Analog Library Reference

Product Version IC23.1
August 2023

Contents

1

Introduction to Analog Library

This topic contains information about all the components in the Analog Library (analogLib). The analogLib library is a library within the Virtuoso Analog Design Environment. You can access the library from the following path:

<your_install_dir>`/tools/dfII/etc/cdslib/artist/analogLib`

Make sure you specify this path in the search path of the Set Library Search Path form.

The Analog Library (`analogLib`) is a library within the Virtuoso Analog Design Environment. The `analogLib` library contains basic analog components, such as resistors, capacitors and transistors that are used in building complex analog blocks, such as amplifiers.

This manual contains information about all the components in the Analog Library. The information presented in this manual is intended for integrated circuit designers and assumes that you are familiar with analog design and the following:

- The applications used to design and develop integrated circuits in the Virtuoso Studio design environment, notably Virtuoso Schematic Editor and Virtuoso Analog Design Environment.
- Component description format (CDF), which lets you create and describe your own components for use with Virtuoso Schematic Editor and Virtuoso Analog Design Environment.

The analogLib library contains basic components, such as resistor, capacitance, and transistor. These basic analog parts are used in building complex analog blocks, such as amplifiers.

The components in analogLib are divided into 10 categories, such as Actives, Analysis, Parasitics and so on. For each component in analogLib multiple views, such as the symbol view and simulator specific views are available. For some components, the schematic view might also be available.

Each component may be supported by different simulators, such as spectre or auCdl. The simulators supported in the Cadence Analog Design Environment are:

- spectre
- ams
- auCdl
- auLvs
- hspiceD
- UltraSim

Licensing Requirements

For information on licensing in the Virtuoso Studio design environment, see *Virtuoso Software Licensing and Configuration Guide*.

Parameters in the Analog Library

This topic lists all the basic parameters that you specify at the time of adding a component to a design. The *Add Instance* form may not show all the parameters at once. Depending on what values you specify for some parameters, more fields may appear in the Add Instance form. You can display the complete list of parameters for each component using the Edit CDF form.

Following are the steps to display the parameters for a component using the Add Instance form as well as the Edit CDF form.

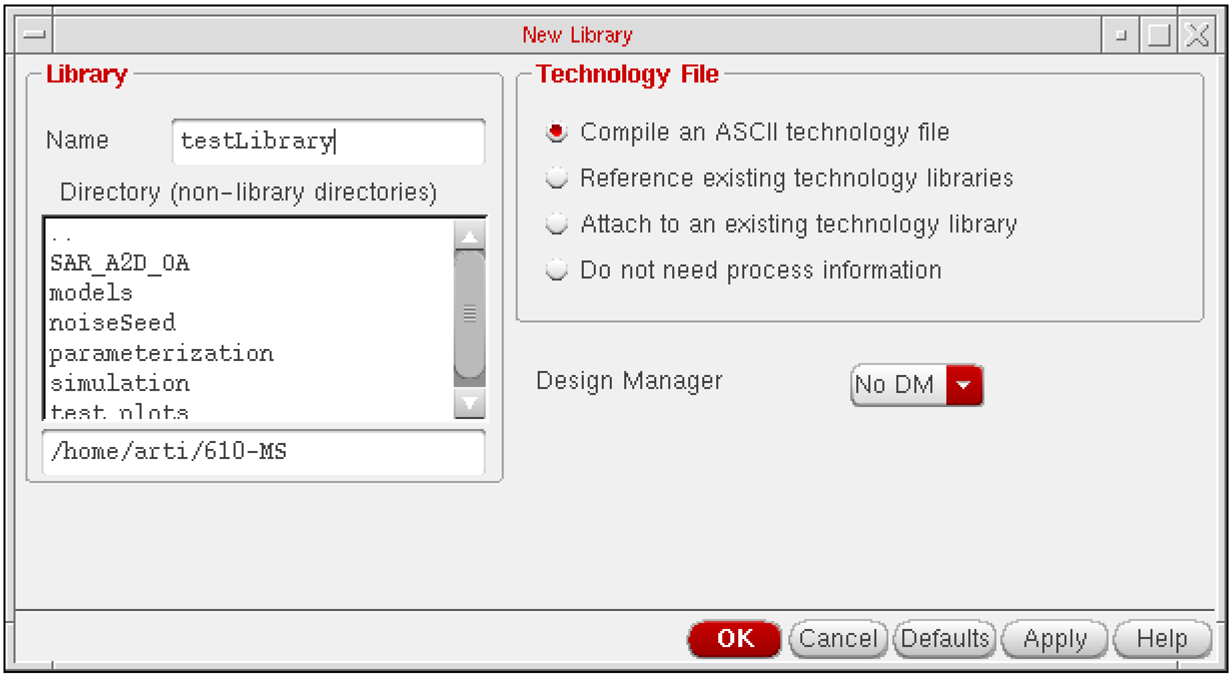
For these series of steps, you will create a library and cell.

1. Type `icms&` in the xterm window.

   The CIW (Cadence Information Window) appears.

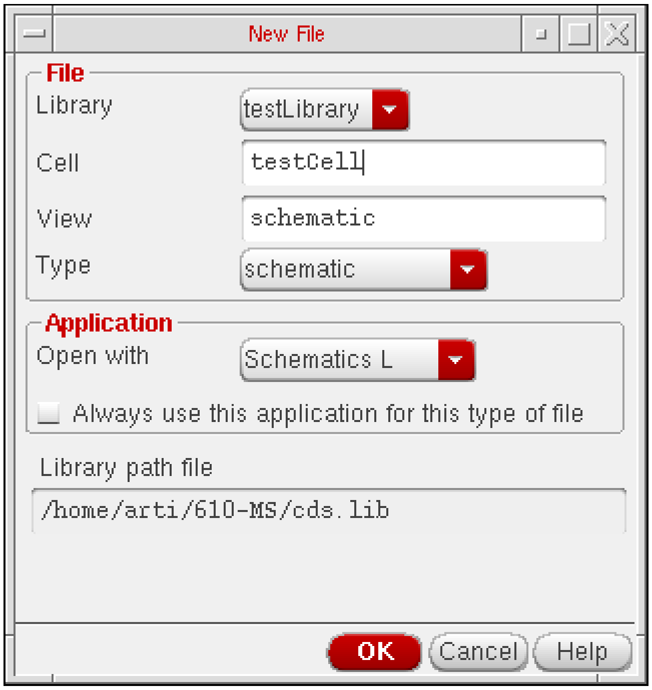
2. Select *File->Close* to close all the *What's New* windows.
3. Select *File->New->Library* from CIW.

4. Type `testLibrary` in the Name field and select the *Don't need process information* radio button.



5. Click *OK*.
6. Select *File->New->Cellview* from CIW.

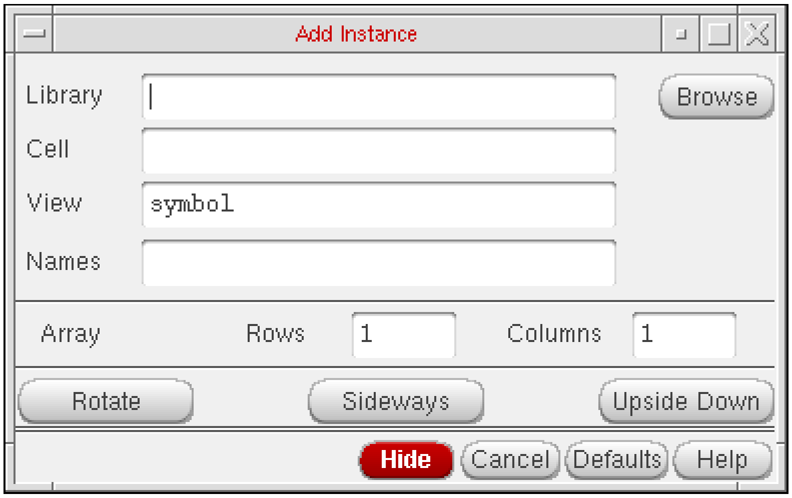
7. Select *testLibrary* in the Library Name field.



8. Type `testCell` in the Cell Name field and schematic in the View Name field.
9. Select *Schematic* from the Type list box.
10. Select the application from the Open with Tool list box and click *OK*.

    The new cell is opened in Virtuoso Schematic Editor.
11. Select *Create ->Instance* or click the *Create Instance* icon from the toolbar.

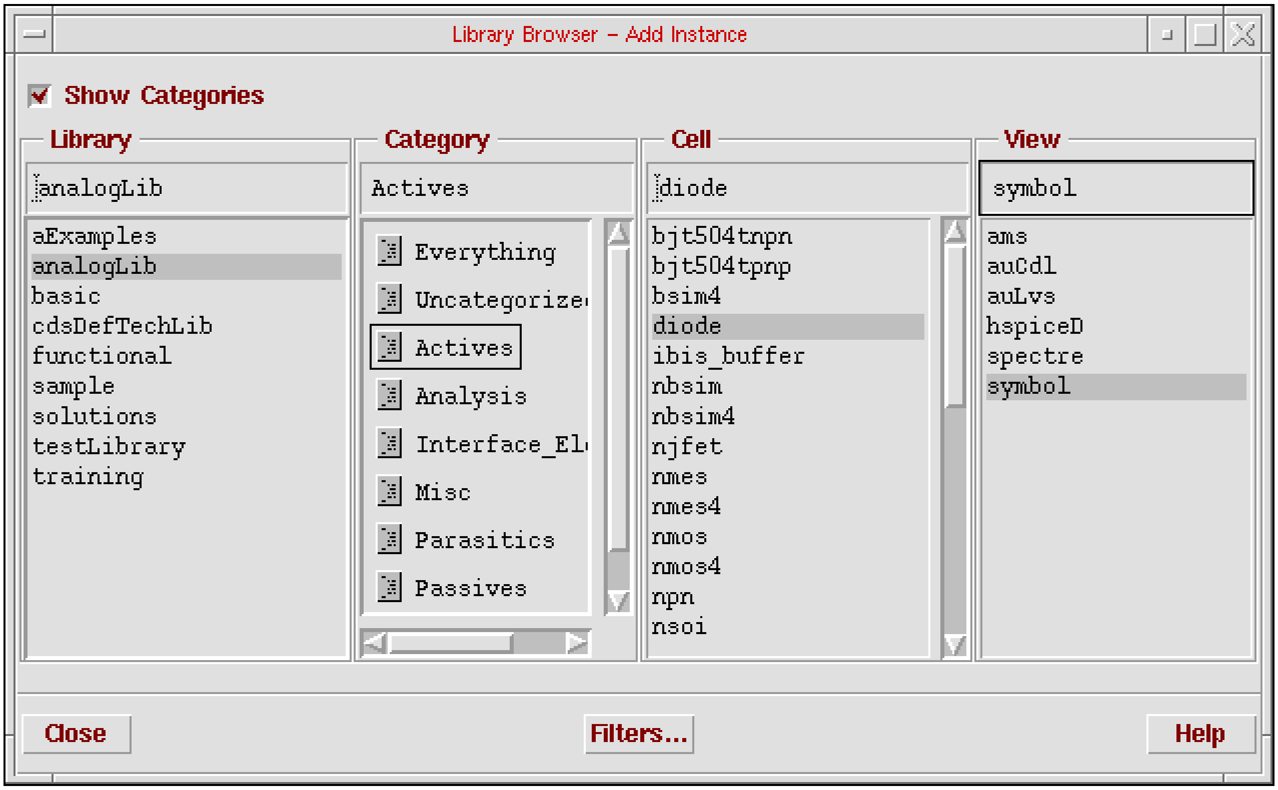
12. Click *Browse* from the Add Instance form.



13. Make sure that the *Show Categories* check box is selected in Library Browser.

14. Select `analogLib`, `Actives`, and `diode` from the Library, Category, and Cell list boxes respectively.

The View list box displays a list of the simulators that support the selected component. The `symbol` view applies to all components.

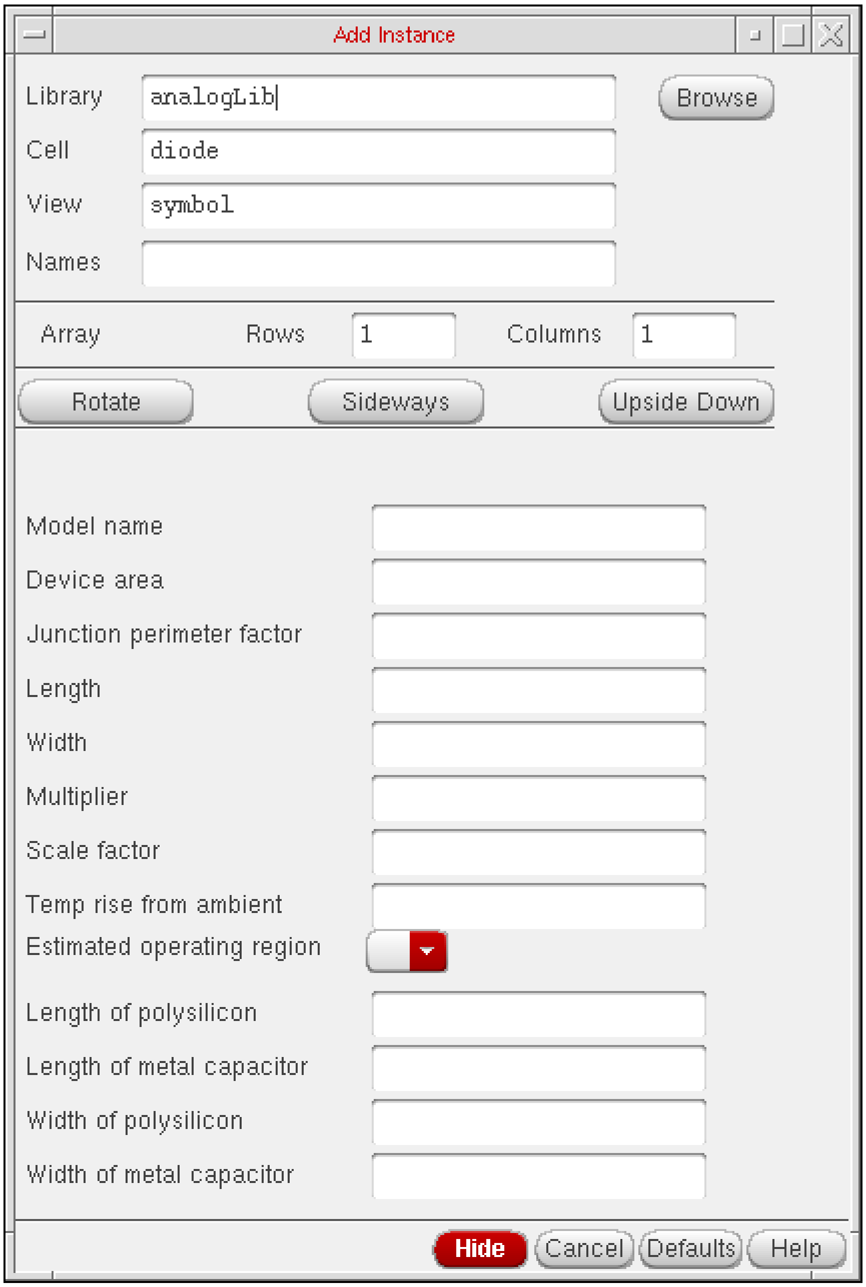


15. Select *symbol* from the View list box and click *Close*.

Notice the outline of the diode component when you move your cursor in the Virtuoso Schematic Editing window.

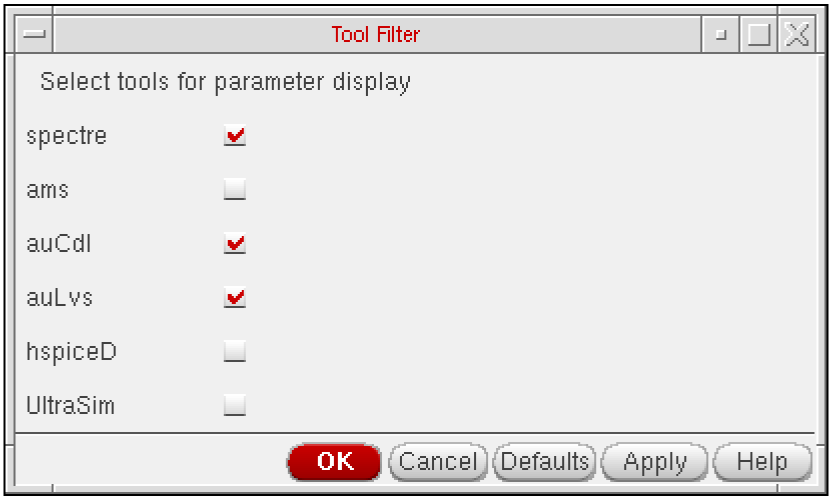
16. Click to place the component in the Virtuoso Schematic Editing window.

In the Add Instance form, notice that the library, cell, and view names appear in the Library, Cell, and View fields. The parameters for the selected component are also displayed.



These parameters are supported by the default simulators. To determine which simulators support which parameters, perform the following steps.

17. Select *Options->Tool Filter* from the Virtuoso Schematic Editing window. The Tool Filter form appears.

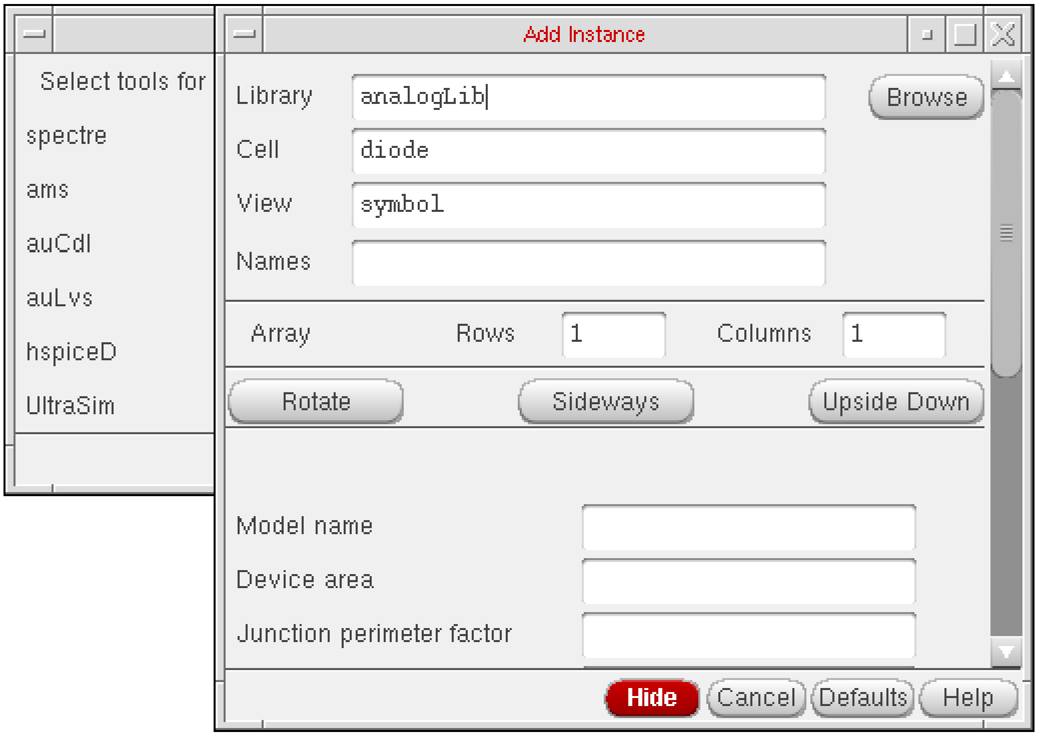


Notice that the default simulators are spectre, auCdl, and auLvs.

18. Deselect all tools and select only `spectre` from the Tool Filter form.

19. Click *Apply*.

Notice that the list of parameters in the Add Instance form changes to display only those parameters that are applicable for Spectre for the diode component.



In this way, you can identify those parameters of an analogLib component that are supported by specific simulators.

Note: The properties of components are retrieved from their corresponding CDF parameters.

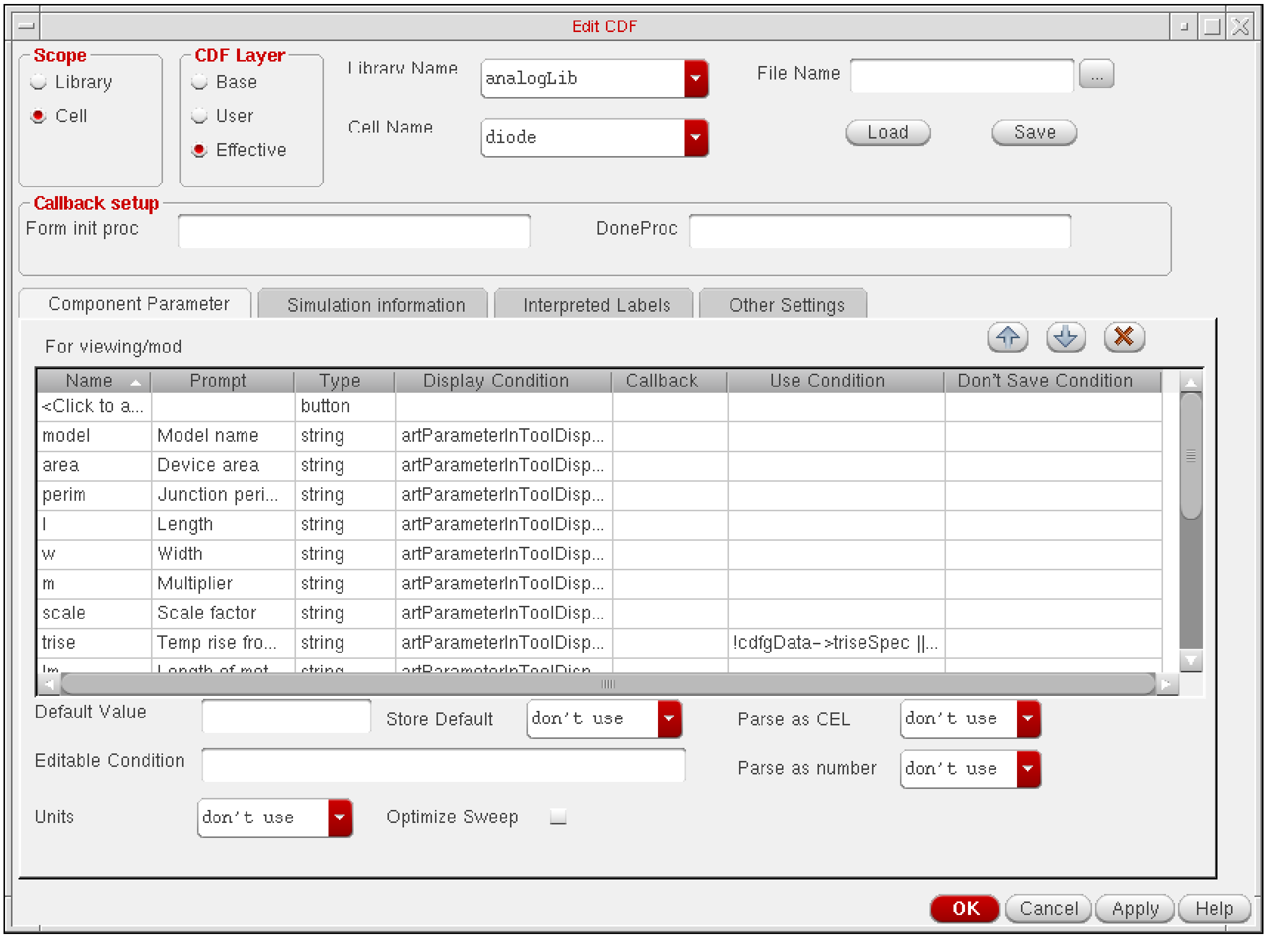
To view and edit the complete list of parameters for a component, perform the following steps.

1. Select *Tools->CDF->Edit* from CIW.

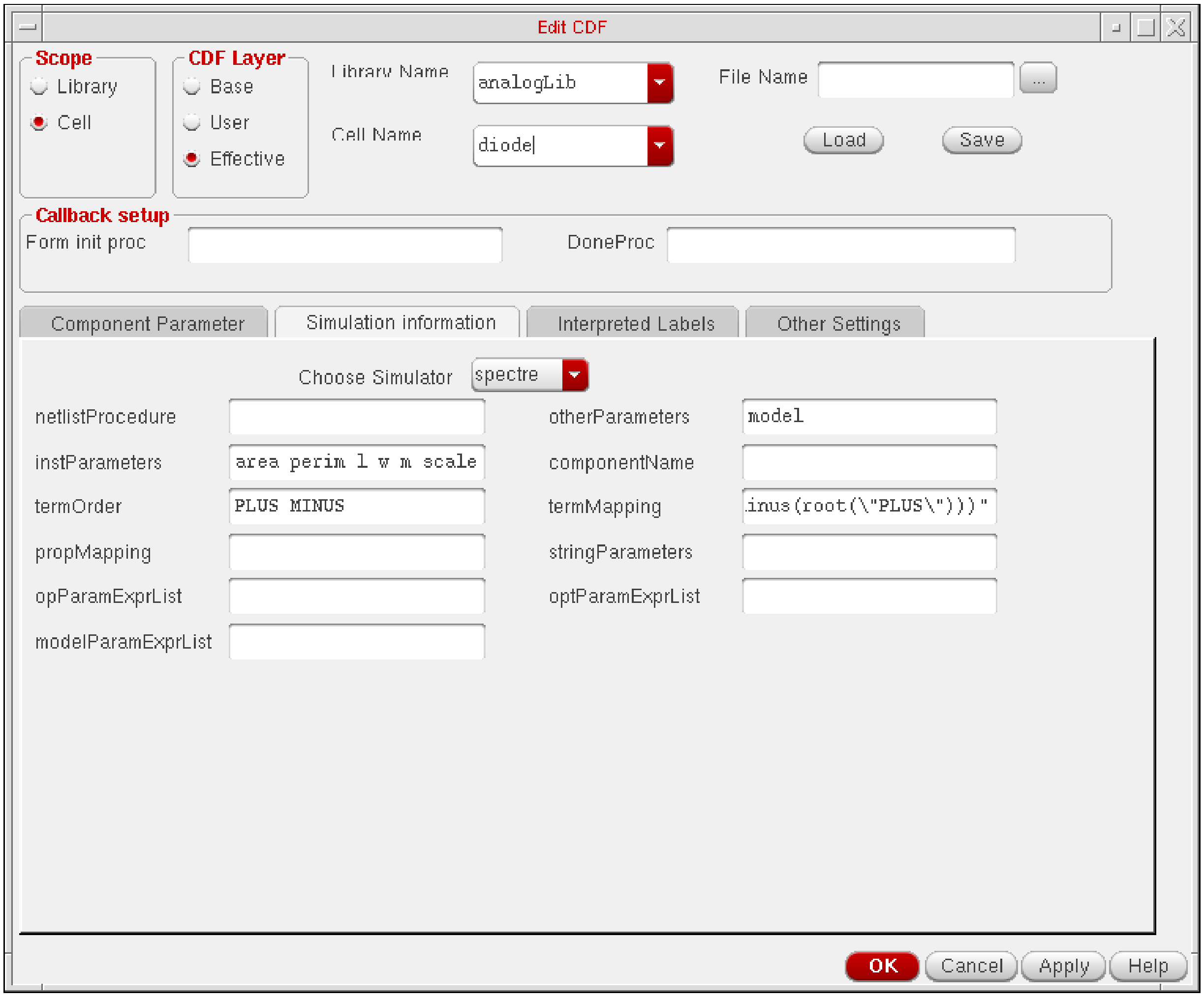
   The *Edit CDF* form appears.

2. Click *Browse* and select the library and cell names.

The Edit CDF form displays the complete list of parameters for the selected component as shown below:



Click *Simulation Information* and select a simulator to view the list of parameters that the simulator supports. The fields appear blank for those simulators that do not support the selected component.



For more information on viewing and editing the CDF descriptions of a component, refer to the *Component Description Format User Guide*. For modifying the simulation information refer to *Chapter 4, Component Description Format User Guide*.

Caution

As far as possible, use the standard analogLib components shipped with an IC release. Do not mix or merge analogLib components with internal simInfo or CDF parameters from an older release with those from a newer release. For example, if you modify a local copy of the pcccs/spectre cell from the IC5032 release, create a sub-circuit, and later try to netlist the design using a newer release, such as IC5033, then the sub-circuit might not work correctly. This is because the base-level cell CDF information in the IC5032 release and the IC5033 release might not be the same.

Note about this manual

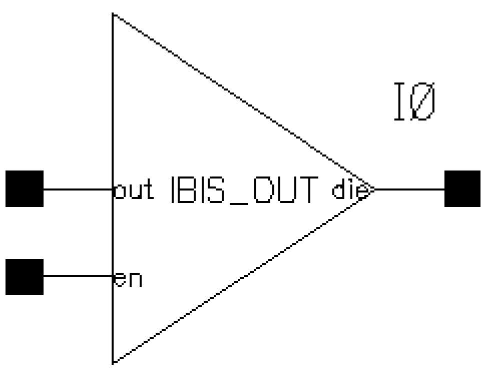
Although, multiple simulators may be supporting each component in analogLib, the descriptions, syntax, and examples used in this book are specific to Spectre. Components supported primarily by hspiceD are listed in Appendix B.

2

Active Components

All components listed in the **Actives** category require a defined model card. Each element maps to a specific Spectre primitive with respect to its instance parameters.

Symbol: ibis_buffer



IBIS buffer

The IBIS buffer model is based on the IBIS (I/O Buffer Information Specification) standard, version 3.2. The package and board models are not included in the buffer, they have to be added as separate subcircuits.

The ibis_buffer component is a p-cell that can have different pin combinations based on the selected buffer type. The supported buffer types are:

- input
- output
- io
- tristate

- opendrain and opensink
- ioopendrain and ioopensink
- opensource
- ioopensource
- terminator
- inputecl
- outputecl
- ioecl
- tristateecl

The following table lists the different pin combinations based on the buffer type. The presence of a pin is denoted by Y, absence of a pin is denoted by N, and optional pin is denoted by O.

Buffer Type	die/ pad pin	input	outp ut	enable	ground	pow er	ground clamp	power clamp	inverted die/pad pin
input	Y	Y	N	N	N	N	O	O	O
output	Y	N	Y	N	O	O	O	O	N
io	Y	Y	Y	Y	O	O	O	O	O
tristate	Y	N	Y	Y	O	O	O	O	N
opendrain opensink	Y	N	Y	N	O	N	O	N	N
ioopendrain ioopensink	Y	Y	Y	Y	O	N	O	N	O
opensource	Y	N	Y	N	N	O	N	O	N
ioopensource	Y	Y	Y	Y	N	O	N	O	O
terminator	Y	N	N	N	N	N	O	O	N
inputecl	Y	Y	N	N	N	N	O	O	O
outputecl	Y	N	Y	N	O	O	O	O	N
ioecl	Y	Y	Y	Y	O	O	O	O	O

Buffer Type	die/ pad pin	input	outp ut	enable	ground	pow er	ground clamp	power clamp	inverted die/pad pin
tristateecl	Y	N	Y	Y	O	O	O	O	N

For each buffer type there can be four variants, internal_power, external_power, differential _input, and diff_inp_and_ext_pwr. Therefore, ibis_buffer can have 44 variants as shown in the following table.

	Buffer Type	Varia nt	die/ pad	in	out	en	gnd	pwr	gnd_ c	pwr_ c	inv_ die
1	input	1	Y	Y	N	N	N	N	N	N	N
2	input	2	Y	Y	N	N	N	N	Y	Y	N
3	input	3	Y	Y	N	N	N	N	N	N	Y
4	input	4	Y	Y	N	N	N	N	Y	Y	Y
5	output	1	Y	N	Y	N	N	N	N	N	N
6	output	2	Y	N	Y	N	Y	Y	Y	Y	N
7	io	1	Y	Y	Y	Y	N	N	N	N	N
8	io	2	Y	Y	Y	Y	Y	Y	Y	Y	N
9	io	3	Y	Y	Y	Y	N	N	N	N	Y
10	io	4	Y	Y	Y	Y	Y	Y	Y	Y	Y
11	tristate	1	Y	N	Y	Y	N	N	N	N	N
12	tristate	2	Y	N	Y	Y	Y	Y	Y	Y	N
13	opendrain	1	Y	N	Y	N	N	N	N	N	N
14	opendrain	2	Y	N	Y	N	Y	N	Y	N	N
15	ioopendrain	1	Y	Y	Y	Y	N	N	N	N	N
16	ioopendrain	2	Y	Y	Y	Y	Y	N	Y	N	N
17	ioopendrain	3	Y	Y	Y	Y	N	N	N	N	Y
18	ioopendrain	4	Y	Y	Y	Y	Y	N	Y	N	Y
19	opensource	1	Y	N	Y	N	N	N	N	N	N

	Buffer Type	Variant	die/pad	in	out	en	gnd	pwr	gnd_c	pwr_c	inv_die
20	opensource	2	Y	N	Y	N	N	Y	N	Y	N
21	ioopensource	1	Y	Y	Y	Y	N	N	N	N	N
22	ioopensource	2	Y	Y	Y	Y	N	Y	N	Y	N
23	ioopensource	3	Y	Y	Y	Y	N	N	N	N	Y
24	ioopensource	4	Y	Y	Y	Y	N	Y	N	Y	Y
25	terminator	1	Y	N	N	N	N	N	N	N	N
26	terminator	2	Y	N	N	N	N	N	Y	Y	N
27	inputecl	1	Y	Y	N	N	N	N	N	N	N
28	inputecl	2	Y	Y	N	N	N	N	Y	Y	N
29	inputecl	3	Y	Y	N	N	N	N	N	N	Y
30	inputecl	4	Y	Y	N	N	N	N	Y	Y	Y
31	outputecl	1	Y	N	Y	N	N	N	N	N	N
32	outputecl	2	Y	N	Y	N	Y	Y	Y	Y	N
33	ioecl	1	Y	Y	Y	Y	N	N	N	N	N
34	ioecl	2	Y	Y	Y	Y	Y	Y	Y	Y	N
35	ioecl	3	Y	Y	Y	Y	N	N	N	N	Y
36	ioecl	4	Y	Y	Y	Y	Y	Y	Y	Y	Y
37	tristateecl	1	Y	N	Y	Y	N	N	N	N	N
38	tristateecl	2	Y	N	Y	Y	Y	Y	Y	Y	N
39	opensink	1	Y	N	Y	N	N	N	N	N	N
40	opensink	2	Y	N	Y	N	Y	N	Y	N	N
41	ioopensink	1	Y	Y	Y	Y	N	N	N	N	N
42	ioopensink	2	Y	Y	Y	Y	Y	N	Y	N	N
43	ioopensink	3	Y	Y	Y	Y	N	N	N	N	Y
44	ioopensink	4	Y	Y	Y	Y	Y	N	Y	N	Y

Based on the model you have selected, you can create two types of ibis_buffer:

- with an external model card

  This is the default option. If you specify the model name the netlist is as follows:

```
b1 (1 2 3) "Model name" <other instance parameters>
```

  For example, the netlist of an ibis_buffer with buffer type = `tristate`, buffer variant = `internal_power`, model name = `SN74_OUT_33_Typ_27degC`, polarity = `inv`, differential threshold = `1.2V`, delay time = `1ms`, delay schedule = `yes`, different element delays = `1p`, `2p`, `5p`, and `2p`, is as follows:

```
I65 (net013 net011 net012) SN74_OUT_33_Typ_27degC polarity=inv \
vdiff=1.2 delay=1m delay_schedule=[1p 2p 5p 2p]
```

- with an IBIS file

  If you specify an IBIS buffer file, then three additional parameters are displayed. In this case the netlist is as follows:

```
b1 (1 2 3) ibis_buffer file="IBIS file name" model="IBIS model name"
corner="IBIS model corner" <other instance parameters>
```

  For example, the netlist with the additional parameters IBIS filename = `~/main.scs`, IBIS modelname = `IBIS_Model`, corner = `typical` is as follows:

```
I65 (net013  net011 net012) ibis_buffer file="~/main.scs" \
model="IBIS_Model" corner=typical polarity=inv vdiff=1.2 delay=1m \
delay_schedule=[1p 2p 5p 2p]
```

Command-line help

```
spectre -h ibis_buffer
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Select IBIS Buffer Type	bufferType	x	-	-	x	-
param0	param0	x	-	-	x	-
Select IBIS Buffer Variant	bufferVariant2	x	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Select IBIS Buffer Variant	bufferVariant4	x	-	-	x	-
IBIS Entry Method	ibisEntryMethod	x	-	-	x	-
Model name	model	x	-	-	x	-
IBIS file name	ibisFile	x	-	-	x	-
IBIS model name	ibisModelName	x	-	-	x	-
IBIS corner	ibisCorner	x	-	-	x	-
Polarity of the buffer	polarity	x	-	-	x	-
Differential threshold	vdiff	x	-	-	x	-
Delay Time	delay	x	-	-	x	-
Delay Schedule	ibisDelaySchedule	x	-	-	x	-
Rise on delay	rise_on_dly	x	-	-	x	-
Rise off delay	rise_off_dly	x	-	-	x	-
Fall on delay	fall_on_dly	x	-	-	x	-
Fall off delay	fall_off_dly	x	-	-	x	-

Syntax/Synopsis

```
Name ( die [inp] [out] [en] [gnd] [pwr] [gnd_c] [pwr_c] [inv_die] ) ModelName
        <parameter=value> ...
```

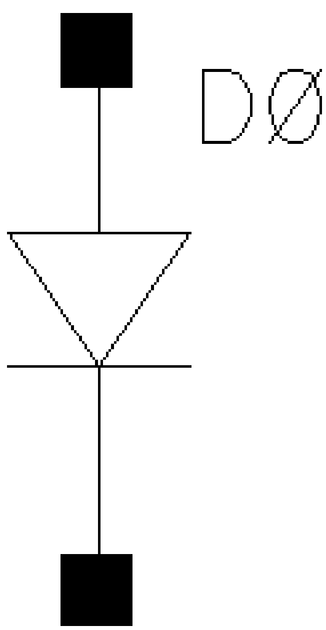
```
Name ( die [inp] [out] [en] [gnd] [pwr] [gnd_c] [pwr_c] [inv_die] )
        ibis_buffer <parameter=value> ...
```

Example

```
I65 (net013 net011 net012) SN74_OUT_33_Typ_27degC polarity=inv \
vdiff=1.2 delay=1m delay_schedule=[1p 2p 5p 2p]
```

```
b1 (1 2 3) ibis_buffer file="IBIS file name" model="IBIS model name"
corner="IBIS model corner" <other instance parameters>
```

Symbol: diode



Junction Diode

The junction diode model includes nonlinear junction capacitance and reverse breakdown.

Command-line help

```
spectre -h diode
```

Component Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	x	-
Device area	`area`	x	-	-	x	-
Device initially off	`off`	-	-	-	x	-
Initial diode voltage	`Vd`	-	-	-	x	-
Junction perimeter factor	`perim`	x	-	-	-	-
Length	`l`	x	-	-	x	-
Width	`w`	x	-	-	x	-
Multiplier	`m`	x	-	-	x	-
Scale factor	`scale`	x	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-
Periphery of junction	`pj`	-	-	-	x	-
Width of polysilicon	`wp`	x	-	-	x	-
Length of polysilicon	`lp`	x	-	-	x	-
Width of metal capcitor	`wm`	x	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Length of metal capcitor	`lm`	x	-	-	x	-
Temperature difference	`dtemp`	-	-	-	x	-

Syntax/Synopsis

```
Name ( a c ) ModelName <parameter=value> ...
```

In the forward operation the voltage on the anode (‘a’) is more positive than the voltage on the cathode (‘c’).

Model Synopsis

```
model ModelName diode <parameter=value> ...
```

Example

```
d0 (dp dn) pdiode l=3e-4 w=2.5e-4 area=1
```

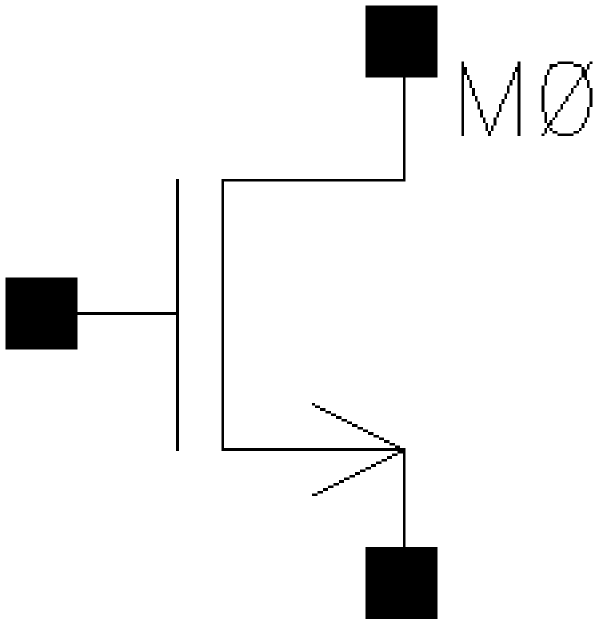
Sample model statement

```
model pdiode diode is=1.8e-5 rs=1.43 n=1.22 nz=2.31 gleak=6.2e-5
rsw=10 isw=6.1e-10 ibv=0.95e-3 tgs=2 ik=1.2e7 fc=0.5 cj=1.43e-3
pb=0.967 mj=0.337 cjsw=2.76e-9 vjsw=0.94 jmax=1e20
```

Additional Information

This device is supported within the altergroups.

Symbol: nbsim



N-type BSIM Field Effect Transistor

nbsim is an n-channel BSIM model.

Command-line help

For related information on MOS, use any of the following help commands:

```
spectre -h bsim1

spectre -h bsim2

spectre -h bsim3

spectre -h bsim3v3
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	x	-
Bulk node connection	`bn`	-	-	-	-	-
Multiplier	`m`	x	x	x	x	-
Width	`w`	x	x	x	x	-
Length	`l`	x	x	x	x	-
Drain diffusion area	`ad`	x	-	-	x	-
Source diffusion area	`as`	x	-	-	-	-
Drain diffusion periphery	`pd`	x	-	-	x	-
Source diffusion periphery	`ps`	x	-	-	x	-
Drain diffusion res squares	`nrd`	x	-	-	x	-
Source diffusion res squares	`nrs`	x	-	-	x	-
Drain diffusion length	`ld`	x	-	-	-	-
Source diffusion length	`ls`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
NQS flag	`nqsmod`	x	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-
Device initially off	`off`	-	-	-	x	-
Drain source initial voltage	`Vds`	-	-	-	x	-
Gate source initial voltage	`Vgs`	-	-	-	x	-
Bulk source initial voltage	`Vbs`	-	-	-	x	-
Additional drain resistance	`rdc`	x	-	-	x	-
Additional source resistance	`rsc`	x	-	-	x	-
Dist. OD & poly(one side)	`sa`	x	-	-	-	-
Dist. OD & poly(other side)	`sb`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Dist. betn neighbour fingers	sd	x	-	-	-	-
Temperature difference	dtemp	-	-	-	x	-
Source/ drain selector	geo	x	-	-	x	-

Syntax/Synopsis

```
Name ( d g s b ) ModelName <parameter=value> ...
```

Model Synopsis

```
model ModelName bsim1 <parameter=value> ...
```

Example

Sample Instance Statement

```
m1 (1 2 0 0) nchmod l=5u w=10u as=40u ad=40u pd=28u ps=28u m=1
```

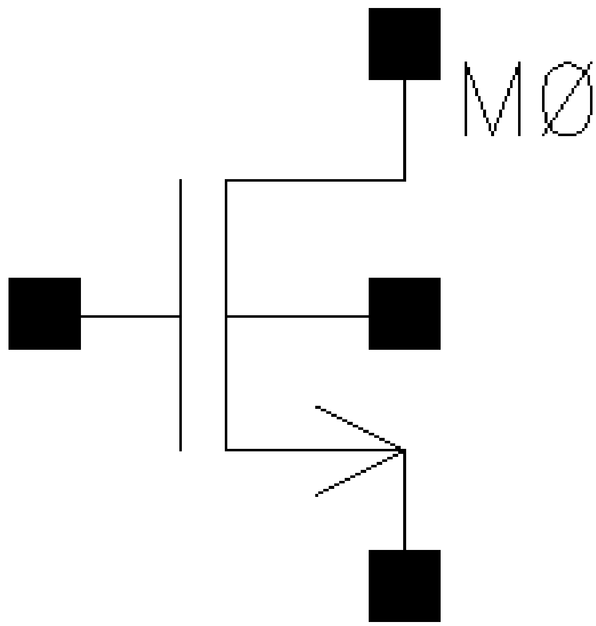
Sample Model Statement

```
model nchmod bsim1 vfb0=-0.5 lvfb=0.5 wvfb=0.3 phi0=0.8 eta0=0.056 k1=0.5 muz=454
eg=0.99 gap1=5.5e-04 trs=1e-3 trd=1e-3 xpart=0.5 rs=10 rd=10
```

Additional Information

This device is supported within the altergroups.

Symbol: nbsim4



N-type BSIM MOS transistor (4 terminals)

BSIM4 is the version-4.21 of the bsim model. BSIM4 transistors require you to use a model statement.

Command-line help

For related information on MOS, use any of the following help commands:

```
spectre -h bsim4
```

```
spectre -h bsim1
```

```
spectre -h bsim2
```

```
spectre -h bsim3
```

```
spectre -h bsim3v3
```

CDF Parameters

The CDF parameters for `nbsim4` are the same as the CDF parameters for `nbsim`.

Syntax/Synopsis

```
Name ( d g s b ) ModelName <parameter=value> ...
```

Model Synopsis

```
model ModelName bsim4 <parameter=value> ...
```

Example

```
m4 (0 2 1 1) pchmod w=2u l=0.8u as=250p ad=250p pd=168p ps=168p m=1
```

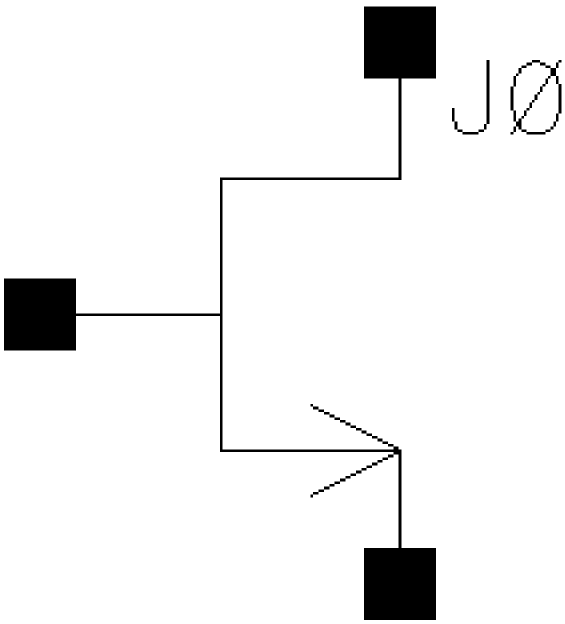
Sample Model Statement

```
model pchmod bsim4 type=p mobmod=0 capmod=2 version=4.21 toxe=3e-9
cdsc=2.58e-4 cdscb=0 cdscd=6.1e-8 cit=0 nfactor=1.1 xj=9e-8
vfb=0.76vsat=9.2e4 at=3.3e4 a0=1.1 ags=1.0e-20 a1=0 ngate=9e19
vth0=-0.42a1=0 a2=1 delta=0.014 pvag=1e-20 pclm=6.28e-4 pdits=0.2
pditsl=2.3e6pditsd=0.23 fprout=0.2 pdiblcb=3.4e-8 pdiblc1=0.81
drout=0.56pdiblc2=9.84e-6 pscbe1=8.14e8 pscbe2=9.58e-07 lint=5e-9
wint=5e-9dmcg=5e-6 dmci=5e-6 dmdg=5e-6 dmcgt=6e-7 dwj=4.5e-8
rsh=6cgso=7.43e-10 cgdo=7.43e-10 cgbo=2.56e-11 cgsl=1e-14
cgdl=1e-14ckappas=0.5 ckappad=0.5 noff=0.9 voffcv=0.02 acde=1 moin=15
xpart=0kt1l=0 kt2=2.2e-2 lpe0=5.75e-8 lpeb=2.3e-10 dvt0=2.89
dvt1=0.53dvt2=-3.2e-2 dvt0w=0 dvt1w=0 dvt2w=0 dvtp0=7.32e-7
dvtp1=0.12dsub=0.058 eta0=0.001 u0=4.19e-2 ua=8.7e-16 ub=3.06e-18
k1=0.33uc=4.6e-13 ute=-1.5 ua1=4.31e-9 ub1=7.61e-18 uc1=-5.6e-11
k2=-1.87e-2rdsw=369.4 rdw=184.7 rsw=184.7 prwg=3.22e-8 prwb=6.8e-11
wr=1rdswmin=0 rdwmin=0 rswmin=0 prt=0 b0=-1e-20 k3=80 k3b=0
w0=2.5e-6b1=0 keta=-0.047 alpha0=7.4e-2 alpha1=0.005 beta0=30
```

Additional Information

This device is supported within the altergroups.

Symbol: njfet



N-type Junction Field Effect Transistor

The JFET model is derived from the FET model of Shichman and Hodges. JFETs require you to use a model statement.

Command-line help

```
spectre -h jfet
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Bulk node connection	`bn`	-	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Device area	`area`	x	-	-	x	-
Device initially off	`off`	-	-	-	x	-
Drain source initial voltage	`Vds`	-	-	-	x	-
Gate source initial voltage	`Vgs`	-	-	-	x	-
Gate to bulk and src voltage	`Vgbs`	-	-	-	x	-
Multiplier	`m`	x	-	-	x	-
Estimated operating region	`region`	x	-	-	-	-
Width	`w`	-	-	-	x	-
Length	`l`	-	-	-	x	-
Temperature difference	`dtemp`	-	-	-	x	-

Syntax/Synopsis

```
Name ( d g s [b] ) ModelName <parameter=value> ...
```

You do not have to specify the back gate terminal when you use the four-terminal model. If left unspecified, the substrate is connected to ground.

Model Synopsis:

```
model ModelName jfet <parameter=value> ...
```

Example

```
jf1 (net1 net2 0) jmod area=1
```

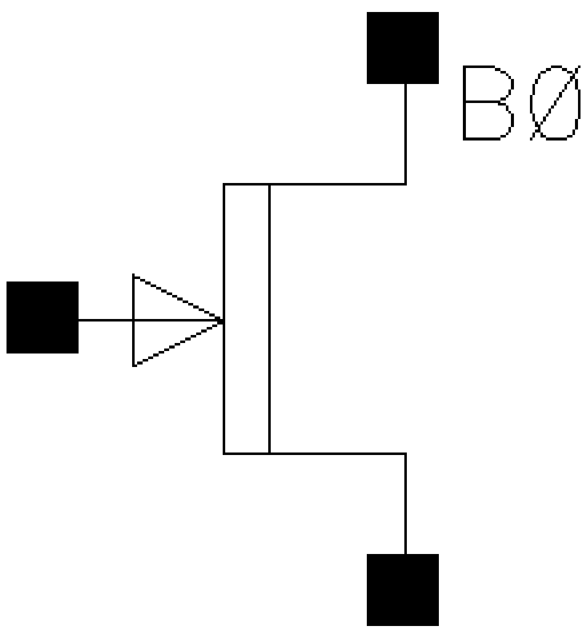
Sample Model Statement:

```
model jmod jfet beta=9e-5 lambda=0 type=n vt0=-18.7 rd=10 rs=10 cgs=1.3e-13 pb=0.65
```

Additional Information

This device is supported within the altergroups.

Symbol: nmes



N-type MES FET Transistor

The GaAs MESFET model was derived from the model by H. Statz and others at Raytheon. This model is completely symmetric and is modified slightly to make it charge conserving. GaAs MESFET instances require that you use a model statement.

Command-line help

```
spectre -h gaas
```

```
spectre -h tom2

spectre -h tom3
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Bulk node connection	`bn`		-	-	-	-
Device area	`area`	x	-	-	x	-
Device initially off	`off`	-	-	-	x	-
Drain source initial voltage	`Vds`	-	-	-	x	-
Gate source initial voltage	`Vgs`	-	-	-	x	-
Bulk source initial voltage	`Vbs`	-	-	-	x	-
Multiplier	`m`	x	-	-	x	-
Estimated operating region	`region`	x	-	-	-	-
Width	`w`	-	-	-	x	-
Length	`l`	-	-	-	x	-
Temperature difference	`dtemp`	-	-	-	x	-

Syntax/Synopsis

```
Name ( d g s ) ModelName <parameter=value> ...
```

Model Synopsis:

```
model ModelName gaas <parameter=value> ...
```

Example

```
m1 (1 2 0) nmes area=1 m=2
```

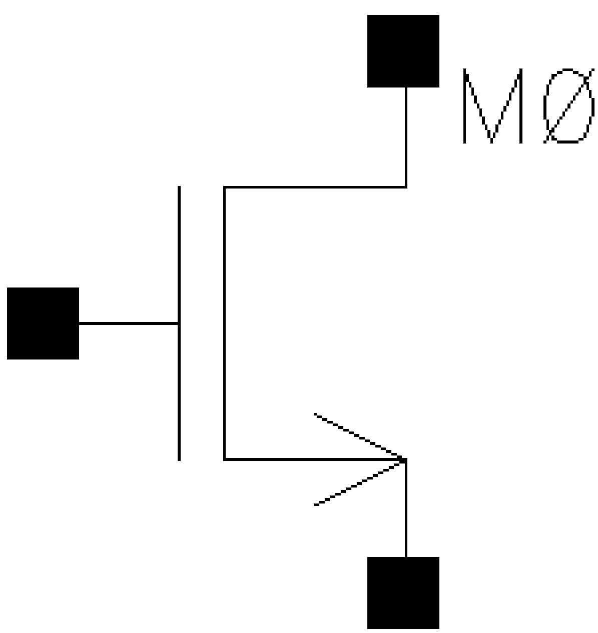
Sample Model Statement:

```
model nmes gaas type=n vto=-2 beta=0.06 lambda=0 b=0.25 rs=3.65 alpha=1.9 rd=1.98
is=1.1e-9 n=1.28 fc=0.5 cgs=0.365e-12
```

Additional Information

This device is supported within the altergroups.

Symbol: nmos



N-type Generic MOS Transistor (3 terminals)

Command-line help

For related information on MOS, use any of the following help commands:

```
spectre -h mos0
```

```
spectre -h mos1
```

```
spectre -h ekv
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	x	-
Bulk node connection	`bn`	-	-	-	-	-
Multiplier	`m`	x	x	x	x	-
Width	`w`	x	x	x	x	-
Length	`l`	x	x	x	x	-
Drain diffusion area	`ad`	x	-	-	x	-
Source diffusion area	`as`	x	-	-	x	-
Drain diffusion periphery	`pd`	x	-	-	x	-
Source diffusion periphery	`ps`	x	-	-	x	-
Drain diffusion res squares	`nrd`	x	-	-	x	-

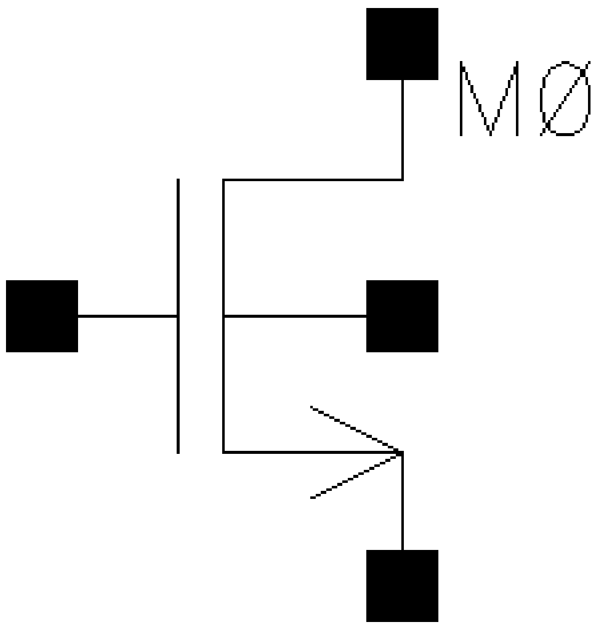
CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Source diffusion res squares	nrs	x	-	-	x	-
Drain diffusion length	ld	x	-	-	-	-
Source diffusion length	ls	x	-	-	-	-
Device initially off	off	-	-	-	x	-
Drain source initial voltage	Vds	-	-	-	x	-
Gate source initial voltage	Vgs	-	-	-	x	-
Bulk source initial voltage	Vbs	-	-	-	x	-
Temp rise from ambient	trise	x	-	-	-	-
Estimated operating region	region	x	-	-	-	-
Hot-electron degradation	degradation	x	-	-	-	-
Additional drain resistance	rdc	x	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Additional source resistance	`rsc`	x	-	-	x	-
Dist. OD & poly(one side)	`sa`	x	-	-	-	-
Dist. OD & poly(other side)	`sb`	x	-	-	-	-
Dist. betn neighbour fingers	`sd`	x	-	-	-	-
Temperature difference	`dtemp`	-	-	-	x	-
Source/ drain selector	`geo`	x	-	-	x	-

Example

```
M0 (net3 net1 net2) nmos
```

Symbol: nmos4



N-type Generic MOS Transistor (4 terminals)

Command-line help

For related information on MOS, use any of the following help commands:

```
spectre -h mos0
```

```
spectre -h mos1
```

```
spectre -h ekv
```

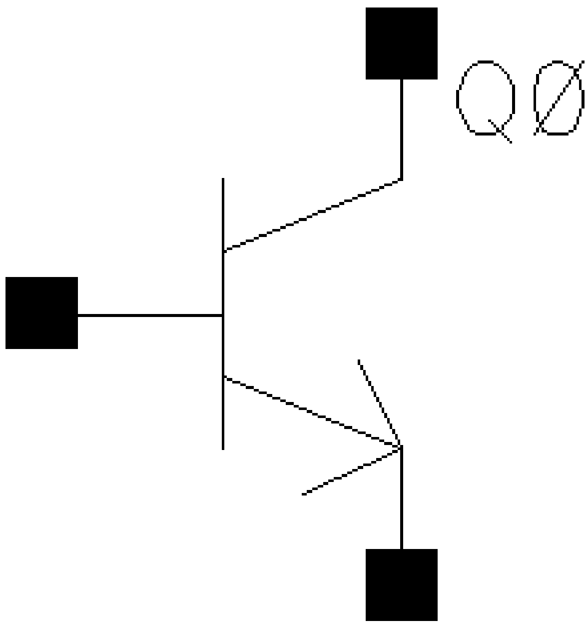
CDF Parameters

The CDF parameters for `nmos4` are the same as the CDF parameters for `nmos`.

Example

```
M0 (net1 net3 net4 net2) nmos4
```

Symbol: npn



Generic Bipolar Transistor (NPN)

npn is an ntype bjt.

Command-line help

`spectre -h bjt`

`spectre -h bjt2`

`spectre -h bjt3`

`spectre -h bjt301`

`spectre -h vbic`

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	x	-
Bulk node connection	`bn`	-	-	-	-	-
Device area	`area`	x	-	-	x	-
Base-emitter voltage	`Vbe`	-	-	-	x	-
Collector-emitter voltage	`Vce`	-	-	-	x	-
Device initially off	`off`	-	-	-	x	-
Multiplier	`m`	x	-	-	x	-
Temp rise from ambient	`trise`	x	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-
Temperature difference	`dtemp`	-	-	-	x	-
Base area	`areab`	-	-	-	x	-
Collector area	`areac`	-	-	-	x	-
Temp Rise Specifier	`triseSpec`	x	-	-	-	-
dtmp -Temp rise from ambient	`dtmp`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
dtemp - Temp rise from ambient	`dtempn`	-	-	-	-	-

Syntax/Synopsis

```
Name ( c b e [s] ) ModelName <parameter=value> ...
```

You do not have to specify the substrate terminal. If you do not specify it, the substrate is connected to ground.

Model Synopsis:

```
model ModelName bjt <parameter=value> ...
```

Example

```
q1 (vcc net3 minus) npn_mod region=fwd area=1 m=1
```

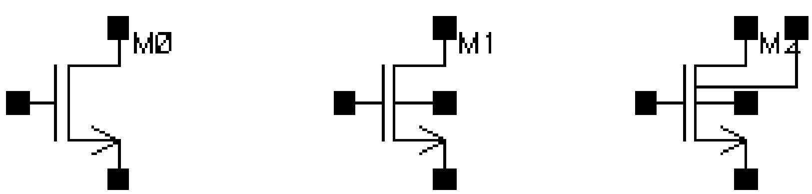
Following is a sample model statement:

```
model npn_mod bjt type=npn is=10e-13 bf=200 va=58.8 ikf=5.63e-3 rb=700 rbm=86
re=3.2 cje=0.352e-12 pe=0.76 me=0.34 tf=249e-12 cjc=0.34e-12 pc=0.55
```

Additional Information

This device is supported within the altergroups.

Symbol: ntft



N-Type Poly-Si TFT (NTFT)

ntft is an n-type polysilicon tft. It can have a maximum of five terminals with drain, gate and source being mandatory terminals and substrate and thermal being optional.

The diagrams show the terminal with none, one or two optional nodes selected.

Command-line help

```
spectre -h psitft
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-	-	-	-
Optional Nodes	`Opins`	x	-	-	-	-	-	-	-
Optional Bulk Node _B	`bulknode`	x	-	-	-	-	-	-	-
Optional Thermal Node _T	`pinT`	x	-	-	-	-	-	-	-
Width	`w`	x	-	-	-	-	-	-	-
Length	`l`	x	-	-	-	-	-	-	-
Drain diffusion res squares	`nrd`	x	-	-	-	-	-	-	-
Source diffusion res squares	`nrs`	x	-	-	-	-	-	-	-
Multiplier	`m`	x	-	-	-	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Thermal resistance	`rth0`	x	-	-	-	-	-	-	-
Thermal capacitance	`cth0`	x	-	-	-	-	-	-	-
Num of segments	`nseg`	x	-	-	-	-	-	-	-

Syntax/Synopsis

```
Name ( d g s [b] [t] ) ModelName <parameter=value> ...
```

Model Synopsis

```
model ModelName psitft <parameter=value> ...
```

Sample Instance Statement

```
m4 (0 2 1 1 3) nch w=2u l=0.8u
```

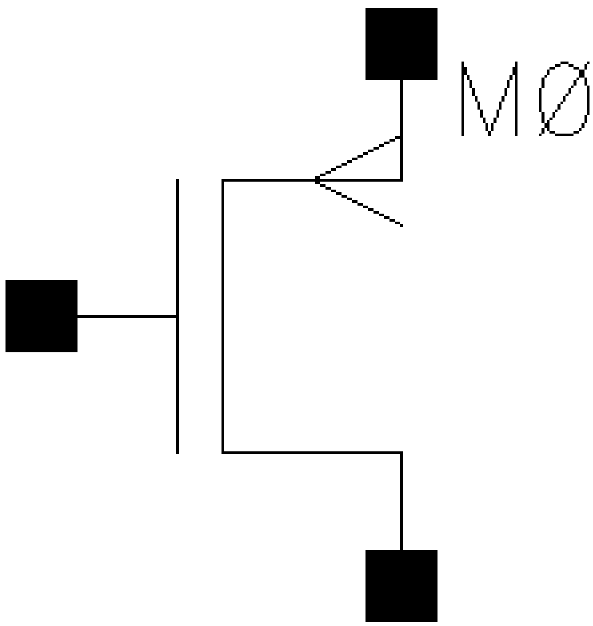
Sample Model Statement

```
model nch psitft type=n
```

Additional Information

This device is supported within the altergroups.

Symbol: pbsim



P-type BSIM MOS Transistor (3 terminals)

pbsim is a p-channel BSIM model.

Command-line help

For related information on MOS, use any of the following help commands:

```
spectre -help bsim1

spectre -help bsim2

spectre -help bsim3

spectre -help bsim3v3
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	x	-
Bulk node connection	`bn`	-	-	-	-	-
Multiplier	`m`	x	x	x	x	-
Width	`w`	x	x	x	x	-
Length	`l`	x	x	x	x	-
Drain diffusion area	`ad`	x	-	-	x	-
Source diffusion area	`as`	x	-	-	x	-
Drain diffusion periphery	`pd`	x	-	-	x	-
Source diffusion periphery	`ps`	x	-	-	x	-
Drain diffusion res squares	`nrd`	x	-	-	x	-
Source diffusion res squares	`nrs`	x	-	-	x	-
Drain diffusion length	`ld`	x	-	-	-	-
Source diffusion length	`ls`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
NQS flag	`nqsmod`	x	-	-	-	-
Device initially off	`off`	-	-	-	x	-
Drain source initial voltage	`Vds`	-	-	-	x	-
Gate source initial voltage	`Vgs`	-	-	-	x	-
Bulk source initial voltage	`Vbs`	-	-	-	x	-
Additional drain resistance	`rdc`	x	-	-	x	-
Additional source resistance	`rsc`	x	-	-	x	-
Temp rise from ambient	`trise`	x	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-
Dist. OD & poly(one side)	`sa`	x	-	-	-	-
Dist. OD & poly(other side)	`sb`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Dist. betn neighbour fingers	sd	x	-	-	-	-
Temperature difference	dtemp	-	-	-	x	-
Source/ drain selector	geo	x	-	-	x	-

Syntax/Synopsis

```
Name ( d g s b ) ModelName <parameter=value> ...
```

Following is the model synopsis:

```
model ModelName bsim1 <parameter=value> ...
```

Example

```
m1 (1 2 0 0) nchmod l=5u w=10u as=40u ad=40u pd=28u ps=28u m=1
```

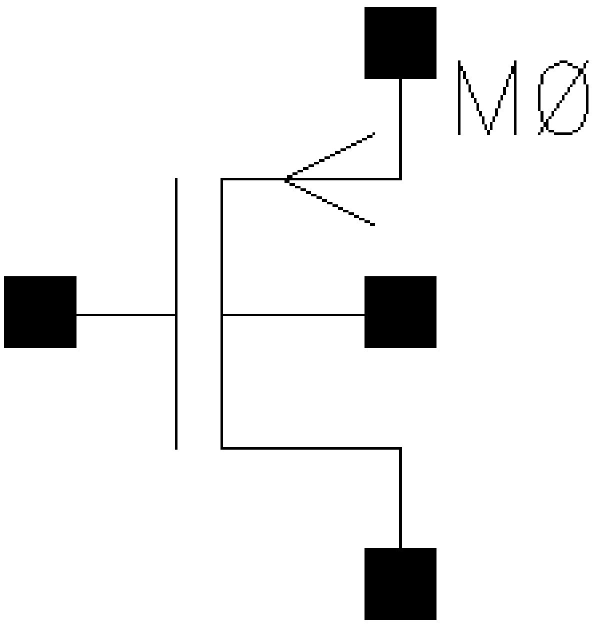
Following is the sample model statement:

```
model nchmod bsim1 vfb0=-0.5 lvfb=0.5 wvfb=0.3 phi0=0.8 eta0=0.056 k1=0.5
muz=454 eg=0.99 gap1=5.5e-04 trs=1e-3 trd=1e-3 xpart=0.5 rs=10 rd=10
```

Additional Information

This device is supported within the altergroups.

Symbol: pbsim4



P-type BSIM MOS transistor (4 terminals)

pbsim is a p-channel BSIM model.

Command-line help

For related information on MOS, use any of the following help commands:

```
spectre -help bsim1

spectre -help bsim2

spectre -help bsim3

spectre -help bsim3v3
```

CDF Parameters

The CDF parameters for `pbsim4` are the same as the CDF parameters for `pbsim`.

Syntax/Synopsis

```
Name ( d g s b ) ModelName <parameter=value> ...
```

Following is the model synopsis:

```
model ModelName bsim1 <parameter=value> ...
```

Example

```
m1 (1 2 0 0) nchmod l=5u w=10u as=40u ad=40u pd=28u ps=28u m=1
```

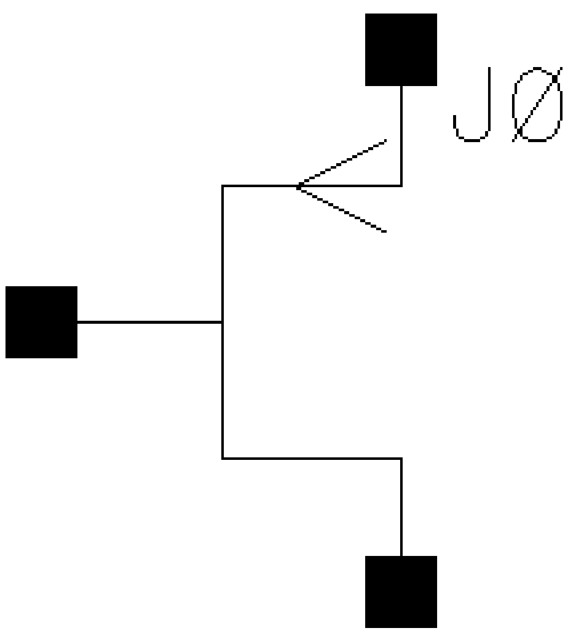
Following is the sample model statement:

```
model nchmod bsim1 vfb0=-0.5 lvfb=0.5 wvfb=0.3 phi0=0.8 eta0=0.056 k1=0.5
muz=454 eg=0.99 gap1=5.5e-04 trs=1e-3 trd=1e-3 xpart=0.5 rs=10 rd=10
```

Additional Information

This device is supported within the altergroups.

Symbol: pjfet



P-type Junction Field Effect Transistor

The JFET model is derived from the FET model of Shichman and Hodges. JFETs require you to use a model statement.

Command-line help

```
spectre -h jfet
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Bulk node connection	`bn`	-	-	-	-	-
Device area	`area`	x	-	-	x	-
Device initially off	`off`	-	-	-	x	-
Drain source initial voltage	`Vds`	-	-	-	x	-
Gate source initial voltage	`Vgs`	-	-	-	x	-
Gate to bulk and src voltage	`Vgbs`	-	-	-	x	-
Multiplier	`m`	x	-	-	x	-
Width	`w`	-	-	-	x	-
Length	`l`	-	-	-	x	-
Temperature difference	`dtemp`	-	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Estimated operating region	`region`	x	-	-	-	-

Syntax/Synopsis

```
Name ( d g s [b] ) ModelName <parameter=value> ...
```

You do not have to specify the back gate terminal when you use the four-terminal model. If left unspecified, the substrate is connected to ground.

Model Synopsis:

```
model ModelName jfet <parameter=value> ...
```

Example

```
jf1 (net1 net2 0) jmod area=1
```

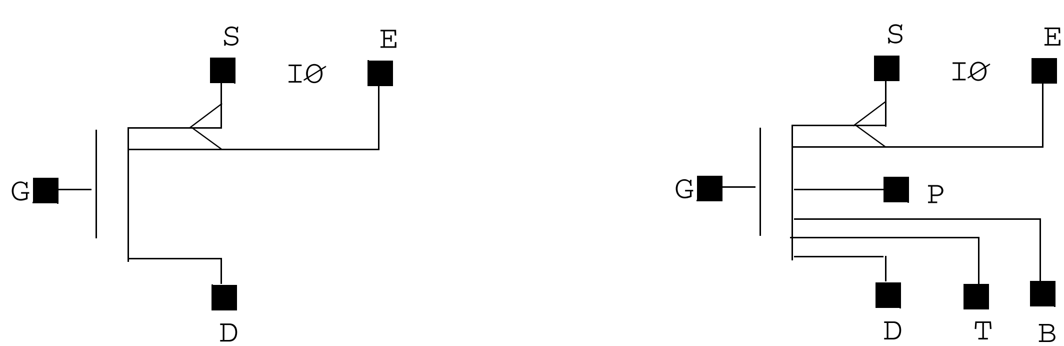
Following is a sample model statement:

```
model jmod jfet beta=9e-5 lambda=0 type=n vt0=-18.7 rd=10 rs=10 cgs=1.3e-13 pb=0.65
```

Additional Information

This device is supported within the altergroups.

Symbol: psoip



P-type BSIM SOI model

psoip is a p-type BSIM SOI model

Command-line help

```
spectre -h bsimsoi
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Temperature Node Present	`TnodeOut( earlier this was bn,therefore check)`	x	-	-	-	-
Thermal Node(T)	Tnode	x	-	-	-	-
Ext. Body Contact (PinP)	PinP	x	-	-	-	-
Body Node	BodyNodePin	x	-	-	-	-
Width	`w`	x	-	-	-	-
Length	`l`	x	-	-	-	-
Source diffusion area	`as`	x	-	-	-	-
Drain diffusion area	`ad`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Source diffusion periphery	ps	x	-	-	-	-
Drain diffusion periphery	pd	x	-	-	-	-
Drain diffusion res squares	nrd	x	-	-	-	-
Source diffusion res squares	nrs	x	-	-	-	-
Multiplier	m	x	-	-	-	-

Syntax/Synopsis

```
Name ( d g s e [p] [b] [t] ) ModelName <parameter=value> ...
```

Following is the model synopsis:

```
model ModelName bsimsoi <parameter=value> ...
```

Example

```
I7 (0 net9 vdd! vdd!) bsimsoi w=1u l=1u
```

Following is the sample model statement:

```
model psoip_model bsimsoi type = p beta0 = 0 dvt = -0.032 delta = 0.01 k1 = 0.6
xbjt = 1 kt1 = -0.11 ndif = -1 noif = 1 vsdfb = 0 vevb = 0.075 dvt1 = 0.53
```

Additional Information

In psoip, there are four optional parameters in the CDF properties of the p-cell:

- Temperature Node present (Tnode Out)
- Thermal Node (T)
- External Body contact (P)

- Body Node (B)

There can be a number of permutaions and combinations for these pins, however, only following seven permutaions are supported:

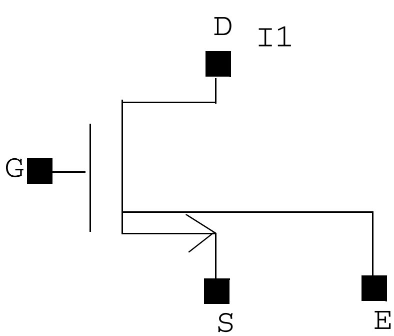
if Tnodeout is not selected:

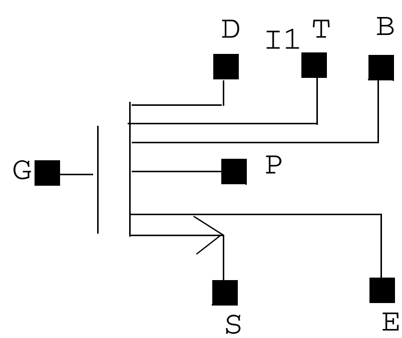
```
4 nodes: D G S E

5 nodes: D G S E P

6 nodes: D G S E P B

7 nodes: D G S E P B T
```

if Tnodeout is selected:

```
5 nodes: D G S E T

6 nodes: D G S E P T

7 nodes: D G S E P B T
```

Symbol: nsoip





N-type BSIM SOI model

nsoip is a n-type BSIM SOI model

Command-line help

```
spectre -h bsimsoi
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Temperature Node Present	`TnodeOut(earlier this was bn,therefore check)`	x	-	-	-	-
Thermal Node(T)	Tnode	x	-	-	-	-
Ext. Body Contact (PinP)	PinP	x	-	-	-	-
Body Node	BodyNodePin	x	-	-	-	-
Width	`w`	x	-	-	-	-
Length	`l`	x	-	-	-	-
Source diffusion area	`as`	x	-	-	-	-
Drain diffusion area	`ad`	x	-	-	-	-
Source diffusion periphery	`ps`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Drain diffusion periphery	pd	x	-	-	-	-
Drain diffusion res squares	nrd	x	-	-	-	-
Source diffusion res squares	nrs	x	-	-	-	-
Multiplier	m	x	-	-	-	-

Syntax/Synopsis

```
Name ( d g s e [p] [b] [t] ) ModelName <parameter=value> ...
```

Following is the model synopsis:

```
model ModelName bsimsoi <parameter=value> ...
```

Example

```
I6 (vdd! net9 0 0 ) bsimsoi w=1u 1=1u
```

Following is the sample model statement:

```
model nsoip_model bsimsoi type = n beta0 = 0 dvt = -0.032 delta = 0.01 k1 = 0.6
xbjt = 1 kt1 = -0.11 ndif = -1 noif = 1 vsdfb = 0 vevb = 0.075 dvt1 = 0.53
```

Additional Information

In nsoip, there are four optional parameters in the CDF properties of the n-cell:

- Temperature Node present (Tnode Out)
- Thermal Node (T)
- External Body contact (P)
- Body Node (B)

There can be a number of permutaions and combinations for these pins, however, only following seven permutaions are supported:

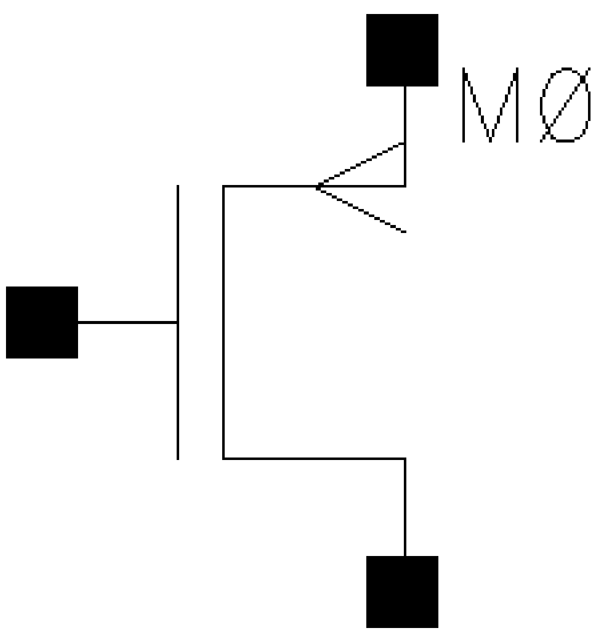
if Tnodeout=0 or not given (default is 0)

```
4 nodes: D G S E

5 nodes: D G S E P

6 nodes: D G S E P B

7 nodes: D G S E P B T
```

if Tnodeout=1

```
5 nodes: D G S E T

6 nodes: D G S E P T

7 nodes: D G S E P B T
```

Symbol: pmos



P-type Generic MOS Transistor (3 terminals)

Command-line help

For related information on MOS, use any of the following help commands:

```
spectre -h mos0
```

```
spectre -h mos1
```

```
spectre -h ekv
```

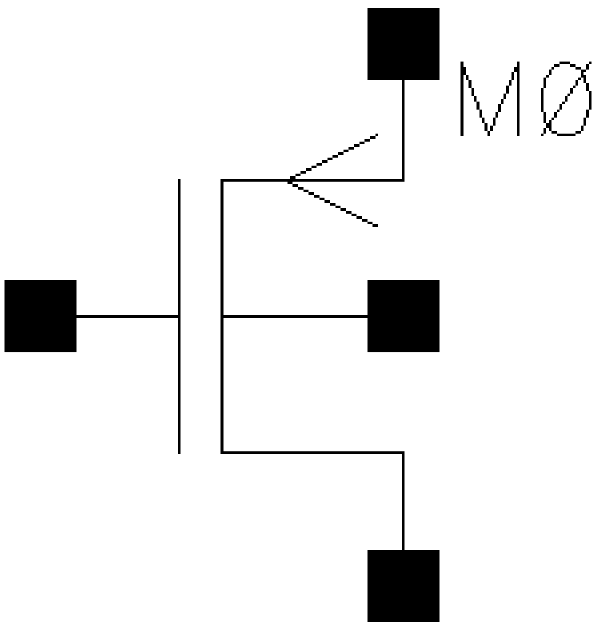
CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Bulk node connection	`bn`	-	-	-	-	-
Multiplier	`m`	x	x	x	x	-
Width	`w`	x	x	x	x	-
Length	`l`	x	x	x	x	-
Drain diffusion area	`ad`	x	-	-	x	-
Source diffusion area	`as`	x	-	-	x	-
Drain diffusion periphery	`pd`	x	-	-	x	-
Source diffusion periphery	`ps`	x	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Drain diffusion res squares	nrd	x	-	-	x	-
Source diffusion res squares	nrs	x	-	-	x	-
Drain diffusion length	ld	x	-	-	-	-
Source diffusion length	ls	x	-	-	-	-
Device initially off	off	-	-	-	x	-
Drain source initial voltage	Vds	-	-	-	x	-
Gate source initial voltage	Vgs	-	-	-	x	-
Bulk source initial voltage	Vbs	-	-	-	x	-
Temp rise from ambient	trise	x	-	-	-	-
Estimated operating region	region	x	-	-	-	-
Hot-electron degradation	degradation	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Additional drain resistance	`rdc`	x	-	-	x	-
Additional source resistance	`rsc`	x	-	-	x	-
Dist. OD & poly(one side)	`sa`	x	-	-	-	-
Dist. OD & poly(other side)	`sb`	x	-	-	-	-
Dist. betn neighbour fingers	`sd`	x	-	-	-	-
Temperature difference	`dtemp`	-	-	-	x	-
Source/ drain selector	`geo`	x	-	-	x	-

Symbol: pmos4



P-type Generic MOS Transistor (4 terminals)

Command-line help

For related information on MOS, use any of the following help commands:

`spectre -h mos0`

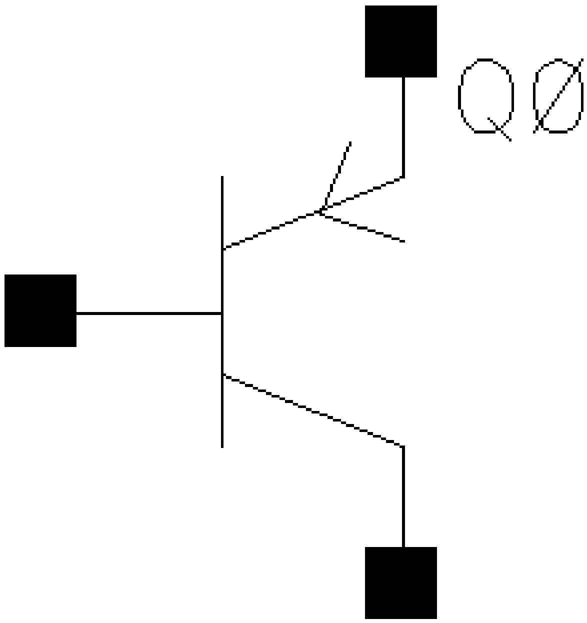
`spectre -h mos1`

`spectre -h ekv`

CDF Parameters

The CDF parameters for `pmos4` are the same as the CDF parameters for `pmos`.

Symbol: pnp



Generic Bipolar Transistor (PNP)

pnp is a p-type bjt.

Command-line help

```
spectre -h bjt
```

```
spectre -h bjt2
```

```
spectre -h bjt3
```

```
spectre -h bjt301
```

```
spectre -h vbic
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	x	-
Bulk node connection	`bn`	-	-	-	-	-
Device area	`area`	x	-	-	x	-
Multiplier	`m`	x	-	-	x	-
Temp rise from ambient	`trise`	x	-	-	-	-
Temp Rise Specifier	`triseSpec`	x	-	-	-	-
dtmp -Temp rise from ambient	`dtmp`	x	-	-	-	-
dtemp - Temp rise from ambient	`dtempn`	-	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-
Base-emitter voltage	`Vbe`	-	-	-	x	-
Collector-emitter voltage	`Vce`	-	-	-	x	-
Device initially off	`off`	-	-	-	x	-
Temperature difference	`dtemp`	x	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Base area	areab	-	-	-	x	-
Collector area	areac	-	-	-	x	-

Syntax/Synopsis

```
Name ( c b e [s] ) ModelName <parameter=value> ...
```

You do not have to specify the substrate terminal. If you do not specify it, the substrate is connected to ground.

Following is the model synopsis:

```
model ModelName bjt <parameter=value> ...
```

Example

```
q1 (vcc net3 minus) npn_mod region=fwd area=1 m=1
```

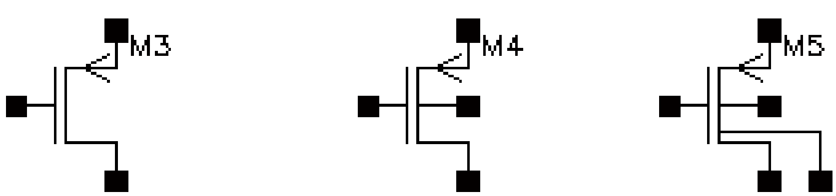
Following is a sample model statement:

```
model npn_mod bjt type=npn is=10e-13 bf=200 va=58.8 ikf=5.63e-3 rb=700
rbm=86 re=3.2 cje=0.352e-12 pe=0.76 me=0.34 tf=249e-12 cjc=0.34e-12
pc=0.55
```

Additional Information

This device is supported within the altergroups.

Symbol: ptft



P-Type Poly-Si TFT (PTFT)

ptft is a p-type polysilicon tft. It can have a maximum of five terminals with drain, gate and source being mandatory terminals and substrate and thermal being optional.

The diagrams show the terminal with none, one or two optional nodes selected.

Command-line help

```
spectre -h psitft
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-	-	-	-
Optional Nodes	`Opins`	x	-	-	-	-	-	-	-
Optional Bulk Node _B	`pinB`	x	-	-	-	-	-	-	-
Optional Thermal Node _T	`pinT`	x	-	-	-	-	-	-	-
Width	`w`	x	-	-	-	-	-	-	-
Length	`l`	x	-	-	-	-	-	-	-
Drain diffusion res squares	`nrd`	x	-	-	-	-	-	-	-
Source diffusion res squares	`nrs`	x	-	-	-	-	-	-	-
Multiplier	`m`	x	-	-	-	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Thermal resistance	`rth0`	x	-	-	-	-	-	-	-
Thermal capacitance	`cth0`	x	-	-	-	-	-	-	-
Num of segments	`nseg`	x	-	-	-	-	-	-	-

Syntax/Synopsis

```
Name ( d g s [b] [t] ) ModelName <parameter=value> ...
```

Model Synopsis

```
model ModelName psitft <parameter=value> ...
```

Sample Instance Statement

```
m4 (0 2 1 1 3) nch w=2u l=0.8u
```

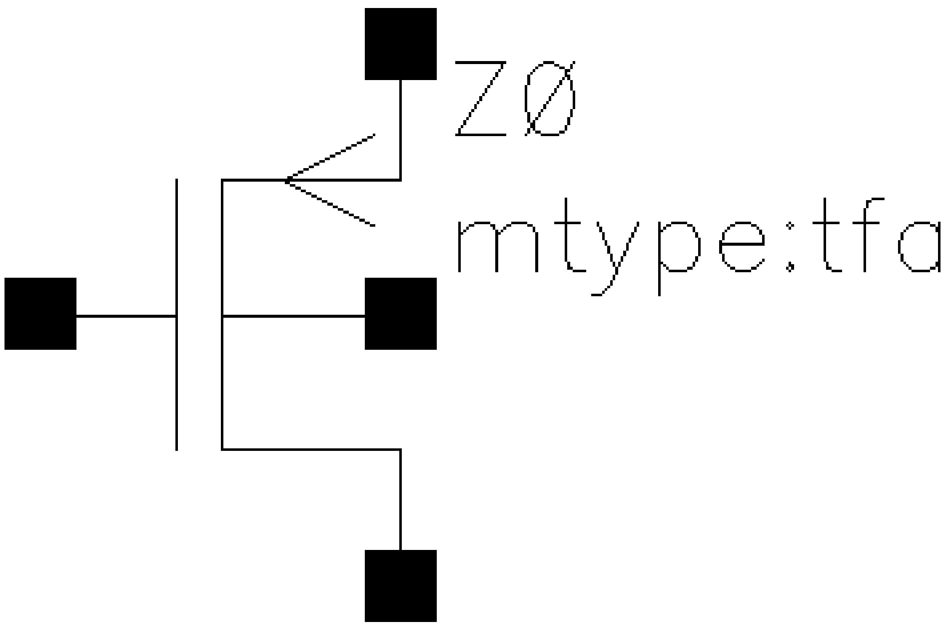
Sample Model Statement

```
model nch psitft type=p
```

Additional Information

This device is supported within the altergroups.

Symbol: psoi

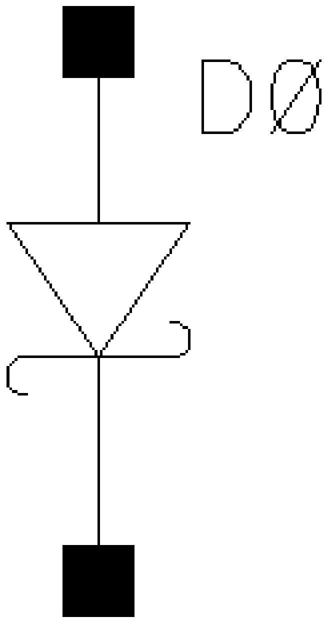


Independent Resistive Source

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Bulk node connection	bn	-	-	x	-	-
Multiplier	m	-	x	x	-	-
Width	w	-	x	x	-	-
Length	l	-	x	x	-	-

Symbol: schottky



Schottky Diode

A special type of diode that has a low forward-voltage drop leading to greater system efficiency.

Command-line help

```
spectre -h diode
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-

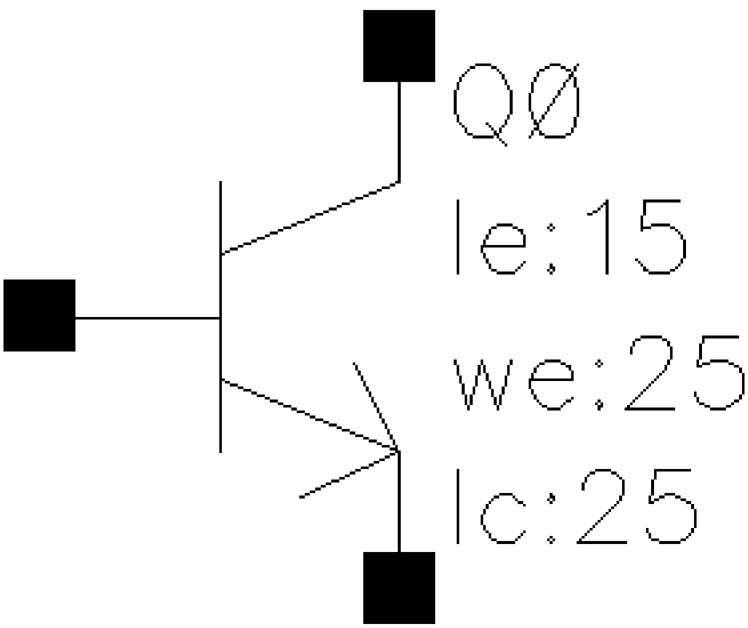
CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Device area	`area`	x	-	-	x	x
Device initially off	`off`	-	-	-	x	x
Initial diode voltage	`Vd`	-	-	-	x	x
Junction perimeter factor	`perim`	x	-	-	-	-
Length	`l`	x	-	-	x	x
Width	`w`	x	-	-	x	x
Multiplier	`m`	x	-	-	x	x
Scale factor	`scale`	x	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-
Periphery of junction	`pj`	-	-	-	x	x
Width of polysilicon	`wp`	-	-	-	x	x
Length of polysilicon	`lp`	-	-	-	x	x
Width of metal capacitator	`wm`	-	-	-	x	x
Length of metal capacitator	`lm`	-	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Temperature difference	`dtemp`	-	-	-	x	x

Example

```
D0 (net1 net2) schottky
```

Symbol: usernpn



User Specific NPN Bipolar Transistor (3 terminals)

Command-line help

```
spectre -h bjt
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	x	-
Bulk node connection	`bn`	-	-	-	-	-
Device area	`area`	x	-	-	x	-
Base-emitter voltage	`Vbe`	-	-	-	x	-
Collector-emitter voltage	`Vce`	-	-	-	x	-
Device initially off	`off`	-	-	-	x	-
Emitter length	`le`	x	-	-	-	-
Emitter width	`we`	x	-	-	-	-
Collector length	`lc`	x	-	-	-	-
Temp Rise Specifier	`triseSpec`	x	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-
dtmp -Temp rise from ambient	`dtmp`	x	-	-	-	-
dtemp - Temp rise from ambient	`dtemp`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Estimated operating region	`region`	x	-	-	-	-

Syntax/Synopsis

```
Name ( c b e [s] ) ModelName <parameter=value> ...
```

You do not have to specify the substrate terminal. If you do not specify it, the substrate is connected to ground.

Model Synopsis:

```
model ModelName bjt <parameter=value> ...
```

Example

Sample Instance Statement:

```
q1 (vcc net3 minus) npn_mod region=fwd area=1 m=1
```

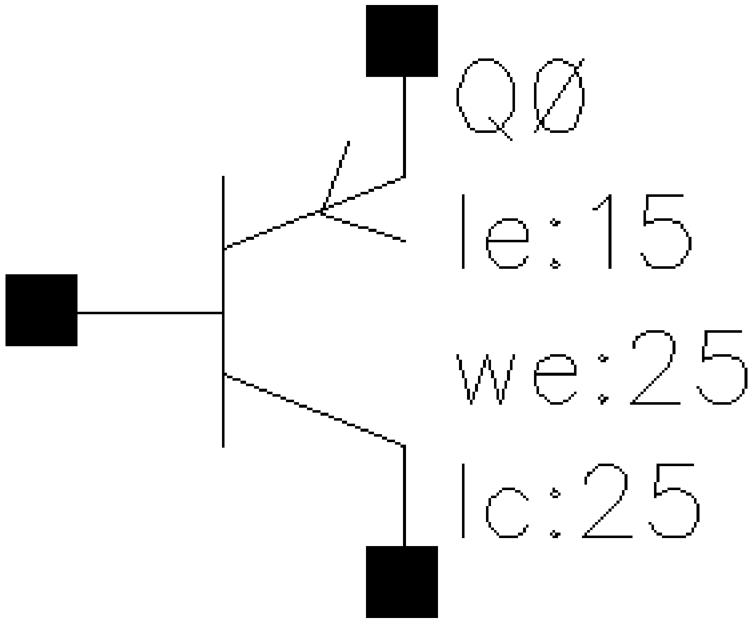
Sample Model Statement:

```
model npn_mod bjt type=npn is=10e-13 bf=200 va=58.8 ikf=5.63e-3 rb=700 rbm=86
re=3.2 cje=0.352e-12 pe=0.76 me=0.34 tf=249e-12 cjc=0.34e-12 pc=0.55
```

Additional Information

This device is supported within the altergroups.

Symbol: userpnp



User Specific PNP Bipolar Transistor (3 terminals)

Command-line help

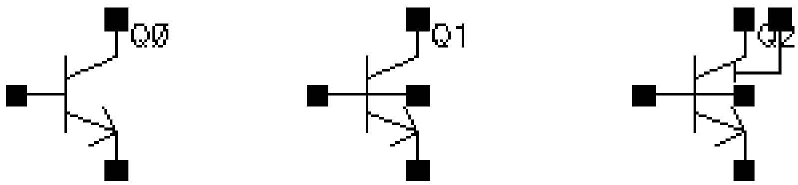
```
spectre -h bjt
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Bulk node connection	`bn`	-	-	-	-	-
Device area	`area`	x	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Base-emitter voltage	`Vbe`	-	-	-	x	x
Collector-emitter voltage	`Vce`	-	-	-	x	x
Device initially off	`off`	-	-	-	x	x
Emitter length	`le`	x	-	-	-	x
Emitter width	`we`	x	-	-	-	x
Collector length	`lc`	x	-	-	-	x
Temp Rise Specifier	`triseSpec`	x	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-
dtmp -Temp rise from ambient	`dtmp`	x	-	-	-	-
dtemp - Temp rise from ambient	`dtemp`	x	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-

Symbol: vnpn



Variable Bipolar Transistor (VNPN)

vnpn is a variable terminal n-type bjt. It can have a maximum of five terminals with collector, emitter and base being mandatory terminals and substrate and thermal being optional.

The diagrams show the terminal with none, one or two optional nodes selected.

Command-line help

```
spectre -h bjt

spectre -h bjt2

spectre -h bjt3

spectre -h bjt301

spectre -h vbic
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-	-	-	-
Optional Nodes	`Opins`	x							

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Optional Substrate Node _S	`pinS`	x	-	-	-	-	-	-	-
Optional Thermal Node _T	`pint`	x	-	-	-	-	-	-	-
Optional Thermal Node _dT	`pindt`	x	-	-	-	-	-	-	-
Temp Rise Specifier	`triseSpec`	x	-	-	-	-	-	-	-
Device area	`area`	x	-	-	-	-	-	-	-
Multiplier	`m`	x	-	-	-	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-	-	-	-
dtmp -Temp rise from ambient	`dtmp`	x	-	-	-	-	-	-	-
dtemp - Temp rise from ambient	`dtempn`	x	-	-	-	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-	-	-	-
Self Heating Switch	`self_heating`	x	-	-	-	-	-	-	-
Length of Emitter Window	`le0`	x	-	-	-	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Width of Emitter Window	`be0`	x	-	-	-	-	-	-	-
Number of emitter contacts	`ne`	x	-	-	-	-	-	-	-
Number of base contacts	`nb`	x	-	-	-	-	-	-	-
Location of collector contact	`location`	x	-	-	-	-	-	-	-
Number of collector contacts	`ncbjt`	x	-	-	-	-	-	-	-
Contact configuration	`order`	x	-	-	-	-	-	-	-
Number of structures in parallel	`npas`	x	-	-	-	-	-	-	-

Syntax/Synopsis

```
Name ( c b e [s] ) ModelName <parameter=value> ...
```

You do not have to specify the substrate terminal. If you do not specify it, the substrate is connected to ground.

Model Synopsis

```
model ModelName bjt <parameter=value> ...
```

Example

```
q1 (vcc net3 minus) npn_mod region=fwd area=1 m=1
```

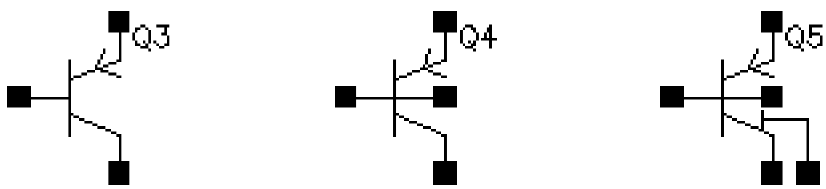
Following is a sample model statement:

```
model npn_mod bjt type=npn is=10e-13 bf=200 va=58.8 ikf=5.63e-3 rb=700 rbm=86
re=3.2 cje=0.352e-12 pe=0.76 me=0.34 tf=249e-12 cjc=0.34e-12 pc=0.55
```

Additional Information

This device is supported within the altergroups.

Symbol: vpnp



Variable Bipolar Transistor (VPNP)

vpnp is a variable terminal p-type bjt. It can have a maximum of five terminals with collector, emitter and base being mandatory terminals and substrate and thermal being optional.

The diagrams show the terminal with none, one or two optional nodes selected.

Command-line help

```
spectre -h bjt
```

```
spectre -h bjt2
```

```
spectre -h bjt3
```

```
spectre -h bjt301
```

```
spectre -h vbic
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-	-	-	-
Optional Nodes	`Opins`	x							
Optional Substrate Node _S	`pinS`	x	-	-	-	-	-	-	-
Optional Thermal Node _T	`pint`	x	-	-	-	-	-	-	-
Optional Thermal Node _dT	`pindt`	x	-	-	-	-	-	-	-
Temp Rise Specifier	`triseSpec`	x	-	-	-	-	-	-	-
Device area	`area`	x	-	-	-	-	-	-	-
Multiplier	`m`	x	-	-	-	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-	-	-	-
dtmp -Temp rise from ambient	`dtmp`	x	-	-	-	-	-	-	-
dtemp - Temp rise from ambient	`dtempn`	x	-	-	-	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Self Heating Switch	`self_heating`	x	-	-	-	-	-	-	-
Length of Emitter Window	`le0`	x	-	-	-	-	-	-	-
Width of Emitter Window	`be0`	x	-	-	-	-	-	-	-
Number of emitter contacts	`ne`	x	-	-	-	-	-	-	-
Number of base contacts	`nb`	x	-	-	-	-	-	-	-
Location of collector contact	`location`	x	-	-	-	-	-	-	-
Number of collector contacts	`ncbjt`	x	-	-	-	-	-	-	-
Contact configuration	`order`	x	-	-	-	-	-	-	-
Number of structures in parallel	`npas`	x	-	-	-	-	-	-	-

Syntax/Synopsis

```
Name ( c b e [s] ) ModelName <parameter=value> ...
```

You do not have to specify the substrate terminal. If you do not specify it, the substrate is connected to ground.

Following is the model synopsis:

```
model ModelName bjt <parameter=value> ...
```

Example

```
q1 (vcc net3 minus) npn_mod region=fwd area=1 m=1
```

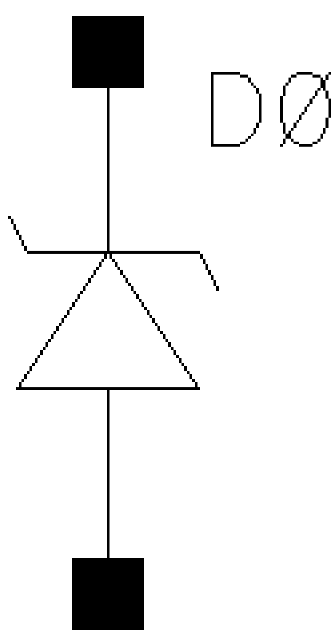
Following is a sample model statement:

```
model npn_mod bjt type=npn is=10e-13 bf=200 va=58.8 ikf=5.63e-3 rb=700
rbm=86 re=3.2 cje=0.352e-12 pe=0.76 me=0.34 tf=249e-12 cjc=0.34e-12
pc=0.55
```

Additional Information

This device is supported within the altergroups.

Symbol: zener



Zener Diode

It has p-n junction in reverse bias to use the zener effect to maintain a constant voltage.

Command-line help

```
spectre -h diode
```

CDF Parameters

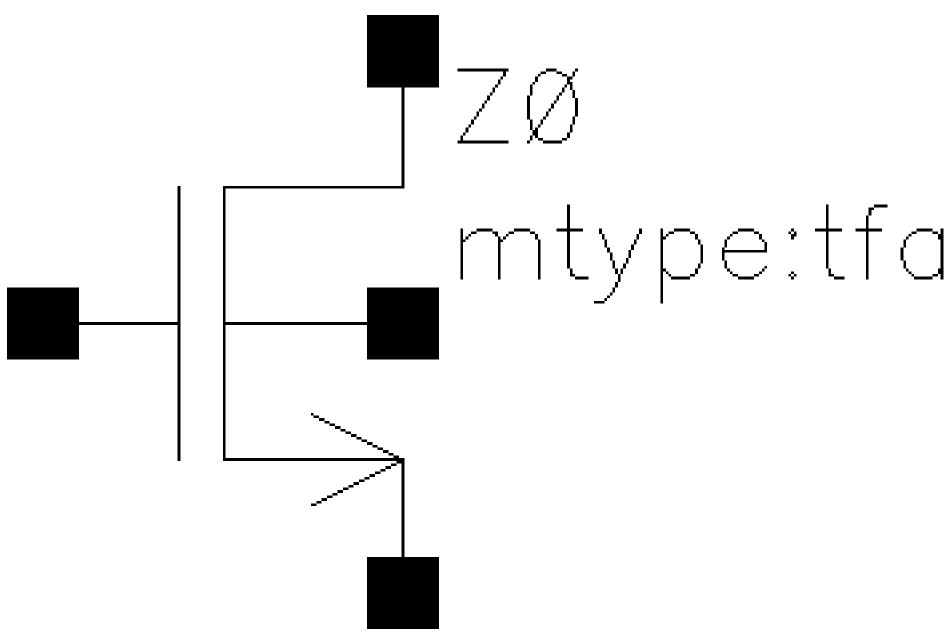
CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	x	x
Device area	`area`	x	-	-	x	x
Device initially off	`off`	-	-	-	x	x
Initial diode voltage	`Vd`	-	-	-	x	x
Junction perimeter factor	`perim`	x	-	-	-	-
Length	`l`	x	-	-	x	x
Width	`w`	x	-	-	x	x
Multiplier	`m`	x	-	-	x	x
Scale factor	`scale`	x	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-
Periphery of junction	`pj`	-	-	-	x	x
Width of polysilicon	`wp`	-	-	-	x	x
Length of polysilicon	`lp`	-	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Width of metal capacitator	`wm`	-	-	-	x	x
Length of metal capacitator	`lm`	-	-	-	x	x
Temperature difference	`dtemp`	-	-	-	x	x

auCdl and auLvs Components

The following components are supported only by auCdl or auLvs.

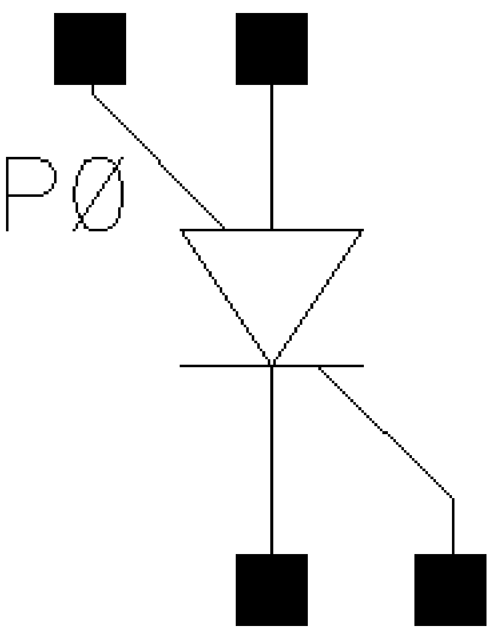
Symbol: nsoi



CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Bulk node connection	`bn`	-	-	x	-	-
Multiplier	`m`	-	x	x	-	-
Width	`w`	-	x	x	-	-
Length	`l`	-	x	x	-	-

Symbol: scr



Silicon Controlled Rectifier

scr is a conventional rectifier controlled by a gate signal. Although the main circuit is a rectifier, the application of a forward voltage is not enough for conduction. Therefore, a gate signal controls the rectifier conduction.

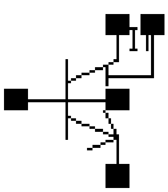
CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Bulk node connection	bn	-	-	x	-	-

Example

```
P231   32   5   21   8   7   PSCR
PN01   25   14   18   2   PMOD   IC=-.8 .8 -15
```

Symbol: bjt504tnpn



Compact Bipolar-Transistor Model

The bjt504 model provides a detailed description of a vertical integrated NPN transistor.

Command-line help

```
spectre -h bjt502
```

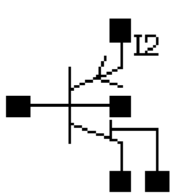
CDF Parameters

CDF Parameter Label	CDF Parameter	spec tre	spect reS	cdsSp ice	auC dl	auL vs	hspic eS	hspi ceD	Ultra Sim
Model name	`model`	x	-	-	-	-	-	-	-
Number of devices in parallel	`mult`	x	-	-	-	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-	-	-	-
Multiplicit y factor	`m`	x	-	-	-	-	-	-	-
Alias of mult	`area`	x	-	-	-	-	-	-	-

Syntax/Synopsis

```
Name ( c b e s ) ModelName <parameter=value> ...
```

Symbol: bjt504tpnp



Compact Bipolar-Transistor Model

The bjt504 model provides a detailed description of a vertical integrated PNP transistor.

Command-line help

```
spectre -h bjt502
```

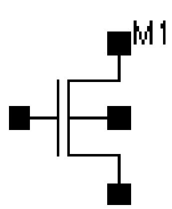
CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-	-	-	-
Number of devices in parallel	`mult`	x	-	-	-	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-	-	-	-
Multiplicity factor	`m`	x	-	-	-	-	-	-	-
Alias of mult	`area`	x	-	-	-	-	-	-	-

Syntax/Synopsis

```
Name ( c b e s ) ModelName <parameter=value> ...
```

Symbol: bsim4



BSIM Effective MOS Transistor

This component is a simple BSIM MOS transistor.

Command-line help

```
spectre -h bsim4
```

CDF Parameters

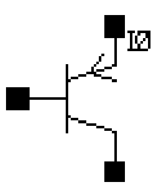
CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-	-	-	-
Integral-1st distribution func	`sca`	x	-	-	-	-	-	-	-
Integral-2nd distribution func	`scb`	x	-	-	-	-	-	-	-
Integral-3rd distribution func	`scc`	x	-	-	-	-	-	-	-
Distance to a single well edge	`sc`	x	-	-	-	-	-	-	-
shift in 0-bias threshold vth0	`delvo`	x	-	-	-	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Gate contact-channel edge	`xgw`	x	-	-	-	-	-	-	-
Number of gate contacts	`ngcon`	x	-	-	-	-	-	-	-

Syntax/Synopsis

```
Name ( d g s b ) ModelName <parameter=value> ...
```

Symbol: vbic



VBIC Bipolar Transistor

This component is a bipolar transistor.

Command-line help

```
spectre -h vbic
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-	-	-	-
Optional Node configuration	`vbicOpNodes`	x	-	-	-	-	-	-	-
Device area	`area`	x	-	-	-	-	-	-	-
Multiplier	`m`	x	-	-	-	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-	-	-	-
Temperature difference	`dtemp`	x	-	-	-	-	-	-	-

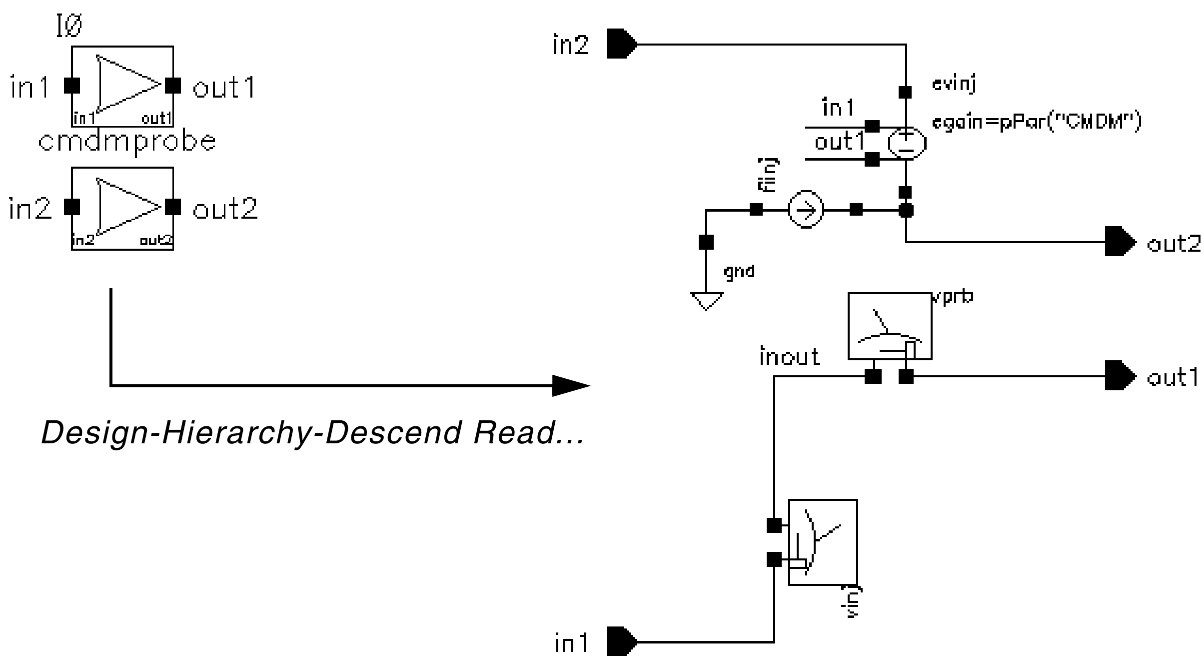
Syntax/Synopsis

```
Name ( c b e [s] [dt] [tl] ) ModelName <parameter=value> ...
```

3

Analysis Specific Components

Symbol: cmdmprobe



Common Model Differential Model Probe

This is a Spectre subcircuit component used in Spectre stability analysis for measuring differential stability. It measures the common-mode stability when CMDM is set to 1 and differential-mode stability when CMDM is set to -1.

The subcircuit consists of two iprobes and two controlled sources that can be viewed by:

➤ Selecting *Design->Hierarchy->Descend Read* or *Descend Edit* from the Virtuoso Schematic Reading window.

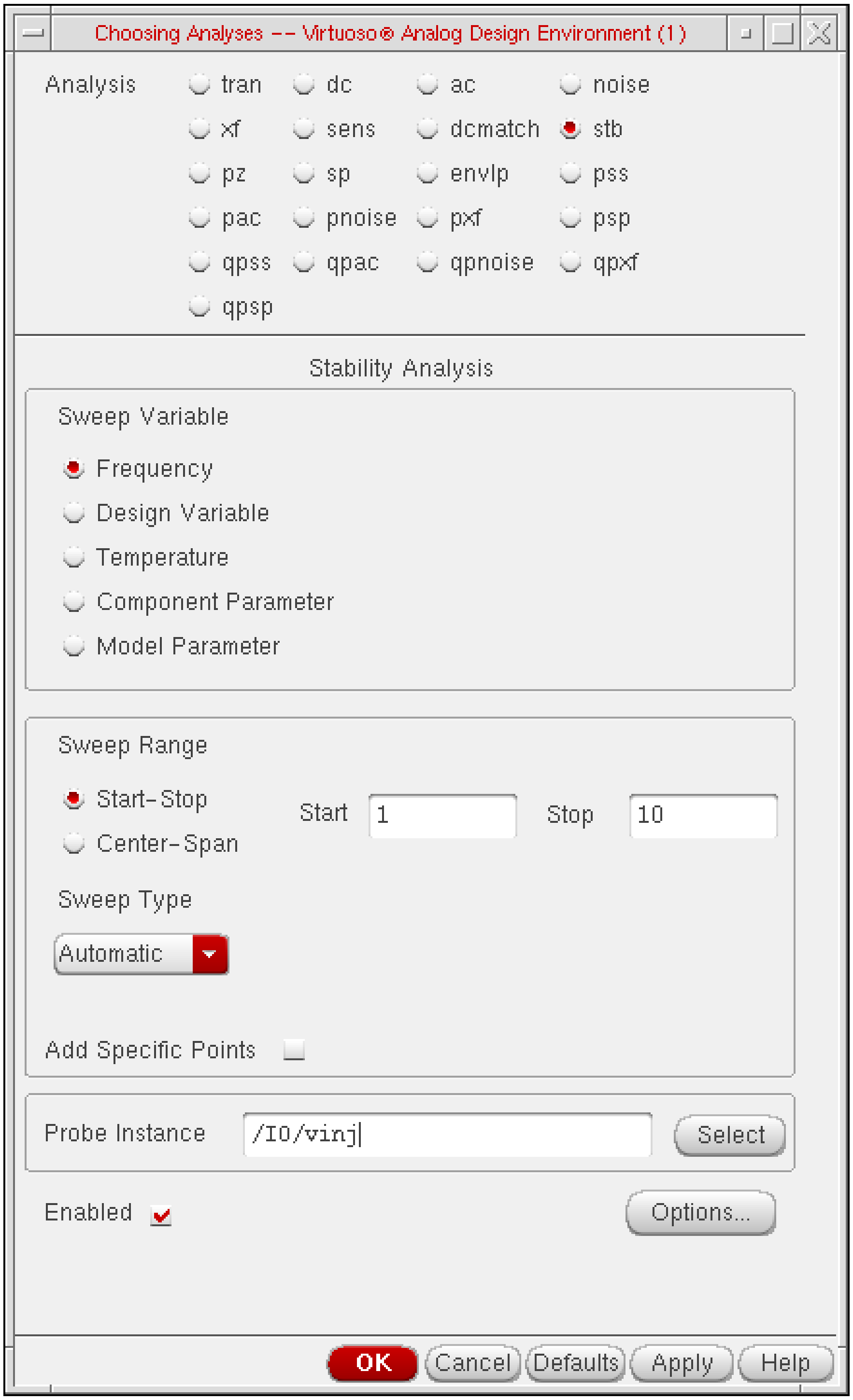
The above illustration shows the cmdmprobe subcircuit.

The subcircuit has two probes, `Vinj` and `Iinj`, designated for stability analysis. Their values are set to zero in a normal circuit simulation. To perform a stability analysis, specify the `vinj` probe in the Probe Instance field of the Choosing Analysis form. The process and the calculation is automated in Spectre's stability (stb) analysis. `Iinj` is internally placed to simplify the use model.

The steps to perform stability analysis in Spectre are as follows:

1. Add the cmdmprobe instance to the design.
2. Select the *Tools->Analog Environment* menu option.
3. Select *Analyses->Choose...* to display the Choosing Analyses form.
4. Select the *stb* radio button.
5. Specify the *Start* Sweep Range and the *Stop* Sweep Range.
6. Click *Select* and then select the cmdmprobe instance from the design.

The <Inst_id>/vinj automatically appears in the Probe Instance field.



7. Click *OK*.

8. Select the *Simulation->Netlist and Run* menu option to generate the netlist in Spectre Direct.

Example

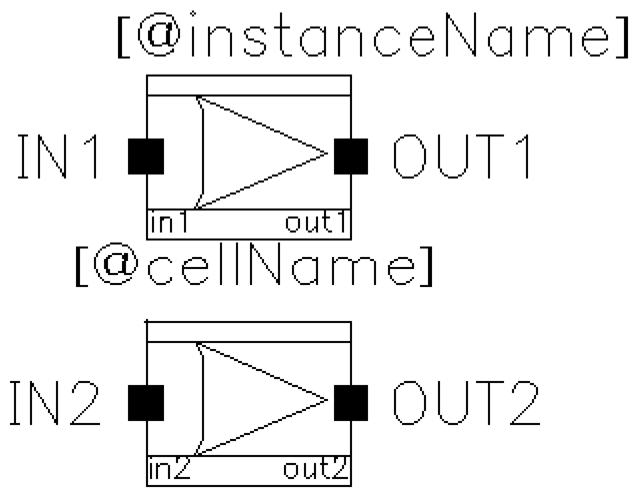
For the instance I107 and Sweep Range between 1 and 10, the netlist is as follows:

```
stb stb start=1 stop=10 probe=I107.vinj annotate=status
// Library name: analogLib
// Cell name: cmdmprobe
// View name: schematic
subckt cmdmprobe in1 in2 out1 out2
parameters CMDM=1
     evinj (in2 out2 in1 out1) vcvs gain=CMDM
     vprb (inout out1) iprobe
     vinj (inout in1) iprobe
     fiinj (0 out2) pcccs gain=CMDM probes=[ vprb vinj ] coeffs=[ 0 1 1 ]
ends cmdmprobe
// End of subcircuit definition.

// Library name: testLib
// Cell name: testCell
// View name: schematic
I107 (net048 net047 net046 net045) cmdmprobe CMDM=1
I111 (net080 net079 net078 net077) cmdmprobe CMDM=-1
```

Component Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
CMDM	CMDM	x	-	-	-	-

Symbol: diffstbprobe



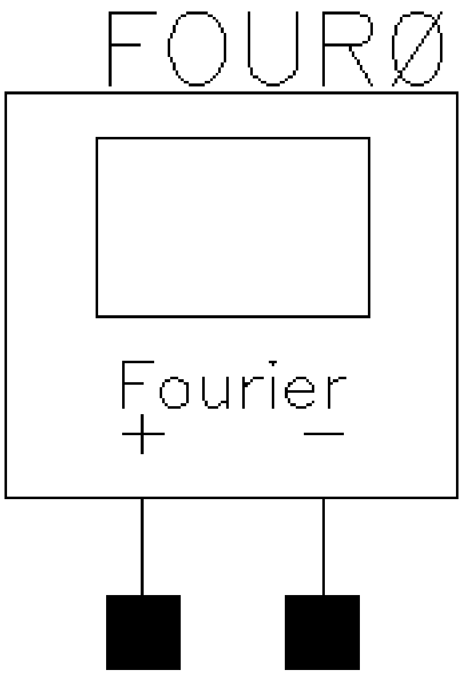
Differential Stability Probe

This is a Spectre subcircuit component used in Spectre stability analysis for measuring differential stability for multi-loop circuits, such as differential feedback circuit.

Component Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
CMDM	`CMDM`	x	-	-	-	-
lxRemoveDevice	`lxRemoveDevice`	-	x	-	-	-

Symbol: fourier



Ratiometric Fourier Analyzer

It measures the Fourier coefficients of two different signals at a specified fundamental frequency without loading the circuit.

Command-line help

```
spectre -h fourier
```

Component Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Fundamental frequency	`fund`	x	-	-	-	-
Minimum no. of time points	`points`	x	-	-	-	-
Active	`active`	x	-	-	-	-
Order of interpolation	`order`	x	-	-	-	-
Number of harmonics	`harms`	x	-	-	-	-
No. of reference Harmonics	`refharms`	x	-	-	-	-
Scale factor	`scale`	x	-	-	-	-
No. of desired harmonics	`harmsvec`	x	-	-	-	-
No. of reference harmonics	`refharmsvec`	x	-	-	-	-
First harmonics computed	`firstharm`	x	-	-	-	-
First of reference harmonics	`reffirstharm`	x	-	-	-	-
Nomalizing harmonic	`normharm`	x	-	-	-	-
Norm of reference harmonics	`refnormharm`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
To print fourier results on	`where`	x	-	-	-	-

Syntax/Synopsis

```
Name ( [p] [n] [pr] [nr] ) ModelName <parameter=value> ...
Name ( [p] [n] [pr] [nr] ) fourier <parameter=value> ...
```

The signal between terminals ‘p’ and ‘n’ is the test or numerator signal. The signal between terminals ‘pr’ and ‘nr’ is the reference or denominator signal. Fourier analysis is performed on terminal currents by specifying the ‘term’ or ‘refterm’ parameters. If both ‘term’ and ‘p’ or ‘n’ are specified, then the terminal current becomes the numerator and the node voltages become the denominator. By mixing voltages and currents, it is possible to compute large signal immittances.

Following is the model synopsis:

```
model ModelName fourier <parameter=value> ...
```

Example

```
four1 (1 0) fourmod harms=50
```

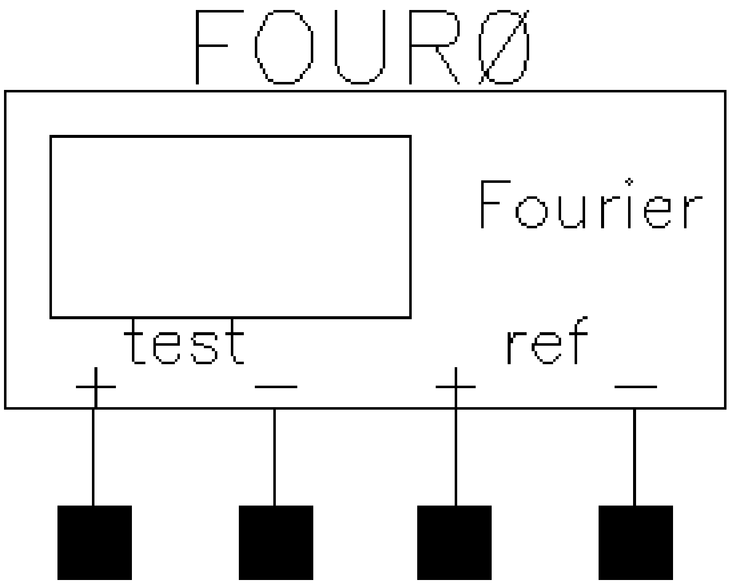
Following is the sample model statement:

```
model fourmod fourier fund=900M points=2500 order=2
```

Additional Information

This device is not supported within the altergroups.

Symbol: fourier2ch



Ratiometric Fourier Analyzer With Reference Terminals

Command-line help

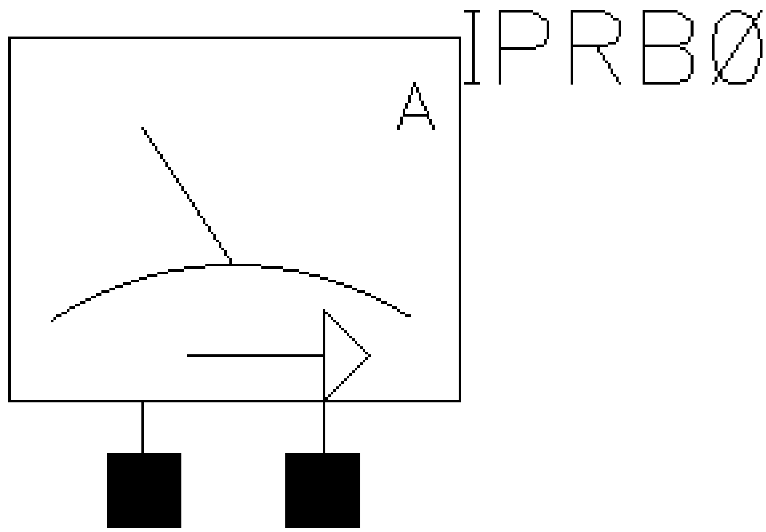
```
spectre -h fourier
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Fundamental frequency	`fund`	x	-	-	-	-
Minimum no. of time points	`points`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Active	`active`	x	-	-	-	-
Order of interpolation	`order`	x	-	-	-	-
Number of harmonics	`harms`	x	-	-	-	-
No. of reference Harmonics	`refharms`	x	-	-	-	-
Scale factor	`scale`	x	-	-	-	-
No. of desired harmonics	`harmsvec`	x	-	-	-	-
No. of reference harmonics	`refharmsvec`	x	-	-	-	-
First harmonics computed	`firstharm`	x	-	-	-	-
First of reference harmonics	`reffirstharm`	x	-	-	-	-
Nomalizing harmonic	`normharm`	x	-	-	-	-
Norm of reference harmonics	`refnormharm`	x	-	-	-	-
To print fourier results on	`where`	x	-	-	-	-

Symbol: iprobe



Current Probe

Current through the probe is computed and is defined as positive if it flows from the input node, through the probe, to the output node. Since the current variable gets the name of the 'iprobe' instance, you cannot create an 'iprobe' with the same name as a circuit node.

Command-line help

```
spectre -h iprobe
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Fundamental frequency	`fund`	x	-	-	-	-
Minimum no. of time points	`points`	x	-	-	-	-
Active	`active`	x	-	-	-	-
Order of interpolation	`order`	x	-	-	-	-
Number of harmonics	`harms`	x	-	-	-	-
No. of reference Harmonics	`refharms`	x	-	-	-	-
Scale factor	`scale`	x	-	-	-	-
Dummy DC voltage	`vdummy`	-	-	-	-	x
lxRemoveDevice	`lxRemoveDevice`	-	x	-	-	-

Syntax/Synopsis

```
Name ( in out ) iprobe
```

Example

```
ip (1  0) iprobe
```

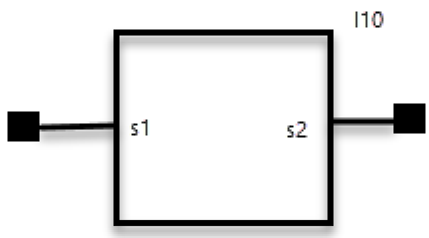
Additional Information

This device is not supported within the altergroups.

Symbol: sprobe

sprobe

A special testbench that enables in-situ probing of bi-directional impedances, without breaking the circuit.



Command-line help

```
spectre -h sprobe
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Reference Resistance	`r0`	x	-	-	-	-
Series Resistance	`rs`	x	-	-	-	-
Probe feed Resistance	`rfeed`	x	-	-	-	-

Syntax/Synopsis

```
Name ( in out ) sprobe
```

Example

sprobe Instance

```
I1 (net7 net4) sprobe rs=0.01 rfeed=1e8 r0=50
```

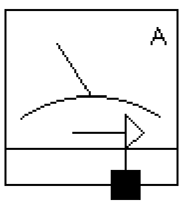
sprobe analysis in one sp analysis

```
sp sp sprobes=[I1 I0] …
```

sprobe and normal sp analysis in one sp analysis.

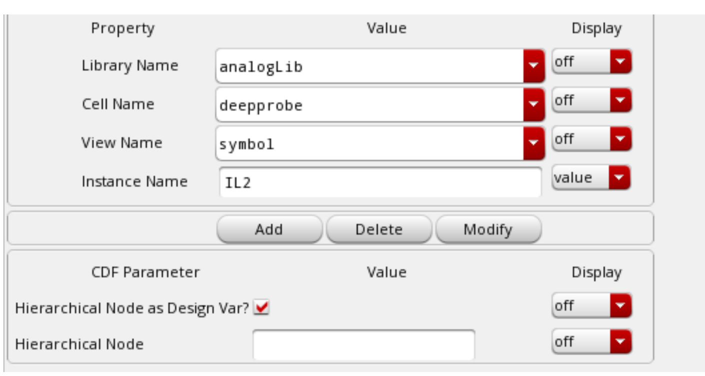
```
sp sp ports=[PORT0] sprobes=[I1 I0]
```

Symbol: deepprobe



It is a single pin device connected to an internal hierarchy net that lets you probe down through the design hierarchy. You can make a connection from the top-level testbench to an internal net within a sub-block inside the hierarchy by connecting a named wire to a deepprobe terminal. With this component, you can also short internal nets, connect two internal nets, or inject pulses on any internal net in the design.

If you select the *Hierarchical Node as Design Var?* check box in the Edit Object Properties form, the value in the *Hierarchical Node* field is considered a variable and netlisted accordingly. Otherwise, the value for *Hierarchical Node* is considered a string value and netlisted directly.



Command-line help

```
spectre -h iprobe
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Hierarchical Node	probeNode	x	-	-	x	-	-	-	-

Syntax/Synopsis

```
Name ( in out ) iprobe
```

Example

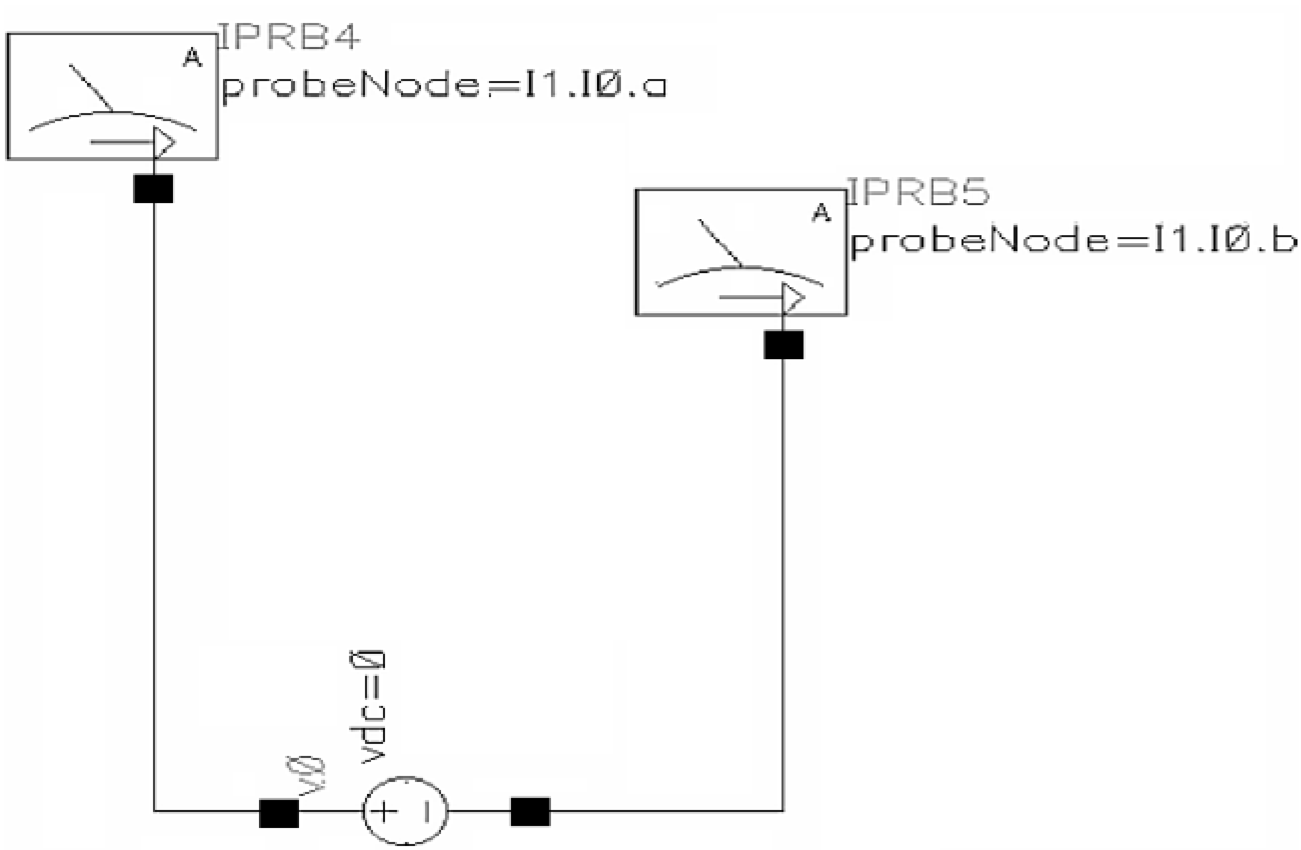
The following example shows the netlist syntax of a deepprobe element.

```
IPRB0 (I1.b c) iprobe
```

Here I1.b is the name of the hierarchical net and c is the name of the net connected to the deepprobe element.

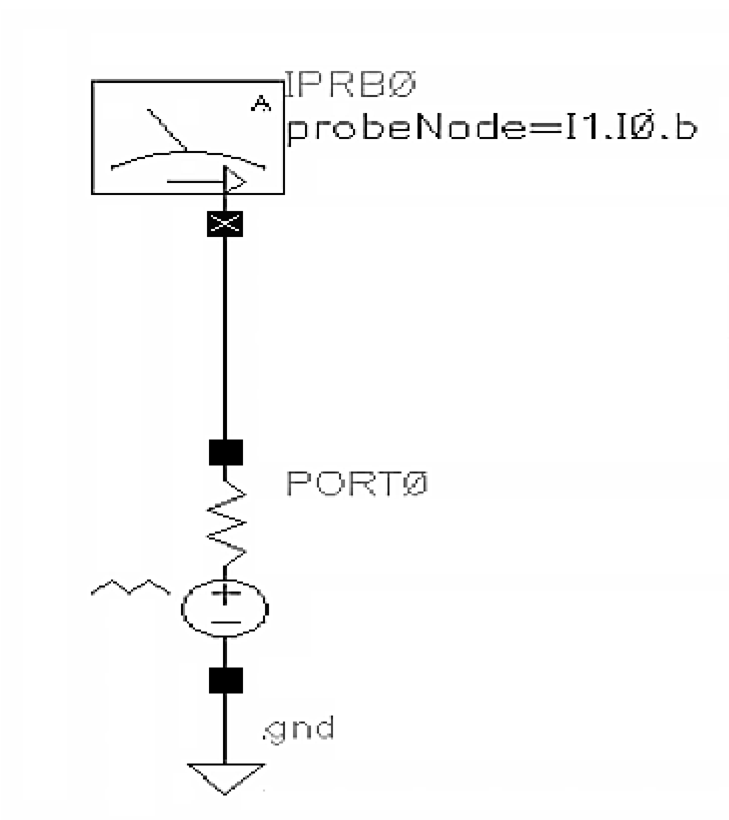
The net name must be the same as it appears in the netlist. For example a member of a bussed net (bus<5>) may appear in the netlist as bus\<5\>. So if that is within the I1 instance at the top level, you should enter I1.bus\<5\>.

The following example shows two deepprobe elements being used to short two internal nets.



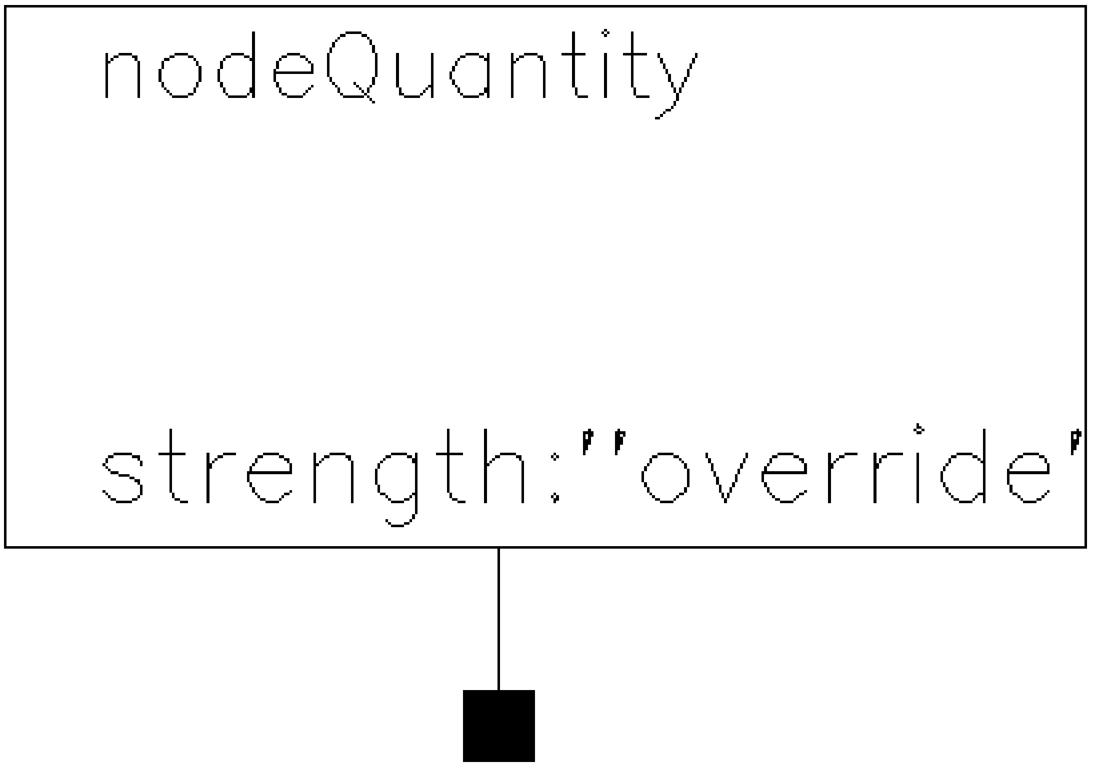

```
IPRB5 (I1.I0.b net7) iprobe
IPRB4 (I1.I0.a net6) iprobe
v0 (net6 net7) vsource dc=0 type=dc
```

The following example shows a deepprobe element being used to inject pulses on an internal net.




```
IPRB0 (I1.I0.b net8) iprobe
PORT0 (net8 0) port r=50 type=pwl wave=[ 0 0 5n 0 ]
```

Symbol: nodeQuantity



Set Node Quantities

Quantities contain information about specific types of signals, such as their units, absolute tolerances, and maximum allowed change per Newton iteration. Use the ‘quantity’ statement to create new quantities or to redefine properties of an existing quantity. Use this statement to set the quantities for a particular node.

For example, to indicate that the node ‘net1’ is used for thermal signals, you could use the following node statement.

```
i17 (net1) node value=Temp flow=Pwr
```

‘Temp’ and ‘Pwr’ are predefined quantities.

Command-line help

```
spectre -h node
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Flow	`flow`	x	-	-	-	x
Value	`value`	x	-	-	-	x
Strength	`strength`	x	-	-	-	x

Syntax/Synopsis

```
Name ( 1 [2] ... ) node <parameter=value> ...
```

Example

```
node1 (1 2 3) node value="T" flow="W" strength=override //Must define T and W with
quantity statement.
```

Additional Information

This device is not supported within the altergroups.

Symbol: simulinkCoupler

simulinkCoupler is a Pcell. In order to cosimulate between AMS and simulink, two types of couplers are required. The Pcell is the coupler which is used on AMS side. On simulink side, simulinkCoupler will be used. The couplers communicate to each other through a TCP/IP network socket connection.

CDF Parameters

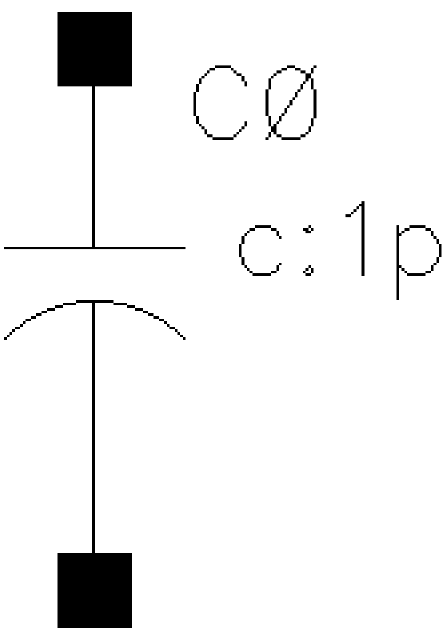
CDF Parameter Label	CDF Parameter
Coupler domain	`a_or_d`
Number of input pins	`n_inp`
Number of output pins	`n_outp`
Show advanced options	`advUser`
Initial coupler output voltage	`init_val`
Simulink(R) hostname	`hostname`
Socket port	`sockPort`
Sim response timeout	`sockTimeout`

4

Parasitic Components

You can use the parasitic components of anlogLib to account for the effect of parasitics on analog circuits. By accounting for the effect of parasitics, you can improve the accuracy of your circuit simulations. These components are usually used during Diva extraction and are placed in an extracted view. Although these components are similar to the normal components, they appear only in extracted views.

Symbol: pcapacitor



Parasitic Capacitor

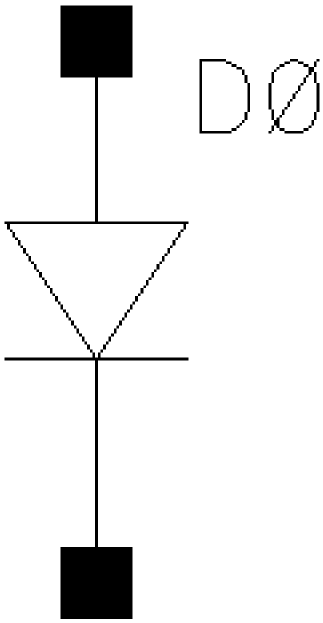
Command-line help

```
spectre -h capacitor
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Capacitance	c	x	x	x	x	x
Initial condition	ic	x	-	-	x	x
Model name	model	x	-	-	-	-
Width	w	x	-	-	x	x
Length	l	x	-	-	x	x
Multiplier	m	x	-	-	x	x
Scale factor	scale	x	-	-	x	x
Temp rise from ambient	trise	x	-	-	-	-
Temperature coefficient 1	tc1	x	-	-	x	x
Temperature coefficient 2	tc2	x	-	-	x	x
Number of Polynomial Coeffs	polyCoef	-	-	-	x	x
Temperature difference	dtemp	-	-	-	x	x

Symbol: pdiode



Parasitic Diode

Command-line help

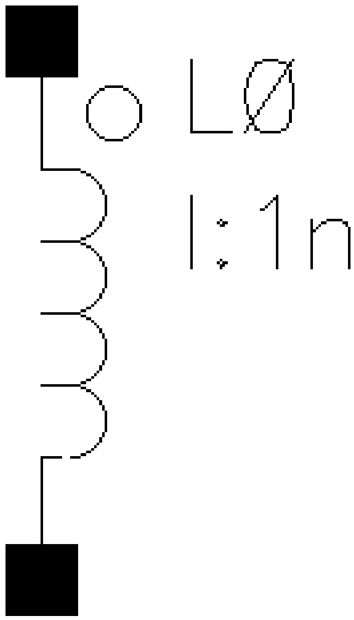
```
spectre -h diode
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Device area	`area`	x	-	-	x	x
Device initially off	`off`	-	-	-	x	x
Initial diode voltage	`Vd`	-	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Junction perimeter factor	`perim`	x	-	-	-	-
Length	`l`	x	-	-	x	x
Width	`w`	x	-	-	x	x
Multiplier	`m`	x	-	-	x	x
Scale factor	`scale`	x	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-
Estimated operating region	`region`	x	-	-	-	-
Periphery of junction	`pj`	-	-	-	x	x
Width of polysilicon	`wp`	-	-	-	x	x
Length of polysilicon	`lp`	-	-	-	x	x
Width of metal capcitor	`wm`	-	-	-	x	x
Length of metal capcitor	`lm`	-	-	-	x	x
Temperature difference	`dtemp`	-	-	-	x	x

Symbol: pinductor



Parasitic Inductor

Command-line help

`spectre -h inductor`

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Inductance	`l`	x	x	x	x	x
Initial condition	`ic`	x	-	-	x	-
Model name	`model`	x	-	-	-	-
Resistance	`r`	x	-	-	x	x
Multiplier	`m`	x	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Temp rise from ambient	trise	x	-	-	-	-
Scale factor	scale	-	-	-	x	x
Number of Polynomial Coeffs	polyCoef	-	-	-	x	-
Temperature coefficient 1	tc1	-	-	-	x	x
Temperature coefficient 2	tc2	-	-	-	x	x
Temperature difference	dtemp	-	-	-	x	x

Syntax/Synopsis

```
Name ( 1 2 ) ModelName <parameter=value> ...
Name ( 1 2 ) inductor <parameter=value> ...
```

Following is the model synopsis:

```
model ModelName inductor <parameter=value> ...
```

Example

Following is a sample instance statement without model:

```
l33 (0 net29) inductor l=10e-9 r=1 m=1
```

Following is a sample instance statement with model:

```
l33 (0 net29) ind l=10e-9 r=1 m=1
```

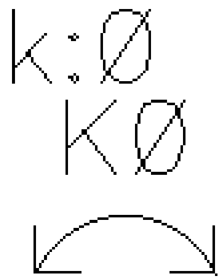
Following is the sample model statement:

```
model ind inductor l=6e-9 r=1 tc1=1e-12 tc2=1e-12 tnom=25
```

Additional Information

This device is supported within the altergroups.

Symbol: pmind



Parasitic Mutual Inductor

Command-line help

```
spectre -h mutual_inductor
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
First coupled inductor	`ind1`	x	-	-	-	-
Second coupled inductor	`ind2`	x	-	-	-	-
Coupling coefficient	`k`	x	-	-	x	x

Syntax/Synopsis

```
Name mutual_inductor <parameter=value> ...
```

Example

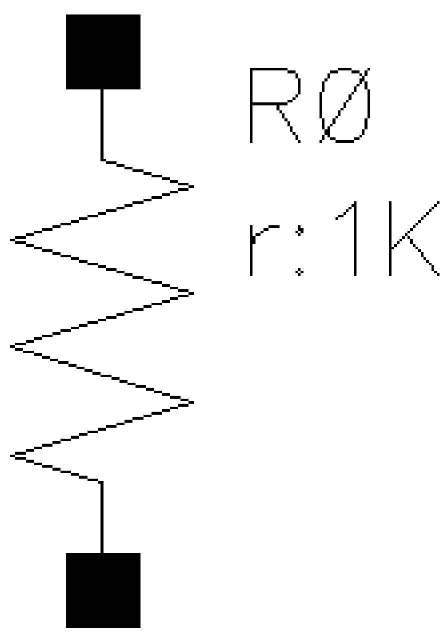
Sample instance statement with two inductors:

```
l1 (1 0) inductor
l2 (2 0) inductor
ml1 mutual_inductor coupling=1 ind1=l1 ind2=l2
```

Additional Information

This device is not supported within the altergroups.

Symbol: presistor



Parasitic Resistor

Command-line help

```
spectre -h resistor
```

CDF Parameters

CDF Parameter	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Resistance	r	x	x	x	x	x
Temperature coefficient 1	tc1	x	-	-	x	x
Temperature coefficient 2	tc2	x	-	-	x	x
Model name	model	x	-	-	-	-
Length	l	x	-	-	x	x
Width	w	x	-	-	x	x
Resistance Form	resform	x	-	-	-	-
Multiplier	m	x	-	-	x	x
Scale factor	scale	x	-	-	x	x
Temp rise from ambient	trise	x	-	-	-	-
Generate noise?	isnoisy	x	-	-	-	-
Capacitance connected	hrc	-	-	-	x	x
Temperature difference	dtemp	-	-	-	x	x
AC resistance	ac	-	-	-	x	-
Capacitance	c	-	-	x	-	-

Syntax/Synopsis

```
Name ( 1 2 ) ModelName <parameter=value> ...
Name ( 1 2 ) resistor <parameter=value> ...
```

Following is the model synopsis:

```
model ModelName resistor <parameter=value> ...
```

Example

Following is a sample without model:

```
r1 (1 2) resistor r=1.2K m=2
```

Following is a sample with model:

```
r1 (1 2) resmod l=8u w=1u
```

Following is a sample model statement:

```
model resmod resistor rsh=150 l=2u w=2u etch=0.05u tc1=0.1 tnom=27 kf=1
```

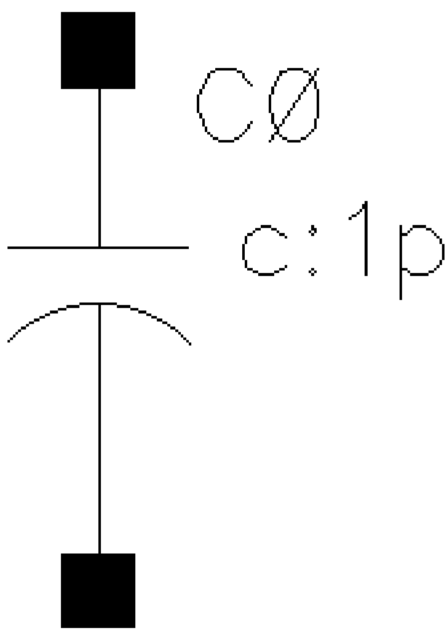
Additional Information

This device is supported within the altergroups.

5

Passive Components

Symbol: cap



Two Terminal Capacitor

You can assign the capacitance or let Spectre compute it from the physical length and width of the capacitor. In either case, the capacitance can be a function of temperature or applied voltage.

Command-line help

```
spectre -h capacitor
```

Component Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Capacitance	`c`	x	x	x	x	x
Initial condition	`ic`	x	-	-	x	x
Model name	`model`	x	-	-	-	-
Width	`w`	x	-	-	x	x
Length	`l`	x	-	-	x	x
Multiplier	`m`	x	-	-	x	x
Scale factor	`scale`	x	-	-	x	x
Temp rise from ambient	`trise`	x	-	-	-	-
Number of Polynomial Coeffs	`polyCoef`	x	-	-	x	x
PolyCoeff 1	`c1`	x	-	-	x	x
Temperature coefficient 1	`tc1`	x	-	-	x	x
Temperature coefficient 2	`tc2`	x	-	-	x	x
Temperature difference	`dtemp`	-	-	-	x	x
Capacitor Area	`area1`	x	x	x	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Capacitor Perimeter	`perim1`	-	x	x	x	x

Syntax/Synopsis

```
Name ( 1 2 ) ModelName <parameter=value> ...
Name ( 1 2 ) capacitor <parameter=value> ...
```

Model Synopsis:

```
model ModelName capacitor <parameter=value> ...
```

Example

Following is a sample without model:

```
c2 (1 0) capacitor c=2.5u tc1=1e-8
```

Following is a sample with model:

```
c2 (1 0) proc_cap c=2.5u tc1=1e-8
```

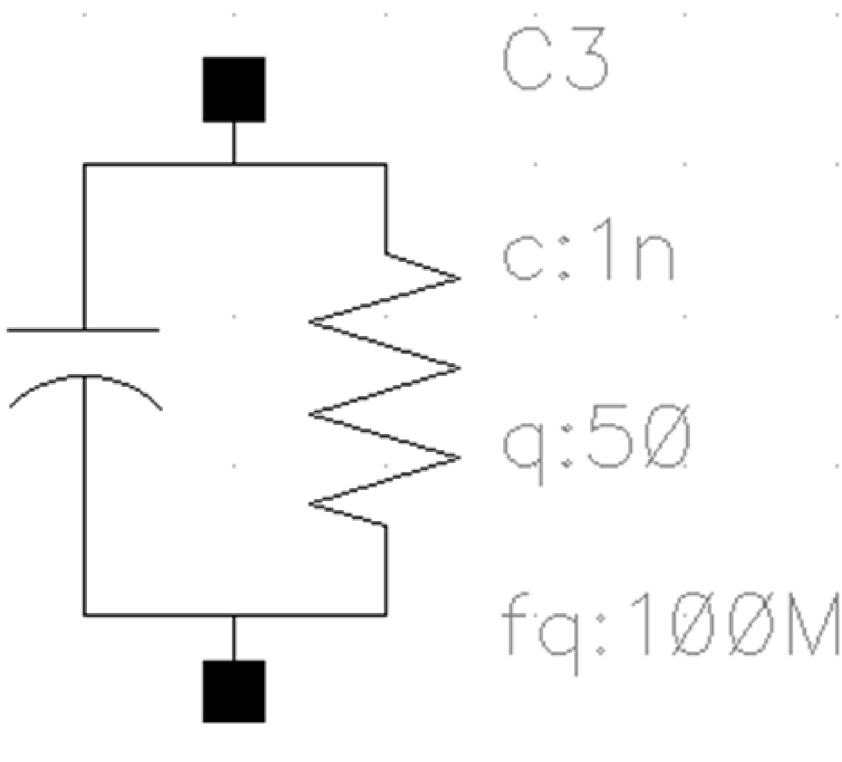
Following is a sample model statement:

```
model proc_cap capacitor c=2u tc1=1.2e-8 tnom=25
```

Additional Information

This device is supported within the altergroups.

Symbol: capq



Two Terminal Capacitor Resistor

The `capq` instance is a capacitor with the Q factor `q`' specified at a particular frequency `fq`.

The parameter mode specifies the exact form of the frequency dependence of the real part of the admittance `Re(Y)`. The equations are written in terms of admittance Y, where by default,

- `w=2*Pi*freq`
- `wq=2*Pi*fq`
- `fq=1.0e8 Hz`. This is the frequency at which `q` is measured.
- `mode=1`. This is the integer parameter that selects the frequency dependence.
- `alph=0`. This is the scaling factor for `Q`.
  - `qf=q*(freq/fq)^alph`
  - `Y=2*pi*freq*c*(1/qf+j)`

The following table describes the various modes and the corresponding equations:

Mode	Description
	Constant real part of the admittance Re(Y)=const.
	Equation: `Re(Y)=wq*c/q=const; Im(Y)=w*C`.
`2`	Re(Y) increases proportional to sqrt(freq).
	Equation: `Re(Y)=c*sqrt(wq*w)/q; Im(Y)=w*C`.
`3`	Re(Y) increases linearly with frequency.
	Equation: `Re(Y)=w*c/q; Im(Y)=w*C`.
`4`	Re(Y) decreases proportional to (freq/fq)^alph.
	Equation: `Re(Y)=w*c/Qf; Im(Y)=w*c; Qf=q*(freq/fq)^alph`.

Command-line help

```
spectre -h capq
```

Component Parameters

CDF Parameter Label	CDF Parameter	spectre
Alpha parameter	`alph`	x
Capacitance	`c`	x
Frequency for C and Q	`fq`	x
Generate noise?	`isnoisy`	x
Mode	`mode`	x
Model name	`model`	x
Multiplier	`m`	x
Quality factor	`q`	x
Temp rise from ambient	`trise`	x

Example

```
C0 (net1 net2) capq c=1n q=50 fq=100M mode=4 alph=0.35
```

Additional Information

This device is supported within the altergroups.

Symbol: core

Magnetic Core with Hysteresis

This component models the magnetic hysteresis, with air gap, frequency, and temperature effects. The model is based on the AWB model for magnetic cores and windings. The user has to specify the core's material and geometric parameters to model the hysteresis.

The material parameters to specify are the 'Br', 'Bm', and 'Hc' of the core. The geometric parameters are the area, magnetic path length, and the air gap of the core.

You can specify the magnetic path length in one of the following ways:

- Give the length directly in cm.
- Or give the outer and inner diameter of the core.

Cores without terminals represent complete magnetic loops. Cores with terminals are fragments that you can use as building blocks to build models of complicated core structures.

Command-line help

```
spectre -h core
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Device area	`area`	x	-	-	-	-
Physical length	`len`	x	-	-	-	-
Inner diam of toroidal core	`idiam`	x	-	-	-	-
Outer diam of toroidal core	`od`	x	-	-	-	-
Gap length	`gap`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Total Num of windings	`numOfL`	-	-	-	x	-
Name of winding 1	`l1 - l20`	-	-	-	x	-
Initial magnetization of core	`mag`	-	-	-	x	-

Syntax/Synopsis

```
Name ... ModelName <parameter=value> ...
```

Model Synopsis

```
model ModelName core <parameter=value> ...
```

Example

Following is a sample instance statement:

```
c1 (1 0) core_mod area=1.2 len=8.1 id=0.55 gap=0.25
```

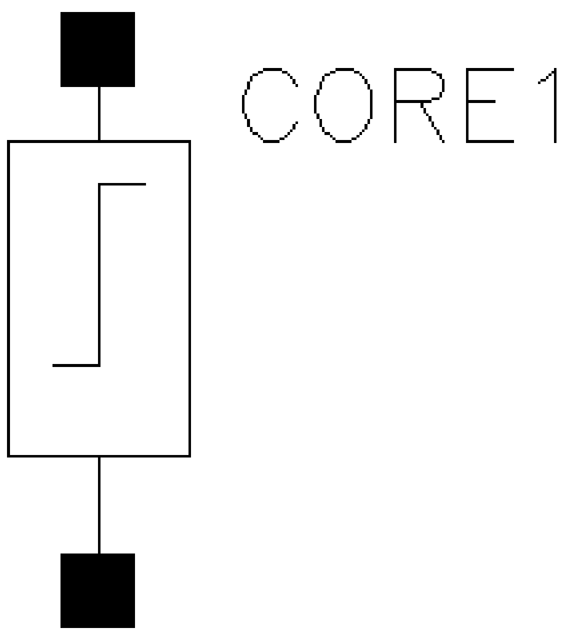
Following is a sample model statement:

```
model core_mod core len=7.7 area=0.85 br=1e3 bm=5e3 hc_t1=0.2 p1_f1=2.08
f1=10e3 p2_f2=50 f2=100K bflux=1e3 density=4.75
```

Additional Information

This device is not supported within the altergroups.

Symbol: corefragment



Magnetic Core with Hysteresis

Cores without terminals represent complete magnetic loops. Cores with terminals are fragments that you can use as building blocks to build models of complicated core structures.

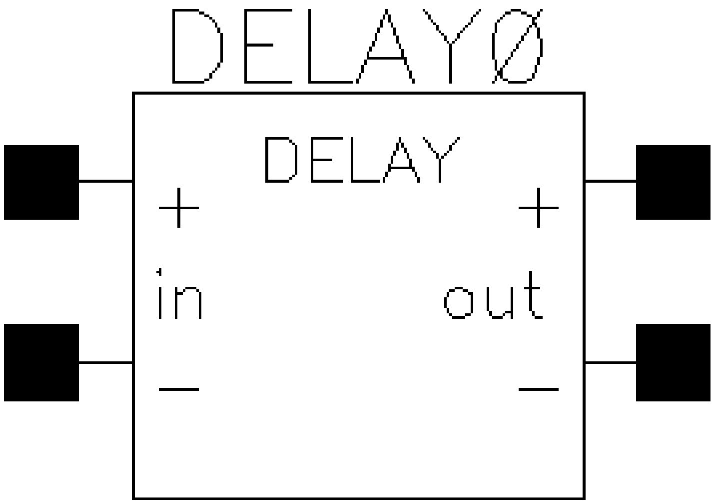
Command-line help

```
spectre -h core
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Device area	`area`	x	-	-	-	-
Physical length	`len`	x	-	-	-	-
Inner diam of toroidal core	`idiam`	x	-	-	-	-
Outer diam of toroidal core	`od`	x	-	-	-	-
Gap length	`gap`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-

Symbol: delay



Delay Line

The delay line model is a four terminal device with zero output impedance and infinite input impedance. The output between nodes ‘p’ and ‘n’ is the input voltage between nodes ‘ps’ and ‘ns’ delayed by the time delay ‘td’ and scaled by ‘gain’.

Command-line help

```
spectre -h delay
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Delay time	`td`	x	-	-	-	-
Gain	`gain`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ps ns ) delay <parameter=value> ...
```

Example

```
dl1(outp outn cntrlp cntrln) delay td=10n gain=1.5
```

Additional Information

This device is not supported within the altergroups.

Symbol: delayline

Delayline

The delayline element is a lossless transmission line section with a specified delay time T_d and characteristic impedance Z_0. The ABCD matrix of a lossless transmission line section is given by:

$$ABCD = \begin{bmatrix} \cos(\omega T_d) & jZ_d \sin(\omega T_d) \\ \dfrac{j\sin(\omega T_d)}{Z_0} & \cos(\omega T_d) \end{bmatrix}$$

Command-line help

```
spectre -h mtline
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Characteristic impedance	`zo`	x	-	-	x	x
Delay time	`td`	x	-	-	x	x

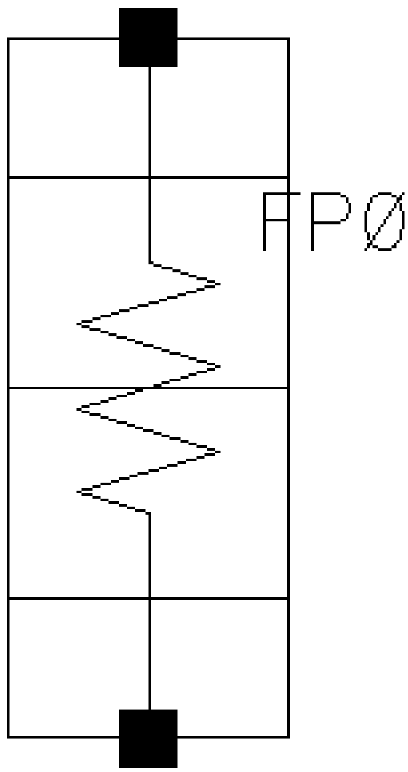
Syntax/Synopsis

```
Name ( in out ) mtline <parameter=value> ...
```

Example

```
DL0 ( net1 net2 ) mtline z0=50 td=1n
```

Symbol: fracpole



Fractional Impedance/Admittance Pole

The circuit is a one-port that exhibits poles and zeros that are real and spaced evenly in a logarithmic sense over the frequency range. The impedance exhibited by one port approximates a fractional pole slope between -1 and 0 in the frequency range. In other words, if the impedance is plotted on a log-log scale, it has a negative slope equal to the fraction specified. If the user requested half a pole, the slope will be -1/2, and so on. Since it is a lumped approximation, the slope is not exact but slowly oscillates about the specified value.

Command-line help

```
spectre -h fracpole
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Low freq. limit for approx.	`f0`	x	-	-	-	-
Freq 1 to Freq50High freq. limit for approx.	`f1`	x	-	-	-	-
Unity intercept point	`coef`	x	-	-	-	-
Slope of imp on log/ log scale	`slope`	x	-	-	-	-
Num of lumps in approx.	`lumps`	x	-	-	-	-
Num of lumps/dec in approx.	`dec`	x	-	-	-	-
Profile	`profile`	x	-	-	-	-
Multiplicity factor	`m`	x	-	-	-	-
Initial condition	`ic`	x	-	-	-	-
Res. for initial conds.	`rforce`	x	-	-	-	-

Syntax/Synopsis

```
Name ( 1 2 ) ModelName <parameter=value> ...
Name ( 1 2 ) fracpole <parameter=value> ...
```

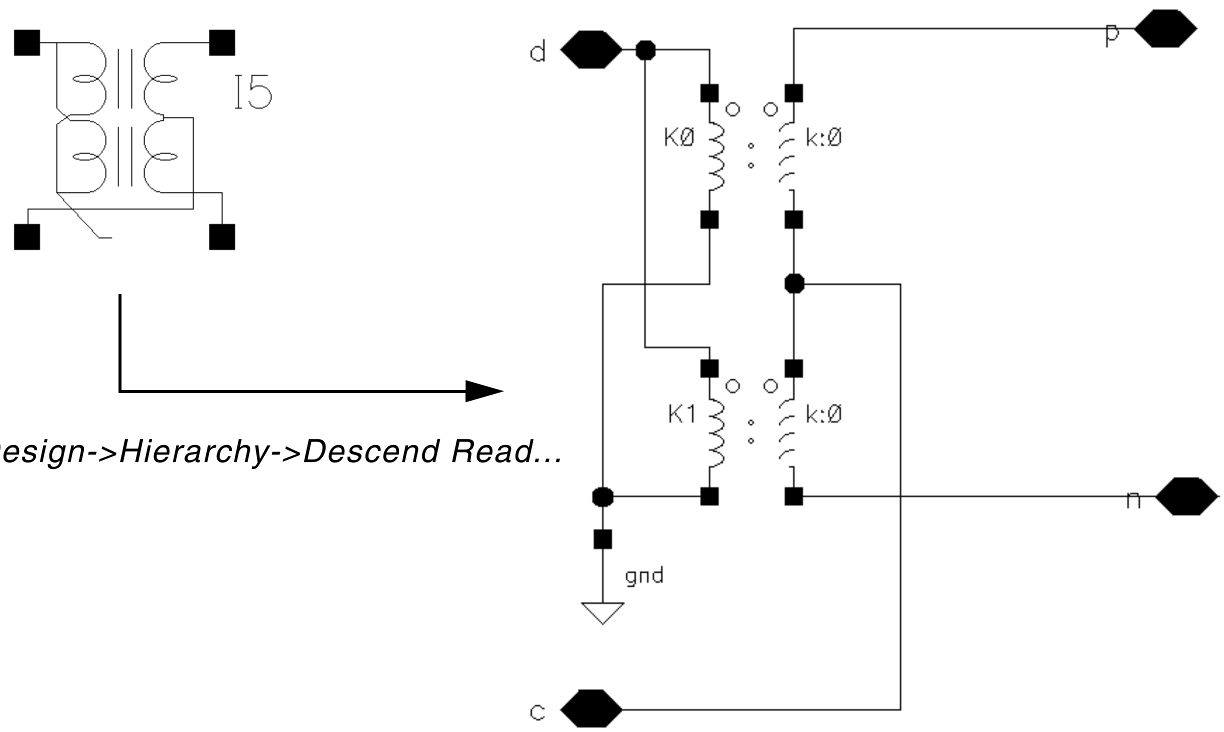
Model Synopsis:

```
model ModelName fracpole <parameter=value> ...
```

Additional Information

This device is supported within the altergroups.

Symbol: ideal_balun



Balun

The balun is a bidirectional balanced-unbalanced convertor that can be used in circuits that require single or differential signal transformation. Although a passive network (including the transformer) is used to achieve balun, this implementation employs a three-port network. It requires three ports (or nodes) because the reference nodes are always at the global ground, single, blip, and bal_n.

The balun is used for converting ground-referred differential-mode (d) and common-mode (c) signals to balanced positive (p) and negative (n) signals.The balun is accurate at all frequencies including DC, because it uses ideal transformers.

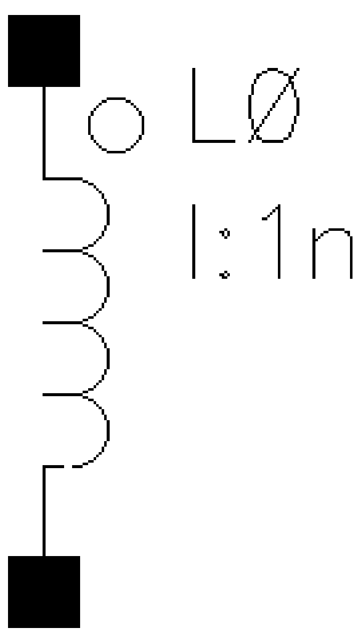
```
subckt balun (d c p n)
    T1 (d 0 p c) transformer n1=2
    T2 (d 0 c n) transformer n1=2
ends balun
```

Notice that the balun is bidirectional, you can use, as inputs or outputs, either the unbalanced signals (d for differential mode and c for common-mode) or the balanced signals (p for positive and n for negative).

Component Parameters

ideal_balun has no component parameters.

Symbol: ind



Two Terminal Inductor

The inductance of this component can be a function of temperature or branch current. If you do not specify the inductance in the instance statement, it is taken from the model.

Command-line help

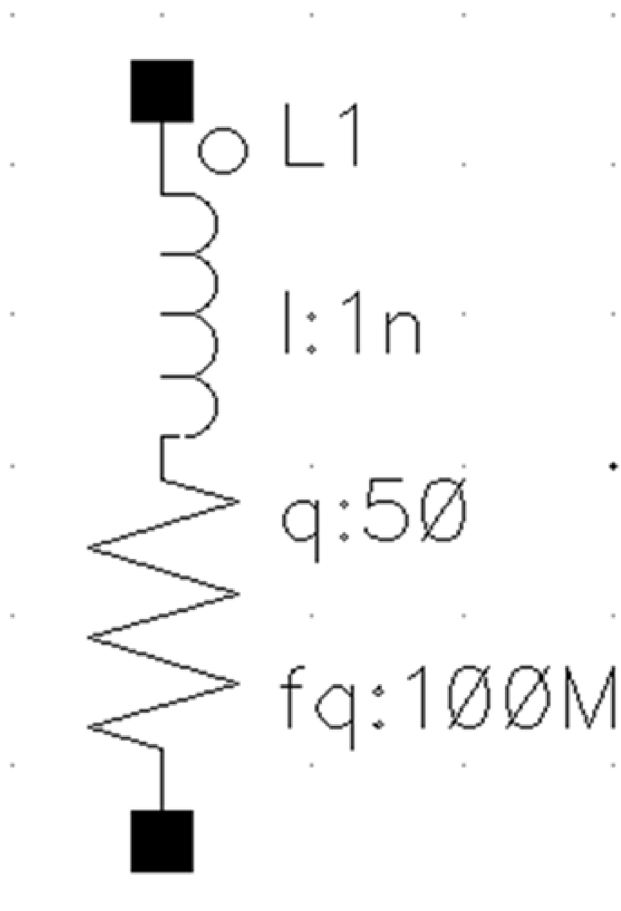
```
spectre -h inductor
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Use S-parameters	useSParamsCheckBox	x	-	-	-	-
Browse s-parameter file	nportFileB	x	-	-	-	-
S-parameter data file	dataFile	x	-	-	-	-
Model name	model	x	-	-	-	-
Inductance	l	x	x	x	x	x
Resistance	r	x	-	-	x	x
Multiplier	m	x	-	-	x	x
Temp rise from ambient	trise	x	-	-	-	-
Initial condition	ic	x	-	-	x	-
Temperature coefficient 1	tc1	-	-	-	x	x
Temperature coefficient 2	tc2	-	-	-	x	x
Generate noise?	isnoisy	x	-	-	-	-
Scale factor	scale	-	-	-	x	x
Number of Polynomial Coeffs	polyCoef	-	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Poly Coeff 1	`c1 - c20`	-	-	-	x	-
Temperature difference	`dtemp`	-	-	-	x	x

Symbol: indq



Two Terminal Series Inductor Resistor

The inductance of this component can be a function of temperature or branch current. If you do not specify the inductance in the instance statement, it is taken from the model statement.

The `indq` component also has the frequency-dependent Q-factor with four modes of frequency dependence. This component has the following optional parameters:

- `fq=1e8 Hz`. This is the frequency at which `l` and `q` are measured.
- `mode=0`. This is the integer parameter that selects the frequency dependence.
- `alph=0`. This is the scaling factor for `Q`.
- `rdc=0.0`. This is the DC resistance used in `mode=2` and `mode=3`.

- `qf=q*(freq/fq)^alph, Z=2*pi*freq*L*(1/qf+j).`

Command-line help

```
spectre -h inductor
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl
Use S-parameters	useSParamsCheck Box	x	x
Browse s-parameter file	nportFileB	x	x
S-parameter data file	dataFile	x	x
Alpha parameter	alph	x	x
Mode	mode	x	x
Model name	model	x	x
Inductance	l	x	x
Multiplier	m	x	-
Temp rise from ambient	trise	x	x
Generate noise?	isnoisy	x	x
Quality factor	q	x	-
Rdc in mode 2 and 3	rdc	x	x
Frequency for L and Q	fq	x	-

Example

```
L1 (net1 net2) inductor l=1n q=50 fq=100M mode=4 alph=0.55
LL0 vout net7 1n $[LP]
```

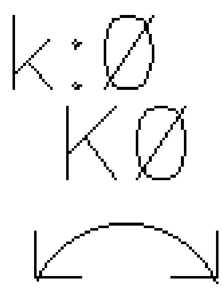
Additional Information

This device is supported within the altergroups.

During auCdl netlisting, the `Multiplier`, `Quality Factor`, and `Frequency for L and Q` CDF parameters are not netlisted. The indq component is netlisted similar to the `ind` component.

For more information, see indq.

Symbol: mind



Mutual Inductor

It couples two previously specified inductors. There is no limit to the number of inductors that you can couple or to the number of couplings to a particular inductor, but you must specify separate mutual inductor statements for each coupling. Using the ‘dot’ convention, place a ‘dot’ on the first terminal of each inductor.

The mutual inductor modifies the constitutive equations of two isolated inductors to:

```
v1 = L11*di1/dt + M*di2/dt
v2 = M*di1/dt + L22*di2/dt
```

where the mutual inductance, M, is computed from the coupling coefficient, k, using k = |M|/sqrt(L11*L22).

Command-line help

```
spectre -h mutual_inductor
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
First coupled inductor	`ind1`	x	-	-	-	-
Second coupled inductor	`ind2`	x	-	-	-	-
Coupling coefficient	`k`	x	-	-	x	x

Syntax/Synopsis

```
Name mutual_inductor <parameter=value> ...
```

Example

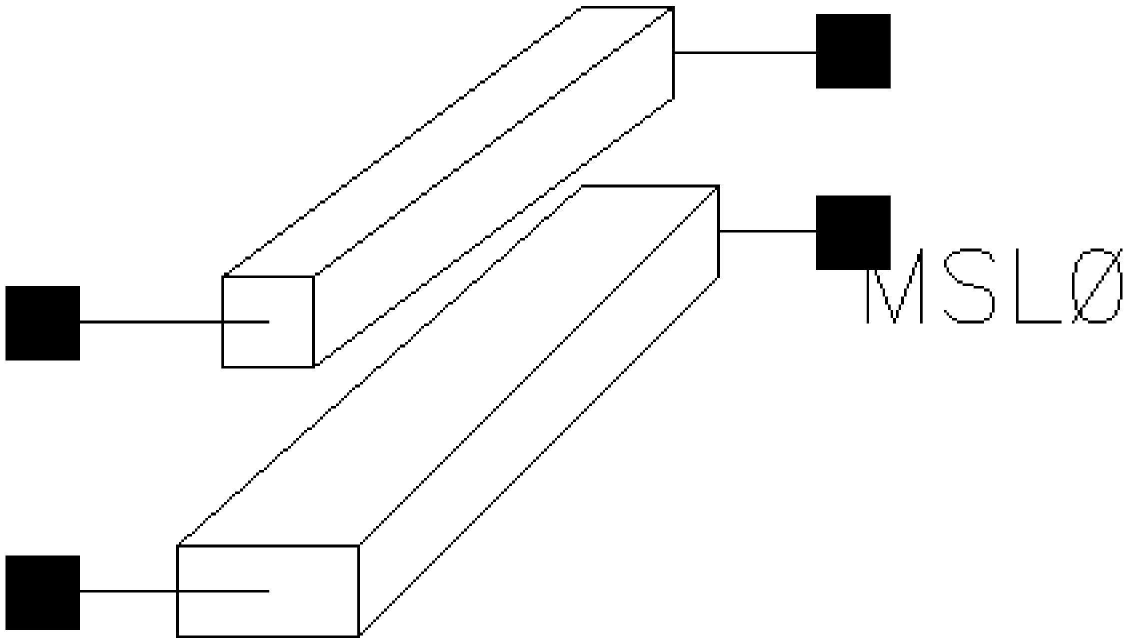
Following is a sample instance statement with two inductors:

```
l1 (1 0) inductor
l2 (2 0) inductor
ml1 mutual_inductor coupling=1 ind1=l1 ind2=l2
```

Additional Information

This device is not supported within the altergroups.

Symbol: msline



Microstrip Line

It is a microstrip line based on the equations of Hammerstad and Jensen. The model contains a thickness correction to the width and frequency dependent permittivity and characteristic impedance. The dispersion equations are those of Kirschning and Jansen.

Command-line help

```
spectre -h msline
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Length	l	x	-	-	-	-
Width	w	x	-	-	-	-
Substrate height	h	x	-	-	-	-
Conductor thickness	t	x	-	-	-	-
Relative permittivity	eps	x	-	-	-	-
Multiplier	m	x	-	-	-	-
Max signal frequency	fmax	x	-	-	-	-

Syntax/Synopsis

```
Name ( t1 b1 t2 b2 ) msline <parameter=value> ...
```

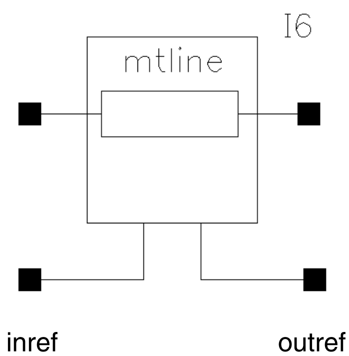
Example

```
tl1 (in 0 out 0) msline l=0.15 w=0.01 h=0.01
```

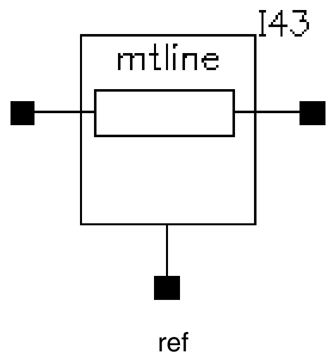
Additional Information

This device is supported within the altergroups.

Symbol: mtline



mtline without sub-circuit
(even number of pins)



mtline with sub-circuit
(odd number of pins)

Multi-Conductor Transmission Line

It is characterized by constant RLCG matrices or frequency dependent RLCG data. An mtline can have as many conductors as defined in the *Num of lines (excluding ref.)* field. However, there must be at least two conductors, with one conductor used as a reference, to define terminal voltages. The reference conductor can be ground. The order of the conductors is the same as the order of data in the input.

All of the conductors are assumed to have the same length. The input to `mtline` are conductor length, per-unit-length resistance (R), inductance (L), capacitance (C), and conductance (G) matrices. As these matrices are symmetric, either a full matrix description or a lower half matrix description can be used.

You can use mtline in four different modes:

- *RLGC* - Lets you specify the per-unit-length constant RLGC matrices and frequency dependent RLGC data file
- *FieldSolver* - Lets you specify the 2-D field solver geometry and material property information
- *S-parameter* - Lets you specify the S-Parameter data
- *Tline* - Lets you specify the old single-conductor tline parameters (to ease migration)

Parameters for the mtline component

This section describes the following CDF parameters for the `mtline` component:

Num of lines (excluding ref.), lets you specify the number of lines excluding the reference lines. The reference conductor is used as a return path. There is no upper limit on number of conductors that mtline can have in Spectre. However, there must be at least two conductors with one conductor used as reference to define terminal voltages.The reference conductor can be ground. The order of the conductors is the same as the order of the data in the input.

Model name lets you specify the name of the model to be associated with the `mtline` component.

Physical length lets you specify the physical length of the line, required in order to perform the transmission line simulation. All the conductors in an mtline instance are assumed to have the same length, and to be uniform along the length. Default = `0.01m`.

When using S-Parameter data, the physical length of the line must be specified.

In the Tline use model, physical length is used with Propagation velocity normalized to specify the electrical length of the line.

Multiplicity factor lets you specify the multiplicity factor of the `mtline` component. The valuemust be a nonzero real number. This number lets you specify a number of mtline components in parallel. Default=`1`.

Max signal frequency lets you specify the maximum signal frequency used to determine the relevant range of rational fitting used in the 2D field solver. Default = `25e09 Hz`.

Spectre uses the rational fitting algorithm to build a stable model that approximates the desired transmission line characteristics. The *Max signal frequency* is used to determine the relevant range of rational fitting. The accuracy of the mtline model depends on how well the rational approximation is over frequency range from fmax to fmin. When constant RLGC matrices are provided, fmin is set to 1Hz and fmax defaults to 25GHz. A good estimate of the Max signal frequency is three times the inverse of rise time in the input signal. When a RLGC data file is provided, the lowest frequency point in the data file is used as fmin' and the largest frequency point in the data file is used as fmax. You must provide sufficient data points to cover both low frequencies and high-frequencies to obtain an accurate, stable model.

Type of Input lets you select a type of input, and displays additional fields required for the specified type of input in the form. Possible values are: *RLGC*, *FieldSolver*, *Tline*, and *S-Parameter*.

If you select `RLGC` or `S-Parameter` as the type of input, you can select the *RLGC data file as Design Var?* check box or the *S-Parameter file as Design Var?* check box.These check boxes let you use a design variable to specify the RLGC data file or the S-Parameter file.

RLGC data file as Design var?, when selected, lets you use a design variable to specify the RLGC data file. This check box is shown only when you select the *RLGC* option for *Type of Input*.

RLGC data file lets you specify the RLGC data file that contains the frequency dependent RLGC data. This field is shown only when the *RLGC* option for *Type of Input* is selected and the *use lmg subckt* check box is not selected on the Edit Properties form.

The RLGC data file parameter can be used to store the 2-D field solver output for use in subsequent simulations. If the file parameter is given, mtline checks the existence of the file. If the file does not exist, the RLGC model is generated by the field solver and the output is stored in a file. If the file exists, mtline checks if the RLGC data stored in the file matches the mtline 2-D field solver input. If it does not match, a new set of RLGC data is generated and the file is overwritten. Otherwise, the data is reused. If the RLCG data file parameter is not given, the RLGC data is stored in the `input.rlgc` file in the `simulation/circuitName/spectre/schematic/netlist` directory.

use lmg subckt, when selected, shows the *LMG subcircuit file* field and hides all RLCG-specific fields. This field is shown only when you select the *RLGC* option for *Type of Input*.

LMG subcircuit file lets you specify the name of the LMG subcircuit file. This field is shown only when you select the *use lmg subckt* check box.

Enter RLCG etc. matrices, when selected, displays the following additional fields. This check box is shown only when you select the *RLGC* option for *Type of Input*. The following fields are vectors. For example, if you want to multiply your R matrix per unit length by a design variable `myScale`, you need to surround the expression with parentheses. For example, enter it as `(myScale*50)`. If you do not add the parentheses, Spectre fails during simulation.

- ***R matrix per unit length*** lets you specify the resistance matrix per unit length. Units: `Ohm/m`.
- ***L matrix per unit length*** lets you specify the inductance matrix per unit length. Units: `H/m`.
- ***G matrix per unit length*** lets you specify the conductor matrix per unit length. Units: `S/m`.
- ***C matrix per unit length*** lets you specify the capacitance matrix per unit length. Units: `F/m`.

- ***Skin effect res matrix per unit length*** lets you specify the skin effect resistance matrix per unit length. Units: `Ohm/m*sqrt(Hz)`
- ***Dielectric loss cond matrix per unit length*** lets you specify the dielectric loss conductance matrix per unit length. Units: `S/m*Hz`

  `mtline` supports LC, RC, RGC, RLG, RLC, and RLGC transmission line systems. For example,

  - When only L and C matrices are provided, a lossless transmission line system is modeled.
  - When only R and C matrices are provided, an RC transmission line system is modeled.

  Since the per-unit-length RLGC matrices are generally symmetric, either a full matrix description or a lower half matrix description is accepted. You enter the matrix as a series of numbers, e. g. `50 10 1 10 50 10 1 10 50` (full matrix) or `50 10 50 1 10 50` (half matrix). Spectre determines whether the matrix is full or half matrix depending on the number of entries. For example, for a `3x3` matrix, if you enter six entries, Spectre knows that it is a half matrix. If you enter nine entries, Spectre knows that it is a full matrix. Spectre complains if the number of entries does not make either a full or a half matrix.

S-parameter file as Design Var?, when selected, lets you use a design variable to specify the S-Parameter file. This check box is shown only when you select the *S-parameter* option for *Type of Input*.

S-parameter File lets you specify the data file that contains the frequency dependent RLGC data or S-Parameter data file. This field is shown only when you select the *S-parameter* option for *Type of Input*.

You can also specify Y- or Z-parameters. Spectre parses the data file and determines whether the data is in S-, Y-, or Z-parameter format.

The supported S-Parameter data file formats include Touchstone, Spectre and CITIfile.

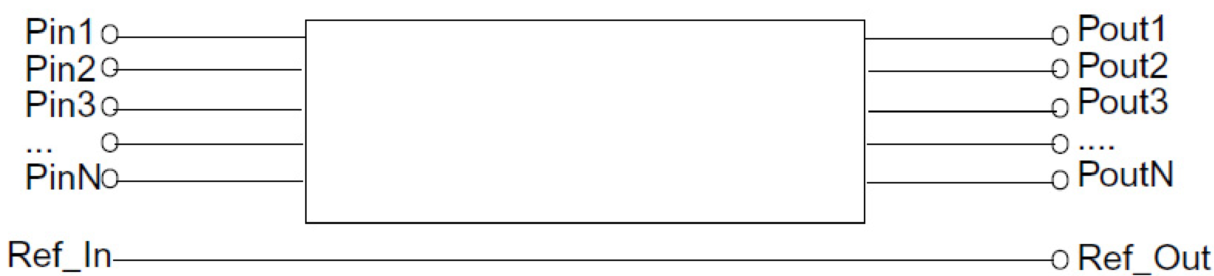
Spectre reads the comment line in the S-Parameter data file to determine whether the frequency data is in Hz, MHz, GHz, etc. It also determines the characteristic impedance and format of the data (real,imag), (mag,deg), (mag,rad), (db,deg), or (db,rad).

Spectre imports the S-Parameter data describing a transmission line system. mtline converts the frequency dependent S-Parameter to frequency dependent RLGC data and stores the results in the `input.rlgc` file,located in `simulation/CircuitName/spectre/schematic/netlist` directory, for reuse in subsequent simulations.

When the file parameter corresponds to S-Parameter data, `mtline` first checks the existence of the `input.rlgc` file to determine if the S-to-RLGC extraction has been performed in a previous simulation.

The ordering of the S-Parameter input file should be in the format of input ports followed by the output ports of the transmission line system, or Pin1, Pin2, Pin3, ..., Pout1, Pout2, Pout3, ...

The S-Parameter data file for use with mtline should have the S-Parameter data interpreted in the following order.



Frequency scale factor lets you specify the frequency scale factor for frequency-dependent RLGC data and S-Parameter data. This field is shown only when you select the *RLGC* or *S-parameter* option for *Type of Input*. The default value is `1`.

Generate noise? lets you control whether the `mtline` component must generate noise. Possible values are *yes* and *no*.

Transmission line type, lets you choose the transmission line type. Possible values are microstrip line (*microstrip*), stripline (*stripline*), coplanar waveguide (*coplanar*), and substrate lossy line(*sublossline*). Default = *sublossline*. If you need to include dielectric loss in your microstrip or stripline model, use the sublossline transmission line type.

Model type, lets you specify the model type. For each line configuration, you can choose one of three model types: *lossless*, *narrowband*, or *wideband*. Default = *wideband*.

In the lossless model, the internal inductance of the conductor is disregarded by setting the frequency value high; 30GHz for cases without substrate loss and 15 GHz for cases with substrate loss, and ignoring the value of Max signal frequency (fmax).

For the narrowband model, the RLGC data is calculated at the Max signal frequency (fmax, default is 25GHz) and assumed to be constant over the frequency of interest. The narrowband model is valid near fmax.

With the wideband model, true frequency dependent RLGC data is calculated over the frequency of interest (DC to fmax). For most applications, choose the wideband model as it provides the best model accuracy.

Number of dielectric layers lets you specify the number of dielectric layers. Dielectric layers are stacked above the ground plane (when `numgnd=1`), or between the ground planes (when `numgnd=2`). There can be up to `12` dielectric layers.This field is shown only when you select the *FieldSolver* option for *Type of Input* and *coplanar* or *sublossline* for *Transmission line type*.

Number of Ground Planes lets you specify the number of ground planes, This field is shown only when you select the *FieldSolver* option for *Type of Input* and *coplanar* or *sublossline* for *Transmission line type*.

Rel dielectric const of layers(er) lets you specify the relative dielectric constant of the dielectric layer. It is a vector type that handles different layer geometries and layer properties. When the number of elements in the vector is less than the number of layers, the value of the last element in the vector is applied to all of the remaining layers.This field is shown only when you select the *FieldSolver* option for *Type of Input*.

Dielectric layer thickness (d) lets you specify the dielectric layer thickness. It is a vector type to handle different layer geometries and layer properties. When the number of elements in the vector is less than the number of layers, the value of the last element in the vector is applied to all of the remaining layers. Units = `meters`.This field is shown only when you select the *FieldSolver* option for *Type of Input*.

Dielectric loss type lets you specify the dielectric loss type. The loss value is specified using the *Dielectric layer loss* parameter. Possible values are *sigma* and *tangent*. A particular dielectric layer can be lossy, and either the loss tangent parameter (`tan = sigma/(w*ep0)`) or the loss sigma parameter (`sigma = tan*w*ep0`) can be used. Default value: `tangent`.This field is shown only when you select the *FieldSolver* option for *Type of Input* and *coplanar* or *sublossline* for *Transmission line type*.

Dielectric layer loss lets you specify the dielectric layer loss. The loss can be in terms of dielectric conductivity or tangent loss, determined by the *Dielectric loss type* parameter.This field is shown only when you select the *FieldSolver* option for *Type of Input* and *coplanar* or *sublossline* for *Transmission line type*.

Signal line width lets you specify the signal line width, This field is shown only when you select the *FieldSolver* option for *Type of Input*. When the number of elements in the vector is less than the number of layers, the value of the last element in the vector is applied to all of the remaining layers. Units: `meters`.

Signal line thickness lets you specify the signal line thickess. This field is shown only when you select the *FieldSolver* option for *Type of Input*. When the number of elements in the vector is less than the number of layers, the value of the last element in the vector is applied to all of the remaining layers. Units: meters.

Signal line height (h) lets you specify the signal line height. This field is shown only when you select the *FieldSolver* option for *Type of Input*. The distance between the signal line and ground plane at the bottom of the 2-D interconnect cross section. When the number of elements in the vector is less than the number of layers, the value of the last element in the vector is applied to all of the remaining layers. Units: `meters`.

Signal line spacing lets you specify the signal line spacing(the distance between the signal lines). This field is shown only when you select the *FieldSolver* option for *Type of Input*. It can be negative in order to describe overlapping signal lines. When the number of elements in the vector is less than the number of layers, the value of the last element in the vector is applied to all the remaining layers. Units: `meters`.

Gnd Plane thickness lets you specify the ground plane thickness. This field is shown only when you select the *FieldSolver* option for *Type of Input*. When the number of elements in the vector is less than the number of layers, the value of the last element in the vector is applied to all of the remaining layers. Units: `meters`.

Ground plane conductivity lets you specify the ground plane conductivity. This field is shown only when you select the *FieldSolver* option for *Type of Input*. Units: `S/m`.

Signal line conductivity lets you specify the signal line conductivity. This field is shown only when you select the *FieldSolver* option for *Type of Input*. Units: `S/m`.

Charecteristic impedance lets you specify the characteristic impedance of lossless line. This field is shown only when you select the *Tline* option for *Type of Input*. Default: `50`. Units: `Ohms`.

Delay Time lets you specify the time delay of a lossless line in seconds; a measure of the electrical length. This field is shown only when you select the *Tline* option for *Type of Input*. Units: `seconds`.

Frequency lets you specify the reference frequency, which is used in conjunction to the normalized length to specify electrical length of line.This field is shown only when you select the *Tline* option for *Type of Input*. Units: `Hz`.

Normalized length, lets you specify the normalized electrical length in wavelengths at the specified reference frequency of a lossless line.This field is shown only when you select the *Tline* option for *Type of Input*. Default: `0.25`.

Propagation velocity normalized lets you specify the propagation velocity of the line given as a multiple of c, the speed of light in free space (`vel <= 1`). This field is shown only when you select the *Tline* option for *Type of Input*. `vel=c/sqrt(år)`.

Corner frequency, lets you specify the corner frequency for skin effect. This is the frequency where skin depth equals the wall thickness of the conductor.This field is shown only when you select the *Tline* option for *Type of Input*. Default: `0`. Units: `Hz`.

DC series res/Length lets you specify the DC series resistance per unit length.This field is shown only when you select the *Tline* option for *Type of Input*. Default: `0`. Units: `Ohm/m`.

Loss resistance per unit length lets you specify the conductor (series) resistance per unit length at conductor loss frequency.This field is shown only when you select the *Tline* option for *Type of Input*. Default: `0`. Units: `Ohm/m`.

- `seriesr` = `2*z0*alphac` (when `alphac` is given)
- `seriesr` = `2*z0*fc/(2*qc*c*vel)` { when `qc` is given }

where `seriesr` is the *Loss resistance per unit length*, `c` is the speed of light. `z0` is *Characteristic Impedance*, `fc` is *Conductor loss frequency*, `alphac` is *Conductor loss at fc* , and `qc` is *Conductor loss quality factor*.

Conductor loss at fc lets you specify the conductor loss at the conductor loss frequency (low loss approximation). This field is shown only when you select the *Tline* option for *Type of Input*. Default: `0`. Units: `dB/m`.

Conductor loss quality factor lets you control the conductor loss quality factor at conductor loss frequency (low loss approximation). This field is shown only when you select the *Tline* option for *Type of Input*. Default: `infinity`.

Dielectric loss frequency lets you specify the dielectric loss measurement frequency. It is used in conjunction with *Dielectric loss quality factor*. This field is shown only when you select the *Tline* option for *Type of Input*. Units: `Hz`.

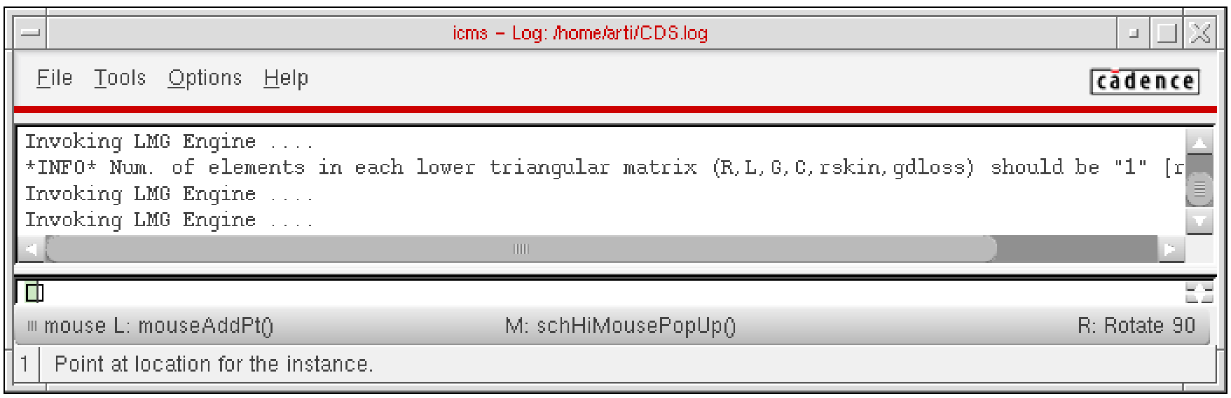
Loss conductance per unit length, lets you specify the dielectric (shunt) conductance per unit length at conductor loss measurement frequency. This field is shown only when you select the *Tline* option for *Type of Input*.

Dielectric loss lets you specify the dielectric loss (low loss approximation). This field is shown only when you select the *Tline* option for *Type of Input*.

Dielectric loss quality factor lets you specify the dielectric loss quality factor at dielectric loss measurement frequency (low loss approximation). This field is shown only when you select the *Tline* option for *Type of Input*.

Conductor loss frequency lets you specify the conductor loss frequency. It is used in conjunction with *Loss resistance per unit length*, *Conductor loss at fc*, and *Conductor loss quality factor*. This field is shown only when you select the *Tline* option for *Type of Input*.

The number of elements that you need to specify for the R/L/G is determined by the number of lines that you specify. For example, if the number of lines (n) is 3, then you need to specify (n*2) 6 elements each for R, L, and G. This information is displayed in CIW as follows.



Command-line help

```
spectre -h mtline
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Num of lines (excluding ref.)	n	x	-	-	-	-
Model name	model	x	-	-	-	-
Physical length	len	x	-	-	-	-
Multiplicity factor	mf	x	-	-	-	-

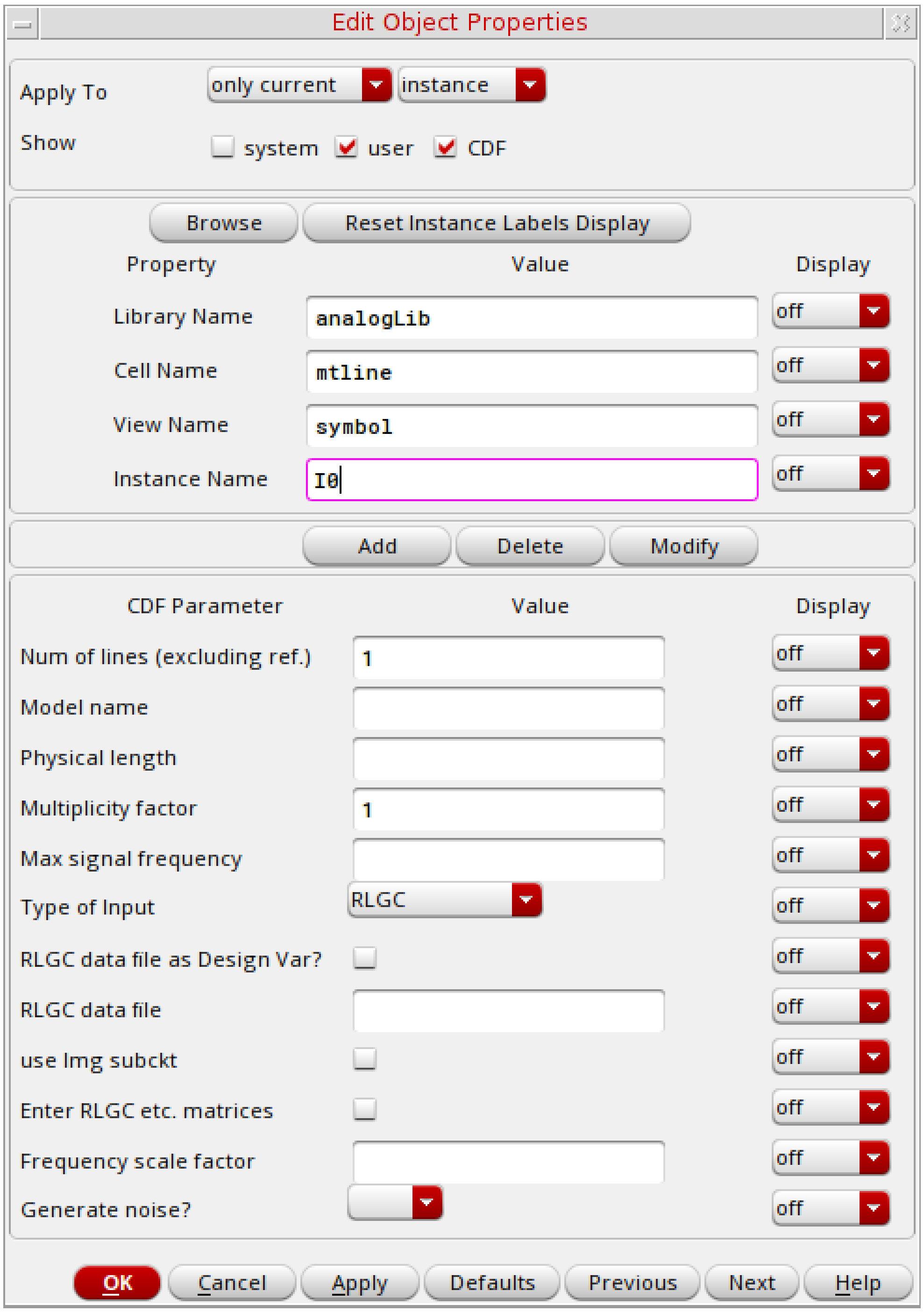
CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Max signal frequency	`fmax`	x	-	-	-	-
Type of Input	`modelType`	x	-	-	-	-
RLCG data file as Design var?	`rlgc_file_as_var`	x	-	-	-	-
RLCG data file	`file`	x	-	-	-	-
use lmg subckt	`uselmg`	x	-	-	-	-
LMG subcircuit file	`subcktfile`	x	-	-	-	-
Enter RLCG etc. matrices	`entermatrices`	x	-	-	-	-
R matrix per unit length	`R`	x	-	-	-	-
L matrix per unit length	`L`	x	-	-	-	-
G matrix per unit length	`G`	x	-	-	-	-
C matrix per unit length	`C`	x	-	-	-	-
Skin effect res matrix per unit length	`rskin`	x	-	-	-	-
Dielectric loss cond matrix per unit length	`gdloss`	x	-	-	-	-
S-parameter file as Design var?	`sparam_file_as_var`	x	-	-	-	-
S-parameter File	`file1`	x	-	-	-	-
Frequency scale factor	`freqscale`	x	-	-	-	-
Generate noise?	`isnoisy`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Transmission line type	linetype	x	-	-	-	-
Model type	modeltype	x	-	-	-	-
Number of dielectric layers	numlayer	x	-	-	-	-
Number of Ground Planes	numgnd	x	-	-	-	-
Rel dielectric const of layers(er)	er	x	-	-	-	-
Dielectric layer thickness (d)	layerthickness	x	-	-	-	-
Dielectric loss type	dlosstype	x	-	-	-	-
Dielectric layer loss	dloss	x	-	-	-	-
Signal line width	linewidth	x	-	-	-	-
Signal line thickness	linethickness	x	-	-	-	-
Signal line height (h)	lineheight	x	-	-	-	-
Signal line spacing	linespace	x	-	-	-	-
Gnd Plane thickness	gndthickness	x	-	-	-	-
Ground plane conductivity	gndsigma	x	-	-	-	-
Signal line conductivity	linesigma	x	-	-	-	-
Charecteristic impedance	z0	x	-	-	-	-
Delay Time	tdmt	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Frequency	`fmt`	x	-	-	-	-
Normalized length	`nlmt`	x	-	-	-	-
Propagation velocity normalized	`velmt`	x	-	-	-	-
Corner frequency	`corner`	x	-	-	-	-
DC series res/ Length	`dcr`	x	-	-	-	-
Loss resistance per unit length	`seriesr`	x	-	-	-	-
Conductor loss at fc	`alphac`	x	-	-	-	-
Conductor loss quality factor	`qc`	x	-	-	-	-
Dielectric loss frequency	`fd`	x	-	-	-	-
Loss conductance per unit length	`shuntg`	x	-	-	-	-
Dielectric loss	`alphad`	x	-	-	-	-
Dielectric loss quality factor	`qd`	x	-	-	-	-
Conductor loss frequency	`fcmt`	x	-	-	-	-

Example

For adding a mtline with a sub-circuit, specify the following parameters:



The netlist for an example of mtline with a sub-circuit:

```
I0 (net15 net16 net039 net040 net14) tline2
include "./w_subckt/tline2.scs"
```

The netlist for an example of mtline without a sub-circuit, with n=10, and entermatrices=nil:

```
I1 (net11 net12 net031 net032 net033 net034 net9 net10) mtline len=1.000m \
m=1 file="./wo_subckt/w_line.dat" freqscale=2 fmax=100
```

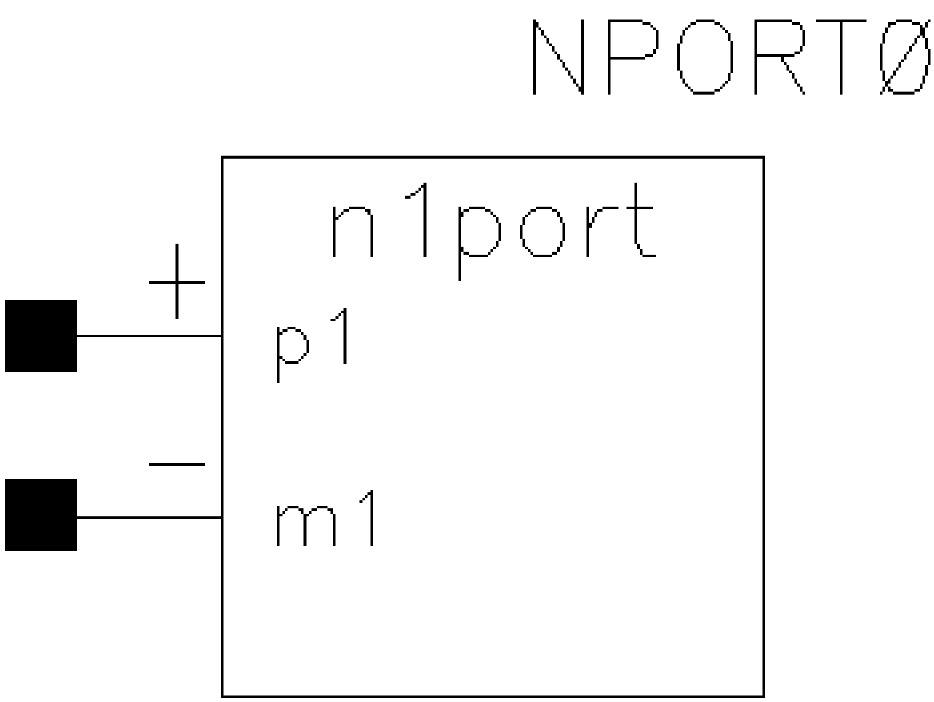
The netlist for the example of mtline without a sub-circuit, n=3, and entermatrices=t.

```
I2 (net7 net8 net023 net024 net025 net026 net5 net6) mtline len=5.000m m=1 \
r=[1K 1K 0 1K 0 1K] l=[418e-9 125e-9 418e-9 125e-9 125e-9 418e-9] \
g=[23e-6 34e-6 4e-6 3e-6 6e-6 1e-6] c=[94e-12 -22e-12 94e-12 \
-22e-12 -22e-12 94e-12] rskin=[3 4 1 1 1 1] gdloss=[1 2 3 1 1 1] \
file="./w_subckt/w_line.dat" freqscale=4 fmax=200
```

Additional Information

This device is not supported within the altergroups. The *Edit Object Properties* or *CDF Parameters* of the component `mtline` in analogLib are dynamic and subject to change based on the usage of IC Spectre or MMSIM Spectre installation.

Symbol: nport

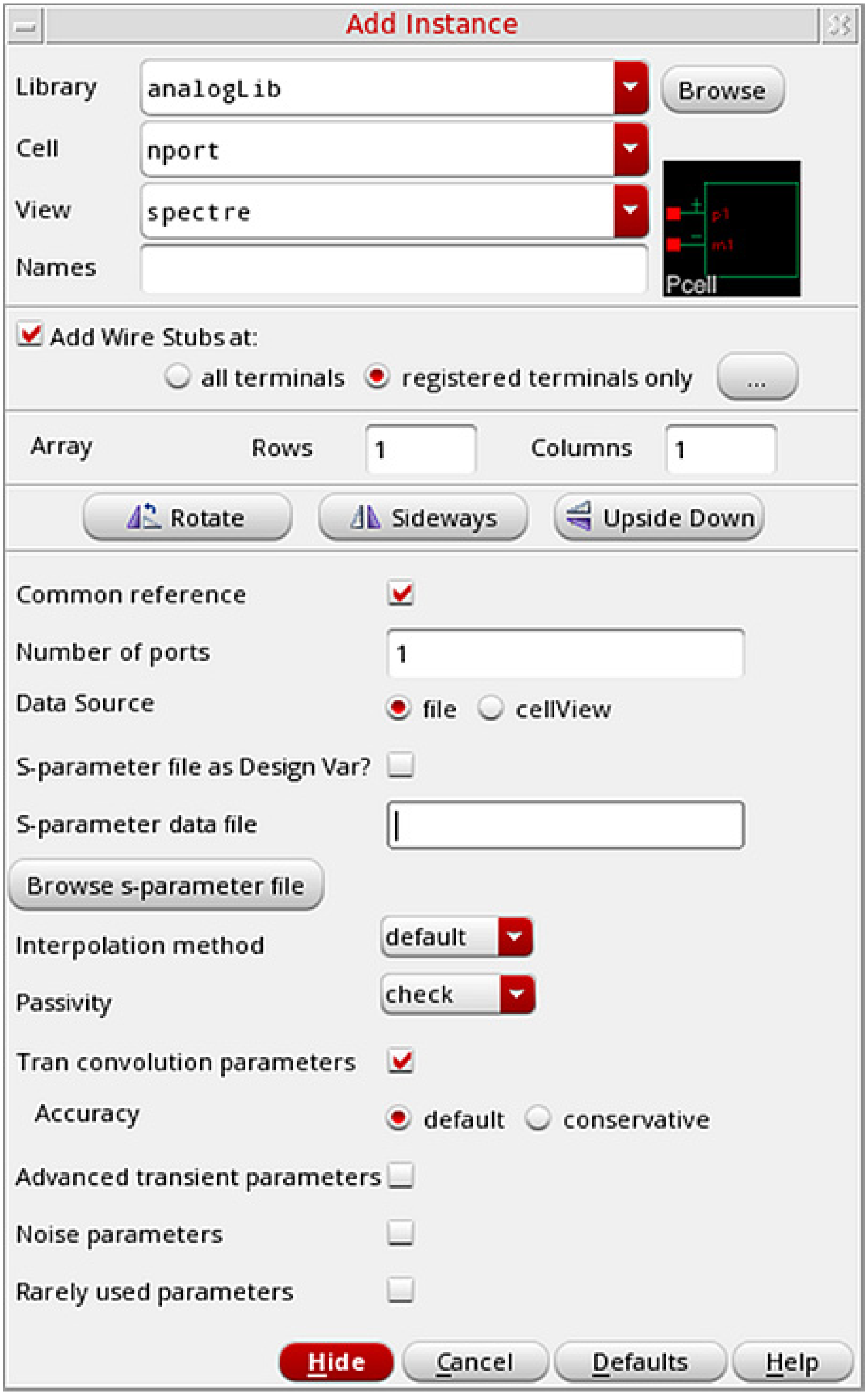


Linear N Port

An `N-port` takes its characteristics from an S-Parameter data file. An N-port can have as many ports as there are in the `N-port` described in the S-Parameter data file. Each pair of terminals in the `nport` instance statement represents one port. Because there is no limit to the number of ports, there is no limit to the number of terminals. However, the terminals must be given in pairs and there must be at least one pair. The order of the pairs is the same as the order of the ports in the data file.

Important

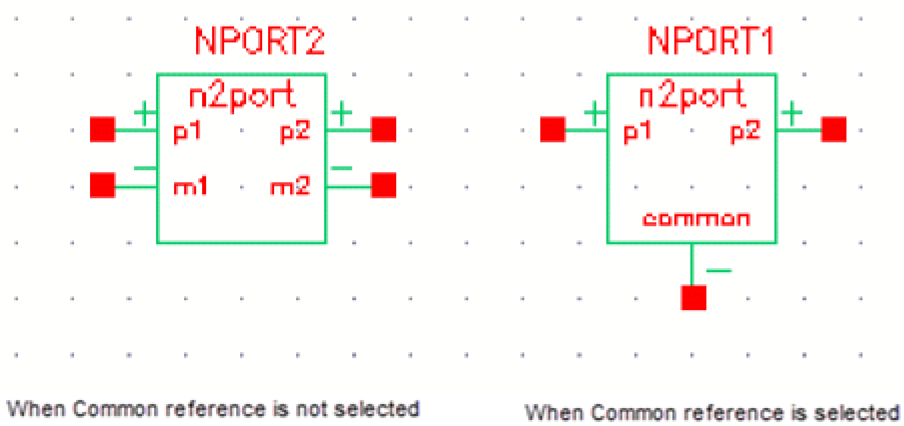
When using Spectre, we strongly recommended that you use `nport` instead of the deprecated `n1port`, `n2port`, `n3port`, and `n4port`, as these devices are retained strictly for legacy purposes and for supporting third-party simulators, such as Hspice.


Add Instance
Library
analogLib
Browse
Cell
nport
View
spectre
Names
Pcell
Add Wire Stubs at:
all terminals
registered terminals only
...
Array
Rows
1
Columns
1
Rotate
Sideways
Upside Down
Common reference
Number of ports
1
Data Source
file
cellView
S-parameter file as Design Var?
S-parameter data file
Browse s-parameter file
Interpolation method
default
Passivity
check
Tran convolution parameters
Accuracy
default
conservative
Advanced transient parameters
Noise parameters
Rarely used parameters
Hide
Cancel
Defaults
Help

Parameters for the nport component

This section describes the following CDF parameters for the `nport` component:

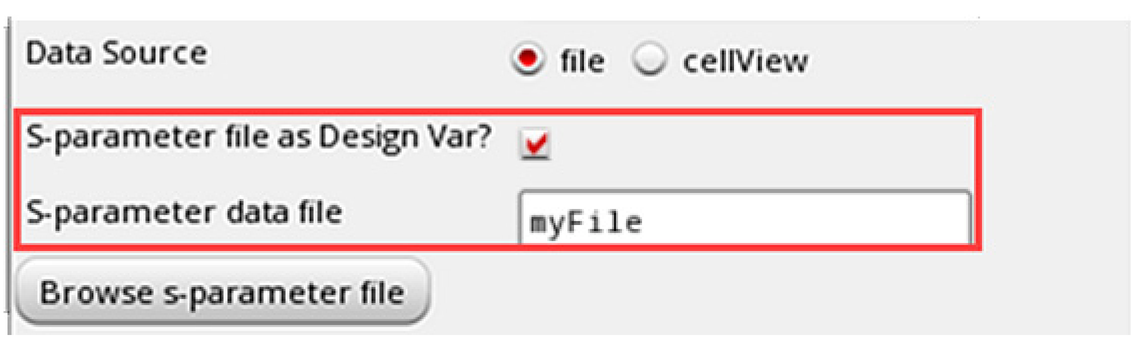
Common reference, when deselected, shows the plus and minus pins for all the individual ports. If you select the check box, the symbol redraws with a single common ground reference pin at the bottom of the symbol. This eliminates the need to add ground connections to each port of the symbol.



Number of ports must be set to the number of ports specified in the S-Parameter data file. This field controls the number of ports shown on the `nport` symbol.

Data Source lets you specify the source of the S-Parameter data file for the `nport` symbol. Possible options are *file* and *cellView*.

- ***file*** lets you specify the S-Parameter file as a design variable or the name and path to the S-Parameter file. Selecting this option displays the *S-parameter file as Design Var?* check box and the *S-parameter data file* field.



- ***cellview*** lets you specify the library, cell, and view of the S-Parameter file. Selecting this option displays the *S-parameter Library*, *S-parameter Cell*, and *S-parameter View* fields and the *Browse* button. You can enter details in the fields or click *Browse* to select the S-Parameter cellview using the Choose s-parameter cellView form.

The `nport` instance is printed in the netlist with the S-Parameter text cellview path. For example:

```
NPORT0 ( net1 net1) nport \
file="/scratch/ade_data/<user>/EAV_RAK/libs/Two_Stage_Opamp/OpAmp_AC_top/
text/text.txt"
```

S-parameter file as Design Var? allows you to specify an S-Parameter file as a design variable when you select the *S-parameter file as Design Var?* check box in the Add Instance or the Edit Object Properties form.

You must add this S-Parameter file as a design variable in an ADE Explorer cellview, which uses the same schematic as the `nport`. After running a simulation with the new design variable, you can view the netlist file, and further change the *S-parameter data file* for different corners. For more information, see *Edit Object Properties – Instance and Block* in the *Virtuoso Schematic Editor User Guide*.

S-parameter data file allows you to specify the name of the S-Parameter data file. This file contains parameters, frequencies, or model information that can be analyzed by the Spectre simulator. This field is shown only when you select the *file* option for *Data Source*.

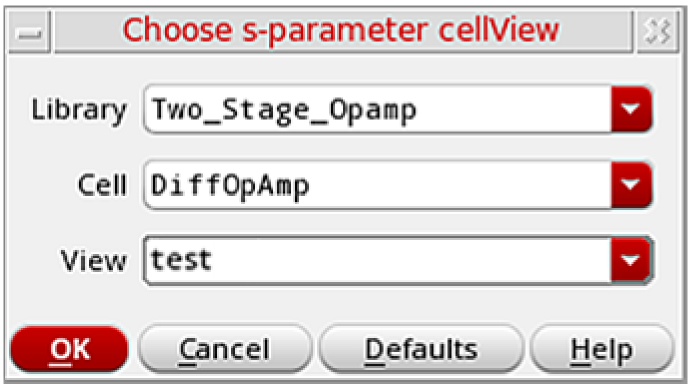
Browse s-parameter file, when clicked, lets you browse to a location and specify the S-Parameter data file. This check box is shown only when you select the *file* option for *Data Source* and when you do not select the *S-parameter file as Design Var?* check box.

S-parameter Library allows you to specify the name of the library that contains the S-Parameter cellview. This field is shown only when you select the *cellview* option for *Data Source*.

S-parameter Cell allows you to specify the name of cell that contains the S-Parameter view. This field is shown only when you select the *cellview* option for *Data Source*.

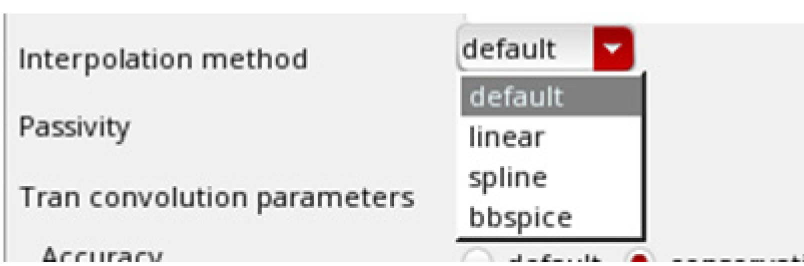
S-parameter View allows you to specify the name of the S-Parameter view. This field is shown only when you select the *cellview* option for *Data Source*.

Browse, when clicked, displays the Choose S-Parameter cellView form. This form lets you browse to a location and specify the S-Parameter cellview. The *Browse* button is shown only when you select the *cellview* option for *Data Source*.



Interpolation method controls the interpolation method for S-Parameter data and is valid only for `datafmt` = `spectre/touchstone/citi/bnp`. The supported methods are *default*, *linear*, *spline*, and *bbspice*. In general, the recommended method is *default*.

Linear and *spline* control the sampling of the S-Parameter data for the convolution-based method. In both methods, the S-Parameter data is sampled using a linear frequency spacing from zero to three times the highest frequency in the S-Parameter data file in order to calculate the impulse response of the transfer function.



- When you choose the ***default*** interpolation method, Spectre uses the default for `interp` according to the global option `nport_default_interp=auto_switch`.

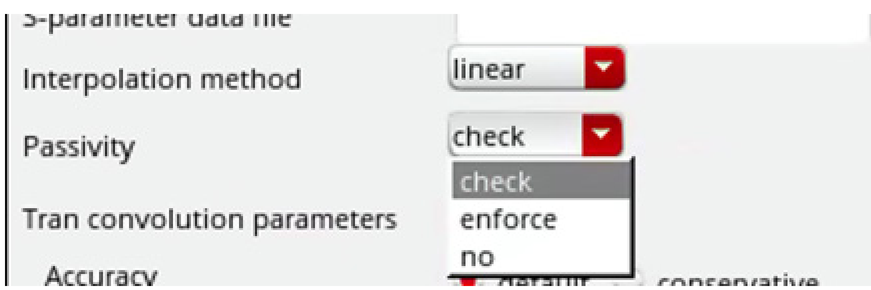
  If `nport_default_interp` is set to `auto_switch`, `nport` automatically switches the interpolation method based on the analysis. It chooses *bbspice* for pss shooting Newton analysis, and *linear* for analyses, such as `ac`, `dc`, and `sp`. See `spectre -h nport` for information on how `nport_default_interp` works for your particular version of Spectre.

All `nport` elements in the netlist that do not have `interp` set will have `interp` set to the value specified in the global option `nport_default_interp`. If an `nport` instance has the `interp` option explicitly specified, the instance option takes priority over the global option. Possible values for `nport_default_interp` are `spline`, `linear`, `bbspice`, and `auto_switch`. For more information, see Interpolation Method.

- When ***linear*** is selected as the interpolation method, linear interpolation is used to get a data point needed in the sample that is not directly in the S-Parameter file.
- ***Spline*** uses a cubic spline algorithm. Cubic spline can occasionally introduce errors when there are rapid changes in the transfer functions defined in the S-Parameter file near the sample point.
- ***Bbspice*** is used to do the rational fit. Bbspice uses a rational model to represent the S-Parameter data.

Passivity checks and enforces the passivity of S-Parameters. Spectre always checks to determine if the S-Parameter data is passive. Due to poor measurement accuracy, the S-Parameter data may be non-passive. Non-passive S-Parameter data may lead to non-converging or even unstable time domain simulations. The *Passivity* option controls detection and enforcement of S-Parameter simulation model passivity.

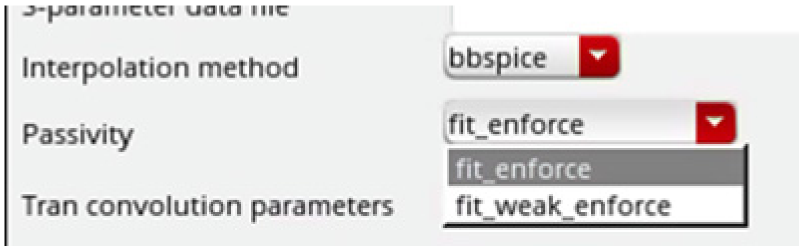
- For `interp` options of `linear` and `spline`, *Passivity* may be set to *no*, *check* or *enforce*.



- ***check*** is default, where the simulation will only check the S-Parameter data at each frequency point and report non-passive data.
- ***enforce*** ensures that the simulator correct the data to ensure passivity if the original data is non-passive.

Note: Passivity options of *no*, *check* and *enforce* are interpreted as passivity=fit_enforce for interp=bbspice.

- For interp=bbspice, *Passivity* may be set to *fit_weak_enforce* or *fit_enforce*. The default value is *fit_enforce*.



  - With passivity=fit_enforce, the simulator always attempts to create a passive simulation model and favors passivity over accuracy.
  - With passivity=fit_weak_enforce, the simulator does not create a passive model if the passivity enforcement phase results in significant accuracy loss.

Tran convolution parameters controls the accuracy parameter for transient convolution. When you select this check box, the *Accuracy* option is displayed.

- ***Accuracy*** lets you set the accuracy to *default* or *conservative*.
  - ***conservative*** as the *Accuracy* default provides exceptional accuracy. This causes more frequency-domain sample points, and produces a more accurate impulse response at the cost of runtime.
  - ***default*** accuracy is set tight enough that *conservative* should only be needed in rare instances.

Advanced transient parameters controls the maximum sampling points, frequency and the impulse response truncation. Selecting the check box displays the corresponding options.

Caution

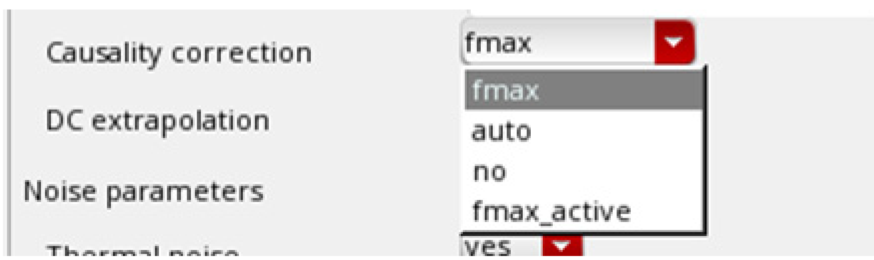
These properties are provided for S-Parameter simulation experts only and apply to linear or spline interpolation. It is strongly recommended that you do not change the default values of these properties.

- ***Max sampling points*** defines the maximum number of frequency points to be sampled in the adaptive algorithm. The default is 131072. In every case, encountered so far, the actual number of samples taken by the adaptive algorithm is much smaller than the default. In extremely unusual cases, it can be raised to 262144.

- ***Max frequency of interest*** (`fmax`) controls the highest frequency for the frequency domain sampling of the S-Parameter file. The default is three times the highest frequency in the S-Parameter file. This property should not be changed.

- ***Impulse response truncation*** is used to deliberately cut off the tail of the impulse response which might theoretically continue to infinite time. Leave this property at the default of `1e-4`, which corresponds to a gain of -`80 dB`.

- ***Causality correction*** list contains four choices: *no*, *fmax*, *auto*, or *fmax_active*. *fmax* is the default and is highly recommended. Causality enforcement is required in order to have reasonable results from an `nport` in either the DC or transient-based analyses. *Causality correction* is performed by setting the transfer function between the highest frequency in the S-Parameter file and three times this frequency so that the data becomes causal. The data within the frequencies specified in the S-Parameter file is unchanged. Setting `causality` to `fmax` or `auto` overrides the setting of the *High freq extrapolation* property.



  - ***no*** does not add a causality check.

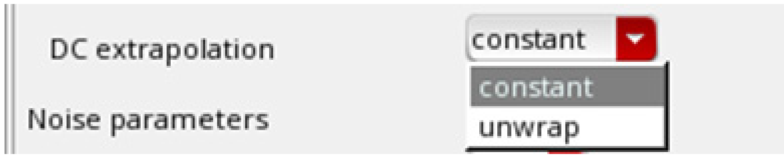
    Note: Setting the *Causality correction* check to *no* is incredibly risky unless you are absolutely sure that the S-Parameter file is causal as described.

  - ***fmax*** retains the data in the frequency range of the S-Parameter file, and then adds a transfer function above the frequency range in the S-Parameter file to force the system to be causal. This transfer function extends to the setting of Max frequency of interest, which defaults to three times the highest frequency in the S-Parameter file. If you suspect that the maximum frequency of interest needs to be changed, use causality *Auto* instead, if you are not an expert.

  - ***auto*** applies the causality correction in a similar manner to choosing fmax. *auto* can also vary the maximum frequency of interest if it needs to get a causal time-domain model.

  - ***fmax_active*** enhances causality correction for active devices to improve the simulation accuracy. This option is only available for *linear* interpolation (`interp=linear`). *bbspice* (interp=bbspice) should never be used when the S-Parameter file represents an active device because *bbspice* enforces *passivity*.

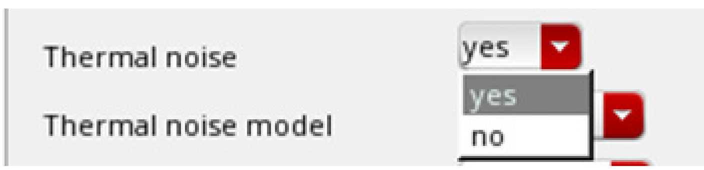
- ***DC extrapolation*** can be set to constant or unwrap. The default is *constant*.



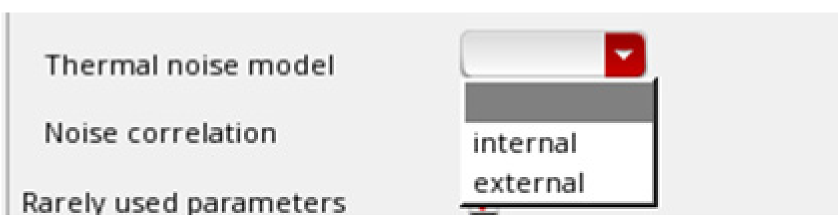
  - ***constant*** projects the first point down to zero frequency at exactly the same level.
  - ***unwrap*** does an estimation based on the first few frequency points in the S-Parameter file.

Noise parameters controls the `nport` noise parameters. Selecting the check box displays the corresponding options.

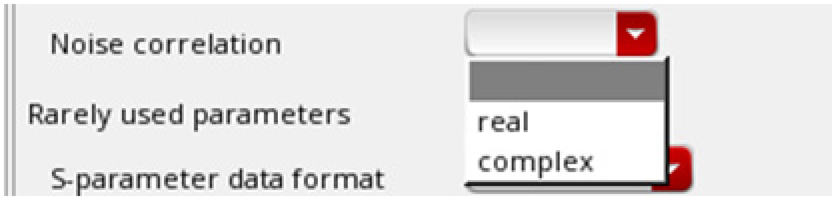
- ***Thermal noise*** lets you specify if `nport` should generate noise. Possible values are *no* and *yes*. Thermal noise defaults to *yes*. Set the value to *no* if you want to disable noise production.



- ***Thermal noise model*** defaults to external, which reads the noise parameters in the S-Parameter file if it is available, and if not, it uses an internal noise model. Internal forces the *internal* noise model.



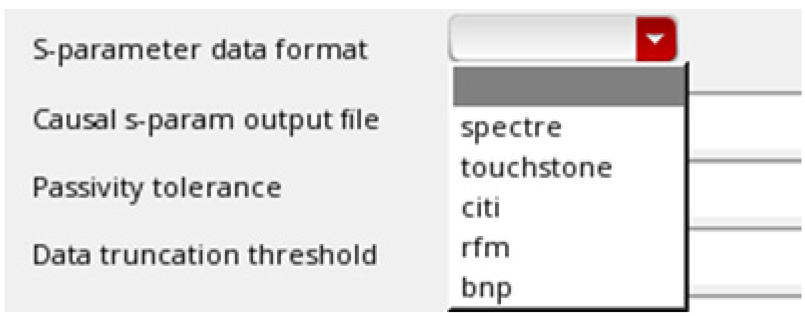
- ***Noise correlation*** set to *real* forces the `nport` noise correlation matrix to be real-valued. The parameter is used for backward compatibility only. Its value is determined automatically and its use is not recommended because it can lead to an incorrect answer. The simulator will generate a warning if the noise correlation matrix is complex while the value of `noisecorr` is set to *real*. Possible values are *real* and *complex*.



Rarely used parameters controls if you want to use the related parameters for `nport`. Selecting the check box displays the corresponding options.

Note: Only two parameters are commonly used—*Causal s-param output file*—*Additional parameter list*.

- ***S-parameter data format*** controls the format of the S-Parameter data file. If this parameter is not specified, Spectre detects the format by itself. Possible values are *spectre*, *touchstone*, *citi*, *rfm* and *bnp*.

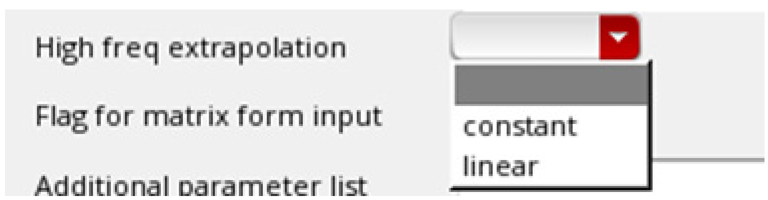


- ***Causal s-param output file*** contains a filename beginning with a slash (/). Specifying a filename in this field causes the S-Parameter data after causality correction to be placed in the specified file. This file will be created the first time the causal impulse response is calculated. This file can then be plotted directly in the waveform display tool. Most of the time, the causal impulse response calculated matches the original data provided in the S-Parameter file up to the maximum frequency provided in the S-Parameter file. To make the time-domain model causal, data is added based on the *fmax* option to make the model causal.

- ***Passivity tolerance*** is only used when the *Passivity* property is set to check or enforce. *Passivity tolerance* does not need to be set. The default is `1e-6`. This defines how close to unity gain should be modified by the passivity check and enforcement. *Passivity* will be enforced and/or reported when the gain is (1 - Passivity tolerance) or greater.

- ***Data truncation threshold*** defaults to `1e-3`, which corresponds to `-60dB` gain. When the cross coupling terms become smaller than the *Data truncation threshold*, they are ignored. Cross coupling is the coupling from one port to another port.
- ***Frequency sampling interval*** sets the delta frequency for the sampling from zero to the maximum frequency of interest. Leave this property at the default value. With adaptive sampling, this should never be necessary. If used, this delta should be a power of two divisor of the maximum frequency of interest.
- ***Multiplier*** specifies how many `nport` devices to put in parallel. This is rarely used.
- ***Scale factor*** scales the frequency of the S-Parameter file. For example, many S-Parameter files have the frequency in GHz. In this case, set the *Scale factor* to `1e9`.
- ***High freq extrapolation*** is ignored when causality correction is applied. The *High freq extrapolation* field can be set to constant or linear.

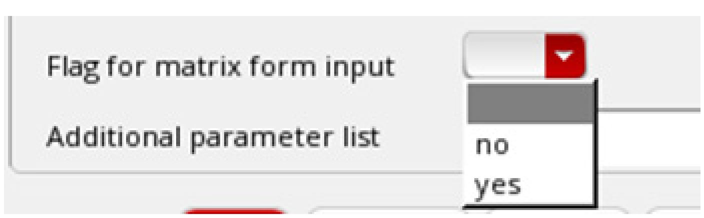
  Note: This property should not be used.



  - ***constant*** maintains the same amplitude and phase as the last point in the S-Parameter file to infinite frequency, if the causality check is not run.
  - ***linear*** keeps the amplitude constant at the last frequency point, but the phase increases linearly with frequency.

- ***Flag for matrix form input*** should not be set. In the past, each time the simulation ran, the impulse response was calculated for every port of every instance of the `nport` every time the simulation was started. In some cases, especially with a large number of ports, this could take considerable time. This flag was provided so the step of calculating the impulse response could be skipped. Since the impulse response is cached and available for re-use at any time, this property should never be needed.

  Note: This is a deprecated parameter that should not be set.



- ***Prioritize Accuracy Range*** is used to specify a frequency band of interest to prioritize the accuracy of *bbspice* fitting at this band. The parameter takes a vector where the first element is the start frequency point and the second element is the end frequency point.

  The frequency band is printed in the netlist as `bbsfreqband=[...]`. For example:

  ```
  NPORT1 (net3 net4) nport bbsfreqband=[2G 3G]
  ```

  Here, the start frequency is `2G` and the end freqency is `3G`.

- ***Additional parameter list*** is typically used to unlock new features. When this feature is used, a warning message is issued. This warning can be ignored.

Caution

It is strongly recommended that you only set the following properties on the Edit Object Properties or Add instance form for `nport`***:***

- ❑ The Number of ports in the S-Parameter file
- ❑ The S-Parameter data file name.
- ❑ The Common reference terminal.
- ❑ Interpolation methods - *default*, *linear*, *spline* or *bbspice*. The *default* interpolation method is typically recommended.

Caution

It is strongly recommended that you leave the following parameters set to their default values for nearly all applications:

- ❑ Tran convolution parameters
- ❑ Advanced transient parameters
- ❑ Noise parameters
- ❑ Rarely used parameters

Command-line help

```
spectre -h nport
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Common Reference	`nmode`	x	-	-	x	-
Number of Ports	`p`	x	-	-	x	-
S-parameter file as Design Var?	`sparam_file_as_var`	x	-	-	-	-
Browse and select s-data file	`nportFileB`	x	-	-	x	-
Causality correction	`causality`	x	-	-	-	-
Flag for matrix form input	`matrixform`	x	-	-	-	-
Matrix entry data file	`matrixfile`	x	-	-	-	-
Type of Port 1 to Type of Port20	`porttype1` to `porttype20`	x	-	-	-	-
Quantity of Port1 to Quantity of Port20	`portquantity1` to `portquantity20`	x	-	-	-	-
Multiplier	`m`	x	-	-	x	-
Scale Factor	`scale`	x	-	-	-	-
Interpolation Method	`interp`	x	-	-	-	-
Max frequency of interest	`fmax`	x	-	-	x	-
Frequency sampling interval	`fdelta`	x	-	-	-	-
Max order impulse response	`maxn`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Impulse response truncation	`imptrunc`	x	-	-	-	-
Noise parameters	`noiseParaLabel`	x	-	-	x	-
Rarely used parameters	`otherParaLabel`	x	-	-	x	-
Data truncation threshold	`datatrunc`	x	-	-	-	-
Thermal Noise	`thermalnoise`	x	-	-	-	-
Use Smooth Data Windowing	`usewindow`	x	-	-	-	-
S-parameter data format	`datafmt`	x	-	-	-	-
Thermal noise model	`noisemodel`	x	-	-	-	-
S-parameter Data File	`dataFile`	x	-	-	x	-
Noise correlation matrix	`noisecorr`	x	-	-	-	-
DC extrapolation	`dcextrap`	x	-	-	-	-
High Frequency Extrapolation	`hfextrap`	x	-	-	-	-
Passivity	`passivity`	x	-	-	-	-
Passivity	`passivity_bbspipce`	x	-	-	-	-
Passivity Tolerance	`pabstol`	x	-	-	-	-
Tran convolution parameters	`tranParaLabel`	x	-	-	-	-
Advanced transient parameters	`tranAdvanParaLabel`	x	-	-	x	-
Accuracy	`accuracyMode`	x	-	-	-	-
Causal s-param output file	`outFile`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Additional Parameter List	additionalParam	x	-	-	-	-
Prioritize Accuracy Range	prioritizeAccuracy Range	x	-	-	-	-
Start Frequencyt	startFrequency	x	-	-	-	-
End Frequencyt	endFrequency	x	-	-	-	-
Model name	hmname	-	-	-	x	-
Enable mixed mode	mixedmode	-	-	-	x	-
The order of indices	datatype	-	-	-	x	-
Characteristic impedance	zo	-	-	-	x	-
Hspice S-parameter data format	datafmtHspice	-	-	-	x	-
Hspice Interpolation method	interpolation	-	-	-	x	-
Enable passive checker	passive	-	-	-	x	-
Delay frequency	delayfreq	-	-	-	x	-
Extracts a system delay	delayhandle	-	-	-	x	-
Temperature difference	dtemp	-	-	-	x	-
High freq extrapolate method	highpass	-	-	-	x	-
Linear interpolation data type	intdattyp	-	-	-	x	-
Low freq extrapolate method	lowpass	-	-	-	x	-
Enable noise passive checker	noipassivechk	-	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Precondition factor keyword	`precfac`	-	-	-	x	-
Enable rational function	`rational_func`	-	-	-	x	-
Reuse rational function data	`rational_func_reuse`	-	-	-	x	-
Method of smooth	`smooth`	-	-	-	x	-
Width of the smoothing window	`smoothpts`	-	-	-	x	-
Stamping method	`stamp`	-	-	-	x	-

Syntax/Synopsis

```
model ndata nport file="sparam.data" scale=1
```

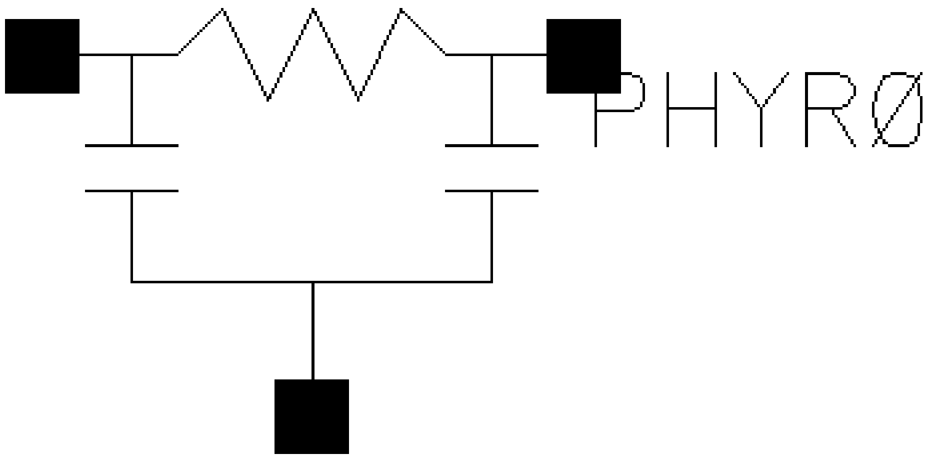
Example

```
x1 (a1 0 b1 0 b3 0) ndata file="sparam 2.data"
```

Additional Information

This device is not supported within the altergroups.

Symbol: phyres



Physical Resistor

It consists of a two terminal resistor (tied between ‘t1’ and ‘t2’) and two diodes (tied between ‘t1’-‘t0’ and ‘t2’-‘t0’). The diodes are junction diodes. Under normal operation, the two diodes are reverse biased, but the parameter ‘subtype’ can reverse the direction of the diodes. If you do not specify ‘t0’, ground is assumed. The instance parameters always override model parameters. If you do not specify the instance resistance value, it is calculated from the model parameters.

Command-line help

```
spectre -h phy_res
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	-	-	-	-
Resistance	`r`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Capacitance	`c`	x	-	-	-	-
Length	`l`	x	-	-	-	-
Width	`w`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Lin temp co of lin cap	`tc1c`	x	-	-	-	-
Quad temp co of lin cap	`tc2c`	x	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-

Syntax/Synopsis

```
Name ( 1 2 [0] ) ModelName <parameter=value> ...
```

Following is the model synopsis:

```
model ModelName phy_res <parameter=value> ...
```

Example

Following is the sample instance statement:

```
res1 (net9 vcc) resphy l=1e-3 w=2e-6
```

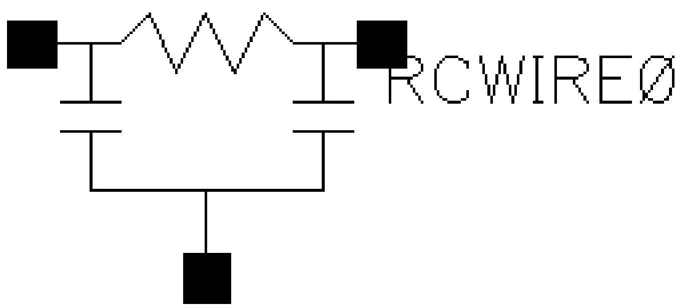
Following is the sample model statement:

```
model resphy phy_res rsh=85 tc1=1.53e-3 tc2=4.67e-7 etch=0 cj=1.33e-3
cjsw=3.15e-10 tc1c=9.26e-4
```

Additional Information

This device is supported within the altergroups.

Symbol: rcwireload



RC Wire Load

A wire model of a two terminal resistor with an optional third terminal at the instance level. If the third terminal is not specified then the two-terminal resistance model is used with the third terminal as ground.

In RC wire load model, R represents the interconnect metal or poly resistance and C represents substrate capacitance from node to ground.

You can specify the capacitance explicitly or allow it to be computed from the physical length and width of the resistor. The model parameter `cratio` can be used to allocate the parasitic capacitance of the wire element between the model's input capacitor and the output capacitor. The value of each capacitor, as a function of temperature, is represented as linear temperature coefficient of capacitor (tc1c) and quadratic temperature coefficient of capacitor (tc2c).

For details refer to `spectre` help.

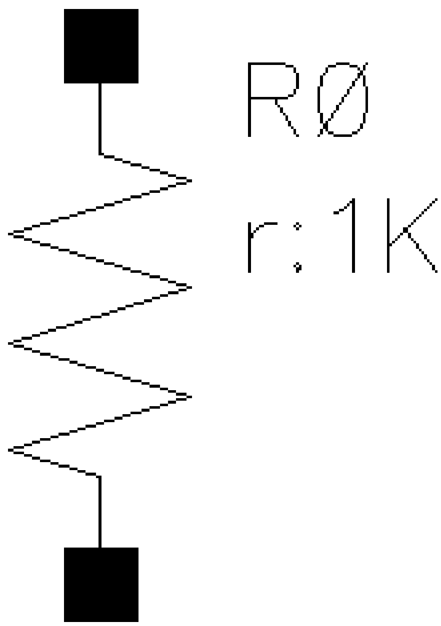
Command-line help

```
spectre -h resistor
```

Component Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	x	x	x	x	-
Resistance	`r`	x	x	x	x	-
Length	`l`	x	-	-	-	-
Width	`w`	x	-	-	-	-
Resistance Form	`resform`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Scale factor	`scale`	x	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Generate noise?	`isnoisy`	x	-	-	-	-
Capacitance	`c`	x	-	-	-	-
Lin temp co of lin cap	`tc1c`	x	-	-	-	-
Quad temp co of lin cap	`tc2c`	x	-	-	-	-

Symbol: res



Two Terminal Resistor

You can give the resistance explicitly or allow it to be computed from the physical length and width of the resistor. In either case, the resistance can be a function of temperature or applied voltage.

If R(inst) is not given, R(inst) = R(model), if R(model) is given, then R(inst) = Rsh * (L - 2 * etchl) / (W - 2 * etch).

If the polynomial coefficients vector (‘coeffs=[c1 c2 ...]’) is specified, the resistor is nonlinear. When ‘nonlinform’ is set to ‘g’, the resistance is:

```
R(V) = dV / dI
     = R(inst) / (1 + c1 * V + c2 * V^2 + ...).
```

Command-line help

```
spectre -h resistor
```

Component Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Resistance	`r`	x	x	x	x	x
Temperature coefficient 1	`tc1`	x	-	-	x	x
Temperature coefficient 2	`tc2`	x	-	-	x	x
Model name	`model`	x	-	-	-	-
Length	`l`	x	-	-	x	x
Width	`w`	x	-	-	x	x
Resistance Form	`resform`	x	-	-	-	-
Multiplier	`m`	x	-	-	x	x
Scale factor	`scale`	x	-	-	x	x
Temp rise from ambient	`trise`	x	-	-	-	-
Generate noise?	`isnoisy`	x	-	-	-	-
Capacitance connected	`hrc`	-	-	-	x	x
Temperature difference	`dtemp`	-	-	-	x	x
AC resistance	`ac`	-	-	-	x	-

Syntax/Synopsis

```
Name ( 1 2 ) ModelName <parameter=value> ...
Name ( 1 2 ) resistor <parameter=value> ...
```

Following is the model synopsis:

```
model ModelName resistor <parameter=value> ...
```

Example

Following is a sample instance statement without model:

```
r1 (1 2) resistor r=1.2K m=2
```

Following is a sample instance statement with model:

```
r1 (1 2) resmod l=8u w=1u
```

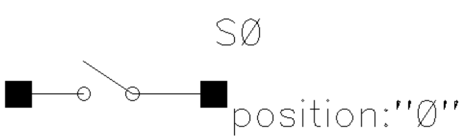
Following is a sample model statement:

```
model resmod resistor rsh=150 l=2u w=2u etch=0.05u tc1=0.1 tnom=27 kf=1
```

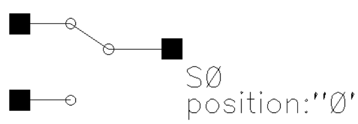
Additional Information

This device is supported within the altergroups.

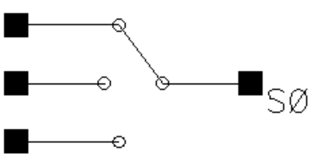
Symbol: spxtswitch



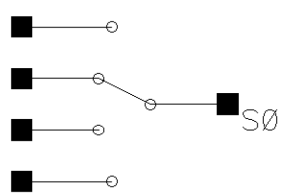
sp1tswitch



sp2tswitch



sp3tswitch



sp4tswitch

sp1tswitch - Ideal Switch With 1 Position

sp2tswitch - Ideal Switch With 2 Positions

sp3tswitch - Ideal Switch With 3 Positions sp4tswitch - Ideal Switch With 4 Positions

Ideal switch is a single-pole multiple-throw switch with infinite ‘off’ resistance and zero ‘on’ resistance. The switch is provided to allow you to reconfigure your circuit between analyses. You can only change the switch state between analyses (using the alter statement), not during an analysis.

Command-line help

```
spectre -h switch
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Switch position	`position`	x	-	-	-	-
DC position	`dcPosition`	x	-	-	-	-
AC position	`acPosition`	x	-	-	-	-
Tran position	`tranPosition`	x	-	-	-	-
IC position	`icPosition`	x	-	-	-	-
Offset voltage	`offset`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Parameter Type	`paramTyp`	x	-	-	-	-

Syntax/Synopsis

```
Name ( t0 t1 ... ) switch <parameter=value> ...
```

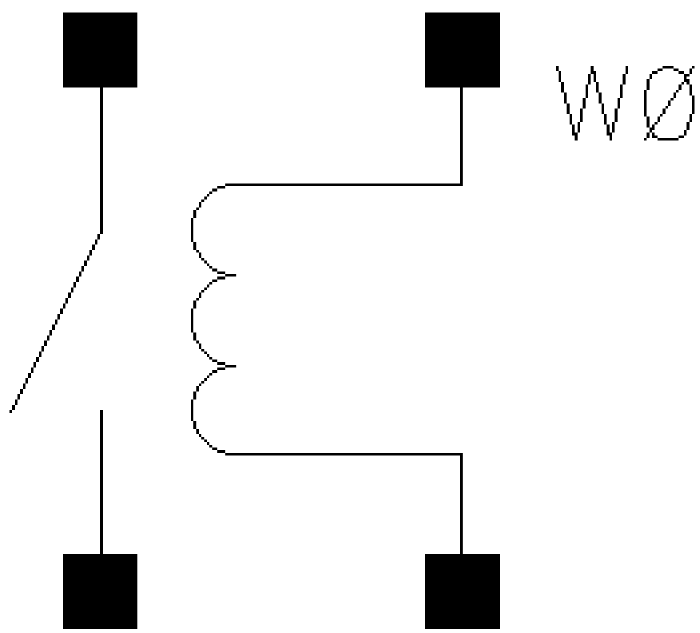
Example

```
sw1 (t1 t2 t3) switch dc_position=0 ac_position=1 tran_position=2
```

Additional Information

This device is not supported within the altergroups.

Symbol: switch



Four Terminal Relay

The four-terminal relay is a voltage controlled relay tied between terminals ‘t1’ and ‘t2’. The voltage between terminals ‘ps’ and ‘ns’ controls the relay resistance. The relay resistance varies nonlinearly between ‘ropen’ and ‘rclosed’, the open relay resistance and closed relay resistance, respectively. These resistance values correspond to control voltages of ‘vt1’ and ‘vt2’ respectively. The four parameters, ‘vt1’, ‘vt2’, ‘ropen’, and ‘rclosed’, can be instance or model parameters.

Command-line help

```
spectre -h relay
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Open/ close voltage	vsw	-	-	-	-	-
Delay time	td	-	-	-	-	-
Time interval for switching	ts	-	-	-	-	-
Open switch resistance	ro	x	-	-	-	-
Close switch resistance	rc	x	-	-	-	-
Open voltage	vt1	x	-	-	-	-
Closed voltage	vt2	x	-	-	-	-
Multiplier	m	x	-	-	-	-
Estimated operating region	region	x	-	-	-	-
Resistance	r	-	-	-	-	-

Syntax/Synopsis

```
Name ( 1 2 ps ns ) ModelName <parameter=value> ...
Name ( 1 2 ps ns ) relay <parameter=value> ...
```

Following is the model synopsis:

```
model ModelName relay <parameter=value> ...
```

Example

Following is a sample instance statement:

```
rel1 (1 2 ps ns) my_relay ropen=1G rclosed=2
```

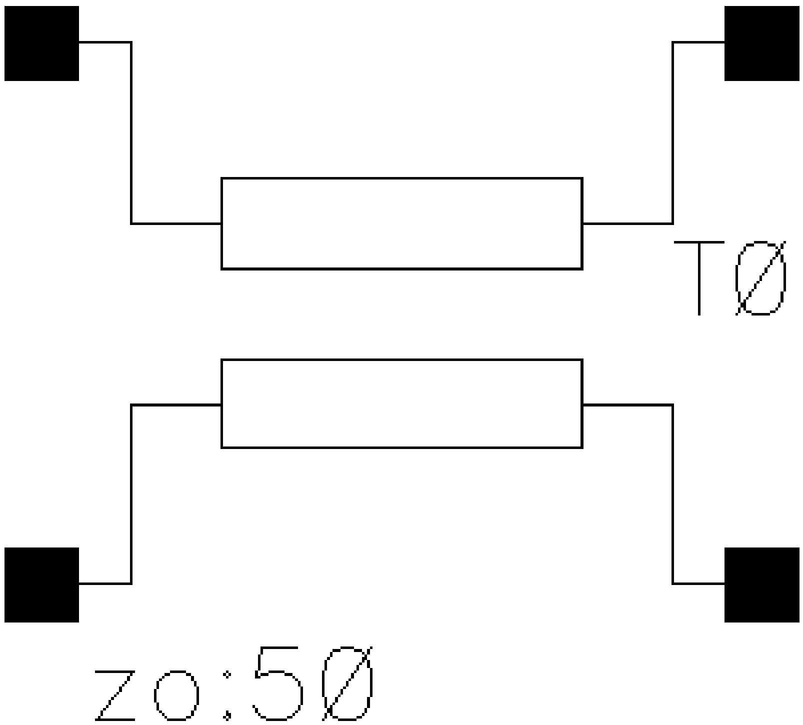
Following is a sample model statement:

```
model my_relay relay vt1=2.5 vt2=5 ropen=100M rclosed=0.1
```

Additional Information

This device is not supported within the altergroups.

Symbol: tline



Transmission Line (Lossy or Lossless)

The transmission line model includes dielectric and conductor loss effects. The conductor loss includes skin effect assuming finite or infinite conductor thickness.

Only the odd mode is modeled, so only the voltage difference across each port is important. The absolute voltage of each terminal is not significant. Also, the current into one node of a port equals the current leaving the other node of the port.

Command-line help

```
spectre -h tline
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Characteristic impedance	`zo`	x	-	-	x	x
Delay time	`td`	x	-	-	x	x
Frequency	`freq`	x	-	-	x	x
Normalized length	`nl`	x	-	-	x	x
Voltage 1	`v1`	-	-	-	x	x
Current 1	`i1`	-	-	-	x	x
Voltage 2	`v2`	-	-	-	x	x
Current 2	`i2`	-	-	-	x	x
Model name	`model`	x	-	-	-	-
Propogation velocity normalized	`vel`	x	-	-	-	x
Physical length	`len`	x	-	-	x	x
Multiplier	`m`	x	-	-	-	-
Loss resistance per unit length	`rs`	x	-	-	-	-
Loss conductance per unit length	`g`	x	-	-	-	-

Syntax/Synopsis

```
Name ( t1 b1 t2 b2 ) ModelName <parameter=value> ...
```

```
Name ( t1 b1 t2 b2 ) tline <parameter=value> ...
```

Following is the model synopsis:

```
model ModelName tline <parameter=value> ...
```

Example

Following is a sample instance statement:

```
t1 (1 0 2 0) lmodel z0=100
```

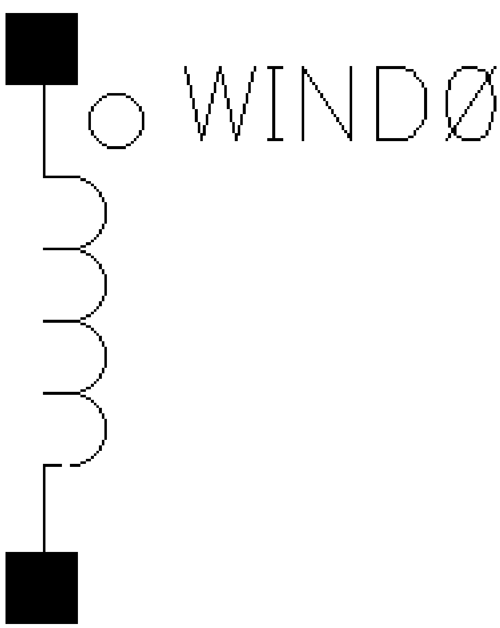
Following is a sample model statement:

```
model lmodel tline f=10M z0=50 alphac=8501 fc=10M dcr=88
```

Additional Information

This device is supported within the altergroups.

Symbol: winding



Winding for Magnetic Core

A winding is used in conjunction with magnetic cores to model coils and transformers with hysteresis. Each winding must be associated with a single core, though a core may have any number of windings.

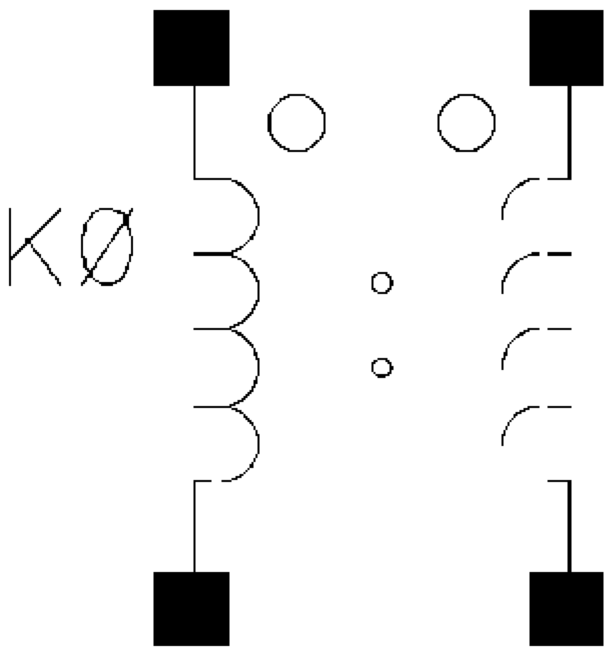
Command-line help

```
spectre -h winding
```

Component Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Name of core	`core`	x	-	-	-	-
Num of turns on winding	`turn`	x	-	-	x	-
Res of the winding	`resis`	x	-	-	x	-
Multiplier	`m`	x	-	-	-	-
Initial condition	`ic`	x	-	-	x	-
Resistance	`r`	-	-	-	-	-

Symbol: xfmr



Linear Two Winding Ideal Transformer

Winding 1 connects terminals ‘t1’ and ‘b1’, and winding 2 connects ‘t2’ and ‘b2’. The number of turns on windings 1 and 2 are given by ‘n1’ and ‘n2’ respectively, where ‘n2’ must not be zero. The absolute number of turns of each winding is not important, only the ratio of ‘n1’ to ‘n2’. Current through winding 1 is computed.

An ideal transformer is modeled, so it acts as a transformer at DC. In particular, it implements

```
v1/v2 = n1/n2
i1/i2 = -n2/n1
```

Command-line help

```
spectre -h transformer
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Coupling coefficient	`k`	-	-	-	x	x
Primary inductor	`pi`	-	-	-	x	x
Secondary inductor	`si`	-	-	-	x	x
Number of turns on primary	`n1`	x	-	-	-	-
Number of turns on secondary	`n2`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-

Syntax/Synopsis

```
Name ( t1 b1 t2 b2 ) transformer <parameter=value> ...
```

Example

```
tr1 (1 0 2 0) transformer n1=3 n2=3 m=2
```

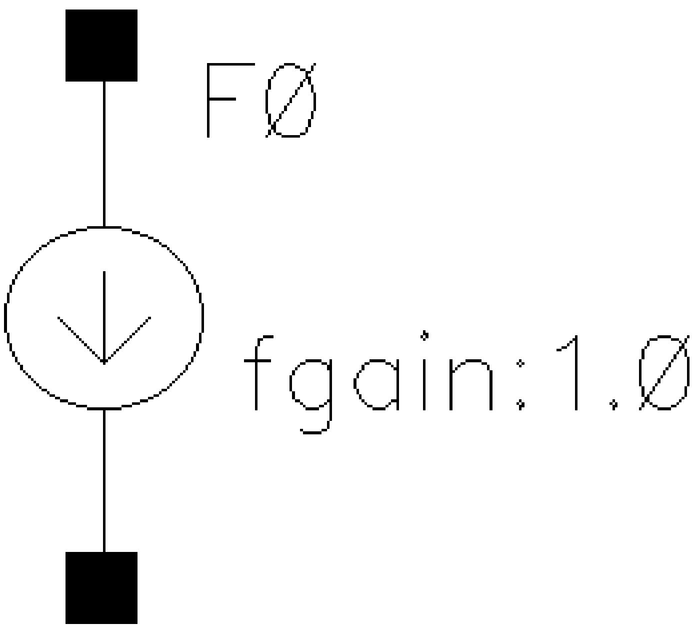
Additional Information

This device is not supported within the altergroups.

6

Sources - Dependent Components

Symbol: cccs



Linear Current Controlled Current Source

A current-controlled source detects the current with a probe device. A valid probe is a component instance in the circuit that naturally computes current. For example, probes can be voltage sources (independent or controlled), inductors, transmission lines, microstrip lines, N-ports, and transformers. If the probe device computes more than one current (such as transmission lines, microstrip lines, and N-ports), the index of the probe port through which the controlling current flows needs to be specified. Positive current exits the source node and enters the sink node of the controlled source.

Note: Component cccs uses the same values of parameters `fgain`, `maxm`, `minm`, `m` for both Spectre and hspiceD simulators.

Command-line help

```
spectre -h cccs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Name of voltage source	vref	x	-	-	x	x
Current gain	fgain	x	-	-	x	x
Initial condition	ic	-	-	-	-	x
Port	port	x	-	-	-	-
Type of transfer char	trfType	x	-	-	-	-
Multiplier	m	x	-	-	x	x
Type of Source	typesrc	x	-	-	-	-
Maximum Output Current	maxm	x	-	-	x	-
Minimum Output Current	minm	x	-	-	x	-
Absolute Output Current	absol	x	-	-	-	-
Smoothing Factor	smoothing	x	-	-	-	-
Type	csType	-	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Current gain (Obsolete)	hfgain	-	-	-	x	-
Maximum output current (Obsolete)	maxi	-	-	-	x	x
Minimum output current (Obsolete)	mini	-	-	-	x	x
Scale factor	scale	-	-	-	x	x
Multiplier (Obsolete)	hm	-	-	-	x	
Temperature coefficient 1	tc1	-	-	-	x	x
Temperature coefficient 2	tc2	-	-	-	x	x
Absolute value	habs	-	-	-	x	-
Initial condition	hic	-	-	-	x	-
Delta	delta	x	-	-	x	x
Number of controlling pairs	xypairs	-	-	-	x	x
Delay Time	htd	-	-	-	x	-
Absolute value	abs	x	-	-	-	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Delay Time	`td`	-	-	-	-	x
Controlling Volt 1	`x1 - x20`	-	-	-	x	x

Syntax/Synopsis

```
Name ( sink src ) cccs <parameter=value> ...
```

Example

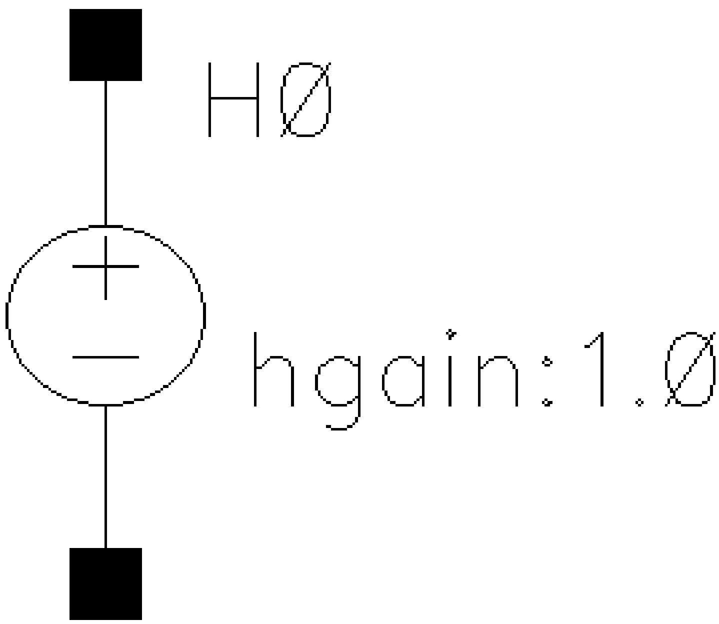
```
vcs (pos gnd) cccs gain=2.5 probe=v1 m=1
```

Note: v1 is an instance of a voltage source

Additional Information

This device is supported within the altergroups. This device can also model ideal digital gates.

Symbol: ccvs



Linear Current Controlled Voltage Source

A current-controlled source senses the current with a probe device. A valid probe is a component instance in the circuit that naturally computes current. For example, probes can be voltage sources (independent or controlled), inductors, transmission lines, microstrip lines, N-ports, and transformers.

If the probe device computes more than one current (such as transmission lines, microstrip lines, and N-ports), the index of the probe port through which the controlling current flows needs to be specified. Current through the controlled voltage source is calculated and is defined to be positive if it flows from the positive terminal, through the source, to the negative terminal.

Note: Component ccvs uses the same values of parameters `hgain`, `maxm`, and `minm` for both Spectre and hspiceD simulators.

Command-line help

```
spectre -h ccvs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Name of voltage source	vref	x	-	-	x	x
Transresistance	hgain	x	-	-	x	x
Initial condition	ic	-	-	-	-	x
Port	port	x	-	-	-	-
Type of transfer char	trfType	x	-	-	-	-
Multiplier	m	x	-	-	-	-
Type of Source	typesrc	x	-	-	-	-
Minimum Output Voltage	minm	x	-	-	x	-
Maximum Output Voltage	maxm	x	-	-	x	-
Absolute Output Voltage	absol	x	-	-	-	-
Smoothing Factor	smoothing	x	-	-	-	-
Type	csType	-	-	-	x	x
Transresistance (Obsolete)	hhgain	-	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Maximum output voltage (Obsolete)	maxv	-	-	-	x	x
Minimum output voltage (Obsolete)	minv	-	-	-	x	x
Scale factor	scale	-	-	-	x	x
Temperature coefficient 1	tc1	x	-	-	x	x
Temperature coefficient 2	tc2	x	-	-	x	x
Absolute value	habs	-	-	-	x	-
Initial condition	hic	-	-	-	x	-
Delta	delta	x	-	-	x	x
Number of controlling pairs	xypairs	-	-	-	x	x
Delay Time	htd	-	-	-	x	-
Absolute value	abs	x	-	-	-	x
Delay Time	td	-	-	-	-	x
Controlling Volt 1	x1 - x20	-	-	-	x	x

Syntax/Synopsis

```
Name ( p n ) ccvs <parameter=value> ...
```

Example

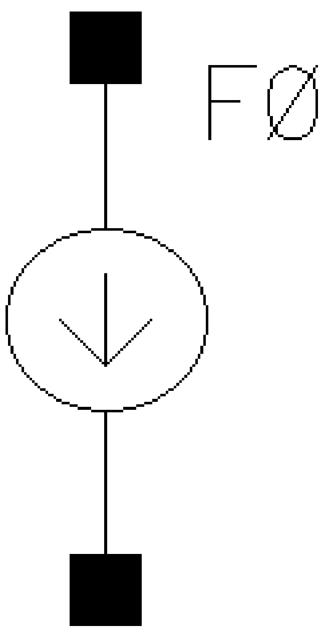
```
vvs (pos gnd) ccvs rm=1 probe=v1 m=1
```

Note: v1 is an instance of a voltage source

Additional Information

This device is supported within the altergroups. This device can also model ideal digital gates.

Symbol: pcccs



Polynomial Current Controlled Current Source

A vector of coefficients specifies the polynomial function that defines the relationship between the output current and the controlling currents. You must specify at least one coefficient.

For a polynomial in N variables a1, a2, ... an, the polynomial function F(a0,a1,...,an) is given by:

```
F = c0 + c1 * a1 + c2 * a2 + ...
+ c(m+1) * a1^2 + c(m+2) * a1 * a2 +...
+ c(2m+1) * a2^2 + c(2m+2) * a2 * a3 +...
```

where the ‘c’s are coefficients of the polynomial terms, and `m` is the multiplier.

Command-line help

```
spectre -h pcccs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Max Coefficient Number	polyCoef	x	-	-	-	-
Poly Coeff 0	c0	x	-	-	-	-
Poly Coeff 1	c1	x	-	-	-	-
Poly Coeff 2	c2	x	-	-	-	-
Poly Coeff 3	c3	x	-	-	-	-
Poly Coeff 4	c4	x	-	-	-	-
Number of Probes	probeCnt	x	-	-	-	-
Probe 1	p1	x	-	-	-	-
Port 1	port1	x	-	-	-	-
Probe 2	p2	x	-	-	-	-
Port 2	port2	x	-	-	-	-
Probe 3	p3	x	-	-	-	-
Port 3	port3	x	-	-	-	-
Probe 4	p4	x	-	-	-	-
Port 4	port4	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Gain	gain	x	-	-	-	-
Multiplier	m	x	-	-	-	-
Maximum Output Current	maxm	x	-	-	-	-
Minimum Output Current	minm	x	-	-	-	-
Absolute Output Current	absol	x	-	-	-	-
Smoothing Factor	smoothing	x	-	-	-	-
Temperature coefficient 1	tc1	x	-	-	-	-
Temperature coefficient 2	tc2	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ) pcccs <parameter=value> ...
```

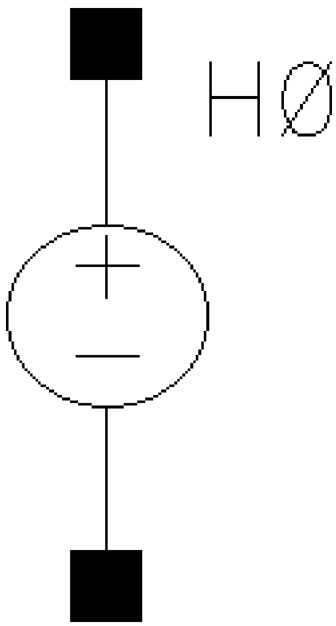
Example

```
vpc (net1 0) pcccs probes=[vb vc ve vlp vpn] coeffs=[0 8.8e6 -8.8e6 9e6 8e6 -9e6]
```

Additional Information

This device is supported within the altergroups.

Symbol: pccvs



Polynomial Current Controlled Voltage Source

The polynomial function defining the relationship between the output voltage and the controlling currents is specified by a vector of coefficients. At least one coefficient must always be specified. Current through the voltage source is calculated and is defined as positive if it flows from the positive terminal, through the source, to the negative terminal.

For a polynomial in N variables a1, a2, ... an, the polynomial function F(a0,a1,...,an) is given by:

```
F = c0 + c1 * a1 + c2 * a2 + ...
+ c(m+1) * a1^2 + c(m+2) * a1 * a2 +...
+ c(2m+1) * a2^2 + c(2m+2) * a2 * a3 +...
```

where the ‘c’s are coefficients of the polynomial terms, and `m` is the multiplier.

Command-line help

```
spectre -h pccvs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Max Coefficient Number	polyCoef	x	-	-	-	-
Poly Coeff 0	c0	x	-	-	-	-
Poly Coeff 1	c1	x	-	-	-	-
Poly Coeff 2	c2	x	-	-	-	-
Poly Coeff 3	c3	x	-	-	-	-
Poly Coeff 4	c4	x	-	-	-	-
Number of Probes	probeCnt	x	-	-	-	-
Probe 1	p1	x	-	-	-	-
Port 1	port1	x	-	-	-	-
Probe 2	p2	x	-	-	-	-
Port 2	port2	x	-	-	-	-
Probe 3	p3	x	-	-	-	-
Port 3	port3	x	-	-	-	-
Probe 4	p4	x	-	-	-	-
Port 4	port4	x	-	-	-	-
Gain	gain	x	-	-	-	-
Multiplier	m	x	-	-	-	-
Maximum Output Voltage	maxm	x	-	-	-	-
Minimum Output Voltage	minm	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Absolute Output Voltage	`absol`	x	-	-	-	-
Smoothing Factor	`smoothing`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) pccvs <parameter=value> ...
```

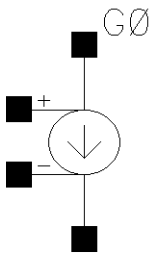
Example

```
ixy (net1 0) pccvs coeffs=[0 1 0 1] probes=[vin1 vin2] gain=2
```

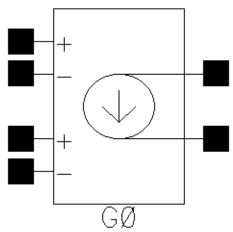
Additional Information

This device is supported within the altergroups.

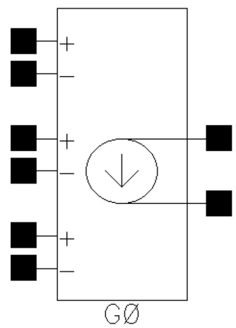
Symbol: pvccs, pvccs2, pvccs3



pvccs



pvccs2



pvccs3

Polynomial Voltage Controlled Current Source

A polynomial voltage controlled current source in which the symbol varies with the number of controlling voltages. A vector of coefficients specifies the polynomial function that defines the relationship between the output current and the controlling voltages. You must specify at least one coefficient. Current exits the source node and enters the sink node.

For a polynomial in M variables a1, a2, ... am, the polynomial function F(a0,a1,...,am) is given by:

```
F = c0 + c1 * a1 + c2 * a2 + ...
+ c(m+1) * a1^2 + c(m+2) * a1 * a2 +...
+ c(2m+1) * a2^2 + c(2m+2) * a2 * a3 +...
```

where the ‘c’s are coefficients of the polynomial terms, and `m` is the multiplier.

The coefficients should be given in the order of the polynomial terms. The order of the polynomial terms is:

1. Lower degree term goes before higher degree term. For example, `a1` is before `a1^2`.
2. For the same degree terms, the term whose first variable has higher degree goes first. If the first variable has the same degree, then check the second variable, and so on. For example, for terms in 3 variables and of 4 degrees, a `1^4` goes before `a1^3*a2`. And `a1^3*a2` goes before `a1^3*a3`.

If you have high degree terms, using `coeff` parameter may not be convenient. You can use a file to specify the nonzero coefficients. You use one line in your file to specify one coefficient. The format is to put the degree of the variables first, then the coefficient. For example, if you have term `1.5*a1*a2^2*a3`, the degrees of `a1`, `a2` and `a3` are `1 2 1`, the coefficient is `1.5`.

So the line in your file is:

```
1 2 1 1.5
```

Command-line help

```
spectre -h pvccs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Max Coefficient Number	`polyCoef`	x	-	-	x	-
Poly Coeff 0	`c0 - c20`	x	-	-	x	-
Initial condition	`ic`	-	-	-	x	-
Scale factor	`scale`	-	-	-	x	-
Absolute value	`abs`	x	-	-	x	-
Gain	`gain`	x	-	-	-	-
Multiplier	`m`	x	-	-	x	-
Maximum Output Current	`maxm`	x	-	-	x	-
Minimum Output Current	`minm`	x	-	-	x	-
Absolute Output Current	`absol`	x	-	-	-	-
Smoothing Factor	`smoothing`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Temperature coefficient 1	`tc1`	x	-	-	x	-
Temperature coefficient 2	`tc2`	x	-	-	x	-
File containing Poly Coeffs	`filecoef`	x	-	-	-	-
Coeffs to be specified in	`coefSpec`	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ps1 ns1 ... ) pvccs <parameter=value> ...
```

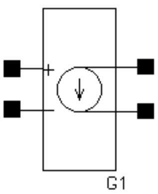
Example

```
v2 (net1 0 net2 0) pvccs coeffs=[0 -2e-3 - 10e-3] gain=2 m=1
```

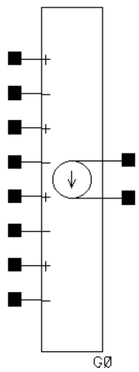
Additional Information

This device is supported within altergroups.

Symbol: pvccsp



pvccsp



pvccsp with 4
controlling voltages

Parameterized Cell Based Polynomial Nonlinear Voltage Controlled Current Source

pvccsp is a Pcell-based polynomial voltage controlled current source in which the symbol varies with the number of controlling voltages. pvccsp is similar to pvccs except that it has one additional parameter (nc) that specifies the number of controlling voltage sources.

Note: The maximum number of controlling voltages is 20. Therefore, if you specify a number greater than 20, the value of this parameter will default to 20.

Command-line help

```
spectre -h pvccs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Num of controlling voltage(s)	`nc`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Max Coefficient Number	`polyCoef`	x	-	-	-	-
Poly Coeff 0	`c0`	x	-	-	-	-
Poly Coeff 1	`c1`	x	-	-	-	-
Poly Coeff 2	`c2`	x	-	-	-	-
Poly Coeff 3	`c3`	x	-	-	-	-
Poly Coeff 4	`c4`	x	-	-	-	-
Gain	`gain`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Maximum Output Current	`maxm`	x	-	-	-	-
Minimum Output Current	`minm`	x	-	-	-	-
Absolute Output Current	`absol`	x	-	-	-	-
Smoothing Factor	`smoothing`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-

Example

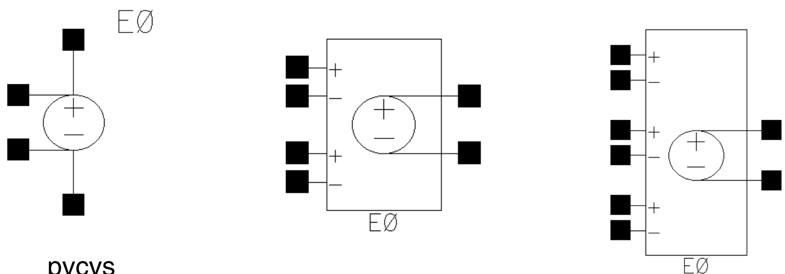
Following is the netlist when the pvccs coefficient is specified as an instance parameter and not specified in a file:

```
G0 (net21 net23 net22 net24 ) pvccs gain=1.0  m=1 coeffs=[ 1 1 1 1 ]  min=1.0
max=3.1 abs=off tc1=0 tc2=0
```

Following is the netlist when the pvccs coefficient is specified in a file:

```
G0 (net21 net23 net22 net24 ) pvccs gain=1.0  m=1 file="abc.coeff"  min=1.0
max=3.1 abs=off tc1=0 tc2=0
```

Symbol: pvcvs, pvcvs2, pvcvs3



pvcvs   pvcvs2   pvcvs3

Polynomial Voltage Controlled Voltage Source

It is a polynomial voltage controlled voltage source in which the symbol varies with the number of controlling voltages. A vector of coefficients specifies the polynomial function that defines the relationship between the output voltage and the controlling voltages. You must specify at least one coefficient. Current through the voltage source is calculated and is defined to be positive if it flows from the positive terminal, through the source, to the negative terminal.

For a polynomial in N variables a1, a2, ... an, the polynomial function F(a0,a1,...,an) is given by:

```
F = c0 + c1 * a1 + c2 * a2 + ...
+ c(m+1) * a1^2 + c(m+2) * a1 * a2 +...
+ c(2m+1) * a2^2 + c(2m+2) * a2 * a3 +...
```

where the ‘c’s are coefficients of the polynomial terms, and `m` is the multiplier.

Command-line help

```
spectre -h pvcvs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Max Coefficient Number	polyCoef	x	-	-	-	-
Poly Coeff 0	c0 - c20	x	-	-	-	-
Initial condition	ic	-	-	-	x	-
Scale factor	scale	-	-	-	x	-
Absolute value	abs	x	-	-	x	-
Gain	gain	x	-	-	-	-
Multiplier	m	x	-	-	-	-
Minimum Output Voltage	minm	x	-	-	x	-
Maximum Output Voltage	maxm	x	-	-	x	-
Absolute Output Voltage	absol	x	-	-	-	-
Smoothing Factor	smoothing	x	-	-	-	-
Temperature coefficient 1	tc1	-	-	-	x	-
Temperature coefficient 2	tc2	x	-	-	x	-

Syntax/Synopsis

```
Name ( p n ps1 ns1 ... ) pvcvs <parameter=value> ...
```

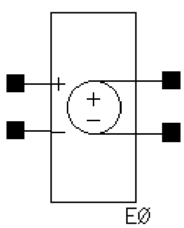
Example

```
v1 (p 0 c1 0) pvcvs coeffs=[0 0 0 0.1 1 1] gain=1
```

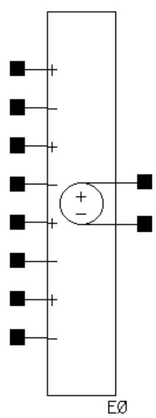
Additional Information

This device is supported within the altergroups.

Symbol: pvcvsp



pvcvsp



pvcvsp with 4
controlling voltages

Parameterized Cell Based Polynomial Nonlinear Voltage Controlled Voltage Source

pvcvsp is a Pcell-based polynomial voltage controlled voltage source in which the symbol varies with the number of controlling voltages. This component is similar to pvcvs except that it has one additional parameter (nc) that specifies the number of the controlling voltage sources.

Note: The maximum number of controlling voltages is 20. Therefore, if you specify a number greater than 20, the value of this parameter will default to 20.

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Num of controlling voltage(s)	`nc`	x	-	-	-	-
Max Coefficient Number	`polyCoef`	x	-	-	-	-
Poly Coeff 0	`c0`	x	-	-	-	-
Poly Coeff 1	`c1`	x	-	-	-	-
Poly Coeff 2	`c2`	x	-	-	-	-
Poly Coeff 3	`c3`	x	-	-	-	-
Poly Coeff 4	`c4`	x	-	-	-	-
Gain	`gain`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Minimum Output Voltage	`min`	x	-	-	-	-
Maximum Output Voltage	`max`	x	-	-	-	-
Absolute Output Voltage	`abs`	x	-	-	-	-
Smoothing Factor	`delta`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-

Example

Following is the netlist when the pvcvs coefficient is passed as an instance parameter and not specified in a file:

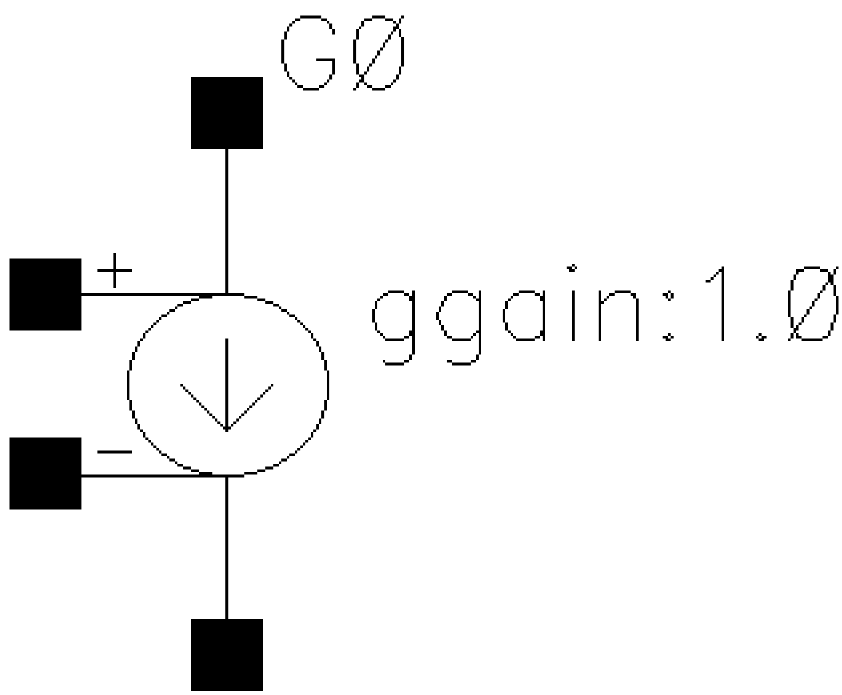
```
E0 (net21 net23 net22 net24 ) pvcvs gain=1.0  m=1 coeffs=[ 1 1 1 1 ]  min=1.0
max=3.1 abs=off tc1=0 tc2=0
```

Following is the netlist when the PVCVS coefficient is specified in a file:

```
E0 (net21 net23 net22 net24 ) pvcvs gain=1.0  m=1 file=”abc.coeff”  min=1.0
max=3.1 abs=off tc1=0 tc2=0
```

Note: The parameters, `polyCoef` and `coefSpec`, are not netlisted.

Symbol: vccs



Linear Voltage Controlled Current Source

Positive current exits the source node and enters the sink node.

Note: Component vccs uses the same values of parameters `ggain`, `maxm`, `minm`, and `m` for both Spectre and hspiceD simulators.

Command-line help

```
spectre -h vccs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Transconductance	ggain	x	-	-	x	x
Initial condition	ic	-	-	-	-	x
Multiplier	m	x	-	-	x	x
Type	csType	-	-	-	x	x
Transconductance (Obsolete)	hggain	-	-	-	x	-
Maximum output current (Obsolete)	maxi	-	-	-	x	x
Minimum output current (Obsolete)	mini	-	-	-	x	x
Maximum output current	maxm	x	-	-	x	-
Minimum output current	minm	x	-	-	x	-
Scale factor	scale	-	-	-	x	x
Multiplier (Obsolete)	hm	-	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Temperature coefficient 1	tc1	-	-	-	x	x
Temperature coefficient 2	tc2	-	-	-	x	x
Absolute value	habs	-	-	-	x	-
Initial condition	hic	-	-	-	x	-
Pwl type	pwlType	-	-	-	x	x
Delta	delta	x	-	-	x	x
Number of controlling pairs	xypairs	-	-	-	x	x
Delay Time	htd	-	-	-	x	-
Absolute value	abs	x	-	-	-	x
Delay Time	td	-	-	-	-	x
Type of input of source	inputtype	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ps ns ) vccs <parameter=value> ...
```

Example

```
v1 (1 0 2 3) gm=-1 m=2
```

Additional Information

This device is supported within the altergroups.

Voltage-Controlled Current Sources (G-Elements)

Voltage-Controlled Capacitor

```
Gxx n+ n- vccap pwl(1) in+ in- [delta = val] [scale = val]
     + [m = val] [tc1 = val] [tc2 = val] x1, y1, x2, y2 ... [ic = val]
```

Voltage-Controlled Current Source

Behavioral

```
Gxx n+ n- [vccs] cur = 'equation' [max = val] [min = val] [scale = val]
```

Linear

```
Gxx n+ n- [vccs] in+ in- transconductance [max = val] [min = val]
     + [m = val] [scale = val] [tc1 = val] [tc2 = val] [abs = 1] [ic = val]
```

Piece-Wise Linear

```
Gxx n+ n- [vccs] pwl(1) in+ in- [delta = val] [scale = val]
     + [m = val] [tc1 = val] [tc2 = val] x1, y1, x2, y2 ... [ic = val]
```

Polynomial

```
Gxx n+ n- [vccs] poly(ndim) in+ in- ... inndim+ inndim-
     + [tc1 = val] [tc2 = val] [scale = val] [max = val]
     + [min = val] [abs = 1] p0 [p1 ...] [ic = vals]
```

Delay Element

```
Gxx n+ n- [vccs] delay in+ in- td = val
     + [tc1 = val] [tc2 = val] [scale = val] [npdelay = val]
```

Voltage-Controlled Resistor

Linear

```
Gxx n+ n- vcr in+ in- transfactor [max = val] [min = val]
     + [m = val] [scale = val] [tc1 = val] [tc2 = val] [ic = val]
```

Piece-Wise Linear

```
Gxx n+ n- vcr pwl(1) in+ in- [delta = val] [scale = val]
     + [m = val] [tc1 = val] [tc2 = val] x1, y1, x2, y2 ... [ic = val]

Gxx n+ n- vcr npwl(1) in+ in- [delta = val] [scale = val]
     + [m = val] [tc1 = val] [tc2 = val] x1, y1, x2, y2 ... [ic = val]

Gxx n+ n- vcr ppwl(1) in+ in- [delta = val] [scale = val]
     + [m = val] [tc1 = val] [tc2 = val] x1, y1, x2, y2 ... [ic = val]
```

Polynomial

```
Gxx n+ n- vcr poly(ndim) in+ in- ... inndim+ inndim-
     + [tc1 = val] [tc2 = val] [scale = val] [max = val]
     + [min = val] [abs = 1] p0 [p1 ...] [ic = vals]
```

Description

Defines voltage-controlled current sources (VCCSs), voltage-controlled resistors (VCRs), and voltage-controlled capacitors (VCCAPs) in behavioral, linear, piece-wise linear, poly, and delay forms. In the behavioral function, the equation can contain terms of node voltages. In linear form, the output value is estimated with '[v(in+)-v(in-)]' multiplied by `transfactor` or `transconductance`, followed by the `scale` and temperature adjustment, before confined with the `abs`, `min`, and `max` parameters. In the piece-wise linear function, at least two pairs of voltage-current (or voltage-resistance, voltage-capacitance) points are required.

Arguments

`n+`, `n-`	Terminals of controlled element.
`in+`, `in-`	Positive and negative controlling nodes.
`vcr`, `vccap`, `vccs`	Keywords for the voltage-controlled resistor, capacitor, and current source elements. **Note:** `vcr`, `vccap`, and `vccs` are reserved words that cannot be used as node names.
`cur` = *'equation'*	Current of the controlled element flowing from `n+` to `n-`. It can be ■ An expression with parameters and functions of node voltages ■ Branch currents of other elements ■ Time, frequency, or temperature
`max` = *val*	Maximum value of the controlled current or resistance.
`min` = *val*	Minimum value of the controlled current or resistance.

`transconductance`	Voltage to current conversion factor.
`transfactor`	Voltage to resistance conversion factor.
`scale` = *`val`*	Scaling factor; scales current by its value (default = 1.0).
`m` = *`val`*	Multiplier (default = 1).
`tc1` = *`val`*	First-order temperature coefficient for the element.
`tc2` = *`val`*	Second-order temperature coefficient for the element.
`abs`	Output current takes its absolute value if `abs` = 1.
`ic` = *`val`*	Initial value of the current source (default = 0.0).
`delta` = *`val`*	A value used to smooth corners of the piece-wise linear function. The default is 1/4 of the smallest distance between break points, and is not to exceed 1/2 of this value.
`x1...`	Voltage drops between the controlling nodes `in+` and `in-`. They must be in ascending order.
`y1...`	Element output value corresponding to `x1...`
`npdelay`	The number of data points used in delay simulations.

The `npwl` and `ppwl` functions are used to interchange the `n+` and `n-` nodes, but use the same transfer function.

npwl

For the `in-` node connected to `n+`, if v(n+,n-) < 0, then the controlling voltage is v(in+,in-). Otherwise, the controlling voltage is v(in+,n-).

For the `in-` node connected to `n-`, if v(n+,n-) > 0, then the controlling voltage is v(in+,in-). Otherwise, the controlling voltage is v(in+,n+).

ppwl

For the `in-` node, connected to `n+`, if v(n+,n-) > 0, then the controlling voltage is v(in+,in-). Otherwise, the controlling voltage is v(in+,n-).

For the `in-` node, connected to `n-`, if v(n+,n-) < 0, then the controlling voltage is v(in+,in-). Otherwise, the controlling voltage is v(in+,n+).

Note: If the `in-` node does not connect to either `n+` or `n-`, the Virtuoso UltraSim simulator changes `npwl` and `ppwl` to `pwl`.

Examples

In the following example

```
G1 1 2 cur = '3.0*sin(v(7)/2)+v(6)^2'
```

defines a `VCCS` connected to nodes `1` and `2`, with its current dependent on the voltage of nodes `6` and `7` in the given form.

In the next example

```
G2 1 2 vccs 5 0 0.5 max = 5 min = 0 m = 2 ic = 0
```

defines a `VCCS` connected to nodes `1` and `2`. Its current is initialized as 0, and is half of the voltage at node `5`. The current is also confined within 0 and 5 amps. The output current is multiplied by 2.

In the next example

```
G3 1 2 vccs pwl(1) 5 0 delta = 0.2 0, 0 0.5,1 1.5,1.5 scale = 1.e-3
```

defines a `VCCS` connected to nodes `1` and `2`, its current controlled by the voltage at node `5`. The current is calculated in a piece-wise linear function with a smoothing parameter of 0.2, and is scaled by 1.e-3 upon output.

In the next example

```
Gres 1 2 vcr pwl(1) 5 4 m = 3 0,0 1,1k 2,1.5k 3,1.8k 4,2.0k 5,2.0k ic = 1k
```

defines a `VCR` connected to nodes `1` and `2`, with its resistance dependent on the voltage difference between nodes `5` and `4` in a piece-wise linear form. The initial resistance is 1k. The output resistance is decreased by 2/3.

In the next example

```
Gcap 1 2 vccap pwl(1) 5 4 m = 3 scale = 1.e-12 0,0 1,10 2,15 3,18 4,20 5,20 ic = 10
```

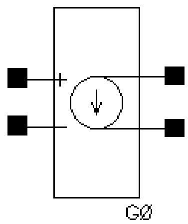
defines a `VCCAP` connected to nodes `1` and `2`, with its capacitance dependent on the voltage difference between nodes `5` and `4` in a piece-wise linear form. The initial capacitance is set to 10 p after being scaled with 1e-12. The output capacitance is increased by a factor of 3.

In the next example

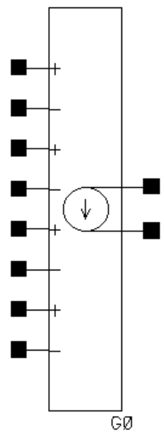
```
Gnmos d s vcr npwl(1) g s m =3 0,5g 1,5meg 2,5k 3,1k 5,50
```

tells the Virtuoso UltraSim simulator to model the source-drain resistor of the n-channel MOSFET which is used as a switch. Based on the `npwl` function, the resistor value (`Gnmos`) does not change when changing the position of the `d` and `s` nodes.

Symbol: vccsp



vccsp



vccsp with 4
controlling voltages

Parameterized Cell Based Voltage Controlled Current Source

vccsp is a Pcell-based voltage controlled current source in which the symbol varies with the number of controlling voltages. vccsp is similar to vccs except that it has one additional parameter (nc) that specifies the number of controlling voltage sources.

Note: The maximum number of controlling voltages is 20. Therefore, if you specify a number greater than 20, the value of this parameter will default to 20.

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Num of controlling voltage(s)	`nc`	x	-	-	-	-
Transconductance	`ggain`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Type of Source	`typesrc`	x	-	-	-	-
Minimum Output Voltage	`min`	x	-	-	-	-
Maximum Output Voltage	`max`	x	-	-	-	-
Absolute Output Voltage	`abs`	x	-	-	-	-
Smoothing Factor	`delta`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-

Example

Following is the netlist for the linear transfer characteristic:

```
G0  (net011 net012 net09 net010) vcvs type=vcvs  m=1  gain =1.0 min=1 max=4
abs=off tc1= 0 tc2=0
```

Following is the netlist for the PWL transfer characteristic when the voltage or voltage pair is not specified in a file:

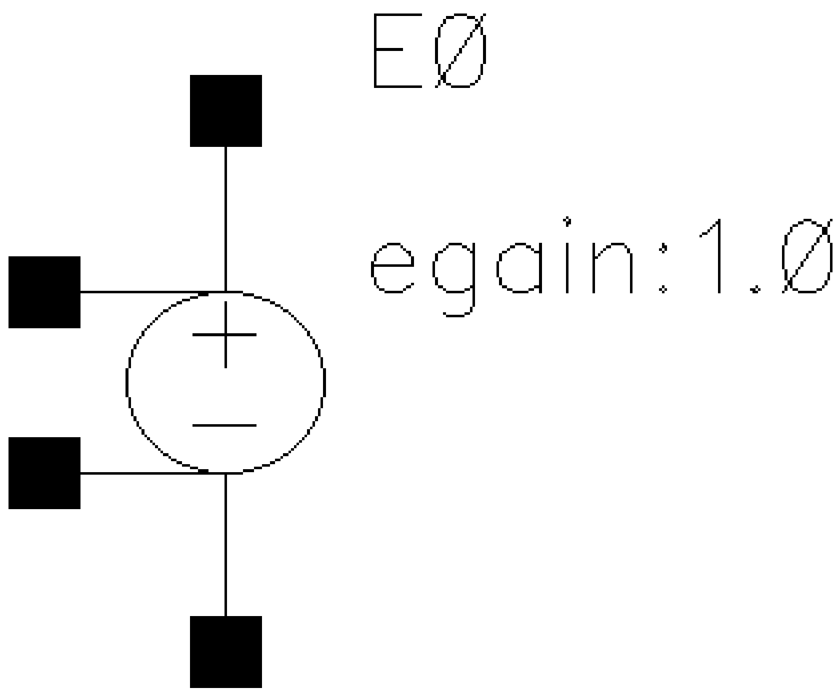
```
G0  (net011 net012 net09 net010) vcvs type=vcvs  m=1 pwl=[1 1 2 4] scale=1
delta=0 tc1=0 tc2=0
```

Following is the netlist for the PWL transfer characteristic when the voltage or voltage pair is specified in a file:

```
G0  (net011 net012 net09 net010) vcvs type=vcvs  m=1 file=”abc” scale=1 delta=0
tc1=0 tc2=0
```

Note: The parameters, `trfType` and `iVectSpec` are not netlisted.

Symbol: vcvs



Linear Voltage Controlled Voltage Source

Current through the voltage source is calculated and is defined to be positive if it flows from the positive terminal, through the source, to the negative terminal.

Note: Component vcvs uses the same values of parameters `egain`, `maxm`, and `minm` for both Spectre and hspiceD simulators.

Command-line help

```
spectre -h vcvs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Voltage gain	egain	x	-	-	x	x
Initial condition	ic	-	-	-	-	x
Multiplier	m	x	-	-	-	-
Type	csType	-	-	-	x	x
Voltage gain (obsolete)	hegain	-	-	-	x	-
Maximum output voltage (obsolete)	maxv	-	-	-	x	x
Minimum output voltage (obsolete)	minv	-	-	-	x	x
Maximum output current	maxm	x	-	-	x	-
Minimum output current	minm	x	-	-	x	-
Scale factor	scale	-	-	-	x	x
Temperature coefficient 1	tc1	-	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Temperature coefficient 2	tc2	-	-	-	x	x
Absolute value	habs	-	-	-	x	-
Initial condition	hic	-	-	-	x	-
Delta	delta	x	-	-	x	x
Number of controlling pairs	xypairs	-	-	-	x	x
Delay Time	htd	-	-	-	x	-
Absolute value	abs	x	-	-	-	x
Delay Time	td	-	-	-	-	x

Syntax/Synopsis

```
Name ( p n ps ns ) vcvs <parameter=value> ...
```

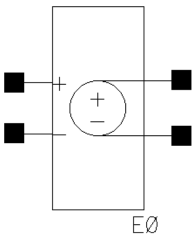
Example

```
e1 (out1 0 pos neg) vcvs gain=10
```

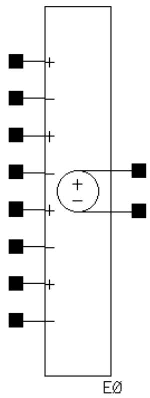
Additional Information

This device is supported within the altergroups. This device can also model ideal digital gates.

Symbol: vcvsp



vcvsp



vcvsp with 4
controlling voltages

Parameterized Cell Based Voltage Controlled Voltage Source

vcvsp is a Pcell-based voltage controlled voltage source in which the symbol varies with the number of controlling voltages.

Note: The maximum number of controlling voltages is 20. Therefore, if you specify a number greater than 20, the value of this parameter will default to 20.

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Num of controlling voltage(s)	`nc`	x	-	-	-	-
Type of transfer char	`trfType`	x	-	-	-	-
Voltage gain	`egain`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Type of source	`typesrc`	x	-	-	-	-
Minimum Output Voltage	`min`	x	-	-	-	-
Maximum Output Voltage	`max`	x	-	-	-	-
Absolute Output Voltage	`abs`	x	-	-	-	-
Smoothing Factor	`delta`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-

Example

Following is the netlist for linear transfer characteristic:

```
G0  (net011 net012 net09 net010) vcvs type=vcvs  m=1  gain =1.0 min=1 max=4
abs=off tc1= 0 tc2=0
```

Following is the netlist for PWL transfer characteristic when the voltage or voltage pair is not specified in a file but passed as an instance parameter:

```
G0  (net011 net012 net09 net010) vcvs type=vcvs  m=1 pwl=[1 1 2 4]   scale=1
delta=0 tc1=0 tc2=0
```

Following is the netlist for PWL transfer characteristic when the voltage or voltage pair is not specified in a file:

```
G0  (net011 net012 net09 net010) vcvs type=vcvs  m=1 file=”abc”   scale=1
delta=0 tc1=0 tc2=0
```

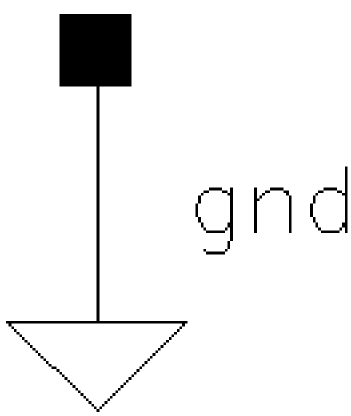
Note: The parameters, `trfType` and `iVectSpec`, are not netlisted.

7

Sources - Global Components

This chapter contains a list of all those components of the Analog Library that netlist as a global net.

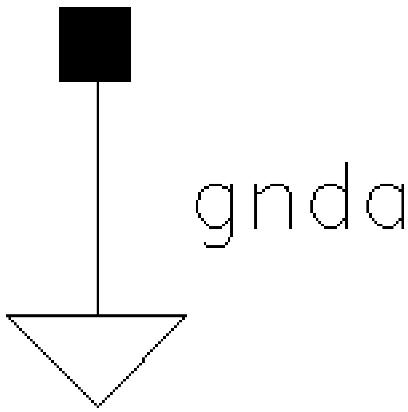
Symbol: gnd



Component Parameters

gnd has no component parameters.

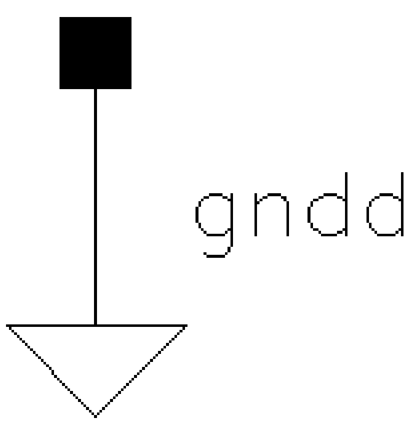
Symbol: gnda



Component Parameters

gnda has no component parameters.

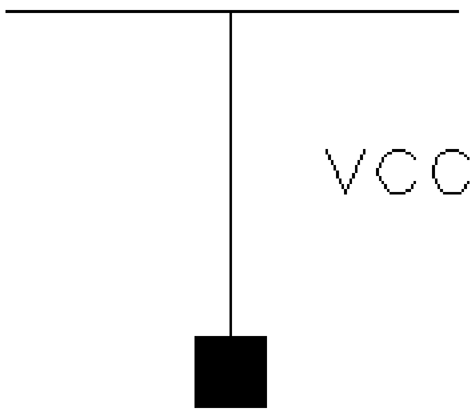
Symbol: gndd



Component Parameters

gndd has no component parameters.

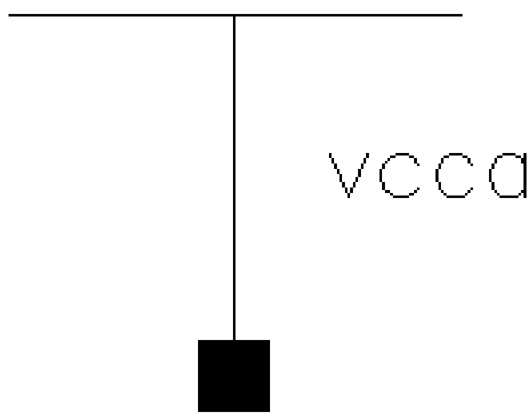
Symbol: vcc



Component Parameters

vcc has no component parameters.

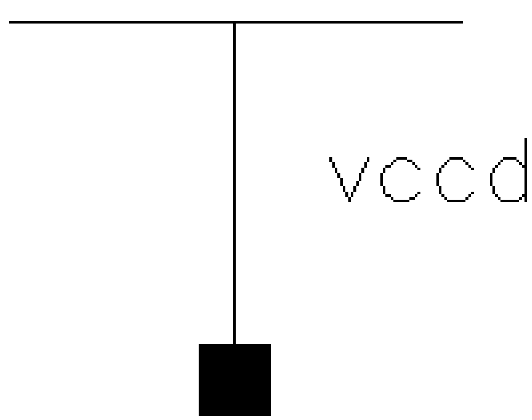
Symbol: vcca



Component Parameters

vcca has no component parameters.

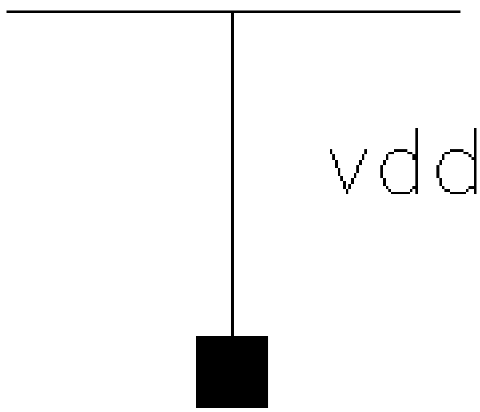
Symbol: vccd



Component Parameters

vccd has no component parameters.

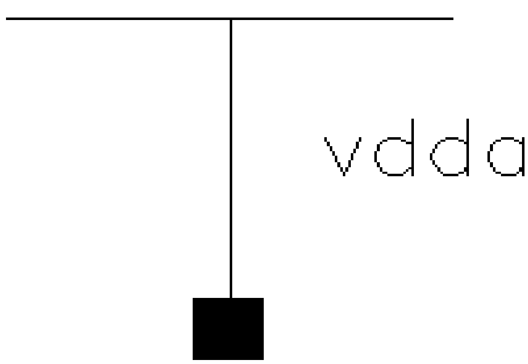
Symbol: vdd



Component Parameters

vdd has no component parameters.

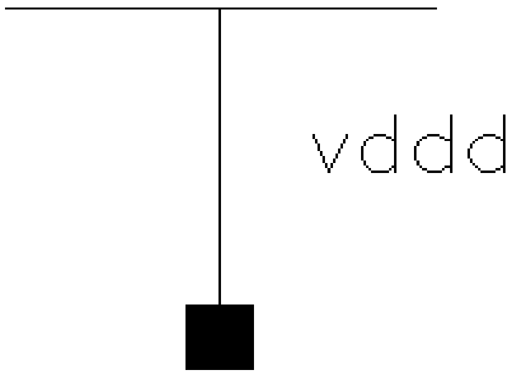
Symbol: vdda



Component Parameters

vdda has no component parameters.

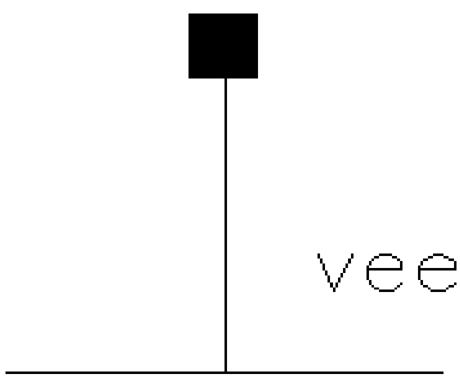
Symbol: vddd



Component Parameters

vddd has no component parameters.

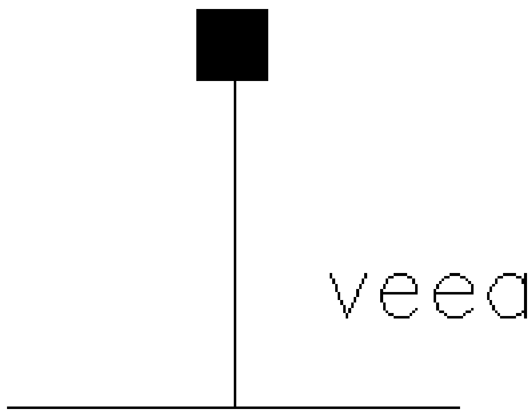
Symbol: vee



Component Parameters

vee has no component parameters.

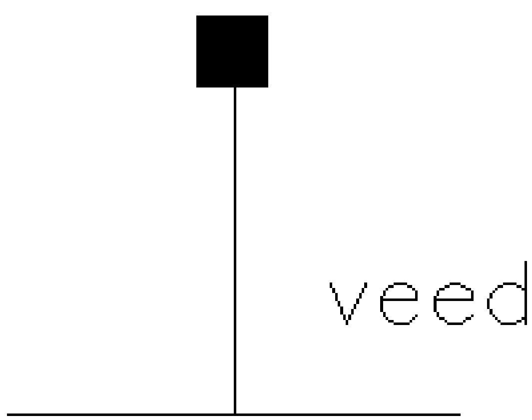
Symbol: veea



Component Parameters

veea has no component parameters.

Symbol: veed



Component Parameters

veed has no component parameters.

Symbol: vrefgnd

This component can be used at either the global level or the subckt level as a local option. For example, if it is in the netlist, option `vrefgnd1` is applied at the subckt level and `vrefgnd2` is applied on the top node, as follows:

```
vrefgnd1 options node_name=n1 subckt=x1
vrefgnd2 options node_name=top_n1
```

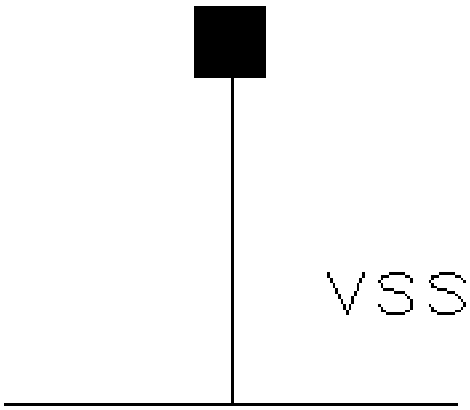
Here,

- `vrefgnd1` is applied to `subcktx1` and node `x1.n1` is used as reference gnd.
- `vrefgnd2` is applied to the top level and node `n1` is used as reference gnd.

Component Parameters

vrefgnd has no component parameters.

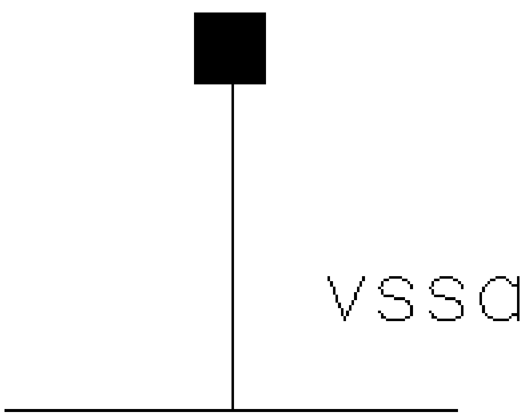
Symbol: vss



Component Parameters

vss has no component parameters.

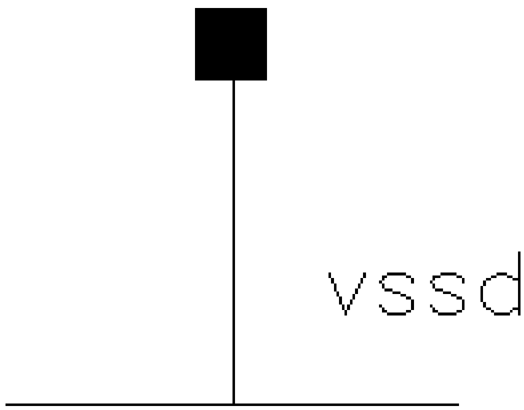
Symbol: vssa



Component Parameters

vssa has no component parameters.

Symbol: vssd



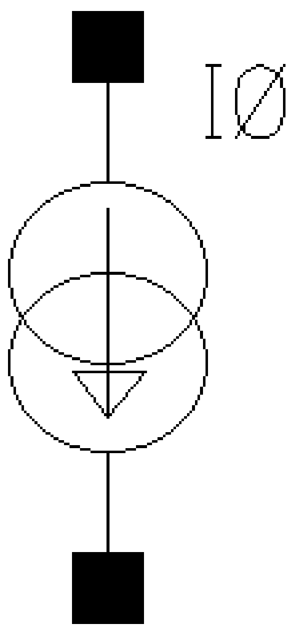
Component Parameters

vssd has no component parameters.

8

Sources - Independent Components

Symbol: idc



Independent DC Current Source

idc is a constant isource.

Command-line help

```
spectre -h isource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	x	-	-	x	x
AC phase	`acp`	x	-	-	x	x
DC current	`idc`	x	-	-	x	x
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Multiplier	`m`	x	-	-	x	x
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Source type	`srcType`	x	-	-	-	-
AC Phase	`acPhase`	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ) isource <parameter=value> ...
```

Positive current exits the source node and enters the sink node.

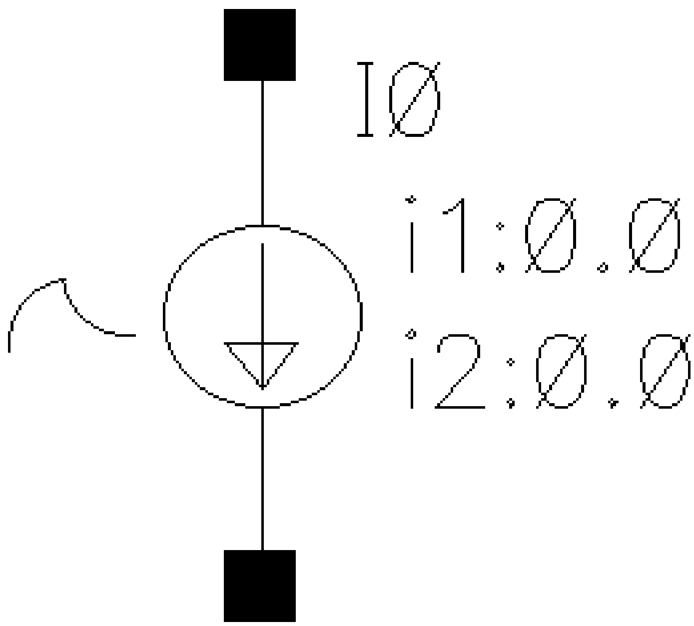
Example

```
i1 (in 0) isource dc=0 type=pulse delay=10n val0=0 val1=500u period=500n

rise=1n fall=1n width=250n
```

Additional Information

This device is supported within the altergroups.

Symbol: iexp



Independent Exponential Current Source

Command-line help

```
spectre -h isource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	x	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC phase	`acp`	x	-	-	x	x
DC current	`idc`	x	-	-	x	x
Current 1	`i1`	x	-	-	x	x
Current 2	`i2`	x	-	-	x	x
Delay time 1	`td1`	x	-	-	x	x
Damping factor 1	`tau1`	x	-	-	x	x
Delay time 2	`td2`	x	-	-	x	x
Damping factor 2	`tau2`	x	-	-	x	x
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/ freq pairs	`FNpairs`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Multiplier	`m`	x	-	-	x	x
Delay time	`td`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
DC source	`dc`	-	-	-	x	x
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Delay Time	`delay`	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ) isource <parameter=value> ...
```

Positive current exits the source node and enters the sink node.

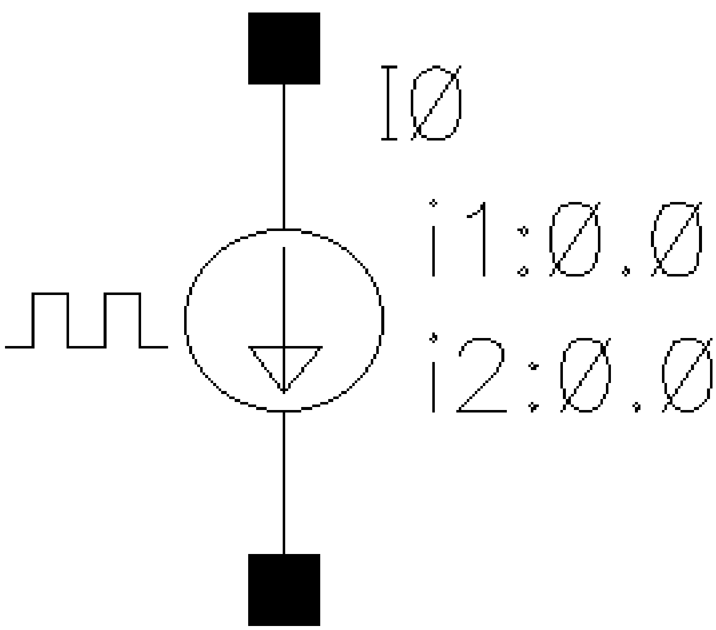
Example

```
i1 (in 0) isource dc=0 type=pulse delay=10n val0=0 val1=500u period=500n
rise=1n fall=1n width=250n
```

Additional Information

This device is supported within the altergroups.

Symbol: ipulse



Independent Pulse Current Source

ipulse is a square wave varying isource.

Command-line help

```
spectre -h isource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	x	-	-	x	x
AC phase	`acp`	x	-	-	x	x
DC current	`idc`	x	-	-		
Current 1	`i1`	x	-	-	x	x
Current 2	`i2`	x	-	-	x	x
Delay time	`td`	x	-	-	x	x
Rise time	`tr`	x	-	-	x	x
Fall time	`tf`	x	-	-	x	x
Pulse width	`pw`	x	-	-	x	x
Period	`per`	x	-	-	x	x
Frequency name for 1/ period	`fundname`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Multiplier	`m`	x	-	-	x	x
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
DC source	`dc`	-	-	-	x	x
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Source type	`srcType`	x	-	-	-	-
Type of rising & falling edge	`edgetype`	x	-	-	-	-
Delay Time	`delay`	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ) isource <parameter=value> ...
```

Positive current exits the source node and enters the sink node.

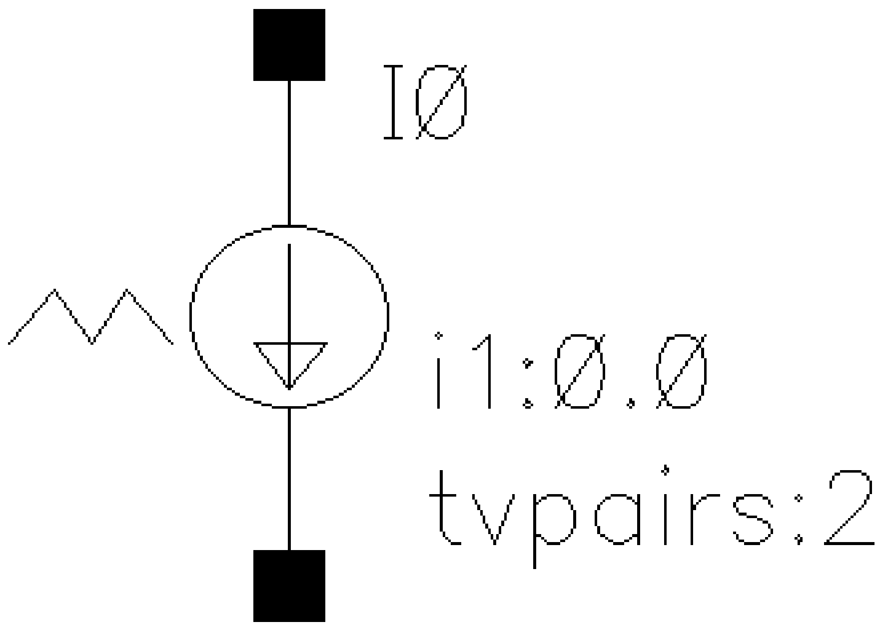
Example

```
i1 (in 0) isource dc=0 type=pulse delay=10n val0=0 val1=500u period=500n
rise=1n fall=1n width=250n
```

Additional Information

This device is supported within the altergroups.

Symbol: ipwl



Independent Piece-Wise Linear Current Source

Command-line help

```
spectre -h isource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Number of pairs of points	`tvpairs`	x	-	-	x	x
AC magnitude	`acm`	x	-	-	x	x
AC phase	`acp`	x	-	-	x	x
DC current	`idc`	x	-	-		

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Time 1	`t1 - t50`	x	-	-	x	x
Current 1	`i1 - i50`	x	-	-	x	x
Frequency name for 1/period	`fundname`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Multiplier	`m`	x	-	-	x	x
Delay time	`td`	x	-	-	x	x
Offset current	`io`	x	-	-	-	-
Scale factor	`scale`	x	-	-	-	-
Time scale factor	`stretch`	x	-	-	-	-
Period of the PWL	`pwlperiod`	x	-	-	-	-
Transition width	`twidth`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
DC source	`dc`	-	-	-	x	x
Repeated function	`rpt`	-	-	-	x	x
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Source type	`srcType`	x	-	-	-	-
Type of rising & falling edge	`edgetype`	x	-	-	-	-
Delay Time	`delay`	x	-	-	-	-
Transition width	`twidth`	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ) isource <parameter=value> ...
```

Positive current exits the source node and enters the sink node.

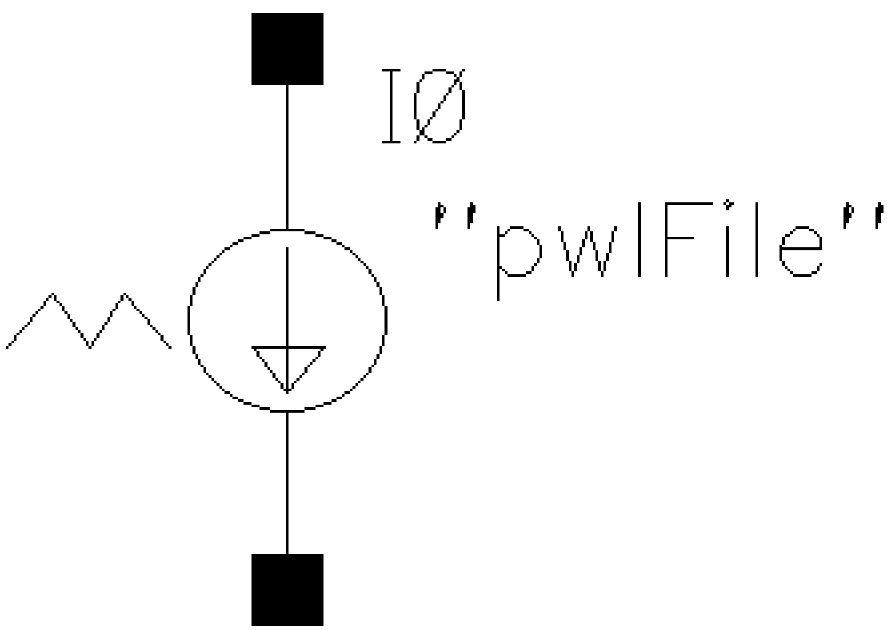
Example

```
i1 (in 0) isource dc=0 type=pulse delay=10n val0=0 val1=500u period=500n
rise=1n fall=1n width=250n
```

Additional Information

This device is supported within the altergroups.

Symbol: ipwlf



Independent Piece-Wise Linear Current Source Based on a File

Command-line help

```
spectre -h isource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	x	-	-	x	-
AC phase	`acp`	x	-	-	x	-
DC current	`idc`	x	-	-	x	-
PWL file name	`fileName`	x	-	-	x	-
Frequency name for 1/period	`fundname`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Delay time	`td`	x	-	-	-	-
Offset current	`io`	x	-	-	-	-
Scale factor	`scale`	x	-	-	-	-
Time scale factor	`stretch`	x	-	-	-	-
Period of the PWL	`pwlperiod`	x	-	-	-	-
Transition width	`twidth`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Source type	`srcType`	x	-	-	-	-
Type of rising & falling edge	`edgetype`	x	-	-	-	-
Delay Time	`delay`	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ) isource <parameter=value> ...
```

Positive current exits the source node and enters the sink node.

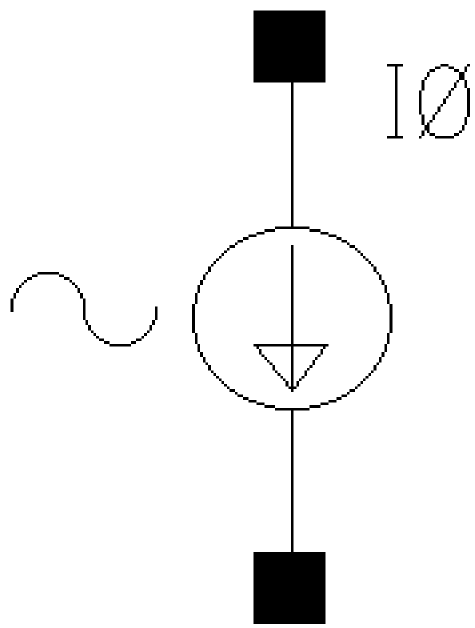
Example

```
i1 (in 0) isource dc=0 type=pulse delay=10n val0=0 val1=500u period=500n
rise=1n fall=1n width=250n
```

Additional Information

This device is supported within the altergroups.

Symbol: isin



Independent Sinusoidal Current Source

isin is a sin wave isource.

Command-line help

```
spectre -h isource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	x	-	-	x	x
AC phase	`acp`	x	-	-	x	x
DC current	`idc`	x	-	-		
Offset current	`io`	x	-	-	x	x
Amplitude	`ia`	x	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Frequency	`freq`	x	-	-	x	x
Delay time	`td`	x	-	-	x	x
Damping factor	`theta`	x	-	-	x	x
First frequency name	`fundname`	x	-	-	-	-
Second frequency name	`fundname2`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Number of FM files	`filenums`	x	-	-	-	-
Name of FM File1	`fmmodfile1`	x	-	-	-	-
Name of FM File2	`fmmodfile2`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Multiplier	`m`	x	-	-	x	x
Initial phase for Sinusoid	`sinephase`	x	-	-	-	-
Amplitude 2	`ia2`	x	-	-	-	-
Initial phase for Sinusoid 2	`sinephase2`	x	-	-	-	-
Frequency 2	`freq2`	x	-	-	-	-
FM modulation index	`fmmodindex`	x	-	-	-	-
FM modulation frequency	`fmmodfreq`	x	-	-	-	-
AM modulation index	`ammodindex`	x	-	-	-	-
AM modulation frequency	`ammodfreq`	x	-	-	-	-
AM modulation phase	`ammodphase`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
DC source	`dc`	-	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Phase delay	`phi`	-	-	-	x	x
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Delay Time	`delay`	x	-	-	-	-
Sine DC level	`sinedc`	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ) isource <parameter=value> ...
```

Positive current exits the source node and enters the sink node.

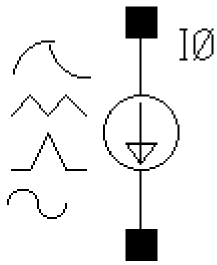
Example

```
i1 (in 0) isource dc=0 type=pulse delay=10n val0=0 val1=500u period=500n
rise=1n fall=1n width=250n
```

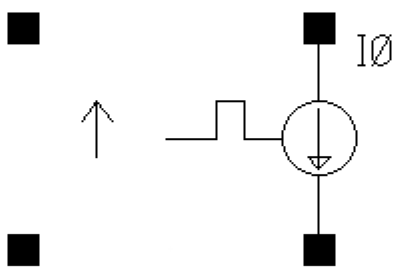
Additional Information

This device is supported within the altergroups.

Symbol: isource



If *Source type* = `prbs`, and *Trigger* = `External rising edge`, `External falling edge`, or `External both edges`, two extra ports are added to the isource symbol as shown below:



You can specify the wave shape for isource by selecting one of the following options from the *Source type* drop-down list box in the Edit Object Properties form:

- *dc*—Generates a dc level from isource. When the Source type is set to dc, the dc and temperature effect parameters are active. The dc setting sets the DC level for all analyses.
- *sine*—Generates sinusoidal waveforms.

  Up to two sinusoids can be generated simultaneously. They are denoted as 1 and 2. You can set the amplitude, frequency, and phase for both individually. The amplitude can be set to either a current or a power level. When you set a power level, the assumption is that the isource is perfectly matched. The source that is internal to isource gets double the amplitude specified by the power in `dBm`. You can also specify sinusoidal AM or FM modulation of sinusoid 1. Sinusoid 2 cannot be modulated.
- *pulse*—Generates a step, a single pulse, or a periodic pulse waveform.

When you specify the current, you are specifying the current when isource is properly terminated, and not the current on the internal current source. Therefore, the current on the internal source is set to twice the value specified on the component.

- *exp*—Generates an exponential waveform. The exponential waveform can generate one exponential pulse, and cannot generate a periodic signal.

  When you specify the current, you are specifying the current when isource is properly terminated, and not the current on the internal current source. Thus, the current on the internal source is set to twice the value specified on isource.

- *pwl*—Generates piecewise linear waveforms that allow an arbitrary input waveform to be generated.

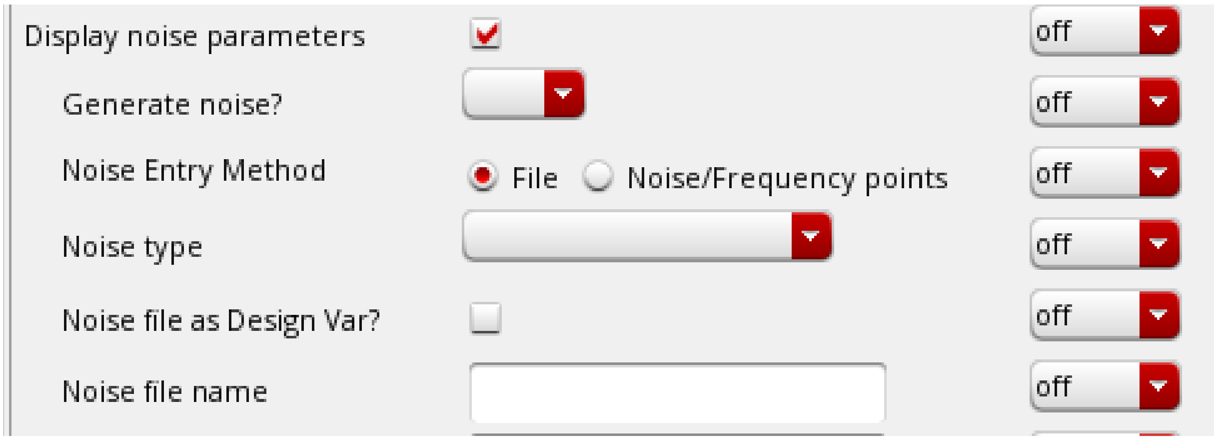
  The input can either be a file that contains time and current pairs, or you can enter the time-current pairs directly in the PWL source properties form. Remember that the current you enter in the piecewise linear file assumes that the isource is properly terminated. The internal current source gets set to double the value specified in the piecewise linear current specifications.

- *bit*—Generates bit sequence or string from isource. The bit source has four states: 1, 0, m, and z, which represent the high, low, middle current, and high-impedance state respectively. It allows patterns defining a sequence of bits.

- *prbs*—PRBS is an acronym for Pseudo-Random Binary Sequence. This source has three modes. It can be used to generate a maximum-length pseudo-random sequence. You can specify the beginning state and tap gains for a Fibonacci PRBS generator. A third mode allows reading an ASCII file that describes the sequence of one and zero events to generate.

Note: The symbol `ibit` is the same as `isource type=bit`.

If you select the *Display noise parameters* check box in the Edit Object Properties form, the *Noise file as Design Var?* check box is displayed. You can select this check box to specify the noise file as a design variable in the *Noise file name* field.



For more information on the available source types, see the section *Source type* in *AnalogLib Components Used in RF Simulation*.

Independent Current Source

The value of the DC current as a function of the temperature is given by:

```
I(T) = I(tnom) * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2].
```

Command-line help

```
spectre -h isource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
DC current	idc	x	-	-	x	x
Source type	srcType	x	-	-	x	-
Frequency name 1	fundname	x	-	-	-	-
Frequency 1	freq	x	-	-	x	x
Amplitude 1 (Ipk)	ia	x	-	-	x	x
Amplitude 2(Ipk)	ia2	x	-	-	x	x
Phase for Sinusoid 1	sinephase	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Sine DC level	`sinedc`	x	-	-	x	-
Sinusoid Ampl 1 (Ipk) to Sinusoid Ampl 9 (Ipk)	`vav1 - vav9`	x	-	-	-	-
File name	`fileName`	x	-	-	x	-
Browse and select file	`selectFile`	x	-	-	x	-
Number of PWL/Time pair	`tvpairs`	x	-	-	x	-
Time 1	`t1 - t50`	x	-	-	x	-
Current 1	`i1 - i50`	x	-	-	x	-
Delay time	`td`	x	-	-	-	x
Type of rising & falling edge	`edgetype`	x	-	-	x	-
Rise time start	`td1`	x	-	-	x	-
Rise time constant	`tau1`	x	-	-	x	-
Fall time start	`td2`	x	-	-	x	-
Fall time constant	`tau2`	x	-	-	x	-
DC offset	`offset`	x	-	-	-	-
Amplitude scale factor	`scale`	x	-	-	-	-
Time scale factor	`stretch`	x	-	-	-	-
Breakpoints	`allbrkpts`	x	-	-	-	-
Period	`pwlperiod`	x	-	-	x	-
Period start time	`pwlperiodstart`	x	-	-	x	-
Transition width	`twidth`	x	-	-	-	-
Period of waveform	`per`	x	-	-	x	-
Display second sinusoid	`numofsines`	x	-	-	-	-
FM modulation index 1	`fmmodindex`	x	-	-	x	-
FM modulation freq 1	`fmmodfreq`	x	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AM modulation index 1	`ammodindex`	x	-	-	x	-
AM modulation freq 1	`ammodfreq`	x	-	-	x	-
AM modulation phase 1	`ammodphase`	x	-	-	-	-
Damping factor 1	`theta`	x	-	-	x	-
Display small signal params	`smallSig`	x	-	-	x	-
PAC Magnitude (Ipk)	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
AC Magnitude (Ipk)	`acm`	x	-	-	x	-
AC phase	`acp`	x	-	-	x	-
XF Magnitude (Ipk)	`xfm`	x	-	-	-	-
Display noise parameters	`noiseParam`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Display modulation params	`modulation`	x	-	-	x	-
Display temperature params	`tempParam`	x	-	-	-	-
Linear temp. coefficient	`tc1`	x	-	-	-	-
Quadratic temp. coeff.	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
Transition reference	`transitionreference`	x	-	-	-	-
Multiplier	`m`	x	-	-	x	x
DC source	`dc`	-	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Phase delay	`phi`	-	-	-	x	x
Number of FM files	`filenums`	x	-	-	-	-
Name of FM File1	`fmmodfile1`	x	-	-	-	-
Name of FM File2	`fmmodfile2`	x	-	-	-	-
Reference Value	`ref`	x	-	-	-	-
Remove Device	`lxRemoveDevice`	-	x	-	-	-
RJ(seed)	`rjseed`	x	-	-	-	-
RJ(rms)	`rjrms`	x	-	-	-	-
PJ(amplitude)	`pjamp`	x	-	-	-	-
PJ(frequency)	`pjfreq`	x	-	-	-	-
PJ(type)	`pjtype`	x	-	-	-	-
PAM modulation	`pam4_modulation`	x	-	-	-	-
PAM4 mapping	`pam4_mapping`	x	-	-	-	-
Threshold	`triggerthreshold`	x	-	-	-	-
High-Z impedance	`highz`	x	-	-	-	-
Min high-Z trans. width	`min_z_transition_width`	x	-	-	-	-
Z state 1 to Z state 50	`Z1 - Z50`	x	-	-	-	-

Note: For HspiceD, parameter `pwlperiod` is supported under the following conditions:

- In case `pwlperiod` is specified and `pwlperiodstart` is not specified, then another current-time pair must be added, where time = `pwlperiod` and current is the same as the current in the last current-time pair.

  But, if the value specified for `pwlperiod` is the same as the time specified in the last current-time pair, then no additional current-time pair is required.

- In case both `pwlperiod` and `pwlperiodstart` are specified, then another current-time pair must be added, where time = (`pwlperiod + pwlperiodstart`) and current is the same as the current in the last current-time pair.

Syntax/Synopsis

```
Name ( sink src [ctl] ) isource <parameter=value> ...
```

Positive current exits the source node and enters the sink node. The third node(ctl) is used as a switch only for prbs.

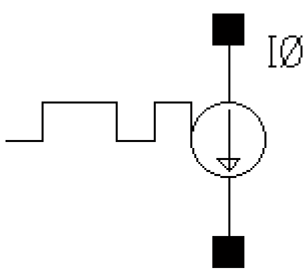
Example

```
i1 (in 0) isource dc=0 type=pulse delay=10n val0=0 val1=500u period=500n
rise=1n fall=1n width=250n
```

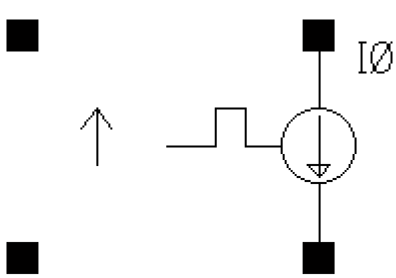
Additional Information

This device is supported within the altergroups.

Symbol: iprbs



If *Trigger* = `External rising edge`, `External falling edge`, or `External both edges`, two extra ports are added to the iprbs symbol as shown below:



Independent Current Source

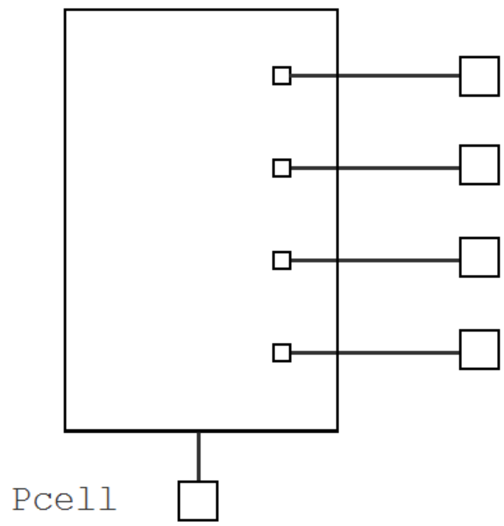
Command-line help

```
spectre -h isource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Bit string	`data`	x	-	-	-	-	-	-	-
Final value for logical 1	`val1`	x	-	-	-	-	-	-	-
Final value for logical 0	`val0`	x	-	-	-	-	-	-	-
Delay time	`delay`	x	-	-	-	-	-	-	-
Rise time	`rise`	x	-	-	-	-	-	-	-
Fall time	`fall`	x	-	-	-	-	-	-	-
Period of waveform	`period`	x	-	-	-	-	-	-	-
Reference Value	`ref`	x	-	-	-	-	-	-	-
Waveform Random Delay Time	`jitter`	x	-	-	-	-	-	-	-
Generates Random Count	`seed`	x	-	-	-	-	-	-	-
Bit	`taps`	x	-	-	-	-	-	-	-

Symbol: multibit



multibit is a Pcell, which allows you to provide a DC stimulus for a bus having multiple bits. The number of bits, the bit pattern, logic high, and logic low voltages can be selected as parameters. The Pcell also supports scalar (single bit) as well as bus outputs.

Note: The multibit device does not support more than 32 bits.

Example

For instance `I0`, the netlist as follows:

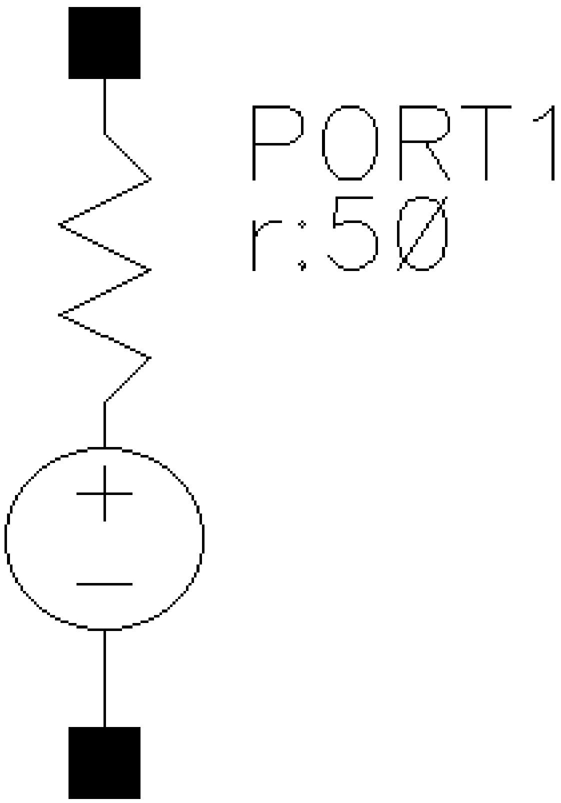
```
// Library name: analogLib
// Cell name: multibit
// View name: schematic
subckt multibit_pcell_0 a0 a1 a2 a3 ref
parameters a3=fmod(int((0)/8),2) vbit1=1 vbit0=0 a2=fmod(int((0)/4),2) \
        a1=fmod(int((0)/2),2) a0=fmod(int((0)/1),2)
    V3 (a3 ref) vsource dc=a3 > 0 ? vbit1 : vbit0 type=dc
    V2 (a2 ref) vsource dc=a2 > 0 ? vbit1 : vbit0 type=dc
    V1 (a1 ref) vsource dc=a1 > 0 ? vbit1 : vbit0 type=dc
    V0 (a0 ref) vsource dc=a0 > 0 ? vbit1 : vbit0 type=dc
ends multibit_pcell_0
// End of subcircuit definition.

// Library name: InhConn
// Cell name: test
```

```
// View name: schematic
I0 (net5 net4 net3 net2 net1) multibit_pcell_0 a3=fmod(int((0)/8),2) \
        vbit1=1 vbit0=0 a2=fmod(int((0)/4),2) a1=fmod(int((0)/2),2) \
        a0=fmod(int((0)/1),2)
```

Component Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Input Mode	mode1	x	-	-	-	x
Number of Bits	numbits	x	-	-	-	x
Expand Bus	expand	x	-	-	-	x
Bit Pattern(MSB...LSB)	pattern	x	-	-	-	x
Decimal Value	dec	x	-	-	-	x
Bit 1 voltage level	vbit1	x	-	-	-	x
Bit 0 voltage level	vbit0	x	-	-	-	x

Symbol: pdc



Independent DC Resistive Source

When Source type=dc, the dc and temperature effect parameters are active and set the DC level for all analyses. The DC voltage sets the DC level of the source for DC analysis. The value must be a real number. If you do not specify the DC value, it is assumed to be the time =0 value of the waveform.

The DC voltage parameter specifies the DC voltage across the port when it is terminated in its reference resistance. In other words, the DC voltage of the internal voltage source is double the user specified DC value, dc. The same is true for the values for the transient, AC, and PAC signals of the port.

For more information on this component refer to Appendix H of the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Command-line help

```
spectre -h port
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Resistance	`r`	x	-	-	-	-
Port number	`num`	x	-	-	-	-
DC voltage	`vdc`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
AC magnitude	`acm`	x	-	-	-	-
AC phase	`acp`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Source type	`srcType`	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) port <parameter=value> ...
```

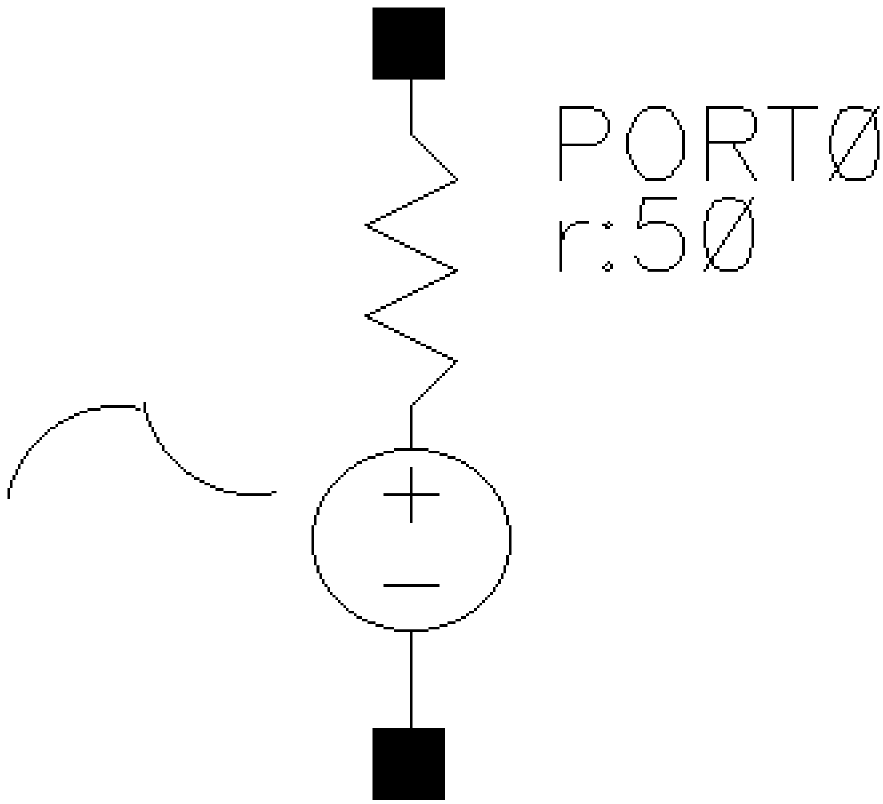
Example

```
p20 (2 0) port num=2 r=50 type=pulse period=1e-9 rise=1e-10 fall=1e-10 val1=1
width=0.5n mag=1
```

Additional Information

This device is not supported within the altergroups.

Symbol: pexp



Independent Exponential Resistive Source

For more information on this component refer to Appendix H of the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Command-line help

```
spectre -h port
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Resistance	`r`	x	-	-	-	-
Port number	`num`	x	-	-	-	-
DC voltage	`vdc`	x	-	-	-	-
Delay time	`td`	x	-	-	-	-
Delay time 1	`td1`	x	-	-	-	-
Damping factor 1	`tau1`	x	-	-	-	-
Delay time 2	`td2`	x	-	-	-	-
Damping factor 2	`tau2`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	acm	x	-	-	-	-
AC phase	acp	x	-	-	-	-
XF magnitude	xfm	x	-	-	-	-
PAC magnitude	pacm	x	-	-	-	-
PAC phase	pacp	x	-	-	-	-
Source type	srcType	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) port <parameter=value> ...
```

Example

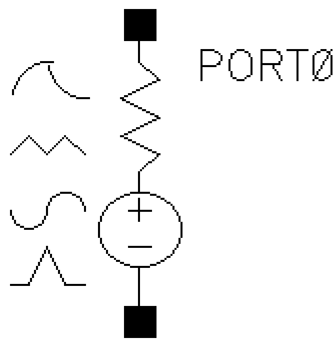
```
p20 (2 0) port num=2 r=50 type=pulse period=1e-9 rise=1e-10 fall=1e-10 val1=1
```

width=0.5n mag=1

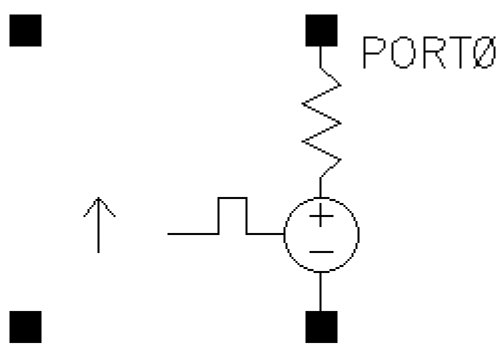
Additional Information

This device is not supported within the altergroups.

Symbol: port



If *Source type* = `prbs`, and *Trigger* = `External rising edge`, `External falling edge`, or `External both edges`, two extra ports are added to the port symbol as shown below:



You can specify the wave shape for the port by selecting one of the following options from the *Source type* drop-down list box in the Edit Object Properties form:

- *dc*—Generates a dc level from the port. When the Source type is set to dc, the dc and temperature effect parameters are active. The dc setting sets the DC level for all analyses.
- *sine*—Generates sinusoidal waveforms.

  Up to two sinusoids can be generated simultaneously. They are denoted as 1 and 2. You can set the amplitude, frequency, and phase for both individually. The amplitude can be set to either a voltage or a power level. When you set a power level, the assumption is that the port is perfectly matched. The source that is internal to the port gets double the amplitude specified by the power in `dBm`. You can also specify sinusoidal AM or FM modulation of sinusoid 1. Sinusoid 2 cannot be modulated.

- *pulse*—Generates a step, a single pulse, or a periodic pulse waveform.

  When you specify the voltage, you are specifying the voltage when the port is properly terminated, and not the voltage on the internal voltage source. Therefore, the voltage on the internal source is set to twice the value specified on the component.

- *exp*—Generates an exponential waveform. The exponential waveform can generate one exponential pulse, and cannot generate a periodic signal.

  When you specify the voltage, you are specifying the voltage when the port is properly terminated, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the port.

- *pwl*—Generates piecewise linear waveforms that allow an arbitrary input waveform to be generated.

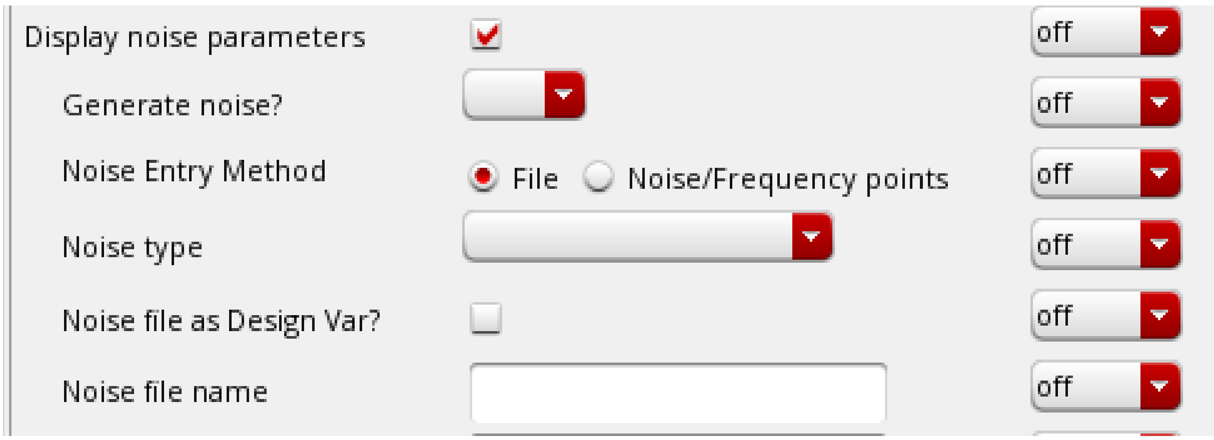
  The input can either be a file that contains time and voltage pairs, or you can enter the time-voltage pairs directly in the PWL source properties form. Remember that the voltages you enter in the piecewise linear file assumes that the port is properly terminated. The internal voltage source gets set to double the value specified in the piecewise linear voltage specifications.

- *pwlz*—Generates piecewise linear waveforms that allow an arbitrary input waveform to be generated. This source type resembles the pwl source type, except that some voltage values can be replaced by the high-impedance state. In addition to voltage-time pairs supported by pwl, pwlz also supports z-state in the waveform. When z-state is active, the voltage source is disconnected from the node and it is put in high-impedance state.

- *bit*—Generates bit sequence or string from the port. The bit source has four states: 1, 0, m, and z, which represent the high, low, middle voltage, and high impedance state respectively. It allows patterns defining a sequence of bits. When the m state is specified, the output voltage is set halfway between `0` state and `1` state voltages.

- *prbs*—PRBS is an acronym for Pseudo-Random Binary Sequence. This source has three modes. It can be used to generate a maximum-length pseudo-random sequence. You can specify the beginning state and tap gains for a Fibonacci PRBS generator. A third mode allows reading an ASCII file that describes the sequence of one and zero events to generate.

If you select the *Display noise parameters* check box in the Edit Object Properties form, the *Noise file as Design Var?* check box is displayed. You can select this check box to specify the noise file as a design variable in the *Noise file name* field.



For more information on the available source types, see the section *Source type* in the chapter *AnalogLib Components Used in RF Simulation* in the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Independent Resistive Source

For more information on this component refer to Appendix H of the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Command-line help

```
spectre -h port
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Resistance	`r`	x	-	-	-	-
Reactance	`x`	x	-	-	-	-
Port number	`num`	x	-	-	-	-
DC voltage	`vdc`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Source type	srcType	x	-	-	-	-
Frequency name 1	fundname	x	-	-	-	-
Frequency 1	freq	x	-	-	-	-
Amplitude 1 (Vpk)	va	x	-	-	-	-
PAC Magnitude (Vpk)	pacm	x	-	-	x	-
Amplitude 1 (dBm)	vaDBm	x	-	-	-	-
Phase for Sinusoid 1	sinephase	x	-	-	-	-
Sine DC level	sinedc	x	-	-	-	-
PJ(amplitude)	pjamp	x	-	-	-	-
PJ(frequency)	pjfreq	x	-	-	-	-
PJ(type)	pjtype	x	-	-	-	-
RJ(rms)	rjrms	x	-	-	-	-
RJ(seed)	rjseed	x	-	-	-	-
Number of PWL/Time pair	tvpairs	x	-	-	-	-
Time 1	t1 - t50	x	-	-	-	-
Voltage 1	v1 - v50	x	-	-	-	-
FM modulation index	fmmodindex	x	-	-	-	-
FM modulation frequency	fmmodfreq	x	-	-	-	-
AC Magnitude (Vpk)	acm	x	-	-	-	-
AM modulation index	ammodindex	x	-	-	-	-
AM modulation frequency	ammodfreq	x	-	-	-	-
AM modulation phase	ammodphase	x	-	-	-	-
Delay time	td	x	-	-	-	-
Amplitude scale factor	scale	x	-	-	-	-
Power of PWL waveform	pwldbm	x	-	-	-	-
Display second sinusoid	numofsines	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Display modulation params	`modulation`	x	-	-	-	-
Display small signal params	`smallSig`	x	-	-	-	-
Display temperature params	`tempParam`	x	-	-	-	-
Display noise parameters	`noiseParam`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Number of FM files	`filenums`	x	-	-	-	-
Name of FM File1	`fmmodfile1`	x	-	-	-	-
Name of FM File2	`fmmodfile2`	x	-	-	-	-
High-Z impedance	`highz`	x	-	-	-	-
Min high-Z trans. width	`min_z_transition_width`	x	-	-	-	-
PAM modulation	`pam4_modulation`	x	-	-	-	-
PAM4 mapping	`pam4_mapping`	x	-	-	-	-
Sinusoid Ampl 1 (Vpk) to Sinusoid Ampl 9 (Vpk)	`vav1 - vav9`	x	-	-	-	-
XF Magnitude (Vpk)	`xfm`	x	-	-	-	-
Z state 1 to Z state 50	`Z1 - Z50`	x	-	-	-	-

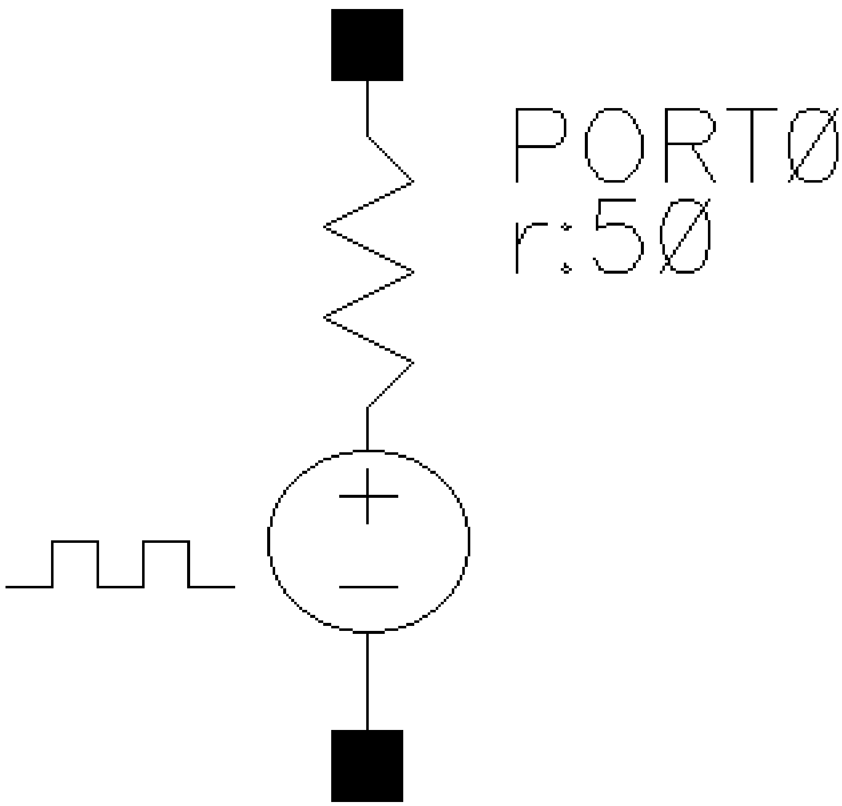
For more information on the jitter parameters: `pjamp`, `pjfreq`, `pjtype`, `rjrms`, and `rjseed` refer to Independent Current Source (isource) section in Spectre® Circuit Simulator Components and Device Models Reference

Additional Information

Power of PWL waveform (`pwldbm`) is an alternative to *Amplitude scale factor* (`scale`). Use `pwldbm` to specify the rms power for the waveform and spectre automatically calculates the correct scale factor.

If `pwldbm` is specified, it overwrites the `scale` parameter.

Symbol: ppulse



Independent Resistive Pulse Source

For more information on this component refer to Appendix H of the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Command-line help

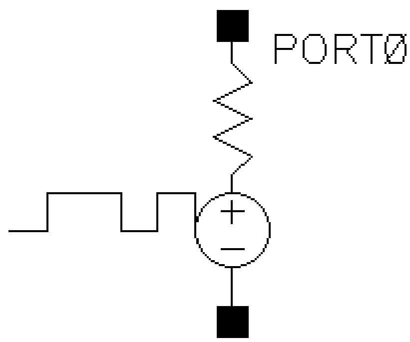
```
spectre -h port
```

CDF Parameters

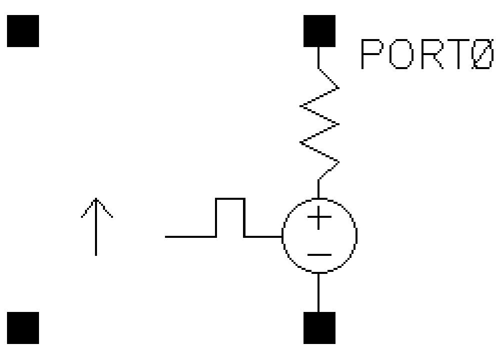
CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Frequency name for 1/period	`fundname`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/ freq pairs	`FNpairs`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Delay Time	`delay`	x	-	-	-	-
Resistance	`r`	x	-	-	-	-
Port number	`num`	x	-	-	-	-
DC voltage	`vdc`	x	-	-	-	-
Delay time	`td`	x	-	-	-	-
Voltage 1	`v1`	x	-	-	-	-
Voltage 2	`v2`	x	-	-	-	-
Period	`per`	x	-	-	-	-
Rise time	`tr`	x	-	-	-	-
Fall time	`tf`	x	-	-	-	-
Pulse width	`pw`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Nominal temperature	`tnom`	x	-	-	-	-
AC magnitude	`acm`	x	-	-	-	-
AC phase	`acp`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-

Symbol: pprbs



If *Trigger* = `External rising edge`, `External falling edge`, or `External both edges`, two extra ports are added to the pprbs symbol as shown below:



Independent Resistive Pulse Source

For more information on this component refer to Appendix H of the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Command-line help

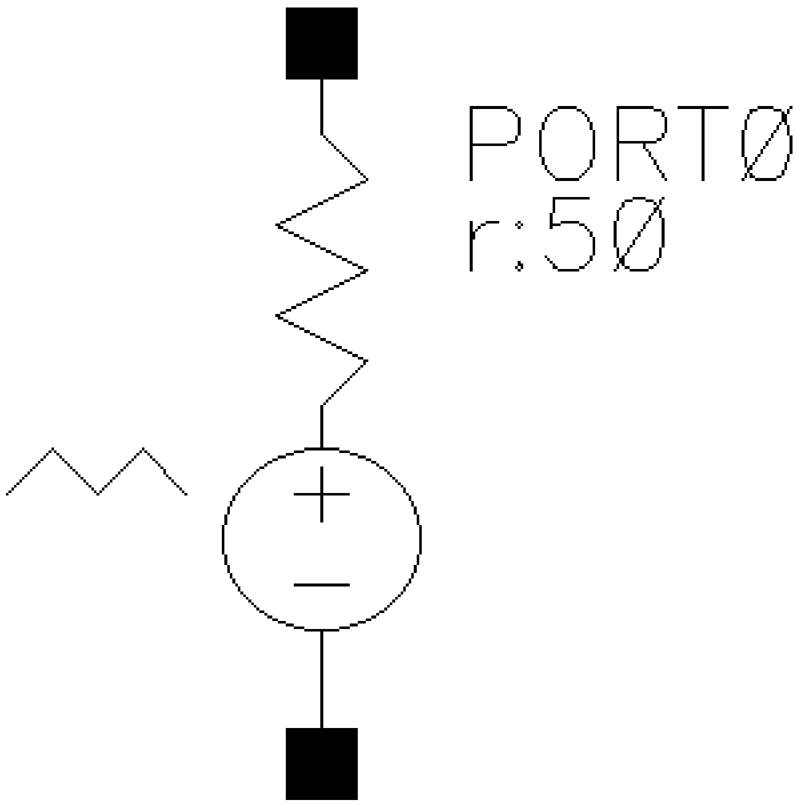
```
spectre -h port
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Bit string	`data`	x	-	-	-	-	-	-	-
Final value for logical 1	`val1`	x	-	-	-	-	-	-	-
Final value for logical 0	`val0`	x	-	-	-	-	-	-	-
Delay time	`delay`	x	-	-	-	-	-	-	-
Rise time	`rise`	x	-	-	-	-	-	-	-
Fall time	`fall`	x	-	-	-	-	-	-	-
Period of waveform	`period`	x	-	-	-	-	-	-	-
Resistance	`r`	x	-	-	-	-	-	-	-
Reference Value	`ref`	x	-	-	-	-	-	-	-
Waveform Random Delay Time	`jitter`	x	-	-	-	-	-	-	-
Generates Random Count	`seed`	x	-	-	-	-	-	-	-
Bit	`taps`	x	-	-	-	-	-	-	-
Multiplier	`m`	x	-	-	-	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Port number	num	x	-	-	-	-	-	-	-

Symbol: ppwl



Independent Piece-Wise Linear Resistive Source

For more information on this component refer to Appendix H of the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Command-line help

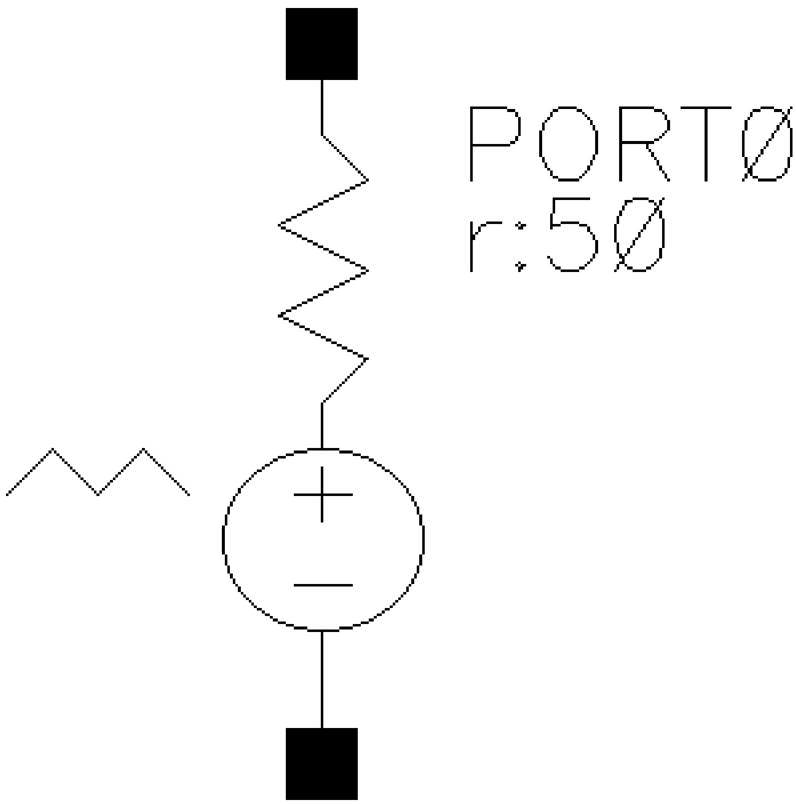
```
spectre -h port
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Frequency name for 1/ period	`fundname`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Number of pairs of points	`tvpairs`	x	-	-	-	-
Time 1	`t1 - t50`	x	-	-	-	-
Voltage 1	`v1 - v50`	x	-	-	-	-
Resistance	`r`	x	-	-	-	-
Port number	`num`	x	-	-	-	-
DC voltage	`vdc`	x	-	-	-	-
Delay time	`td`	x	-	-	-	-
Offset Voltage	`vo`	x	-	-	-	-
Scale factor	`scale`	x	-	-	-	-
Time scale factor	`stretch`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
All are breakpoints	`allbrkpts`	x	-	-	-	-
Period of the PWL	`pwlperiod`	x	-	-	-	-
Transition width	`twidth`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
AC magnitude	`acm`	x	-	-	-	-
AC phase	`acp`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Source type	`srcType`	x	-	-	-	-

Symbol: ppwlf



Independent Piece-Wise Linear Resistive Source Based on File

For more information on this component refer to Appendix H of the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Command-line help

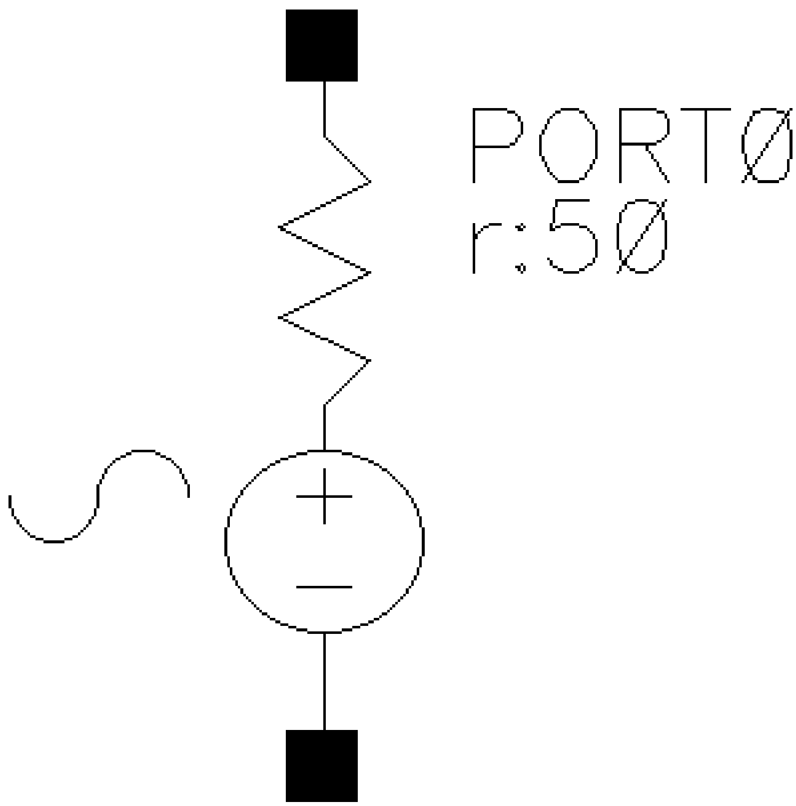
```
spectre -h port
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Frequency name for 1/ period	`fundname`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Resistance	`r`	x	-	-	-	-
Port number	`num`	x	-	-	-	-
DC voltage	`vdc`	x	-	-	-	-
Delay time	`td`	x	-	-	-	-
Offset voltage	`vo`	x	-	-	-	-
Scale factor	`scale`	x	-	-	-	-
Time scale factor	`stretch`	x	-	-	-	-
All are breakpoints	`allbrkpts`	x	-	-	-	-
Period of the PWL	`pwlperiod`	x	-	-	-	-
Transition width	`twidth`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
AC magnitude	`acm`	x	-	-	-	-
AC phase	`acp`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Source type	`srcType`	x	-	-	-	-

Symbol: psin



Independent Sinusoidal Resistive Source

The psin component is used in all RF circuits for SpectreRF and Spectre S-parameter simulations. When you netlist psin in the analog design environment using the Spectre simulator, you can see that psin is the port component in the Spectre simulation. A port is a resistive source that is tied between positive and negative terminals. It is equivalent to a voltage source in series with a resistor, and the reference resistance of the port is the value of the resistor.

For more information on this component refer to Appendix C of the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Command-line help

```
spectre -h port
```

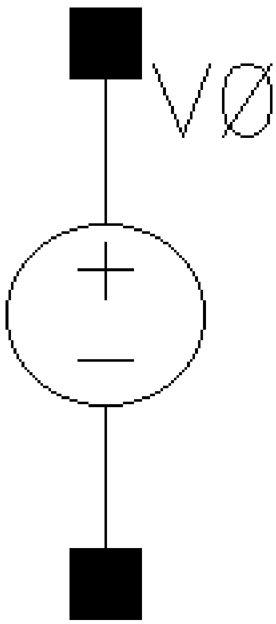
CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Frequency name	`fundname`	x	-	-	-	-
Second frequency name	`fundname2`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Number of FM files	`filenums`	x	-	-	-	-
Name of FM File1	`fmmodfile1`	x	-	-	-	-
Name of FM File2	`fmmodfile2`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Resistance	`r`	x	-	-	-	-
Port number	`num`	x	-	-	-	-
DC voltage	`vdc`	x	-	-	-	-
Source type	`srcType`	x	-	-	-	-
Delay time	`td`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Sine DC level	sinedc	x	-	-	-	-
Amplitude	va	x	-	-	-	-
Amplitude (dBm)	vaDBm	x	-	-	-	-
Initial phase for Sinusoid	sinephase	x	-	-	-	-
Frequency	freq	x	-	-	-	-
Amplitude 2 (Vpk)	va2	x	-	-	-	-
Amplitude 2 (dBm)	vaDBm2	x	-	-	-	-
Initial phase for Sinusoid 2	sinephase2	x	-	-	-	-
Frequency 2	freq2	x	-	-	-	-
FM modulation index	fmmodindex	x	-	-	-	-
FM modulation frequency	fmmodfreq	x	-	-	-	-
AM modulation index	ammodindex	x	-	-	-	-
AM modulation frequency	ammodfreq	x	-	-	-	-
AM modulation phase	ammodphase	x	-	-	-	-
Damping factor	theta	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Multiplier	`m`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
Noise temperature	`noisetemp`	x	-	-	-	-
AC magnitude	`acm`	x	-	-	-	-
AC phase	`acp`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC magnitude (dBm)	`pacmDBm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-

Symbol: vdc



Independent Voltage Source

vdc is a constant vsource.

Command-line help

```
spectre -h vsource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	acm	x	-	-	x	x
AC phase	acp	x	-	-	x	x
DC voltage	vdc	x	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Noise file name	noiseFile	x	-	-	-	-
Number of noise/freq pairs	FNpairs	x	-	-	-	-
Freq 1 to Freq 50	F1 - F50	x	-	-	-	-
Noise 1 to Noise 50	N1 - N50	x	-	-	-	-
XF magnitude	xfm	x	-	-	-	-
PAC magnitude	pacm	x	-	-	-	-
PAC phase	pacp	x	-	-	-	-
Temperature coefficient 1	tc1	x	-	-	-	-
Temperature coefficient 2	tc2	x	-	-	-	-
Nominal temperature	tnom	x	-	-	-	-
Source type	srcType	x	-	-	-	-
AC Phase	acPhase	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) vsource <parameter=value> ...
```

Example

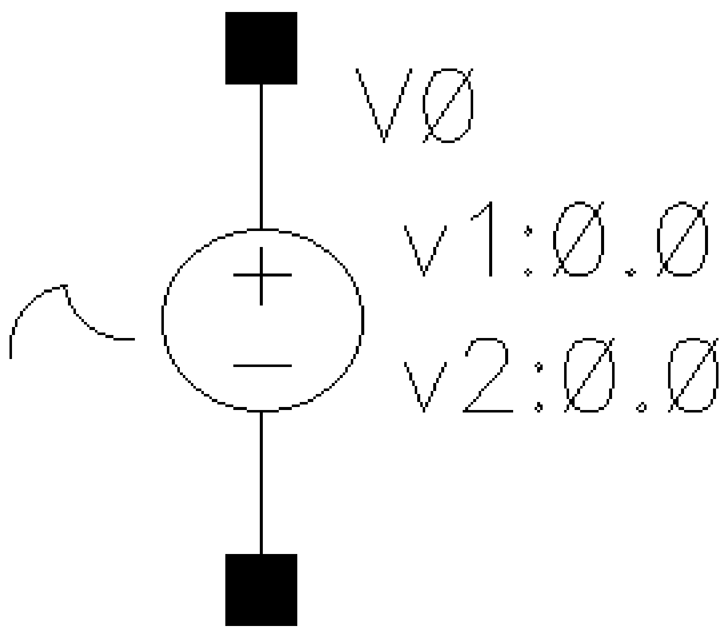
```
vpulse1 (1 0) vsource type=pulse val0=0 val1=5 period=100n rise=10n fall=10n
width=40n
```

```
vpwl1 (1 0) vsource type=pwl wave=[1n 0 1.1n 2 1.5n 0.5 2n 3 5n 5] pwlperiod=5n
```

Additional Information

This device is supported within the altergroups.

Symbol: vexp



Independent Exponential Voltage Source

vexp is an exponential vsource.

Command-line help

```
spectre -h vsource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	x	-	-	x	x
AC phase	`acp`	x	-	-	x	x
DC voltage	`vdc`	x	-	-	-	-
Voltage 1	`v1`	x	-	-	x	x
Voltage 2	`v2`	x	-	-	x	x
Delay time 1	`td1`	x	-	-	x	x
Damping factor 1	`tau1`	x	-	-	x	x
Delay time 2	`td2`	x	-	-	x	x
Damping factor 2	`tau2`	x	-	-	x	x
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Delay time	`td`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Temperature coefficient 2	tc2	x	-	-	-	-
Nominal temperature	tnom	x	-	-	-	-
DC source	dc	-	-	-	x	x
Freq 1 to Freq 50	F1 - F50	x	-	-	-	-
Noise 1 to Noise 50	N1 - N50	x	-	-	-	-
Delay Time	delay	x	-	-	-	-
Source type	srcType	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) vsource <parameter=value> ...
```

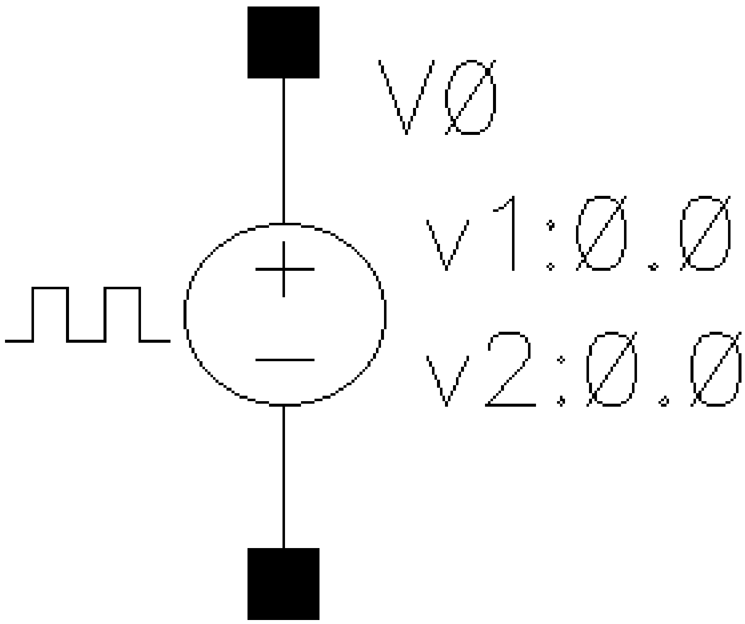
Example

```
vpulse1 (1 0) vsource type=pulse val0=0 val1=5 period=100n rise=10n fall=10n
width=40n
vpwl1 (1 0) vsource type=pwl wave=[1n 0 1.1n 2 1.5n 0.5 2n 3 5n 5] pwlperiod=5n
```

Additional Information

This device is supported within the altergroups.

Symbol: vpulse



Independent Pulse Voltage Source

vpulse is a square wave varying vsource.

Command-line help

```
spectre -h vsource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	x	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC phase	acp	x	-	-	x	x
DC voltage	vdc	x	-	-	-	-
Voltage 1	v1	x	-	-	x	x
Voltage 2	v2	x	-	-	x	x
Delay time	td	x	-	-	x	x
Type of rising & falling edge	edgetype	x	-	-	-	-
Rise time	tr	x	-	-	x	x
Fall time	tf	x	-	-	x	x
Pulse width	pw	x	-	-	x	x
Period	per	x	-	-	x	x
Frequency name for 1/period	fundname	x	-	-	-	-
Noise file name	noisefile	x	-	-	-	-
Number of noise/freq pairs	FNpairs	x	-	-	-	-
Freq 1 to Freq 50	F1 - F50	x	-	-	-	-
Noise 1 to Noise 50	N1 - N50	x	-	-	-	-
XF magnitude	xfm	x	-	-	-	-
PAC magnitude	pacm	x	-	-	-	-
PAC phase	pacp	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
DC source	`dc`	-	-	-	x	x
Source type	`srcType`	x	-	-	-	-
Delay Time	`delay`	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) vsource <parameter=value> ...
```

Example

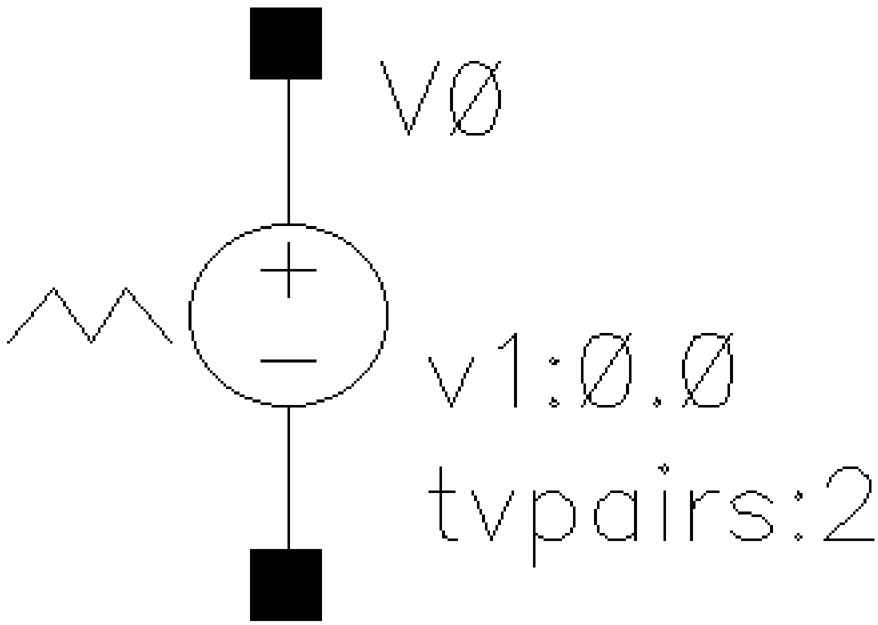
```
vpulse1 (1 0) vsource type=pulse val0=0 val1=5 period=100n rise=10n fall=10n
width=40n
```

```
vpwl1 (1 0) vsource type=pwl wave=[1n 0 1.1n 2 1.5n 0.5 2n 3 5n 5] pwlperiod=5n
```

Additional Information

This device is supported within the altergroups.

Symbol: vpwl



Independent Piece-Wise Linear Voltage Source

vpwl is a piece-wise linear vsource.

Command-line help

```
spectre -h vsource
```

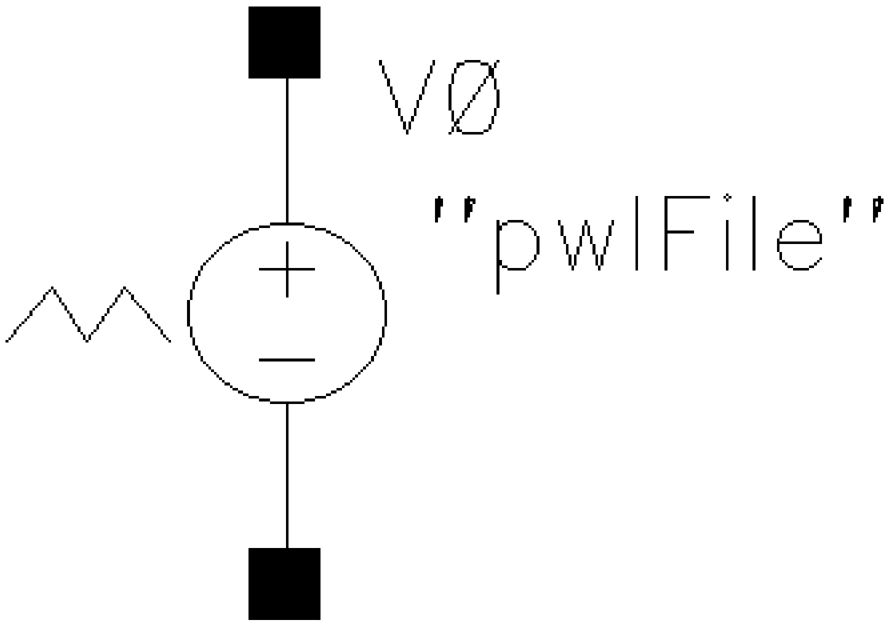
CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Number of pairs of points	`tvpairs`	x	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	x	-	-	x	x
AC phase	`acp`	x	-	-	x	x
DC voltage	`vdc`	x	-	-	-	-
Time 1	`t1 - t50`	x	-	-	x	x
Voltage 1	`v1 - v50`	x	-	-	x	x
Frequency name for 1/ period	`fundname`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Delay time	`td`	x	-	-	x	x
Type of rising & falling edge	`edgetype`	x	-	-	-	-
Offset voltage	`vo`	x	-	-	-	-
Scale factor	`scale`	x	-	-	-	-
Time scale factor	`stretch`	x	-	-	-	-
Period of the PWL	`pwlperiod`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Transition width	`twidth`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
DC source	`dc`	-	-	-	x	x
Repeated function	`rpt`	-	-	-	x	x
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Delay Time	`delay`	x	-	-	-	-
Source type	`srcType`	x	-	-	-	-

Symbol: vpwlf



Independent Piece-Wise Linear Voltage Source Based on File

Command-line help

```
spectre -h vsource
```

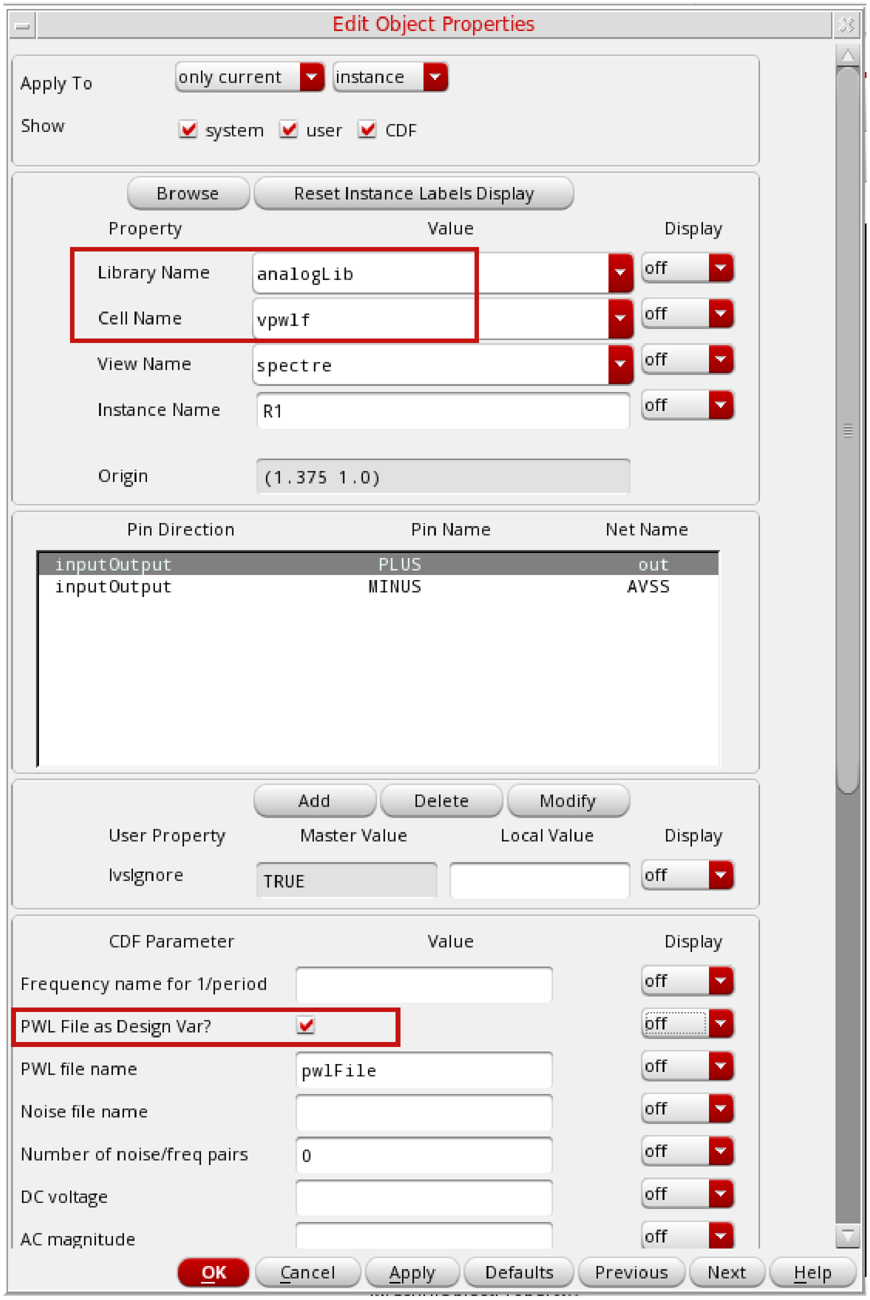
CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	x	-	-	x	-
AC phase	`acp`	x	-	-	x	-
DC voltage	`vdc`	x	-	-	x	-

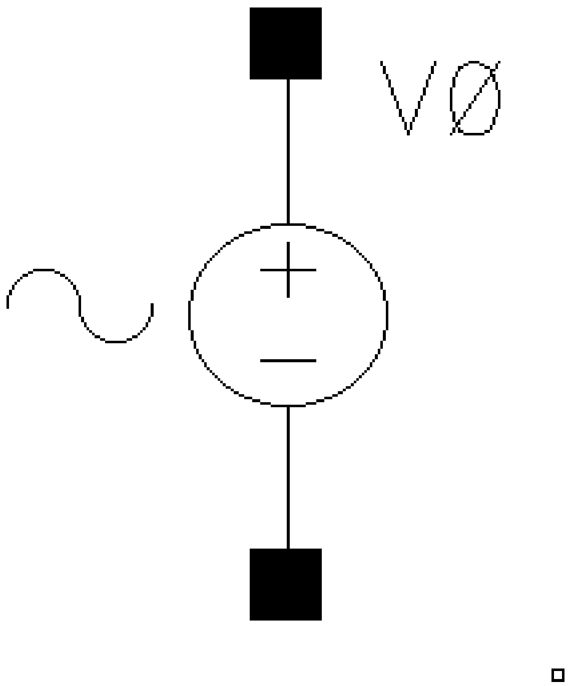
CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
PWL file name	fileName	x	-	-	x	-
Frequency name for 1/period	fundname	x	-	-	-	-
Noise file name	noisefile	x	-	-	-	-
Number of noise/freq pairs	FNpairs	x	-	-	-	-
XF magnitude	xfm	x	-	-	-	-
PAC magnitude	pacm	x	-	-	-	-
PAC phase	pacp	x	-	-	-	-
Delay time	td	x	-	-	-	-
Type of rising & falling edge	edgetype	x	-	-	-	-
Offset voltage	vo	x	-	-	-	-
Scale factor	scale	x	-	-	-	-
Time scale factor	stretch	x	-	-	-	-
Period of the PWL	pwlperiod	x	-	-	-	-
Transition width	twidth	x	-	-	-	-
Temperature coefficient 1	tc1	x	-	-	-	-
Temperature coefficient 2	tc2	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Nominal temperature	tnom	x	-	-	-	-
Freq 1 to Freq 50	F1 - F50	x	-	-	-	-
Noise 1 to Noise 50	N1 - N50	x	-	-	-	-
Delay Time	delay	x	-	-	-	-
Source type	srcType	x	-	-	-	-

If you select `vsource` or `vpwlf` from the *Cell Name* drop-down list of the Edit Object Properties form, you can select the *PWL file as Design Var?* check box to specify the PWL data file as a design variable in the *PWL file name* field.


Edit Object Properties
Apply To
only current
instance
Show
system
user
CDF
Browse
Reset Instance Labels Display
Property
Value
Display
Library Name
analogLib
Cell Name
vpwlf
View Name
spectre
Instance Name
R1
Origin
(1.375 1.0)
Pin Direction
Pin Name
Net Name
inputOutput PLUS out
inputOutput MINUS AVSS
Add
Delete
Modify
User Property
Master Value
Local Value
lvsIgnore
TRUE
CDF Parameter
Frequency name for 1/period
PWL File as Design Var?
PWL file name
pwlFile
Noise file name
Number of noise/freq pairs
0
DC voltage
AC magnitude
off
OK
Cancel
Apply
Defaults
Previous
Next
Help

Symbol: vsin



Independent Sinusoidal Voltage Source

vsin is a sin wave vsource.

Command-line help

```
spectre -h vsource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	x	-	-	x	x
AC phase	`acp`	x	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
DC voltage	`vdc`	x	-	-	-	-
Offset voltage	`vo`	x	-	-	x	x
Amplitude	`va`	x	-	-	x	x
Frequency	`freq`	x	-	-	x	x
Delay time	`td`	x	-	-	x	x
Damping factor	`theta`	x	-	-	x	x
First frequency name	`fundname`	x	-	-	-	-
Second frequency name	`fundname2`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Number of FM files	`filenums`	x	-	-	-	-
Name of FM File1	`fmmodfile1`	x	-	-	-	-
Name of FM File2	`fmmodfile2`	x	-	-	-	-
XF magnitude	`xfm`	x	-	-	-	-
PAC magnitude	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
Initial phase for Sinusoid	`sinephase`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Amplitude 2 (Vpk)	`va2`	x	-	-	-	-
Initial phase for Sinusoid 2	`sinephase2`	x	-	-	-	-
Frequency 2	`freq2`	x	-	-	-	-
FM modulation index	`fmmodindex`	x	-	-	-	-
FM modulation frequency	`fmmodfreq`	x	-	-	-	-
AM modulation index	`ammodindex`	x	-	-	-	-
AM modulation frequency	`ammodfreq`	x	-	-	-	-
AM modulation phase	`ammodphase`	x	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
DC source	`dc`	-	-	-	x	x
Phase delay	`phi`	-	-	-	x	x
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Delay Time	`delay`	x	-	-	-	-
Source type	`srcType`	x	-	-	-	-
Sine DC level	`sinedc`	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) vsource <parameter=value> ...
```

Example

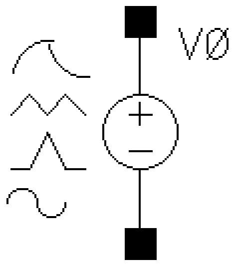
```
vpulse1 (1 0) vsource type=pulse val0=0 val1=5 period=100n rise=10n fall=10n
width=40n
```

```
vpwl1 (1 0) vsource type=pwl wave=[1n 0 1.1n 2 1.5n 0.5 2n 3 5n 5] pwlperiod=5n
```

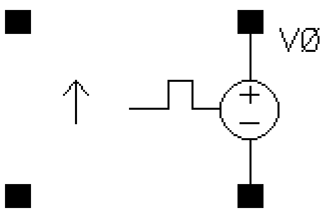
Additional Information

This device is supported within the altergroups.

Symbol: vsource



If *Source type* = `prbs`, and *Trigger* = `External rising edge`, `External falling edge`, or `External both edges`, two extra ports are added to the vsource symbol as shown below:



You can specify the wave shape for vsource by selecting one of the following options from the *Source type* drop-down list box in the Edit Object Properties form:

- *dc*—Generates a dc level from vsource. When the Source type is set to dc, the dc and temperature effect parameters are active. The dc setting sets the DC level for all analyses.
- *sine*—Generates sinusoidal waveforms.

  Up to two sinusoids can be generated simultaneously. They are denoted as 1 and 2. You can set the amplitude, frequency, and phase for both individually. The amplitude can be set to either a voltage or a power level. When you set a power level, the assumption is that the vsource is perfectly matched. The source that is internal to vsource gets double the amplitude specified by the power in `dBm`. You can also specify sinusoidal AM or FM modulation of sinusoid 1. Sinusoid 2 cannot be modulated.
- *pulse*—Generates a step, a single pulse, or a periodic pulse waveform.

  When you specify the voltage, you are specifying the voltage when vsource is properly terminated, and not the voltage on the internal voltage source. Therefore, the voltage on the internal source is set to twice the value specified on the component.

- *exp*—Generates an exponential waveform. The exponential waveform can generate one exponential pulse, and cannot generate a periodic signal.

  When you specify the voltage, you are specifying the voltage when vsource is properly terminated, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on vsource.

- *pwl*—Generates piecewise linear waveforms that allow an arbitrary input waveform to be generated.

  The input can either be a file that contains time and voltage pairs, or you can enter the time-voltage pairs directly in the PWL source properties form. Remember that the voltages you enter in the piecewise linear file assumes that the vsource is properly terminated. The internal voltage source gets set to double the value specified in the piecewise linear voltage specifications.

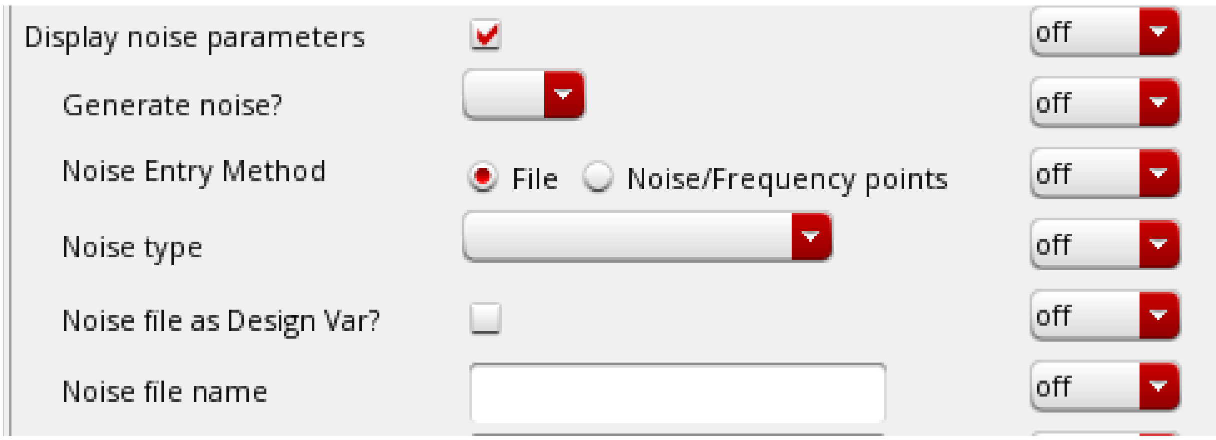
- *pwlz*—Generates piecewise linear waveforms that allow an arbitrary input waveform to be generated. This source type resembles the pwl source type, except that some voltage values can be replaced by the high-impedance state. In addition to voltage-time pairs supported by pwl, pwlz also supports z-state in the waveform. When z-state is active, the voltage source is disconnected from the node and it is put in high-impedance state.

- *bit*—Generates bit sequence or string from vsource. The bit source has four states: `1`, `0`, `m`, and `z`, which represent the high, low, middle voltage, and high impedance state respectively. When the `m` state is specified, the output voltage is set halfway between `0` state and `1` state voltages. This source type lets you create simple or nested patterns defining a sequence of bits.

  Note: Nested patterns are supported only for Spectre.

- *prbs*—PRBS is an acronym for Pseudo-Random Binary Sequence. This source has three modes. It can be used to generate a maximum-length pseudo-random sequence. You can specify the beginning state and tap gains for a Fibonacci PRBS generator. A third mode allows reading an ASCII file that describes the sequence of one and zero events to generate.

If you select the *Display noise parameters* check box in the Edit Object Properties form, the *Noise file as Design Var?* check box is displayed. You can select this check box to specify the noise file as a design variable in the *Noise file name* field.



For more information on the available source types, see the section *Source type* in the chapter *AnalogLib Components Used in RF Simulation* in the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Independent Voltage Source

Current through the source is computed and is defined to be positive if it flows from the positive node, through the source, to the negative node.

The value of the DC voltage as a function of the temperature is given by:

```
V(T) = V(tnom) * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2].
```

Command-line help

```
spectre -h vsource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
DC voltage	`vdc`	x	-	-	x	-
Source type	`srcType`	x	-	-	x	-
Frequency name 1	`fundname`	x	-	-	-	-
Frequency 1	`freq`	x	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Amplitude 1 (Vpk)	`va`	x	-	-	x	x
Phase for Sinusoid 1	`sinephase`	x	-	-	-	-
Sine DC level	`sinedc`	x	-	-	x	-
Browse and select file	`selectFile`	x	-	-	x	-
File name	`fileName`	x	-	-	x	-
Number of PWL/Time pair	`tvpairs`	x	-	-	x	-
Sinusoid Ampl 1 (Vpk) to Sinusoid Ampl 9 (Vpk)	`vav1 - vav9`	x	-	-	-	-
Time 1	`t1 - t50`	x	-	-	x	-
Voltage 1	`v1 - v50`	x	-	-	x	-
Delay time	`td`	x	-	-	x	x
Type of rising & falling edge	`edgetype`	x	-	-	x	-
Pattern Parameter Data	`data`	x	-	-	x	-
Rise time start	`td1`	x	-	-	x	-
Rise time constant	`tau1`	x	-	-	x	-
Fall time start	`td2`	x	-	-	x	-
Fall time constant	`tau2`	x	-	-	x	-
DC offset	`offset`	x	-	-	-	-
Amplitude scale factor	`scale`	x	-	-	-	-
Time scale factor	`stretch`	x	-	-	-	-
Breakpoints	`allbrkpts`	x	-	-	-	-
Period of waveform	`per`	x	-	-	x	-
FM modulation index 1	`fmmodindex`	x	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
FM modulation frequency 1	`fmmodfreq`	x	-	-	x	-
AM modulation index 1	`ammodindex`	x	-	-	x	-
AM modulation frequency 1	`ammodfreq`	x	-	-	x	-
AM modulation phase 1	`ammodphase`	x	-	-	-	-
Display second sinusoid	`numofsines`	x	-	-	-	-
Damping factor 1	`theta`	x	-	-	x	-
Display small signal params	`smallSig`	x	-	-	x	-
PAC Magnitude (Vpk)	`pacm`	x	-	-	-	-
PAC phase	`pacp`	x	-	-	-	-
AC Magnitude (Vpk)	`acm`	x	-	-	x	-
AC phase	`acp`	x	-	-	x	-
XF Magnitude (Vpk)	`xfm`	x	-	-	-	-
Display noise parameters	`noiseParam`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Noise 1 to Noise 50	`N1 - N50`	x	-	-	-	-
Display modulation params	`modulation`	x	-	-	x	-
Display temperature params	`tempParam`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Linear temp. coefficient	`tc1`	x	-	-	-	-
Quadratic temp. coeff.	`tc2`	x	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-
DC source	`dc`	-	-	-	x	x
Offset voltage	`vo`	-	-	-	x	x
Phase delay	`phi`	-	-	-	x	x
Repeated function	`rpt`	-	-	-	-	x
Period	`pwlperiod`	x	-	-	x	-
PAM modulation	`pam4_modulation`	x	-	-	-	-
PAM4 mapping	`pam4_mapping`	x	-	-	-	-
Period start time	`pwlperiodstart`	x	-	-	x	-
Transition width	`twidth`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-
Delay Time	`delay`	x	-	-	-	-
Number of FM files	`filenums`	x	-	-	-	-
Name of FM File1	`fmmodfile1`	x	-	-	-	-
Name of FM File2	`fmmodfile2`	x	-	-	-	-
Reference Value	`ref`	x	-	-	-	-
Remove Device	`lxRemoveDevice`	-	x	-	-	-
RJ(seed)	`rjseed`	x	-	-	-	-
RJ(rms)	`rjrms`	x	-	-	-	-
PJ(amplitude)	`pjamp`	x	-	-	-	-
PJ(frequency)	`pjfreq`	x	-	-	-	-
PJ(type)	`pjtype`	x	-	-	-	-
Taps	`lfsrtaps`	x	-	-	x	-
Seed	`lfsrseed`	x	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Seed	`seed`	x	-	-	x	-
Transition reference	`transitionreference`	x	-	-	-	-
LFSR Mode	`lfsrmode`	x	-	-	x	-
Threshold	`triggerthreshold`	x	-	-	-	-
Rise Delay	`td01`	x	-	-	x	-
Fall Delay	`td10`	x	-	-	x	-
High-Z impedance	`highz`	x	-	-	-	-
Min high-Z trans. width	`min_z_transition_width`	x	-	-	-	-
Z state 1 to Z state 50	`Z1 - Z50`	x	-	-	-	-

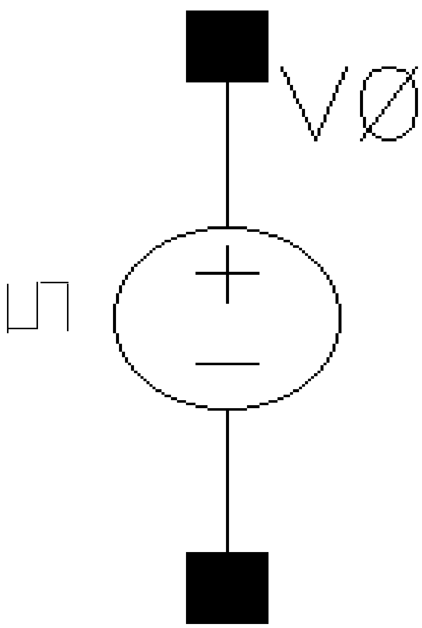
Note: For HspiceD, parameter `pwlperiod` is supported under the following conditions:

- ❑ In case `pwlperiod` is specified and `pwlperiodstart` is not specified, then another voltage-time pair must be added, where time = `pwlperiod` and voltage is the same as the voltage in the last voltage-time pair.

  But, if the value specified for `pwlperiod` is the same as the time specified in the last voltage-time pair, then no additional voltage-time pair is required.

- ❑ In case both `pwlperiod` and `pwlperiodstart` are specified, then another voltage-time pair must be added, where time = (`pwlperiod + pwlperiodstart`) and voltage is the same as the voltage in the last voltage-time pair.

Symbol: vbit



Independent Voltage Source

Command-line help

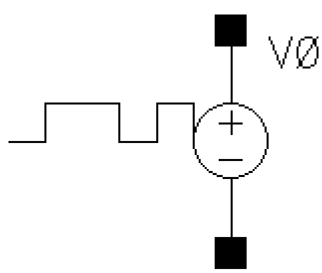
```
spectre -h vsource
```

CDF Parameters

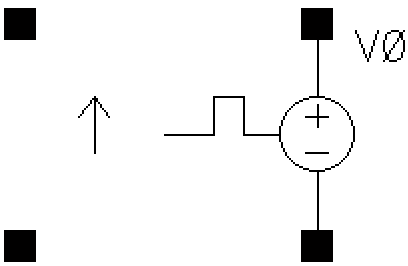
CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Bit string	`data`	x	-	-	-	-	-	-	-
Starting bit when repeating	`rptstart`	x	-	-	-	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Repeat times	`rpttimes`	x	-	-	-	-	-	-	-
Final value for logical 1	`val1`	x	-	-	-	-	-	-	-
Final value for logical 0	`val0`	x	-	-	-	-	-	-	-
Delay time	`delay`	x	-	-	-	-	-	-	x
Rise time	`rise`	x	-	-	-	-	-	-	-
Fall time	`fall`	x	-	-	-	-	-	-	-
Period of waveform	`period`	x	-	-	-	-	-	-	-
Source type	`type`	x	-	-	-	-	-	-	-

Symbol: vprbs



If *Trigger* = `External rising edge`, `External falling edge`, or `External both edges`, two extra ports are added to the vprbs symbol as shown below:



Independent Voltage Source

Command-line help

`spectre -h vsource`

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Bit string	`data`	x	-	-	-	-	-	-	-
Final value for logical 1	`val1`	x	-	-	-	-	-	-	-
Final value for logical 0	`val0`	x	-	-	-	-	-	-	-
Delay time	`delay`	x	-	-	-	-	-	-	-
Rise time	`rise`	x	-	-	-	-	-	-	-
Fall time	`fall`	x	-	-	-	-	-	-	-
Period of waveform	`period`	x	-	-	-	-	-	-	-
Reference Value	`ref`	x	-	-	-	-	-	-	-

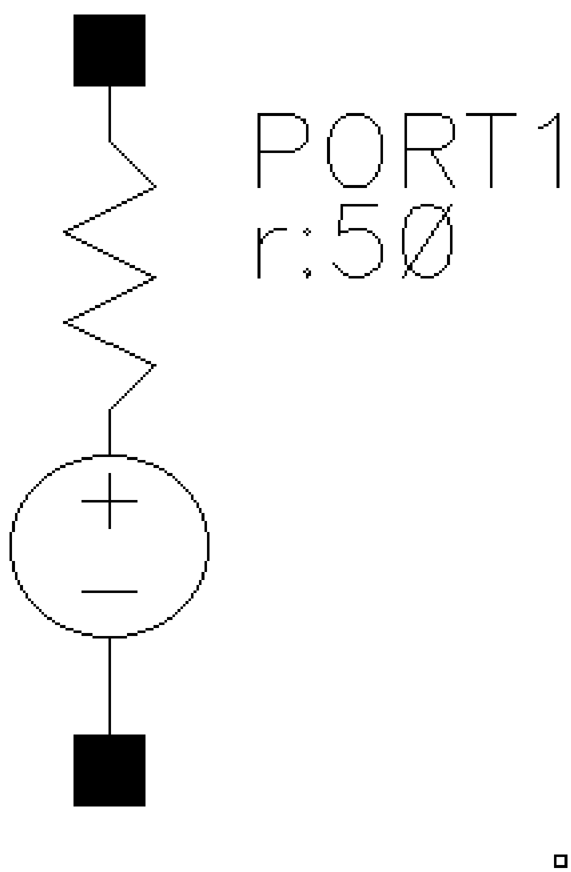
CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Waveform Random Delay Time	`jitter`	x	-	-	-	-	-	-	-
Generates Random Count	`seed`	x	-	-	-	-	-	-	-
Bit	`taps`	x	-	-	-	-	-	-	-

9

Ports

Sources

Symbol: pdc

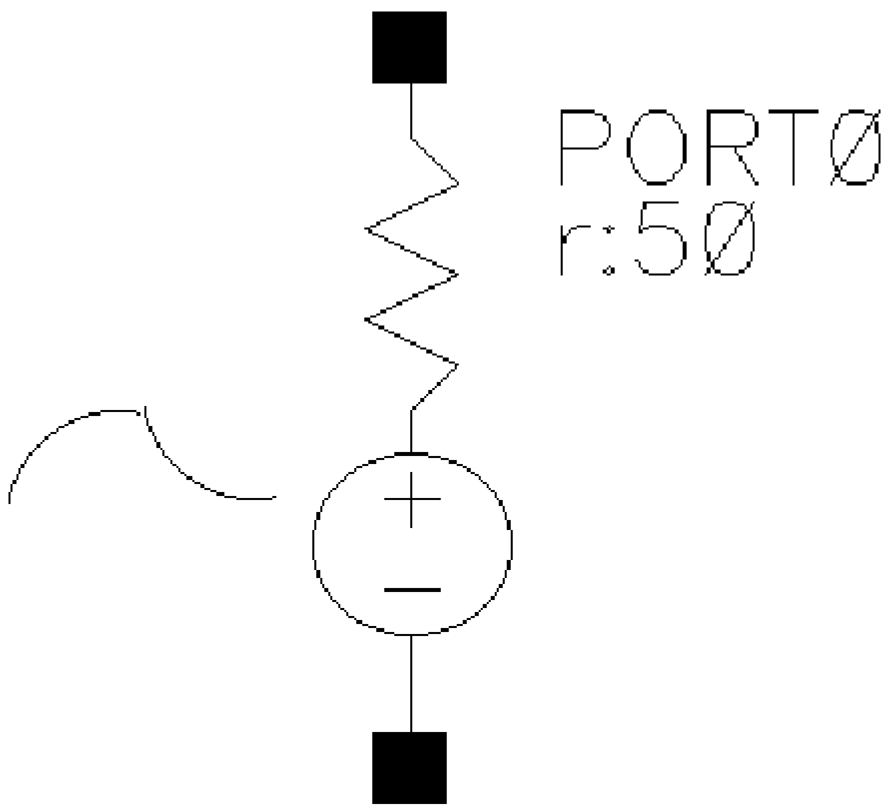


Independent DC Resistive Source

This component is the same as pdc described in the Chapter 8, “Symbol: pdc.”

For more information on ports refer to *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Symbol: pexp

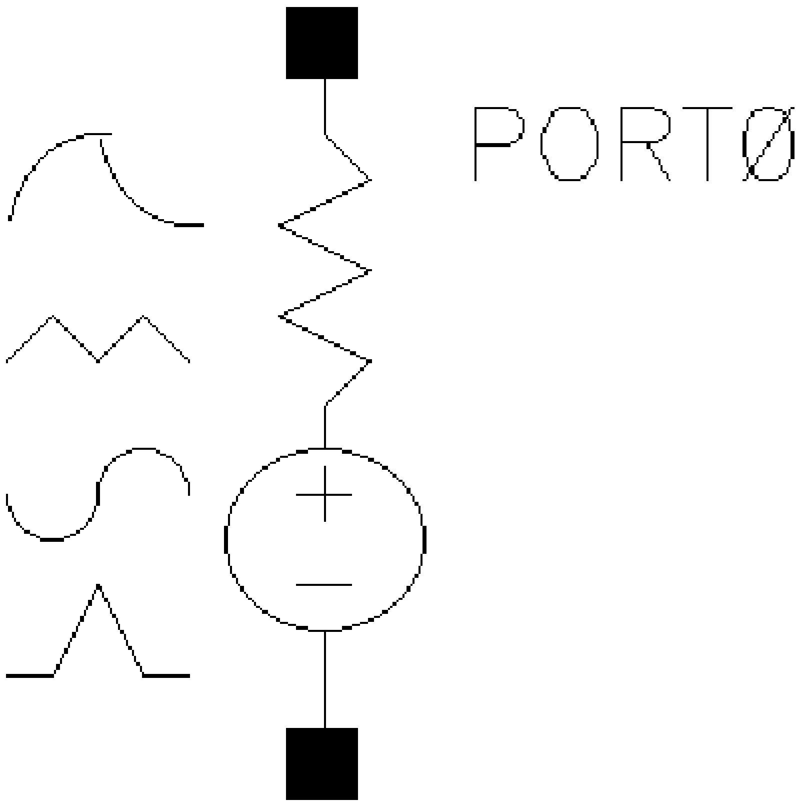


Independent Exponential Resistive Source

This component is the same as pexp described in the Chapter 8, “Symbol: pexp.”

For more information on ports refer to *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Symbol: port

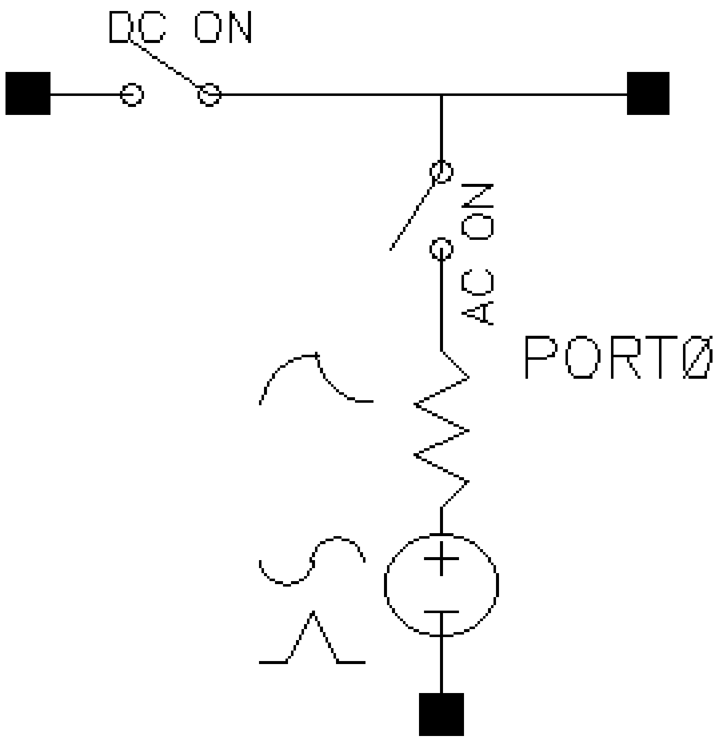


Independent Resistive Source

This component is the same as port described in the Chapter 8, "Symbol: port."

For more information on ports refer to *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Symbol: port3t



Independent Resistive Source

You can define a three-terminal independent resistive source with an ideal choke inductor and an ideal blocking capacitor. They work like switches to terminate or connect appropriate branch depending on the type of analysis.

For more information on ports refer to *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

Command-line help

```
spectre -h port
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Frequency name 1	`fundname`	x	-	-	-	-
Frequency name 2	`fundname 2`	x	-	-	-	-
Noise file name	`noisefile`	x	-	-	-	-
File name	`fileName`	x	-	-	-	-
Display second sinusoid	`numofsines`	x	-	-	-	-
Display modulation params	`modulation`	x	-	-	-	-
Display small signal params	`smallSig`	x	-	-	-	-
Display temperature params	`tempParam`	x	-	-	-	-
Display noise parameters	`noiseParam`	x	-	-	-	-
Number of noise/freq pairs	`FNpairs`	x	-	-	-	-
Freq 1 to Freq 50	`F1 - F50`	x	-	-	-	-
Number of PWL/Time pair	`tvpairs`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Time 1	`t1 - t50`	x	-	-	-	-
Voltage 1	`v1 - v50`	x	-	-	-	-
Number of FM files	`filenums`	x	-	-	-	-
Name of FM File1	`fmmodfile1`	x	-	-	-	-
Name of FM File2	`fmmodfile2`	x	-	-	-	-
Resistance	`r`	x	-	-	-	-
Reactance	`x`	x	-	-	-	-
Choke ind for net analyser	`lchock`	x	-	-	-	-
Blocking cap for net analyser	`cblock`	x	-	-	-	-
Port number	`num`	x	-	-	-	-
DC voltage	`vdc`	x	-	-	-	-
Source type	`srcType`	x	-	-	-	-
Delay time	`td`	x	-	-	-	-
Frequency 1	`freq`	x	-	-	-	-
Amplitude 1 (Vpk)	`va`	x	-	-	-	-
Amplitude 1 (dBm)	`vaDBm`	x	-	-	-	-
Phase for Sinusoid 1	`sinephase`	x	-	-	-	-
Sine DC level	`sinedc`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
FM modulation index	fmmodindex	x	-	-	-	-
FM modulation frequency	fmmodfreq	x	-	-	-	-
AM modulation index	ammodindex	x	-	-	-	-
AM modulation frequency	ammodfreq	x	-	-	-	-
AM modulation phase	ammodphase	x	-	-	-	-
PAC magnitude	pacm	x	-	-	-	-
PAC phase	pacp	x	-	-	-	-
Power of PWL waveform	pwldbm	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n [choke]... ) port <parameter=value> ...
```

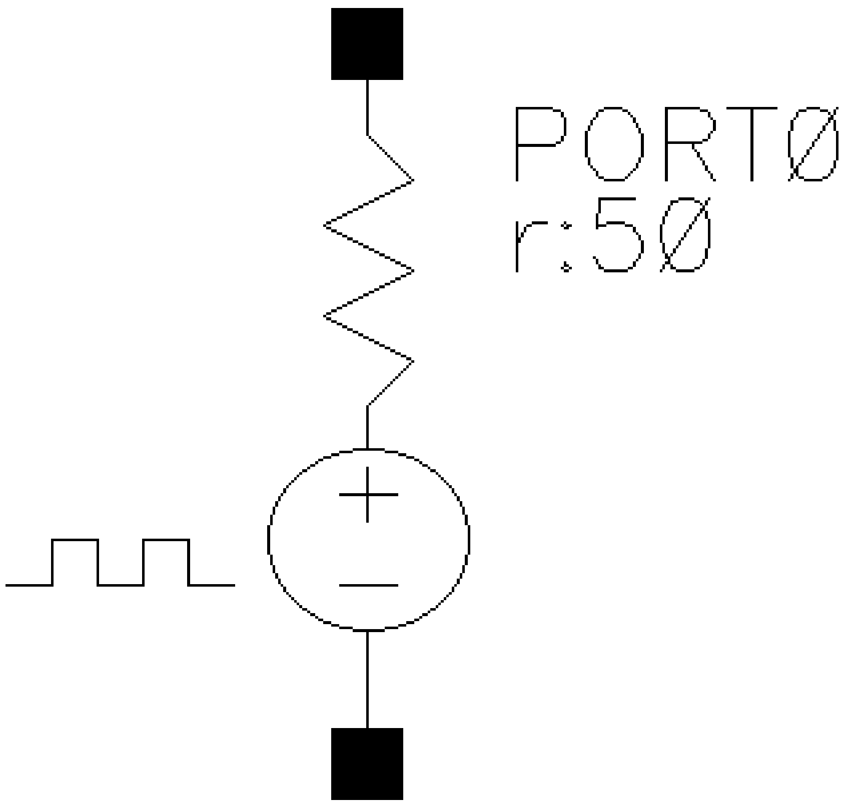
Sample Instance Statement

```
p20 (2 0) port num=2 r=50 type=pulse period=1e-9 rise=1e-10 fall=1e-10 val1=1
width=0.5n mag=1
p30 (2 0 choke) port num=1 r=50 lchoke=0.1 cblock=0.00001 type=pulse period=1e-8
rise=1e-8 fall=1e-10
```

Additional Information

This device is not supported within altergroup.

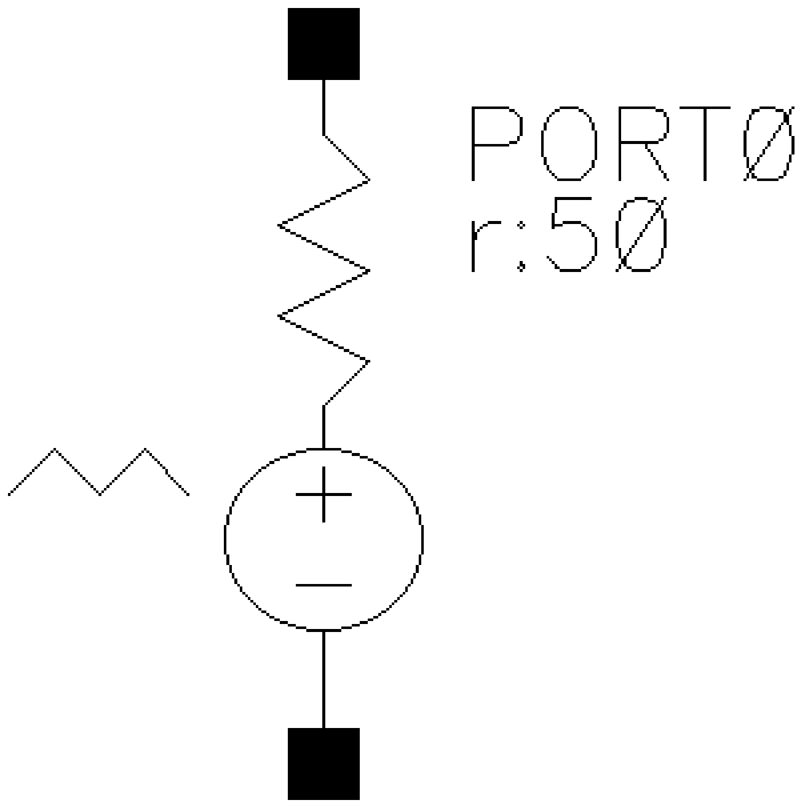
Symbol: ppulse



Independent Resistive Pulse Source

This component is the same as ppulse described in the Chapter 8, “Symbol: ppulse,”

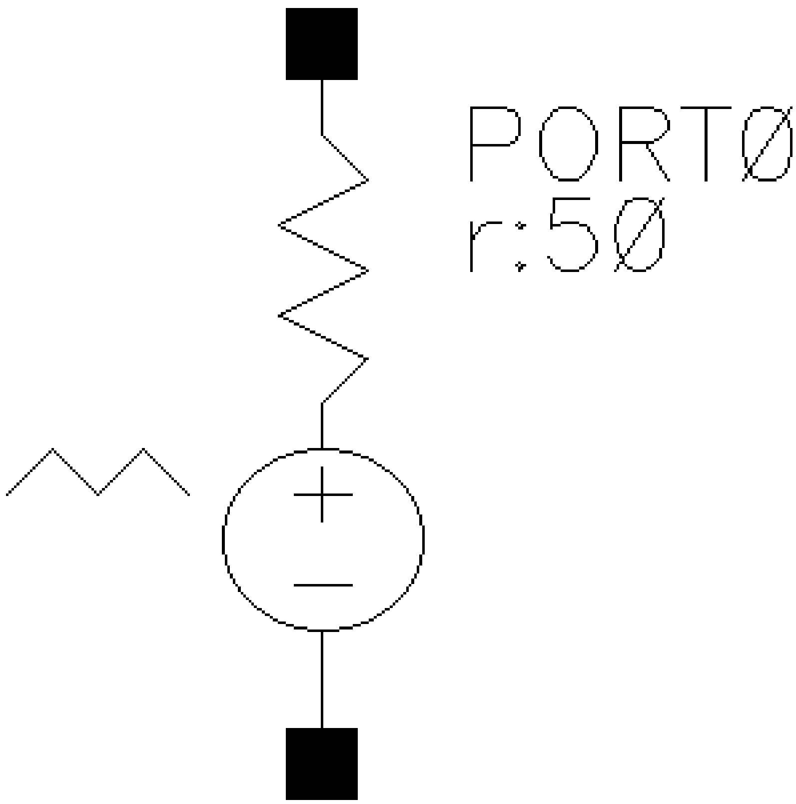
Symbol: ppwl



Independent Piece-Wise Linear Resistive Source

This component is the same as ppwl described in the Chapter 8, “Symbol: ppwl,”

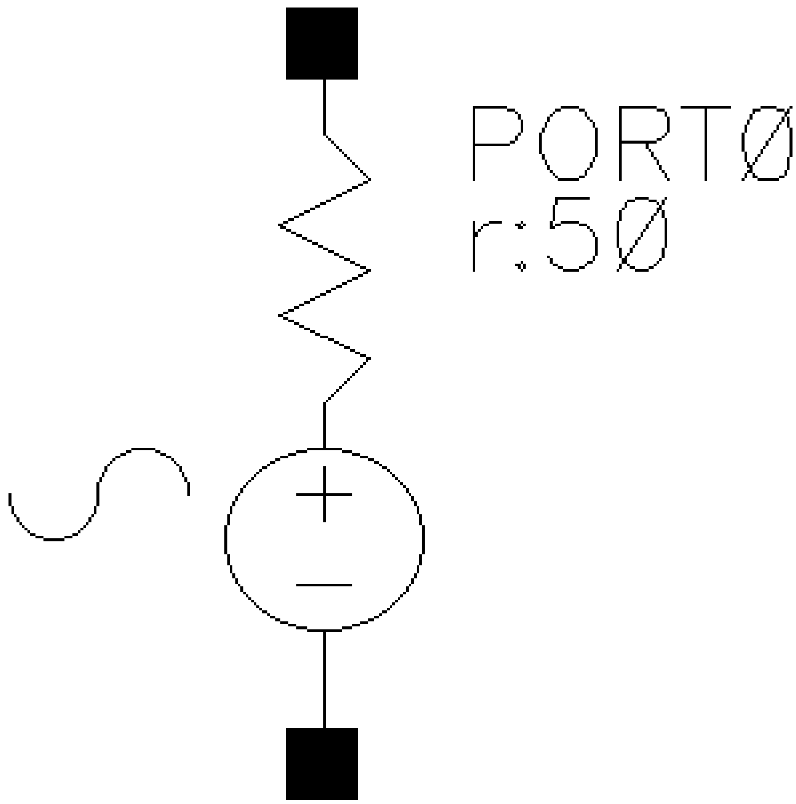
Symbol: ppwlf



Independent Piece-Wise Linear Resistive Source Based on File

This component is the same as ppwlf described in the Chapter 8, “Symbol: ppwlf,”

Symbol: psin



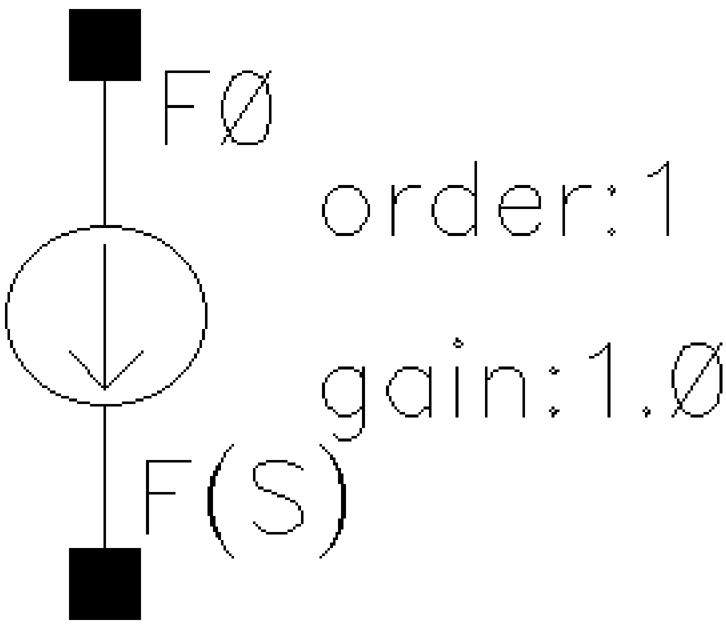
Independent Sinusoidal Resistive Source

This component is the same as psin described in the Chapter 8, “Symbol: psin.”

10

Sources - Z_S_Domain Components

Symbol: scccs



S-Domain Linear Current Controlled Current Source

The device output is defined through a transfer function given as a ratio of two polynomials in the complex variable `s`. Polynomials can be specified in terms of either coefficients or roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

Command-line help

```
spectre -h scccs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Specification type	spec	x	-	-	-	-
Order of transfer function	order	x	-	-	-	-
Probe Device Name	probe	x	-	-	-	-
Coef. of num. const. term	a0	x	-	-	-	-
Coef. of num. 1st term	a1	x	-	-	-	-
Coef. of den. const. term	b0	x	-	-	-	-
Coef. of den. 1st term	b1	x	-	-	-	-
Port	port	x	-	-	-	-
Gain	gain	x	-	-	-	-
Multiplier	m	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ) scccs <parameter=value> ...
```

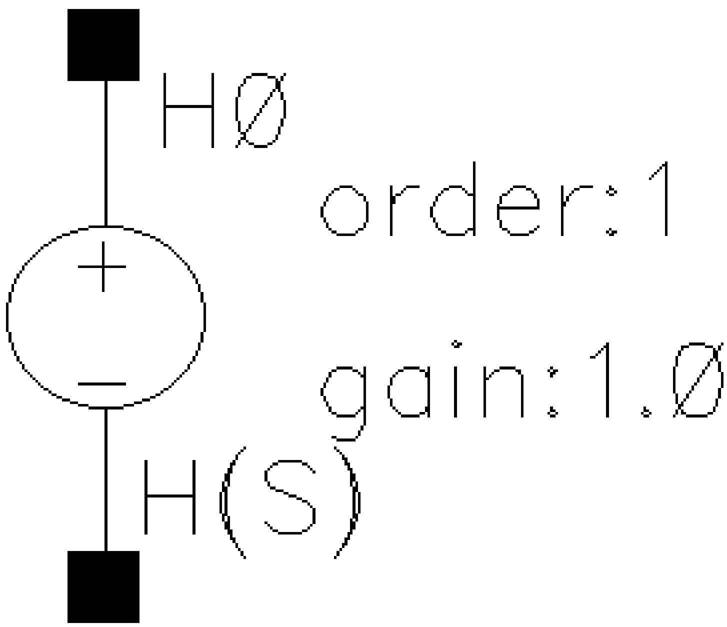
Example

```
l1 (2 1) inductor l=15
sc1 (1 0) scccs probe=l1 zeros=[0 6 0 -6 2 -8 2 8] poles=[-1 0 0 64 0 -64 -2 8 -2 -8]
```

Additional Information

This device is not supported within the altergroups.

Symbol: sccvs



S-Domain Linear Current Controlled Voltage Source

The device output is defined through a transfer function given as a ratio of two polynomials in the complex variable s. Polynomials can be specified in terms of either coefficients or roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

Command-line help

```
spectre -h sccvs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Specification type	`spec`	x	-	-	-	-
Order of transfer function	`order`	x	-	-	-	-
Probe Device Name	`probe`	x	-	-	-	-
Coef. of num. const. term	`a0`	x	-	-	-	-
Coef. of num. 1st term	`a1`	x	-	-	-	-
Coef. of den. const. term	`b0`	x	-	-	-	-
Coef. of den. 1st term	`b1`	x	-	-	-	-
Port	`port`	x	-	-	-	-
Gain	`gain`	x	-	-	-	-
Multiplier	`m`	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) sccvs <parameter=value> ...
```

Example

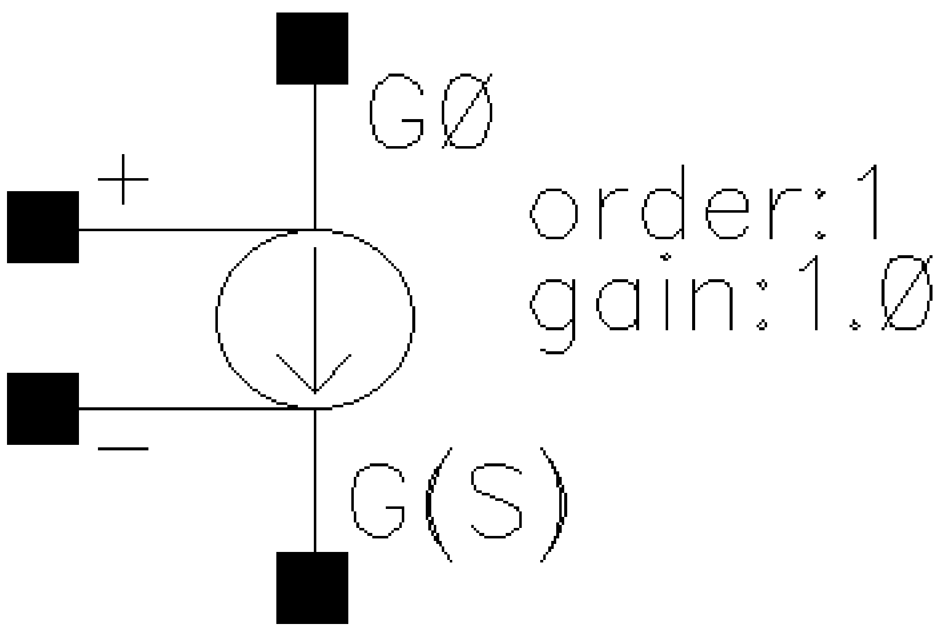
```
myv (1 0) vsource type=sine freq=10K
```

```
scc1 (2 0) sccvs probe=myv gain=0.5 numer=[2] denom=[5]
```

Additional Information

This device is not supported within the altergroups.

Symbol: svccs



S-Domain Linear Voltage Controlled Current Source

The device output is defined through a transfer function given as a ratio of two polynomials in the complex variable `s`. Polynomials can be specified in terms of either coefficients or roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

Command-line help

```
spectre -h svccs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Specification type	`spec`	x	-	-	-	-
Order of transfer function	`order`	x	-	-	-	-
Coef. of num. const. term	`a0`	x	-	-	-	-
Coef. of num. 1st term	`a1`	x	-	-	-	-
Coef. of den. const. term	`b0`	x	-	-	-	-
Coef. of den. 1st term	`b1`	x	-	-	-	-
Gain	`gain`	x	-	-	-	-
Multiplier	`m`	-	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ps ns ) svccs <parameter=value> ...
```

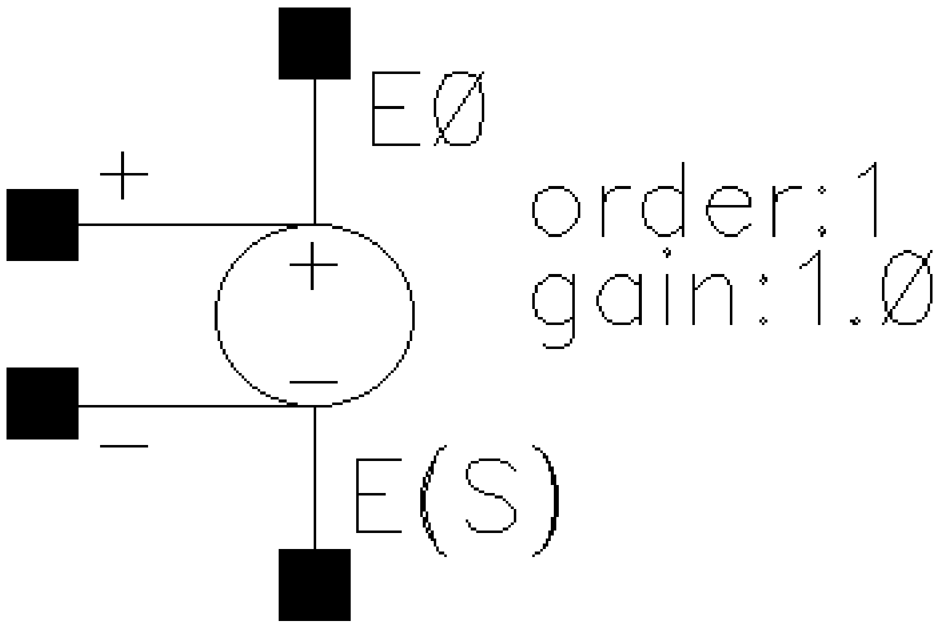
Example

```
s2 (1 0 control 0) svccs gain=0.4 numer=[2 3] denom=[4 5 1]
```

Additional Information

This device is not supported within the altergroups.

Symbol: svcvs



S-Domain Linear Voltage Controlled Volatge Source

The device output is defined through a transfer function given as a ratio of two polynomials in the complex variable s. Polynomials can be specified in terms of either coefficients or roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

Command-line help

```
spectre -h svcvs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Specification type	spec	x	-	-	-	-
Order of transfer function	order	x	-	-	-	-
Coef. of num. const. term	a0	x	-	-	-	-
Coef. of num. 1st term	a1	x	-	-	-	-
Coef. of den. const. term	b0	x	-	-	-	-
Coef. of den. 1st term	b1	x	-	-	-	-
Gain	gain	x	-	-	-	-
Multiplier	m	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ps ns ) svcvs <parameter=value> ...
```

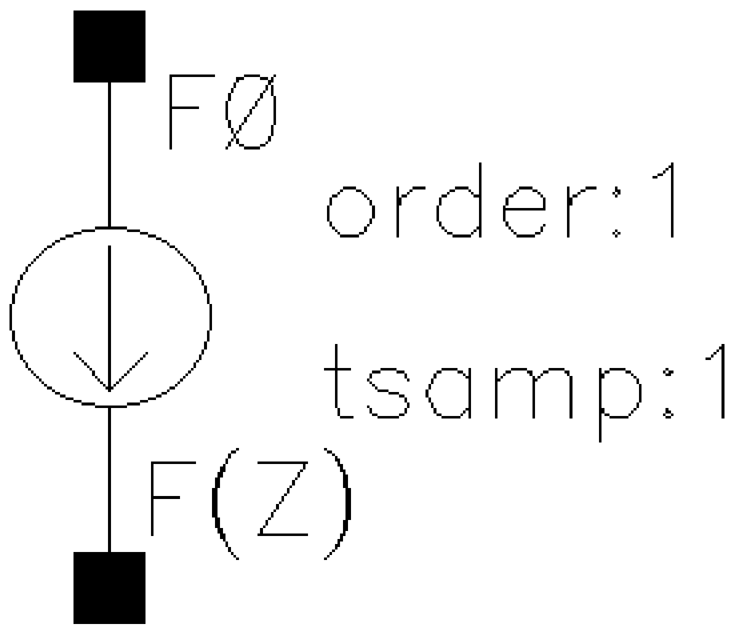
Example

```
e1 (1 0 control 0) svccs gain=5 poles=[-1 0 1 0] zero=[0 0 1 0]
```

Additional Information

This device is not supported within the altergroups.

Symbol: zcccs



Z-Domain Linear Current Controlled Current Source

The output is defined with a transfer function given as the ratio of two polynomials in the complex variable `z`. Each polynomial can be specified using either its coefficients or its roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

Command-line help

```
spectre -h zcccs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Specification type	`spec`	x	-	-	-	-
Order of transfer function	`order`	x	-	-	-	-
Probe Device Name	`probe`	x	-	-	-	-
Coef. of num. const. term	`a0`	x	-	-	-	-
Coef. of num. 1st term	`a1`	x	-	-	-	-
Coef. of den. const. term	`b0`	x	-	-	-	-
Coef. of den. 1st term	`b1`	x	-	-	-	-
Port	`port`	x	-	-	-	-
Sampling period	`tsamp`	x	-	-	-	-
Delay time	`td`	x	-	-	-	-
Transaction time	`tt`	x	-	-	-	-
Gain	`gain`	x	-	-	-	-
Polynomial argument	`polyarg`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
S to Z Transformation	sxz	x	-	-	-	-
Multiplier	m	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ) zcccs <parameter=value> ...
```

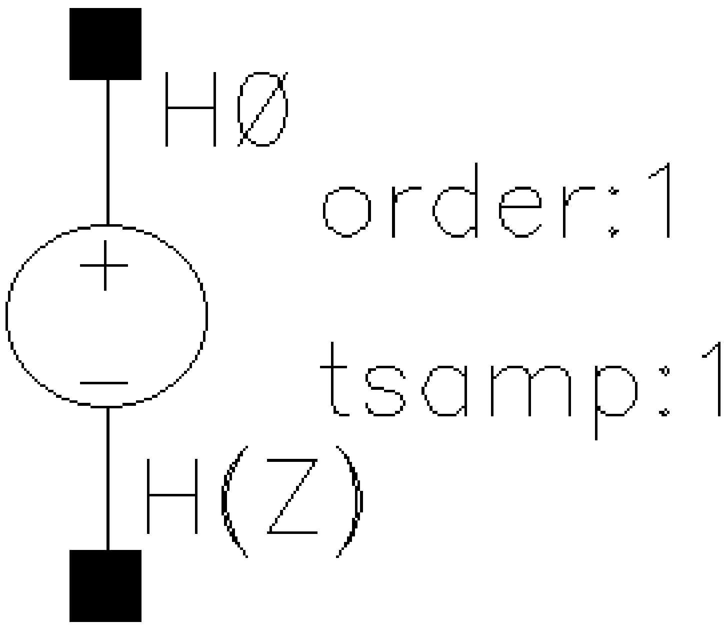
Example

```
va (1 0) vsource type=sine freq=10K
z2 (2 0) zcccs probe=va gain=1 ts=4.9e-5 tt=1e-5 polyarg=inservez
numer=[1 -1] denom=[1 0]
```

Additional Information

This device is not supported within the altergroups.

Symbol: zccvs



Z-Domain Linear Current Controlled Voltage Source

The output is defined with a transfer function given as the ratio of two polynomials in the complex variable `z`. Each polynomial can be specified using either its coefficients or its roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

Command-line help

```
spectre -h zccvs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Specification type	`spec`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Order of transfer function	order	x	-	-	-	-
Probe Device Name	probe	x	-	-	-	-
Coef. of num. const. term	a0	x	-	-	-	-
Coef. of num. 1st term	a1	x	-	-	-	-
Coef. of den. const. term	b0	x	-	-	-	-
Coef. of den. 1st term	b1	x	-	-	-	-
Port	port	x	-	-	-	-
Sampling period	tsamp	x	-	-	-	-
Delay time	td	x	-	-	-	-
Transaction time	tt	x	-	-	-	-
Gain	gain	x	-	-	-	-
Polynomial argument	polyarg	x	-	-	-	-
S to Z Transformation	sxz	x	-	-	-	-
Multiplier	m	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) zccvs <parameter=value> ...
```

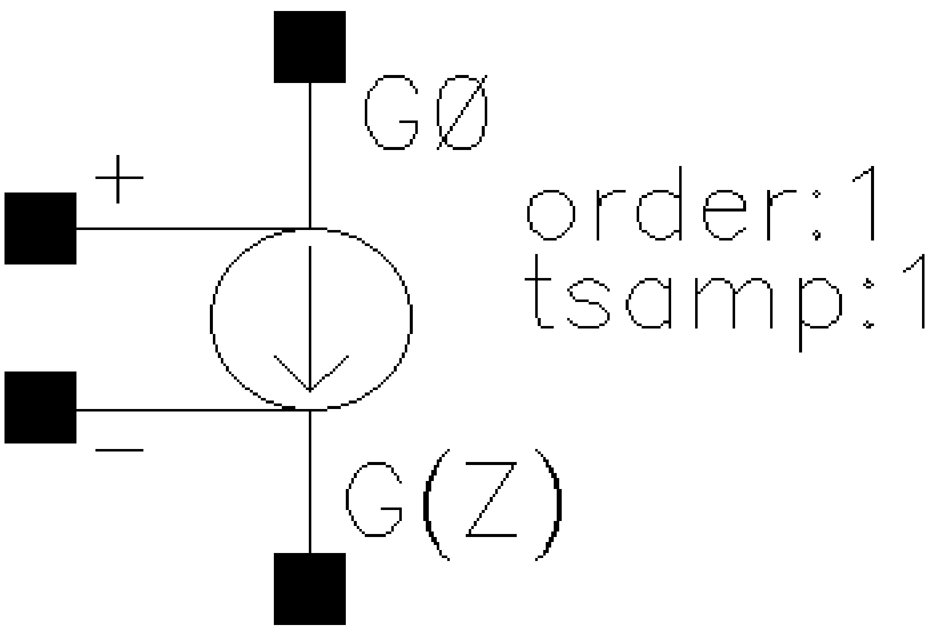
Example

```
va (1 0) vsource type=sine freq=10K
z2 2 0 zccvs probe=va gain=-2 ts=5e-5 tt=1.1e-5 numer=[1 -1]
```

Additional Information

This device is not supported within the altergroups.

Symbol: zvccs



Z-Domain Linear Voltage Controlled Current Source

The output is defined with a transfer function given as the ratio of two polynomials in the complex variable `z`. Each polynomial can be specified using either its coefficients or its roots.

The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

Command-line help

```
spectre -h zvccs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Polynomial argument	`polyarg`	x	-	-	-	-
S to Z Transformation	`sxz`	x	-	-	-	-
Specification type	`spec`	x	-	-	-	-
Order of transfer function	`order`	x	-	-	-	-
Coef. of num. const. term	`a0`	x	-	-	-	-
Coef. of num. 1st term	`a1`	x	-	-	-	-
Coef. of den. const. term	`b0`	x	-	-	-	-
Coef. of den. 1st term	`b1`	x	-	-	-	-
Sampling period	`tsamp`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Delay time	td	x	-	-	-	-
Transaction time	tt	x	-	-	-	-
Gain	gain	x	-	-	-	-
Multiplier	m	x	-	-	-	-

Syntax/Synopsis

```
Name ( sink src ps ns ) zvccs <parameter=value> ...
```

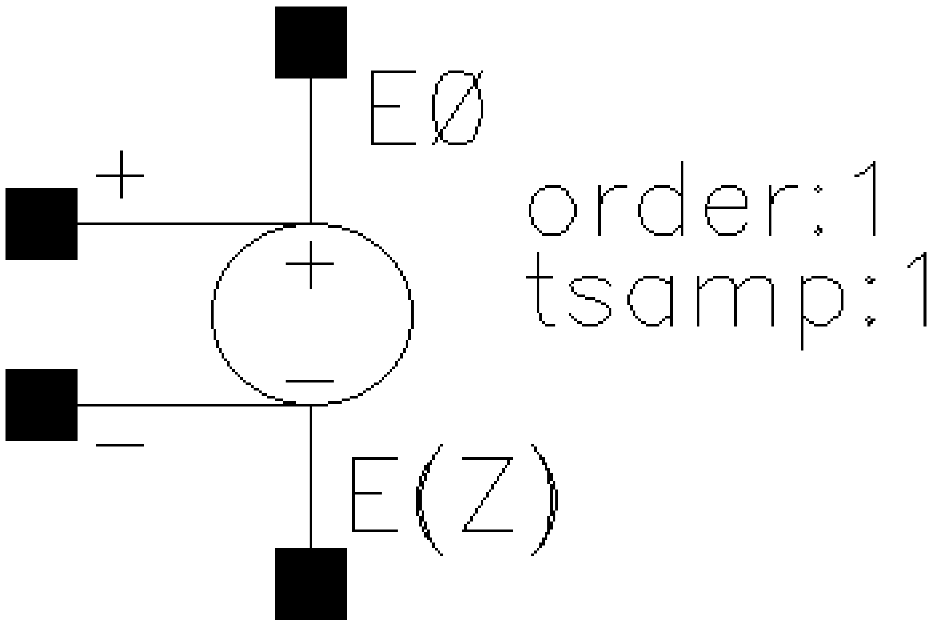
Example

```
va (1 0) vsource type=sine freq=10K
z1 (2 0 1 0) zvccs gain=2 ts=4.5e-5 tt=1e-5 zeros=[-1 0] poles=[0 0]
```

Additional Information

This device is not supported within the altergroups.

Symbol: zvcvs



Z-Domain Voltage Controlled Voltage Source

The output is defined with a transfer function given as the ratio of two polynomials in the complex variable `z`. Each polynomial can be specified using either its coefficients or its roots. The roots of the numerator are the zeros of the transfer function and the roots of the denominator are the poles.

To use the ‘s’ to ‘z’ transformation, set the optional ‘sxz’ parameter to one of the transformation methods - forward differences, backward differences, or bilinear. When the ‘sxz’ parameter is specified, the transfer function specification is assumed to be given in the complex variable s and it will be transformed to the complex variable z using the indicated method.

Command-line help

```
spectre -h zvcvs
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Polynomial argument	`polyarg`	x	-	-	-	-
S to Z Transformation	`sxz`	x	-	-	-	-
Specification type	`spec`	x	-	-	-	-
Order of transfer function	`order`	x	-	-	-	-
Coef. of num. const. term	`a0`	x	-	-	-	-
Coef. of num. 1st term	`a1`	x	-	-	-	-
Coef. of den. const. term	`b0`	x	-	-	-	-
Coef. of den. 1st term	`b1`	x	-	-	-	-
Sampling period	`tsamp`	x	-	-	-	-
Delay time	`td`	x	-	-	-	-
Transaction time	`tt`	x	-	-	-	-
Gain	`gain`	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Multiplier	`m`	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ps ns ) zvcvs <parameter=value> ...
```

Example

```
va (1 0) vsource type=sine freq=10K
z3 (3 0 1 0) zvcvs gain=-1 ts=4e-5 tt=1e-5 numer=[-1 -1]
```

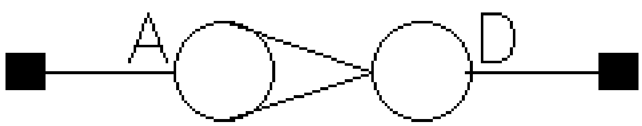
Additional Information

This device is not supported within the altergroups.

11

Interface Elements

Symbol: MOS_a2d



Interface Element for MOS - Metal Oxide Semiconductor Analog to Digital Convertor

A general Interface Element (IE) for mixed signal. To match your design, you will need to change the Base CDF. The default is for 5V logic. Do not manually place this component in your schematic as the IE placement is done automatically by the mixed signal netlister.

Node	Name
Input	A
Output	D

Command-line help

```
spectre -h a2d
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Macro Name	`macro`	-	-	-	-	-
Level 0 threshold	`a2d_v0`	x	-	-	-	-
Level 1 threshold	`a2d_v1`	x	-	-	-	-
Time to x state	`a2d_tx`	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) a2d <parameter=value>
```

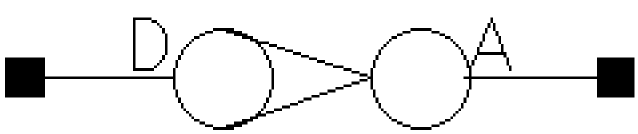
Example

```
I0 a2d timex=1m v1=1.5 vh=3.5
```

Additional Information

This device is not supported within the altergroups.

Symbol: MOS_d2a



Interface Element for MOS - Metal Oxide Semiconductor Digital to Analog Convertor

A general Interface Element (IE) for mixed signal. To match your design, you will need to change the Base CDF. The default is for 5V logic. Do not manually place this component in your schematic as the IE placement is done automatically by the mixed signal netlister.

Node	Name
Input	D
Output	A

Command-line help

```
spectre -h d2a
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Macro name	`macro`	-	-	-	-	-
Level 0 voltage	`d2a_vl`	x	-	-	-	-
Level 1 voltage	`d2a_vh`	x	-	-	-	-
Rise time	`d2a_tr`	x	-	-	-	-
Fall time	`d2a_tf`	x	-	-	-	-
Level X voltage	`d2a_vx`	x	-	-	-	-
Level Z voltage	`d2a_vz`	x	-	-	-	-

Syntax/Synopsis

```
Name ( p n ) d2a <parameter=value> ...
```

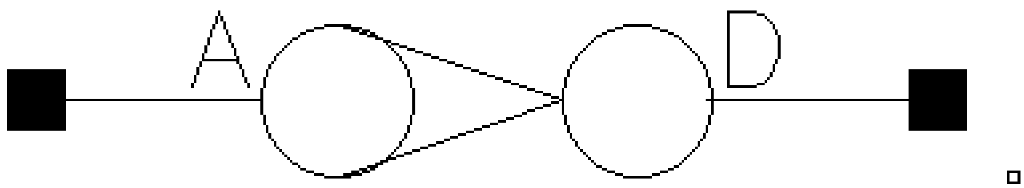
Example

```
I1 d2a fall=2n rise=3n val1=5 val0=0 valx=1.25
```

Additional Information

This device is not supported within the altergroups.

Symbol: TTL_a2d



Interface Element for TTL - Transistor to Transistor Logic Analog to Digital Convertor

An Interface Element (IE) for TTL that is used as an analog-to-digital interface for mixed-signal simulations. The analog-to-logic converter transfers analog waveforms to a logic simulator.

Command-line help

```
spectre -h a2d
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Macro Name	macro	x	-	-	-	-
Level 0 threshold	a2d_v0	x	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Level 1 threshold	`a2d_v1`	x	-	-	-	-
Time to x state	`a2d_tx`	x	-	-	-	-

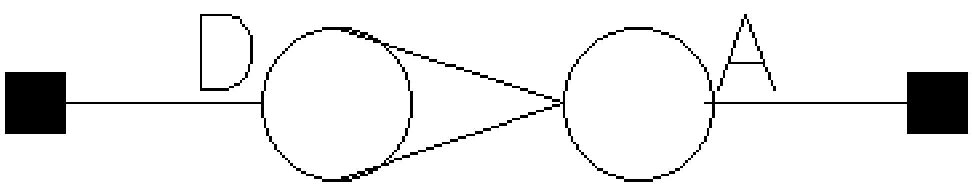
Syntax/Synopsis

Example

```
I3 interfaceElement timex=1m v1=1.5 vh=3.5
```

Additional Information

Symbol: TTL_d2a



Interface Element for TTL - Transistor to Transistor Digital to Logic Analog Convertor

An Interface Element (IE) for TTL that is used as a digital-to-analog interface for mixed-signal simulations.

Command-line help

```
spectre -h d2a
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Macro Name	macro	x	-	-	-	-
Level 0 voltage	d2a_vl	x	-	-	-	-
Level 1 voltage	d2a_vh	x	-	-	-	-
Rise time	d2a_tr	x	-	-	-	-
Fall time	d2a_tf	x	-	-	-	-
Level X voltage	d2a_vx	x	-	-	-	-
Level Z voltage	d2a_vz	x	-	-	-	-

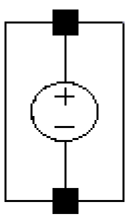
Example

```
I3 interfaceElement fall=2n rise=3n val1=5 val0=0
```

12

Behavioral Model

Symbol: bsource



Behavioral Source

This component can be used to model a resistor, inductor, capacitor, voltage or current source as a behavioral component. Using `bsource` you can express the value of a resistance, capacitance, voltage or current as a combination of node voltages, branch currents, time expression, and built-in expressions.

Command-line help

```
spectre -h bsource
```

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Model type	behav_param	x	-	-	-	-	-	-	-
Expression	expr	x	-	-	-	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
Multiplier	`m`	x	-	-	-	-	-	-	-
Temp rise from ambient	`trise`	x	-	-	-	-	-	-	-
Nominal temperature	`tnom`	x	-	-	-	-	-	-	-
Temperature coefficient 1	`tc1`	x	-	-	-	-	-	-	-
Temperature coefficient 2	`tc2`	x	-	-	-	-	-	-	-
Max value of bsource expr	`max_val`	x	-	-	-	-	-	-	-
Min value of bsource expr	`min_val`	x	-	-	-	-	-	-	-
Flicker noise coefficient	`kf`	x	-	-	-	-	-	-	-
Flicker noise exponent	`af`	x	-	-	-	-	-	-	-
Generate noise?	`isnoisy`	x	-	-	-	-	-	-	-
Lin temp co of lin cap	`tc1c`	x	-	-	-	-	-	-	-
Quad temp co of lin cap	`tc2c`	x	-	-	-	-	-	-	-

CDF Parameter Label	CDF Parameter	spectre	spectreS	cdsSpice	auCdl	auLvs	hspiceS	hspiceD	UltraSim
DC-Mismatch parameter	`mr`	x	-	-	-	-	-	-	-
Initial condition	`ic`	x	-	-	-	-	-	-	-
Implementation Type	`ctype`	x	-	-	-	-	-	-	-
White noise expression	`white_noise`	x	-	-	-	-	-	-	-
Flicker noise expression	`flicker_noise`	x	-	-	-	-	-	-	-
Model name	`model`	x	-	-	-	-	-	-	-

Syntax/Synopsis

```
name (node1 node2) bsource behav_param param_list
```

A

List of All CDF Parameters

CDF Parameter Label	CDF Parameter	Description	Default
# of lumps in element	`lumps`		
Absolute Output Current	`absol`		0
Absolute Value	`abs`		
Absolute value	`habs`		
AC magnitude	`acm`		
AC Model	`acModel`	Alters nport behavior in small signal analyses: sp, ac and xf. Possible values are freqdomain and timedomain.	freqdomain
AC phase	`acp`		
AC Phase	`acPhase`		
AC position	`acPosition`	Position to which switch is set at the start of AC analysis.	-
AC resistance	`ac`		-
Active	`active`	Whether Fourier analysis should be performed or skipped. Possible values are no or yes.	-
Accuracy	`accuracyMode`	Whether accuracy should be default or conservative.	default
Additional drain resistance	`rdc`		-
Additional parameter list	`additionalParam`		-

CDF Parameter Label	CDF Parameter	Description	Default
Additional source resistance	`rsc`		-
Advanced transient parameters	`tranAdvanParaLabel`		-
All are breakpoints *Breakpoints*	`allbrkpts`		-
Alias of mult	`area`		-
Alpha parameter	`alph`	Scaling factor for `Q`.	-
AM modulation frequency	`ammodfreq`		-
AM modulation index	`ammodindex`		-
AM modulation phase	`ammodphase`		-
Amplitude 1 (Ipk)	`ia`		-
Amplitude 1 (dBm)	`vaDBm`		-
Amplitude 1 (Vpk)	`va`		-
Amplitude 2 (Ipk)	`ia2`		-
Amplitude 2 (Vpk)	`va2`		-
Amplitude 2 (dBm)	`vaDBm2`		-
Anode gate voltage	`Vag`		-
Base area	`areab`		-
Base-emitter voltage	`Vbe`		-
Bit 1 voltage level	`vbit1`	Bit 1 voltage level.	1
Bit 0 voltage level	`vbit0`	Bit 0 voltage level.	0
Bit Pattern(MSB...LSB)	`pattern`	Bit pattern.	0000
Browse and select file	`selectFile`	Can be used to open the file browser for selecting a waveform file. The path of the selected file is stored in the `fileName` parameter.	`nil`

CDF Parameter Label	CDF Parameter	Description	Default
Browse and select s-data file	`nportFileB`	Opens the file browser for selecting an s-data file. This check box is shown only when you do not select the *S-parameter file as Design Var?* check box.	`nil`
Blocking cap for net analyser	`cblock`	Blocking capacitance for network analyser.	0.0001 F
Body contact area	`ab`		-
Body diffusion resistor square	`nrb`		-
Body-source initial voltage	`Vbys`		-
Browse S-parameter file	`nportFileB`	Lets you browse to a location and specify the S-Parameter data file.	-
Bulk node connection	`bn`		-
Bulk source initial voltage	`Vbs`		-
Capacitance	`c`	Capacitance	1p F
Capacitor Area	`area`	Area of capacitor	1
Capacitor Perimeter	`perim`	Perimeter for capacitor	0
Capacitance connected	`hrc`		-
Carrier frequency	`fc`		-
Cathode gate voltage	`Vcg`		
Causality Correction	`causality`	Possible values are fmax, auto, no, fmax_active.	fmax
Characteristic impedance	`zo`	Characteristic impedance of lossless line.	50
checkFlag	`checkFlag`	Checks the state of the flag.	1

CDF Parameter Label	CDF Parameter	Description	Default
check box status	`checkBoxValue`	Checks the status of the check box.	1
Check Passivity	`passivity`	Check and enforce passivity of s parameters.	no
Chock ind for net analyser	`lchock`	Chock inductor for network analyser.	0.1 H
Close switch resistance	`rc`	Resistance of a fully closed relay.	-
Closed voltage	`vt2`	Relay resistance is ‘rclosed’ at this voltage.	-
CMDM	`CMDM`		1
Coef. of den. 1st term	`b1`		-
Coef. of den. const. term	`b0`		-
Coef. of num. 1st term	`a1`		-
Coef. of num. const. term	`a0`		-
Collector area	`areac`		-
Collector length	`lc`		-
Collector-emitter voltage	`Vce`		-
Common Reference	`nmode`	When selected, redraws the symbol with a single common reference pin at the bottom and eliminates the need to add reference connections to each port of the symbol	-
Conductor thickness	`t`		-
Contact configuration	`order`		-
Controlling Volt 1	`x1`		-
Controlling Volt 2	`x2`		-
Corresp Element 1	`y1`		-

CDF Parameter Label	CDF Parameter	Description	Default
Corresp Element 2	`y2`		-
Coupler domain	`a_or_d`	specify the domain to which the coupler belongs: analog or digital	analog
Coupling coefficient	`k`	Coupling coefficient	0
Current 1 - Current 50	`i1 - i50`		-
Current 2	`i2`		-
Current eqn	`cur`		-
Current gain	`fgain`		-
Current gain (Obsolete)	`hfgain`	**Note:** Parameter hfgain is obsolete. Parameter fgain is used for both Spectre and HspiceD instead.	-
Damping factor	`theta`		-
Damping factor 1 *Rise time constant*	`tau1`	Rise time constant for exponential wave.	
Damping factor 2 *Fall time constant*	`tau2`	Fall time constant for exponential wave.	
Data truncation threshold	`datatrunc`		
DC current	`idc`		
DC extrapolation	`dcextrap`	Long delay DC extrapolation method: constant or unwrap	constant
DC position	`dcPosition`	Position to which switch is set at the start of DC analysis.	0
DC source	`dc`		
DC voltage	`vdc`		
Decimal Value	`dec`	Value of Bit Pattern in Decimal.	0
Delay Schedule	`ibisDelaySchedule`		NO

CDF Parameter Label	CDF Parameter	Description	Default
Delay Time	`delay`		
Delay time	`td`	Time delay	1n
Delay Time	`htd`	Time delay	
Delay time 1 *Rise time start*	`td1`	Rise start time for exponential wave	
Delay time 2 *Fall time start*	`td2`	Fall start time for exponential wave.	
Delta	`delta`		
Device area	`area`	Transistor area factor.	
Device initially off	`off`		
Dielectric loss cond matrix per unit length	`gdloss`	Dielectric loss conductance matrix per unit length.	
Differential threshold	`vdiff`		
Display modulation params	`modulation`		
Display noise parameters	`noiseParam`		
Display second sinusoid	`numofsines`		
Display small signal params	`smallSig`		
Distance to a single well edge	sc		
shift in 0-bias threshold vth0	`delvo`		
Frequency for Q/ Frequency for L and Q	`fq`	Frequency at which quality factor and inductance are measured.	
Gate contact-channel edge	`xgw`		

CDF Parameter Label	CDF Parameter	Description	Default
Expand Bus	`expand`	Expand bus. Possible values are yes and no.	Yes
Input Mode	`mode1`	Input mode for Bit pattern. Possible values are binary, hexadecimal and octal.	Binary
Number of Bits	`numbits`	Number of Bits.	4
Number of gate contacts	`ngcon`		
Number of input pins	`n_inp`	Number of input ports. Min:0 and Max:100	1
Number of output pins	`n_outp`	Number of output ports. Min:0 and Max:100	1
Mode	`mode`	Integer parameter that selects the frequency dependence.	
Rdc	`rdc`	DC resistance	0.0
Show advanced options	`advUser`	When selected, the parameters under it will be shown. By default the parameetrs are not shown	1
Initial coupler output voltage	`init_val`	Initial value of for interpolation. Sets the analog output value for simulation time to 0	0
Simulink(R) hostname	`hostname`	Hostname of the master simulator.	local host
Socket port	`sockPort`	TCP port number for socket connection. This parameter must be set to the same value for coupler of both simulators. It should be greater than 1024	5023

CDF Parameter Label	CDF Parameter	Description	Default
Sim response timeout	`sockTimeout`	Seconds to wait for an answer from the master simulator during simulation. Increase this value if the master simulator needs long calculation time per sample/frame. It should be greater than 30.	120
DC-Mismatch parameter	`mr`	DC-Mismatch parameter. Valid only for r.	-
Delay frequency	`delayfreq`		-
Display temperature params	`tempParam`		
Dist. betn & poly(one side)	`sa`		
Dist. betn OD & poly(otherside)	`sb`		
Dist. betn neighbour fingers	`sd`		
Drain diffusion area	`ad`		
Drain diffusion length	`ld`		
Drain diffusion periphery	`pd`		
Drain diffusion res squares	`nrd`		
Drain source initial voltage	`Vds`		
Dummy DC voltage	`vdummy`		0
Emitter length	`le`		
Emitter width	`we`		
Enable mixed mode	`mixedmode`		-
Enable noise passive checker	`noipassivechk`		-

CDF Parameter Label	CDF Parameter	Description	Default
Enable passive checker	`passive`		-
Enable rational function	`rational_func`	Enables the rational function when using hspiceD.	t
End Frequency	`endFrequency`	Specifes the end frequency point in `bbsfreqband`.	
Enter RLCG etc. matrices	`entermatrices`		
Estimated operating region	`region`	Estimated operating region. Possible values are off, on or breakdown.	-

<table>
<tr><th>CDF Parameter Label</th><th>CDF Parameter</th><th>Description</th><th>Default</th></tr>
<tr><td>Expression</td><td><code>expr</code></td><td>Behavioral expression. Depending on the value of <code>behave_param</code>, this can be either <code>simple_expr</code> or <code>generic_expr</code>.

<code>simple_expr</code> – Spectre expression which contains the following:
■ Netlist parameters
■ Current simulation time, <code>$time</code>
■ Current frequency, <code>$freq</code>
■ Node voltage, <code>v(a,b)</code>, where <code>a</code> and <code>b</code> are nodes in the Spectre netlist, or <code>v(a)</code>, which is the voltage between node a and ground
■ Branch currents, <code>i("inst_id:index")</code>, where <code>inst_id</code> is an instance name given in the netlist and <code>index</code> is the port index that starts from <code>1</code>.
Default value of <code>index</code> is <code>1</code>.
■ Note: If the value of the port index is set to <code>0</code>, <code>simple_expr</code> treats it as the default value <code>1</code>.

<code>generic_expr</code> – <code>simple_expr</code> or <code>ddt()</code> of <code>simple_expr</code> or <code>idt()</code> of <code>simple_expr</code>.</td><td>0</td></tr>
</table>

CDF Parameter Label	CDF Parameter	Description	Default
Extracts a system delay	`delayhandle`		-
Fall Delay	`td10`	Time delay for 1 to 0 transition.	-
Fall on delay	`fall_on_dly`		
Fall off delay	`fall_off_dly`		
Fall time	`d2a_tf`		2n s
Fall time	`fall`	Time for transition to fall from Level 1 voltage to Level 0 voltage.	2n s
Fall time	`tf`	Time for transition to fall from Level 1 voltage to Level 0 voltage.	2n s
First coupled inductor	`ind1`	Inductor to be coupled	
First harmonics computed	`firstharm`	First harmonic computed for the test (numerator) channel.	“ “
First of reference harmonics	`reffirstharm`	First harmonic computed for the reference (denominator) channel.	“ “
Flag for matrix form input	`matrixform`	Flag for matrix form input. Possible values are `no` or `yes`.	
Flicker noise expression	`flicker_noise`	Generates pink noise with given power at 1 Hz that varies in proportion to `1/f^exp`. Noise contributions with the same tag are combined for a module.	-
*Flicker noise coefficien*t	`kf`	Flicker noise co-efficient. Valid for r and g elements.	`0`
Flicker noise exponent	`af`	Flicker noise exponent. Valid for r and g elements.	`2`
Flow	`flow`	Flow quantity	
FM modulation frequency	`fmmodfreq`		

CDF Parameter Label	CDF Parameter	Description	Default
FM modulation index	`fmmodindex`		
Freq 1 to Freq 50	`F1 - F50`		
Frequency	`freq`	Reference frequency (used in conjunction to the normalized length to specify electrical length of line).	
Frequency 1	`freq`		
Frequency 2	`freq2`		
Frequency name 1	`fundname`		
Frequency name for 1/ period	`fundname`		
Frequency sampling interval	`fdelta`		
Frequency scale factor	`freqscale`	The frequency scale factor for frequency-dependent RLGC data and S-Parameter data.	
Front gate-source voltage	`Vgfs`		
Fundamental frequency	`fund`		-
Gain	`gain`	Gain Parameter.	1.0
Gap length	`gap`	Gap length	
Gate source initial voltage	`Vgs`		
Gate to bulk and src voltage	`Vgbs`		
Generate noise?	`isnoisy`	Should resistor generate noise. Possible values are no or yes.	-
Hierarchical Node	`probeNode`	Hierarchical net name in Spectre syntax. The net name should be as it appears in the netlist.	-

CDF Parameter Label	CDF Parameter	Description	Default
High Frequency Extrapolation	`hfextrap`	Long delay high-frequency extrapolation method. Possible values are constant and linear.	constant
High freq extrapolate method	`highpass`	Possible values are 0,1,2,3,4.	3
High freq. limit for approx.	`f1`	High frequency limit for the approximation.	f1=1.0e6 Hz
High-Z impedance	`highz`	Impedance of high z state.	-
Hot-electron degradation	`degradation`		
Hspice Interpolation method	`interpolation`	Possible values are: linear, spline, step, hybrid.	linear
Hspice S-parameter data format	`datafmtHspice`	Possible values are fqmodel, touchstne, citi, rfm, bnp.	touchstone
IBIS Entry Method	`ibisEntryMethod`		
IBIS file name	`ibisFile`		
IBIS model name	`ibisModelNameo`		
IBIS corner	`ibisCorner`	Specify the corner of an IBIS buffer. Possible corner parameters are typical, maximal, minimal, fast, and slow.	typical
IC position	`icPosition`	Position to which switch is set at the start of IC analysis (precedes transient analysis).	0

CDF Parameter Label	CDF Parameter	Description	Default
Implementation Type	`ctype`	Different implementations of a capacitor. When the value is `1`, bsource current is `ddt(cap*V(node1, node2))`, where `cap` is the bsource capacitor value with temp effect, mfactor effect, scale effect and so on. `V(node1, node2)` is the voltage between the bsource terminals. When the value is `2`, the current is `ddt(cap)`. When the value is 0 or any other value, the current value is `cap*ddt(V(node1, node2))`.	0
Impulse response truncation	`imptrunc`		
Inductance	`l`		1n
Initial condition	`ic`	Initial condition	-
Initial condition	`hic`		
Initial diode voltage	`Vd`		-
Initial magnetization of core	`mag`		
Initial phase for Sinusoid	`sinephase`		
Initial phase for Sinusoid 2	`sinephase2`		
Inner diam of toroidal core	`idiam`	Inner diameter of toroidal core	
Integral-1st distribution func	`sca`		

CDF Parameter Label	CDF Parameter	Description	Default
Integral-2nd distribution func	`scb`		
Integral-3rd distribution func	`scc`		
Internal junction voltage	`Vbcc`		
Interpolation Method	`interp`	Method to interpolate S-Parameter data. Possible values are `default`, `spline`, `linear`, `bbspice`, and `auto_switch`.	`default`
Invoke 'LMG' parameter extraction tool	`firelmg`		
Junction perimeter factor	`perim`		-
Length	`l`	Length of the resistor	-
Length of Emitter Window	`le0`		
Length of metal capacitor	`lm`	Length of metal capacitor	-
Length of polysilicon	`lp`	Length of polysilicon capacitor	-
Level 0 threshold	`a2d_v0`		1.5 V
Level 0 voltage	`d2a_vl`	Final value for logical 0.	0 V
Level 1 threshold	`a2d_v1`		3.5 V
Level 1 voltage	`d2a_vh`		5 V
Level X voltage	`d2a_vx`		
Level Z voltage	`d2a_vz`		

CDF Parameter Label	CDF Parameter	Description	Default
LFSR Mode	`lfsrmode`	When set for PRBS, the registerlength defines the width of the LFSR and the LFSR works in Maximum Length Sequence mode. Specify seed and taps, Specify bit file, PN32, PN16, PN2, PN3, PN4, PN5, PN6, PN7, PN8, PN9, PN10, PN11, PN12, PN13, PN14, PN15, PN17, PN18, PN19, PN20, PN21, PN22, PN23, PN24, PN25, PN26, PN27, PN28, PN29, PN30, and PN31.	PN32
Lin temp co of lin cap	`tc1c`	Linear temperature coefficient of capacitor.	
Linear interpolation data type	`intdattyp`	Data type of linear interpolation. Values: RI, DBSA, MA.	MA
LMG subcircuit file	`subcktfile`		
Location of collector contact	`location`		
Loss conductance per unit length	`g`	Dielectric (shunt) conductance per unit length	
Loss resistance per unit length	`rs`		
Low freq extrapolate method	`lowpass`	Values: 0,1,2,3,	1
Low freq. limit for approx.	`f0`	Low frequency limit for the approximation.	1.0 Hz
lxRemoveDevice.	`lxRemoveDevice`	Removes a device during netlisting.	(short(PLUS MINUS))
Macro name	`macro`		
Matrix entry data file	`matrixfile`	Matrix entry data file name.	
Max	`max`		-

CDF Parameter Label	CDF Parameter	Description	Default
Max Coefficient Number	`polyCoef`		0
Max order impulse response	`maxn`		
Max signal frequency	`fmax`	Maximum signal frequency	
Maximum output current (Obsolete)	`maxi`		
Maximum output current	`maxm`	Sets the Voltage gain for both Spectre and HspiceD.	
Maximum output resistance	`maxr`		
Maximum output voltage (Obsolete)	`maxv`	**Note:** Parameter `maxv` is obsolete. Parameter `maxm` is used to set the Voltage gain for both Spectre and HspiceD instead.	
Max value of bsource expr	`max_val`	Maximum value of bsource expression. Valid for all behavioral elements, but used with i and v elements to clip the current or voltage between the specified values.	-
Method of smooth	`smooth`	Possible values are 0,1,2,3,4.	0
Min	`min`		-
Min high-Z trans. width	`min_z_transition_width`	Minimum width of transition from z-state to a non z-state. The width of transition is set as 1e-3*(z-state duration).	-
Minimum no. of time points	`points`		-
Minimum output current (Obsolete)	`mini`		
Minimum output current	`minm`		

CDF Parameter Label	CDF Parameter	Description	Default
Minimum output resistance	`minr`		
Minimum output voltage (Obsolete)	`minv`		
Min value of bsource expr	`min_val`	Minimum value of bsource expression. Valid for all behavioral elements, but used with i and v elements to clip the current or voltage between the specified values.	-
Model name	`model`	Specifies the name of the model to be associated with the component.	-
Model name	`hmname`	Specifies the name of the model to be associated with the component.	MDN

CDF Parameter Label	CDF Parameter	Description	Default
Model type	behav_param	Type of behavioral source. It can be one of the following: c=simple_expr – Capacitance between the nodes g=simple_expr – Conductance between the nodes i=generic_expr – Current through bsource l=simple_expr – Inductance between the nodes phi=simple_expr – Flux in the bsource device q=simple_expr – Charge in bsource device r=simple_expr – Resistance between the nodes v=generic_expr – Voltage across the nodes	v
Model type	modeltype	Model type. Possible values are *lossless*, *narrowband*, and *wideband*.	
Modulation frequency	fm		
Modulation index	mdi		
Multiplicity factor	mf	Multiplicity factor.	1
Multiplier (Obsolete)	hm	**Note:** Parameter hm is obsolete. Parameter m is used for both Spectre and HspiceD instead.	
Name of core	core	Name of core around which winding is wrapped.	-

CDF Parameter Label	CDF Parameter	Description	Default
Name of FM File1 *Name of FM File2*	`fmmodfile1` `fmmodfile2`	Name of files that contain data for Frequency Modulated waveform for a sinesoid source.	-
Name of the model	`modelName`		
Name of voltage source	`vref`		
Name of winding 1	`l1`		
No. of reference Harmonics	`refharms`	Number of harmonics for reference (denominator) channel, if an array is not given. The harmonics start from ‘reffirstharm’ and go up to ‘reffirstharm’ + ‘harms’ - 1.	-
No. of Harmonics for PSS	`pssharms`		
Noise correlation matrix	`noisecorr`	Type of noise correlation matrix: real or complex	
Noise 1 to Noise 50	`N1 - N50`		
Noise file name	`noisefile`	Name of file containing excess spot noise data in the form of frequency-noise pairs.	
Noise parameters	`noiseParaLabel`	.	
Noise temperature	`noisetemp`		
Nominal temperature	`tnom`	Parameter measurement temperature. Default set by options.	
Normalized length	`nl`	Normalized electrical length in wavelengths at ‘f’ of a lossless line.	
Nomalizing harmonic	`normharm`	Normalizing harmonic for the test (numerator) channel.	“ “
Nomalizing reference harmonic	`refnormharm`	Normalizing harmonic for the reference (denominator) channel.	“ “

CDF Parameter Label	CDF Parameter	Description	Default
NQS flag	`nqsmod`		
Num of controlling voltage(s)	`nc`		
Num of lines (excluding ref.)	`n`	Number of lines.	1
Number of turns on secondary	`n2`	Number of turns on winding 2.	
Num of turns on winding	`turn`	Number of turns on winding.	-
Number of base contacts	`nb`		
Number of collector contacts	`ncbjt`		
Number of emitter contacts	`ne`		
Number of controlling pairs	`xypairs`		
Number of desired harmonics	`harmsvec`	Array of desired harmonics for test (numerator) channel.	`0`
Number of devices in parallel	`mult`		
Number of FM files	`filenums`	Number of files that contain data for Frequency Modulated waveform for a sinesoid source. You can specify a max of 2 files.	none
Number of harmonics	`harms`	Number of harmonics for test (numerator) channel, if an array is not given. The harmonics start from ‘firstharm’ and go up to ‘firstharm’ + ‘harms’ - 1.	-
Num of lumps in approx.	`lumps`	Number of lumps used in the approximation.	

CDF Parameter Label	CDF Parameter	Description	Default
Num of lumps/dec in approx.	`dec`	Number of lumps per decade used in the approximation.	1.0
Number of noise/freq pairs	`FNpairs`		
Number of PWL/Time pairs	`tvpairs`		
Number of pairs of points	`tvpairs`		
Number of Polynomial Coeffs *Max Coefficient Number*	`polyCoef`		0
Number of ports	`padNum`		1
Number of Ports	`p`		1
Number of reference harmonics	`refharmsvec`	Array of desired harmonics for reference (denominator) channel.	0
Number of Probes	`probeCnt`		
Num of segments	`nseg`		
Number of structures in parallel	`npas`		
Number of turns on primary	`n1`	Number of turns on winding 1.	
Offset constant	`oc`		
Offset current	`io`		
Offset voltage *DC offset*	`offset`	Offset voltage in series with common terminal	-
Offset voltage	`vo`	Offset voltage in series with common terminal	-
Open switch resistance	`ro`	Resistance of a fully open relay.	

CDF Parameter Label	CDF Parameter	Description	Default
Open voltage	`vt1`	Relay resistance is 'ropen' at this voltage	
Open/close voltage	`vsw`		
Optional Nodes	`Opins`		
Optional Node Configuration	`soipOpNodes`	The options PinP, pinP_pinB, and pinP_pinB_Tnode correspond to each pin in the component. To know about the pins that these options correspond to, type spectre -h.	
Optional Node Configuration	`vbicOpNodes`	Substrate Node and Substrate & Temp. Node configurations.	
Optional Bulk Node_B	`pinB`		
Optional Substrate Node_S	`pinS`		
Optional Thermal Node_T	`pinT`		
Optional Thermal Node_dT	`pindT`		
Order of interpolation	`order`	Order of interpolation	-
Outer diam of toroidal core	`od`	Outer diameter of toroidal core	
Causal s-param output file	`outFile`	File used for storing the equivalent S-Parameter data based on corresponding time-domain model. The file format is touchstone. The instance name is added as a suffix and the file extension is added automatically.	-
PAC magnitude	`pacm`	Periodic AC analysis magnitude	
PAC magnitude (dBm)	`pacmDBm`		
PAC phase	`pacp`	Periodic AC analysis phase	

CDF Parameter Label	CDF Parameter	Description	Default
PAM4 mapping	`pam4_mapping`	Specifies a mapping from a pair of bit to 4-level voltages. Possible values are `0123`, `0132`, `0213`, `0231`, `0312`, `0321`, `1023`, `1032`, `1203`, `1230`, `1302`, `1320`, `2013`, `2031`, `2103`, `2130`, `2301`, `2310`, `3012`, `3021`, `3102`, `3120`, `3201`, and `3210`. PAM-4 signals have four distinct levels represented by 00, 01, 11 and 10 respectively. The transition between these levels depends on the value specified for this parameter. For example, 1203 represents the transition as 01 10 00 11. *PAM4 mapping* is only visible when you select `bit` *Source type* with *pam4* modulation.	`0132`
PAM modulation	`pam4_modulation`	Specifies the type of amplitude modulation; effective for *bit* and *prbs* sources. Possible values are: ■ `none`– Default behavior of the source. ■ `pam3`– Enables PAM3 modulation in the source. ■ `pam4`– Enables PAM4 modulation in the source.	`none`
param0	`param0`		
Parameter Type	`paramTyp`	Input type for other paramters. Possible values are cyclic and string.	cyclic

CDF Parameter Label	CDF Parameter	Description	Default
Passivity	`passivity`	Possible values are check, enforce, and no.	check
Passivity	`passivity_bbspice`	Possible values are fit_enforce and fit_weak_enforce.	check
Passivity Tolerance	`pabstol`	Absolute tolerance of passivity criteria.	1e-6

<table>
<tr><th>CDF Parameter Label</th><th>CDF Parameter</th><th>Description</th><th>Default</th></tr>
<tr><td>Pattern Parameter Data</td><td><code>data</code></td><td>Specifies the sequence in which the bits are to be arranged. It can be used to create both simple and nested bit patterns.

Note: Nested patterns are supported only for Spectre.

In case of nested patterns, ensure that the specified value conforms to the following rules:

■ An opening bracket for a pattern to be multiplied is preceded only by a pattern multiplier.

■ A pattern multiplier is preceded only by a space character or an opening bracket.

■ Every opening/closing bracket has a corresponding closing/opening bracket.

■ A closing bracket for a pattern to be multiplied is followed only by a space character or a closing bracket.

For example, if <code>data</code> = <code>4(01) 2(11001) 10</code>, then the final bit sequence is: <code>01 01 01 01 11001 11001 10</code>.</td><td>-</td></tr>
<tr><td>Period

Period of waveform</td><td><code>per</code></td><td></td><td></td></tr>
<tr><td>Period of the PWL

Period</td><td><code>pwlperiod</code></td><td>Period of the periodic PWL waveform</td><td></td></tr>
</table>

CDF Parameter Label	CDF Parameter	Description	Default
Period start time	`pwlperiodstart`	Period start time of the periodic PWL waveform	
Periphery of junction	`pj`		-
Phase delay	`phi`		
Phase for Sinusoid 1	`sinephase`		
Physical length	`len`	Effective physical length of magnetic path (used with 'vel' to specify electrical length of line).	
PJ(amplitude)	`pjamp`	When set for PRBS source or Bit source, the source has a periodic jitter whose amplitude is pjamp and whose frequency is pjfreq.	
PJ(frequency)	`pjfreq`	When set for PRBS source or Bit source, the source has a periodic jitter whose amplitude is pjamp and whose frequency is pjfreq.	
PJ(type)	`pjtype`	For PRBS source or Bit source, pjtype defines the type of periodic jitter. Possible valudes are sine, sawtooth, and square.	sine
Polarity of the buffer	`polarity`	Possible values: inv, noninv, or blank	inv
Poly Coeff 0	`c0`	Polynomial coefficients. At least one must be given.	
Poly Coeff 1	`c1`	Polynomial coefficients. At least one must be given.	
Poly Coeff 2	`c2`		
Poly Coeff 3	`c3`		
Poly Coeff 4	`c4`		
Polynomial argument	`polyarg`		

CDF Parameter Label	CDF Parameter	Description	Default
Port	`port`		
Port 1	`port1`		
Port 2	`port2`		
Port 3	`port3`		
Port 4	`port4`		
Port number	`num`		
Power of PWL waveform	`pwldbm`	Power of PWL waveform in dBm.	
Primary inductor	`pi`		
Precondition factor keyword	`precfac`		0.75
Prioritize Accuracy Range	`prioritizeAccuracyRange`	Specifes a frequency band of interest to prioritize the accuracy of *bbspice* fitting at this band.	t
Probe 1	`p1`	Devices through which the controlling currents flow.	
Probe 2	`p2`	Index of the probe ports through which the controlling currents flow.	
Probe 3	`p3`		
Probe 4	`p4`		
Probe Device Name	`probe`		
Profile	`profile`	Specifies what happens outside the range of approximation. Possible values are ff, df, fd, or dd. It is dd if abs(slope) >= 0.5 and ff otherwise.	-

CDF Parameter Label	CDF Parameter	Description	Default
Propogation velocity normalized	`vel`	Propagation velocity of the line given as a multiple of ‘c’, the speed of light in free space. (vel <= 1).	
Pulse width	`pw`		
PWL file name	`fileName`		
Pwl type	`pwlType`		
Quad temp co of lin cap	`tc2c`	Quadratic temperature coefficient of capacitor.	
Quantity of Port1 to Quantity of Port20	`portquantity1` to `portquantity20`	Quantities of ports. Use 0 for voltage and 1 for current.	
Rarely used parameters	`otherParaLabel`	Rarely used parameters.	-
Reactance	`x`	Reactance, that can have real number values. It can either be positive or negative.	-
Reference Value	`ref`	Sets the crossing reference for the control node. This parameter applies only when the Prbs source operates as a 3 or 4 terminal device. When the voltage across terminals 3 and 4 drops below ref, the output of the source is set to 0. If terminal 4 is not specified, it is assumed to be connected to ground.	-
Relative permittivity	`eps`	Substrate permittivity relative to a vacuum.	
Remove Device	`lxRemoveDevice`	Added to the auCDL view of a device to indicate that the device is shorted.	
Repeated function	`rpt`		

CDF Parameter Label	CDF Parameter	Description	Default
Res. for initial conds.	`rforce`	Resistance used when forcing initial conditions.	1.0 Ohm
Res of the winding	`resis`	Resistance of the winding.	-
Resistance	`r`	Resistance	1K Ohms
Resistance Form	`resform`	Default is ‘yes’ if ‘r < thresh’. Possible values are no or yes.	-
Reuse rational function data	`rational_func_reuse`	Possible values are 0,1, 2.	1
Rise Delay	`td01`	Time delay for 0 to 1 transition.	-
Rise on delay	`rise_on_dly`		
Rise off delay	`rise_off_dly`		
Rise time	`d2a_tr`		3n s
Rise time	`rise`	Time for transition to rise from Level 0 voltage to Level 1 voltage.	3n s
Rise time	`tr`	Time for transition to rise from Level 0 voltage to Level 1 voltage.	3n s
RJ(seed)	`rjseed`	The seed for random number generator, used in generating random jitter for PRBS sources.	1
RJ(rms)	`rjrms`	When set for PRBS source or Bit source, the source has a normally distributed random jitter, whose mean is zero and whose standard deviation is rjrms.	
RLCG data file	`file`	RLGC data file name.	
RLCG data file as Design var?	`rlgc_file_as_var`	Checks if the RLGC data file is used as a design variable.	
S to Z Transformation	`sxz`		

CDF Parameter Label	CDF Parameter	Description	Default
S-parameter Data File	`dataFile`	S-Parameter data file name.This file contains parameters, frequencies, or model information that can be analyzed by the Spectre simulator.	
S-parameter File	`file1`	S-parameter file name.	
S-parameter file as Design Var?	`sparam_file_as_var`	Checks if the S-parameter file is used as a design variable.	-
S-parameter data format	`datafmt`	Possible values are spectre, touchstone,citi, rfm, bnp.	
Sampling period	`tsamp`		
Scale factor *Amplitude scale factor*	`scale`	Scale factor	-
Second coupled inductor	`ind2`	Inductor to be coupled	
Second frequency name	`fundname2`		
Secondary inductor	`si`		
Seed	`seed`	Set registerlength=[2 ... 32] to choose a Maximum Length Sequence or define a custom PRBS by use of the parameters, lfsrtaps and lfsrseed.	-

CDF Parameter Label	CDF Parameter	Description	Default
Seed	`lfsrseed`	For PRBS source, lfsrseed is an integer array which sets the initial state of the LFSR. Array elements sets the locations of non-zero bits. Locations are 1-based and ordered from MSB to LSB of the LFSR. For example, assume lfsrtaps=[6] and lfsrseed=[1 3 5]. The width of the register is then 6 bits and the initial state is `101010`.	-
Signal amplitude	`sa`		
Signal frequency	`fs`		
Select IBIS Buffer Type	`bufferType`		
Select IBIS Buffer Variant	`bufferVariant2`		
Select IBIS Buffer Variant	`bufferVariant4`		
Self Heating Switch	`sel_heating`		
Sine DC level	`sinedc`		
Sinusoid Ampl 1 to Sinusoid Ampl 9	`vav1 - vav9`		
Skin effect res matrix per unit length	`rskin`		
Slope of imp on log/log scale	`slope`	Slope of the impedance when plotted on a log-log scale.	0.5
Smoothing Factor	`smoothing`		
Source diffusion area	`as`		
Source diffusion length	`ls`		
Source diffusion periphery	`ps`		
Source diffusion res squares	`nrs`		

CDF Parameter Label	CDF Parameter	Description	Default
Source type	`srcType`		
Source/drain selector	`geo`		
Specification type	`spec`		
Stamping method	`stamp`	Possible values are Y, S, YSTS, SSTS, DEFMBED.	Y
Start Frequency	`startFrequency`	Specifes the start frequency point in `bbsfreqband`.	t
Strength	`strength`	Quantity strength. Possible values are indifferent, suggest, insist, or override.	override
Subckt file	`modelFile`		
Substrate height	`h`		
Switch position	`position`	Switch position (0, 1, 2, ...).	0
Taps	`lfsrtaps`	For PRBS source, lfsrtaps is an integer array which sets the location of LFSR taps. Locations are 1-based and ordered form MSB to LSB of the LFSR. The largest element of the taps array is equal to the width of the LFSR.	-
Temp Rise Specifier	`triseSpec`		
dtmp -Temp rise from ambient	`dtmp`		
dtemp -Temp rise from ambient	`dtempn`		
Temp rise from ambient	`trise`	Temperature rise from ambient	-
Temperature coefficient 1 *Linear temp. coefficient*	`tc1`	Linear temperature coefficient First order temperature coefficient.	-

CDF Parameter Label	CDF Parameter	Description	Default
Temperature coefficient 2 *Quadratic temp. coeff.*	`tc2`	Quadratic temperature coefficient Second order temperature coefficient.	-
Temperature difference	`dtemp`		-
The order of indices	`datatype`		-
Thermal Node(T)	`Tnode`		-
Thermal Noise	`thermalnoise`	Thermal noise. Possible values are no or yes.	yes
Thermal noise model	`noisemodel`	Possible values are internal and external.	
Thermal resistance	`rth0`		
Thermal capacitance	`cth0`		
Threshold	`triggerthreshold`	For PRBS, when triggerthreshold is set and the source is instantiated with optional control terminals (terminals 3 and optionally 4; if terminal 4 is unspecified it is assumed to be connected to ground), triggerthreshold defines the crossing threshold for the trigger event. The event causes the emission of the next PRBS pulse.	
Time 1	`t1`		
Time 2	`t2`		
Time interval for switching	`ts`		
Time scale factor	`stretch`		
Time to x state	`a2d_tx`		1m s
Total Num of windings	`numOfL`		

CDF Parameter Label	CDF Parameter	Description	Default
Tran position	`tranPosition`	Position to which switch is set at the start of transient analysis.	0
Tran convolution parameters	`tranParaLabel`	Accuracy parameters for transient convolution.	0
Transaction time	`tt`		
Transconductance	`ggain`		
Transconductance (Obsolete)	`hggain`	**Note:** Parameter `hggain` is obsolete. Parameter `ggain` is used for both Spectre and HspiceD instead.	
Transition reference	`transitionrefe rence`	Defines the voltage swing for the duration of rise and fall time, as a percentage of val1 - val0. For example, transitionreference =100 means that the output voltage transitions from val0 to val1 in rise seconds. 90 means that it transitions from 0.1*(val1-val0) to 0.9*(val1-val0) in rise seconds, 80 means from 0.2*(val1-val0) to 0.8*(val1-val0), etc. Possible values are 100, 90, 80, 70 and 60.	100
Transition width	`twidth`		
Transmission line type	`linetype`	Transmission line type. Possible values are `microstrip`, `stripline`, `coplanar`, and `sublossline`.	
Transresistance	`hgain`		
Transresistance (Obsolete)	`hhgain`	**Note:** Parameter `hhgain` is obsolete. Parameter `hgain` is used for both Spectre and HspiceD instead.	

CDF Parameter Label	CDF Parameter	Description	Default
Type	`csType`		
Type of input of source	`inputtype`	Type of input of the source. Possible values are `single`, `and`, `nand`, `or`, `nor`, `npwl`, or `ppwl`.	`nil`
Type of input	`modelType`	Type of input selected for the component. Possible values are `RLGC`, `FieldSolver`, `Tline`, `and S-parameter`.	`nil`
Type of Port 1 to Type of Port20	`porttype1 - porttype20`	Types of ports. Use 0 for input port, 1 for output port, and 2 if the port is both input and output.	
Use S-parameters	`useSParamsCheckBox`	Controls whether S-parameters are specified or not.	
White noise expression	`white_noise`	Generates white noise with given power. Noise contributions with the same tag are combined for a module.	-
Width	`w`		
Width of Polysilicon	`wp`		
Width of metal capacitor	`wm`		
Width of the smoothing window	`smoothpts`		
Ext. Body Contact(PinP)	`PinP`		
Body Node	`BodyNodePin`		
Width of Emitter Window	`wemw`		
Value	`value`		
Volt/res conversion factor	`transfactor`		

CDF Parameter Label	CDF Parameter	Description	Default
Voltage eqn	`vol`		
Voltage gain	`egain`	Sets the Voltage gain for both Spectre and HspiceD.	
Voltage gain (obsolete)	`hegain`	**Note:** Parameter `hegain` is obsolete. Parameter `egain` is used to set the Voltage gain for both Spectre and HspiceD instead.	
Type of transfer char	`trfType`		
Type of Source	`typesrc`		
XF magnitude	`xfm`		
Type of rising & falling edge	`edgetype`		
Voltage 1	`v1`		
Voltage 2	`v2`		
Unity intercept point	`coef`		
use Img subckt	`uselmg`		
Use smooth data windowing	`usewindow`	Possible values are yes and no.	
To print fourier results on	`where`	Where Fourier results should be printed. Possible values are screen, logfile, and both.	logfile
Z state 1 to Z state 50	`Z1 - Z50`	Disable voltage N and netlist z as the value.	No
Dielectric loss cond matrix per unit length	`gdloss`	Dielectric loss conductance matrix per unit length.	
Frequency scale factor	`freqscale`	The frequency scale factor for frequency-dependent RLGC data and S-parameter data.	
Multiplicity factor	`mf`	Multiplicity factor.	1

CDF Parameter Label	CDF Parameter	Description	Default
Model name	`model`	Specifies the name of the model to be associated with the component.	-
Number of dielectric layers	`numlayer`	Number of dielectric layers.	1
Number of Ground Planes	`numgnd`	Number of ground planes,	1
Num of lines (excluding ref.)	`n`	Number of lines.	1
Conductor loss frequency	`fcmt`	Conductor loss frequency. Used in conjunction with `seriesr`, `qc`, or `alphac`.	
Frequency	`fmt`	Reference frequency, used in conjunction with `nl` to specify electrical length of line.	
Normalized length	`nlmt`	Normalized electrical length in wavelengths at the specified reference frequency of a lossless line.	
Propagation velocity normalized	`velmt`	Propagation velocity of the line given as a multiple of c, the speed of light in free space (vel <= 1).	
Corner frequency	`corner`	Corner frequency for skin effect.	
DC series res/Length	`dcr`	DC series resistance per unit length.	
Loss resistance per unit length	`seriesr`	Conductor (series) resistance per unit length at `fc`.	
Conductor loss at fc	`alphac`	Conductor loss measurement frequency(low loss approximation).	

CDF Parameter Label	CDF Parameter	Description	Default
Conductor loss quality factor	`qc`	Conductor loss quality factor at conductor loss measurement frequency (low loss approximation).	
Dielectric loss frequency	`fd`	Dielectric loss measurement frequency.	
Loss conductance per unit length	`shuntg`	Dielectric (shunt) conductance per unit length at conductor loss measurement frequency.	
Dielectric loss	`alphad`	Dielectric loss (low loss approximation).	
Dielectric loss quality factor	`qd`	Dielectric loss quality factor at dielectric loss measurement frequency (low loss approximation).	
Rel dielectric const of layers(er)	`er`	Relative dielectric constant.	
Dielectric layer thickness (d)	`layerthickness`	Dielectric layer thickness.	
Dielectric loss type	`dlosstype`	Dielectric loss type. The loss value is specified using the `dloss` parameter. Possible values are *sigma* and *tangent*.	
Dielectric layer loss	`dloss`	Dielectric layer loss. The loss can be in terms of dielectric conductivity or tangent loss, determined by the dlosstype parameter.	
Quality Factor/Q	`q`	Quality factor specified for capq and indq.	
Signal line width	`linewidth`	Signal line width,	
Signal line thickness	`linethickness`	Signal line thickess.	
Signal line height (h)	`lineheight`	Signal line height.	
Signal line spacing	`linespace`	Signal line spacing.	

CDF Parameter Label	CDF Parameter	Description	Default
Gnd Plane thickness	`gndthickness`	Ground plane thickness.	
Ground plane conductivity	`gndsigma`	Ground plane conductivity.	
Signal line conductivity	`linesigma`	Signal line conductivity.	
Charecteristic impedance	`z0`	Characteristic impedance of lossless line.	50 Ohms
Delay Time	`tdmt`	Time delay of a lossless line in seconds. A measure of the electrical length.	
Physical length	`len`	Effective physical length of magnetic path (used with 'vel' to specify electrical length of line).	
RLCG data file	`file`	RLGC data file name.	
RLCG data file as Design var?	`rlgc_file_as_var`	Checks if the RLGC data file is used as a design variable.	
S-parameter File	`file1`	S-parameter file path. Uses the S-parameter file name only when *S-parameter file as Design Var?* is enabled.	
S-parameter file as Design Var?	`sparam_file_as_var`	Checks if the S-parameter file is used as a design variable.	-
Transmission line type	`linetype`	Transmission line type. Possible values are `microstrip`, `stripline`, `coplanar`, and `sublossline`.	
Type of input	`modelType`	Type of input selected for the component. Possible values are `RLGC`, `FieldSolver`, `Tline, and S-parameter`.	`nil`

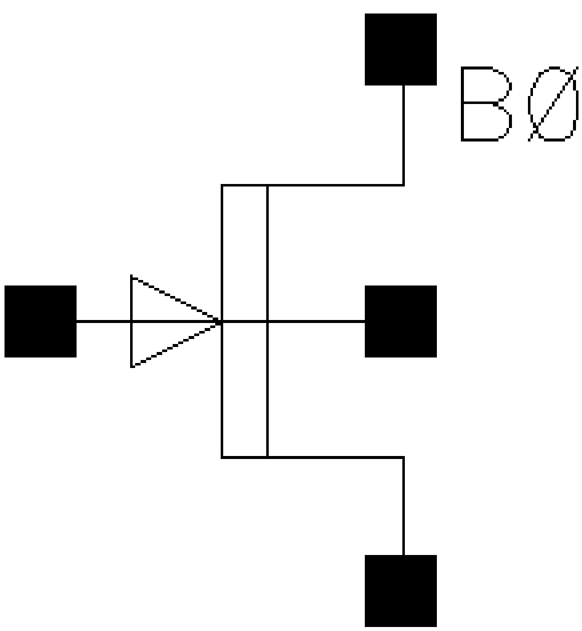
B

hspiceD Components

This section lists all the Analog Library components that are primarily supported by hspiceD.

Active Components

Symbol: nmes4



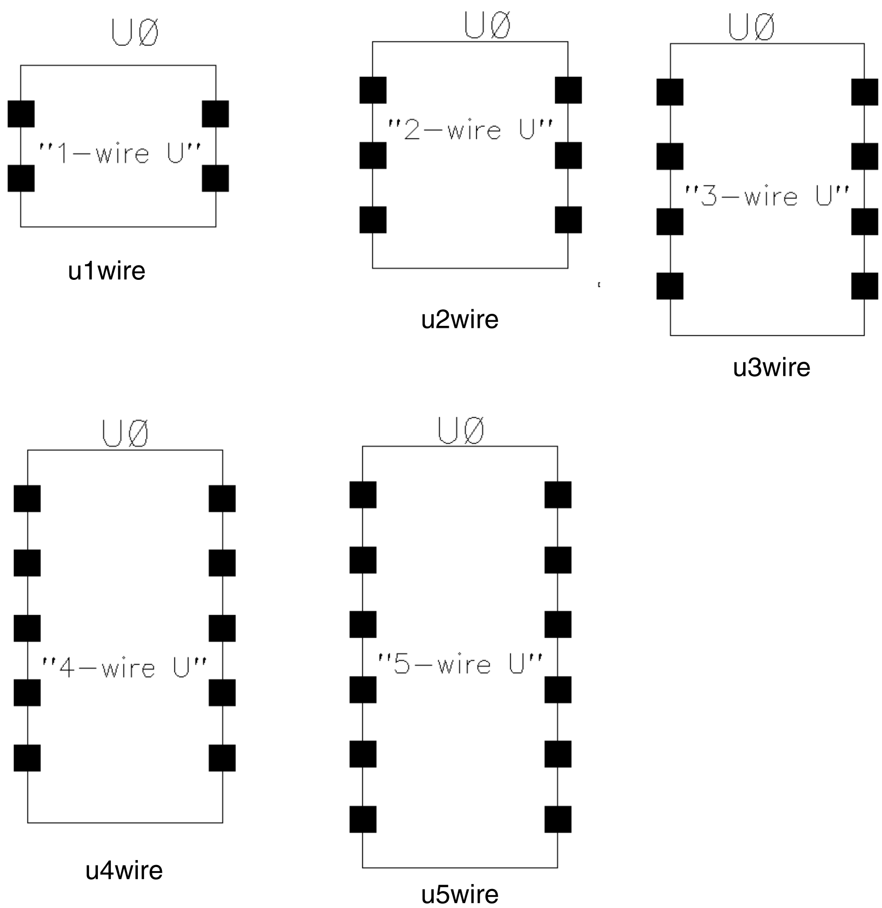
N-type MES FET Transistor with 4 Terminals

CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Device area	`area`	-	-	-	x	-
Device initially off	`off`	-	-	-	x	-
Drain source initial voltage	`Vds`	-	-	-	x	-
Gate source initial voltage	`Vgs`	-	-	-	x	-
Bulk source initial voltage	`Vbs`	-	-	-	x	-
Width	`w`	-	-	-	x	-
Length	`l`	-	-	-	x	-
Model name	`model`	-	-	-	x	-
Multiplier	`m`	-	-	-	x	-
Temperature difference	`dtemp`	-	-	-	x	-

Passive Components

Symbol: uxwire

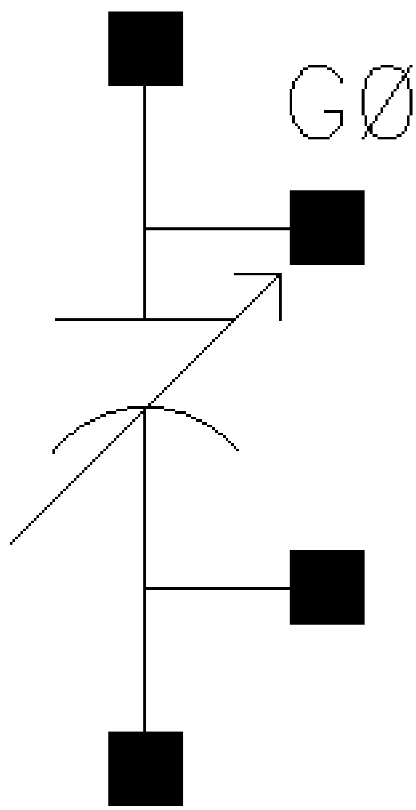


CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Model name	`model`	-	-	-	x	-
Length	`l`	-	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
# of lumps in element	`lumps`	-	-	-	x	-

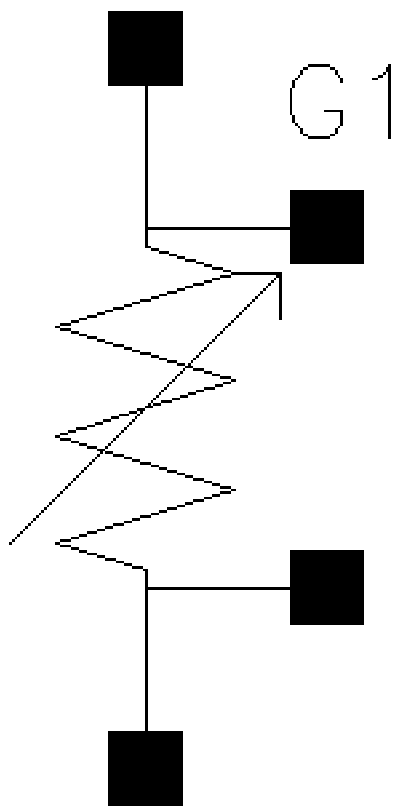
Symbol: vccap



CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Scale factor	scale	x	-	-	x	-
Multiplier	hm	x	-	-	x	-
Temperature coefficient 1	tc1	x	-	-	x	-
Temperature coefficient 2	tc2	x	-	-	x	-
Initial condition	hic	x	-	-	x	-
Delta	delta	x	-	-	x	-
Type	csType	x	-	-	x	-
Number of controlling pairs	xypairs	x	-	-	x	-
Controlling Volt 1	x1 - x20	x	-	-	x	-
Corresp Element 1	y1 -y20	x	-	-	x	-

Symbol: vcres



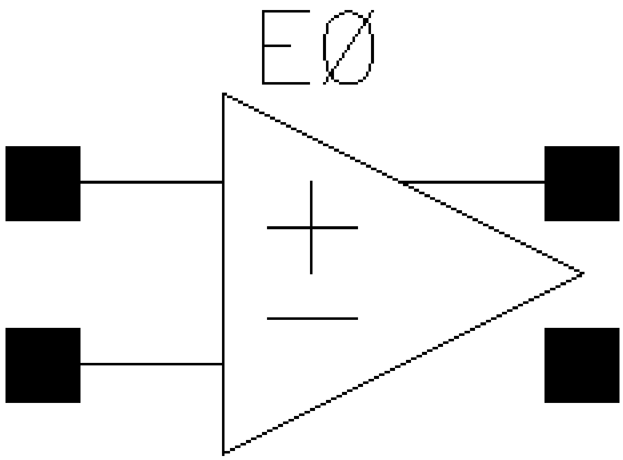
CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Type	`csType`	x	-	-	x	x
Volt/res conversion factor	`transfactor`	x	-	-	x	x

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Maximum output resistance	`maxr`	x	-	-	x	-
Minimum output resistance	`minr`	x	-	-	x	-
Scale factor	`scale`	x	-	-	x	x
Multiplier	`hm`	x	-	-	x	-
Temperature coefficient 1	`tc1`	x	-	-	x	x
Temperature coefficient 2	`tc2`	x	-	-	x	x
Initial condition	`hic`	x	-	-	x	-

Sources - Dependent Components

Symbol: iopamp

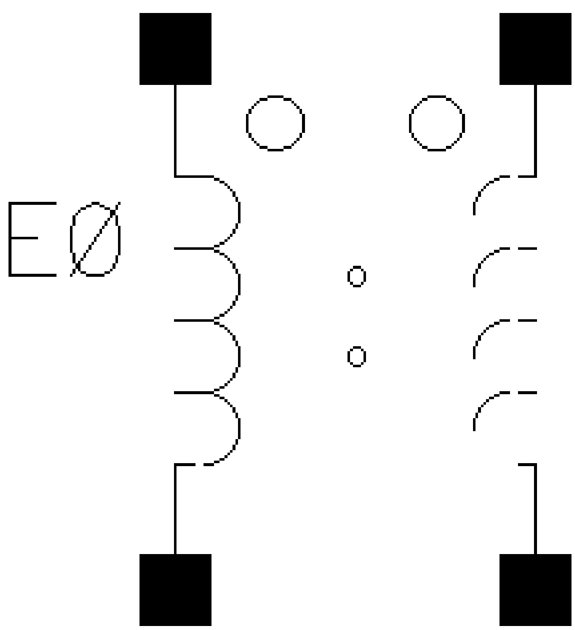


CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Scale factor	scale	-	-	-	x	-
Multiplier	hm	-	-	-	x	-
Temperature coefficient 1	tc1	-	-	-	x	-
Temperature coefficient 2	tc2	-	-	-	x	-
Initial condition	hic	-	-	-	x	-
Delta	delta	-	-	-	x	-
Type	csType	-	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Number of controlling pairs	`xypairs`	-	-	-	x	-
Controlling Volt 1	`x1 - x20`	-	-	-	x	-
Corresp Element 1	`y1 -y20`	-	-	-	x	-

Symbol: ixfmr

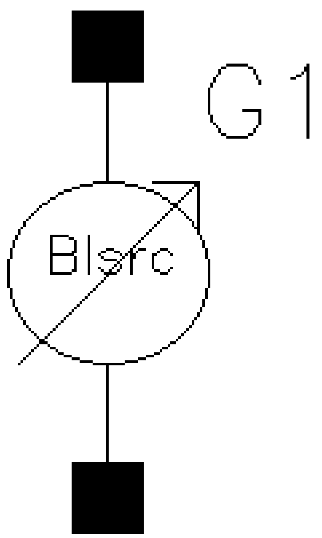


CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Coupling coefficient	`k`	-	-	-	x	-

Sources - Independent Components

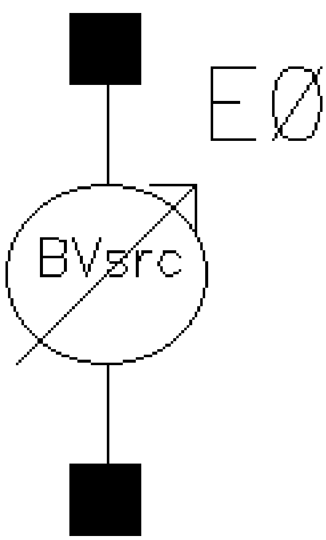
Symbol: bcs



CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Current eqn	`cur`	-	-	-	x	-
Min	`min`	-	-	-	x	-
Max	`max`	-	-	-	x	-
Scale factor	`scale`	-	-	-	x	-
Multiplier	`hm`	-	-	-	x	-

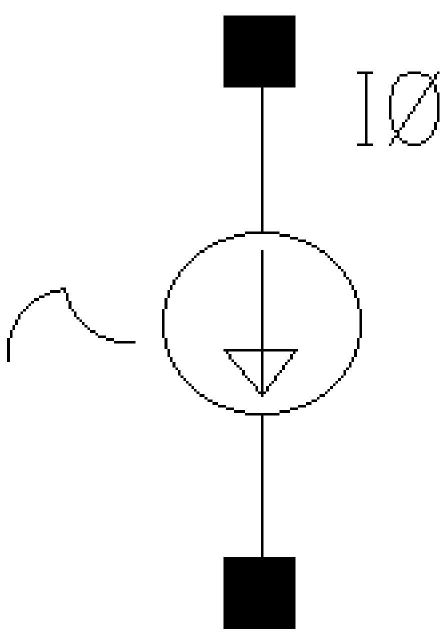
Symbol: bvs



CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Voltage eqn	`vol`	-	-	-	x	-
Min	`min`	-	-	-	x	-
Max	`max`	-	-	-	x	-

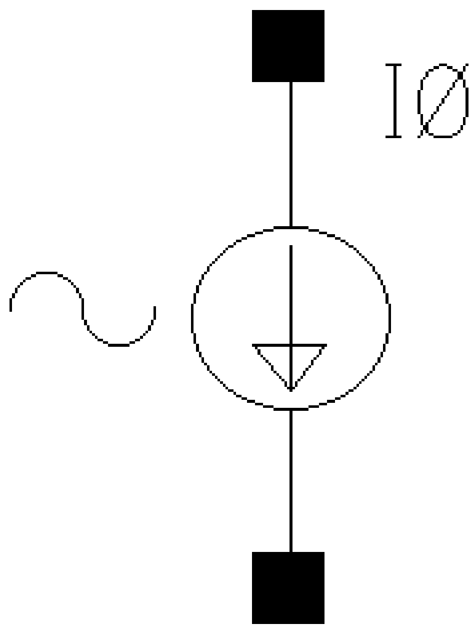
Symbol: iam



CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Signal amplitude	`sa`	-	-	-	x	-
Carrier frequency	`fc`	-	-	-	x	-
Modulation frequency	`fm`	-	-	-	x	-
Offset constant	`oc`	-	-	-	x	-
Delay time	`td`	-	-	-	x	-
DC source	`dc`	-	-	-	x	-
Multiplier	`m`	-	-	-	x	-
AC magnitude	`acm`	-	-	-	x	-
AC phase	`acp`	-	-	-	x	-

Symbol: isffm

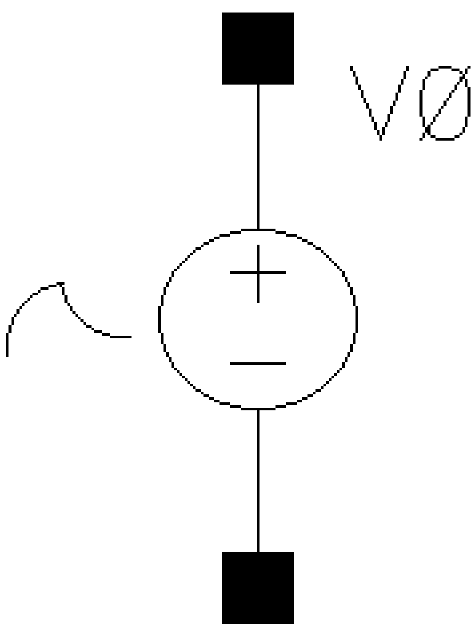


CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	-	-	-	x	-
AC phase	`acp`	-	-	-	x	-
DC current	`idc`	-	-	-	-	-
Offset current	`io`	-	-	-	x	-
Amplitude	`ia`	-	-	-	x	-
Frequency	`freq`	-	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Modulation index	`mdi`	-	-	-	x	-
Signal frequency	`fs`	-	-	-	x	-
Multiplier	`m`	-	-	-	x	-

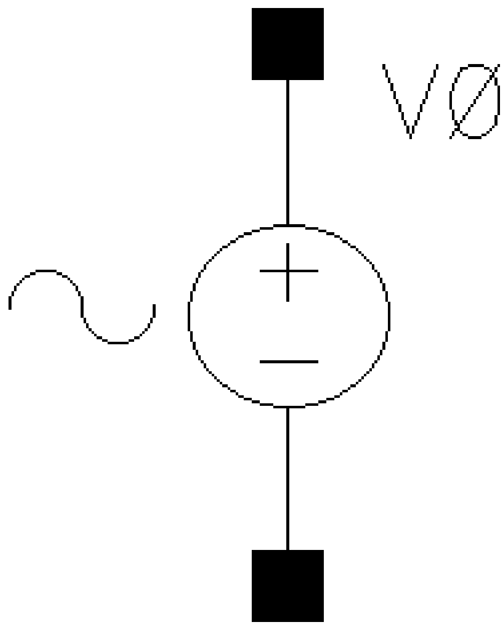
Symbol: vam



CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Signal amplitude	`sa`	-	-	-	x	-
Carrier frequency	`fc`	-	-	-	x	-
Modulation frequency	`fm`	-	-	-	x	-
Offset constant	`oc`	-	-	-	x	-
Delay time	`td`	-	-	-	x	-
DC source	`dc`	-	-	-	x	-
AC magnitude	`acm`	-	-	-	x	-
AC phase	`acp`	-	-	-	x	-

Symbol: vsffm



CDF Parameters

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
AC magnitude	`acm`	-	-	-	x	-
AC phase	`acp`	-	-	-	x	-
DC voltage	`vdc`	-	-	-	-	-
Offset voltage	`vo`	-	-	-	x	-
Amplitude	`va`	-	-	-	x	-
Frequency	`freq`	-	-	-	x	-
Modulation index	`mdi`	-	-	-	x	-

CDF Parameter Label	CDF Parameter	spectre	auCdl	auLvs	hspiceD	UltraSim
Signal frequency	fs	-	-	-	x	-
DC source	dc	-	-	-	x	-

Index

P

R

S

T

U

V

W

X

Z