTol	None	File

Notes/Equations

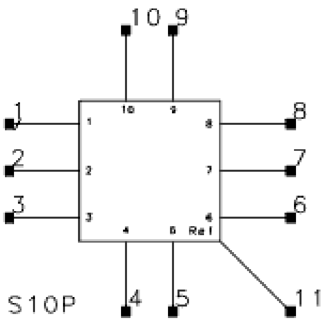
1. For information on data file formats, refer to *Working with Data Files* (cktsim).
2. The number of terminals increases sequentially from 5 to 9, and is equal to the number of ports of the component.
   Ref is the common terminal; it is normally grounded, but can be used in non-grounded mode.
3. If 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. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.
5. Block is used only when Type=Dataset. Specify the name of an S-parameter data

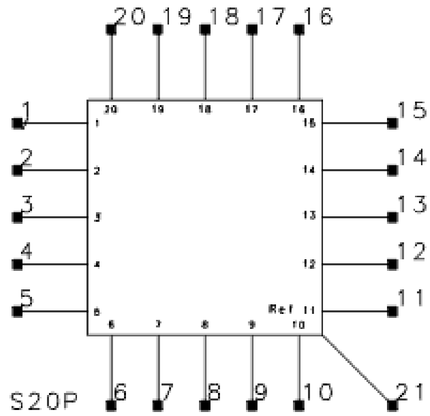
block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.

6. For the InterpMode parameter: interpolation of S-, Y-, or Z-parameters, PortZ reference impedance (and noise parameters for S2P) vs. simulation variable freq can be linear, cubic spline, cubic, or lookup by actual freq value. The two parts of each complex parameter (RI, MA, dBA) are interpolated independently.
7. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation.
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
8. ExtrapMode specifies the extrapolation mode. An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
9. If the component temperature Temp is less than -273°C, then the component does not generate any noise. If the S-parameters describe a passive device, then Temp and Twiss's theorem are used to calculate its noise performance. If the S-parameters describe an active device, no noise is generated.
10. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
    Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
    If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.
11. For time-domain analysis, the frequency-domain S-parameters are used.
12. This component has no default artwork associated with it.

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

Symbol





Parameters

Name	Description	Units	Default
File Name	Name of data file containing 10 to 20-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is *.s<#>p* where # ranges from 10 to 20 and the default directory is *<wrk>/data* where *<wrk>* is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
CheckPassivity	Check passivity or not. Yes=check; No=do not check	None	no
ImpNoncausalLength	Non-causal function impulse response order (value type: integer)	None	None
ImpMode	Convolution mode (value type: integer)	None	None
ImpMaxFreq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order (value type: integer)	None	None
ImpWindow	Smoothing window (value type: integer)	None	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None
EnforcePassivity	Sets passivity enforcement for S-parameters in transient/convolution simulations. If *Yes* is selected, passivity in this particular component is enforced. If *No* is selected, passivity is not enforced. If *Auto* is selected, the decision follows the setting of ImpEnforcePassivity in the Transient controller. This setting overwrites ImpEnforcePassivity in the Transient controller for only this component.	None	Auto
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, extrapMode, Temp, UseLogFreq, CheckPassivity, ImpNoncausalLength, ImpMode, ImpMaxFreq, ImpDeltaFreq, ImpMaxOrder, ImpWindow, ImpRelTol, ImpAbsTol	None	File

Notes/Equations

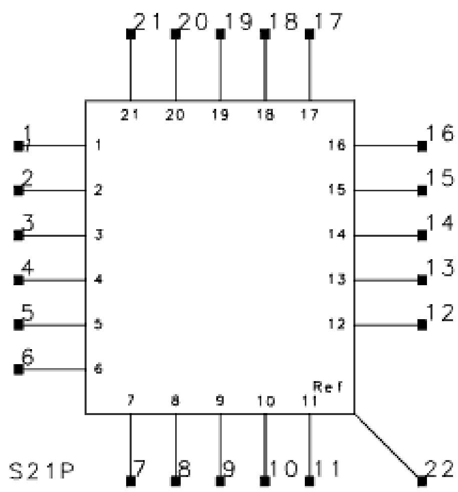
1. The items S11P through S99P cannot be selected from the component palette. They are accessed by typing the appropriate name (such as S12P or S98P) into the right entry box (directly above the viewing area), pressing *Enter* , then moving the cursor to the viewing area to place the item.
2. The number of terminals increases sequentially from 10 to 20, and is equal to the number of ports of the component.
   Ref is the common terminal; it is normally grounded, but can be used in non-grounded mode.
3. For information on data file formats, refer to *Working with Data Files* (cktsim).
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. When an SnP component references a CITIfile, and there is more than one data block

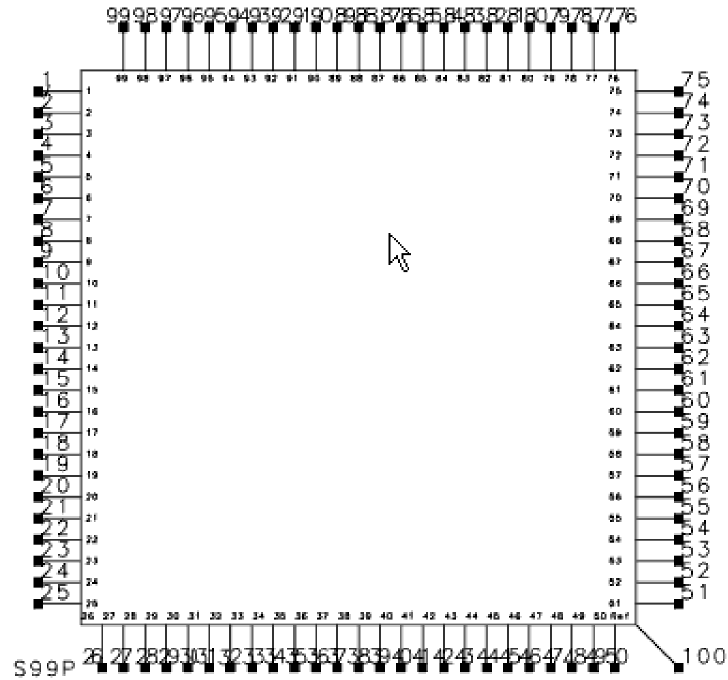
in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.

6. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
7. 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.
8. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
9. ExtrapMode specifies the extrapolation mode. An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when extrapolation occurs, no interpolation is performed; the value of the nearest data point is returned.
10. 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.
11. 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.
12. For time-domain analysis, the frequency-domain S-parameters are used.
13. This component has no default artwork associated with it.

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

Symbol





Parameters

Name	Description	Units	Default
File Name	Name of data file containing 21 to 99-port S-, Y-, or Z-parameters for this component. The file extension and directory path are optional. Default extension is *.s<#>p* where # ranges from 21 to 99 and the default directory is *<wrk>/data* where *<wrk>* is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
CheckPassivity	Check passivity or not. Yes=check; No=do not check	None	no
ImpNoncausalLength	Non-causal function impulse response order (value type: integer)	None	None
ImpMode	Convolution mode (value type: integer)	None	None
ImpMaxFreq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order (value type: integer)	None	None
ImpWindow	Smoothing window (value type: integer)	None	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None
EnforcePassivity	Sets passivity enforcement for S-parameters in transient/convolution simulations. If *Yes* is selected, passivity in this particular component is enforced. If *No* is selected, passivity is not enforced. If *Auto* is selected, the decision follows the setting of ImpEnforcePassivity in the Transient controller. This setting overwrites ImpEnforcePassivity in the Transient controller for only this component.	None	Auto
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, extrapMode, Temp, UseLogFreq, CheckPassivity, ImpNoncausalLength, ImpMode, ImpMaxFreq, ImpDeltaFreq, ImpMaxOrder, ImpWindow, ImpRelTol, ImpAbsTol	None	File

Notes/Equations

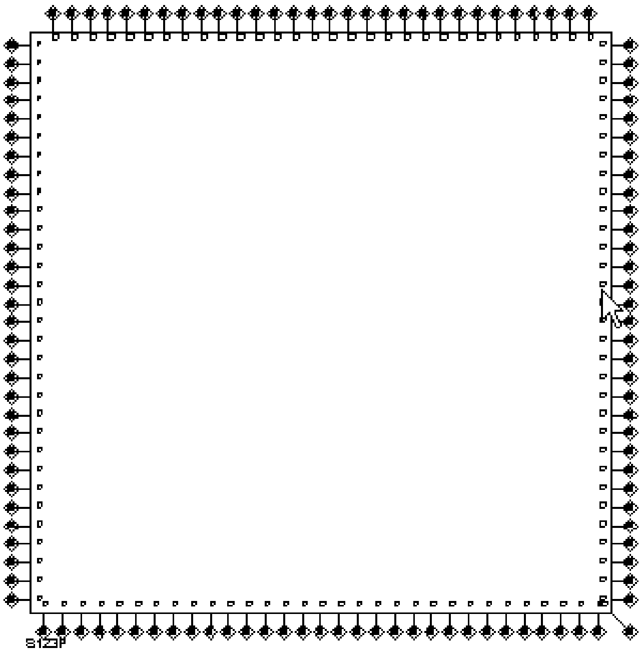
1. S11P through S99P support up to 99-port networks. They cannot be selected from the component palette; they are accessed by typing the appropriate name (such as S11P or S99P) into the field above the viewing area, pressing Enter, then moving the cursor to the viewing area to place the item.
2. Ref 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. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the

first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.

6. Block is used only when Type=Dataset. Specify the name of an S-parameter data block when there are multiple S-parameter data blocks in a dataset file. If Block remains blank, the first S-parameter data block in the dataset file will be used.
7. 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.
8. InterpDom defines the domains in that the two parts of a complex dependent variable are interpolated:
   Rectangular: transform to (real, imag) before interpolation
   Polar: transform to (mag, angle) before interpolation
   DB: transform to (dB, angle) before interpolation
   Data Based: (Series IV compatibility) uses Polar for S-parameters, Rectangular for Y- and Z-parameters
9. ExtrapMode specifies the extrapolation mode. An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.
   Interpolation Mode: when extrapolation occurs, the interpolation mode specified by InterpMode is used for extrapolation.
   Constant Extrapolation: when olation occurs, no interpolation is performed; the value of the nearest data point is returned.
10. 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.
11. 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.
12. For time-domain analysis, the frequency-domain S-parameters are used.
13. This component has no default artwork associated with it.

SnP component (n>99)

Symbol



Parameters

Name	**Description**	**Units**	**Default**
File Name	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 *<wrk>/data* where *<wrk>* is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile	None	Touchstone
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2, Floor Value Lookup or 4, Ceiling Value Lookup or 5, Floor Value Lookup or 6	None	0
Interpolation Domain (or InterpDom)	Interpolation domain: Data Based, Rectangular, Polar, DB	None	Data Based
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
Temp	Physical temperature	°C	27.0
CheckPassivity	Check passivity of the S-parameter data or not. Yes=check; No=do not check	None	no
ImpNoncausalLength	Non-causal function impulse response order (value type: integer)	None	None
ImpMode	Convolution mode (value type: integer)	None	None
ImpMaxFreq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order (value type: integer)	None	None
ImpWindow	Smoothing window (value type: integer)	None	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None
EnforcePassivity	Sets passivity enforcement for S-parameters in transient/convolution simulations. If *Yes* is selected, passivity in this particular component is enforced. If *No* is selected, passivity is not enforced. If *Auto* is selected, the decision follows the setting of ImpEnforcePassivity in the Transient controller. This setting overwrites ImpEnforcePassivity in the Transient controller for only this component.	None	Auto
Display	Display parameter on schematic: File, Type, Block, InterpMode, InterpDom, extrapMode, Temp, UseLogFreq, CheckPassivity, ImpNoncausalLength, ImpMode, ImpMaxFreq, ImpDeltaFreq, ImpMaxOrder, ImpWindow, ImpRelTol, ImpAbsTol	None	File

Notes/Equations

1. SnP component (n>99) is used to create SnP components with a port number larger than 99.
2. SnP(n>99) component is not a built-in component. SnP(n>99) components cannot be selected from the *Data Items* component library palette (which is the case for n=1:10), nor can they be typed into the field above the design viewing area (which is the case for n=10:99). Instead, for n>99, you must create SnP components by following the procedure given in the Example section below.

Tip

1. When an SnP component references a CITIfile, and there is more than one data block in the CITIfile, the simulator picks the first data block as S-parameter data. If the first data block is not S-parameter data, an error message might be reported by SNPC_COMPONENT_NAME.CMP1.
2. An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.

Example

Procedure for ADS2011

1. From the Main window select **Tools** > **Command Line** and type:
   generate_snp_component(112 , "MyLibrary")
   where:
   112 is the number of the ports.
   "MyLibrary" is the name of the library you want the new component to be created in.
   This generates all the required files including a new cell S112P in the specified library and a file S112P in the workspace directory and loads the device into ADS.
2. Type the name of the component (e.g., S112P) in the schematic component history list to place it into the Schematic window.

To delete a SnP component, all the files generated using the procedures described above need to be deleted.
These are the Cell SnP and the file <curr_wrk>/SnP.

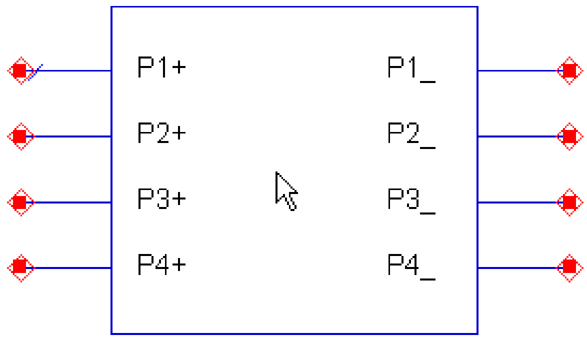
SnP_Diff component

Description

Unlike SnP components which have one global reference node and only the "plus" terminals for each port, SnP_Diff components are SnP Data Items which have two terminals (plus/minus) per port models.

> **Note**
> The SnP_Diff component is not a differential (or mixed-mode) SnP component. Since the minus terminals of an SnP_Diff component can be connected to nodes other than Ground, caution must be taken when using the component. Incoming current through the plus terminals must be the same as the current leaving through the minus terminals. If this requirement is violated, S-parameter data will be useless.

Symbol



Parameters

Name	Description	Units	Default
File Name	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 *<wrk>/data* where <wrk> is your current workspace directory.	None	None
File Type	File type: Touchstone, Dataset, CITIfile	None	None
Block Name	(for Type=Dataset) Name of S-parameter data block	None	None
InterpMode	Interpolation mode: Linear, Cubic Spline, Cubic, Value Lookup	None	None
InterpDom	Interpolation domain: Data Based (polar for S and rectangular for Y and Z), Rectangular, Polar, DB	None	None
ExtrapMode	Extrapolation mode: Interpolation Mode, Constant Extrapolation	None	None
Temp	Physical temperature	°C	Celsius
ImpNoncausalLength	Non-causal function impulse response order	Integer	None
ImpMode	Convolution mode	Integer	None
ImpMaxFreq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order	Integer	None
ImpWindow	Smoothing window	Integer	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None
EnforcePassivity	Sets passivity enforcement for S-parameters in transient/convolution simulations. If *Yes* is selected, passivity in this particular component is enforced. If *No* is selected, passivity is not enforced. If *Auto* is selected, the decision follows the setting of ImpEnforcePassivity in the Transient controller. This setting overwrites ImpEnforcePassivity in the Transient controller for only this component.	None	Auto

Notes/Equations

1. SnP_Diff components cannot be selected from the component palette; they are accessed by typing the appropriate name (such as S12P_Diff) into the field above the viewing area and press Enter, then moving the cursor to the viewing area to place the item
2. SnP_Diff component is not a built-in component. You must create this component by following the procedure given below

Procedure for ADS2011

1. From the Main window select **Tools** > **Command Line** and type:
   generate_snp_diff_component(12 , "MyLibrary")
   where:
   12 is the number of the ports.
   "MyLibrary" is the name of the library you want the new component to be created in.
   This generates all the required files including a new cell S12P_Diff in the specified library and a file S12P_Diff in the workspace directory and loads the device into ADS.
2. Type the name of the component (e.g., S12P_Diff) in the schematic component history list to place it into the Schematic window.

To delete a SnP_Diff component, all the files generated using the procedures described above need to be deleted.
These are the Cell SnP_Diff and the file <curr_wrk>/SnP_Diff

Hint
An extrapolation warning is issued when an extrapolation occurs on *freq* in an S-parameter simulation. For all other analysis types, status level in the analysis controller must be set to 3 or higher to see extrapolation warnings.

VAR (Variables and Equations Component)

Symbol

Parameters

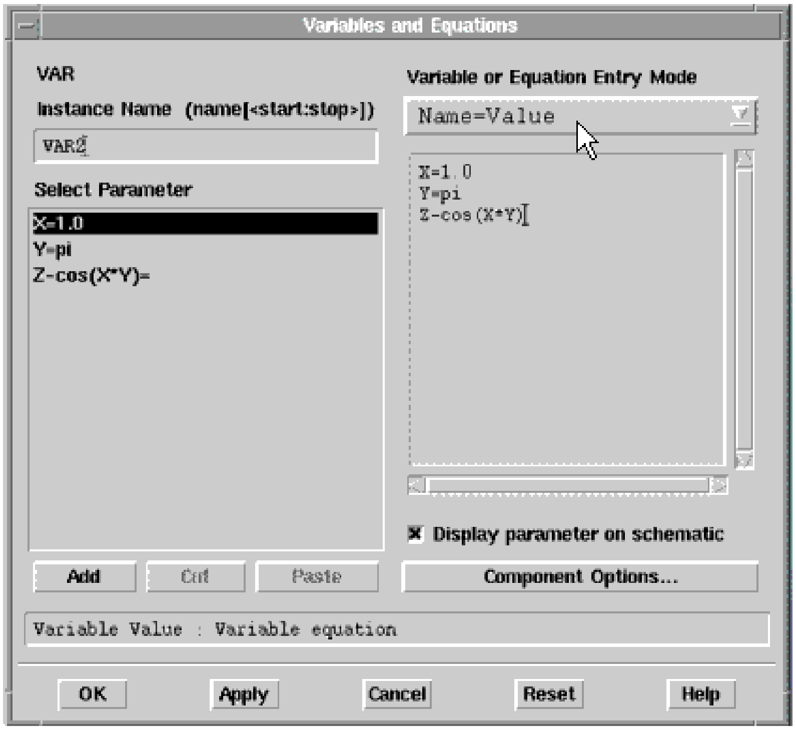
Name	Description	Units	Default
X	Name of variable or equation	None	1.0

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.

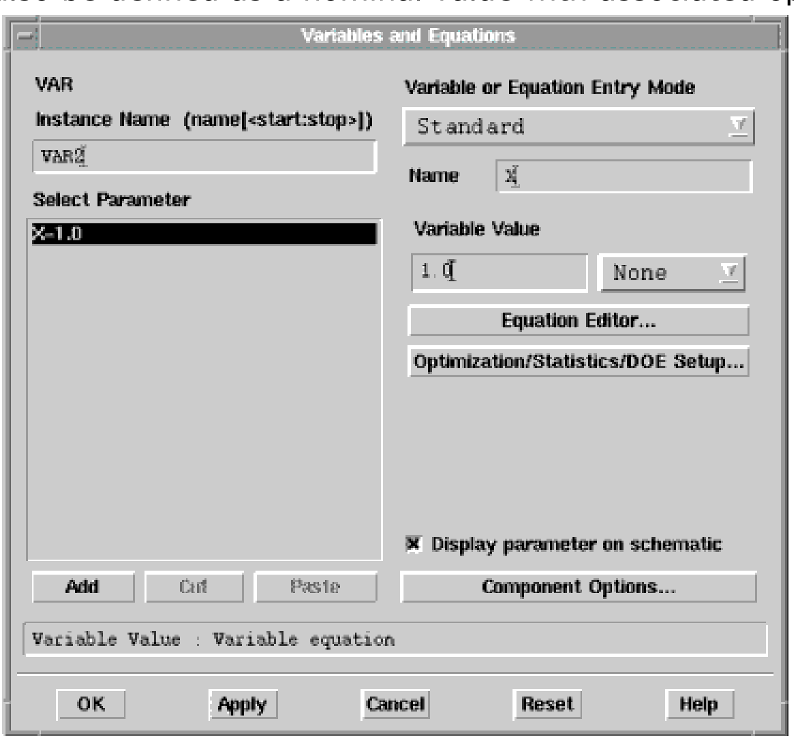
   > **Note**
   > There is a limitation of 1800 characters in a variable equation. Using more than 1800 characters may cause ADS to crash.

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).



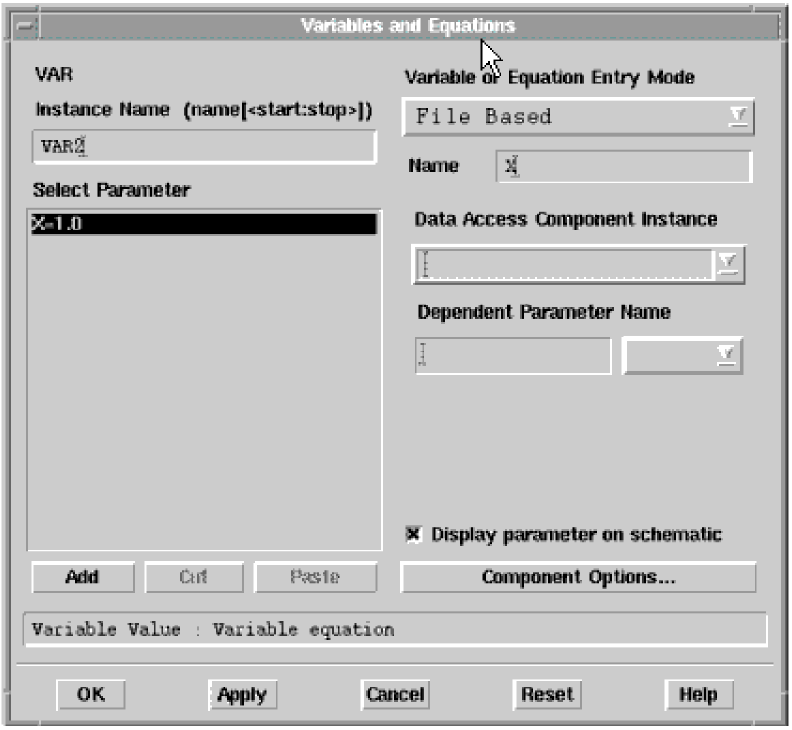
Note that
expression X has a numeric value;
expression Y uses a predefined constant;
expression Z uses a predefined function.

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.



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 variable to be created, as identified in the DataAccessComponent. Data Access Component Instance is the instance name of the

particular DataAccessComponent that you are referencing. Dependent Parameter Name is the name of a DataAccessComponent parameter for which you want to include data.



3. Pre-defined Built-in Constants
The pre-defined built-in constants available for use in an equation are here.

```
e          = 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=S11, where the results of S11 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 *Introduction to Simulator Expressions* (expsim) for a complete list of simulator expressions. A function that returns a real value effectively has a zero value imaginary term.

5. Equation Editor Syntax
   Mathematical expressions entered equations can include the following items.
   - Blank spaces Blank spaces within an expression are ignored; they can be used to improve readability. For example,
     4 * (x + .1) evaluates the same as
     4*(x+.1)
   - Numerical constants Real numbers such as 12.68, exponential notation numbers such as 1e6 or 25.1e3, pi can be used, and complex numbers can be defined. For example, z = x + j*y.
   - LHS form assignment The LHS assignment takes the form of integer, double, complex or string dependent on what form is associated with the RHS. For example,
     X=4, Y=4.0, W=1.0+j*3.0, Z="4"
     associates the form of integer, double, complex and string to X, Y, W and Z, respectively. The LHS form is important when subsequently used in following expressions.
   - Mathematical operators Standard operators are available:

```
** exponentiation
exponentiation
* multiplication
/ division
+ addition
- subtraction
```

> **Note**
> To avoid returning incorrect results when using ** with very large integers, convert integers to real numbers first.

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 view 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.

Scope: Nested or Global

When referencing VAR items in hierarchical designs, set the Scope option to indicate the levels, from a hierarchical standpoint, that recognize the expressions defined in the VAR. To set Scope, click Component Options.

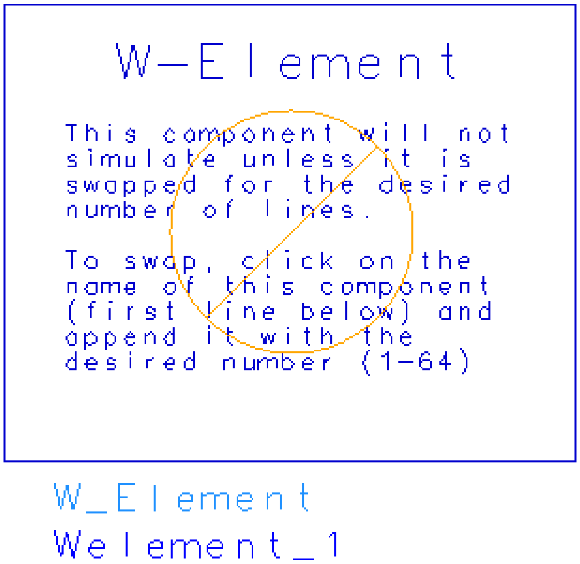
Nested sets the scope so VAR item expressions are recognized within the design containing the VAR item, as well as within any subnetworks (designs at lower levels) referenced by the design containing the VAR item. This is the default.

Global sets the scope so VAR item expressions are recognized throughout the entire design, no matter what level in the design hierarchy the VAR item is placed.

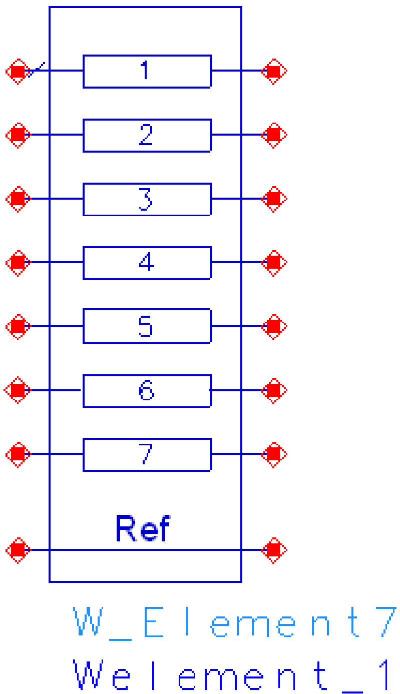
W_Element (Multi-Conductor Transmission Lines)

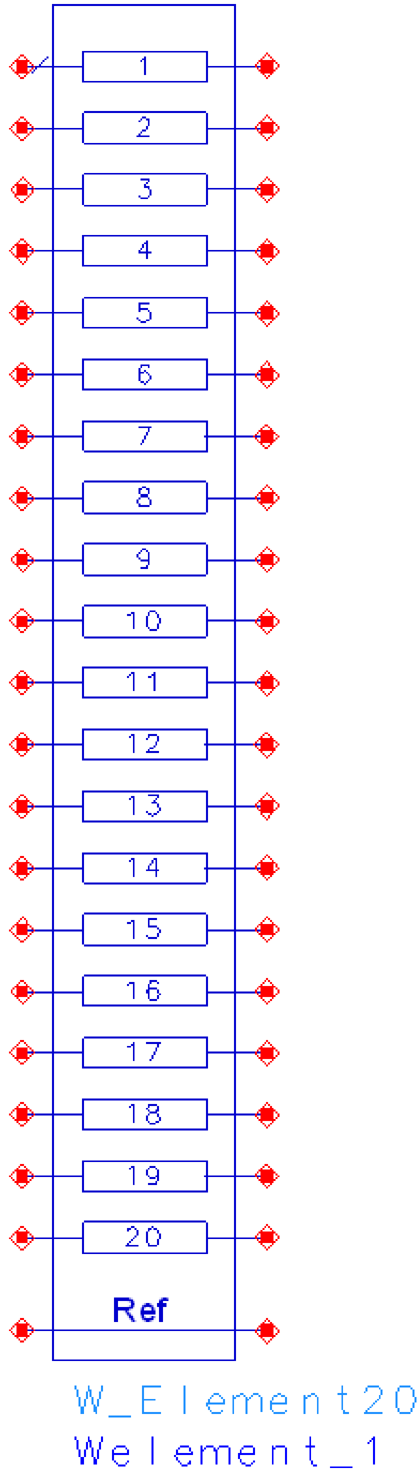
Symbols

Generic Symbol (interface to specific symbols for 1-64 lines)



Specific Symbol (examples of a 7-line and 20-line symbol)





Signal line numbers, as shown in the symbols, are relevant to the data ordering (rows and columns of the RLGC matrices). Please note that the easiest way to place a specific symbol in the Schematic window is to drop a generic symbol and edit (on screen) its name. The generic component name "W_Element" needs to be appended with the desired number of lines, for example changing it to "W_Element7" or "W_Element20".

> W_Element components are supported for HSPICE compatibility. For more information, see *Compatible Features and Limitations* (hspice).

Library: Analog/RF Network > Data Items

Parameters

Parameters

Name	Description	Units	Default
N	number of transmission lines	none	1
Length	length of all transmission lines	m	0.001
Model_type	model type: 0 – analytical (static matrices) model, 1- tabular model (frequency dependent matrices)	none	0
RLGCfile	name of the file containing static RLGC data (for Model_type=0)	none	none
Lfile	name of the file containing frequency dependent **L** matrix (for Model_type=1)	none	none
Cfile	name of the file containing frequency dependent **C** matrix (for Model_type=1)	none	none
Rfile	name of the file containing frequency dependent **R** matrix (for Model_type=1)	none	none
Gfile	name of the file containing frequency dependent **G** matrix (for Model_type=1)	none	none
Ldata	array defining the **L** matrix (either static for Model_type=0 or frequency dependent for Model_type=1)	H/m	none
Cdata	array defining the **C** matrix (either static for Model_type=0 or frequency dependent for Model_type=1)	F/m	none
Rdata	array defining the **R** matrix (either static DC matrix **Rdc** for Model_type=0 or frequency dependent matrix **R** for Model_type=1)	Ohm/m	0.0
Gdata	array defining the **G** matrix (either static DC matrix **Gdc** for Model_type=0 or frequency dependent matrix **G** for Model_type=1)	S/m	0.0
RSdata	array defining the **Rs** matrix (applies to Model_type=0)	Ohm/m	0.0
GDdata	array defining the **Gd** matrix (applies to Model_type=0)	S/m	0.0
Fgd	cut-off frequency of dielectric loss (applies to Model_type=0) – zero means infinity	Hz	infinity
LfreqSweep	type of frequency sweep in Ldata array or in Lfile: 0 – linearFromTo, 1 – logFromTo, 2 – logPtsPerDec, 3 – logPtsPerOct, 4 – points (applies to Model_type=1)	none	0
LfreqStart	frequency of the first dataset in the Ldata list or in Lfile (applies to Model_type=1)	Hz	0.0
LfreqStop	frequency of the last dataset in the Ldata list or in Lfile (applies to Model_type=1)	Hz	none
LfreqsPerDec	starting from LfreqStart, the number of frequencies per decade corresponding to the subsequent datasets in Ldata or in Lfile (applies to Model_type=1)	none	10
LfreqsPerOct	starting from LfreqStart, the number of frequencies per octave corresponding to the subsequent datasets in Ldata or in Lfile (applies to Model_type=1)	none	10
CfreqSweep	type of frequency sweep in Cdata array or in Cfile: 0 – linearFromTo, 1 – logFromTo, 2 – logPtsPerDec, 3 – logPtsPerOct, 4 – points (applies to Model_type=1)	none	0
CfreqStart	frequency of the first dataset in the Cdata list or in Cfile (applies to Model_type=1)	Hz	0.0
CfreqStop	frequency of the last dataset in the Cdata list or in Cfile (applies to Model_type=1)	Hz	none
CfreqsPerDec	starting from CfreqStart, the number of frequencies per decade corresponding to the subsequent datasets in Cdata or in Cfile (applies to Model_type=1)	none	10
CfreqsPerOct	starting from CfreqStart, the number of frequencies per octave corresponding to the subsequent datasets in Cdata or in Cfile (applies to Model_type=1)	none	10
RfreqSweep	type of frequency sweep in Rdata array or in Rfile: 0 – linearFromTo, 1 – logFromTo, 2 – logPtsPerDec, 3 – logPtsPerOct, 4 – points (applies to Model_type=1)	none	0
RfreqStart	frequency of the first dataset in the Rdata list or in Rfile (applies to Model_type=1)	Hz	0.0
RfreqStop	frequency of the last dataset in the Rdata list or in Rfile (applies to	Hz	none

	Model_type=1)		
RfreqsPerDec	starting from RfreqStart, the number of frequencies per decade corresponding to the subsequent datasets in Rdata or in Rfile (applies to Model_type=1)	none	10
RfreqsPerOct	starting from RfreqStart, the number of frequencies per octave corresponding to the subsequent datasets in Rdata or in Rfile (applies to Model_type=1)	none	10
GfreqSweep	type of frequency sweep in Gdata array or in Gfile: 0 – linearFromTo, 1 – logFromTo, 2 – logPtsPerDec, 3 – logPtsPerOct, 4 – points (applies to Model_type=1)	none	0
GfreqStart	frequency of the first dataset in the Gdata list or in Gfile (applies to Model_type=1)	Hz	0.0
GfreqStop	frequency of the last dataset in the Gdata list or in Gfile (applies to Model_type=1)	Hz	none
GfreqsPerDec	starting from GfreqStart, the number of frequencies per decade corresponding to the subsequent datasets in Gdata or in Gfile (applies to Model_type=1)	none	10
GfreqsPerOct	starting from GfreqStart, the number of frequencies per octave corresponding to the subsequent datasets in Gdata or in Gfile (applies to Model_type=1)	none	10

Range of Usage

The diagonal entries of the L and C matrices must be positive. The off-diagonal elements of the C, G and Gd matrices must be non-positive. The diagonal entries of R and G matrices must be non-negative. See Notes/Equations for further details.

Notes/Equations

1. The available connection nodes include a reference line, and thus the total number of connection nodes is 2(N+1).
2. The matrices can be static (Model_type=0, default) or tabular frequency dependent (Model_type=1). For Model_type=0, all the matrices are specified either directly, or in a file. For Model_type=1, the data format can be set independently for each matrix.
3. All the matrices are assumed to be symmetric and real-valued. Each matrix must be specified using the lower triangular part only, which contains the diagonal and sub-diagonal elements. The order is row centric, that is the first element of the first row, followed by the first two elements of the second row, then the first three elements of the third row, etc. The total number of the matrix element specified must be exactly N(N+1)/2.
4. If a file name parameter is specified, any corresponding, directly specified data will be ignored. For example, if RLGCfile is specified and Model_type=0 then all of Ldata, Cdata, Rdata, Gdata, RSdata and GDdata arrays, even if specified will be ignored. If any of the optional matrices are missing in the file, their default zero values will be used.
5. Any of the matrices specified directly using the parameters Ldata, Cdata, Rdata, Gdata, RSdata and GDdata need to use the list() function with an appropriate number of entries (size of the array). For further details see Notes 6, 7 and 8.
6. For Model_type=0, the size of the arrays Ldata, Cdata, Rdata, Gdata, RSdata and GDdata arrays must be exactly N(N+1)/2.
7. For Model_type=1, the frequency dependent tabular data consists of a number of matrices (datasets). The corresponding frequencies are implicitly defined by the type and the details of the frequency sweep. These sweeps are set individually for each of the matrices. For linear and logarithmic sweeps, the first entry in the data list, or in the data file, is the number K of datasets to follow. Thus the total number of numeric entries must be exactly 1+K*N*(N+1)/2.
8. For Model_type=1 and for the "points" type of the sweep, the data consists of pairs, where each pair is formed by a frequency value followed by the matrix entries at that

frequency. The first data entry in the data list (or a file) is the number of such pairs (the number of points). Thus, for K points the data must contain exactly 1+K*(1+N*(N+1)/2) numeric entries.

9. The format of the data files is restricted. The files may only contain comment lines (or partial lines) starting with the asterisk character "*", and the numeric data without any units or scaling factors. The data can be delimited by one or more of the following characters: " \t\n,;()[]{}". For further details see Notes 10 and 11.
10. For Model_type=0, only the RLGCfile parameter is relevant. The first numeric entry in an RLGCfile must be an integer N specifying the number of lines and thus the size of all the matrices. Only the L and C matrices are required and thus the total number of numeric entries must be 1+K*N*(N+1)/2 where K is an integer between 2 and 6. The order of the matrices is L, C, Rdc, Gdc, Rs and Gd.
11. For Model_type=1, only Lfile, Cfile, Rfile and Gfile parameters are relevant. Each file contains frequency dependent data for one matrix type only. The number and the meaning of numeric entries in any of such files is exactly the same as in a directly specified array for the specified type of the frequency sweep, as in Notes 7 and 8, respectively.
12. All parameters defining frequency sweeps are relevant to Model_type=1 tabular data only.
13. For all types of log sweeps the start frequency must be specified and positive.
14. Stop frequency must be specified for the linear and for the "logFromTo" sweep types.
15. A positive integer number of points per decade, or per octave must be specified for the "logPtsPerDec" or "logPtsPerOct" type of sweeps, respectively.

Equations

R, L, G and C are N-dimensional matrices representing the per-unit-length parameters of the N coupled transmission lines. These parameters enter the telegrapher's equations describing the coupled lines. In the W-element model these matrices can be frequency dependent. Frequency scaling is applied differently for different model types.

Analytical model (Model_type=0)

For the static RLGC matrices (Model_type=0) the frequency dependence is analytically predefined, as follows.
At the frequency f, the resulting matrix R is defined as:

$$R = R_{dc} + R_s \cdot (1 + j) \cdot \sqrt{f}$$

where the first term models the ohmic resistances and the second term models the skin effect. Similarly, the resulting matrix G is defined as:

$$G = G_{dc} + \frac{f}{\sqrt{1 + \left(\frac{f}{Fgd}\right)^2}} G_d$$

where the first term models the dielectric leakage and the second term models dipole rotation related dielectric losses. Rdc, Rs, Gdc and Gd matrices are defined by the Rdata, RSdata, Gdata and GDdata parameters, respectively, or are given in the RLGCfile. Fgd is a component parameter. The L and C matrices are static and do not vary with frequency.

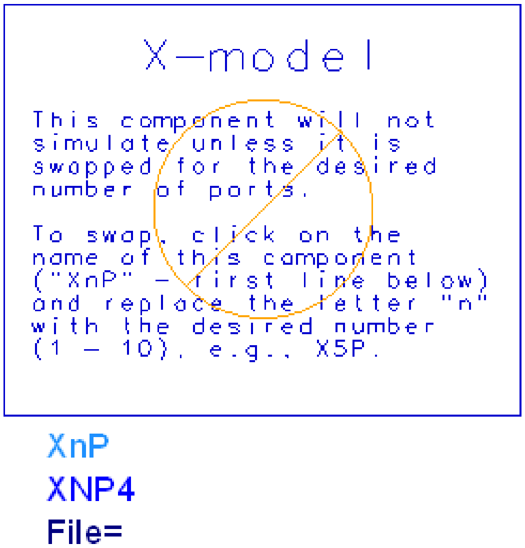
Tabular model (Model_type=1)

For the tabular data, the R, L, G and C matrices are determined by interpolation from the given data in Rdata, Ldata, Gdata and Cdata arrays, respectively (or in the Rfile, Lfile, Gfile or Cfile). The interpolator assumes that the data represents the frequency points established by the frequency sweep parameters.

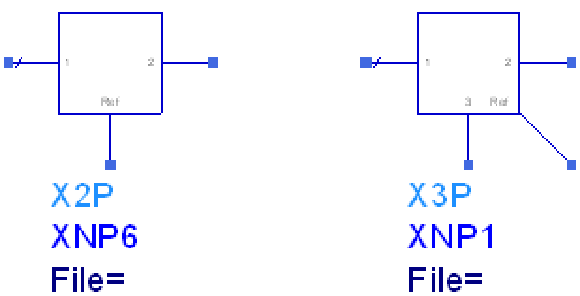
XnP Components (X1P - X10P)

Symbols

Generic Symbol (interface to specific symbols for 1-10 ports)



Specific Symbols (examples of a 2-port and 3-port symbol)



Please note that, except for the 2-port and 3-port symbols, the way to place a specific symbol in the Schematic window is to drop the generic symbol and edit (on screen) its name. The letter "n" in the generic component name "XnP" needs to be replaced by the desired number of ports, for example the numeric "5", leading to "X5P". Please also note that the labels "XNP6" and "XNP1" in the above examples are instance names and thus the numerics "6" and "1" have no meaning regarding the number of ports.
The ports of the XnP components have a common reference node. The ith port of the component is established by a pair of nodes with the i pin as the "plus" node and Ref pin as the "minus" node.

Parameters

Name	Description	Units	Default
File File Name	Name of data file containing X-parameters for this component	None	None
File Type (or Type)	File type: Dataset, GMDIF (Generic MDIF)	None	Dataset
Interpolation Method (or InterpMode)	Interpolation method: Linear or 0, Cubic Spline or 1, Cubic or 2	None	Linear
Interpolation Domain (or InterpDom)	Interpolation domain: Rectangular, Polar	None	Polar
Extrapolation Method (or ExtrapMode)	Extrapolation method: Interpolation Mode, Constant Extrapolation	None	Constant Extrapolation
EnableNormalization	Volterra normalization option (yes/no)	None	YES

Range of Usage

1. The XnP components should be used within the frequency range and the large-signal operating conditions covered by the data.
2. The XnP components can be used in all simulations. However, the XnP components are not intended for, and should not be used in general transient simulations. The availability of the components in transient simulations comes with a limited accuracy and is provided only to facilitate TAHB.

Notes/Equations

1. The XnP components facilitate simulation of behavioral models described in terms of n-port X-parameter data.
2. The data file (parameter "File") needs to contain X-parameter data for an n-port.
3. Files of Dataset type are binary and can be generated using the *X-Parameter Generator* (xparam). The **View Dataset** button can be used to see the information about the independent and dependent variables in the dataset. Files of GMDIF type are ASCII and can be generated using either the *X-Parameter Generator* (xparam) or the NVNA instrument. Version 2.0 X-parameter GMDIF files as well as earlier versions produced by NVNA are supported. The **Edit** button can used to view ASCII files. GMDIF files are suitable for cross-platform exchanges. Dataset files make the results available for Data Display plotting.
4. For information on data file formats, refer to *Working with Data Files* (cktsim) in the Using Circuit Simulators.
5. The X-parameter file format allows gaps in port numbering. For example, data can be present for ports 1, 2 and 5. The highest port number in the data establishes the required number of ports in the symbol - in the latter example the X5P component has to be used. The unused ports (3 and 4 in the example) will be open-circuited at the component side.
6. The XnP components use tabular data and, therefore, their simulation inherently involves interpolation. The Interpolation Method parameter selects the interpolation technique to be applied to the data. The Linear mode is the fastest and in most cases provides sufficiently good results.
7. The Interpolation Domain parameter refers to how complex data is interpolated and has two settings:
    - **Rectangular** mode: provide good results for most applications and
    - **Polar** (default) mode: is a better choice for frequency sweeps.
8. Extrapolation is not desired. However if needed, the Extrapolation Method parameter specifies the extrapolation mode. There are two possible settings:
    - **Interpolation Mode**: when extrapolation occurs, the interpolation mode specified by "Interpolation Method" is used for extrapolation.
    - **Constant Extrapolation**: when extrapolation occurs, no interpolation is

performed; the value of the nearest data point is returned.

9. The *EnableNormalization* parameter enables or disables Volterra normalization of the X-parameter data. If it is set to **yes** the data is normalized which improves the quality of the interpolation.
10. Some older X-parameter files (prior to Version 2.0) may contain data that is already normalized (such files used to be termed "PHD model files"). For such files EnableNormalization=no setting will be ignored, i.e., the already normalized data will be used during interpolation.
11. Without loss of generality, simplified X-parameter model equations are shown for a single-tone large signal at port 1 at the fundamental frequency.

RF case:

$$b_{ik} = X^{B}_{ik}(|a_{11}|)P^{k} + \sum_{(j,l)\neq(1,1)} (X^{S}_{ik,jl}(|a_{11}|)P^{k-l}a_{jl} + X^{T}_{ik,jl}(|a_{11}|)P^{k+l}a^{*}_{jl})$$

for
i,j=1,2,...,total_number_of_ports
k,l=1,2,...,total_number_of_harmonics
where,

a_{jl}	incident wave at input port *j* and harmonic *l* - the asterisk denotes complex conjugation
b_{ik}	reflected wave at output port *i* and harmonic *k*
$P = \frac{a_{11}}{\lvert a_{11} \rvert}$	phase of the incident wave at port 1 and harmonic 1; this incident wave serves as a phase reference (see *Reference Signal* (xparam))
X^{B}_{ik}	B-type X-parameter - measured reflected wave (power definition) at output port *i* and harmonic *k* at the large-signal operating conditions
$X^{S}_{ik,jl}$	S-type X-parameter providing the small-signal added-contribution to the reflected wave at output port *i* and harmonic *k* due to a small-signal incident wave at input port *j* and harmonic *l* measured under the large-signal operating conditions
$X^{T}_{ik,jl}$	T-type X-parameter providing the small-signal added-contribution to the reflected wave at output port *i* and harmonic *k* due to a phase-reversed small-signal incident wave at input port *j* and harmonic *l* measured under the large-signal operating conditions

The power definition of incident and reflected waves is used. The reference impedance for the waves can be different for different ports, and can be complex.

In the above equations only $|a_{11}|$ is shown as an independent variable. The list of independent variables usually contains other quantities such as the fundamental frequencies, other large-signal incident waves, port loadings as well as DC biasing conditions. For more information, refer to *X-parameter XnP File Format* (cktsim) in *Working with Data Files* (cktsim).

The DC equations can define either the current

$$I_{i} = X^{I}_{i}(|a_{11}|, V_{DC}) + \sum_{(j,l)\neq(1,1)} \mathrm{Re}(X^{Y}_{i,jl}(|a_{11}|, V_{DC})a_{jl})$$

or the voltage

$$V_{i} = X^{V}_{i}(|a_{11}|, I_{DC}) + \sum_{(j,l)\neq(1,1)} \mathrm{Re}(X^{Z}_{i,jl}(|a_{11}|, I_{DC})a_{jl})$$

where

I_i	DC current at output port *i*
V_i	DC voltage at output port *i*
V_{DC}	applied input port voltage (e.g., a DC voltage source)
I_{DC}	applied input port current (e.g., a DC current source)
X_i^I	I-type X-parameter - DC current measured at output port *i* under the large-signal operating conditions
X_i^V	V-type X-prameter - DC voltage measured at output port *i* under the large-signal operating conditions
$X_{i,jl}^Y$	Y-type X-parameter providing the small-signal contribution to the DC current at output port *i* due to a small-signal incident wave at input port *j* and harmonic *l* measured under the large-signal operating conditions
$X_{i,jl}^Z$	Z-type X-parameter providing the small-signal contribution to the DC voltage at output port *i* due to a small-signal incident wave at input port *j* and harmonic *l* measured under the large-signal operating conditions

12. Depending on what X-parameters are present in the X-parameter file the ports can be considered as unused, RF or DC_only, which can be categorized further as:
    - RF_no_DC
    - RF_with_VDC
    - RF_with_IDC
    - VDC_only
    - IDC_only

    Ports that do not have any X-parameters associated with, are unused and are kept open-circuited at all frequencies.
    A port is DC_only if there are no X-parameters of type B, S or T associated with that port, and no Y or Z type X-parameters for which it is an input port.
    A port is a VDC port if VDC applied to that port is one of the independent variables and/or there exists the X-parameter of type I associated with that port, and/or there exist X-parameters of type Y for which it is the output port. X-parameters of type V or Z (output port) are not allowed for VDC ports.
    Similarly, a port is an IDC port if IDC applied to that port is one of the independent variables and/or there exists the X-parameter of type V associated with that port, and/or there exist X-parameters of type Z for which it is the output port. X-parameters of type I or Y (output port) are not allowed for IDC ports.
    VDC_only ports are kept open-circuited at all non-zero frequencies.
    IDC_only ports are kept short-circuited at all non-zero frequencies.
    RF ports are matched at all frequencies for which there are no X-parameters associated with.
    RF_only ports are short-circuited at DC.
13. This component does not generate any noise.

References

1. D. E. Root *et al*., "Broad-Band Poly-Harmonic Distortion (PHD) Behavioral Models From Fast Automated Simulations and Large-Signal Vectorial Network Measurements," *IEEE Trans. MTT*, vol. 53, no. 11, pp. 3656-3664, November 2005.
2. J. Verspecht and D. E. Root, "Poly-Harmonic Distortion Modeling," *IEEE Microwave Theory and Techniques Microwave Magazine*, pp. 44-57, June, 2006.
3. J. Verspecht, D. Gunyan, J. Horn, J. Xu, A. Cognata, and D.E. Root, "Multi-tone, Multi-Port, and Dynamic Memory Enhancements to PHD Nonlinear Behavioral Models from Large-Signal Measurements and Simulations," 2007 IEEE MTT-S Int. Microwave

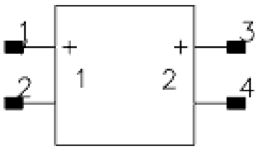
Symp. Dig., (Honolulu, HI), pp. 969-972, June 2007.
4. J. Horn *et al.*, "X-parameter Measurement and Simulation of a GSM Handset Amplifier", Proc. 3rd European Microwave Integrated Circuits Conf., (Amsterdam, The Netherlands), pp. 135-138, October 2008.

Equation-Based Linear Components

- *Chain (2-Port User-Defined Linear Chain)* (ccsim)
- *Hybrid (2-Port User-Defined Linear Hybrid)* (ccsim)
- *S1P Eqn to S6P Eqn (1- to 6-Port S-parameters, Equation-Based)* (ccsim)
- *Y1P Eqn to Y6P Eqn (1- to 6-Port Y-parameters, Equation-Based)* (ccsim)
- *Z1P Eqn to Z6P Eqn (1- to 6-Port Z-parameters, Equation-Based)* (ccsim)

Chain (2-Port User-Defined Linear Chain)

Symbol



Parameters

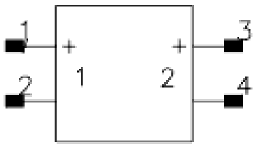
Name	Description	Units	Default
A	Reverse voltage gain (v1/v2 with i2=0)	None	None
B	Reverse transresistance (v1/i2 with v2=0)	Ohm	None
C	Reverse transconductance (i1/v2 with i2=0)	S	None
D	Reverse current gain (i1/i2 with v2=0)	None	None
ImpNoncausalLength	Non-causal function impulse response order	Integer	None
ImpMode	cOnvolution mode	Integer	None
ImpMax Freq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order	Integer	None
ImpWindow	Smoothing window	Integer	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None

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

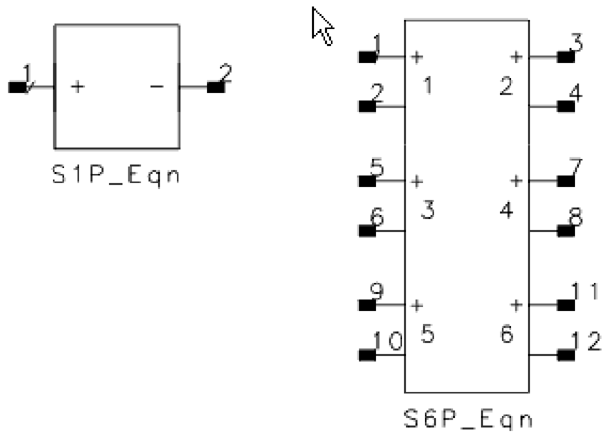
Name	Description	Units	Default
H11	Input impedance (v1/i1 with v2=0)	Ohm	None
H12	Reverse voltage gain (v1/v2 with i1=0)	None	None
H21	Forward current gain (i2/i1 with v2=0)	None	None
H22	Output conductance (i2/v2 with i1=0)		None
ImpNoncausalLength	Non-causal function impulse response order	integer	None
ImpMode	Convolution mode	integer	None
ImpMax Freq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order	Integer	None
ImpWindow	Smoothing window	None	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None

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.
5. When Hij is a rational transfer function of s = j*omega = j*2*PI*freq and specified in terms of poles and zeros using function *eval_pole_zero()* (expsim), the pole/zero data will be directly used in convolution to achieve better accuracy and efficiency.

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

Symbol



Parameters

Name	Description	Units	Default
S[i, j]	S-parameter in real and imaginary format	None	None
Z[i]	Port i reference impedance		None
NFmin	Minimum noise figure (obsolete-not used)	None	None
Rn	Noise resistance (obsolete-not used)	None	None
Sopt	Optimum noise match (obsolete-not used)	None	None
Temp	Device noise temperature	°C	
ImpNoncausalLength	Non-causal function impulse response order	Integer	None
ImpMode	Convolution mode	Integer	None
ImpMax Freq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order	Integer	None
ImpWindow	Smoothing window	Integer	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None

Range of Usage

1 ≤ i, j ≤ port number

Notes/Equations

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

value.

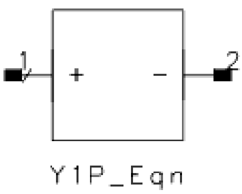
2. If a value is not entered for S[i,j], it is set to a zero default value (0, 0) and cannot be modified later. If S[i,j] is initially defined (even as zero), it can be modified and swept. It can also be swept indirectly, by sweeping a variable that it depends on. State currents are available for the port.
3. Port polarity is indicated by a minus (-) and a plus sign (+) on each port. The port can be made reciprocal by setting Recip=YES. By declaring the device to be reciprocal, S[i,j]] is always forced to equal S[j,i]. Only one of the two can be defined.
4. If NFmin, Sopt, and Rn are used to characterize noise in S2P_Eqn, the following relation must be satisfied for a realistic model.

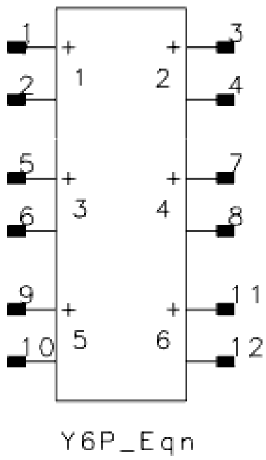
$$\frac{Rn}{Zo} \geq \frac{To(Fmin-1)|1+Sopt|^2}{T\,4} \frac{(1-|S_{11}|^2)}{|1-Sopt\ S_{11}|^2}$$

   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.
5. If the component temperature Temp is < -273°C, the component does not generate any noise. For S2P_Eqn only, if noisy 2-port parameters (minimum noise figure NFmin, optimum source reflection coefficient Sopt and effective noise source resistance Rn) are specified, these parameters are used to calculate the device's noise performance, independent of Temp. If the S-parameters describe a passive device, Temp and Twiss's theorem are used to calculate noise performance; if the S-parameters describe an active device, (i.e., the S-parameters are not passive), non-real noise is generated resulting in meaningless noise data. Further, if the network is not passive, a warning is issued to the Simulation Status window.
6. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.
7. S[i,j] can be made dependent on frequency by using the global variable freq. For example, you can use a brick wall lowpass filter by using S21=if(freq<1 GHz), then 1 else 0.

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

Symbol





Parameters

Name	Description	Units	Default
Y[i,j]	Y-parameter in real and imaginary format		
Temp	Temperature	°C	
ImpNoncausalLength	Non-causal function impulse response order	Integer	None
ImpMode	Convolution mode	Integer	None
ImpMax Freq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order	Integer	None
ImpWindow	Smoothing window	Integer	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None

Range of Usage

1 ≤ i, j ≤ port number

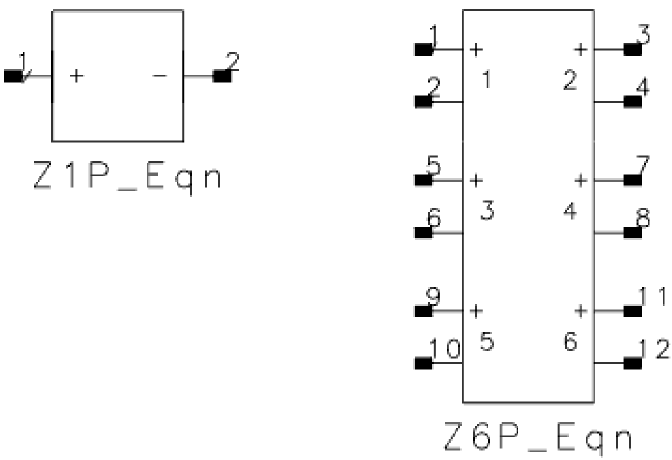
Notes/Equations

1. To enter a value for Y[i,j], use the syntax a+j*b or complex (a,b). Expressions can be used as an entry, and the first syntax is in fact a special case with j being a reserved symbol representing complex(0,1). Thus, the first syntax can be further simplified if either real or imaginary part is zero. For example, you can enter just Y[1,1]=3 instead of Y[1,1]=3+j*0. Also, subtraction, negative numbers and parentheses can be used as desired. The following four entries Y[1,1]=0+j*2, Y[1,1]=j*2, Y[1,1]=0-j*(-2) Y[1,1]=-j*(-2) are all supported and represent the same value. Similarly, the entries Y[1,1]=-j*j and Y[1,1]=1.0 represent the same value.
2. If a value is not entered for Y[i,j], it is set to a zero default value (0, 0) and cannot be modified later. If Y[i,j] is initially defined (even as zero), it can be modified and swept. It can also be swept indirectly, by sweeping a variable that it depends on. State currents are available for the port.

3. Port polarity is indicated by a minus (-) and a plus sign (+) on each port. The port can be made reciprocal by setting Recip=YES. By declaring the device to be reciprocal, Y[i,j]] is always forced to equal Y[j,i]. Only one of the two can be defined.
4. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
   Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
   If ImpMode, ImpMaxFreq, or ImpMaxOrder are not specified, they default to the global ImpMode specified by the transient analysis controller item.
5. Y[i,j] can be made dependent on frequency by using the global variable freq.
6. When Y[i,j] is a rational transfer function of s = j*omega = j*2*PI*freq and specified in terms of poles and zeros using function *eval_pole_zero()* (expsim), the pole/zero data will be directly used in convolution to achieve better accuracy and efficiency.

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

Symbol



Parameters

Name	Description	Units	Default
Z[i,j]	Z-parameter in real and imaginary format	None	None
C[i]	Port 1 controlling current (refer to Notes)	None	None
Temp	Temperature	°C	
ImpNoncausalLength	Non-causal function impulse response order	Integer	None
ImpMode	Convolution mode	Integer	None
ImpMax Freq	Maximum frequency to which device is evaluated		None
ImpDeltaFreq	Sample spacing in frequency		None
ImpMaxOrder	Maximum impulse response order	Integer	None
ImpWindow	Smoothing window	Integer	None
ImpRelTol	Relative impulse response truncation factor	None	None
ImpAbsTol	Absolute impulse response truncation factor	None	None

Range of Usage

1 ≤ i, j ≤ port number

Notes/Equations

1. To enter a value for Z[i,j], use the syntax a+j*b or complex (a,b). Expressions can be used as an entry, and the first syntax is in fact a special case with j being a reserved symbol representing complex(0,1). Thus, the first syntax can be further simplified if either real or imaginary part is zero. For example, you can enter just Z[1,1]=3 instead of Z[1,1]=3+j*0. Also, subtraction, negative numbers and parentheses can be used as desired. The following four entries Z[1,1]=0+j*2, Z[1,1]=j*2, Z[1,1]=0-j*(-2) Z[1,1]=-j*(-2) are all supported and represent the same value. Similarly, the entries Z[1,1]=-j*j and Z[1,1]=1.0 represent the same value.
2. If a value is not entered for Z[i,j], it is set to a zero default value (0, 0) and cannot be modified later. If Z[i,j] is initially defined (even as zero), it can be modified and

swept. It can also be swept indirectly, by sweeping a variable that it depends on. State currents are available for the port.

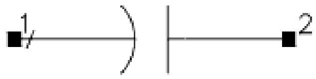
3. Port polarity is indicated by a minus (-) and a plus sign (+) on each port. The port can be made reciprocal by setting Recip=YES. By declaring the device to be reciprocal, Z[i,j]] is always forced to equal Z[j,i]. Only one of the two can be defined.
4. Allowed values for ImpMode are 1 (Discrete) and 2 (PWL Continuous).
Allowed values for ImpWindow are 0 (Rectangle) and 1 (Hanning).
If these values are not specified, they default to the corresponding global parameter values specified by the transient analysis controller item.
5. The C[i] parameter can be used to model the mutual coupling between ZnP_Eqn and other components in the circuit. For example, Z1P_Eqn_A is used to model a one-port block and Z1P_Eqn_B is used to model another one-port block. C[1] can be used to model the mutual coupling between Z1P_Eqn_A and Z1P_Eqn_B.
6. Z[i,j] can be made dependent on frequency by using the global variable freq.

Lumped Components

- *C (Capacitor)* (ccsim)
- *CAPP2 Conn (Chip Capacitor (Connector Artwork))* (ccsim)
- *CAPP2 Pad1 (Chip Capacitor (Pad Artwork))* (ccsim)
- *CAPP2 Space (Chip Capacitor (Space Artwork))* (ccsim)
- *CAPQ (Capacitor with Q)* (ccsim)
- *C Conn (Capacitor (Connector Artwork))* (ccsim)
- *C dxdy (Capacitor (Delta X - Delta Y))* (ccsim)
- *C Model (Capacitor Model)* (ccsim)
- *C Pad1 (Capacitor (Pad Artwork))* (ccsim)
- *CQ Conn (Capacitor with Q (Connector Artwork))* (ccsim)
- *CQ Pad1 (Capacitor with Q (Pad Artwork))* (ccsim)
- *CQ Space (Capacitor with Q (Space Artwork))* (ccsim)
- *C Space (Capacitor (Space Artwork))* (ccsim)
- *DC Block (DC Block)* (ccsim)
- *DC Feed (DC Feed)* (ccsim)
- *DICAP (Dielectric Laboratories Di-cap Capacitor)* (ccsim)
- *DILABMLC (Dielectric Laboratories Multi-Layer Chip Capacitor)* (ccsim)
- *InDQ2 (Inductor with Q)* (ccsim)
- *INDQ (Inductor with Q)* (ccsim)
- *L (Inductor)* (ccsim)
- *L Conn (Inductor (Connector Artwork))* (ccsim)
- *L Model (Inductor Model)* (ccsim)
- *L Pad1 (Inductor (Pad Artwork))* (ccsim)
- *LQ Conn (Inductor with Q (Connector Artwork))* (ccsim)
- *LQ Pad1 (Inductor with Q (Pad Artwork))* (ccsim)
- *LQ Space (Inductor with Q (Space Artwork))* (ccsim)
- *L Space (Inductor (Space Artwork))* (ccsim)
- *Mutual (Mutual Inductor)* (ccsim)
- *PLC (Parallel Inductor-Capacitor)* (ccsim)
- *PLCQ (Parallel Inductor-Capacitor with Q)* (ccsim)
- *PRC (Parallel Resistor-Capacitor)* (ccsim)
- *PRL (Parallel Resistor-Inductor)* (ccsim)
- *PRLC (Parallel Resistor-Inductor-Capacitor)* (ccsim)
- *R (Resistor)* (ccsim)
- *R Conn (Resistor (Connector Artwork))* (ccsim)
- *R dxdy (Resistor (Delta X - Delta Y))* (ccsim)
- *reluctance* (ccsim)
- *R Model (Resistor Model)* (ccsim)
- *R Pad1 (Resistor (Pad Artwork))* (ccsim)
- *R Space (Resistor (Space Artwork))* (ccsim)
- *Short (Short)* (ccsim)
- *SLC (Series Inductor-Capacitor)* (ccsim)
- *SLCQ (Series Inductor-Capacitor with Q)* (ccsim)
- *SMT Pad (SMT Bond Pad)* (ccsim)
- *SRC (Series Resistor-Capacitor)* (ccsim)
- *SRL (Series Resistor-Inductor)* (ccsim)
- *SRLC (Series Resistor-Inductor-Capacitor)* (ccsim)
- *TF3 (3-Port Transformer)* (ccsim)
- *TF (Transformer)* (ccsim)

C (Capacitor)

Symbol



Parameters

Name	Description	Units	Default
C	Capacitance	pF	1.0
Temp	Temperature	°C	None
Trise	Temperature rise over ambient	°C	None
Tnom	Nominal temperature	°C	25
TC1	Temperature coefficient; per degree Celsius	1/°C	None
TC2	Temperature coefficient; per degree Celsius squared	$1/°C^2$	None
wBV	Breakdown voltage warning		None
InitCond	Initial condition for transient analysis	None	None
Model	Name of a capacitor model to use	None	None
Width	Physical width for use with a model		None
Length	Physical length for use with a model		None
_M	Number of capacitors in parallel	None	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:
$$C' = C - Cj \times (Length - 2 \times Narrow) \times -(Width - 2 \times Narrow)$$

$$+ Cjsw \times 2 \times (Length + Width - 4 \times Narrow)$$

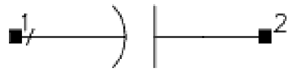
5. _M is used to represent the number of capacitors in parallel and defaults to 1. If a capacitor model is used, an optional scaling parameter Scale can also be defined on the model; it defaults to 1. The effective capacitance that will be simulated is C × Scale × M.
6. When InitCond is explicitly specified, 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.
7. The postive polarity of the initial conditions is applied to pin 1 (reference capacitor symbol above), therefore the terminal with the curved line is positive for a positive voltage.
8. DC Operating Point Information lists the DC operating point parameters that can be sent to the dataset.

DC Operating Point Information

Name	Description	Units
C	Capacitance	F

CAPP2_Conn (Chip Capacitor (Connector Artwork))

Symbol



Parameters

Name	Description	Units	Default
C	Capacitance	pF	1.0
TanD	Dielectric loss tangent	None	0.001
Q	Quality factor	None	50.0
FreQ	Resistance frequency for Q	MHz	300.0
FreqRes	Resistance frequency	MHz	500.0
Exp	Exponent for frequency dependence of Q	None	2.0
Temp	Temperature	°C	25

Range of Usage

C, Q, FreqQ, FreqRes ≥ 0

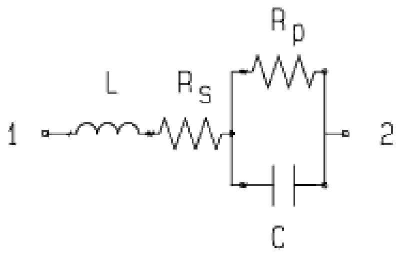
Notes/Equations

1. The series resistance Rs is determined by the Q and the parallel resistance Rp is determined by TanD.
   The frequency-dependence of Q is given by
   Q (f) = Q (FQ) (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

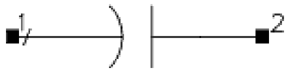
1. C. Bowick, RF Circuit Design, Howard Sams & Co., 1987.
2. The RF Capacitor Handbook, American Technical Ceramics Corp., September 1983.

Equivalent Circuit



CAPP2_Pad1 (Chip Capacitor (Pad Artwork))

Symbol



Parameters

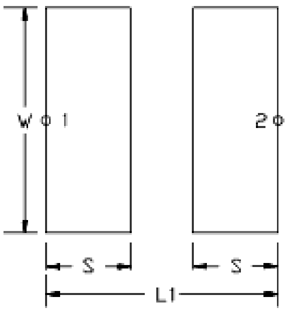
Name	Description	Units	Default
C	Capacitance	pF	1.0
TanD	Dielectric loss tangent	None	0.001
Q	Quality factor	None	50.0
FreqQ	Resistance frequency for Q	MHz	300.0
FreqRes	Resonance frequency	MHz	500.0
Exp	Exponent for frequency dependence of Q	None	2.0
W	(ADS Layout option) Width of pad	mil	25.0
S	(ADS Layout option) Spacing	mil	10.0
L1	(ADS Layout option) Pin-to-pin distance	mil	50.0
Temp	Temperature	°C	25

Range of Usage

C, Q, FreqQ, FreqRes ≥ 0

Notes/Equations

1. The series resistance Rs is determined by the Q and the parallel resistance Rp is determined by TanD.
   The frequency-dependence of Q is given by
   Q (f) = Q (FreqQ) (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.

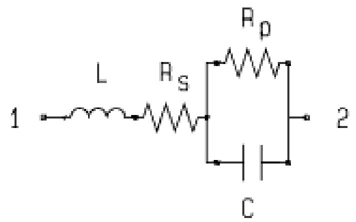


References

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

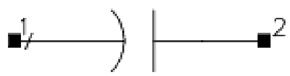
2. The RF Capacitor Handbook, American Technical Ceramics Corp., September 1983.

Equivalent Circuit



CAPP2_Space (Chip Capacitor (Space Artwork))

Symbol



Parameters

Name	Description	Units	Default
C	Capacitance	pF	1.0
TanD	Dielectric loss tangent	None	0.001
Q	Quality factor	None	50.0
FreQ	Resistance frequency for Q	MHz	300.0
FreqRes	Resistance frequency	MHz	500.0
Exp	Exponent for frequency dependence of Q	None	2.0
L1	(ADS Layout option) Pin-to-pin distance	mil	50.0
Temp	Temperature	°C	25

Range of Usage

C, Q, FreqQ, FR ≥ 0

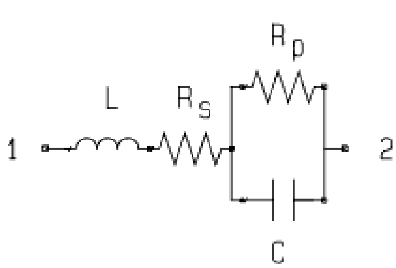
Notes/Equations

1. The series resistance Rs is determined by the Q and the parallel resistance Rp is determined by TanD.
   The frequency-dependence of Q is given by
   Q (f) = Q (FreqQ) (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

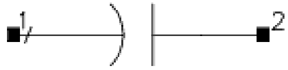
1. C. Bowick, RF Circuit Design, Howard Sams & Co., 1987.
2. The RF Capacitor Handbook, American Technical Ceramics Corp., September 1983.

Equivalent Circuit



CAPQ (Capacitor with Q)

Symbol



Parameters

Name	Description	Units	Default
C	Capacitance	pF	1.0
Q	Quality factor	None	50.0
F	Frequency at which Q is defined	MHz	100.0
Mode	Frequency dependence mode of Q; options (also refer to notes):	None	1
Temp	Temperature	°C	Celsius

Range of Usage

F ≥ 0

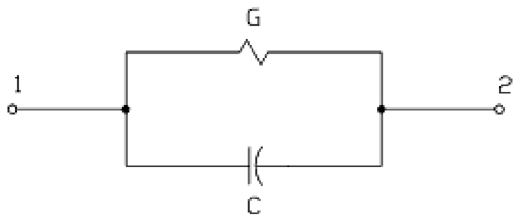
Notes/Equations

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

Mode Setting	Q	G
proportional to freq	Q(f) = Q(F) × f/F	G(f) = G(F)
proportional to sqrt (freq)	Q(f) = Q(F) × $\sqrt{f/F}$	G(f) = G(F) × $\sqrt{f/F}$
constant	Q(f) = Q(F)	G(f) = G(F) × f/F

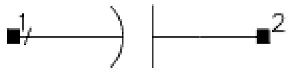
   If F is set to zero, then Q is assumed to be infinite; where f = simulation frequency, F = reference frequency, G = conductance of capacitor
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component has no default artwork associated with it.

Equivalent Circuit


G
1
2
C

C_Conn (Capacitor (Connector Artwork))

Symbol



Parameters

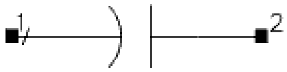
Name	Description	Units	Default
C	Capacitance	pF	1.0

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

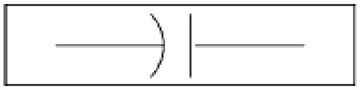
Name	Description	Units	Default
C	Capacitance	pF	1.0
dx	Delta X	mil	50.0
dy	Delta Y	mil	50.0
Temp	Temperature	°C	25

Notes/Equations

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

C_Model (Capacitor Model)

Symbol



Parameters

Name	Description	Units	Default
C (Cap)	Capacitance	pF	1.0
Cj	Capacitance per area		None
Cjsw (Capsw)	Sidewall or periphery capacitance		None
Length (L) †	Length		None
Width (W) †	Width		None
Narrow (Etch)	Length and width narrowing due to etching		None
Tnom	Nominal temperature	°C	25
Trise	Temperature rise over ambient	°C	None
TC1	Temperature coefficient; per degree Celsius	1/°C	None
TC2	Temperature coefficient; per degree Celsius squared	1/°C^2	None
wBV	Breakdown voltage (warning)		None
Scale (Scalec)	Capacitance scaling factor	None	1
Coeffs	Nonlinear capacitor polynomial coefficients (syntax: list(c1,c2,...))	None	None
AllParams	Data Access Component (DAC) Based Parameters	None	None

† Each instance parameter whose dimension contains a power of meter will be multiplied by the Scale to the same power. For example, a parameter with a dimension of *m* will be multiplied by *scale*^{1} and a parameter with a dimension of m^2 will be multiplied by *scale*^{2} . Note that only parameters whose dimensions contain meter are scaled. For example, a parameter whose dimension contains *cm* instead of meter is not scaled.

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 *Design Kit Development* (dkarch).

```
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 *ADS Simulator Input Syntax* (cktsim).
Example:

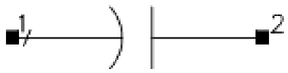
```
model  mimCap C_Model \
 Cje=1e-3 Cjsw=1e-10 Tc1=-1e-3 \
 Coeffs=list(1,2)
```

Notes/Equations

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. If C for the instance is not given, C for the model will be used.
The charge as a function of the applied voltage is:
$$Q(V) = C \times V \times \left(1 + \left(\frac{1}{2}\right) \times c1 \times V + \left(\frac{1}{3}\right) \times c2 \times V^2 + \ldots\right)$$

C_Pad1 (Capacitor (Pad Artwork))

Symbol

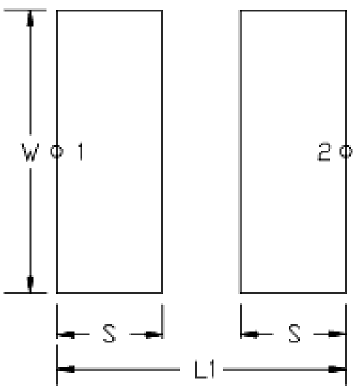


Parameters

Name	Description	Units	Default
C	Capacitance	pF	1.0
W	Width of pad	mil	25.0
S	Spacing	mil	10.0
L1	(ADS Layout option) Pin-to-pin distance	mil	50.0

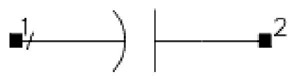
Notes/Equations

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



CQ_Conn (Capacitor with Q (Connector Artwork))

Symbol



Parameters

Name	Description	Units	Default
C	Capacitance	pF	1.0
Q	Quality factor	None	50.0
F	Frequency at which Q is defined	MHz	100.0
Mode	Frequency dependence mode of Q; options (also refer to notes): 1=proportional to freq 2=proportional to sqrt(freq) 3=constant	None	1
Temp	Temperature	°C	25

Range of Usage

F ≥ 0

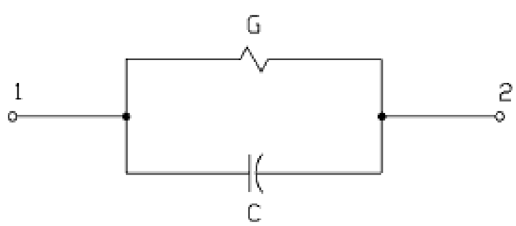
Notes/Equations

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

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

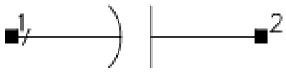
   If F is set to zero, then Q is assumed to be infinite
   where, f = simulation frequency
   F = reference frequency
   G = conductance of capacitor
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component is a single connection in layout. For example, it can be used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

Equivalent Circuit



CQ_Pad1 (Capacitor with Q (Pad Artwork))

Symbol



Parameters

Name	Description	Units	Default
C	Capacitance	pF	1.0
Q	Quality factor	None	50.0
F	Frequency at which Q is defined	MHz	100.0
Mode	Frequency dependence mode of Q; options (also refer to notes): 1=proportional to freq 2=proportional to sqrt(freq) 3=constant	None	1
W	(ADS Layout option) Width of pad	mil	25.0
S	(ADS Layout option) Spacing	mil	10.0
L1	(ADS Layout option) Pin-to-pin distance	mil	50.0
Temp	Temperature	°C	25

Range of Usage

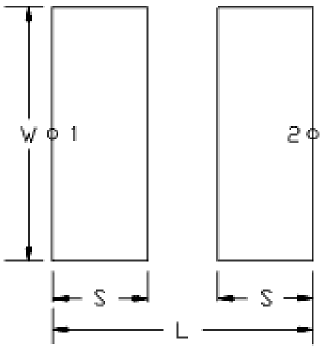
F ≥ 0

Notes/Equations

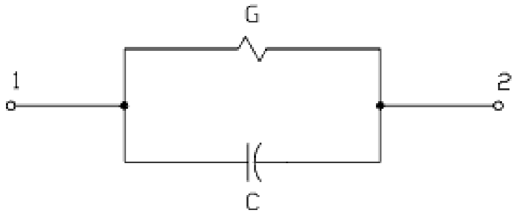
1. $$Q = \frac{B}{G} = \frac{2\pi f C}{G}$$
   where:

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

   If F is set to zero, then Q is assumed to be infinite. where f = simulation frequency
   F = reference frequency
   G = conductance of capacitor
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component's artwork is composed of two rectangular pads with pins on the outer edges, as shown:

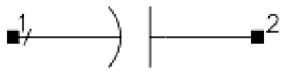


Equivalent Circuit



CQ_Space (Capacitor with Q (Space Artwork))

Symbol



Parameters

Name	Description	Units	Default
C	Capacitance	pF	1.0
Q	Quality factor	None	50.0
F	Frequency at which Q is defined	MHz	100.0
Mode	Frequency dependence mode of Q; options (also refer to notes): 1=proportional to freq 2=proportional to sqrt(freq) 3=constant	None	1
L1	(ADS Layout option) Pin-to-pin distance	mil	50.0
Temp	Temperature	°C	25

Range of Usage

F ≥ 0

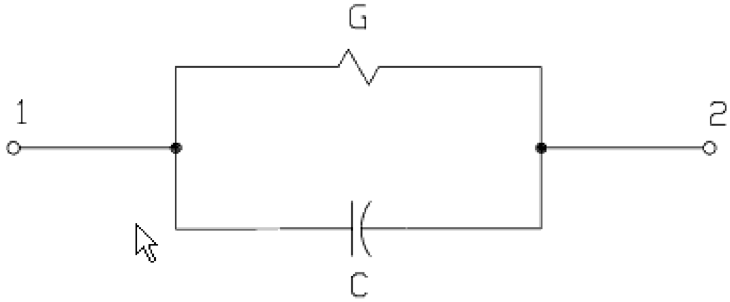
Notes/Equations

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

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

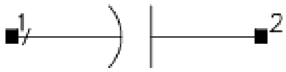
   If F is set to zero, then Q is assumed to be infinite. where f = simulation frequency
   F = reference frequency
   G = conductance of capacitor
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component is represented as a connected gap in layout---into which a custom artwork object can be inserted.

Equivalent Circuit


G
1
2
C

C_Space (Capacitor (Space Artwork))

Symbol



Parameters

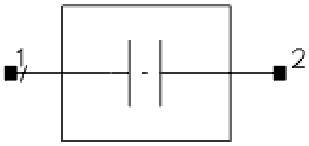
Name	Description	Units	Default
C	Capacitance	pF	1.0
L1	(ADS Layout option) Pin-to-pin distance	mil	50.0

Notes/Equations

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

DC_Block (DC Block)

Symbol



Parameters

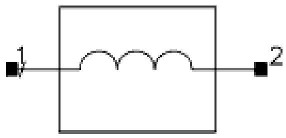
Name	Description	Units	Default
Mode	Integer values denote functionality:-1=blocks AC but feeds DC0=feeds both AC and DC1=blocks DC but feeds AC	Integer	1 †
C	DC block capacitance		None
L	DC feed inductance		None
Gain	Current gain	None	None
wImax	Maximum current (warning)		None
† Refer to DC_Feed and Short components for other default settings.			

Notes/Equations

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

DC_Feed (DC Feed)

Symbol



Parameters

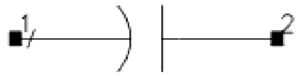
Name	Description	Units	Default
Mode	Integer values denote functionality:-1=blocks AC but feeds DC0=feeds both AC and DC1=blocks DC but feeds AC	None	-1 †
C	DC block capacitance		None
L	DC feed inductance		None
Gain	Current gain	None	None
wImax	Maximum current (warning)		None
† Refer to DC_Block and Short components for other default settings.			

Notes/Equations

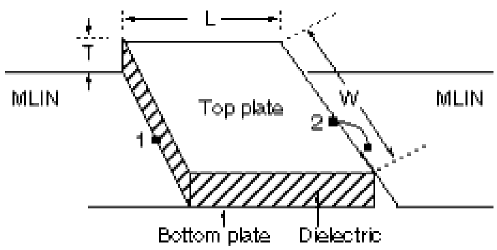
1. The C and L parameters are used for transient simulation only because short for DC_Feed is non-causal for transient simulation.

DICAP (Dielectric Laboratories Di-cap Capacitor)

Symbol



Illustration



Parameters

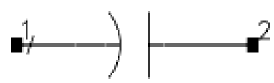
Name	Description	Units	Default
W	Width of metal plates and dielectric	mil	25.0
L	Length of metal plates and dielectric	mil	25.0
T	Thickness of dielectric	mil	4.0
Er	Dielectric constant	None	2.50
TanDeL	Dielectric loss tangent value at 1 MHz	None	0.001
R0	Series resistance at 1 GHz	Ohm	0.01
Cond1_Layer	(for Layout option) Layer on which the bottom plate is drawn	select from layer options	cond
Diel_Layer	(for Layout option) Layer on which the dielectric is drawn	select from layer options	diel
Cond2_Layer	(for Layout option) Layer on which the airbridge is drawn	select from layer options	cond2
Temp	Temperature	°C	25

Notes/Equations

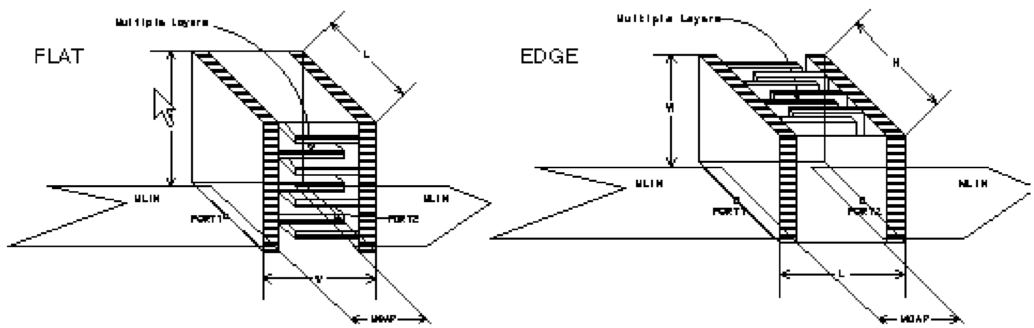
1. This is the Di-cap capacitor model by Dielectric Laboratories Incorporated; for the parameter values, please contact Dielectric Laboratories.
2. DICAP is a single-layer capacitor that behaves as lossy parallel plate transmission lines. Pin 1 is on the bottom metal plate; pin 2 is on the top metal 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

Name	Description	Units	Default
C0	Nominal capacitance	pF	1.0
TanDel	Dielectric loss tangent value at 1 MHz	None	0.001
RO	Bulk resistivity of termination at 1 MHz	Ohm	0.01
Rt	Termination loss resistance at 1 MHz	Ohm	0.01
Re	Electrode loss resistance per electrode at 1 GHz	Ohm	0.01
Mount	Mounting orientation: flat or edge	None	flat
W	Width of dielectric	mil	30.0
L	Length of dielectric	mil	60.0
H	Height of dielectric	mil	30.0
Cond1_Layer	(for Layout option) Layer on which the bottom plate is drawn	select from layer options	cond
Diel_Layer	(for Layout option) Layer on which the dielectric is drawn	select from layer options	diel
Cond2_Layer	(for Layout option) Layer on which the airbridge is drawn	select from layer options	cond2
Temp	Temperature	°C	25

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.

InDQ2 (Inductor with Q)

Symbol

Parameters

Name	Description	Units	Default
L	Inductance	nH	1.0
Q	Quality factor	None	50.0
F	Frequency at which Q is given	MHz	100.0
Mode	Mode loss mode for this device:	None	1
Rdc	Series constant resistance associated with the device, for Mode=2, 3, and 4 only	Ohm	0.0

Range of Usage

F ≥ 0

Notes/Equations

1. Rdc is the value of *R* at dc; that is, 0 Hz. It can be thought of as the *starting value* for *R* . In Mode 1, the value of Rdc is ignored; in modes 2, 3, and 4 it is used in the calculations.
2. The equivalent circuit for the INDQ2 component can be thought of as a complex impedance with the value
   Z = r1 + j × x1
   where r1 and x1 are calculated using the following equations:
   here
   freq = simulation frequency
   F = reference frequency
   Rdc = dc resistance
   - For Mode 1:
     ω = 2 × π × freq
     w_q = 2 × π × F
     r1 = w_q × L/Q + Rdc
     x1 = ω × L
   - For Mode 2:
     ω = 2 × π × freq
     w_q = 2 × π × F
     rt1 = w_q × L/Q - Rdc
     qt1 = w_q × L/rt1
     rac = sqrt(ω × w_q) × L/qt1
     r1 = Rdc + rac

x1 = rac + ω × L × (1-1/qt1)

- For Mode 3:
  ω = 2 × π × freq
  w_q = 2 × π × F
  rq1 = w_q × L/Q
  rq2 = sqrt(rq1 × rq1 - Rdc × Rdc)
  qt = w_q × L/rq2
  rac = ω × L/qt
  r1 = sqrt(Rdc × Rdc + rac × rac)
  x1 = ω × Lindq
- For Mode 4:
  ω = 2 × π × freq
  w_q = 2 × π × F
  rt1 = w_q × L/Q - Rdc
  qt1 = w_q × L/rt1
  rac = sqrt(ω × w_q) × L/qt1
  r1 = Rdc + rac
  x1 = ω × L
  Thus using r1 and x1 from the above calculations, we can say the final equivalent circuit is a Resistor and Inductor in series, where the resistance value is set by r1 and the inductance value would be x1/(2 × π × freq)

3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component has no default artwork associated with it.
5. This component does not generates noise.

INDQ (Inductor with Q)

Symbol

Parameters

Name	Description	Units	Default
L	Inductance	nH	1.0
Q	Quality factor	None	50
F	Frequency at which Q is given	MHz	100.0
Mode	Loss mode for this device:	None	1
Rdc	Resistance for modes 2 and 3	Ohm	0.0
Temp	Temperature	°C	25

Range of Usage

$F \geq 0$

Notes/Equations

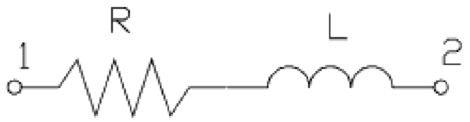
1. Rdc is the value of *R* at dc; that is, 0 Hz. It can be thought of as the *starting value* for *R* . In Mode 1, the value of Rdc is ignored; in modes 2 and 3 it is used in the calculations.
2. The equivalent circuit for the INDQ component can be thought of as a complex impedance with the value
   $Z = r1 + j \times x1$
   where r1 and x1 are calculated using the following equations:
   where
   freq = simulation frequency
   F =reference frequency
   Rdc = dc resistance (used in modes 2 and 3 only)
   - For Mode 1:
     $\omega = 2 \times \pi \times freq$
     $w\_q = 2 \times \pi \times F$
     $r1 = w\_q \times L/Q$
     $x1 = \omega \times L$
   - For Mode 2:
     $\omega = 2 \times \pi \times freq$
     $w\_q = 2 \times \pi \times F$
     $rt1 = w\_q \times L/Q - Rdc$
     $qt1 = w\_q \times L/rt1$

rac = sqrt(ω × w_q) × L/qt1
r1 = Rdc + rac
x1 = rac + ω × L × (1-1/qt1)

- For Mode 3:
  ω = 2 × π × freq
  w_q = 2 × π × F
  rq1 = w_q × L/Q
  rq2 = sqrt(rq1 × rq1 - Rdc × Rdc)
  qt = w_q × L/rq2
  rac = ω × L/qt
  r1 = sqrt(Rdc × Rdc + rac × rac)
  x1 = ω × Lindq
  Thus using r1 and x1 from the above calculations, we can say the final equivalent circuit is a Resistor and Inductor in series, where the resistance value is set by r1 and the inductance value would be x1/(2 × π × freq)

3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component has no default artwork associated with it.
5. For an inductor model with Q proportional to frequency, refer to *InDQ2 (Inductor with Q)* (ccsim).

Equivalent Circuit



L (Inductor)

Symbol

Parameters

Name	Description	Units	Default
L	Inductance	nH	1.0
R	Series resistance		None
Temp	Nominal temperature	°C	25
Trise	Temperature rise over ambient	°C	None
Tnom	Nominal temperature; see Notes	°C	None
TC1	Temperature coefficient; per degree Celsius	1/°C	None
TC2	Temperature coefficient; per degree Celsius squared	$1/°C^2$	None
InitCond	Initial condition for transient analysis	None	None
Noise	Noise generation option: yes=1, no=0	None	yes
Model	Model instance name	None	None
_M	Number of inductors in parallel	None	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:
   $\langle i^2 \rangle = 4kT/R$.
4. If a model name is given, then values that are not specified on the inductor instance are taken from the model values. Typical values that can be defaulted are the inductance, series resistance, nominal temperature and temperature coefficients.
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. *DC Operating Point Information* lists the DC operating point parameters that can be sent to the dataset.

Name	Description	Units
I	Current	A
L	Inductance	H

L_Conn (Inductor (Connector Artwork))

Symbol

Parameters

Name	Description	Units	Default
L	Inductance	nH	1.0

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	Units	Default
L	Inductance	nH	1.0
R	Series resistance		None
Tnom	Nominal temperature	°C	None
Trise	Temperature rise over ambient	°C	None
TC1	Temperature coefficient; per degree Celsius	1/°C	None
TC2	Temperature coefficient; per degree Celsius squared	$1/°C^2$	None
Scale (Scalei)	Scaling factor	None	1
Kf	Flicker noise coefficient	None	0.0
Af	Flicker noise exponent	None	0.0
Coeffs	Nonlinear inductor polynomial coefficients (syntax: list(c1,c2,...))	None	None
All Params	Data Access Component (DAC) based parameters	None	None

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 *Design Kit Development* (dkarch).

```
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 *ADS Simulator Input Syntax* (cktsim).
Example:

```
model bondWire L_Model
```

```
Tc1=20e-6
Coeffs=list(1,2)
```

Notes/Equations

1. This model supplies values for an inductor L. This allows some common inductor values to be specified in a model.
2. Kf and Af add flicker noise using the equation:
$$<i^2> \;= Kf \times I_{dc}^{Af}/f$$
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:
$$L(I) \;=\; \frac{dFlux(I)}{dl} = L(1 + c1 \times I + c2 \times I^2 + \dots)$$
where *L* is the inductance of the instance, and *ck* is the *k* -th entry in the Coeffs list.
If L for the instance is not given, L for the model will be used.
The branch flux as a function of the inductor current is:
$$Flux(I) \;=\; L \times I \times \left(1 + \left(\frac{1}{2}\right) \times c1 \times I + \left(\frac{1}{3}\right) \times c2 \times I^2 + \dots\right)$$

L_Pad1 (Inductor (Pad Artwork))

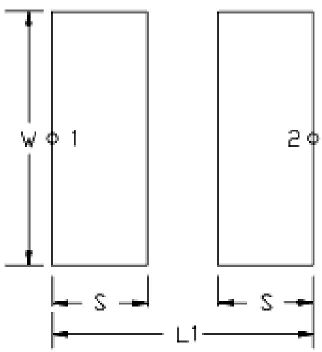
Symbol

Parameters

Name	Description	Units	Default
L	Inductance	nH	1.0
W	(ADS Layout option) Width of pad	mil	25.0
S	(ADS Layout option) Spacing	mil	10.0
L1	(ADS Layout option) Length	mil	50.0

Notes/Equations

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



LQ_Conn (Inductor with Q (Connector Artwork))

Symbol

Parameters

Name	Description	Units	Default
L	Inductance	nH	1.0
Q	Quality factor	None	50.0
F	Reference frequency for Q	MHz	100.0
Mode	Frequency dependence mode of Q; options (also refer to notes): 1=proportional to freq 2=proportional to sqrt(freq) 3=constant	None	1
Temp	Temperature	°C	25

Range of Usage

F ≥ 0

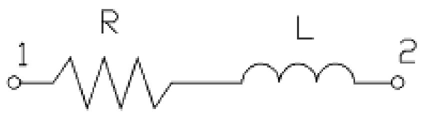
Notes/Equations

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

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

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

Equivalent Circuit



LQ_Pad1 (Inductor with Q (Pad Artwork))

Symbol

Parameters

Name	Description	Units	Default
L	Inductance	nH	1.0
Q	Quality factor	None	50.0
F	Reference frequency for Q	MHz	100.0
Mode	Loss mode for this device; options (also refer to notes): 1=proportional to freq 2=proportional to sqrt(freq) 3=constant	None	1
Temp	Temperature	°C	25
W	(ADS Layout option) Width of pad	mil	25.0
S	(ADS Layout option) Spacings	mil	10.0
L1	(ADS Layout option) Pin-to-pin distance	mil	50.0

Range of Usage

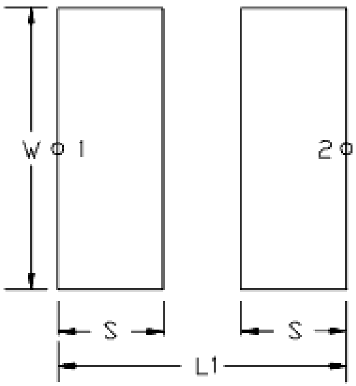
F ≥ 0

Notes/Equations

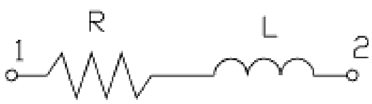
1. $$Q = \frac{2\pi f L}{R}$$
   where:

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

   If F is set to zero, then Q is assumed to be infinite. where f = simulation frequency
   F = reference frequency
   G = conductance of capacitor
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component's artwork is composed of two rectangular pads with pins on the outer edges, as shown:



Equivalent Circuit



LQ_Space (Inductor with Q (Space Artwork))

Symbol

Parameters

Name	Description	Units	Default
L	Inductance	nH	1.0
Q	Quality factor	None	50.0
F	Reference frequency for Q	MHz	100.0
Mode	Loss mode for this device; options (also refer to notes): 1=proportional to freq 2=proportional to sqrt(freq) 3=constant	None	1
Temp	Temperature	°C	25
L1	(ADS Layout option) Pin-to-pin distance	mil	50

Range of Usage

F ≥ 0

Notes/Equations

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

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

   If F is set to zero, then Q is assumed to be infinite. where f = simulation frequency
   F = reference frequency
   G = conductance of capacitor
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component is represented as a connected gap in layout---into which a custom artwork object can be inserted.

Equivalent Circuit

L_Space (Inductor (Space Artwork))

Symbol

Parameters

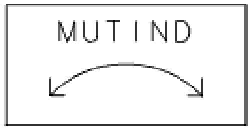
Name	Description	Units	Default
L	Inductance	nH	1.0
L1	(ADS Layout option) Length	mil	50.0

Notes/Equations

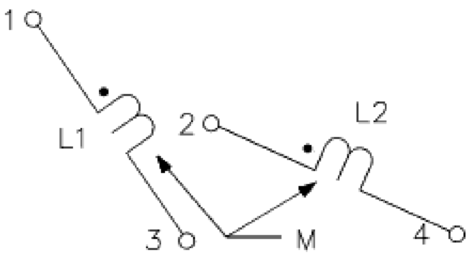
1. This component is represented as a connected gap in layout---into which a custom artwork object can be inserted.

Mutual (Mutual Inductor)

Symbol



Illustration



Parameters

Name	Description	Units	Default
K	Mutual inductor coupling coefficient; -1.0<= k <=1.0	None	0.5
M	Mutual inductance		None
Inductor1	ID of inductor 1	a string	None
Inductor2	ID of inductor 2	a string	None

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.

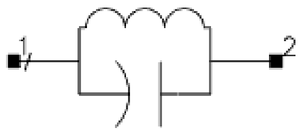
Note To edit *string* parameters on a schematic, highlight the parameter and enter a value enclosed with double quote symbols.

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.
5. *DC Operating Point Information* lists the DC operating point parameters that can be sent to the dataset.

Name	Description	Units
M	Mutual inductance	H

PLC (Parallel Inductor-Capacitor)

Symbol



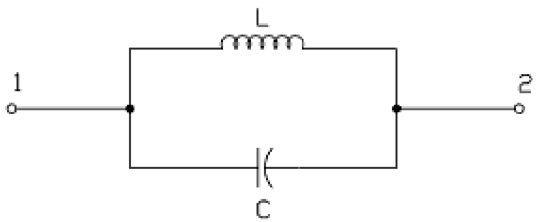
Parameters

Name	Description	Units	Default
L	Inductance	nH	1.0
C	Capacitance	pF	1.0

Notes/Equations

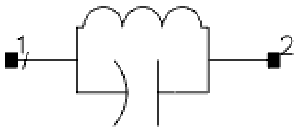
1. Use for high Q circuits rather than individual components in parallel.
2. This component has no default artwork associated with it.

Equivalent Circuit



PLCQ (Parallel Inductor-Capacitor with Q)

Symbol



Parameters

Name	Description	Units	Default
L	Capacitance	nH	120.0
Ql	Quality factor of inductor	None	50.0
Fl	Frequency at which Q is defined	MHz	100.0
Mode	Frequency dependence mode of inductor Q; options (also refer to notes):	None	1
C	Capacitance	pF	21.0
Qc	Quality factor of capacitor	None	100.0
Fc	Frequency at which capacitor Q is given	MHz	100.0
ModC	Frequency dependence mode of capacitor Q; options (also refer to notes):	None	1
Rdc	Resistance for modes 2 and 3	Ohm	0.0
Temp	Temperature	°C	25

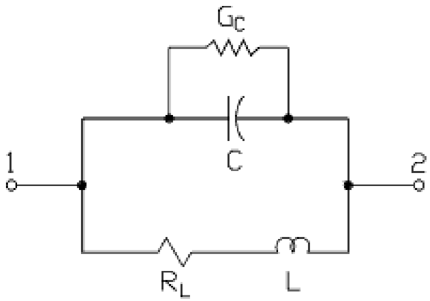
Notes/Equations

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) fs/Fl	proportional to freq	Qc(Fc) fs/Fc
proportional to sqrt (freq)	Q(f) = Q(F) $\sqrt{f/F}$	G(f) = G(F) $\sqrt{f/F}$	proportional to sqrt (freq)
constant	Ql(Fl)	constant	Qc(Fc)
where R = resistance of inductor G= conductance of capacitor fs = simulation frequency Fc, Fl = specified Q frequencies			

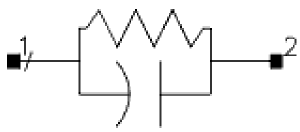
3. For time-domain analysis, the frequency-domain analytical model is used.
4. This component has no default artwork associated with it.

Equivalent Circuit



PRC (Parallel Resistor-Capacitor)

Symbol



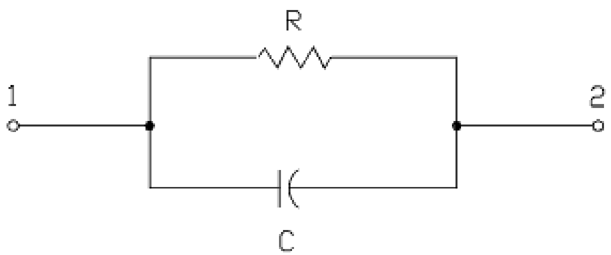
Parameters

Name	Description	Units	Default
R	Resistance	Ohm	1.0
C	Capacitance	pF	1.0
Temp	Temperature	°C	25

Notes/Equations

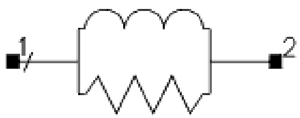
1. This component has no default artwork associated with it.

Equivalent Circuit



PRL (Parallel Resistor-Inductor)

Symbol



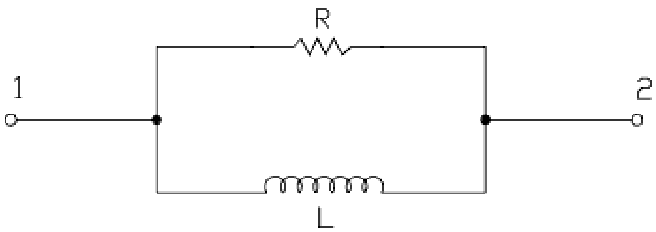
Parameters

Name	Description	Units	Default
R	Resistance	Ohm	1.0
C	Capacitance	nH	1.0
Temp	Temperature	°C	25

Notes/Equations

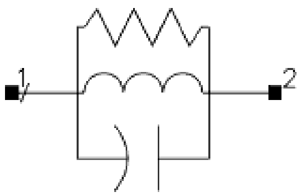
1. This component has no default artwork associated with it.

Equivalent Circuit



PRLC (Parallel Resistor-Inductor-Capacitor)

Symbol



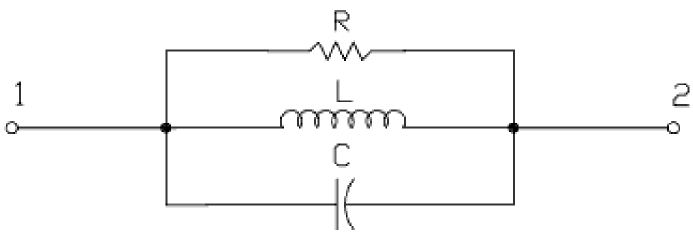
Parameters

Name	Description	Units	Default
R	Resistance	Ohm	1.0
L	Inductance	nH	1.0
C	Capacitance	pF	1.0
Temp	Temperature	°C	25

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	Units	Default
R	Resistance	Ohm	50
Temp	Temperature	°C	25
Trise	Temperature rise above ambient	°C	None
Tnom	Nominal temperature; see Note	°C	None
TC1	Temperature coefficient; per degree Celsius	1/°C	None
TC2	Temperature coefficient; per degree Celsius squared	$1/°C^2$	None
Noise	Resistor thermal noise option: yes=enable; no=disable	None	yes
wPmax	Maximum power dissipation (warning)		None
wImax	Maximum current (warning)		None
Model	Name of a resistor model to use	None	None
Width	Physical width for use with a model		None
Length	Physical length for use with a model		None
_M	Number of resistors in parallel	None	1
C	Capacitance	F	0.0

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 TempTnom, then the simulated resistance value is given by:
   $R' = R\,[1 + TC1\,(Temp - Tnom) + TC2\,(Temp - Tnom)^2]$
2. If Temp is not explicitly specified, it defaults to the global temperature specified in the options item.
3. The resistor generates thermal noise:
   $<i^2> = 4kT/R$
   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. *DC Operating Point Information* lists the DC operating point parameters that can be sent to the dataset.

Name	Description	Units
I	Current	A
Power	DC power dissipated	W
R	Resistance	Ohms
V	Resistor voltage	V

R_Conn (Resistor (Connector Artwork))

Symbol

Parameters

Name	Description	Units	Default
R	Resistance	Ohm	50
Temp	Temperature	°C	25

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

Name	Description	Units	Default
R	Resistance	Ohm	0
dx	Delta X	mil	50.0
dy	Delta Y	mil	50.0
Temp	Temperature	°C	25

Notes/Equations

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

reluctance

Parameters

Name	Description	Units	Default
reluctance	Reluctance matrix data. In general, K will be sparse and only non-zero values in the matrix need be given. Each matrix entry is represented by a triplet (r,c,val). See Note 4, 5, 6, 7.	H^{-1}	no value unless set by user
file	The data files should contain three columns of data. Each row should contain an (r,c,value) triplet separated by white space. See Note 4, 5, 6, 7.		no value unless set by user

Netlist Format

Inline form:

```
reluctance: <InstanceName> n1p n1n [n2p n2n ....... nNp nNn]
reluctance=list(r1, c1, val1 [, r2, c2, val2, ......] )
```

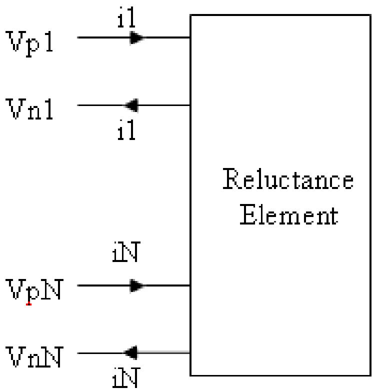
External file reference form:

```
reluctance: <InstanceName> n1p n1n [n2p n2n ....... nNp nNn]
file="<Filename>"
```

For more information about the ADS circuit simulator netlist format, refer to *ADS Simulator Input Syntax* (cktsim).

Notes/Equations

1. The reluctance element is used to model the parasitic inductive effect of interconnects, which is a must for the accurate simulation of high-speed digital, analog and RF designs.
2. The parasitic inductive effect is traditionally modeled by the inductance matrix. The reluctance matrix is the inverse of the inductance matrix. This new approach of using the reluctance matrix has the following advantages over the traditional one:
   - The reluctance matrix is easier to extract in a stable manner.
   - The reluctance matrix is much sparser than the inductance matrix and allows for more efficient simulation.
3. The reluctance element is a linear N-Port network, the relationship between the port voltages and currents is governed by the following matrix equation:

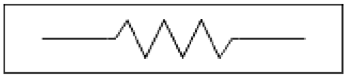


$$\begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1N} \\ K_{21} & K_{22} & & \\ \vdots & & \ddots & \\ K_{N1} & & & K_{NN} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} = \begin{bmatrix} di_1/dt \\ di_2/dt \\ \vdots \\ di_N/dt \end{bmatrix}$$

4. The value *r* and *c* are integers referring to a pair of inductors from the list of terminal nodes. If there are 2*N terminal nodes, there will be N inductors, and the *r* and *c* values must be in the range [1,N]. The *val* value is a reluctance value for the (*r* , *c*) matrix location.
5. Only terms along and above the diagonal are specified for the reluctance matrix. The simulator fills in the lower triangle to ensure symmetry. If you specify lower diagonal terms, the simulator converts that entry to the appropriate upper diagonal term. If multiple entries are supplied for the same (*r* , *c*) location, only the first one is used, and a warning will be issued indicating some entries are ignored.
6. All diagonal entries of the reluctance matrix must be assigned a positive value. Otherwise, a warning will be issued to point out the non-positive diagonal entries. The simulation will still proceed with the values specified by the user.
7. The reluctance matrix should be positive definite. The simulator checks that off-diagonal terms should be smaller in magnitude than the diagonal term in the same row/column.

R_Model (Resistor Model)

Symbol



Parameters

Name	Description	Units	Default
R	Resistance	Ohm	50
Rsh	Sheet resistance		None
Length (L) †	Length		None
Width (W) †	Width		None
Narrow	Length and width narrowing due to etching		None
Tnom	Nominal temperature	°C	None
Trise	Temperature rise over ambient	°C	None
TC1	Temperature coefficient; per degree Celsius	1/°C	None
TC2	Temperature coefficient; per degree Celsius squared	1/°C^2	None
wPmax	Maximum power dissipation (warning)		None
wImax	Maximum current (warning)		None
Scale (Scaler)	Resistance scaling factor	None	1
AllParams	Data Access Component (DAC) Based Parameters	None	None
Dw (Etch)	Width narrowing due to etching, in specified units	None	None
Dl (Etchl)	Length narrowing due to etching, in specified units	None	None
Kf	Flicker noise coefficient	None	0.0
Af	Flicker noise current exponent	None	0.0
Wdexp	Flicker noise W exponent	None	0.0
Ldexp	Flicker noise L exponent	None	0.0
Weexp	Flicker noise Weff exponent	None	0.0
Leexp	Flicker noise Leff exponent	None	0.0
Fexp	Flicker noise frequency exponent	None	1.0
Coeffs	Nonlinear resistor polynomial coefficients (syntax: list(c1,c2,...))	None	None
Shrink	Shrink Factor for Length and Width	None	1.0
Cap	Default parasitic capacitance	F	0.0
Capsw	Sidewall fringe capacitance	F/m	0.0
Cox	Bottom-Wall capacitance	F/m^2	0.0
Di	Relative dielectric constant	None	0.0
Tc1c	First order temperature coefficient for capacitance	1/°C	0.0
Tc2c	Second order temperature coefficient for capacitance	1/°C^2	0.0
Thick	Dielectric thickness	m	0.0
Cratio	Ratio to allcate parasitic capacitance	None	0.5

† Each instance parameter whose dimension contains a power of meter will be multiplied by the Scale to the same power. For example, a parameter with a dimension of *m* will be multiplied by $scale^1$ and a parameter with a dimension of m^2 will be multiplied by $scale^2$. Note that only parameters whose dimensions contain meter are scaled. For example, a parameter whose dimension contains *cm* instead of meter is not scaled.

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 *Design Kit Development* (dkarch).

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

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

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

Notes/Equations

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. Flicker noise is modeled using the equation:
   $$<i^2> = \frac{Kf \quad I_{DC}^{Af} / f^{Fexp}}{W^{Wdexp}\, L^{Ldexp}\, Weff^{Weexp}\, Leff^{Leexp}}$$
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:Leexp
   $$R(V) = \frac{dV}{dI} = \frac{R}{(1 + c1 \times V + c2 \times V^2 + \ldots)}$$
   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_Pad1 (Resistor (Pad Artwork))

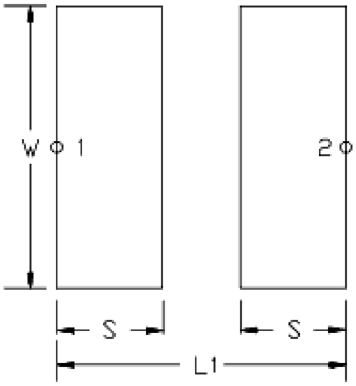
Symbol

Parameters

Name	Description	Units	Default
R	Resistance	Ohm	50.0
W	(ADS Layout option) Width of pad	mil	25.0
S	(ADS Layout option) Spacing	mil	10.0
L1	(ADS Layout option) Pin-to-pin distance	mil	50.0
Temp	Temperature	°C	25

Notes/Equations

1. For transient and convolution analysis, 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

Name	Description	Units	Default
R	Resistance	Ohm	50
L1	(ADS Layout option) Pin-to-pin distance	mil	50.0
Temp	Temperature	°C	25

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

Name	Description	Units	Default
Mode	Integer values denote functionality:	None	0
C	DC block capacitance		None
L	DC feed inductance		None
Gain	Current gain	None	None
SaveCurrent	Save branch current: yes or no	None	no
wImax	Maximum current (warning)		None

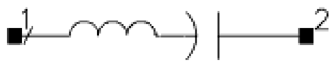
Notes/Equations

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. *DC Operating Point Information* lists the DC operating point parameters that can be sent to the dataset.

Name	Description	Units
I	Current	A

SLC (Series Inductor-Capacitor)

Symbol



Parameters

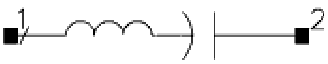
Name	Description	Units	Default
L	Inductance	nH	1.0
C	Capacitance	pF	1.0

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

Name	Description	Units	Default
L	Inductance	nH	120.0
Ql	Quality factor of inductor	None	50.0
Fl	frequency at which Q is defined	MHz	100
ModL	Frequency dependence mode of inductor Q; options (also refer to notes):	None	1
C	Capacitance	pF	21.0
Qc	Quality factor of capacitor	None	100.0
Fc	Frequency at which capacitor Q is given	MHz	100.0
ModC	Frequency dependence mode of capacitor Q; options (also refer to notes):	None	1
Rdc	Resistance for modes 2 and 3	Ohm	0.0
Temp	Temperature	°C	25

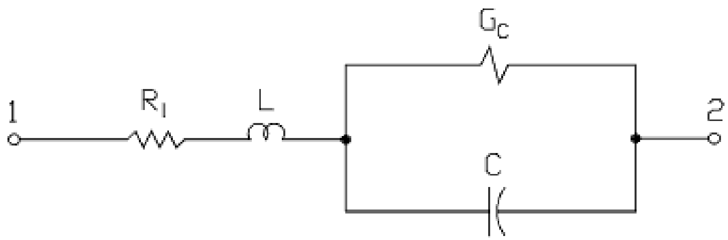
Notes/Equations

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) fs/Fl	proportional to freq	Qc(Fc) fs/Fc
proportional to sqrt(freq)	Ql(Fl) $\sqrt{f_s / Fl}$	proportional to sqrt(freq)	Qc(Fc) $\sqrt{f_s / Fc}$
constant	Ql(Fl)	constant	Qc(Fc)
where R = resistance of inductor G= conductance of capacitor fs = simulation frequency 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.

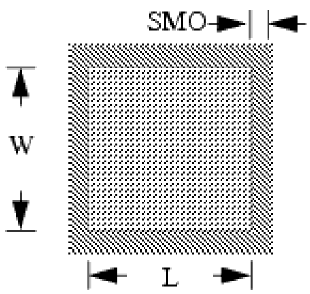
Equivalent Circuit



SMT_Pad (SMT Bond Pad)

Symbol

Illustration



Parameters

Name	Description	Units	Default
W	Width of pad	mil	10.0
L	Length of pad	mil	25.0
PadLayer	Layer of pad	select from layer options	bond
SMO	Solder mask overlap	mil	5.0
SM_Layer	Solder mask layer	select from layer options	solder_mask
PO	Pad offset from connection pin	mil	0

Range of Usage

W ≥ 0
L ≥ 0

Notes/Equations

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

Name	Description	Units	Default
R	Inductance	Ohm	1.0
C	Capacitance	pF	1.0
Temp	Temperature	°C	25

Notes/Equations

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

SRL (Series Resistor-Inductor)

Symbol

Parameters

Name	Description	Units	Default
R	Resistance	Ohm	1.0
C	Inductance	nH	1.0
Temp	Temperature	°C	25

Notes/Equations

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

SRLC (Series Resistor-Inductor-Capacitor)

Symbol

Parameters

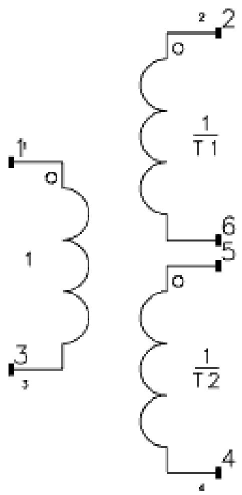
Name	Description	Units	Default
R	Resistance	Ohm	1.0
L	Inductance	nH	1.0
C	Capacitance	pF	1.0
Temp	Temperature	°C	25

Notes/Equations

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

TF3 (3-Port Transformer)

Symbol



Parameters

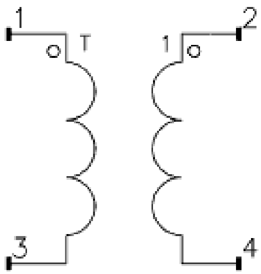
Name	Description	Units	Default
T1	Turn 1	None	1.00
T2	Turn 2	None	1.00

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.

TF (Transformer)

Symbol



Parameters

Name	Description	Units	Default
T	Turns ratio T1/T2	None	1.00

Notes/Equations

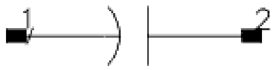
1. The turns ratio T is the ratio of turns in the primary to turns in the secondary (T:1). A turns ratio less than 1 describes a transformer in which there are more turns in the secondary than in the primary.
2. The TF component is a Hybrid component with the parameters H12=T and H21=-T. For more information on Hybrid components, refer to "Hybrid (2-Port User-Defined Linear Hybrid)".
3. Parasitic inductances of the primary and secondary are not modeled. To do this, use the mutual inductance component "Mutual" located on the *Lumped-Components* palette. The ends that are in phase are identified by a small open circle on the schematic symbol.
4. Because this is an ideal transformer, the impedance transformation is the same at DC as it is at nonzero frequencies.
5. This component passes DC.

Miscellaneous Circuit Components

- *CAPP2 (Chip Capacitor)* (ccsim)
- *CIND (Ideal Torroidal Inductor)* (ccsim)
- *RIND (Rectangular Inductor)* (ccsim)
- *XFERP (Physical Transformer)* (ccsim)
- *XFERRUTH (Ruthroff Transformer)* (ccsim)
- *XFERTAP (Tapped Secondary Ideal Transformer)* (ccsim)

CAPP2 (Chip Capacitor)

Symbol



Parameters

Name	Description	Units	Default
C	Capacitance	pF	1.0
TanD	Dielectric loss tangent	None	0.001
Q	Quality factor	None	50.0
FreqQ	Reference frequency for Q	MHz	300.0
FreqRes	Resonance frequency	MHz	500.0
Exp	Exponent for frequency dependence of Q	None	2.0
Temp	Temperature	°C	25

Range of Usage

C, Q, FreqQ, FreqRes ≥ 0

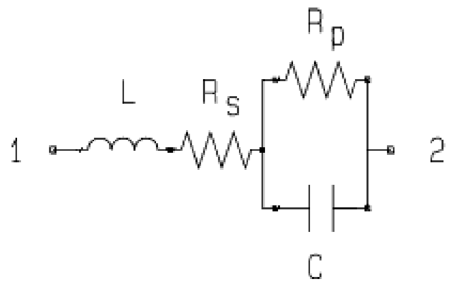
Notes/Equations

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 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.

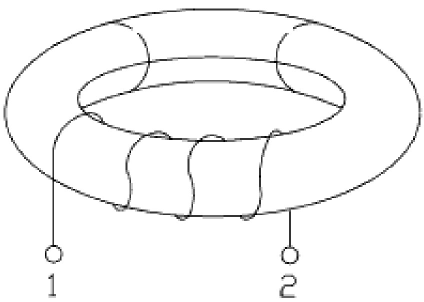
Equivalent Circuit



CIND (Ideal Torroidal Inductor)

Symbol

Illustration



Parameters

Name	Description	Units	Default
N	Number of units	None	10.0
AL	Inductance index	nH	0.15

Range of Usage

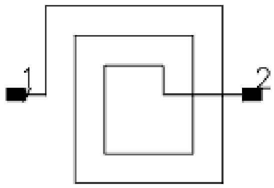
N, AL > 0

Notes/Equations

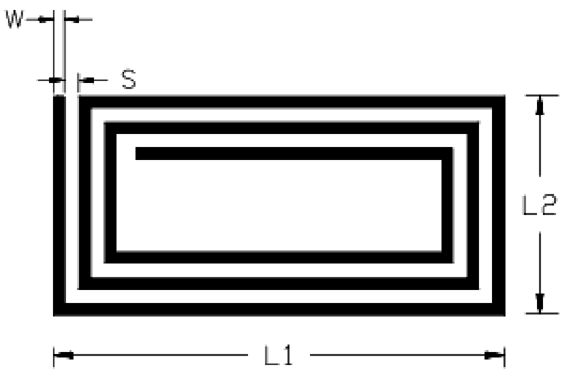
1. The inductance is given by
   $L = N^2 \times AL$
2. This component has no default artwork associated with it.

RIND (Rectangular Inductor)

Symbol



Illustration



Parameters

Name	Description	Units	Default
N	Number of turns (need not be an integer)	None	3.0
L1	Length of second outermost segment	mil	30.0
L2	Length of outermost segment	mil	20.0
W	Conductor width	mil	1.0
S	Conductor spacing	mil	1.0
T	Conductor thickness	mil	0.2
Rho	Conductor resistivity (relative to copper)	None	1.0
FR	Resonant frequency	MHz	10.0
Temp	Physical temperature	°C	25

Range of Usage

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

Notes/Equations

Notes/Equations

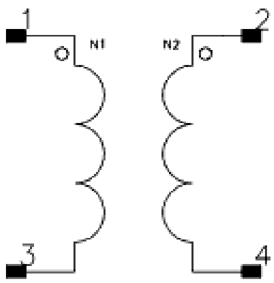
1. For time-domain analysis, an impulse response obtained from the frequency-domain analytical model is used.
2. This component has no default artwork associated with it.

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

Name	Description	Units	Default
N	Turns ratio N1/N2	None	1.0
Lp	Magnetizing inductance	nH	200.0
Rc	Core loss resistance	Ohm	1000.0
K	Coefficient of coupling	None	0.9
R1	Primary loss resistance	Ohm	5.0
R2	Secondary loss resistance	Ohm	5.0
C1	Primary capacitance	uF	25.0
C2	Secondary capacitance	uF	25.0
C	Interwinding capacitance	uF	0.1
Temp	Temperature	°C	25

Range of Usage

$0 < K < 1$

Notes/Equations

1. Primary leakage:

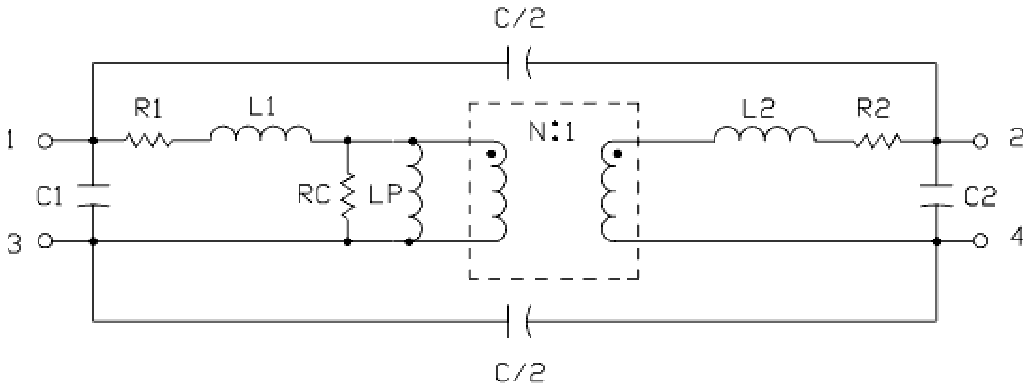
   $$L_1 = LP\left(\frac{1}{K} - 1\right)$$

   Secondary leakage:

   $$L_2 = \frac{L_1}{N^2}$$

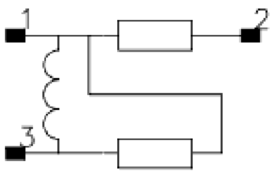
2. This component has no default artwork associated with it.

Equivalent Circuit



XFERRUTH (Ruthroff Transformer)

Symbol



Parameters

Name	Description	Units	Default
N	Number of turns	None	1.0
AL	Inductance index	nH	1.0
Z	Characteristic impedance of transmission line	Ohm	50.0
E	Electrical length of transmission line	deg	90
F	Reference frequency for electrical length	MHz	1.0

Range of Usage

N > 0
AL > 0

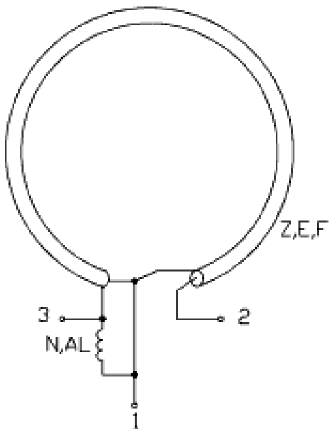
Notes/Equations

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

References

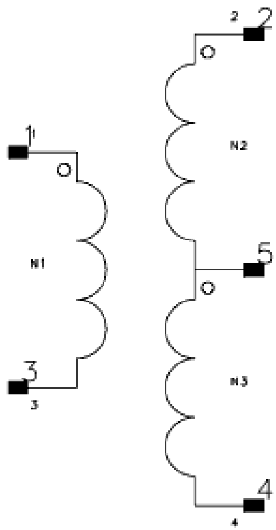
1. H. L. Krauss and C. W. Allen. "Designing torroidal transformers to optimize wideband performance," *Electronics,* pp. 113-116, August 16, 1973.
2. O. Pitzalis, Jr. and T. P. Couse. "Practical Design Information for Broadband Transmission Line Transformers," *Proceedings of the IEEE* , Vol. 56, No. 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


Z,E,F
3
N,AL
2
1

XFERTAP (Tapped Secondary Ideal Transformer)

Symbol



Parameters

Name	Description	Units	Default
N12	Turns ratio, N1/N2	None	1.414
N13	Turns ratio, N1/N3	None	1.414
L1	Primary winding inductance	nH	400.0
K	Coupling coefficient	None	0.9

Range of Usage

0 < K < 1

Notes/Equations

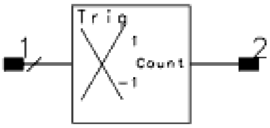
1. This component has no default artwork associated with it.

Probe Components

- *Counter (Counter Component)* (ccsim)
- *EyeDiff Probe (Differential Eye Measurement Probe)* (ccsim)
- *Eye Probe (Eye Measurement Probe)* (ccsim)
- *I Probe (Current Probe)* (ccsim)
- *OscPort2 (Differential Oscillator Port)* (ccsim)
- *OscPort (Grounded Oscillator Port)* (ccsim)
- *OscTest (Grounded Oscillator Test)* (ccsim)
- *P Probe2 (Differential Power Probe)* (ccsim)
- *P Probe (Grounded Power Probe)* (ccsim)
- *SP Probe (S-Parameter Probe)* (ccsim)
- *SProbe2 (SProbe2 Component)* (ccsim)
- *SProbePair2 (SProbePair2 Component)* (ccsim)
- *TimeDelta (Time Delta Component)* (ccsim)
- *TimeFrq (Time Frequency Component)* (ccsim)
- *TimePeriod (Time Period Component)* (ccsim)
- *TimeStamp (Time Stamp Component)* (ccsim)
- *WaveformStats (WaveformStats Component)* (ccsim)

Counter (Counter Component)

Symbol



Parameters

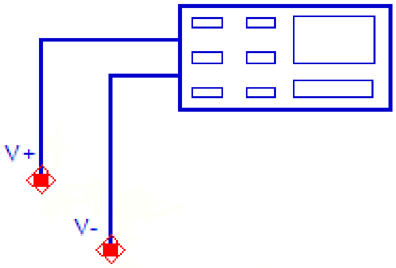
Name	Description	Units	Default
Direction	Direction one	None	1
Thresh	Threshold one	V	0

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.

EyeDiff_Probe (Differential Eye Measurement Probe)

Symbol



Parameters

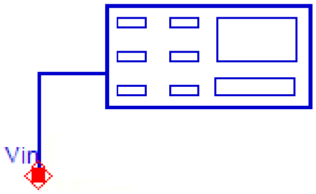
Name	Description	Units	Default
Data Rate	Signal data rate	Gbps	1
Number of time points per UI	Number of time points for eye histogram binning	None	1000
Amplitude resolution	Voltage resolution for eye histogram binning	V	0.001
Lower boundary (%)	Lower boundary for eye level computation	None	40
Upper boundary (%)	Upper boundary for eye level computation	None	60
Lower threshold (%)	Lower amplitude threshold for rise/fall time computation	None	20
Upper threshold (%)	Upper amplitude threshold for rise/fall time computation	None	80
Disable transient analysis output	Disables storage of transient waveforms to save disk space	None	yes

Notes/Equations

1. Provides a differential input for designs requiring use of the Eye_Probe component. For details about parameters and measurements, see *Eye Probe* (ccsim).
2. The component is located on the Simulation-Transient palette.

Eye_Probe (Eye Measurement Probe)

Symbol



Parameters

Name	Description	Units	Default
Data Rate	Signal data rate	Gbps	1
BER width and height	The bit error rate of a probability contour for *WidthAtBER* and *HeightAtBER* measurements	None	1e-12
BER contour	List of bit error rates for computing probability contours	None	list(1e-12, 1e-11, 1e-10)
Extrapolate in bit-by-bit mode	Enables BER extrapolation in bit-by-bit mode	None	no
Use eye mask	Enables the eye mask specification and outputs the eye mask to dataset	None	no
Disable transient analysis output	Disables storage of transient waveforms to save disk space	None	yes
Number of time points	Number of time points per 2 UI for eye histogram binning	None	400
Amplitude resolution	Voltage resolution for eye histogram binning	V	0.001
Lower boundary (%)	Lower boundary for eye level computation	None	40
Upper boundary (%)	Upper boundary for eye level computation	None	60
Lower threshold (%)	Lower amplitude threshold for rise/fall time computation	None	20
Upper threshold (%)	Upper amplitude threshold for rise/fall time computation	None	80
Position from crossing level	Offset from eye crossing level for timing bathtub measurement	V	0.0
Position from crossing time	Offset from eye crossing time for amplitude bathtub measurement	None (UI)	0.5

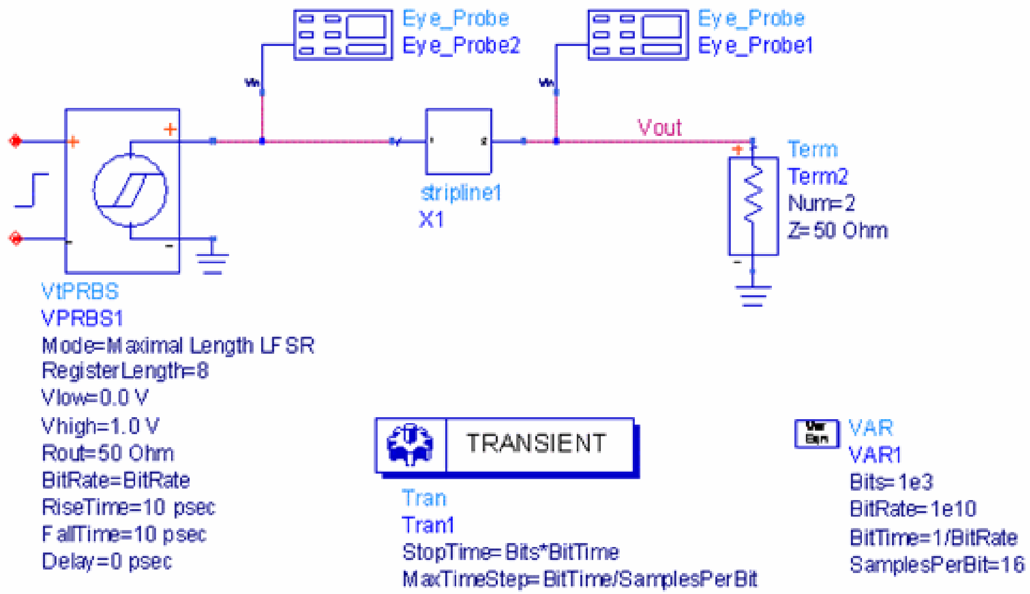
Notes/Equations

1. **Description**
   The Eye_Probe component displays eye diagrams and extracts statistical eye diagram metrics such as probability bathtubs, BER contours, eye width, eye height, RMS jitter and rise/fall time. For many of the measurements, it uses the same algorithms as the *Eye Diagram FrontPanel* (data) utility. Unlike the Eye Diagram FrontPanel, Eye_Probe does not require stored circuit waveforms and does not rely on waveform post-processing in the data display. As a result, Eye_Probe measurements are significantly faster, require little disk space, and consume far less computer memory.

   Eye_Probe components work with the *Transient and Convolution* (cktsimtrans) and *Channel* (cktsimchan) simulation controllers. They produce no output in any other simulation analysis. In all cases, Eye_Probe components are open-circuited, which means they do not have any electrical effect on the circuit.
2. Eye_Probe component can only be attached to the Rx component in a Channel Simulation.

3. **Location**
The Eye_Probe is located on the Simulation-Transient and Simulation-ChannelSim component palettes, and uses the following icon:

4. **Example Circuit**
After placing the Eye_Probe on the schematic, connect the probe to a circuit node as shown in the following figure. Two Eye_Probe components are shown in this example, but any number of probes may be placed in a circuit:



5. **Parameters Setup Dialog**
To set Eye_Probe parameters, double-click on the symbol to open the component setup dialog box. There are two tabs:

- **Parameters**: Sets Eye_Probe parameters common to all measurements.
- **Measurements**: Sets specific measurements for output to the data display.

By default, the **Parameters** tab shows a subset of commonly used parameters. To access all of the available parameters, click *More* to expand the parameters tab.

Setting Up the Parameters Tab
The following table describes how parameter settings interact:

Parameter	Description
Data rate	This parameter should be set to the data rate of the incoming data stream. In transient simulation, *Data rate* is a required parameter. In channel simulation, this parameter is automatically synchronized to the *Bit rate* parameter of the TX source.
BER width and height, BER contour	These parameters control the output of Contour, WidthAtBER and HeightAtBER measurements, as discussed below.
Use Eye Mask	This parameter let you specify an eye mask and view it in the dataset, as described in detail in section Eye Masks below.
Number of time points per UI, Amplitude resolution	These parameters set the horizontal and vertical resolution of the eye histogram. Using finer resolution increases measurement accuracy at the expense of larger disk storage and run-time memory requirements.
Lower boundary (%), Upper boundary (%)	Boundary parameters define the eye diagram boundary for the computation of quantities such as amplitude, height, and SNR. Lower and upper limits are defined as a percentage of a UI.
Lower threshold (%), Upper threshold (%)	Threshold parameters define the amplitude swing for rise and fall time computation.
Disable transient analysis output	Use this parameter to prevent the output of transient waveforms to the data display. By default, simulation controllers save all waveforms in a hierarchical design up to a specified level of hierarchy (see *Selectively Saving and Controlling Simulation Data* (cktsim)). Waveform storage requires a large amount of disk space in long transient simulations. This option saves disk space by preventing default waveform storage from transient simulations. If the waveform at the node to which the probe is connected is required, Eye_Probes provide an optional waveform measurement, to be discussed later.

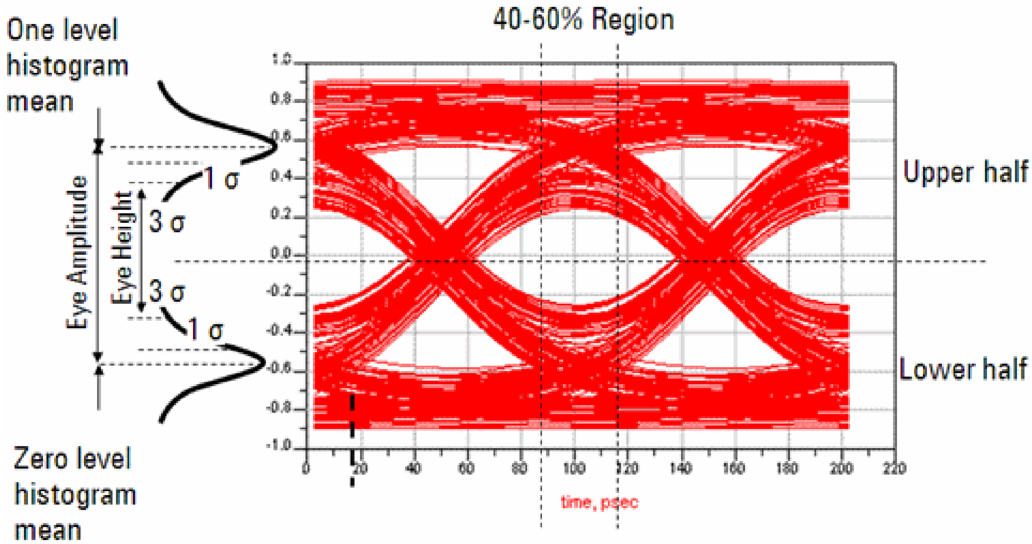
Setting Up the Measurements Tab
The Measurements tab offers a selection of measurements to be processed and sent to the data display. To move a measurement into the list of *Selected* measurements, click a measurement name in the *Available* list, then click **Add**. The default selection includes *Density*, *Height*, *Level1*, *Level0*, and *Width*. The available measurements are described in the following section.

6. **Eye Histogram Measurement Equations**
Eye measurements are computed from the statistical properties of the 3-dimensional eye histogram. The eye histogram is computed by counting the number of signal crossings in the time-voltage plane. Histogram resolution is defined by the resolution parameters on the Eye_Probe's setup dialog box on the Parameters tab.

Eye measurements can be loosely characterized as shown in the following table:

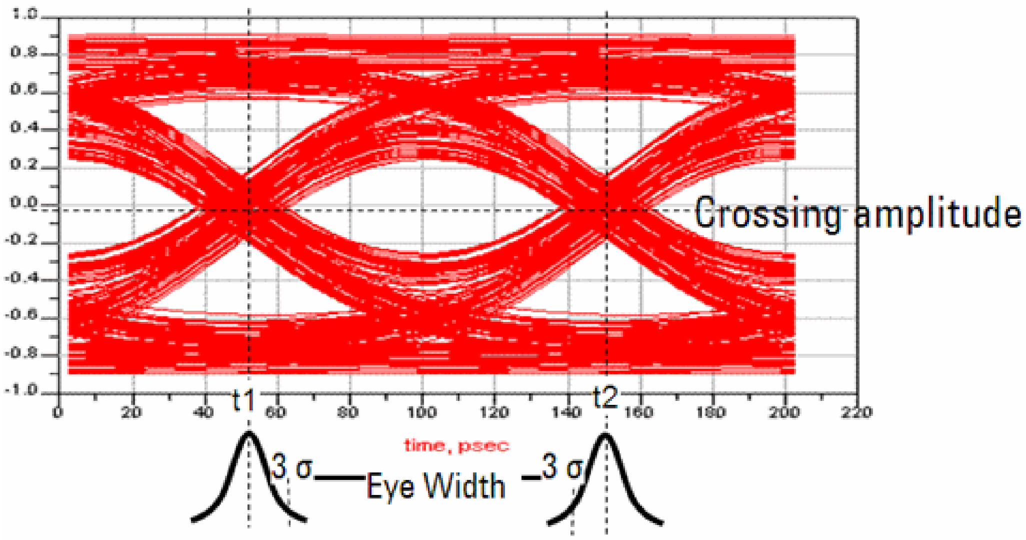
Characterization	Measurements
Vertical measurements	Amplitude, Height, HeightDB, Level1, Level0, LevelMean, SNR
Horizontal measurements	FallTime, JitterPP, JitterRMS, RiseTime, Width
BER measurements	Contour, Bathtub, WidthAtBER, HeightAtBER
Miscellaneous measurements	Density, Waveform, CheckMaskViolation

Vertical measurements are computed from the properties of the eye histogram in the region defined by the *Eye Level Boundaries* parameters, as shown in following figure:



Vertical Measurement	Description
Amplitude	Computed as the difference between *Level1* and *Level0*.
Height	The distance between the 3-sigma points of the logic-1 and logic-0 histograms, measured across the eye level boundary. Height is calculated according to the formula: $Height=(Level1-3*sigma_{level1}) - (Level0+3*sigma_{level0})$
HeightDB	Calculated as 10log10(Height).
Level1	The mean of the eye vertical histogram in the upper half of the eye level boundary.
Level0	The mean of the eye vertical histogram in the lower half of the eye level boundary.
LevelMean	Defined as the average of *Level1* and *Level0*.
SNR	SNR (signal-to-noise ratio) is calculated as $SNR=Amplitude/(sigma_{level1} + sigma_{level0})$.

Horizontal measurements are computed from the statistics of the crossing histogram at certain amplitude thresholds as shown in the following figure:



Horizontal Measurement	**Description**
JitterPP	The peak-to-peak value, or width, of the crossing time histogram.
JitterRMS	The standard deviation of the crossing time histogram.
RiseTime and FallTime	These measurements calculate the time between the mean crossings of the upper and lower amplitude thresholds. The upper and lower thresholds default to 80% and 20%, respectively, and are controlled by the *Amplitude Threshold* parameters located on the Eye_Probe's Parameters tab. The thresholds are a percentage of the eye *Amplitude*, defined above as Amplitude = Level1 - Level0.
Width	Width is calculated according to the formula: Width=(t_2 - 3*sigma$_{t2}$) - (t_1 + 3*sigma$_{t1}$). t_2 and t_1 are the mean values of the crossing time histograms.

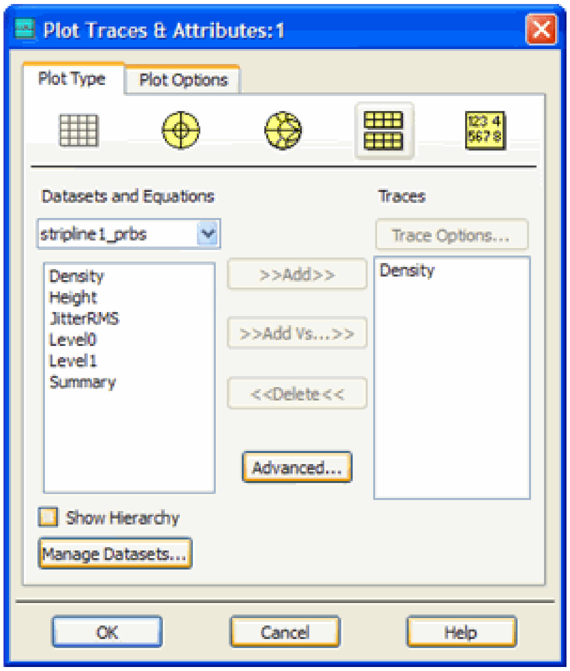
BER measurements are described in the following table:

BER Measurement	**Description**
Contour	This measurement computes contours of constant error probability and outputs them to dataset as *BERContour*. Any number of contours may be specified using the eye probe *BER contour* parameter. The contours are computed over 1 UI. If a contour doesn't exist because its BER is below the system's minimum BER, a warning is printed in the status window and the contour is not output to the dataset.
Bathtub	This measurement computes the timing and amplitude bathtubs and outputs them to dataset as *TimingBathtub* and *VoltageBathtub*, respectively. The voltage reference for the timing bathtub is automatically set at the eye crossing level. Similarly, the timing reference for the voltage bathtub is automatically set at 0.5 UI offset from the eye crossing. The timing and amplitude references may be changed using the eye probe parameters *Position from crossing level* and *Position from crossing time*.
WidthAtBER and HeightAtBER	Computes the maximum width and height of a contour at a BER specified using the *BER width and height* parameter

Miscellaneous measurements are described in the following table:

Miscellaneous Measurement	**Description**
Density	A temperature view of the 3-dimensional eye histogram. The eye is automatically centered and displayed over 2UI on the horizontal axis. Color indicates the number of crossings of a segment in the time-voltage plane. Blue areas are *cold*, whereas red indicates comparatively more crossings.
Waveform	Stores the waveform at the node to which the probe is connected. By default, Eye_Probe components suppress standard waveform output from the transient simulation controller. This option gives access to the waveform at the probe node, while suppressing default transient output and therefore preserving disk space.
CheckMaskViolation	Compares the *Density* measurement to the specified eye mask and reports results as *MaskViolated* in the data display. If the mask is violated by the eye density plot, *MaskViolated* is set to 1.

7. **Output**
The Eye_Probe stores the chosen measurements in a dataset for viewing in the data display as shown in the following figure:



In addition to the chosen measurements, a Summary table is available to display all of the measurements in a convenient format, as shown in the following figure. In case of multiple probes, the measurement is prefixed by the Eye_Probe's instance name.

measurement	Summary
Level1	0.488
Level0	0.010
Height	0.454
JitterRMS	7.014E-13

8. Measurement availability in transient and channel simulations

The eye probe component works with transient and channel simulations (both bit-by-bit and statistical). Some measurements are not available in all three analysis modes. This table summarizes the availability of eye probe measurements as a function of analysis type:

Measurement	Transient	Bit-by-bit Channel Simulation	Statistical Channel Simulation
LevelMean	X	X	X
HeightAtBER		X	X
JitterRMS	X	X	X
RiseTime	X	X	X
CheckMaskViolation	X	X	X
SNR	X	X	X
Amplitude	X	X	X
HeightDB	X	X	X
FallTime	X	X	X
WidthAtBER		X	X
JitterPP	X	X	X
Bathtub		X	X
Waveform	X	X	
Contour		X	X
Level1	X	X	X
Density	X	X	X
Height	X	X	X
Level0	X	X	X
Width	X	X	X

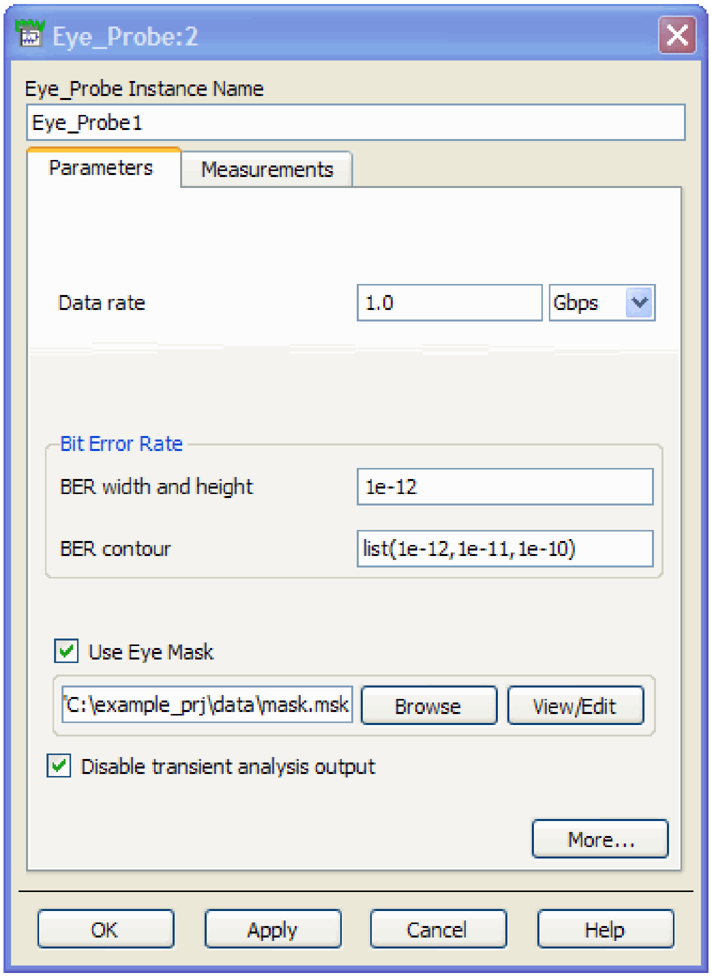
9. **Eye Masks**
Eye_Probe has features that help define, edit and display eye masks. Optionally, the eye probe automatically checks and reports eye density violations of the specified mask. The mask may be viewed by selecting the *Mask* measurement in the data display.

Eye mask definition and display
Eye masks are specified in text files compatible with Agilent oscilloscope mask definitions, as shown below:

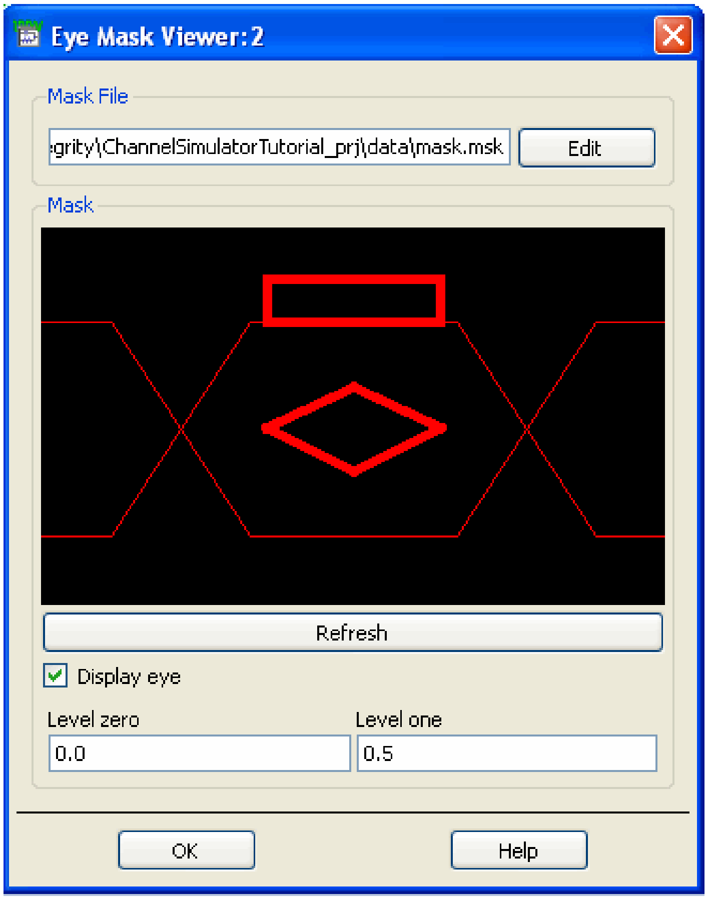
```
1 /* This is the first polygon index */
4 /* This is the number of vertices */
0.25,0.25
0.5,0.35
0.75,0.25
0.5,0.15

2 /* This is the second polygon index */
4 /* This is the number of vertices */
0.25, 0.5
0.25, 0.6
0.75,0.6
0.75,0.5
```

The mask file consists of sections specifying mask polygons. Any number of polygons may be specified. The first entry in the polygon definition section is the polygon index. The second entry specifies the number of polygon vertices. The vertices are specified as comma-delimited time-voltage coordinates of the mask. The voltage coordinate specifies absolute amplitude in Volts; the time coordinate is relative to the eye crossing, a value of 0.0 corresponds to the first crossing point of the eye and a value of 1.0 corresponds to the second crossing point. C-language-style comments are permitted, as shown in the example.

To view a mask, copy-and-paste the example mask shown above and save it to a file. In this example, the file is named mask.msk. Save the file to the data directory of your ADS workspace. On the eye probe Parameters tab, enable *Use Eye Mask* and click Browse to select mask.msk, as shown below:



If you wish to edit or view the mask, click **View/Edit** to bring up the mask editing dialog box, as shown below:

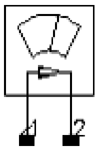


The View/Edit dialog box lets you preview the mask. By clicking **Edit**, you can make changes to the mask file and view the updates by pressing **Refresh**.

1. **Warning Messages**
   When *Disable transient analysis output* is enabled on the Parameters tab, the Eye_Probe sends a warning message to the simulation status window as a reminder that the default output from the transient simulation is suppressed.
2. **Error Conditions**
   Eye_Probe uses automatic algorithms to detect eye crossing thresholds and other eye parameters. If the eye is closed or highly distorted, these automated algorithms may fail, resulting in an all-zero output to the data display.

I_Probe (Current Probe)

Symbol



Parameters

Name	Description	Units	Default
Mode	Mode: 0 => short; (not used)	None	(0)
C	DC block capacitance (transient only)		None
L	DC feed inductance (transient only)		None
Gain	Current gain	None	None
SaveCurrent	Save branch current: yes or no	None	yes
wImax	Maximum current (warning)		None

Notes/Equations

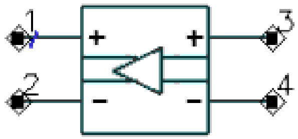
1. The positive current flow direction is assumed to be from pin 1 to pin 2.
2. To measure a branch current, an ammeter must be connected in that branch before performing the analysis.
3. The following table lists the DC operating point parameters that can be sent to the dataset.

Name	Description	Units
I	Current	A

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

OscPort2 (Differential Oscillator Port)

Symbol



Parameters

Name	Description	Units	Default
Mode	Oscillator mode: automatic, small signal loop gain, or large signal loop gain	None	Automatic
V	Initial guess at fundamental voltage (automatic mode only)		None
Z	Initial value for Z_0, in ohms (all modes)	Ohm	1.1
NumOctaves	Number of octaves to search (automatic mode only)	None	2
Steps	Number of steps per search octave (automatic mode only)	None	10
FundIndex	Fundamental number for oscillator (automatic mode only)	None	1
Harm	Harmonic or fundamental for oscillator	None	1
MaxLoopGainStep	Maximum arc length continuation step size during loop-gain search (automatic mode only)	None	None
FreqPlan	Sweep plan for frequency (small and large signal loop gain modes only)	None	None
VinjPlan	Sweep plan for injected loop voltage (large signal loop gain mode only)	None	None

Notes/Equations

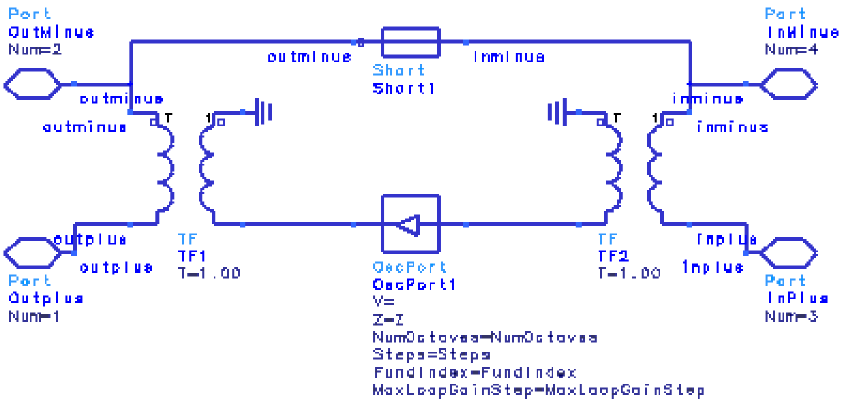
1. This is a special device used for an oscillator analysis. Do not use more than one oscillator test element (OscTest, OscPort, OscPort2) in a circuit.
2. NumOctaves specifies the total number of octaves over which the oscillator search is done. Half of the octaves are below the initial frequency and 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.

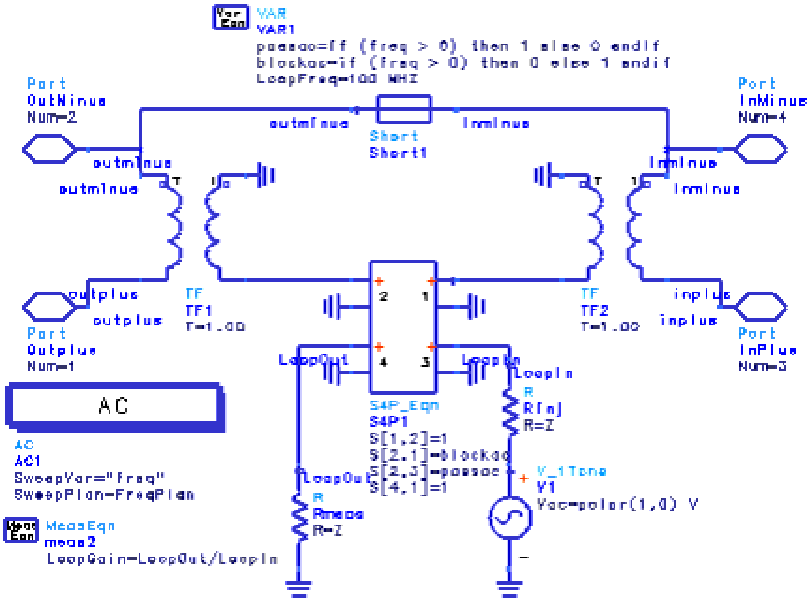
5. This device can be used for both single-ended and differential oscillator topologies. For single-ended oscillators, the negative pins of this element should be grounded. For differential oscillators, it should be connected differentially into the oscillator loop.
6. Provided the circuit produces at 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 (Z0) to a value approximately half-way between the source and load impedances presented by the circuit at the point of insertion.
     To minimize the analysis time, set the probe impedance to a factor of 10 below the load impedance, and a factor of 10 above the source impedance (provided the source and load impedances are sufficiently far apart). Doing this effectively reduces to zero the dependence of the small signal loop gain on Z_0.

     Negative Resistance Oscillators
   - Insert probe between a negative and positive impedance in the circuit. There should be no other signal paths between these two parts of the circuit. Typically, the probe is inserted between the resonator and the effective negative resistance.
   - You can point the arrow of the probe at either the negative impedance node or the positive impedance node.
   - Set the initial probe impedance to any reasonable value. To minimize the analysis time, it should be at least a factor of two higher or lower than the magnitude of the passive load impedance.
     The frequency is specified on the harmonic balance analysis component. The value for Z is chosen based on impedance levels in the circuit and the degree of non-linearity in the circuit. Do not use either 1 or 0 for Z as this will cause convergence problems.
     If the oscillator analysis fails, and this test indicates that the circuit should oscillate, the failure may be due to the fact that the circuit is too nonlinear. This problem can sometimes be solved by trying different impedance values of OscPort (determined by the Z attribute). Lower impedance values usually seem to work better. Also try reversing the OscPort direction.
     Another approach is to try to get the oscillator to oscillate at some nicer parameter value and then to sweep the parameter value to the desired value. The parameter may be bias, self-bias impedance, some gain controlling value, or another factor. In short, anything that will make the oscillator more linear, yet still let it oscillate.
     FundIndex is used for selecting which fundamental tone is considered the unknown frequency during oscillator analysis.
     The FundIndex default is 1, which means that Freq[1] on the Harmonic Balance

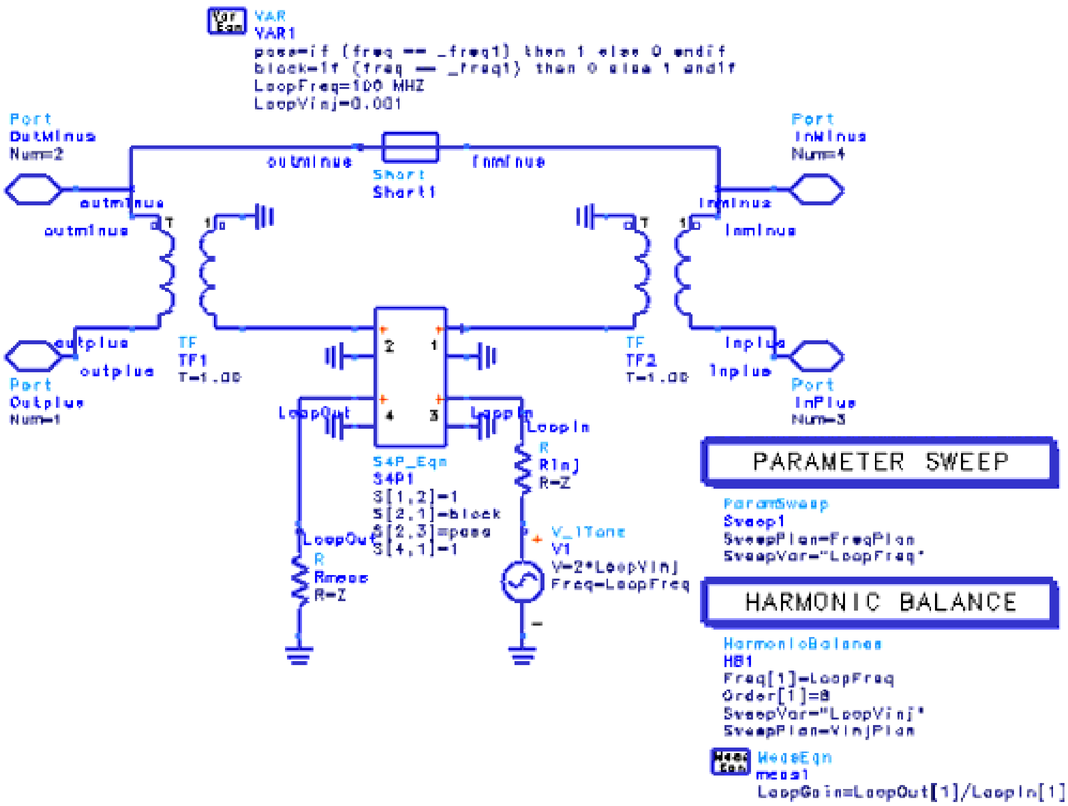
controller is the unknown frequency. This should be changed only if a larger multi-tone system is simulated, such as an oscillator and mixer. In this case, the user may want Freq[1] to be a known driven source and Freq[2] to be the unknown frequency used by the oscillator; for this, set FundIndex=2. For the best harmonic balance solution, the frequency that causes the most nonlinearity should be Freq[1].
Harm is used to make a circuit oscillator on a harmonic of the fundamental frequency rather than directly on the fundamental. For example, the circuit may consist of a 2GHz oscillator followed by a divide-by-two circuit. In this case, the harmonic balance analysis would be set up with Freq[1]=1 GHz, and OscPort2 would have Harm=2. (Note that successful simulation of an oscillator and divider will most likely require that transient-assisted harmonic balance be used.)
The equivalent circuit for OscPort2 in automatic mode is shown.



7. The small-signal loop gain mode is used to examine the small signal linear behavior of the oscillator feedback loop. In this mode, the OscPort2 element behaves as an analysis controller. Any simulation controllers should be disabled before using the OscPort2 in this mode. The analysis calculates and places in the dataset a complex value called LoopGain which is the small signal loopgain of the oscillator.
The range of frequencies over which to analyze loop gain should be specified with a SweepPlan item. The name of this SweepPlan should then be assigned to the parameter FreqPlan.
This device is used to evaluate the ability of a closed-loop system to produce one or more complex conjugate pole pairs in the right-half-plane (RHP) of a pole/zero diagram. This device measures the open-loop gain and phase of the closed-loop system. These results must be plotted on a polar graph (Nyquist diagram) to properly interpret them.
The number of clockwise encirclements of the l + j0 point indicates the number of RHP poles that were produced due to the feedback. The total number of RHP poles is the sum of the number of clockwise encirclements plus the number of RHP poles present in the individual networks that comprise the closed-loop system.
An important aspect of this last point is that traditional feedback or negative resistance topology systems may be unstable even though the l + j0 point is not encircled in a Nyquist diagram. For example, in a negative resistance topology circuit, if the reference impedance of the OscTest device is set equal to the passive load impedance, the measured loop gain is zero. The circuit will oscillate, however, because the negative resistance one-port generates an RHP pole prior to being configured with the remaining part of the system.
The equivalent circuit for OscPort2 in small signal loop gain mode is shown.



8. The large-signal loop gain mode is used to examine the behavior of the oscillator feedback loop as a function of frequency and injected voltage. It can be used to observe the compression of loop gain as the loop voltage is increased. In this mode, the OscPort2 element behaves as an analysis controller. Any simulation controllers should be disabled before using the OscPort2 in this mode. The analysis calculates and places in the dataset a complex value called LoopGain which is the large signal loop gain of the oscillator. The circuit will sustain oscillation at the point at which the magnitude of LoopGain equals one and the phase of LoopGain equals zero.
   The range of frequencies over which to analyze loop gain should be specified with a SweepPlan item. The name of this SweepPlan should then be assigned to the parameter FreqPlan.
   The range of voltages over which to analyze loop gain should be specified with a SweepPlan item. The name of this SweepPlan should then be assigned to the parameter VinjPlan. Initially the sweep should be done with a logarithmic sweep to determine where the oscillator loop goes into compression. Once this range is estimated, a linear sweep can be done to zero in north injected voltage that causes oscillation to be sustained.
   A useful way to interpret results from this analysis is to plot the phase of LoopGain against LoopGain in decibels. Lines of constant frequency will be plotted with values at each voltage value. The circuit will oscillate at the frequency and voltage associated with the (0,0) point on the graph.
   The equivalent circuit for OscPort2 in large signal loop gain mode is shown.



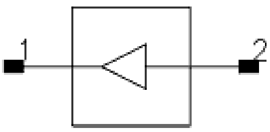
In both small signal and large signal loop gain modes, this element injects a test signal into an oscillator circuit for stimulating oscillations. The specialized directional coupler has zero electrical length and is invisible to normal circuit simulation. It injects a fundamental frequency test signal, blocks the fundamental frequency flow in the feedback path, monitors the signal returned by the feedback path and calculates the loop gain.
The directional coupler in loop gain mode (both small and large signal) is designed to allow the injection of a test signal from port 3 to port 2 as the loop input and to pass the loop output from port 1 to port 4. It does this only at the signal frequency: the AC frequency for small signal loop gain and the fundamental tone for large signal loop gain. All other frequencies, including DC, are coupled from port 1 to port 2.
The scattering matrices follow.

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

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

OscPort (Grounded Oscillator Port)

Symbol



Parameters

Name	Description	Units	Default
V	Initial guess at fundamental voltage		None
Z	Initial value for Z_0	Ohm	1.1
NumOctaves	Number of octaves to search	None	2
Steps	Number of steps per search octave	None	10
FundIndex	Fundamental number for oscillator	None	1
Harm	Harmonic or fundamental for oscillator	None	1
MaxLoopGainStep	Maximum arc length continuation step size during loop-gain search	None	None

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.

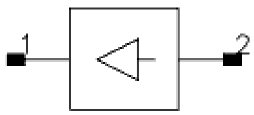
FundIndex is used for selecting which fundamental tone is considered the unknown frequency during oscillator analysis.

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.)

OscTest (Grounded Oscillator Test)

Symbol



Parameters

Name	Description	Units	Default
Port_Number	Port number	None	1
Z	Port impedance	Ohm	1.1
Start	Start frequency	GHz	1.0
Stop	Stop frequency	GHz	10.0
Points	Number of frequency points	None	101

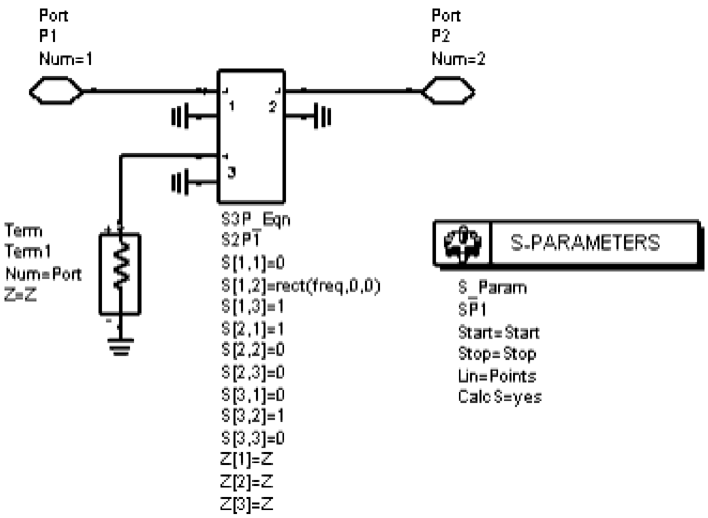
Notes/Equations

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.

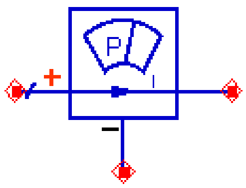
If the oscillator analysis fails, and this test indicates that the circuit should oscillate, the failure may be because the circuit is too nonlinear. This problem can be solved by trying different impedance values for OscTest (determined by the Z attribute). Lower impedance values usually work better,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



P_Probe2 (Differential Power Probe)

Symbol



Parameters

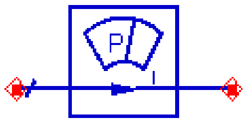
N/A

Notes/Equations

P_Probe2 measures dissipated power in DC, Transient, AC, HB, and Circuit Envelope simulations. The probe samples current Ip through the terminal designated by the arrow sign and voltage Vp between the + and - terminals. Please refer to *P Probe (Grounded Power Probe)* (ccsim) for details about power computation.

P_Probe (Grounded Power Probe)

Symbol



Parameters

N/A

Notes/Equations

P_Probe measures dissipated power in DC, Transient, AC, HB, and Circuit Envelope. The probe samples current Ip through the pair of terminals designated by the arrow sign, and voltage Vp from either of the terminals to ground.

- For DC, AC, and HB simulations, the probe computes average power at each analysis frequency f as:

  If (f>0)

  $P = ½*Vp*conj(Ip)$ (1)

  If (f=0)

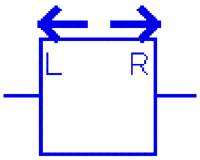
  $P(0) = Vp*Ip$ (2)
- For transient simulations, the probe computes instantaneous power given by

  $p(t) = v(t)*i(t)$ (3)
- For Circuit Envelope, the probe computes instantaneous envelope power at each harmonic, as given by eq. (1)-(2).
- The probe outputs 0 in SP simulations but it doesn't affect simulation results.
- The following table lists the measurement result that can be sent to the dataset

Name	Description	Units
p	Power	W

SP_Probe(S-Parameter Probe)

Symbol



Parameters

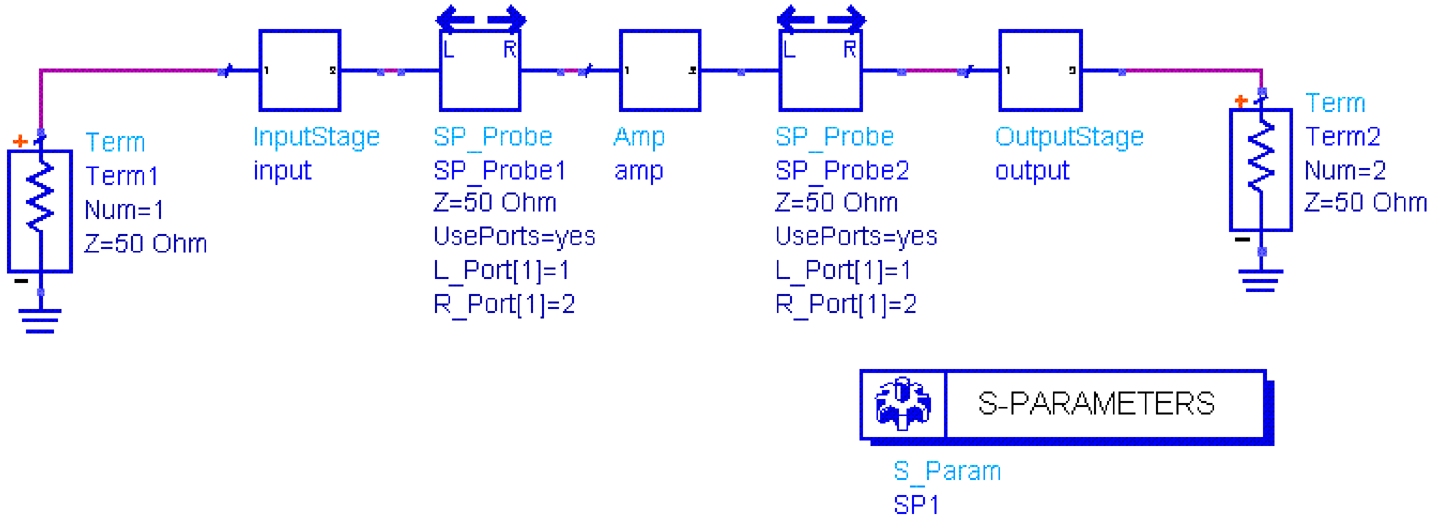
Name	Description	Units	Default
Z	Termination impedance	Ohms	50.0
UsePorts	Enables or disables the use of the ports specified using the L_Port and R_Port parameters. When this parameter is set to a value of 'no', the simulator will only calculate the S11 looking to the left and to the right of the probe. In other words all the other ports will be ignored. When this parameter is set to a value of 'yes' the simulator will calculate the S-Parameters taking into account the ports specified in the parameters L_Port and R_Port.	None	no
L_Port	Left Looking Port (repeatable)	None	1
R_Port	Right Looking Port (repeatable)	None	2

Notes/Equations

1. This component probes for the network parameters of the circuit when an S-Parameter analysis is performed.
2. An example showing how to use the SP_Probe in S-parameter simulations is located in the *SP_Probe_how_to_wrk*. To access the example from the ADS Main window, select *File > Open > Example > Tutorial > SP_Probe_how_to_wrk*, then open the design *A_readme*. For information about this example, see *Using SP_Probe in ADS* (examples) in the *Examples Documentation*.
3. When the initial DC operating point is performed, this component behaves like a short.
4. Similarly when any other analysis besides the small signal S-Parameters is being performed, this component behaves just like a short.
5. The network parameters that are computed are the S, Y and Z parameters.
6. When the small signal noise analysis is performed, this component computes the Sopt, Rn, NF, NFmin, Te and Icor for the network.
7. There will be 2 sets of network parameters, the first are the parameters looking to the left of the component and the second set are the parameters looking right.
8. The port terminations used when computing the left looking parameters are specified using the parameter L_Port, and the termination numbers specified in the R_Port parameter will be used when computing the right looking parameters.
9. The parameters L_Port and R_Port will only be used when the parameter UsePorts is set to *yes*.
10. To clarify how the network parameters are saved, it is best to use the following

simple, 3 stage design.

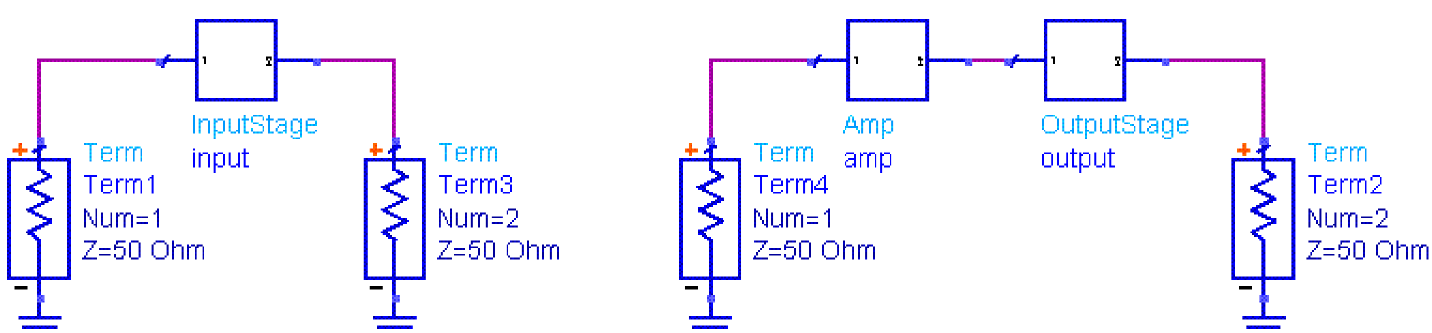
11. When a probe is computing the network parameters all the other probes behave like a short.



These five sets of network parameters will be computed:

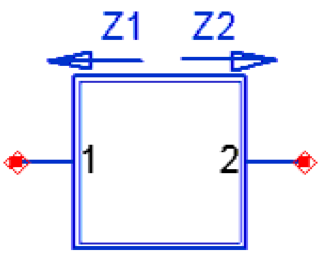
- SP1.S: This is the S-Parameters for the complete network, between Term 1 and Term 2. Note that SP_Probe1 and SP_Probe2 act like shorts.
- SP1.SP_Probe1.L.S: This is the S-Parameters between Term 1 and the port labeled L of SP_Probe1. Note that SP_Probe1 is open and SP_Probe2 is short.
- SP1.SP_Probe1.R.S: This is the S-Parameters between the port labeled R of SP_Probe1 and Term 2. Note that SP_Probe1 is open and SP_Probe2 is short.
- SP1.SP_Probe2.L.S: This is the S-Parameters between Term 1 and the port labeled L of SP_Probe2. Note that SP_Probe1 is short and SP_Probe2 is open.
- SP1.SP_Probe2.R.S: This is the S-Parameters between the port labeled R of SP_Probe1 and Term 2. Note that SP_Probe1 is short and SP_Probe2 is open.

Keep in mind that when a probe is computing the network parameters all the other probes behave like a short. For instance when the probe SP_Probe1 is computing the network parameters the following is the equivalent circuits. The equivalent circuit on the left corresponds to the left looking port of SP_Probe1 and the equivalent circuit on the right corresponds to the right looking port of the SP_Probe1 device.



SProbe2 (SProbe2 Component)

Symbol

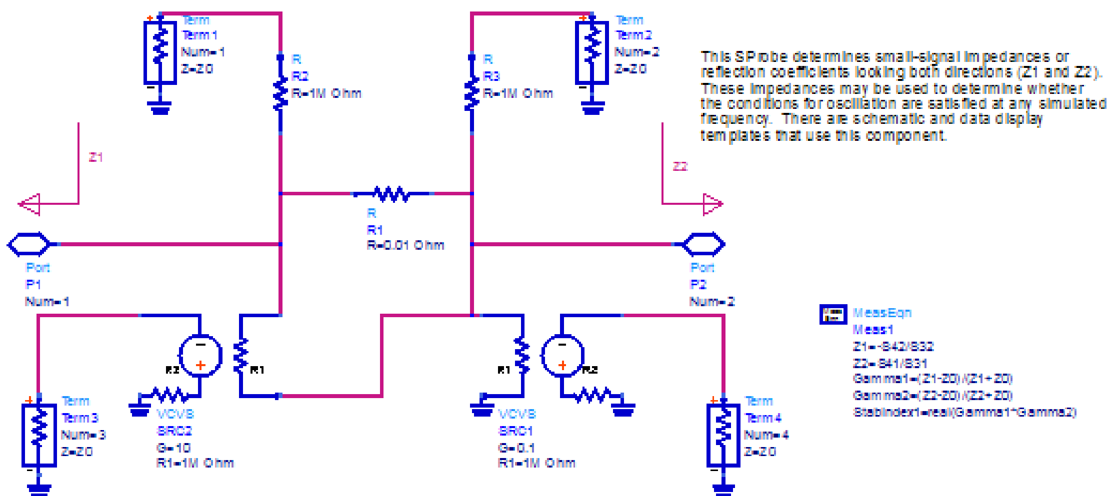


Parameters

Name	Description	Units	Default
Z0	Port impedance	None	50

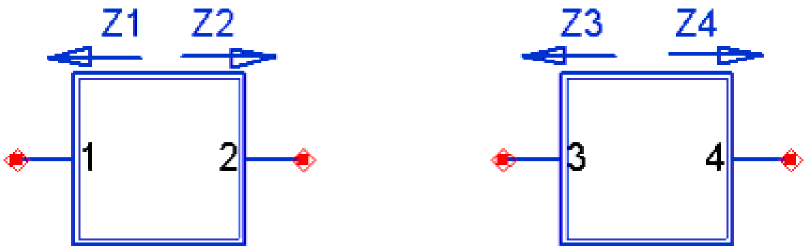
Notes/Equations

1. The SProbe2 component is used to determine small-signal impedances or reflection coefficients looking both directions (Z1 and Z2). These impedances may be used to determine whether the conditions for oscillation are satisfied at any simulated frequency.
2. This component can be inserted anywhere into a circuit without loading it.
3. SProbe2 is intended to be used with the template SProbeT, which includes the SProbe2 component (in a Schematic, click **Insert** > **Template** > **SProbeT**).
4. Do not combine the SProbe2 with other simulations that require Term components, including sources that have built-in Terms (e. g., P_1Tone).
5. The equivalent circuit of SProbe2 is shown. When compared to the obsolete SProbe component, the SProbe2 circuit does not include the S-parameters and Sweep Plan controllers. This is the only difference between them. It is intended that the S-parameters and Sweep Plan controllers be placed at the same level as the SProbe2 component.



SProbePair2 (SProbePair2 Component)

Symbol

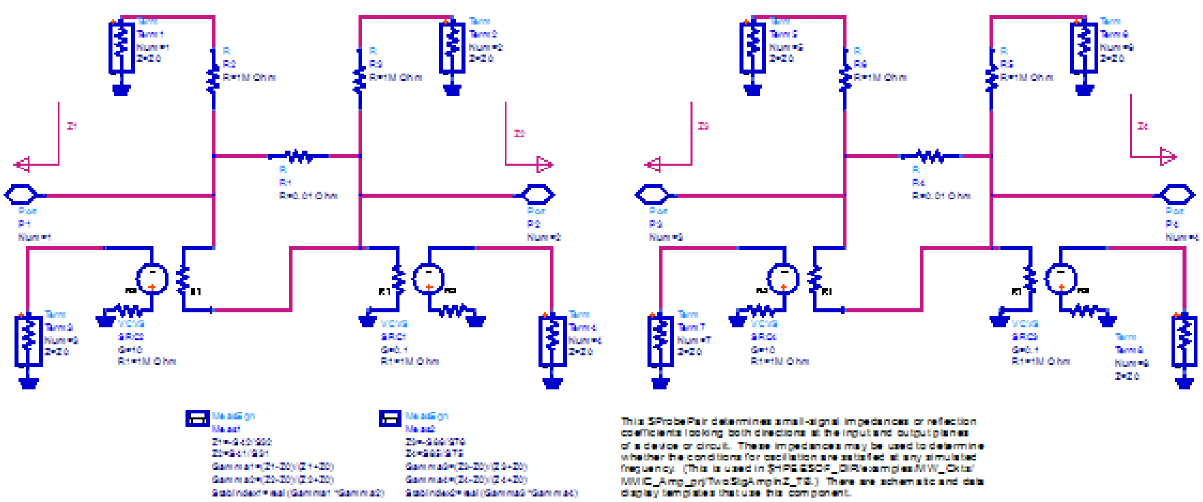


Parameters

Name	Description	Units	Default
Z0	Port impedance	None	50

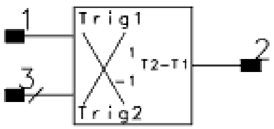
Notes/Equations

1. The SProbePair2 component is used to determine small-signal impedances or reflection coefficients looking both directions at the input and output planes of a device or circuit. These impedances may be used to determine whether the conditions for oscillation are satisfied at any simulated frequency.
2. This component can be inserted anywhere into a circuit without loading it.
3. The SProbePair2 component is intended to be used with the template SProbePairT, which includes the SProbePair 2 component (in a Schematic, click **Insert** > **Template** > **SProbePairT**).
4. Do not combine the SProbePair2 with other simulations that require Term components, including sources that have built-in Terms (e.g., P_1Tone).
5. SProbePair2 is used in the following example: $HPEESOF_DIR/examples/MW_Ckts/MMIC_Amp_wrk/TwoStgAmpInZ_TB
6. The equivalent circuit of SProbePair2 is shown below. When compared to the obsolete SProbePair component, the SProbePair2 circuit does not include the S-parameters and Sweep Plan controllers. This is the only difference. It is intended that the S-parameters and Sweep Plan controllers be placed at the same level as the SProbePair2 component.



TimeDelta (Time Delta Component)

Symbol



Parameters

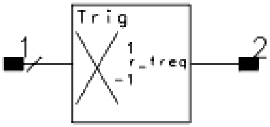
Name	Description	Units	Default
Direction1	Direction one	None	1
Direction2	Direction two	None	1
Thresh1	Threshold one	V	0
Thresh2	Threshold two	V	0
Scale	Scale factor	None	1e+6

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

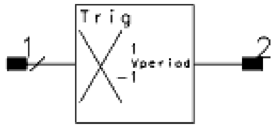
Name	Description	Units	Default
Direction	Trigger Direction	None	1
Thresh	Trigger Threshold	V	0
Scale	Scale factor	None	1e-6

Notes/Equations

1. TimeFrq generates an output voltage proportional to the inverse of the time between two consecutive triggers, which is basically the frequency of the triggering event. 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 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 inverse of the time difference between the last two trigger events (i.e. the triggering frequency) 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 too large and affect the simulation accuracy.

TimePeriod (Time Period Component)

Symbol



Parameters

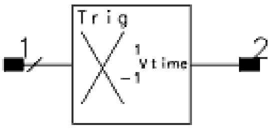
Name	Description	Units	Default
Direction	direction one	None	1
Thresh	threshold one	V	0
Scale	scale factor	None	1e+6

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

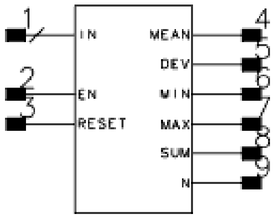
Name	Description	Units	Default
Direction	Direction one	None	1
Thresh	Threshold one	V	0
Scale	Scale factor	None	1e+6

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

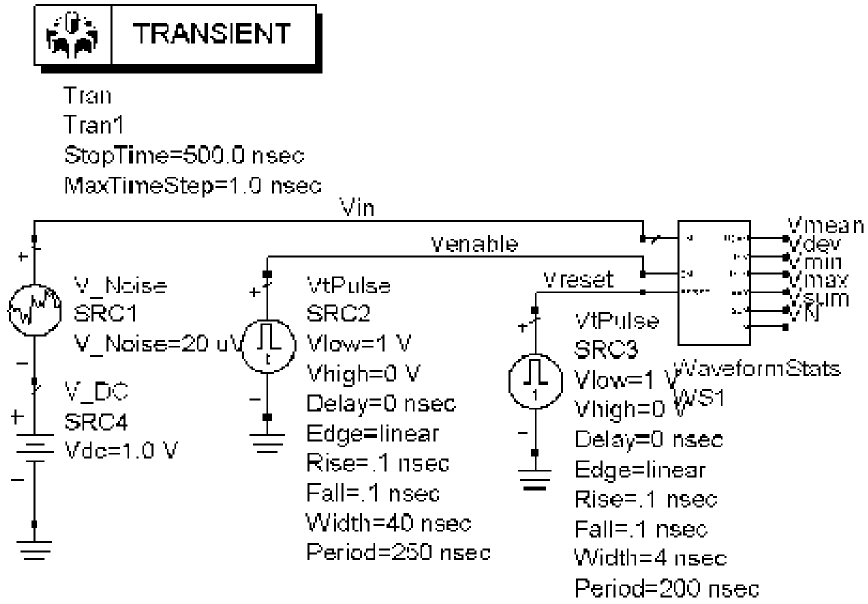


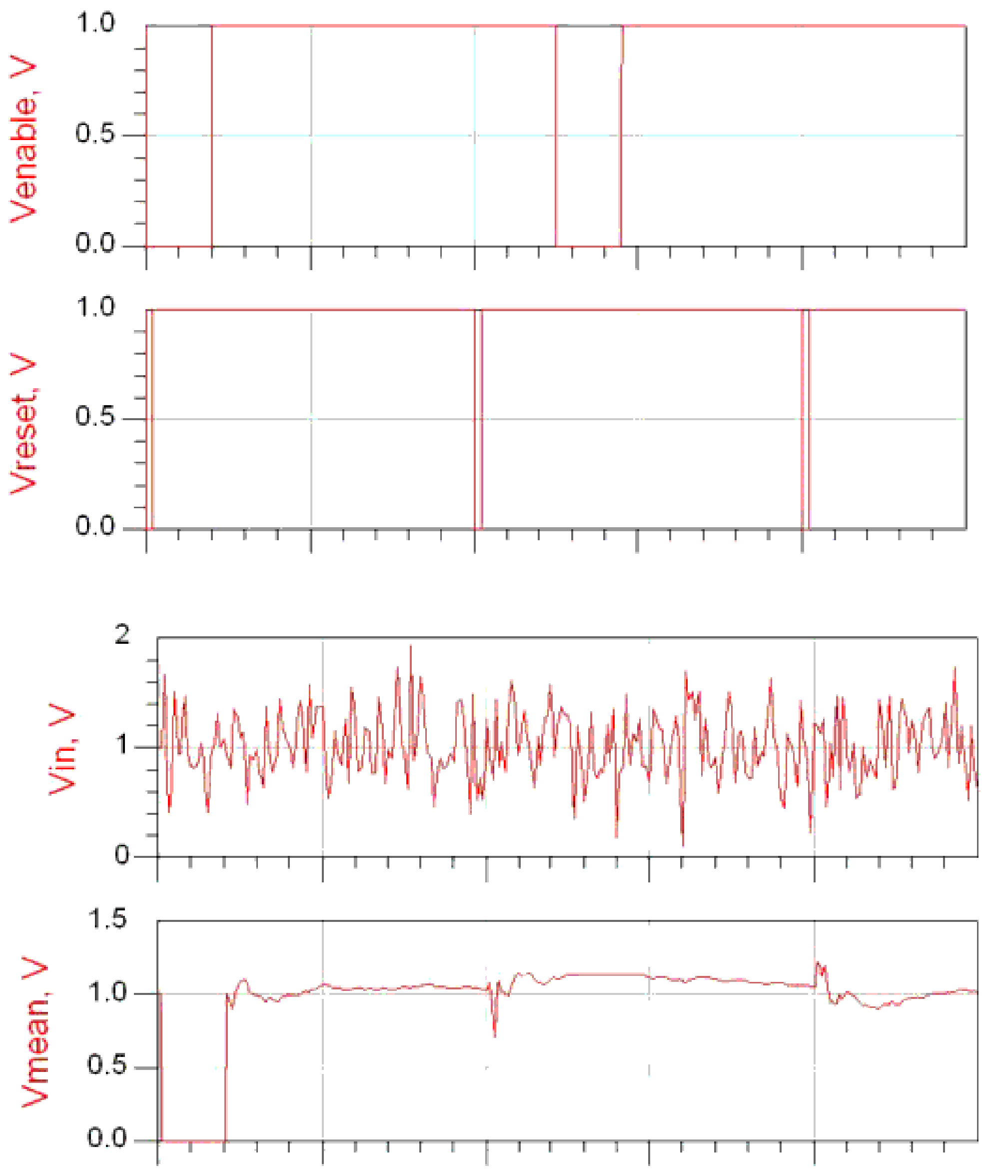
Parameters

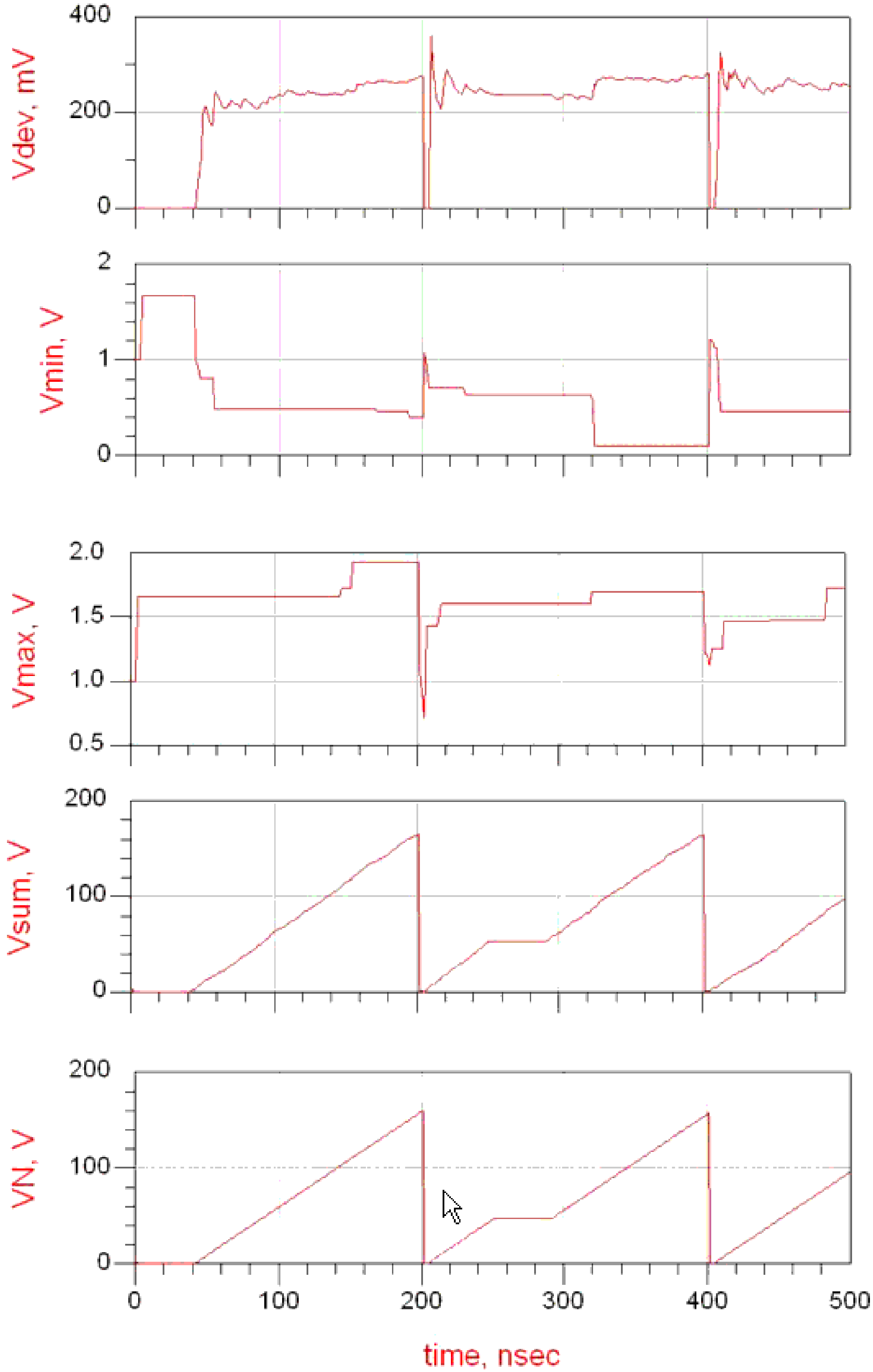
none

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
Vreset, V
Vin, V
Vmean, V
1.0
0.5
0.0
2
1
0
1.5


400
200
0
Vdev, mV
2
1
0
Vmin, V
2.0
1.5
1.0
0.5
Vmax, V
200
100
0
Vsum, V
200
100
0
VN, V
0
100
200
300
400
500
time, nsec