core (Linear Magnetic Core)

Associated Symbols: `core`

License Requirements: None

Part Category: Magnetic Templates

Related Topics: Introduction to Magnetic Templates
corenl - nonlinear core (Jiles-Atherton)
corenl2 - nonlinear core (Preisach)
wind - winding

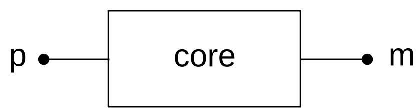
Functional Description

The **core** template models a linear magnetic core (note the magnetic connection points). It is normally used with the winding template **wind** to model an inductor or a transformer. You may either specify the inductance coefficient (`al`), or specify the three properties length (`len`), cross-sectional area (`area`), and relative permeability (`ur`). Specifying a value for `al` overrides the values for `len`, `area`, and `ur`.

Template Description Sections

core Connection Points

Name	Type	Description
p	magnetic	positive end of core
m	magnetic	negative end of core



Linear magnetic core (core)

core Symbol Properties

Property		
primitive	Description:	This symbol calls the template **core**, which models a linear magnetic core. It is normally used with the **wind** template to function as an inductor or transformer in an electrical circuit. You may either specify a value for its overall inductance coefficient (`al`) or specify the three values for length (`len`), cross-sectional area (`area`), and relative permeability (`ur`). Specifying a value for `al` overrides values for the other three.
ref	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, will be assigned by the schematic capture tool
	Example Input:	Can be any alpha-numeric string
al	Description:	Inductance coefficient. Overrides `len`, `area`, and `ur` properties (see Usage Notes).
	Default (units):	undef (nH/turns^2)
	Example Input:	85

Property		
`len`	Description:	Magnetic path length
	Default (units):	undef (m)
	Example Input:	1.27m
`area`	Description:	Cross-sectional area of magnetic path
	Default (units):	undef (m^2)
	Example Input:	6.35e-5
`ur`	Description:	Core relative permeability
	Default (units):	1 (—)
	Example Input:	1.2
`geo_units`	Description:	Allows the geometry units (`len`, `area`) to be specified in inch, meter, or cm.
	Default (units):	meter (—)
	Example Input:	cm
`ratings`	Description:	Structure of maximum ratings for this device.
	Default:	Available stress ratings for this model are: `bmax`. All values are set to undef by default.
	Example Input:	(bmax=5k)

Property		
part_type	Description:	Part type string. Limited to 9 characters.
	Default (units):	mag core (—)
	Example Input:	lin core
part_class	Description:	Part class string. Limited to 18 characters.
	Default (units):	lin generic (—)
	Example Input:	component
include_ stress	Description:	A flag to allow stress analysis in the netlisted template (see External Parameters).
	Default (units):	1 (—)
	Example Input:	To turn off stress measurement: include_stress=0

core Stress Arguments

Stress arguments are intended for use with the stress analysis, which is a part of the InSpecs Stress Analysis Option.

Name	Default	Units/Values	Description
bmax	undef	T	maximum flux density

core Post Processing Information

The variables in the following table are available for post-processing. You can specify them in a simulator signal list or as arguments to the `extract` command.

Name	Type	Units	Description
`mmf`	`val mmf`	A•t	`mmf` across core
`pm`	`val pm`	Wb/A•t	permeability of core (`pm` = B/H)
`f`	`var f`	Wb	magnetic flux (see Export Variables)
`b`	`val bsi`	T	flux density (see Export Variables)

core Export Variables

`f, b`

These are post-processing variables that can be referenced at the next higher level of the hierarchy.

core External Parameters

`include_stress`

This is a global parameter declared in `header.sin`. You can assign a value to it for an instance of this template (such as in a netlist) without affecting its global value in the rest of your design. For example, the following allows you to suppress stress analysis for `core.e1` only:

```
core.e1 b 0 = al=85, geo_units=inch, include_stress=0
```

core Usage Notes

The `al` argument is expressed in terms of the other three:

al = (u0 • ur • area)/len

where `u0` is the permeability of free space. This is provided because data sheets usually specify `al` instead of the other arguments.

core Netlist Examples

This example specifies `al` and suppresses stress analysis by using the external parameter `include_stress`.

```
core.e1 b 0 = al=85, geo_units=inch, include_stress=0
```

This example specifies `len`, `area`, and `ur`.

```
core.e2 mid1 0 = len=1.27m, area=6.35e-5, ur=1
```

corenl (Nonlinear Magnetic Core with Temperature Dependence)

Associated Symbols: `corenl`

License Requirements: OPT_TEMPLATE_LIB

Part Category: Magnetic Templates

Related Topics:
Introduction to Magnetic Templates
Magnetic Core Characterization Tool
core - linear magnetic core
corenl2 - nonlinear magnetic core with hysteresis
wind - winding

Functional Description

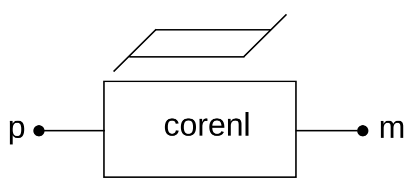
The **corenl** template models a nonlinear magnetic core (note the magnetic connection points). It is normally used with the **wind** template to model an inductor or a transformer in an electrical circuit. Because of the proprietary information of the Jiles-Atherton model, this template is encrypted so you can't read it past the argument declarations (see Model Arguments).

Template Description Sections

- Connection Points
- Symbol Properties
- Model Arguments
- Stress Arguments
- Post-Processing Information
- Export Variables

corenl Connection Points

Name	Type	Description
p	magnetic	positive end of core
m	magnetic	negative end of core



Nonlinear magnetic core (corenl)

corenl Symbol Properties

Property		
`primitive`	Description:	This symbol calls the template **corenl**, which models a nonlinear magnetic core. It is normally used with the **wind** template to function as an inductor or a transformer in an electrical circuit. You must specify either a string value for the argument `matl` or specify numeric values for all the other model arguments except `ptemp` and `tau`. Specifying a value for `matl` overrides the values for `model`. `Matl` is a string specifying a commercially available core material; `model` allows you to specify a group of core characteristics at as many different temperatures (`ptemp`) as desired.
`ref`	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, will be assigned by the schematic capture tool
	Example Input:	Can be any alpha-numeric string
`len`	Description:	Magnetic path length
	Default (units):	*Value required* (m)
	Example Input:	1.27m

Property		
area	Description:	Cross-sectional area of magnetic path
	Default (units):	*Value required* (m^2)
	Example Input:	5.9e-5
matl	Description:	Core material name (from library). This overrides the saber_model property. See Usage Notes.
	Default (units):	none (—)
	Example Input:	“3c8” (Note that the quotation marks are required.)
sf	Description:	Stacking factor adjustment to area in a laminated core. This is the ratio of area that is magnetically active.
	Default (units):	1 (—)
	Example Input:	1

Property		
saber_model	Description:	saber_model automatically maps to the template argument model, which is a grouping of ten B vs H model arguments defining a material at a specified temperature (ptemp).
	Default (units):	none (—)
	Example Input:	[(ptemp=27,ui=2700,hc=0.2, uhc=6000,br=1000,hsat=2.5, bsat=4500,hmax=15, bmax=4800, tau=0.6u,taulim=50n), (ptemp=100,ui=2100,hc=0.2, uhc=4300,br=900,hsat=2.0, bsat=3300,hmax=15,bmax=3600, tau=0.6u,taulim=50n)]
ja_model	Description:	Jiles-Atherton model arguments, used only if matl and saber_model are undefined.
	Default (units):	none (—)
b0	Description:	Initial flux density (B) in addition to anhysteretic. Units depend on units selection.
	Default (units):	0 (G if units=gauss; T if units=si)
	Example Input:	0

Property		
tempc	Description:	Core model temperature
	Default (units):	undef (°C)
	Example Input:	37
units	Description:	System of units for nonlinear magnetic properties saber_model and b0.
	Default (units):	gauss (—)
	Values:	gauss si
	Example Input:	si
geo_units	Description:	Allows the geometry units (len, area) to be specified in inch, meter, or cm.
	Default (units):	meter (—)
	Example Input:	cm
temp	Description:	Ambient temperature for temperature effects (see External Parameters).
	Default (units):	27 (°C)
	Example Input:	32

Property		
ratings	Description:	Structure of maximum ratings for this device.
	Default (units):	Available stress ratings for this model are: pdmax_ja, pdmax_jc, tjmax, tjmin, bmax. All values are set to undef by default (see Usage Notes).
	Example Input:	(bmax=5k,pdmax_ja=.25,tjmax=150)
rth_ja	Description:	Thermal resistance from junction to ambient (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	0.2
rth_jc	Description:	Thermal resistance from junction to case (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	0.2
rth_hs	Description:	Thermal resistance of an external heat sink (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	0.2

Property		
part_type	Description:	Part type string. Limited to 9 characters.
	Default (units):	mag core (—)
	Example Input:	nonlin core
part_class	Description:	Part class string. Limited to 18 characters.
	Default (units):	nonlin generic (—)
	Example Input:	component
include_stress	Description:	A flag to allow stress analysis in the netlisted template (see External Parameters).
	Default (units):	1 (—)
	Example Input:	To turn off stress measurement: include_stress=0

corenl Model Arguments

Name	Default	Units/Values	Description
model	undef	—	a grouping of ten B vs H model arguments defining a material at a specified temperature (ptemp)
ui	undef	—	initial relative permeability
uhc	undef	—	relative permeability at coercive level of H (hc). See Magnetic Hysteresis for more details.
bmax	undef	T or G	maximum value of B (units depend on units selection). See Magnetic Hysteresis for more details.
hmax	undef	A•t/m *or* Oe	maximum value of H (units depend on units selection). See Magnetic Hysteresis for more details.
bsat	undef	T or G	value of B at saturation (units depend on units selection). See Magnetic Hysteresis for more details.
hsat	undef	A•t/m or Oe	value of H at saturation; must be less than hmax (units depend on units selection). See Magnetic Hysteresis for more details.
br	undef	T *or* G	residual B (units depend on units selection). See Magnetic Hysteresis for more details.
hc	undef	A•t/m or Oe	coercive force (units depend on units selection). See Magnetic Hysteresis for more details.

Name	Default	Units/Values	Description
ptemp	27	°C	temperature at which model arguments are defined
tau	0	s	time constant for B vs. H frequency dependence.
taulim	0	s	frequency dependence limit (taulim should be specified as much less than tau)
ja_model[*]	[()]	—	grouping of Jiles-Atherton arguments (used only if matl and model are undefined)
a	undef	—	shape argument
alpha	undef	—	mean field argument
c	undef	—	domain wall flexing constant
k	undef	—	domain wall pinning constant
ms	undef	A/m	magnetization saturation
ptemp	27	°C	temperature for a, alpha, c, k, and ms
tau	0	s	time constant for B vs. H frequency dependence
taulim	0	s	frequency dependence limit (taulim should be specified as much less than tau)

corenl Stress Arguments

Stress arguments are intended for use with the `stress` analysis, which is a part of the InSpecs Stress Analysis Option.

Name	Default	Units /Values	Description
bmax	undef	T	maximum flux density
tjmax	undef	°C	maximum internal temperature
tjmin	undef	°C	minimum internal temperature
pdmax_ja	undef	W	maximum power dissipation with rth_ja
pdmax_jc	undef	W	maximum power dissipation with rth_jc and rth_hs

corenl Post Processing Information

The variables in the following table are available for post-processing. You can specify them in a signal list or as arguments to the `extract` command.

Name	Type	Units	Description
mmf	val mmf	A•t	magnetomotive force
hin	val hm	A•t/m	magnetic field strength of H across the magnetic input pins
mtotal	val mm	A/m	total magnetization
hg	val hg	Oersted	magnetic field strength of H
he	val hm	A•t/m	effective internal H field (a computational variable used internal to the template)
mirr	val mm	A/m	irreversible magnetization

Name	Type	Units	Description
mrev	val mm	A/m	reversible magnetization
temp_case	val tc	°C	case temperature
rth_hs_tjmax	val rth	°C/W	maximum heat sink thermal resistance
f	val f	Wb	flux (see Export Variables)
b	var bsi	Tesla	magnetic field density (see Export Variables)
bg	val bg	Gauss	magnetic field density of B (see Export Variables)
pwrd	val p	W	instantaneous power dissipation (see Export Variables)
tempj	val tc	°C	instantaneous junction temperature (see Export Variables)

corenl Export Variables

`pwrd, tempj, f, b, bg`

These are post-processing variables that can be referenced at the next higher level of the hierarchy.

corenl External Parameters

`temp, include_stress`

These are global parameters declared in header.sin. You can assign values to them for an instance of this template (such as in a netlist) without affecting their global values in the rest of your design. For example, the following netlist statement allows you to change the simulation temperature to 58°C for `corenl.e1` only:

```
corenl.e1 b 0 = len=3e-2, area=6e-5, model=[(ui=2700,
  uhc=6000, bmax=4800, hmax=15, bsat=4500, hsat=2.5,
  br=1000, hc=0.2, ptemp=20, tau=50n), (ui=2100, uhc=4300,
  bmax=3600, hmax=15, bsat=3300, hsat=2.0, br=900, hc=0.2,
  ptemp=100, tau=50n)], units=gauss, temp=58
```

corenl Model Description (the Jiles-Atherton Model)

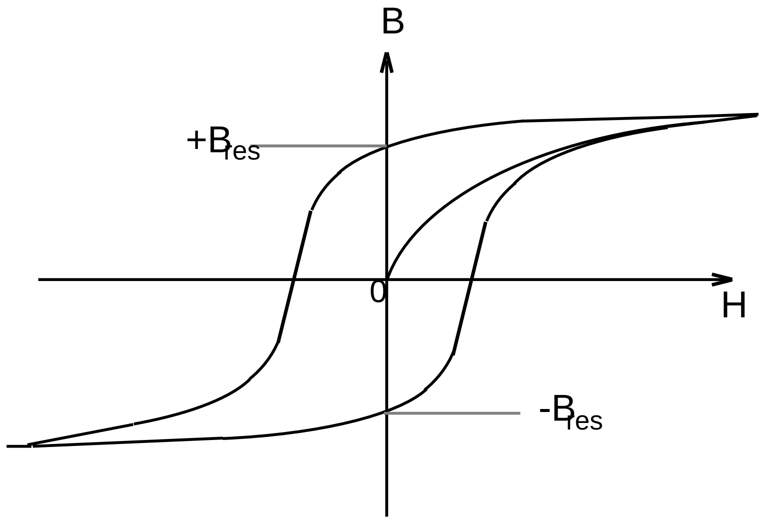
Hysteretic magnetization is characterized by the classic B-H curve described in the Magnetic Fundamentals topic Hysteresis. A basic simulation problem presented by hysteresis is that there is not a unique value of B for any given value of H.

Anhysteretic magnetization provides a method for selecting a unique value of B for a given value of H. The principal uses of the anhysteretic value of B are:

- to provide initial point values for B in a DC analysis
- to provide an equilibrium point for a transient analysis

The Jiles-Atherton model (see the References) provides a workable characterization of anhysteretic magnetization. It consists of a mathematical approach to the theoretical anhysteretic behavior of a core material, given a few basic input arguments for the material. The J-A model derives both the anhysteretic characteristic and the major hysteresis loop for a given material. This procedure has been found to compare favorably with experimental results for many ferromagnetic materials.

Since the value of B at time t is a function not only of the value of H at time t, but also of the initial values of H and B, this means that B_0 can assume any value within the hysteresis envelope, leading to a significant problem in a DC analysis. Even if H = 0 at t_0, B_0 can still lie anywhere between $-B_{res}$ and $+B_{res}$, as shown in the following figure. Although selecting $B_0 = 0$ seems like a reasonable value, it can be grossly inaccurate if $|H_0|$ is large.



Envelope for values of B0

Anhysteretic magnetization provides a more accurate method for selecting a unique value of B for a given value of H. This value of B lies between the upper and lower limits determined by the given value of H on the B-H curve. The principal advantage in using the anhysteretic value of magnetization is that it determines the same value of B for a given value of H regardless of whether H is increasing or decreasing. This means there is no hysteresis—hence the name anhysteretic.

The following equation describes the anhysteretic characteristic of a given material:

$$M_{an} = M_s \cdot (\coth[H_{eff}/A] - A/H_{eff})$$

where:

$H_{eff} = H + \alpha \cdot M_{an}$

M_{an} = anhysteretic magnetization

H_{eff} = effective magnetic field strength

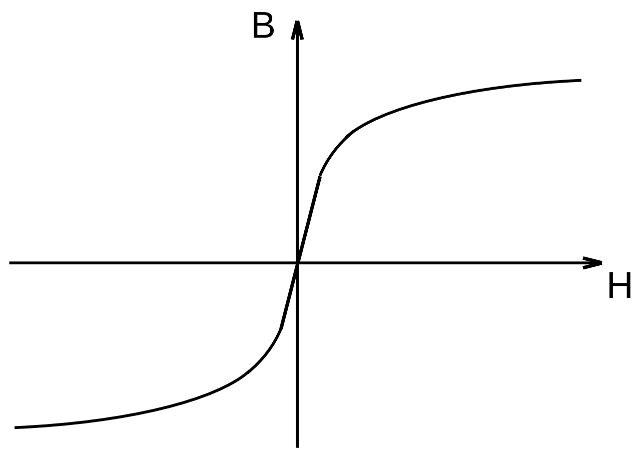
M_s = saturation magnetization

A = shape argument for magnetization

α = mean field argument for interdomain coupling

The J-A model represents the anhysteretic magnetization of a material as a state of magnetic equilibrium. This is an optimum configuration of magnetic characteristics, corresponding to a point of minimum internal energy.

The curve for the anhysteretic level lies between the upper and lower curves of the major hysteresis loop and passes through the origin, as illustrated in the following figure.



Anhysteretic magnetization level

The most significant advantage of this curve is that it provides a continuum of reasonable single values for B_0 for arbitrary values of H_0. The **corenl** templates allows the selection of a value of B_0 that differs from the anhystereticvalue, if desired: however, this is not generally recommended.

This state of anhysteretic magnetization is represented by a value of magnetization, M_{an} (which is related to flux density B) and lies between upper and lower hysteretic values of M (corresponding to upper and lower limits for values of B). The J-A model specifies a tendency for these hysteretic values of M ($M \neq M_{an}$) to approach M_{an}.

corenl Usage Notes

If the device is operating without a heat sink, use `pdmax_ja` for the power dissipation rating and `rth_ja` for the thermal resistance (`rth_ja` must be specified if you want the device temperature rise calculated).

If the device is operating with a heat sink, use `pdmax_jc` for the power dissipation rating and a combination of `rth_jc` and `rth_hs` for the thermal resistance.

This template is principally used as a “building block” template in conjunction with the **wind** template. There are three mutually-exclusive ways to specify characteristics of the **corenl** template:

- The `model` argument allows you to specify values for permeability, flux density, field strength, and coercive force at one or more temperatures. You must specify values for each argument contained within `model`.
- The `ja_model` argument allows you to specify values for Jiles-Atherton arguments (which are less commonly available).
- The `matl` argument allows you to specify a string for a commercially-available core material. The library of models for core materials contains characterization data for ferrite cores, silicon-iron laminate cores, and permalloy cores (see

corenl References).

Material	Manufacturer Name	String Name
Ferrite	Philips Components 3B7	"3B7"
	Philips Components 3B9	"3B9"
	Philips Components 3C6A	"3C6A"
	Philips Components 3C8	"3C8"
	Philips Components 3C85	"3C85"
	Philips Components 3D3	"3D3"
	Philips Components 3E2A	"3E2A"
	Philips Components 3E5	"3E5"
	Philips Components 3F3	"3F3"
	Philips Components 4C4	"4C4"
Silicon-Iron Laminate	Magnetic Metals 3% Silicon Iron, 1 50 EI	“si_re”
Square Permalloy	Magnetics Square Permalloy 80, 50 038 5d	“sq_perm_80”

Specifying a value for `matl` overrides the values for `model`, and `ja_model` is used only if `matl` and `model` are undefined.

corenl Netlist Examples

This example uses model to specify characteristics at two different temperatures (i.e., two values of ptemp), and specifies a value for the external parameter temp.

```
corenl.e1 b 0 = len=3e-2, area=6e-5, model=[(ui=2700,
 uhc=6000, bmax=4800, hmax=15, bsat=4500, hsat=2.5,
 br=1000, hc=0.2, ptemp=20, tau=50n), (ui=2100, uhc=4300,
 bmax=3600, hmax=15, bsat=3300, hsat=2.0, br=900, hc=0.2,
 ptemp=100, tau=50n)], units=gauss, temp=58
```

This example uses matl to specify a predefined, commercially-available core material.

```
corenl.e2 c 0 = len=2.8e-2, area=5.9e-5, matl="3c8",
 tempc=37
```

corenl Additional Examples

Listed below are different ways to specify a core model by changing the netlist entry, with a brief explanation of why each is useful. Refer to Section 3.4 of the *MAST Reference Manual* for more information on using the operators (.. -> and <-).

- If all desired arguments are listed (the simplest specification for a core), use the following:

```
corenl.e3 wb 0 = len=3e-2, area=6e-5,
 model=[(ui=2700, uhc=6000, bmax=4800, hmax=15,
 bsat=4500, hsat=2.5, br=1000, hc=0.2, ptemp=20,
 tau=50n)], tempc=60, units=gauss
```

- If referring to a model specification in more than one netlist entry (useful where several cores will share the same model or similar models), use the following:

```
corenl..model fred = [(ui=2700, uhc=6000,
 bmax=4800, hmax=15, bsat=4500, hsat=2.5, br=1000,
 hc=0.2, ptemp=20, tau=50n)]
```

This example defines the fred model, and must appear before any core that refers to it. The related example below uses the `fred` model with a length of 2 cm and an `area` of 0.5 cm^2.

```
corenl.e4 wb 0 = len=2e-2, area=5e-5, model=fred
```

- If one model references another model (useful if a new model is similar to another model), use the following:

```
corenl..ja_model duff = [(a=8.366, alpha=-100.8u,
 c=0.4500, k=14.2, ms=380777, ptemp=20, tau=600n,
 taulim=50n)]
```

The `duff` model defined above is used with a length of 2 cm and an `area` of 0.5 cm^2 as follows:

```
corenl.e5 wb 0 = len=2e-2, area=5e-5,
 ja_model=duff
```

corenl References

1. *Soft Ferrite Cores Short Form Catalog*, Philips Components, Magnetic Products Group, Saugerties, NY.
2. Colonel William T. McLyman, *Magnetic Core Selection for Transformers and Inductors*, Marcel Dekker Inc., 1982.
3. *Tape Wound Cores Design Manual*, TWC-400, Magnetics, Division of Sprang and Company, 900 E. Butler Road, Butler, PA, 16003.D.C.
4. Jiles and D.L. Atherton, Theory of Ferromagnetic Hysteresis, *Journal of Magnetism and Magnetic Materials*, Vol. 61, 1986.

corenl2 (Nonlinear Magnetic Core with Hysteresis)

Associated Symbols:	`corenl2`
License Requirements:	OPT_TEMPLATE_LIB
Part Category:	Magnetic Templates
Related Topics:	Introduction to Magnetic Templates Magnetic Core Characterization Tool core - linear magnetic core corenl - nonlinear core (Jiles-Atherton) wind - winding

Functional Description

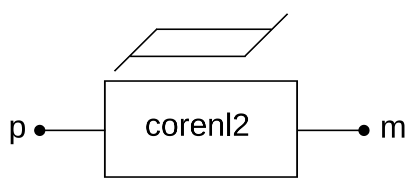
The **corenl2** template is a non-linear reluctance model with hysteresis based on the Preisach theory (see Model Description). It is normally used with the **wind** template to model an inductor or a transformer in an electrical circuit. This implementation of the Preisach model can cover a wide variety of B-H curves and includes frequency dependence to account for dynamic losses (eddy currents) in the core. Limitations of this model include:

- The model does not include temperature dependence.
- The model assumes that the magnetic field is uniformly distributed inside the core.
- This scalar model ignores the vectorial nature of the magnetization process.
- The static B-H curves do not have the accommodation property (i.e., the hysteresis loops always close at the end of any complete cycle).

Template Description Sections

corenl2 Connection Points

Name	Type	Description
p	magnetic	Positive end of core.
m	magnetic	Negative end of core.



Nonlinear magnetic core (corenl2)

corenl2 Symbol Properties

Property		
`primitive`	Description:	This symbol calls the template **corenl2**, which models a nonlinear magnetic core. It is usually used with the **wind** template to function as an inductor or a transformer in an electrical circuit.
`ref`	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, `ref` will be assigned by the schematic capture tool
	Example Input:	Can be any alpha-numeric string.
`len`	Description:	Magnetic path length.
	Default (units):	*Value required* (m)
	Example Input:	`5e-2`
`area`	Description:	Cross-sectional area of magnetic path.
	Default (units):	*Value required* (m^2)
	Example Input:	`5.9e-5`

Property		
`model`	Description:	Structure of input arguments which define the core material per unit volume. (see Model Arguments).
	Default (units):	none (—)
	Example Input:	`(hsat=40, bsat=1, tau=1e-5)`
`ratings`	Description:	Structure of maximum ratings for this device.
	Default (units):	Available stress ratings for this model are: bmax, pmax, pmaxavg. All values are set to undef by default (see Usage Notes).
	Example Input:	`(bmax=2)`
`part_type`	Description:	Part type string. Limited to 9 characters.
	Default (units):	mag core (—)
	Example Input:	`mag core`
`part_class`	Description:	Part class string. Limited to 18 characters.
	Default (units):	nonlin generic (—)
	Example Input:	`nonlin generic`

corenl2 Model Arguments

Model default values are given for Permalloy (2 mil).

Name	Default	Units/Values	Description
bsat	0.76	T	Value of B at saturation.
hsat	11.14	A/m	Value of H at saturation.
n	25	—	Mesh over which the integral of the Preisach distribution function is computed.
prm1	43	—	Shaping argument.
prm2	41	—	Shaping argument.
prm3	26	—	Shaping argument.
prm4	35	—	Shaping argument.
prm5	38	—	Shaping argument.
prm6	1	—	Shaping argument.
prm7	5	—	Shaping argument.
prm8	40	—	Shaping argument.
tau	1e-4	s	Time constant for B vs. H frequency dependence.
taulim	0.15e-4	s	Frequency dependence limit (taulim should be less than tau).

The prm shaping arguments are closely interrelated. The effect of any one of the prm shaping arguments on the B-H curve depends upon the settings of the other prm shaping arguments. To achieve the desired hysteresis curve, the shaping arguments can be adjusted using the Magnetic Core Characterization Tool.

corenl2 Stress Arguments

Stress arguments are intended for use with the `stress` analysis, which is a part of the InSpecs Stress Analysis Option.

Name	Default	Units /Values	Description
`bmax`	`undef`	T	Maximum induction.
`pmax`	`undef`	W	Maximum instantaneous power.
`pmaxavg`	`undef`	W	Average maximum power.

corenl2 Post Processing Information

The variables in the following table are available for post-processing. You can specify them in a signal list or as arguments to the `extract` command.

Name	Type	Units	Description
`mmf`	`val mmf`	A•t	Magnetomotive force.
`dbdt`	`var nu`	T	Time derivative of B.
`h_dc`	`var hm`	—	H field used as input to the static Classical Preisach Model (CPM).
`b_classical`	`val nu`	—	Normalized B returned by the CPM.
`bt`	`val nu`	—	Internal normalized B with reversible component.

Name	Type	Units	Description
h	val hm	A/m	H field (see Export Variables)
b	var bsi	T	Magnetic field density (see Export Variables).
hg	val hg	Oersteds	H field (see Export Variables).
bg	val bg	G	Magnetic field density (see Export Variables).
power	val p	W	Instantaneous power dissipation (see Export Variables).
flux	val f	Wb	Flux (see Export Variables).

corenl2 Export Variables

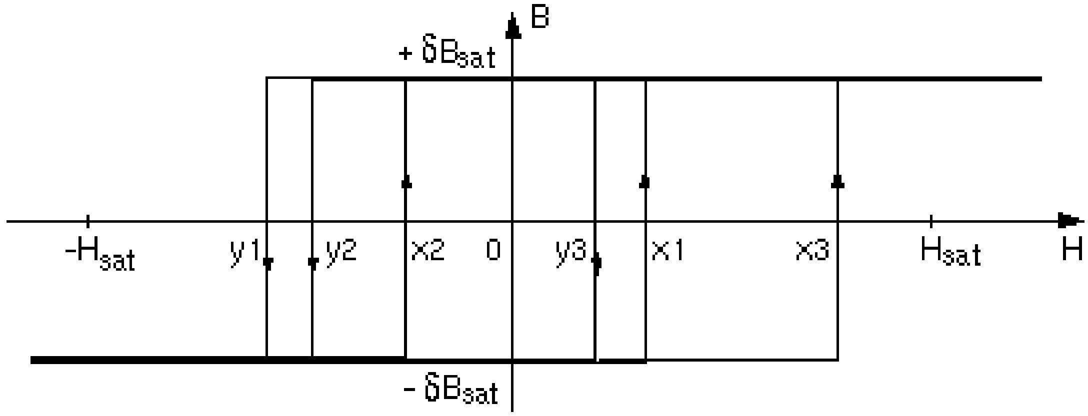
```
h, b, hg, bg, power, flux
```

These are post-processing variables that can be referenced at the next higher level of the hierarchy.

corenl2 Model Description (the Preisach Model)

The **corenl2** template models a nonlinear magnetic core that is based on the Preisach theory, which provides a graphical representation of the hysteresis phenomenon.

The classical Preisach model assumes that the magnetic material is composed of an infinite set of magnetic dipoles with rectangular hysteresis loops. The dipoles can either be in the +δBsat state or they can be in the -δBsat state. Each dipole can be represented in a plane as a point (x, y). The X-coordinate is the threshold for positive transition, and the Y-coordinate is the threshold for negative transition as shown below.

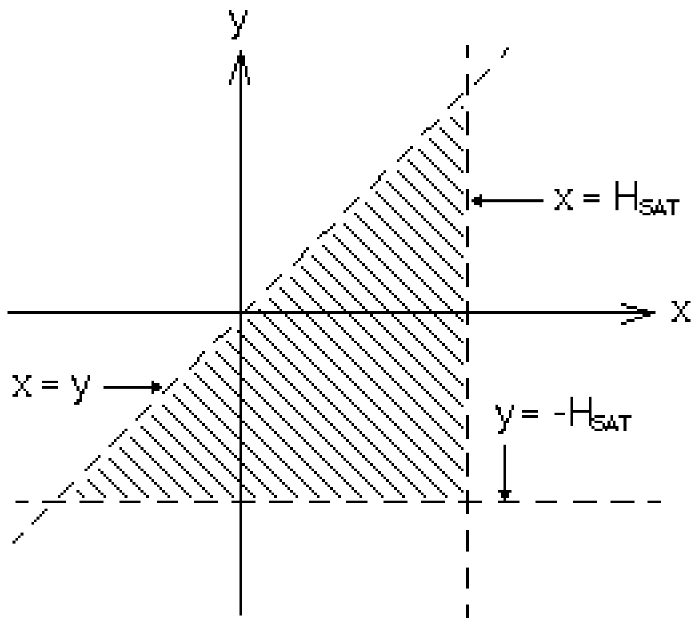


Examples of elemental dipole loops

The following conditions apply to the switching fields x_n and y_n of all dipoles:

x < Hsat	All dipoles have switched to the +δBsat state when H is above Hsat.
y > -Hsat	All dipoles have switched to the -δBsat state when H is below -Hsat.
x > y	The balance of energy over one cycle is always negative.

As a consequence of these three conditions, all the dipoles are located within an area referred to as the Preisach Triangle, shown below:



Preisach Distribution Function

The statistical distribution of x and y is described by the Preisach distribution function, F(x,y). F(x,y)·dx·dy is the probability of finding a dipole whose x value is located between x and x+dx and whose y value is between y and y+dy.

F(x,y) must have the following properties:

- The function is normalized:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(x, y) \cdot dx \cdot dy = 1$$

- The line y = -x is an axis of symmetry: F(x,y) = F(y,-x). This property defines the B-H curve as symmetrical.

- F(x,y) equals 0 outside the Preisach triangle.

Many mathematical definitions for the Preisach distribution function have been proposed. To cover the largest possible variety of hysteresis shapes, the distribution function used in **corenl2** combines a Gaussian and a Lorentzian function and is described by the five static parameters `prm`*n* of the `model` argument.

Calculating the Macroscopic Flux Density

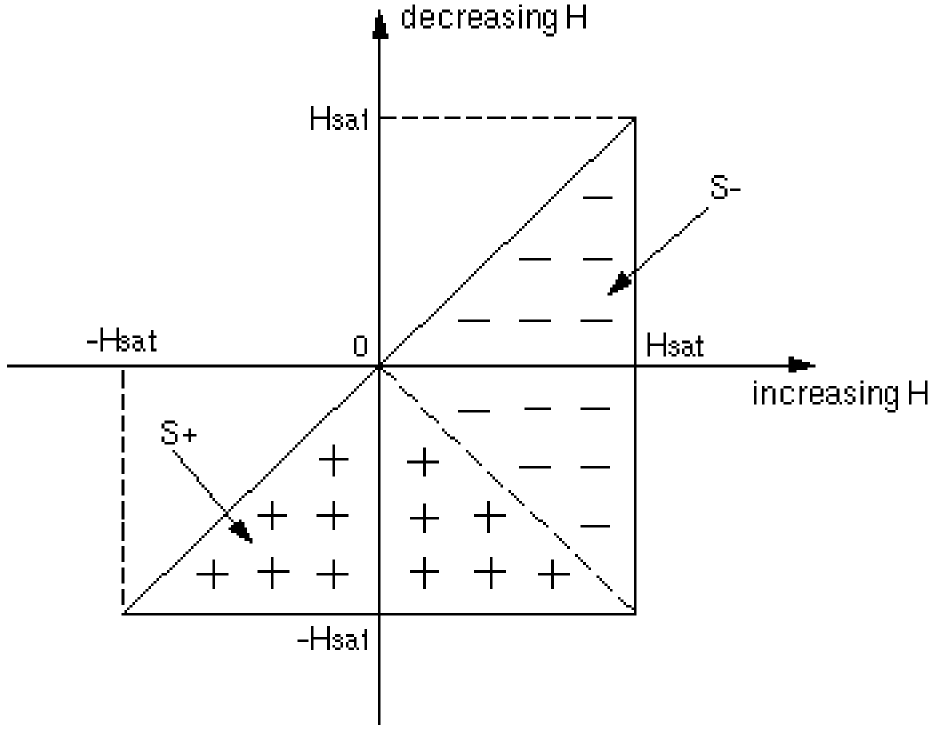
The macroscopic flux density B is obtained by summing the contribution of the dipoles in the $+\delta B_{sat}$ state and the contribution of the dipoles in the $-\delta B_{sat}$ state, corresponding to the integral of the Preisach function inside the triangle:

$$B_{classical} = B_{sat} \cdot \iint_{S+} F(x, y) \cdot dxdy - B_{sat} \cdot \iint_{S-} F(x, y) \cdot dxdy$$

where S+ and S- are the areas of the x-y plane which contain the dipoles in the +δBsat and -δBsat states respectively. These domains are a function of the core history and the current H value.

The Magnetization Process

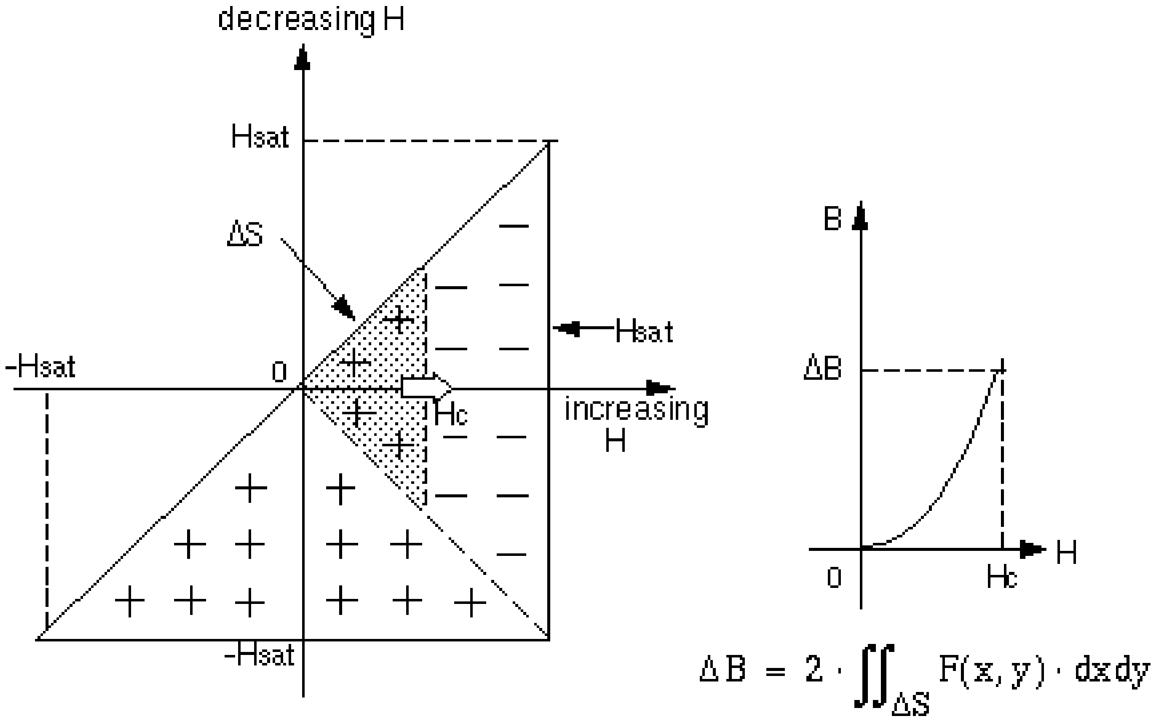
Preisach diagrams are used to illustrate how the domains evolve during a magnetization process. The Preisach diagram for the unmagnetized state is shown below.



Preisach diagram corresponding to the unmagnetized state.

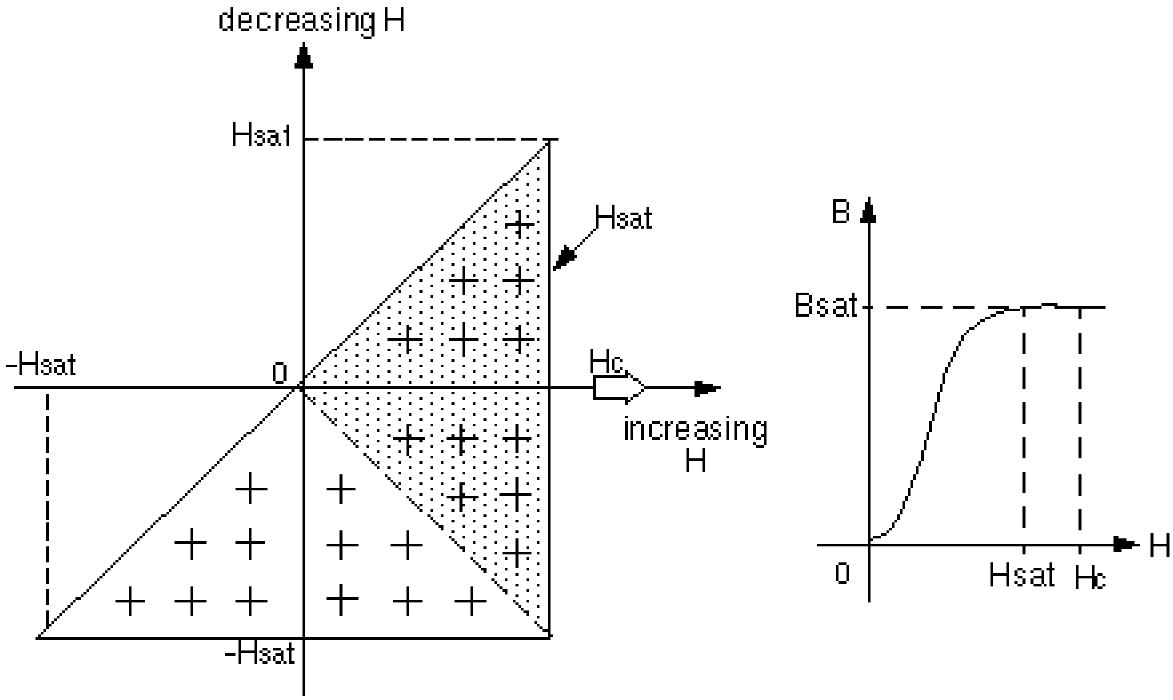
In the unmagnetized state, the two domains are of equal size, the distribution function is symmetrical about the line x = -y, and the macroscopic flux density B is equal to zero.

When H increases from zero to Hc, as shown by the initial magnetization trajectory below, the dipoles in the ΔS triangle change from the -δBsat state to the +δBsat state and contribute to an increasing value of B above zero.



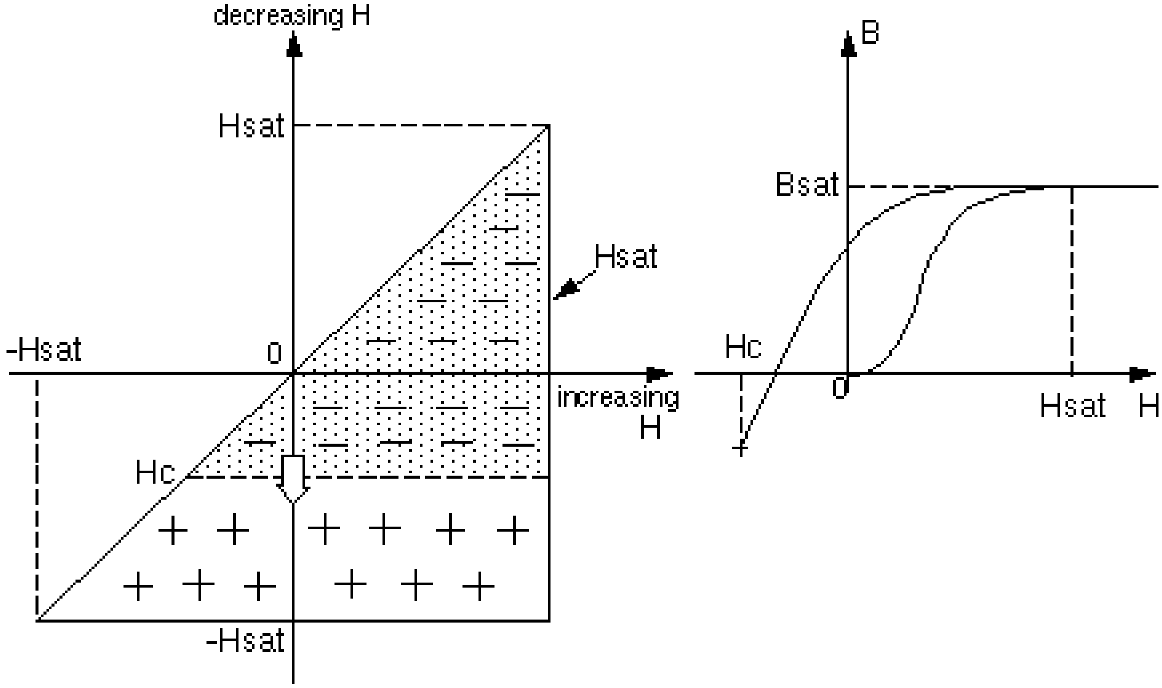
Initial magnetization trajectory

When H increases past Hsat, as shown by the magnetization trajectory at positive saturation below, all the dipoles will have switched to the +δBsat state and the B-H curve becomes flat.



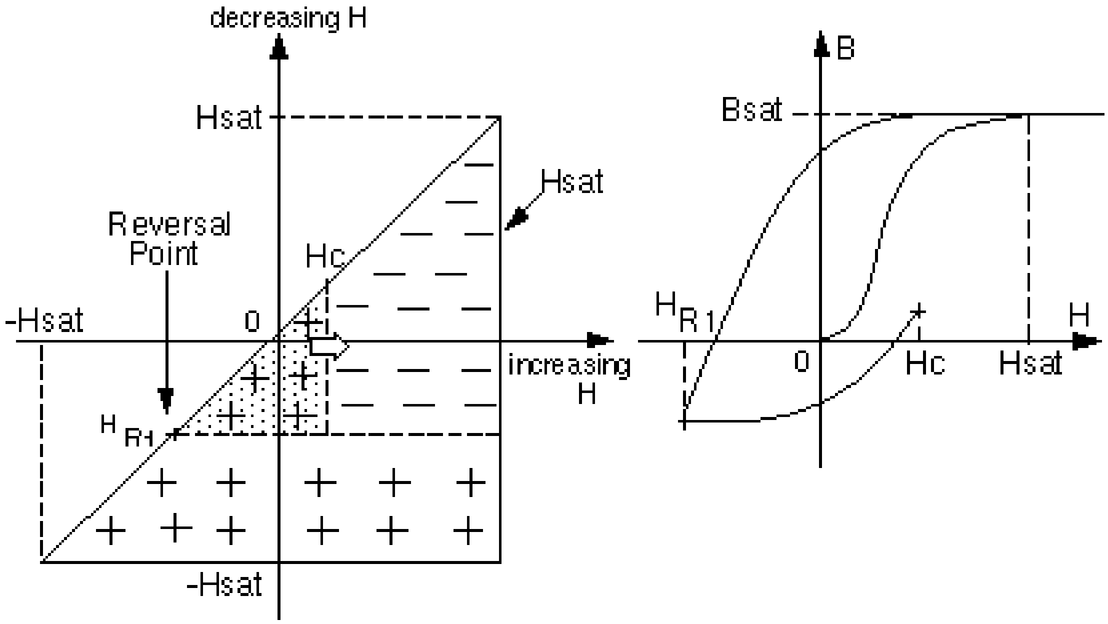
magnetization trajectory at positive saturation

When H decreases from saturation (as shown below), the state of the dipoles in the upper triangle (defined by lower boundary Hc) changes to the -δBsat state.



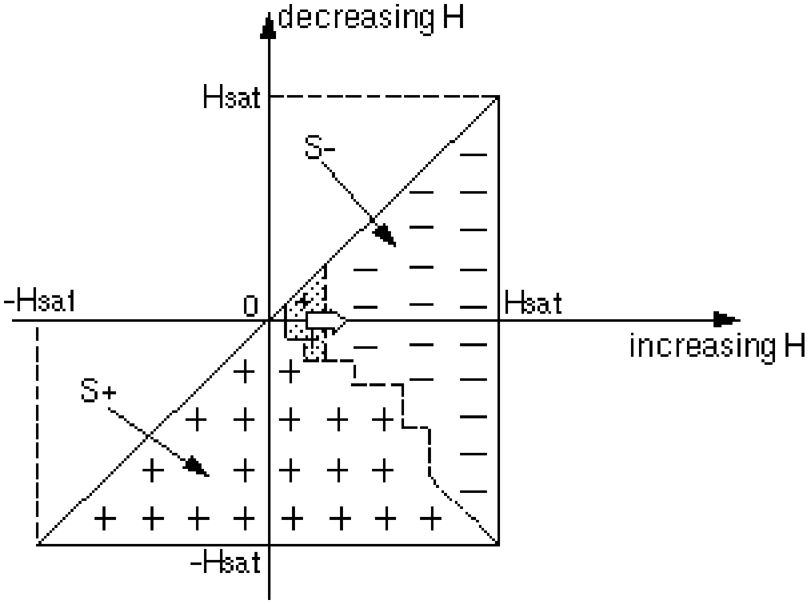
downward magnetization trajectory

If H is increased prior to reaching negative saturation (-Hsat), a reversal point, H_{R1}, is memorized as part of the core history as shown below.



Upward magnetization trajectory

The interface between the S^+ domain of +δBsat dipoles and the S^- domain of -δBsat dipoles is determined both by the core history and by the present state of magnetization. The H values of the reversal points that define the interface (shown below) are recorded in an array of state variables.

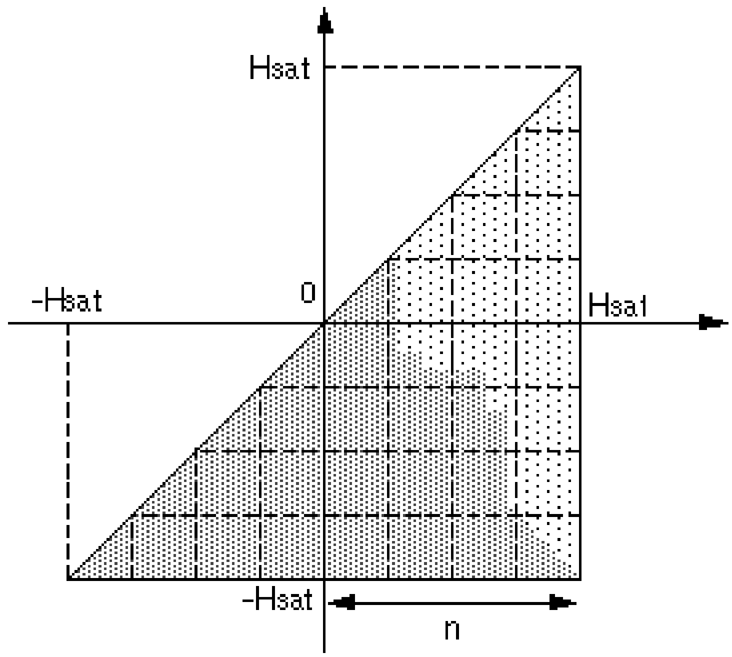


Preisach diagram after several magnetization cycles

Integration of the Preisach Function

From the figure above it is clear that the computation of B corresponds to an incremental integration of the Preisach function over the area recovered by the interface displacement between the previous time point and the current time point. The integral is computed numerically by approximating the Preisach function using a bilinear function which matches the Preisach function at the corner of each cell of the mesh shown below. The integral of this function is analytically defined. The template argument n defines the resolution of the mesh. The default value of n=25 is a trade-off between accuracy and computational speed.

$$F(x,y) \approx A \cdot xy + B \cdot x + C \cdot y + D$$



Preisach triangle mesh

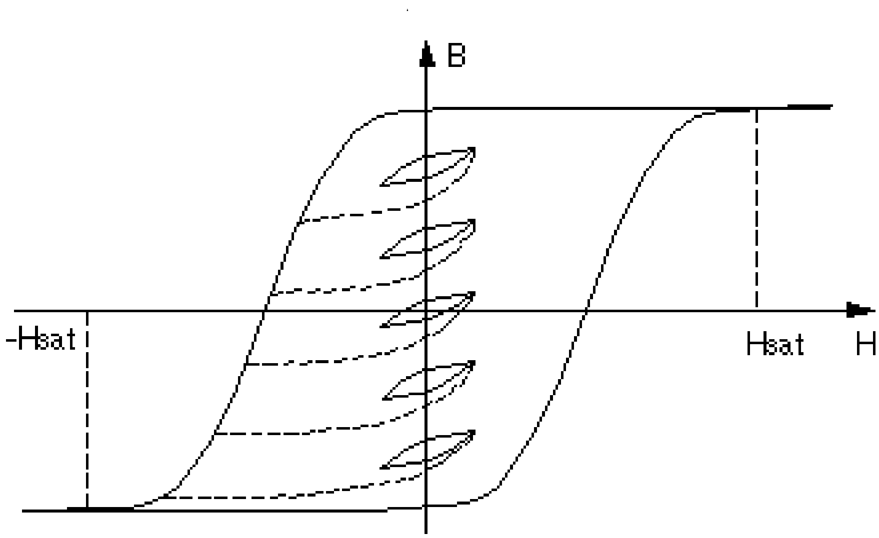
Limitations of the Preisach Model

Being fully irreversible, the classical Preisach model has three limitations:

- The minor loops are congruent in the hysteresis region.

- The induction B is completely flat in the saturation region.
- The initial permeability is zero.

The following figure illustrates these limitations:



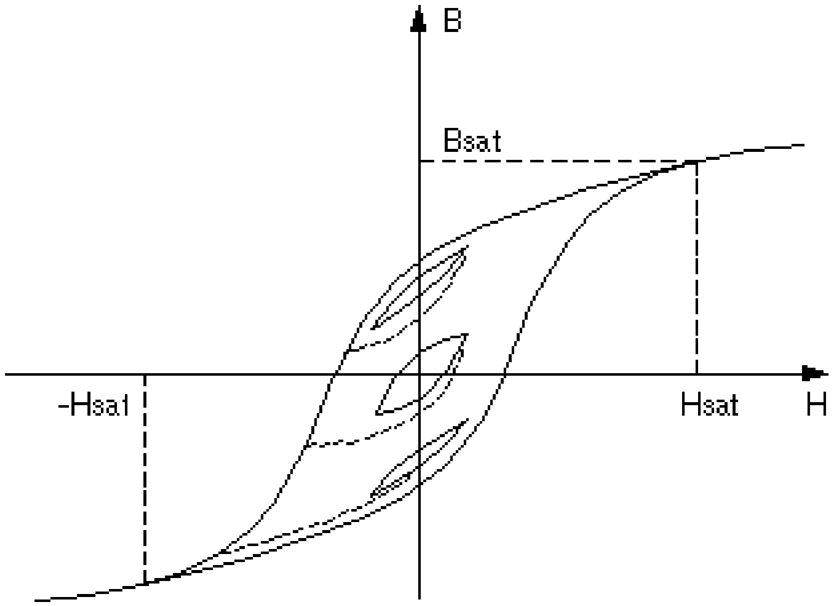
Fully irreversible magnetization

These limitations have been addressed by adding a reversible, energy conserving component in the calculation of the flux density. The reversible and irreversible components are combined as follows:

$$B = f((1-\beta)\cdot B_{classical} + \beta\cdot H)$$

where β corresponds to static parameter `prm7` and f(x) is a non-linear function defined by static parameter `prm6` of the `model` property.

With the inclusion of this reversible component, a larger variety of BH curves can be modeled with **corenl2**.The figure below is an example of the B-H relationship that results from adding the reversible component.



Combination of reversible and irreversible magnetization

Dynamic Behavior in the Preisach Model

The classical Preisach model is a static model. It assumes that the elemental dipoles can change their magnetization as fast as the magnetizing field. In reality the elemental dipoles respond to the external excitation with a certain time constant. This delayed magnetization causes a broadening of the BH loop at high frequency. The eddy currents are another dynamic effect that contributes to the broadening of the BH loop. The alternating magnetic flux in the material generates an electric field around the flux lines that induces currents if the material has electrical conductivity. These eddy currents flow in a plane perpendicular to the magnetic flux.

The dynamic behavior of the model is given by the same differential equation which describes the Jiles-Atherton model:

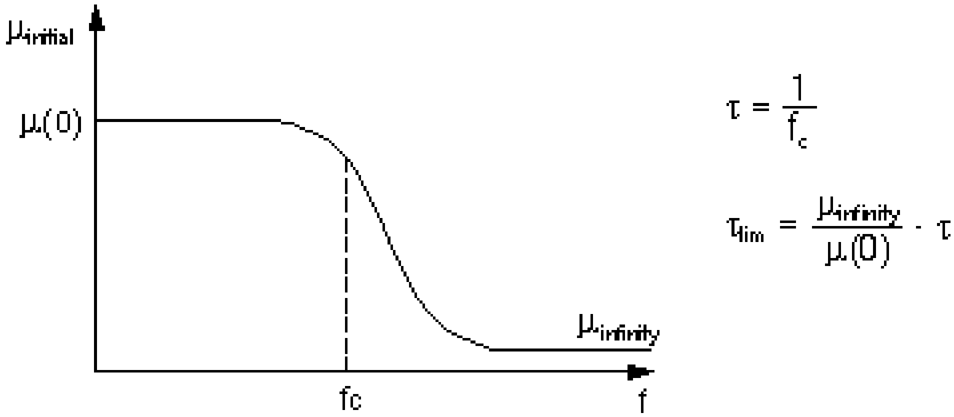
$$H - H_{in} = \tau_{lim} \cdot \frac{dH_{in}}{dt} - \tau \cdot \frac{dH}{dt}$$

where H is the magnetic field effectively applied across the model pins and Hin is the internal value used to calculate B according to the Preisach theory. The delay between H and Hin increases with frequency and results in a widening of the BH loop. This is implemented in **corenl2** as a linear transfer block described by:

$$\frac{H}{H_{in}} = \frac{1 + \tau_{lim} \cdot s}{1 + \tau \cdot s}$$

At low frequencies, the transfer function of the delay has the characteristics of a first order low-pass filter with cut-off frequency $fc=1/\tau$.

This dynamic block also accounts for the dependency of the initial permeability on frequency. The initial permeability is the ratio of change in magnetic induction to the change in magnetizing force in the unmagnetized state. In other words, it is the slope of the first magnetization curve at the point H = 0 (in Gauss-Oersted units). τ_{lim} defines the limiting value of μ at high frequency. The time constants τ and τ_{lim} can be determined from the initial permeability versus frequency characteristic, as shown below. These constants can be optimized to match the loss characteristics.



initial permeability versus frequency curve.

corenl2 Usage Notes

This template is principally used as a “building block” template in conjunction with the **wind** template.

For more information about using **corenl2**, see:

Magnetic Templates

Magnetic Building Block Templates

Magnetic Core Characterization Tool

The model **corenl2** assumes that the magnetic field is uniformly distributed inside the core. The model arguments `hsat` and `bsat` define the saturation levels. Above that point, the BH curve is anhysteretic. These arguments must be given in SI units (Amp/m and Telsa). The arguments `prm1` through `prm8` define the shape of the DC BH loop. The arguments `tau` and `taulim` control the dynamic aspects of the model (widening of the BH loop with increased frequency due to eddy currents). The argument `n` defines the mesh of the Preisach triangle over which the integral of the

Preisach distribution function is computed numerically. The default value n=25 provides a reasonable trade-off between accuracy and computational efficiency.

The component library contains models for commercially available magnetic cores, including ferrite cores, silicon-iron laminate cores, and permalloy cores.

corenl2 References

1. *Soft Ferrite Cores Short Form Catalog*. Philips Components, Magnetic Products Group, Saugerties, NY, 12477.
2. McLyman, Colonel William T. *Magnetic Core Selection for Transformers and Inductors*. Marcel Dekker Inc., 1982.
3. *Tape Wound Cores Design Manual*. TWC-400, Magnetics, Division of Sprang and Company, 900 E. Butler Road, Butler, PA, 16003.
4. Preisach, F. “Uber die magnetische Nachwirkung.” *Zeitschrift fur Physik*, Vol. 94, 1935, pp. 277-302.
5. Philips, D., et al. “Comparison of Jiles and Preisach Hysteresis Models in Magnetodynamics” *IEEE Trans. Magn*., Vol. 31, No. 6, 1995, pp.3551-3553.
6. Hui, S., J.Zhu and V. Ramsen. “A Generalized Dynamic Circuit Model of Magnetic Cores for Low- and High-Frequency Applications.” *IEEE Trans. Power Electronics*, Vol. 11, No. 2, 1996, pp.246-259.
7. Atherton, D., et al. “A New Approach to Preisach Diagrams.” *IEEE Trans. Magn*., Vol. 23, No. 3, 1987, pp. 1856-1854.
8. Torre, E. Della, J. Oti and G. Kadar. “Preisach Modeling and Reversible Magnetization.” *IEEE Trans. Magn*., Vol. 26, No. 6, 1990, pp. 3052-3058.
9. Torre, E. Della, and F. Vajda.”Physical Basis for Parameter

Identification in Magnetic Materials." *IEEE Trans. Magn.*, Vol. 32, No. 5, 1996, pp. 4186-4191.

10. Rouve, L., et al. "Application of Preisach Model to Grain Oriented Steels." *IEEE Trans. Magn.*, Vol. 31, No. 6, 1995, pp. 3557-3559.

11. Mayergoyz, I.D., and G. Friedman. "Generalized Preisach Model of Hysteresis." *IEEE Trans. Magn.*, Vol. 24, No. 1, 1988, pp. 212-217.

lnl (Nonlinear Inductor)

Associated Symbols: `lnl` `lnlh`

License Requirements: None

Part Category: Electrical Templates -- Passive Elements

Related Topics: Introduction to Magnetic Templates
core — Linear magnetic core
corenl — Nonlinear magnetic core
wind — Winding

Functional Description

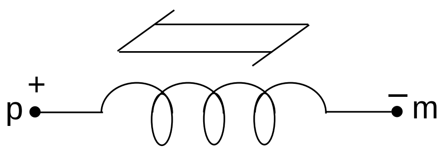
The **lnl** template models an inductor with a nonlinear core. It uses three other library templates (**wind**, **corenl**, **core**) as “building blocks” to implement the model. These templates are included in an internal netlist, and their argument values are passed in from the **lnl** arguments (refer to the Usage Notes).

Template Description Sections

[References](#)

lnl Connection Points

Name	Type	Description
`p`	`electrical`	positive inductor connection
`m`	`electrical`	negative inductor connection



Nonlinear inductor (lnl)

lnl Symbol Properties

Property		
`primitive`	Description:	This symbol calls the template **lnl**, which models an inductor with a non-linear core. It uses three other library templates (**wind**, **corenl**, and **core**) as building blocks. Because **corenl** is used as one of the building blocks, you must specify a string value for `matl` OR numeric values for all the `model` arguments except ptemp and tau. Specifying `matl` overrides the value for `model`.
`ref`	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, will be assigned by the schematic capture tool
	Example Input:	Can be any alpha-numeric string
`n`	Description:	Number of winding turns (see Usage Notes).
	Default (units):	*value required* (turns)
	Example Input:	10

Property		
nk	Description:	Coupling coefficient between windings (-1 <= nk <= 1) (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	1
r	Description:	Winding resistance (see Usage Notes).
	Default (units):	0 (Ω)
	Example Input:	1m
area	Description:	Cross-sectional area of magnetic path
	Default (units):	*value required* (m^2)
	Example Input:	5.9e-5
len_fe	Description:	Magnetic path of nonlinear core
	Default (units):	*value required* (m)
	Example Input:	2e-2
len_air	Description:	Airgap length (linear magnetic path)
	Default (units):	0 (m)
	Example Input:	1m

Property		
matl	Description:	Core material name (from library). This overrides the saber_model property (see Usage Notes).
	Default (units):	undef (—)
	Example Input:	“3c8” (note that the quotation marks are required.)
saber_model	Description:	saber_model automatically maps to the template argument model, which is a grouping of ten B vs H model arguments (see Usage Notes).
	Example Input:	[(ptemp=27, ui=2700, hc=0.2, uhc=6000, br=1000, hsat=2.5, bsat=4500, hmax=15, bmax=4800, tau=0.6u,taulim=50n), (ptemp=100,ui=2100, hc=0.2, uhc=4300, br=900, hsat=2.0, bsat=3300, hmax=15, bmax=3600, tau=0.6u,taulim=50n)]
units	Description:	System of units for nonlinear magnetic arguments (see Usage Notes).
	Default (units):	gauss (—)
	Values:	gauss si
	Example Input:	si

Property		
b0	Description:	Initial flux density (B) in addition to anhysteretic. Units depend on units selection (see Usage Notes).
	Default (units):	0 (T or G)
	Example Input:	0
sf	Description:	Stacking factor adjustment to area (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	1
tempc	Description:	Core model temperature
	Default (units):	undef (°C)
	Example Input:	25
ratings	Description:	Structure of maximum ratings for this device.
	Default:	Available stress ratings for this model are: pdmax_ja, pdmax_jc, tjmax, tjmin, imax, vmax. All values are set to undef by default (see Usage Notes).
	Example Input:	(imax=.1,pdmax_ja=.25,tjmax=150)

Property		
rth_ja	Description:	Thermal resistance from junction to ambient (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_jc	Description:	Thermal resistance from junction to case (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_hs	Description:	Thermal resistance of an external heat sink (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
part_type	Description:	Part type string. Limited to 9 characters.
	Default (units):	inductor (—)
	Example Input:	coil

Property		
part_class	Description:	Part class string. Limited to 18 characters.
	Default (units):	nonlin generic (—)
	Example Input:	component
temp	Description:	Ambient temperature for temperature effects (see External Parameters).
	Default (units):	27 (°C)
	Example Input:	32
include_ stress	Description:	A flag to allow stress analysis in the netlisted template (see External Parameters).
	Default (units):	1 (—)
	Example Input:	To turn off stress measurement: include_stress=0
ja_model	Description:	Jiles-Atherton model arguments, used only if matl and saber_model are undefined (see Usage Notes).
	Default:	none

lnl Model Arguments

Name	Default	Units/Values	Description
ja_model[*]	[( )]	—	grouping of Jiles-Atherton arguments (used only if matl and model are undefined)

lnl Stress Arguments

Stress arguments are intended for use with the stress analysis, which is a part of the InSpecs Stress Analysis Option.

Name	Default	Units/Values	Description
imax	undef	A	maximum winding current
vmax	undef	V	maximum winding voltage
tjmax	undef	°C	maximum internal temperature
tjmin	undef	°C	minimum internal temperature
pdmax_ja	undef	W	maximum power dissipation with rth_ja (see Usage Notes).
pdmax_jc	undef	W	maximum power dissipation with rth_jc and rth_hs (see Usage Notes).

lnl Post Processing Information

The variables in the following table are available for post-processing. You can specify them in a signal list or as arguments to the `extract` command.

Name	Type	Units	Description
v	val v	V	winding voltage
pd_wind	val p	W	winding power dissipation
pd_corenl	val p	W	core power dissipation
temp_case	val tc	°C	case temperature
rth_hs_tjmax	val rth	°C/W	maximum heat sink thermal resistance
pwr	group	W	grouping of power (pd_wind, pd_corenl, pwrd)
f	val f	Wb	flux (see Export Variables)
i	val i	A	winding current (see Export Variables)
pwrd	val p	W	instantaneous power dissipation (see Export Variables)
tempj	val tc	°C	instantaneous junction temperature (see Export Variables)

lnl Export Variables

```
pwrd, tempj, f, i
```

These are post-processing variables that can be referenced at the next higher level of the hierarchy.

lnl External Parameters

`temp, include_stress`

These are global parameters declared in header.sin. You can assign values to them for an instance of this template (such as in a netlist) without affecting their global values in the rest of your design. For example, the following allows you to change the simulation temperature to 58°C for `lnl.l1` only:

```
lnl.l1 na 0 = n=10, len_fe=2e-2, len_air=1m, area=5e-5,
  r=2m, model=[(ui=2700, uhc=6000, bmax=4800, hmax=15,
  bsat=4500, hsat=2.5, br=1000, hc=0.2, ptemp=30,
  tau=50n)], tempc=20, temp=58
```

lnl Usage Notes

If the device is operating WITHOUT a heat sink, use `pdmax_ja` for the power dissipation rating and `rth_ja` for the thermal resistance (`rth_ja` must be specified if you want the device temperature rise calculated). If the device is operating WITH a heat sink, use `pdmax_jc` for the power dissipation rating and a combination of `rth_jc` and `rth_hs` for the thermal resistance.

The value specified for `len_air` is used for calculations in the Arguments section. This accounts for the fringing effect that occurs when using an air gap.

This template contains three "building block" templates (**core**, **corenl**, **wind**), which are called by a netlist included within **lnl**. The building block templates receive values from the following **lnl** arguments: `n`, `len_fe`, `len_air_eq`, `nk`, `sf`, `r`, `model`, `matl`, `units`, `tempc`, `b0`, and `ja_model`. The choices for `units` are `gauss` and `si`, which select the system of magnetic units used for calculations.

You can specify a string value for `matl` OR numeric values for the `model` arguments OR numeric values for the `ja_model` arguments.

Specifying a value for `matl` overrides the values for `model`, and `ja_model` is used only if `matl` and `model` are undefined.

The material library presently contains characterization data for the following materials (see References). The string name of the material is shown between double quote marks (" ").

Material Name (Ferrite)	String Name
Philips Components 3B7	"3B7"
Philips Components 3C8	"3C8"
Philips Components 3C85	"3C85"
Philips Components 3D3	"3D3"
Philips Components 3E2A	"3E2A"
Philips Components 3F3	"3F3"
Philips Components 4C4	"4C4"
Philips Components 3B9	"3B9"
Philips Components 3C6A	"3C6A"

Material Name (Silicon Iron Lamination)	String Name
Magnetic Metals 3% Silicon Iron, 1 50 EI	"si_fe"

Material Name (Square Permalloy)	String Name
Magnetics Square Permalloy 80,50 038 5 d	"sq_perm_80"

lnl Netlist Examples

This example uses `model` to specify core characteristics at 30°C (i.e., `ptemp` = 30), and specifies a value for the external parameter `temp`.

```
lnl.l1 na 0 = n=10, len_fe=2e-2, len_air=1m, area=5e-5,
 r=2m, model=[(ui=2700, uhc=6000, bmax=4800, hmax=15,
 bsat=4500, hsat=2.5, br=1000, hc=0.2, ptemp=30,
 tau=50n)], tempc=20, temp=58
```

This example uses `matl` to specify a predefined, commercially-available core material.

```
lnl.l2 nb 0 = n=10, len_fe=3e-2, len_air=0, area=6e-5,
 nk=1, sf=1, r=1m, matl="3c8", tempc=25, b0=0
```

lnl References

1. *Soft Ferrite Cores Short Form Catalog*, Philips Components, Magnetic Products Group, Saugerties, NY, 12477.
2. Colonel William T. McLyman, *Magnetic Core Selection for Transformers and Inductors*, Marcel Dekker Inc., 1982.
3. *Tape Wound Cores Design Manual*, TWC-400, Magnetics, Division of Sprang and Company, 900 E. Butler Road, Butler, PA, 16003.

shortm (Magnetic Short)

Associated Symbols	`shortm`
License Requirements:	None
Part Category:	Magnetic Templates
Related Topics:	Introduction to Magnetic Templates

Functional Description

The **shortm** template is equivalent to a zero-valued mmf source. There is no mmf drop across the terminals, and there is unrestricted flux flowing through the short from the positive terminal to the negative terminal. Since this flux (`f`) is a `var`, it can be referenced by other templates.

You can use **shortm** to measure flux between two nodes or to temporarily short two nodes without having to change their names.

Template Description Sections

Connection Points

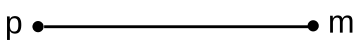
Symbol Properties

Post-Processing Information

Netlist Examples

shortm Connection Points

Name	Type	Description
p	magnetic	positive end of magnetic short
m	magnetic	negative end of magnetic short



Magnetic short (shortm)

shortm Symbol Properties

Property		
`primitive`	Description:	This symbol calls the template **shortm**, which models a magnetic short. This is equivalent to a zero-valued mmf source. There is no mmf drop across the terminals, and there is unrestricted flux flowing through the short from the positive terminal to the negative terminal. Since this flux (`f`) is a var, it can be referenced by other templates. You can use **shortm** to measure flux between two nodes or to temporarily short two nodes without having to change their names.
`ref`	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, will be assigned by the schematic capture tool
	Example Input:	Can be any alpha-numeric string

shortm Post Processing Information

The variables in the following table are available for post-processing. You can specify them in a simulator signal list or as arguments to the `extract` command.

Name	Type	Units	Description
`f`	`var f`	Wb	flux through short
`mmf`	`val mmf`	A·t	mmf across short

shortm Netlist Examples

```
shortm.ground mgnd 0
```

wind (Winding)

Associated Symbols	`wind`
License Requirements:	None
Part Category:	Magnetic Templates
Related Topics:	Introduction to Magnetic Templates **core** - linear magnetic core **corenl** - nonlinear core (Jiles-Atherton) **corenl2** - nonlinear core (Preisach)

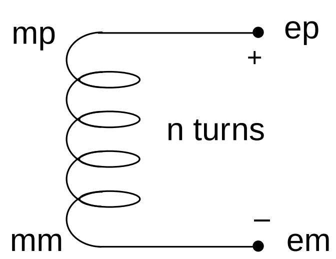
Functional Description

The **wind** template models an inductive winding. It has connection points for both magnetic and electrical circuits; it acts as a converter (or gyrator) between these two types of circuits.

Template Description Sections

wind Connection Points

Name	Type	Description
ep	electrical	positive electrical connection
em	electrical	negative electrical connection
mp	magnetic	positive magnetic connection
mm	magnetic	negative magnetic connection



Winding (wind)

wind Symbol Properties

Property		
`primitive`	Description:	This symbol calls the template **wind**, which models an inductive winding. It has connection points for both magnetic and electrical circuits; it acts as a converter (or gyrator) between these two types of circuits.
`ref`	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, will be assigned by the schematic capture tool
	Example Input:	Can be any alpha-numeric string
`n`	Description:	Number of winding turns
	Default (units):	*value required* (turns)
	Example Input:	10
`r`	Description:	Winding resistance
	Default (units):	0 (Ω)
	Example Input:	1m

Property		
nk	Description:	Coupling coefficient between windings, or effective turns factor (-1 <= nk <= 1).
	Default (units):	1 (—)
	Example Input:	1
area	Description:	Cross-sectional area of magnetic path
	Default (units):	undef (m^2)
	Example Input:	6.35e-5
temp	Description:	Ambient temperature for temperature effects (see External Parameters).
	Default (units):	27 (°C)
	Example Input:	32
ratings	Description:	Structure of maximum ratings for this device.
	Default:	Available stress ratings for this model are: pdmax_ja, pdmax_jc, tjmax, tjmin, amp_per_cmil, cmil_per_amp, vmax, imax. All values are set to undef by default.
	Example Input:	(imax=.1,pdmax_ja=.25,tjmax=150)

Property		
rth_ja	Description:	Thermal resistance from junction to ambient.
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_jc	Description:	Thermal resistance from junction to case.
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_hs	Description:	Thermal resistance of an external heat sink.
	Default (units):	undef (°C/W)
	Example Input:	.2
part_type	Description:	Part type string. Limited to 9 characters.
	Default (units):	winding (—)
	Example Input:	coil

Property		
part_class	Description:	Part class string. Limited to 18 characters.
	Default (units):	generic (—)
	Example Input:	component
include_stress	Description:	A flag to allow stress analysis in the netlisted template (see External Parameters).
	Default (units):	1 (—)
	Example Input:	To turn off stress measurement: include_stress=0

wind Stress Arguments

Stress arguments are intended for use with the stress analysis, which is a part of the InSpecs Stress Analysis Option.

Name	Default	Units/Values	Description
amp_per_cmil	undef	A/0.7854 sq. mil	current density; overridden by cmil_per_amp (see Usage Notes)
cmil_per_amp	undef	0.7854 sq. mil/A	inverse current density overridden by amp_per_cmil (see Usage Notes)

Name	Default	Units/Values	Description
tjmax	undef	°C	maximum internal temperature
tjmin	undef	°C	minimum internal temperature
pdmax_ja	undef	W	maximum power dissipation with rth_ja
pdmax_jc	undef	W	maximum power dissipation with rth_jc and rth_hs
vmax	undef	V	maximum voltage
imax	undef	A	maximum current

wind Post Processing Information

The variables in the following table are available for post-processing. You can specify them in a simulator signal list or as arguments to the extract command.

Name	Type	Units	Description
mmf	val mmf	A-t	mmf drop across winding
v	val v	V	total voltage drop across winding
vdrop	val v	V	voltage drop due to winding resistance
l	val l	H	winding inductance
jinv	val nu	A	inverse current density (1/j)
temp_case	val tc	°C	case temperature

Name	Type	Units	Description
rth_hs_ tjmax	val rth	°C/W	maximum heat sink thermal resistance
i	var i	A	current through winding (see Export Variables)
j	val nu	A	current density (see Export Variables)
f	var f	Wb	flux through winding (see Export Variables)
pwrd	val p	W	instantaneous power dissipation (see Export Variables)
tempj	val tc	°C	instantaneous junction temperature (see Export Variables)

wind Export Variables

```
pwrd, tempj, i, j, f
```

These are post-processing variables that can be referenced at the next higher level of the hierarchy.

wind External Parameters

```
temp, include_stress
```

These are global parameters declared in header.sin. You can assign values to them for an instance of this template (such as in a netlist) without affecting their global values in the rest of your design. For example, the following allows you to change the simulation temperature to 58°C for `wind.l1` only:

```
wind.l1 h1 0 mb 0 = n=10, r=1m, nk=1, temp=58
```

wind Usage Notes

A circular mil is an area of a circle that will fit in a square having a side equal to the circle's diameter. For example, if the radius of a circle was 0.5, one circular mil = $\pi(.5)^2$ = 0.7854 square mil, with a mil equal to 0.001 inch.

If the device is operating without a heat sink, use `pdmax_ja` for the power dissipation rating and `rth_ja` for the thermal resistance (`rth_ja` must be specified if you want the device temperature rise calculated).

If the device is operating with a heat sink, use `pdmax_jc` for the power dissipation rating and a combination of `rth_jc` and `rth_hs` for the thermal resistance.

wind Netlist Examples

This example specifies the required value for `n` and specifies a value for the external parameter `temp`.

```
wind.l1 h1 0 mb 0 = n=10, r=1m, nk=1, temp=58
```

xfr (Linear 2-Winding Transformer)

Associated Symbols: xfr

License Requirements: None

Part Category: Magnetic Templates

Related Topics: Introduction to Magnetic Templates
l — Linear inductor
ml — Mutual inductance

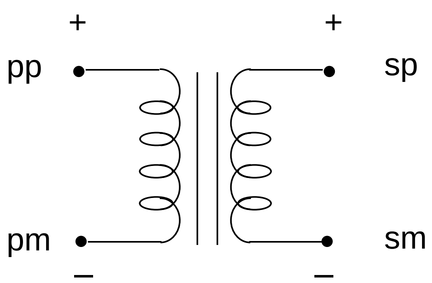
Functional Description

The **xfr** template models a 2-winding transformer template that allows you to specify electrical arguments (lp and ls) or magnetic arguments (np, ns, len, area, and ur). Specifying electrical arguments overrides the magnetic arguments (refer to the Usage Notes).

Template Description Sections

xfr Connection Points

Name	Type	Description
pp	electrical	positive connection of primary winding
pm	electrical	negative connection of primary winding
sp	electrical	positive connection of secondary winding
sm	electrical	negative connection of secondary winding



Linear 2-winding transformer (xfr)

xfr Symbol Properties

Property		
primitive	Description:	This symbol calls the template **xfr**, which models a 2-winding transformer template that allows you to specify electrical properties (lp, ls) or magnetic properties (np, ns, len, area, ur). Specifying electrical properties overrides the magnetic properties.
ref	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, is assigned by the schematic capture tool
	Example Input:	Can be any alpha_numeric string
lp	Description:	Inductance of primary winding. Specifying lp and ls overrides the magnetic properties (np, ns, len, area, ur) (see Usage Notes).
	Default (units):	undef (H)
	Example Input:	3e-2
np	Description:	Number of winding turns in primary winding (see Usage Notes).
	Default (units):	undef (turns)
	Example Input:	60

Property		
ls	Description:	Inductance of secondary winding. Specifying lp and ls overrides the magnetic properties (np, ns, len, area, ur) (see Usage Notes).
	Default (units):	undef (H)
	Example Input:	2e-3
ns	Description:	Number of winding turns in secondary winding (see Usage Notes).
	Default (units):	undef (turns)
	Example Input:	4
len	Description:	Magnetic path length (see Usage Notes).
	Default (units):	undef (m)
	Example Input:	3e-2
area	Description:	Cross-sectional area of magnetic path (see Usage Notes).
	Default (units):	undef (m^2)
	Example Input:	6e-5

Property		
ur	Description:	Relative permeability (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	1
k	Description:	Coupling coefficient (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	0.98
rp	Description:	Winding resistance of primary winding (see Usage Notes).
	Default (units):	0 (Ω)
	Example Input:	1m
rs	Description:	Winding resistance of secondary winding (see Usage Notes).
	Default (units):	0 (Ω)
	Example Input:	1m

Property		
temp	Description:	Ambient temperature for temperature effects (see External Parameters).
	Default (units):	27 (°C)
	Example Input:	32
ratings	Description:	Structure of maximum ratings for this device.
	Default:	Available stress ratings for this model are: pdmax_ja, pdmax_jc, tjmax, tjmin, vpmax, vsmax, ipmax, ismax. All values are set to undef by default (see Usage Notes).
	Example Input:	(ipmax=.1,pdmax_ja=.25,tjmax=150)
rth_ja	Description:	Thermal resistance from junction to ambient (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_jc	Description:	Thermal resistance from junction to case (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2

Property		
`rth_hs`	Description:	Thermal resistance of an external heat sink (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
`part_type`	Description:	Part type string. Limited to 9 characters.
	Default (units):	xformer (—)
	Example Input:	Z-matched
`part_class`	Description:	Part class string. Limited to 18 characters.
	Default (units):	two winding (—)
	Example Input:	linear two winding
`include_stress`	Description:	A flag to allow stress analysis in the netlisted template (see External Parameters).
	Default (units):	1 (—)
	Example Input:	To turn off stress measurement: `include_stress=0`

xfr Stress Arguments

Stress arguments are intended for use with the stress analysis, which is a part of the InSpecs Stress Analysis Option.

Name	Default	Units/Values	Description
pdmax_ja	undef	W	maximum power dissipation with rth_ja
pdmax_jc	undef	W	maximum power dissipation with rth_jc and rth_hs
tjmax	undef	°C	maximum internal temperature
tjmin	undef	°C	minimum internal temperature
vpmax	undef	V	maximum primary voltage
vsmax	undef	V	maximum secondary voltage
ipmax	undef	A	maximum primary current
ismax	undef	A	maximum secondary current

xfr Post Processing Information

Name	Type	Units	Description
vp	val v	V	primary voltage
vs	val v	V	secondary voltage
ip	val i	A	primary current (see Export Variables)
is	val i	A	secondary current (see Export Variables)
pd_lp	val p	W	primary winding power dissipation
pd_ls	val p	W	secondary winding power dissipation
temp_case	val tc	°C	case temperature
rth_hs_tjmax	val rth	°C/W	maximum heat sink thermal resistance
pwrd	val p	W	instantaneous power dissipation (see Export Variables)
tempj	val tc	°C	instantaneous junction temperature (see Export Variables)
v	group	V	grouping of voltages (vp, vs)
i	group	A	grouping of currents (ip, is)
pwr	group	W	grouping of power (pwrd, pd_lp, pd_ls)

xfr Export Variables

```
pwrd, tempj, ip, is
```

These are post-processing variables that can be referenced at the next higher level of the hierarchy.

xfr External Parameters

`pwrd, tempj, ip, is`

These are global parameters declared in `header.sin`. You can assign values to them for an instance of this template (such as in a netlist) without affecting their global values in the rest of your design. For example, the following allows you to change the simulation temperature to 58°C for `xfr.t1` only:

```
xfr.t1 g1 0 h3 h2 = lp=3e-2, ls=2e-3, k=0.98, rp=1m, rs=1m,
  temp=58
```

xfr Usage Notes

This template allows you to specify electrical or magnetic arguments. Specifying values for primary and secondary inductances (`lp`, `ls`) overrides values for the magnetic arguments. The values for `k`, `rp`, and `rs` are used with both groups.

Electrical	`lp, ls`
Magnetic	`np, ns, len, area, ur`

You can specify a value for the coupling coefficient (`k`) in the following range:

$$-1 \le k \le +1$$

For an iron core, the value of `k` is nearly 1. For an air core, `k` assumes a very small positive value. If `k` is specified less than zero, it reverses the polarity of the transformer.

This template uses **l** and **ml** as “building block” templates, calling them from a netlist that is part of the **xfr** template. If `lp` and `ls` are specified, these values are passed directly to the building block inductors (`l1`, `l2`). If the magnetic arguments are used instead, **xfr** calculates the inductances internally (`lpri`, `lsec`) and then passes them to `l1` and `l2` in the netlist.

If the device is operating WITHOUT a heat sink, use pdmax_ja for the power dissipation rating and rth_ja for the thermal resistance (rth_ja must be specified if you want the device temperature rise calculated).

If the device is operating WITH a heat sink, use pdmax_jc for the power dissipation rating and a combination of rth_jc and rth_hs for the thermal resistance.

Other Templates used with this one

The arguments of the following templates use values passed in from **xfr** (see the Usage Notes).

```
l.lp pp pm = l=lpri, r=rp, include_stress=0
l.ls sp sm = l=lsec, r=rs, include_stress=0
ml.ml1 i(l.lp) i(l.ls) = m
```

xfr Netlist Examples

This example uses the electrical arguments lp and ls to directly specify the inductance of the transformer and specifies a value for the external parameter temp.

```
xfr.t1 g1 0 h3 h2 = lp=3e-2, ls=2e-3, k=0.98, rp=1m, rs=1m,
  temp=58
```

This example uses the magnetic arguments np, ns, len, area, and ur to specify the winding and core characteristics of the transformer.

```
xfr.t2 p1 p2 s1 s2 = np=60, ns=4, len=3e-2, area=6e-5,
  ur=1, k=0.98, rp=1m, rs=1m
```

xfrei (Nonlinear 2-Winding Transformer with EI Core)

Associated Symbols: xfrei

License Requirements: None

Part Category: Magnetic Templates

Related Topics: Introduction to Magnetic Templates
core — Linear magnetic core
corenl — Nonlinear magnetic core
wind — Winding

Functional Description

The **xfrei** template models a 2-winding transformer with a nonlinear core in an “EI” configuration. It uses three other library templates (**wind**, **corenl**, **core**) as “building blocks” to implement the model. These templates are included in an internal netlist, and their argument values are passed in from the **xfrei** arguments (refer to the Usage Notes).

Template Description Sections

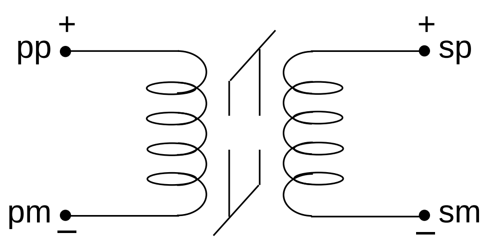
[Usage Notes](#)

[Netlist Examples](#)

[References](#)

xfrei Connection Points

Name	Type	Description
pp	electrical	positive connection of primary winding
pm	electrical	negative connection of primary winding
sp	electrical	positive connection of secondary winding
sm	electrical	negative connection of secondary winding



Nonlinear 2-winding transformer with EI core (xfrei)

xfrei Symbol Properties

Property		
`primitive`	Description:	This symbol calls the template **xfrei**, which models a 2-winding transformer with a nonlinear core in an EI configuration. It uses three other library templates (**wind**, **corenl**, and **core**) as building blocks to implement this model. Because **corenl** is used, you must specify a string value for `matl` OR numerical values for all the model arguments except `ptemp` and `tau`. Specifying a value for `matl` overrides the values for model. Model allows you to specify a group of core characteristics at as many different temperatures (`ptemp`) as desired.
`ref`	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, will be assigned by the schematic capture tool
	Example Input:	Can be any alpha-numeric string
`np`	Description:	Number of winding turns in primary winding (see Usage Notes).
	Default (units):	*value required* (turns)
	Example Input:	10

Property		
rp	Description:	Winding resistance of primary winding (see Usage Notes).
	Default (units):	*value required* (Ω)
	Example Input:	1m
ns	Description:	Number of winding turns in secondary winding (see Usage Notes).
	Default (units):	*value required* (turns)
	Example Input:	10
rs	Description:	Winding resistance of secondary winding (see Usage Notes).
	Default (units):	*value required* (Ω)
	Example Input:	1m
len_fe1	Description:	1st magnetic path length (nonlinear core) (see Usage Notes).
	Default (units):	*value required* (m)
	Example Input:	3.5e-2

Property		
`len_fe2`	Description:	2nd magnetic path length (nonlinear core) (see Usage Notes).
	Default (units):	*value required* (m)
	Example Input:	3.5e-2
`len_fe3`	Description:	3rd magnetic path length (nonlinear core) (see Usage Notes).
	Default (units):	*value required* (m)
	Example Input:	2.0e-2
`area`	Description:	Cross-sectional area of magnetic path (see Usage Notes).
	Default (units):	*value required* (m^2)
	Example Input:	5.9e-5
`len_air`	Description:	Airgap length (linear magnetic path) (see Usage Notes).
	Default (units):	0 (m)
	Example Input:	1m

Property		
matl	Description:	Core material name (from library). This overrides the saber_model property (see Usage Notes).
	Default (units):	undef (—)
	Example Input:	“3c8” (Note that the quotation marks are required.)
sf	Description:	Stacking factor adjustment to area (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	1
saber_model	Description:	saber_model automatically maps to the template argument model, which is a grouping of ten B vs H model arguments (see Usage Notes).
	Example Input:	[(ptemp=27, ui=2700, hc=0.2, uhc=6000, br=1000, hsat=2.5, bsat=4500, hmax=15, bmax=4800, tau=0.6u,taulim=50n), (ptemp=100,ui=2100, hc=0.2, uhc=4300, br=900, hsat=2.0, bsat=3300, hmax=15, bmax=3600, tau=0.6u,taulim=50n)]
ja_model	Description:	Jiles-Atherton model arguments, used only if matl and saber_model are undefined (see Usage Notes).
	Default (units):	undef (—)

Property		
units	Description:	System of units for nonlinear magnetic arguments (see Usage Notes).
	Default:	gauss
	Values:	gauss or si
	Example Input:	si
b0	Description:	Initial flux density (B) in addition to anhysteretic. Units depend on units selection (see Usage Notes).
	Default (units):	0 (G if units=gauss)
	Example Input:	0
tempc	Description:	Core model temperature (see Usage Notes).
	Default (units):	undef (°C)
	Example Input:	25
temp	Description:	Ambient temperature for temperature effects (see External Parameters).
	Default (units):	27 (°C)
	Example Input:	32

Property		
ratings	Description:	Structure of maximum ratings for this device.
	Default:	Available stress ratings for this model are: pdmax_ja, pdmax_jc, tjmax, tjmin, vpmax, vsmax, ipmax, ismax. All values are set to undef by default (see Usage Notes).
	Example Input:	(ipmax=.1,pdmax_ja=.25,tjmax=150)
rth_ja	Description:	Thermal resistance from junction to ambient (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_jc	Description:	Thermal resistance from junction to case (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_hs	Description:	Thermal resistance of an external heat sink (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2

Property		
part_type	Description:	Part type string. Limited to 9 characters.
	Default (units):	xformer (—)
	Example Input:	transfrm
part_class	Description:	Part class string. Limited to 18 characters.
	Default (units):	ei-two winding (—)
	Example Input:	lin two winding
include_ stress	Description:	A flag to allow stress analysis in the netlisted template (see External Parameters).
	Default (units):	1 (—)
	Example Input:	To turn off stress measurement: include_stress=0

xfrei Model Arguments

Name	Default	Units/Values	Description
ja_model [*]	[( )]	—	grouping of Jiles-Atherton arguments (used only if matl and model are undefined)

xfrei Stress Arguments

Stress arguments are intended for use with the `stress` analysis, which is a part of the InSpecs Stress Analysis Option.

Name	Default	Units/Values	Description
`ipmax`	`undef`	A	maximum primary current
`ismax`	`undef`	A	maximum secondary current
`vpmax`	`undef`	V	maximum primary voltage
`vsmax`	`undef`	V	maximum secondary voltage
`tjmax`	`undef`	°C	maximum internal temperature
`tjmin`	`undef`	°C	minimum internal temperature
`pdmax_ja`	`undef`	W	maximum power dissipation with `rth_ja`
`pdmax_jc`	`undef`	W	maximum power dissipation with `rth_jc` and `rth_hs`

xfrei Post Processing Information

The variables in the following table are available for post-processing. You can specify them in a signal list or as arguments to the `extract` command.

Name	Type	Units	Description
`vp`	`val v`	V	primary voltage
`vs`	`val v`	V	secondary voltage

Name	Type	Units	Description
pd_wp	val p	W	power dissipation in primary winding
pd_ws	val p	W	power dissipation in secondary winding
pd_core1	val p	W	power dissipation in corenl.xfrei1
pd_core2	val p	W	power dissipation in corenl.xfrei2
pd_core3	val p	W	power dissipation in corenl.xfrei3
temp_ case	val tc	°C	case temperature
rth_hs_ tjmax	val rth	°C/W	maximum heat sink thermal resistance
v	group	V	grouping of voltages (vp, vs)
i	group	A	grouping of currents (ip, is)
pwr	group	W	grouping of power (pwrd, pd_wp, pd_ws, pd_core1, pd_core2, pd_core3)
ip	val i	A	primary winding current (see Export Variables)
is	val i	A	secondary winding current (see Export Variables)
pwrd	val p	W	instantaneous power dissipation (see Export Variables)
tempj	val tc	°C	instantaneous junction temperature (see Export Variables)

xfrei Export Variables

`pwrd, tempj, ip, is`

These are post-processing variables that can be referenced at the next higher level of the hierarchy.

xfrei External Parameters

`temp, include_stress`

These are global parameters declared in `header.sin`. You can assign values to them for an instance of this template (such as in a netlist) without affecting their global values in the rest of your design. For example, the following allows you to change the simulation temperature to 58°C for `xfrei.t1` only:

```
xfrei.t1 1 0 3 2 = np=60, ns=4, len_fe1=3.5e-2,
  len_fe2=3.5e-2, len_fe3=2e-2, len_air=1e-3, area=6e-5,
  sf=1, model=[(ui=2700, uhc=6000, bmax=4800, hmax=15,
  bsat=4500, hsat=2.5, br=1000, hc=0.2, ptemp=20, tau=50n),
  (ui=2100, uhc=4300, bmax=3600, hmax=15, bsat=3300,
  hsat=2.0, br=900, hc=0.2, ptemp=100, tau=50n)],
  units=gauss, tempc=40, rp=1m, rs=1m, temp=58
```

xfrei Usage Notes

If the device is operating WITHOUT a heat sink, use `pdmax_ja` for the power dissipation rating and `rth_ja` for the thermal resistance (`rth_ja` must be specified if you want the device temperature rise calculated).

If the device is operating WITH a heat sink, use `pdmax_jc` for the power dissipation rating and a combination of `rth_jc` and `rth_hs` for the thermal resistance.

An internal calculation is performed on the value assigned to len_air to account for the fringing effect that occurs when using an air gap.

This template contains three “building block” templates (**core**, **corenl**, **wind**), which are called by a netlist included within **xfrei**. The building block templates receive values from the following **xfrei** arguments: np, ns, len_fe1, len_fe2, len_fe3, len_air, matl, sf, model, units, tempc, rp, rs, b0, and ja_model. The choices for units are gauss and si, which select the system of magnetic units used for calculations.

You can specify a string value for matl OR numeric values for the model arguments OR numeric values for the ja_model arguments. Specifying a value for matl overrides the values for model, and ja_model is used only if matl and model are undefined.

The material library presently contains characterization data for the following materials (see [1], [2], and [3]). The string name of the material is shown between double quote marks (" ").

Material Name (Ferrite)	String Name
Philips Components 3B7	"3B7"
Philips Components 3C8	"3C8"
Philips Components 3C85	"3C85"
Philips Components 3D3	"3D3"
Philips Components 3E2A	"3E2A"
Philips Components 3F3	"3F3"
Philips Components 4C4	"4C4"
Philips Components 3B9	"3B9"
Philips Components 3C6A	"3C6A"

Material Name (Silicon Iron Lamination)	String Name
Magnetic Metals 3% Silicon Iron, 1 50 EI	"si_fe"

Material Name (Square Permalloy)	String Name
Magnetics Square Permalloy 80, 50 038 5 d	"sq_perm_80"

xfrei Netlist Examples

This example uses `model` to characterize the transformer at two different temperatures (i.e., two values of `ptemp`), and specifies a value for the external parameter `temp`.

```
frei.t1 1 0 3 2 = np=60, ns=4, len_fe1=3.5e-2,
  len_fe2=3.5e-2, len_fe3=2e-2, len_air=1e-3, area=6e-5,
  sf=1, model=[(ui=2700, uhc=6000, bmax=4800, hmax=15,
  bsat=4500, hsat=2.5, br=1000, hc=0.2, ptemp=20, tau=50n),
  (ui=2100, uhc=4300, bmax=3600, hmax=15, bsat=3300,
  hsat=2.0, br=900, hc=0.2, ptemp=100, tau=50n)],
  units=gauss, tempc=40, rp=1m, rs=1m, temp=58
```

xfrei References

1. *Soft Ferrite Cores Short Form Catalog*, Philips Components, Magnetic Products Group, Saugerties, NY, 12477.
2. Colonel William T. McLyman, *Magnetic Core Selection for Transformers and Inductors*, Marcel Dekker Inc., 1982.
3. *Tape Wound Cores Design Manua*l, TWC-400, Magnetics, Division of Sprang and Company, 900 E. Butler Road, Butler, PA, 16003.

xfrnl (Nonlinear 2-Winding Transformer)

Associated Symbols: xfrnl

License Requirements: None

Part Category: Magnetic Templates

Related Topics: Introduction to Magnetic Templates
core — Linear magnetic core
corenl — Nonlinear magnetic core
wind — Winding

Functional Description

The **xfrnl** template models a 2-winding transformer with a nonlinear core. It uses three other library templates (**wind**, **core**, **corenl**) as "building blocks" to implement the model. These templates are included in an internal netlist, and their argument values are passed in from the **xfrnl** arguments (refer to the Usage Notes).

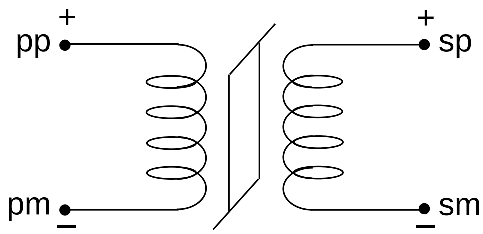
Template Description Sections

Netlist Examples

References

xfrnl Connection Points

Name	Type	Description
pp	electrical	positive connection of primary winding
pm	electrical	negative connection of primary winding
sp	electrical	positive connection of secondary winding
sm	electrical	negative connection of secondary winding



Nonlinear 2-winding transformer (xfrnl)

xfrnl Symbol Properties

Property		
`primitive`	Description:	This symbol calls the template **xfrnl**, which models a 2-winding transformer with a nonlinear core. It uses three other library templates (**wind**, **corenl**, and **core**) as building blocks to implement this model. Because **corenl** is used, you must either specify a string value for `matl` OR numeric values for all the model arguments except `ptemp` and `tau`. Specifying a value for `matl` overrides the values for `saber_model`. `Matl` is a string for a commercially available core material; model allows you to specify a group of core characteristics at as many different temperatures (`ptemp`) as required.
`ref`	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, will be assigned by the schematic capture tool
	Example Input:	Can be any alpha-numeric string
`np`	Description:	Number of winding turns in primary winding
	Default (units):	*value required* (turns)
	Example Input:	60

Property		
rp	Description:	Winding resistance of primary winding (see Usage Notes).
	Default (units):	*value required* (Ω)
	Example Input:	1m
ns	Description:	Number of winding turns in secondary winding (see Usage Notes).
	Default (units):	*value required* (turns)
	Example Input:	4
rs	Description:	Winding resistance of secondary winding (see Usage Notes).
	Default (units):	*value required* (Ω)
	Example Input:	1m
area	Description:	Cross-sectional area of magnetic path (see Usage Notes).
	Default (units):	*value required* (m^2)
	Example Input:	6e-5

Property		
len_fe	Description:	Magnetic path of non-linear core (see Usage Notes).
	Default (units):	*value required* (m)
	Example Input:	3e-2
len_air	Description:	Airgap length (linear magnetic path) (see Usage Notes).
	Default (units):	0 (m)
	Example Input:	1e-3
matl	Description:	Core material name (from library). This overrides the saber_model property (see Usage Notes).
	Default (units):	undef (—)
	Example Input:	"3c8" (Note that the quotation marks are required.)
sf	Description:	Stacking factor adjustment to area (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	1

Property		
saber_model	Description:	saber_model automatically maps to the template argument model, which is a grouping of ten B vs H model properties (see Usage Notes).
	Example Input:	[(ptemp=27, ui=2700, hc=0.2, uhc=6000, br=1000, hsat=2.5, bsat=4500, hmax=15, bmax=4800, tau=0.6u,taulim=50n), (ptemp=100,ui=2100, hc=0.2, uhc=4300, br=900, hsat=2.0, bsat=3300, hmax=15, bmax=3600, tau=0.6u,taulim=50n)]
ja_model	Description:	Jiles-Atherton model arguments, used only if matl and saber_model are undefined (see Usage Notes).
	Default:	undef
units	Description:	System of units for nonlinear magnetic arguments (see Usage Notes).
	Default:	gauss
	Values:	gauss or si
	Example Input:	si
b0	Description:	Initial flux density (B) in addition to anhysteretic. Units depend on units selection (see Usage Notes).
	Default (units):	0 (G if units=gauss)
	Example Input:	0

Property		
`tempc`	Description:	Core model temperature (see Usage Notes).
	Default (units):	undef (°C)
	Example Input:	25
`temp`	Description:	Ambient temperature for temperature effects (see External Parameters).
	Default (units):	27 (°C)
	Example Input:	32
`ratings`	Description:	Structure of maximum ratings for this device.
	Default:	Available stress ratings for this model are: `pdmax_ja`, `pdmax_jc`, `tjmax`, `tjmin`, `vpmax`, `vsmax`, `ipmax`, `ismax`. All values are set to undef by default (see Usage Notes).
	Example Input:	(ipmax=.1,pdmax_ja=.25,tjmax=150)
`rth_ja`	Description:	Thermal resistance from junction to ambient (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2

Property		
rth_jc	Description:	Thermal resistance from junction to case (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_hs	Description:	Thermal resistance of an external heat sink (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
part_type	Description:	Part type string. Limited to 9 characters.
	Default (units):	xformer (—)
	Example Input:	transfrm
part_class	Description:	Part class string. Limited to 18 characters.
	Default (units):	nonlin two winding (—)
	Example Input:	two winding spec

Property		
include_ stress	Description:	A flag to allow stress analysis in the netlisted template (see External Parameters).
	Default (units):	1 (—)
	Example Input:	To turn off stress measurement: include_stress=0

xfrnl Model Arguments

Name	Default	Units/ Values	Description
ja_model[*]	[( )]	—	grouping of Jiles-Atherton arguments (used only if matl and model are undefined)

xfrnl Stress Arguments

Stress arguments are intended for use with the stress analysis, which is a part of the InSpecs Stress Analysis Option.

Name	Default	Units/ Values	Description
ipmax	undef	A	maximum primary current
ismax	undef	A	maximum secondary current
vpmax	undef	V	maximum primary voltage
vsmax	undef	V	maximum secondary voltage
tjmax	undef	°C	maximum internal temperature
tjmin	undef	°C	minimum internal temperature

Name	Default	Units/ Values	Description
pdmax_ja	undef	W	maximum power dissipation with rth_ja
pdmax_jc	undef	W	maximum power dissipation with rth_jc and rth_hs

xfrnl Post Processing Information

The variables in the following table are available for post-processing. You can specify them in a signal list or as arguments to the extract command.

Name	Type	Units	Description
vp	val v	V	primary voltage
vs	val v	V	secondary voltage
pd_wp	val p	W	power dissipation in primary winding
pd_ws	val p	W	power dissipation in secondary winding
pd_core	val p	W	power dissipation in core
temp_ case	val tc	°C	case temperature
rth_hs_ tjmax	val rth	°C/W	maximum heat sink thermal resistance
v	group	V	grouping of voltages (vp, vs)
i	group	A	grouping of currents (ip, is)
pwr	group	W	grouping of power (pwrd, pd_wp, pd_ws)

Name	Type	Units	Description
ip	val i	A	primary winding current (see Export Variables)
is	val i	A	secondary winding current (see Export Variables)
pwrd	val p	W	instantaneous power dissipation (see Export Variables)
tempj	val tc	°C	instantaneous junction temperature (see Export Variables)

xfrnl Export Variables

`pwrd, tempj, ip, is`

These are post-processing variables that can be referenced at the next higher level of the hierarchy.

xfrnl External Parameters

`temp, include_stress`

These are global parameters declared in `header.sin`. You can assign values to them for an instance of this template (such as in a netlist) without affecting their global values in the rest of your design. For example, the following allows you to change the simulation temperature to 58°C for `xfrnl.t1` only:

```
xfrnl.t1 1 0 3 2 = np=60, ns=4, len_fe=3e-2, len_air=1e-3,
  area=6e-5, sf=1, model=[(ui=2700, uhc=6000, bmax=4800,
  hmax=15, bsat=4500, hsat=2.5, br=1000, hc=0.2, ptemp=20,
  tau=50n), (ui=2100, uhc=4300, bmax=3600, hmax=15,
  bsat=3300, hsat=2.0, br=900, hc=0.2, ptemp=100,
  tau=50n)], units=gauss, tempc=25, rp=1m, rs=1m, temp=58
```

xfrnl Usage Notes

If the device is operating WITHOUT a heat sink, use pdmax_ja for the power dissipation rating and rth_ja for the thermal resistance (rth_ja must be specified if you want the device temperature rise calculated). If the device is operating WITH a heat sink, use pdmax_jc for the power dissipation rating and a combination of rth_jc and rth_hs for the thermal resistance.

An internal calculation is performed on the value assigned to len_air to account for the fringing effect that occurs when using an air gap.

This template contains three "building block" templates (**core**, **corenl**, **wind**), which are called by a netlist included within **xfrnl**. The building block templates receive values from the following **xfrnl** arguments: np, ns, len_fe, len_air, matl, sf, model, units, tempc, rp, rs, b0, and ja_model. The choices for units are gauss and si, which select the system of magnetic units used for calculations.

You can specify a string value for matl OR numeric values for the model arguments OR numeric values for the ja_model arguments. Specifying a value for matl overrides the values for model, and ja_model is used only if matl and model are undefined.

The material library presently contains characterization data for the following materials (see [1], [2], and [3]). The string name of the material is shown between double quote marks (" ").

Material Name (Ferrite)	String Name
Philips Components 3B7	"3B7"
Philips Components 3C8	"3C8"
Philips Components 3C85	"3C85"
Philips Components 3D3	"3D3"
Philips Components 3E2A	"3E2A"
Philips Components 3F3	"3F3"

Material Name (Ferrite)	String Name
Philips Components 4C4	"4C4"
Philips Components 3B9	"3B9"
Philips Components 3C6A	"3C6A"

Material Name (Silicon Iron Lamination)	String Name
Magnetic Metals 3% Silicon Iron, 1 50 EI	"si_fe"

Material Name (Square Permalloy)	String Name
Magnetics Square Permalloy 80, 50 038 5 d	"sq_perm_80"

xfrnl Netlist Examples

This example uses `model` to specify characteristics at two different temperatures (i.e., two values of `ptemp`), and specifies a value for the external parameter `temp`.

```
xfrnl.t1 1 0 3 2 = np=60, ns=4, len_fe=3e-2, len_air=1e-3,
  area=6e-5, sf=1, model=[(ui=2700, uhc=6000, bmax=4800,
  hmax=15, bsat=4500, hsat=2.5, br=1000, hc=0.2, ptemp=20,
  tau=50n), (ui=2100, uhc=4300, bmax=3600, hmax=15,
  bsat=3300, hsat=2.0, br=900, hc=0.2, ptemp=100,
  tau=50n)], units=gauss, tempc=25, rp=1m, rs=1m, temp=58
```

xfrnl References

1. *Soft Ferrite Cores Short Form Catalog*, Philips Components, Magnetic Products Group, Saugerties, NY, 12477.
2. Colonel William T. McLyman, *Magnetic Core Selection for*

Transformers and Inductors, Marcel Dekker Inc., 1982.

3. *Tape Wound Cores Design Manual*, TWC-400, Magnetics, Division of Sprang and Company, 900 E. Butler Road, Butler, PA, 16003.

xfr3 (Linear 3-Winding Transformer)

Associated Symbols: xfr3

License Requirements: None

Part Category: Magnetic Templates

Related Topics: Introduction to Magnetic Templates
l — Linear inductor
ml — Mutual inductance

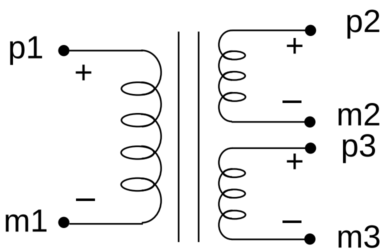
Functional Description

The **xfr3** template models a 3-winding transformer template that allows you to specify electrical arguments (l1, l2, and l3) or magnetic arguments (n1, n2, n3, len, area, and ur). Specifying electrical arguments overrides the magnetic arguments (refer to the Usage Notes).

Template Description Sections

xfr3 Connection Points

Name	Type	Description
p1	electrical	positive connection of first winding
m1	electrical	negative connection of first winding
p2	electrical	positive connection of second winding
m2	electrical	negative connection of second winding
p3	electrical	positive connection of third winding
m3	electrical	negative connection of third winding



Linear 3-winding transformer (xfr3)

xfr3 Symbol Properties

Property		
`primitive`	Description:	This symbol calls the template **xfr3**, which models a 3-winding transformer that allows you to specify electrical arguments (`l1`, `l2`, `l3`) or magnetic arguments (`n1`, `n2`, `n3`, `len`, `area`, `ur`). Specifying electrical properties overrides the magnetic property values.
`ref`	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, will be assigned by the schematic capture tool
	Example Input:	Can be any alpha-numeric string
`l2`	Description:	Inductance of second winding (see Usage Notes).
	Default (units):	undef (H)
	Example Input:	4e-3
`r2`	Description:	Winding resistance of second winding (see Usage Notes).
	Default (units):	0 (Ω)
	Example Input:	1m

Property		
n2	Description:	Number of winding turns in second winding (see Usage Notes).
	Default (units):	undef (turns)
	Example Input:	8
l3	Description:	Inductance of third winding (see Usage Notes).
	Default (units):	undef (H)
	Example Input:	2e-3
n3	Description:	Number of winding turns in third winding (see Usage Notes).
	Default (units):	undef (turns)
	Example Input:	4
r3	Description:	Winding resistance of third winding (see Usage Notes).
	Default (units):	0 (Ω)
	Example Input:	1m

Property		
l1	Description:	Inductance of first winding (see Usage Notes).
	Default (units):	undef (H)
	Example Input:	3e-2
n1	Description:	Number of winding turns in first winding (see Usage Notes).
	Default (units):	undef (turns)
	Example Input:	60
r1	Description:	Winding resistance of first winding (see Usage Notes).
	Default (units):	0 (Ω)
	Example Input:	1m
len	Description:	Magnetic path length (see Usage Notes).
	Default (units):	undef (m)
	Example Input:	3e-2

Property		
area	Description:	Cross-sectional area of magnetic path (see Usage Notes).
	Default (units):	undef (m^2)
	Example Input:	6e-5
ur	Description:	Relative permeability (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	1
k12	Description:	Coupling coefficient between first and second windings (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	0.98
k13	Description:	Coupling coefficient between first and third windings (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	0.98

Property		
k23	Description:	Coupling coefficient between second and third windings (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	0.98
temp	Description:	Ambient temperature for temperature effects (see External Parameters).
	Default (units):	27 (°C)
	Example Input:	32
ratings	Description:	Structure of maximum ratings for this device.
	Default:	Available stress ratings for this model are: pdmax_ja, pdmax_jc, tjmax, tjmin, v1max, v2max, v3max, i1max, i2max, i3max. All values are set to undef by default (see Usage Notes).
	Example Input:	(i1max=.1,pdmax_ja=.25,tjmax=150)
rth_ja	Description:	Thermal resistance from junction to ambient (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2

Property		
rth_jc	Description:	Thermal resistance from junction to case (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_hs	Description:	Thermal resistance of an external heat sink (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
part_type	Description:	Part type string. Limited to 9 characters.
	Default (units):	xformer (—)
	Example Input:	pwr xfrm
part_class	Description:	Part class string. Limited to 18 characters.
	Default (units):	three winding (—)
	Example Input:	lin three winding

Property		
include_ stress	Description:	A flag to allow stress analysis in the netlisted template (see External Parameters).
	Default (units):	1 (—)
	Example Input:	To turn off stress measurement: include_stress=0

xfr3 Stress Arguments

Stress arguments are intended for use with the stress analysis, which is a part of the InSpecs Stress Analysis Option.

Name	Default	Units/ Values	Description
pdmax_ja	undef	W	maximum power dissipation with rth_ja
pdmax_jc	undef	W	maximum power dissipation with rth_jc and rth_hs
tjmax	undef	°C	maximum internal temperature
tjmin	undef	°C	minimum internal temperature
v1max	undef	V	maximum first winding voltage
v2max	undef	V	maximum second winding voltage
v3max	undef	V	maximum third winding voltage
i1max	undef	A	maximum first winding current
i2max	undef	A	maximum second winding current
i3max	undef	A	maximum third winding current

xfr3 Post Processing Information

Name	Type	Units	Description
v1	val v	V	first winding voltage
v2	val v	V	second winding voltage
v3	val v	V	third winding voltage
temp_case	val tc	°C	case temperature
rth_hs_tjmax	val rth	°C/W	maximum heat sink thermal resistance
pd_1	val p	W	first winding power dissipation
pd_2	val p	W	second winding power dissipation
pd_3	val p	W	third winding power dissipation
i1	val i	A	first winding current (see Export Variables)
i2	val i	A	second winding current (see Export Variables)
i3	val i	A	third winding current (see Export Variables)
pwrd	val p	W	instantaneous power dissipation (see Export Variables)
tempj	val tc	°C	instantaneous junction temperature (see Export Variables)
pwr	group	W	grouping of power (pwrd, pd_1, pd_2, pd_3)

xfr3 Export Variables

`pwrd, tempj, i1, i2, i3`

These are post-processing variables that can be referenced at the next higher level of the hierarchy.

xfr3 External Parameters

`temp, include_stress`

These are global parameters declared in `header.sin`. You can assign values to them for an instance of this template (such as in a netlist) without affecting their global values in the rest of your design. For example, the following allows you to change the simulation temperature to 58°C for `xfr3.t1` only:

```
xfr3.t1 g1 0 h5 h4 j3 j2 = l1=3e-2, l2=4e-3, l3=2e-3,
  k12=0.98, k13=0.98, k23=0.98, r1=0.1, r2=0.1, r3=0.1,
  temp=58
```

xfr3 Usage Notes

The linear 3-winding transformer template (**xfr3**) allows you to specify electrical arguments or magnetic arguments:

Electrical	`l1, l2, l3`
Magnetic	`n1, n2, n3, len, area, ur`

Specifying values for winding inductances (`l1`, `l2`, `l3`) overrides values for the magnetic arguments. The values for `k12`, `k13`, `k23`, `r1`, `r2`, and `r3` are used with both groups.

The effective coupling coefficient is defined as:

$$k = \begin{vmatrix} 1 & k12 & k13 \\ k12 & 1 & k23 \\ k13 & k23 & 1 \end{vmatrix}$$

The magnitude of the effective coupling coefficient `k` must be in the range $-1 \le k \le +1$, which results in the constraint:

$$1 + 2 \cdot k12 \cdot k13 \cdot k23 - (k12^2 + k13^2 + k23^2) \quad 0$$

This template uses **l** and **ml** as “building block” templates, calling them from a netlist that is part of the **xfr3** template. If `l1`, `l2`, and `l3` are specified, these values are passed directly to the building block inductors (`l1`, `l2`, `l3`). If the magnetic arguments are used instead, **xfr3** calculates the inductances internally (`l1calc`, `l2calc`, `l3calc`) and then passes them to `l1`, `l2`, and `l3` in the netlist.

If the device is operating WITHOUT a heat sink, use `pdmax_ja` for the power dissipation rating and `rth_ja` for the thermal resistance (`rth_ja` must be specified if you want the device temperature rise calculated).

If the device is operating WITH a heat sink, use `pdmax_jc` for the power dissipation rating and a combination of `rth_jc` and `rth_hs` for the thermal resistance.

Other Templates used with this one

The following template arguments use values passed in from **xfr3** (see Usage Notes).

```
l.l1 p1 m1 = l=l1calc, r=r1, include_stress=0
l.l2 p2 m2 = l=l2calc, r=r2, include_stress=0
l.l3 p3 m3 = l=l3calc, r=r3, include_stress=0
ml.ml12 i(l.l1) i(l.l2) = m12
ml.ml13 i(l.l1) i(l.l3) = m13
ml.ml23 i(l.l2) i(l.l3) = m23
```

xfr3 Netlist Examples

This example uses the electrical arguments `l1`, `l2`, and `l3` to directly specify the inductance of the transformer and specifies a value for the external parameter `temp`.

```
xfr3.t1 g1 0 h5 h4 j3 j2 = l1=3e-2, l2=4e-3, l3=2e-3,
  k12=0.98, k13=0.98, k23=0.98, r1=0.1, r2=0.1, r3=0.1,
  temp=58
```

This example uses the magnetic arguments `n1`, `n2`, `n3`, `len`, `area`, and `ur` to specify the winding and core characteristics of the transformer.

```
xfr3.t2 6 0 10 9 8 7 = n1=60, n2=8, n3=4, len=3e-2,
  area=6e-5, ur=1, k12=0.98, k13=0.98, k23=0.98, r1=1m,
  r2=1m, r3=1m
```

xfr3nl (Nonlinear 3-Winding Transformer)

Associated Symbols:	`xfr3nl`
License Requirements:	None
Part Category:	Magnetic Templates
Related Topics:	Introduction to Magnetic Templates **core** — Linear magnetic core **corenl** — Nonlinear magnetic core **wind** — Winding

Functional Description

The **xfr3nl** template models a 3-winding transformer with a nonlinear core. It uses three other library templates (**wind**, **core**, **corenl**) as "building blocks" to implement the model. These templates are included in an internal netlist, and their argument values are passed in from the **xfr3nl** arguments (refer to the Usage Notes).

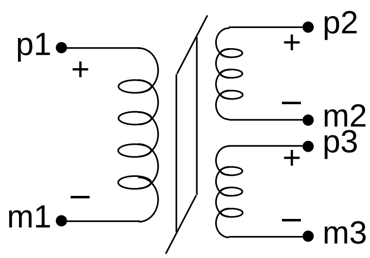
Template Description Sections

Netlist Examples

References

xfr3nl Connection Points

Name	Type	Description
p1	electrical	positive connection of first winding
m1	electrical	negative connection of first winding
p2	electrical	positive connection of second winding
m2	electrical	negative connection of second winding
p3	electrical	positive connection of third winding
m3	electrical	negative connection of third winding



Nonlinear 3-winding transformer (xfr3nl)

xfr3nl Symbol Properties

Property		
`primitive`	Description:	This symbol calls the template **xfr3nl**, which models a 3-winding transformer with a core. It uses three other library templates (**wind**, **corenl**, **core**) as building blocks to implement this model. Because **corenl** is used, you must specify a string value for `matl` OR numeric values for all the model arguments except `ptemp` and `tau`. Specifying a value for `matl` override the values for model. `Matl` is a string for a commercially available core material; model allows you to specify a group of core characteristics at as many different temperatures (`ptemp`) as desired.
`ref`	Description:	Suffix appended to a template name that uniquely identifies a part in a schematic.
	Default:	If not specified, will be assigned by the schematic capture tool
	Example Input:	Can be any alpha-numeric string
`n2`	Description:	Number of winding turns in second winding (see Usage Notes).
	Default (units):	*value required* (turns)
	Example Input:	10

Property		
`r2`	Description:	Winding resistance of second winding (see Usage Notes).
	Default (units):	*value required* (Ω)
	Example Input:	1m
`n3`	Description:	Number of winding turns in third winding (see Usage Notes).
	Default (units):	*value required* (turns)
	Example Input:	10
`r3`	Description:	Winding resistance of third winding (see Usage Notes).
	Default (units):	*value required* (Ω)
	Example Input:	1m
`n1`	Description:	Number of winding turns in first winding (see Usage Notes).
	Default (units):	*value required* (turns)
	Example Input:	60

Property		
`r1`	Description:	Winding resistance of first winding (see Usage Notes).
	Default (units):	*value required* (Ω)
	Example Input:	1m
`area`	Description:	Cross-sectional area of magnetic path (see Usage Notes).
	Default (units):	*value required* (m^2)
	Example Input:	5.9e-5
`len_fe`	Description:	Magnetic path of non-linear core (see Usage Notes).
	Default (units):	*value required* (m)
	Example Input:	2e-2
`len_air`	Description:	Airgap length (linear magnetic path) (see Usage Notes).
	Default (units):	0 (m)
	Example Input:	1m

Property		
matl	Description:	Core material name (from library). This overrides the saber_model property (see Usage Notes).
	Default (units):	undef (—)
	Example Input:	"3c8" (Note that the quotation marks are required.)
sf	Description:	Stacking factor adjustment to area (see Usage Notes).
	Default (units):	1 (—)
	Example Input:	1
saber_model	Description:	saber_model automatically maps to the template argument model, which is a grouping of ten B vs H model arguments (see Usage Notes).
	Example Input:	[(ptemp=27, ui=2700, hc=0.2, uhc=6000, br=1000, hsat=2.5, bsat=4500, hmax=15, bmax=4800, tau=0.6u,taulim=50n), (ptemp=100,ui=2100, hc=0.2, uhc=4300, br=900, hsat=2.0, bsat=3300, hmax=15, bmax=3600, tau=0.6u,taulim=50n)]
ja_model	Description:	Jiles-Atherton model arguments, used only if matl and saber_model are undefined (see Usage Notes).
	Default:	undef

Property		
units	Description:	System of units for nonlinear magnetic arguments (see Usage Notes).
	Default:	gauss
	Values:	gauss or si
	Example Input:	si
b0	Description:	Initial flux density (B) in addition to anhysteretic. Units depend on units selection (see Usage Notes).
	Default (units):	0 (G if units=gauss)
	Example Input:	0
tempc	Description:	Core model temperature (see Usage Notes).
	Default (units):	undef (°C)
	Example Input:	25
temp	Description:	Ambient temperature for temperature effects (see External Parameters).
	Default (units):	27 (°C)
	Example Input:	32

Property		
ratings	Description:	Structure of maximum ratings for this device.
	Default:	Available stress ratings for this model are: pdmax_ja, pdmax_jc, tjmax, tjmin, v1max, v2max, v3max, v4max, i1max, i2max, i3max, i4max. All values are set to undef by default (see Usage Notes).
	Example Input:	(i1max=.1,pdmax_ja=.25,tjmax=150)
rth_ja	Description:	Thermal resistance from junction to ambient (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_jc	Description:	Thermal resistance from junction to case (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2
rth_hs	Description:	Thermal resistance of an external heat sink (see Usage Notes).
	Default (units):	undef (°C/W)
	Example Input:	.2

Property		
part_type	Description:	Part type string. Limited to 9 characters.
	Default (units):	xformer (—)
	Example Input:	transform
part_class	Description:	Part class string. Limited to 18 characters.
	Default (units):	nonlin three wind (—)
	Example Input:	three winding
include_stress	Description:	A flag to allow stress analysis in the netlisted template (see External Parameters).
	Default (units):	1 (—)
	Example Input:	To turn off stress measurement: include_stress=0

xfr3nl Model Arguments

Name	Default	Units/Values	Description
units	gauss	gauss,si	system of units for nonlinear magnetic arguments
tempc	undef	°C	core model temperature (temp can be used instead: see External Parameters)
r1	*value required*	Ω	first winding resistance
r2	*value required*	Ω	second winding resistance
r3	*value required*	Ω	third winding resistance
b0	0	T *or* G	initial flux density (B) in addition to anhysteretic (units depend on units selection)
ja_model [*]	[( )]	—	grouping of Jiles-Atherton arguments (used only if matl and model are undefined)

xfr3nl Stress Arguments

Stress arguments are intended for use with the stress analysis, which is a part of the InSpecs Stress Analysis Option.

Name	Default	Units/Values	Description
i1max	undef	A	maximum first winding current

Name	Default	Units/Values	Description
i2max	undef	A	maximum second winding current
i3max	undef	A	maximum third winding current
v1max	undef	V	maximum first winding voltage
v2max	undef	V	maximum second winding voltage
v3max	undef	V	maximum third winding voltage
tjmax	undef	°C	maximum internal temperature
tjmin	undef	°C	minimum internal temperature
pdmax_ja	undef	W	maximum power dissipation with rth_ja
pdmax_jc	undef	W	maximum power dissipation with rth_jc and rth_hs

xfr3nl Post Processing Information

The variables in the following table are available for post-processing. You can specify them in a signal list or as arguments to the `extract` command.

Name	Type	Units	Description
v1	val v	V	first voltage
v2	val v	V	second voltage
v3	val v	V	third voltage

Name	Type	Units	Description
absv1	val v	V	absolute value of first voltage
absv2	val v	V	absolute value of second voltage
absv3	val v	V	absolute value of third voltage
pd_w1	val p	W	power dissipation in first winding
pd_w2	val p	W	power dissipation in second winding
pd_w3	val p	W	power dissipation in third winding
pd_core	val p	W	power dissipation in core
temp_case	val tc	°C	case temperature
rth_hs_tjmax	val rth	°C/W	maximum heat sink thermal resistance
v	group	V	grouping of voltages (v1, v2, v3)
i	group	A	grouping of currents (i1, i2, i3)
pwr	group	W	grouping of power (pwrd, pd_w1, pd_w2, pd_w3, pd_core)
i1	val i	A	first winding current (see Export Variables)
i2	val i	A	second winding current (see Export Variables)
i3	val i	A	third winding current (see Export Variables)

Name	Type	Units	Description
pwrd	val p	W	instantaneous power dissipation (see Export Variables)
tempj	val tc	°C	instantaneous junction temperature (see Export Variables)

xfr3nl Export Variables

`pwrd, tempj, i1, i2, i3`

These are post-processing variables that can be referenced at the next higher level of the hierarchy.

xfr3nl External Parameters

`temp, include_stress`

These are global parameters declared in header.sin. You can assign values to them for an instance of this template (such as in a netlist) without affecting their global values in the rest of your design. For example, the following allows you to change the simulation temperature to 58°C for xfr3nl.t1 only:

```
xfr3nl.t1 g1 0 h5 h4 k3 k2 = n1=60, n2=8, n3=4,
  len_fe=3e-2, len_air=0, area=6e-5, matl="3c8", sf=1,
  units=gauss, tempc=25, r1=1m, r2=1m, r3=1m, temp=58
```

xfr3nl Usage Notes

If the device is operating WITHOUT a heat sink, use pdmax_ja for the power dissipation rating and rth_ja for the thermal resistance (rth_ja must be specified if you want the device temperature rise calculated). If the device is operating WITH a heat sink, use pdmax_jc for the power dissipation rating and a combination of rth_jc and rth_hs for the thermal resistance.

An internal calculation is performed on the value assigned to len_air to account for the fringing effect that occurs when using an air gap.

This template contains three "building block" templates (**core**, **corenl**, **wind**) that are called by a netlist included within **xfr3nl**. The building block templates receive values from the following **xfr3nl** arguments: n1, n2, n3, len_fe, len_air, matl, sf, model, units, tempc, r1, r2, r3, b0, and ja_model. The choices for units are gauss and si, which select the system of magnetic units used for calculations.

You can specify a string value for matl OR numeric values for the model arguments OR numeric values for the ja_model arguments. Specifying a value for matl overrides the values for model, and ja_model is used only if matl and model are undefined.

The material library presently contains characterization data for the following materials (see [1], [2], and [3]). The string name of the material is shown between double quote marks (" ").

Material Name (Ferrite)	String Name
Philips Components 3B7	"3B7"
Philips Components 3C8	"3C8"
Philips Components 3C85	"3C85"
Philips Components 3D3	"3D3"
Philips Components 3E2A	"3E2A"
Philips Components 3F3	"3F3"
Philips Components 4C4	"4C4"
Philips Components 3B9	"3B9"
Philips Components 3C6A	"3C6A"

Material Name (Silicon Iron Lamination)	String Name
Magnetic Metals 3% Silicon Iron, 1 50 EI	"si_fe"

Material Name (Square Permalloy)	String Name
Magnetics Square Permalloy 80, 50 038 5 d	"sq_perm_80"

xfr3nl Netlist Examples

This example uses `matl` to specify a predefined, commercially-available core material, and specifies a value for the external parameter `temp`.

```
xfr3nl.t1 g1 0 h5 h4 k3 k2 = n1=60, n2=8, n3=4,
  len_fe=3e-2, len_air=0, area=6e-5, matl="3c8", sf=1,
  units=gauss, tempc=25, r1=1m, r2=1m, r3=1m, temp=58
```

xfr3nl References

1. *Soft Ferrite Cores Short Form Catalog*, Philips Components, Magnetic Products Group, Saugerties, NY, 12477.
2. Colonel William T. McLyman, *Magnetic Core Selection for Transformers and Inductors*, Marcel Dekker Inc., 1982.
3. *Tape Wound Cores Design Manual*, TWC-400, Magnetics, Division of Sprang and Company, 900 E. Butler Road, Butler, PA, 16003.