Chapter 12

Using Passive Devices

This chapter describes element and model statements for passive devices. It includes statements for resistors, inductors, capacitors, and assorted magnetic elements and models.

Resistors, inductors, and capacitors are of two types:

- A simple, linear element with a value that depends on temperature, initialization, and scaling.
- An element that refers to a model statement

Use the set of passive elements and model statements to construct a wide range of board and integrated circuit level designs. Passive elements let you include transformers, PC board trace interconnects, coaxial cables and transmission lines in an analysis. The wire element model is specifically designed to model the RC delay and RC transmission line effects of interconnects at both the IC level and the PC board level.

To aid in designing power supplies, a mutual-inductor model includes switching regulators and a number of other magnetic circuits, including a magnetic-core model and element. You can specify precision modeling of passive elements using geometric, temperature, and parasitic model parameters.

This chapter covers the following topics:

- Using the Element Statement
- Using the Resistor Element
- Using the Capacitor Element
- Using the Linear Inductor Element
- Using Magnetics

Using the Element Statement

The element statement specifies the type of element used. It has fields for the element name, the connecting nodes, a component value, and optional parameters.

Syntax

```
NAME node1, node2 …. nodeN <model reference>
value <optional parameters>
```

NAME	Specifies the type and name of element. The first letter in the name field identifies the element as a specific type. For example, C=capacitor, L=inductor, or R=resistor. The remaining letters give the element a unique name.
node1 ... nodeN	Specifies how the element is connected in the netlist
value	Gives the value of the element. For example, C1 2 0 10uF.
<model reference>	Refers to a model when the basic element value does not provide an adequate description

Element Parameters

Element parameters within the element statement describe the device type, device terminal connections, associated model reference, element value, DC initialization voltage or current, element temperature, and parasitics.

The following tables include the complete set of parameters for all the available elements. The netlist field includes the element name (for example, Rxxx or Cxxx), the connecting nodes (for example, n1, n2), and any associated element value parameters (for example, rval in the resistor statement or cval in the capacitor statement).

Table 12-1: R, L, C, and K Element Parameters

	Resistor	Capacitor	Inductor	Mutual Inductor
netlist	Rxxx, n1, n2 mname, rval	Cxxx, n1, n2, mname, cval	Lxxx, n1, n2 mname, lval	Kxxx, Lyyy, Lzzz, kval
temperature	DTEMP, TC1, TC2	DTEMP, TC1, TC2	DTEMP, TC1, TC2	
geometric (scaling)	L, M, W, SCALE	L, M, W, SCALE	M, SCALE	
parasitics	C			
initialization		IC(v)	IC(i)	
AC	AC			

Table 12-2 shows two nonlinear elements. Use nonlinear elements to specify nonlinear inductors and capacitors in a polynomial equation. Specify either a nonlinear or linear element equation with the netlist parameters POLY, C0, and C1.

Table 12-2: Nonlinear L and C Element Parameters

	Nonlinear Inductor	Nonlinear Capacitor
netlist	Lxxx, n1, n2, POLY, C0, C1	Cxxx, n1, n2, POLY, C0, C1
geometric	M	M
initialization	IC(i)	IC(v)

Geometric parameters describe the dimensions of the element.

DTEMP is an element temperature parameter. It represents the difference between the general circuit simulation temperature and the element temperature. Element temperature = circuit temperature + DTEMP.

The magnetic winding element has a secondary resistance parasitic parameter (R) to model secondary parasitics.

The element statement provides initialization parameters for capacitor, inductor, magnetic winding, and transmission line elements to initialize circuits for DC analysis.

Table 12-3: Magnetic and Transmission Line Element Parameters

Function	Magnetic Winding	Mutual Coupling
netlist	Lxxx, n1, n2	Kxxx, Lyyy, mname
geometric	NT	
parasitics	R	
initialization	IC(i)	MAG

Model Statement

Capacitor, resistor, and mutual coupling element statements have associated model capacitor, wire, and magnetic core model statements. The capacitor, wire, and magnetic core model statements use the following syntax:

General form

```
.MODEL mname modeltype <keyword=value>
```

mname	Refers to the model reference name specified in the associated element statement
modeltype	Specifies the model type: R for wire model, C for capacitor model, L for magnetic core model

Model Parameters

The model statement for each element has associated parameters to specify temperature, geometric dimensions, primary and parasitic resistance, and capacitive and magnetic values. You can set both resistance and capacitor values in the wire model, to model interconnect.

Table 12-4: Passive Model Parameters

Function	Wire Model	Capacitor	Saturable Core
Temperature	TC1C, TC1R, TC2C, TC2R, TREF	TC1, TC2, TREF	
Geometric	DLR, DW, L, SHRINK, THICK, W	DEL, L, SHRINK, THICK, W	AC, LC, LG, TC
Resistance	RAC, RES, RSH		
Capacitance	BULK, CAP, CAPSW, COX, DI	CAP, CAPSW, COX, DI	
Magnetic			BS, BR, HC, HCR, HS

Using the Resistor Element

The resistor element uses the following element statement formats:

Format

The general format of the resistor element is:

```
Rxxx n1 n2 <mname> rval <TC1 <TC2>> <SCALE=val>
<M=val> <AC=val>
+ <DTEMP=val> <L=val> <W=val> <C=val>

Rxxx n1 n2 <mname> R=val <TC1=val> <TC2=val >
<SCALE=val> <M=val> <AC=val>
 + <DTEMP=val> <L=val> <W=val> <C=val>
Rxxx n1 n2 R='equation'
```

If you specify mname, the resistor value specification is optional.

AC	AC resistance used in the AC analysis. Default=Reff.	
		$ACeff = AC \cdot SCALE$ (element) / M
C	Capacitance connected from node n2 to bulk. Default=0.0, if C is not specified in the model.	
		$Ceff = C \cdot SCALE$ (element) $\cdot M$
DTEMP	Temperature difference between the element and the circuit. Default=0.0.	
L	Resistor length. Default = 0.0, if L is not specified in the model.	
		$Lscaled = L \cdot SHRINK \cdot SCALE$ (option)
	SHRINK is a model parameter that specifies the fabrication shrink factor.	

M	Multiplier that simulates parallel resistors. For example, to represent two parallel instances of a resistor, set M=2 to multiply the number of resistors by 2. Default=1.0.	
mname	Model name. Use this name in elements to reference the model.	
n1	Positive terminal node name	
n2	Negative terminal node name	
R=equation	Equation that descibes the resistor value as a function of any node voltages, branch currents, and any independent variables such as time, frequency (Hertz), or temperature.	
R=val	Resistance value	
		Reff = R · SCALE (element) / M
Rxxx	Resistor element name. Must begin with “R”, which can be followed by up to 15 alphanumeric characters.	
rval	Resistor value. Can be a value or parameter that can be evaluated.	
SCALE	Element scale factor for resistance and capacitance. Default=1.0.	
TC1	First order temperature coefficient for resistor	
TC2	Second order temperature coefficient for resistor	
W	Resistor width. Default=0.0, if W is not specified in the model. SHRINK is a model parameter.	
		Wscaled = W · SHRINK · SCALE (option)

Examples

```
R1 Rnode1 Rnode2 100
RC1 12 17 1K TC=0.001, 0  1.2
R4 33 0 45 RTC1 RTC2  7
Rxxx  98999999 87654321  1 AC=1e10
```

Resistor Noise Equation

The thermal noise of a resistor is modeled by:

$$inr = \left(NOISE \cdot \frac{4kT}{Rval}\right)^{1/2}$$

where NOISE is a model parameter that defaults to 1. To eliminate the contribution of resistor noise, use the NOISE parameter. To specify the NOISE parameter, use a model for the resistor.

Noise Summary Print out Definitions

RX	Transfer function of thermal noise to the output. This is not noise, but is a transfer coefficient, reflecting the contribution of thermal noise to the output.
TOT, V^2/Hz	Total output noise: $TOT = RX^{2} \cdot inr$

Resistor Temperature Equations

The resistor and capacitor values are modified by temperature values as follows:

$$R(t) = R \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

$$RAC(t) = RAC \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

$$C(t) = C \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

Δt	t - tnom
t	Element temperature in °K: t = circuit temp + DTEMP + 273.15
tnom	Nominal temperature in °K: tnom = 273.15 + TNOM

Wire RC Model

The Hspice wire element RC model is a CRC (pi) model. Use the CRATIO wire model parameter to allocate the parasitic capacitance of the wire element between the model’s input capacitor and output capacitor. This allows for symmetric node impedance for bidirectional circuits such as buses.

Format

The format of the wire element is:

```
.MODEL mname R keyword=value <CRATIO=val>
```

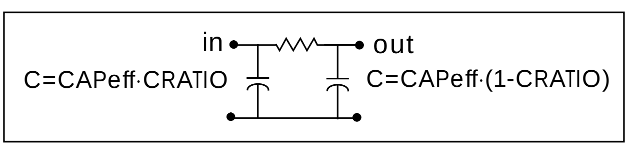
mname	Model name. Elements reference the model with this name.
R	Specifies a wire model
keyword	Any model parameter name

CRATIO	Ratio to allocate the total wire element parasitic capacitance between the capacitor connected to the input node and the capacitor connected to the output node of the wire element pi model. You can assign CRATIO any value between 0 and 1:

0	Assigns all of the parasitic capacitance (CAPeff) to the output node
0.5	Assigns half of the parasitic capacitance to the input node and half to the output node
1	Assigns all of the parasitic capacitance to the input node

The default is 0.5. CRATIO values smaller than 0.5 assign more of the capacitance to the output node than to the input node. Values greater than 0.5 assign more of the capacitance to the input node than to the output node.

If you specify a value outside the range of 0 to 1.0 is specified for CRATIO, Hspice displays a warning, sets CRATIO to 0.5, and continues the analysis.



A resistor referred to as a wire model behaves like an elementary transmission line if you specify an optional capacitor from node n2 to a bulk or ground node in the model statement. The bulk node functions as a ground plane for the wire capacitance.

A wire is described by a drawn length and a drawn width. The resistance of the wire is the effective length multiplied by RSH divided by the effective width.

To avoid syntactic conflicts, if a resistor model exists using the same name as a parameter for rval in the element statement, the model name is taken. In the following example, R1 assumes that REXX refers to the model and not the parameter.

```
.PARAMETER REXX=1
R1 1 2 REXX
.MODEL REXX R RES=1
```

Wire Model Parameters

Name(Alias)	Units	Default	Description
BULK	gnd		default reference node for capacitance
CAP	F	0	default capacitance
CAPSW	F/m	0	sidewall fringing capacitance
COX	F/m^2	0	bottomwall capacitance
DI		0	relative dielectric constant
DLR	m	0	difference between drawn length and actual length (for resistance calculation only). For capacitance calculation, DW is used DLReff=DLR · SCALM
DW	m	0	difference between drawn width and actual width DWeff=DW · SCALM
L	m	0	default length of wire Lscaled=L · SHRINK · SCALM
LEVEL			model selector (not used)
RAC	ohm		default AC resistance (RACeff default is Reff)
RES	ohm	0	default resistance
RSH		0	sheet resistance/square
SHRINK		1	shrink factor
TC1C	1/deg	0	first order temperature coefficient for capacitance
TC2C	$1/deg^2$	0	second order temperature coefficient for capacitance
TC1R	1/deg	0	first order temperature coefficient for resistance
TC2R	$1/deg^2$	0	second order temperature coefficient for resistance
THICK	m	0	dielectric thickness
TREF	deg C	TNOM	temperature reference for model parameters

Name(Alias)	Units	Default	Description
W	m	0	default width of wire Wscaled=W · SHRINK · SCALM

Wire Resistance Calculation

You can specify the wire width and length in both the element and model statements. The element values override the model values. The element width and length are scaled by the option SCALE and the model parameter SHRINK. The model width and length are scaled by the option SCALM and the model parameter SHRINK.

The effective width and length are calculated as follows:

$$Weff = Wscaled - 2 \cdot DWeff$$

$$Leff = Lscaled - 2 \cdot DLReff$$

If element resistance is specified:

$$Reff = \frac{R \cdot SCALE(element)}{M}$$

Otherwise, if $(Weff \cdot Leff \cdot RSH)$ is greater than zero, then:

$$Reff = \frac{Leff \cdot RSH \cdot SCALE(element)}{M \cdot Weff}$$

If $(Weff \cdot Leff \cdot RSH)$ is zero, then:

$$Reff = \frac{RES \cdot SCALE(element)}{M}$$

If AC resistance is specified in the element, then:

$$RACeff = \frac{AC \cdot SCALE(element)}{M}$$

Otherwise, if RAC is specified in the model, RAC is used:

$$RACeff = \frac{RAC \cdot SCALE(element)}{M}$$

If neither are specified, it defaults to:

$$RACeff = Reff$$

If the resistance is less than option RESMIN, it is reset to RESMIN and a warning message is issued.

$$RESMIN = \frac{1}{GMAX \cdot 1000 \cdot M}$$

Wire Capacitance Calculation

The effective length is the scaled drawn length less $2 \cdot$ DLeff. Leff represents the effective length of the resistor from physical edge to physical edge. DWeff is the distance from the drawn edge of the resistor to the physical edge of the resistor. The effective width is the same as the width used in the resistor calculation.

$$Leff = Lscaled - 2 \cdot DLeff$$

$$Weff = Wscaled - 2 \cdot DWeff$$

If the element capacitance C is specified:

$$CAPeff = C \cdot SCALE(element) \cdot M$$

Otherwise, the capacitance is calculated from the Leff, Weff, and COX.

$$CAPeff = M \cdot SCALE(element) \cdot [Leff \cdot Weff \cdot COX + 2 \cdot (Leff + Wef$$

The computation of the bottom wall capacitance COX is based upon a hierarchy of defaults and specified values involving the dielectric thickness THICK, the relative dielectric constant DI, and two absolute dielectric constants εo and εox, as follows:

1. If COX=value is given, that value is used.
2. If COX is not given specifically but THICK (the dielectric thickness) is given and nonzero:

   a. If DI=value is given and nonzero then:

   $$COX = \frac{DI \cdot \varepsilon o}{THICK}$$

   b. If DI is not given, or is zero, then:

   $$COX = \frac{\varepsilon ox}{THICK}$$

   where

   $\varepsilon o = 8.8542149e\text{-}12$ F/meter

   $\varepsilon ox = 3.453148e\text{-}11$ F/meter

   If COX is not given and THICK= 0 is an error.
3. If only the model capacitance CAP is specified, then:
4. $CAPeff = CAP \cdot SCALE(element) \cdot M$

If the capacitance is specified and the bulk node is not specified, then capacitance is not evaluated and a warning message is issued.

Using the Capacitor Element

General form

```
Cxxx n1 n2 <mname> cval <TC1 <TC2>> <SCALE=val>
<IC=val> <M=val> <W=val>
+ <L=val> <DTEMP=val>
```

or

```
Cxxx n1 n2 <mname> C=val <TC1=val> <TC2=val> <IC=val>
<M=val> <W=val> <L=val>
+ <DTEMP=val>
```

or

```
Cxxx n1 n2 C='equation' CTYPE = 0 or 1
```

If a model is chosen for the capacitor, then the specification of CAPVAL is optional.

Cxxx	capacitor element name. Must begin with a “C”, which can be followed by up to 15 alphanumeric characters.
n1	positive terminal node name
n2	negative terminal node name
mname	model name
C	capacitance in farads at room temperature
cval	capacitance value. Can be a value or parameter that can be evaluated.
TC1	first order temperature coefficient

TC2	second order temperature coefficient	
M	multiplier used to simulate multiple parallel devices. Default=1.0.	
W	capacitor width	
		Wscaled = W · SHRINK · SCALE (option)
L	capacitor length	
		Lscaled = L · SHRINK · SCALE (option)
DTEMP	element temperature difference with respect to circuit temperature. Default=0.0.	
C=equation	The capacitor value can be described as a function of any node voltages, branch currents, and any independent variables such as time, frequency (Hertz), or temperature.	

CTYPE	If capacitance C is a function of v(n1, n2), set CTYPE=0. If C is not a function of v(n1, n2), set CTYPE=1. The capacitance charge is calculated differently for the two types. CTYPE must be set properly to provide correct simulation results. C as a function of multiple variables is not recommended. Default=0.
SCALE	element scale factor

$$Ceff = C \cdot SCALE\ (element) \cdot M$$

IV	initial voltage across capacitor in volts. This value is used as the DC operating point voltage.

To avoid syntactic conflicts, if a capacitor model exists using the same name as a parameter for cval in the element statement, the model name is taken. In the following example, C1 assumes that CAPXX refers to the model and not the parameter.

```
.PARAMETER  CAPXX=1
C1  1  2  CAPXX
.MODEL CAPXX C CAP=1
```

Capacitance Model

General form

.MODEL mname C parameter=value

mname	model name
C	specifies a capacitance model
parameter	any model parameter name

Capacitance Parameters

Table 12-5: Capacitance Parameters

Name(Alias)	Units	Default	Description
CAP	F	0	default capacitance value
CAPSW	F/m	0	sidewall fringing capacitance
COX	F/m^2	0	bottomwall capacitance
DEL	m	0	difference between drawn width and actual width or length DELeff = DEL · SCALM
DI		0	relative dielectric constant
L	m	0	default length of capacitor Lscaled = L · SHRINK · SCALM
SHRINK		1	shrink factor
TC1	1/deg	0	first temperature coefficient for capacitance
TC2	$1/deg^2$	0	second temperature coefficient for capacitance
THICK	m	0	insulator thickness
TREF	deg C	TNOM	reference temperature
W	m	0	default width of capacitor Wscaled = W · SHRINK · SCALM

Parameter Limit Checking

HSPICE writes a warning message to the output listing file if a capacitive element value exceeds 0.1 farad. This feature eases identification of elements with missing units or wrong values, particularly those in automatically produced netlists.

Effective Capacitance Calculation

A model can be associated with a capacitor in HSPICE. You can specify some of the parameters in both the element and model descriptions. The element values override the model values. The option SCALE and the model parameter SHRINK scale the element width and length. The option SCALM and the model parameter SHRINK scale the model width and length.

The effective width and length are calculated as follows:

$$Weff = Wscaled - 2 \cdot DELeff$$

$$Leff = Lscaled - 2 \cdot DELeff$$

If the element capacitance C is specified:

$$CAPeff = C \cdot SCALE(element) \cdot M$$

Otherwise, the capacitance is calculated from the Leff, Weff and COX.

$$CAPeff = M \cdot SCALE(element) \cdot$$
$$[Leff \cdot Weff \cdot COX + 2 \cdot (Leff + Weff) \cdot CAPS$$

If COX is not specified, but THICK is not zero, then:

$$COX = \frac{DI \cdot \varepsilon o}{THICK} \qquad \textit{if DI not zero}$$

or

$$COX = \frac{\varepsilon ox}{THICK} \qquad \textit{if DI=0}$$

where

$$\varepsilon o = 8.8542149e\text{-}12 \quad \frac{F}{meter}$$

$$\varepsilon ox = 3.453148e\text{-}11 \quad \frac{F}{meter}$$

If only model capacitance CAP is specified, then

$$Peff = CAP \cdot SCALE(element) \cdot$$

Capacitance Temperature Equation

The capacitance as a function of temperature is calculated as follows:

$$t) = C \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t$$

Δt	t - tnom
t	element temperature in degrees Kelvin
	t=circuit temp + DTEMP + 273.15
tnom	nominal temperature in degrees Kelvin
	tnom=273.15 + TNOM

Polynomial Capacitor Elements

General form

Cxxx n1 n2 POLY c0 c1 … <IC=v>

Cxxx	capacitor element name. Must begin with a “C”, which can be followed by up to 15 alphanumeric characters.
n1, n2	node names
POLY	keyword identifies capacitor as nonlinear polynomial. It is a reserved name.
c0 c1 …	coefficients of a polynomial describing the element value. The capacitance is described as a function of the voltage across the capacitor. The value is:

capacitance = c0 + c1 · v + c2 · v…

IC	initial voltage across capacitor in volts. If the input netlist file contains a .IC statement, the initial conditions in the .IC statement override initial conditions specified in element statements.

Using the Linear Inductor Element

General form

```
Lxxx n1 n2 lval <TC1 <TC2>> <SCALE=val> <IC=val>
<M=val> <DTEMP=val> <R=val>
```

or

```
Lxxx n1 n2 L=val <TC1=val > <TC2=val> <SCALE=val>
<IC=val> <M=val> <DTEMP=val> + <R=val>
```

or

```
Lxxx n1 n2 L='equation' LTYPE = 0 or 1 <R=val>
```

Lxxx	inductor element name. The name must begin with an "L", followed by up to 15 alphanumeric characters.
n1	positive terminal node name
n2	negative terminal node name
TC1	first order temperature coefficient
TC2	second order temperature coefficient
SCALE	element scale factor. Default=1.0.
IC	initial current through the inductor in amperes
L	inductance in henries at room temperature

Leff = L · SCALE (element) / M

M	multiplier used to simulate multiple paralleled inductors. Default=1.0.
DTEMP	element and circuit temperature difference. Default=0.0
R	resistance in ohms of the inductor element
equation	The inductor value can be described as a function of any node voltages, branch currents, and any independent variables such as TIME, frequency (HERTZ), or temperature (TEMPER). The type of variable L depends upon is indicated by the parameter "LTYPE". Most commonly L depends upon I(Lxxx), which is assumed with the default of LTYPE=0 as explained below.
LTYPE	If inductance L is a function of I(Lxxx), set LTYPE to 0. Otherwise, set LTYPE to 1. The inductance flux is calculated differently depending on the value of LTYPE. LTYPE must be set properly to provide correct simulation results. Defining L as function of multiple variables is not recommended. Default=0.

Example

```
LLINK   42   69   1UH
LSHUNT 23 51 10U  .001 0  15  IC=15.7MA
```

Parameter Limit Checking

HSPICE writes a warning message to the output listing file if an inductive element value exceeds 0.1 henry. This feature eases identification of elements with missing units or wrong values, particularly those in automatically produced netlists.

Inductance Temperature Equation

The effective inductance as a function of temperature is provided by the following equation:

$$t)) = L \cdot (1.0 + TC1 \cdot \Delta t + TC2 \cdot \Delta t$$

Δt	t - tnom	
t	element temperature in degrees Kelvin	
		t=circuit temp + DTEMP + 273.15
tnom	nominal temperature in degrees Kelvin	
		tnom=273.15 + TNOM

Mutual Inductor Element

General form

```
Kxxx Lyyy Lzzz kvalue
or
Kxxx Lyyy Lzzz K=val
```

Kxxx	mutual (coupled) inductor element name. Must begin with a “K”, which can be followed by up to 15 alphanumeric characters.
Lyyy, Lzzz	element names of coupled inductors
kvalue, K	the coefficient of mutual coupling. The absolute value of kvalue must be greater than 0 and less than or equal to 1. If kvalue is negative, the direction of coupling is reversed. This reversal is equivalent to reversing the polarity of either of the coupled inductors. If kvalue is a parameter, the syntax of kvalue should be K=parameter.

Example

```
K43 LAA LBB 0.9999
KXFTR L1 L4 K=0.87
```

Using the “dot” convention, a “dot” appears on the first node of each inductor.

Use the mutual inductor statement to specify coupling between more than two inductors. Also, automatically calculate second order coupling effects using the GENK and KLIM options. For example, if you specify three inductors, HSPICE automatically calculates the coupling between L3 and L1, if the coupling between L1, L2 and L2, L3 is given.

Example

```
K1 L1 L2 .98
K2 L2 L3 .87
```

Create coupling between inductors with a separate coupling element. Specify mutual inductance between two inductors by the coefficient of coupling, kvalue, defined by the equation:

$$K = \frac{M}{(L_1 \cdot L_2)^{1/2}}$$

L1, L2	the inductances of the two coupled inductors
M	the mutual inductance between the inductors

Linear branch relation for transient analysis:

$$v_1 = L_1 \cdot \frac{di_1}{dt} + M \cdot \frac{di_2}{dt}$$

$$v_2 = M \cdot \frac{di_1}{dt} + L_2 \cdot \frac{di_2}{dt}$$

Linear branch relation for AC analysis:

$$V_1 = (j \cdot \omega \cdot L_1) \cdot I_1 + (j \cdot \omega \cdot M) \cdot I_2$$

$$V_2 = (j \cdot \omega \cdot M) \cdot I_1 + (j \cdot \omega \cdot L_2) \cdot I_2$$

Note: You must define an inductor reference by a mutual inductor statement, otherwise HSPICE issues an error message and terminates.

Polynomial Inductor Element

General form

```
Lxxx n1 n2 POLY c0 c1 … <IC=val>
```

Lxxx	inductor element name. Must begin with an “L”, which can be followed by up to 15 alphanumeric characters.
n1, n2	node names
POLY	keyword to identify the inductor as nonlinear polynomial
c0 c1 …	coefficients of a polynomial describing the inductor value
IC	initial current through the inductor in amperes
lval	The inductance is described as a function of the instantaneous current, i, through the inductor. The value is computed as:

$$\text{lval} = c0 + c1 \cdot i + c2 \cdot i^2$$

Using Magnetics

You can use several elements and models to analyze switching regulators, transformers, and mutual inductive circuits. These elements include magnetic winding elements, mutual cores, and magnetic core models.

You can use the HSPICE saturable core model for chokes, saturable transformers, and linear transformers. To use the model, you must provide a mutual core statement, specify the core parameters with a .MODEL statement, and provide specification of the windings around each core element with a magnetic winding element statement.

Magnetic Winding Element

General form

Lxxx n1 n2 NT=val <R=val> <IC=val>

Lxxx	the name of the winding, which must begin with an "L" followed by up to 15 alphanumeric characters
n1, n2	node names
NT	number of turns
R	DC resistance
IC	the initial current through the inductor in amperes. The "dot" convention is used to determine the direction of the turns.

Example

```
L3 4 5 NT=50 R=.01
LDRIVE 1 2 NT=100
```

Mutual Core Statement

General form

Kxxx Lyyy <Lzzz
… <Laaa>>
mname
<MAG=val>

Kxxx	the saturable core element name. Must begin with a "K", which can be followed by up to 15 alphanumeric characters.
Lyyy, *Lzzz*,	the names of the windings about the Kxxx element. At least one winding must
Laaa	be specified. There is no limit to the total number of windings. The "dot"
	convention is used to determine the direction of the turns.
mname	model name reference
MAG	initial magnetization of the core. Can be set to -1, 0 or +1, where +1, -1 correspond to positive or negative values of model parameter BS in model "mname". Default=0.0.

Example

```
K2 L1 CHOKE
K5 L3 L5 L7 T1 MAG=1
```

Magnetic Core Model

General form

.MODEL mname L (<pname1=val1>…)

mname	model name. Elements refer to the model by this name.
L	identifies a saturable core model
pname1	each saturable core model can include several model parameters.

Example

```
.MODEL CHOKE L(BS=12K BR=10K HS=1 HCR=.2 HC=.3 AC=1. LC=3.)
```

Obtain the core model parameters from the manufacturer's data. Figure 12-1: illustrates the required *b-h* loop parameters for the model. The model includes core area, length, and gap size, as well as the core growth time constant.

Example

```
*file: bhloop.sp b-h loop nonlinear magnetic core
transformer
* plot in metawaves i(l1 versus 22 to get b-h loop
.option acct method=gear post rmax=.05
.tran 1m 25m
.probe mu=lx0(k1) h=lx1(k1) b=lx2(k1) L1=lv1(l1)
L2=lv1(l2) i(l1)
k1 l1 l2 mag2
l1 1 0 nt=20
l2 2 0 nt=20
r11 1 11 1
v11 11 0 sin (0 5 60
r22 2 22 1
c22 22 0 1
.model mag2 l bs=6k br=3k hs=1 hcr=.1 hc=.8 ac=1
lc=16
.end
```

Magnetic Core Model Parameters

The magnetic core model parameters are described in Table 12-6.

Table 12-6: Definitions of Magnetic Core Model Parameters

Name (Alias)	Units	Default	Description
AC	cm · 2	1.0	core area
BS	Gauss	13000	magnetic flux density at saturation
BR	Gauss	12000	residual magnetization
HC	Oersted	0.8	coercive magnetizing force
HCR	Oersted	0.6	critical magnetizing force
HS	Oersted	1.5	magnetizing force at saturation
LC	cm	3.0	core length
LG	cm	0.0	gap length
TC	s	0.0	core growth time constant

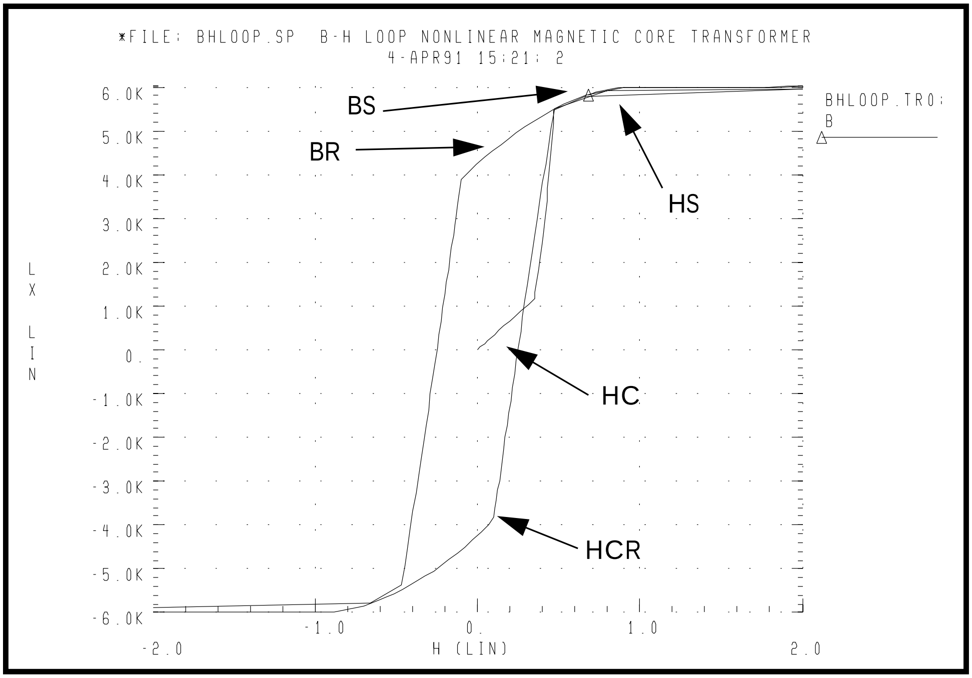


Figure 12-1: Magnetic Saturable Core Model

Jiles-Atherton Ferromagnetic Core Model

The Jiles-Atherton ferromagnetic core model is based on domain wall motion, including both bending and translation. The hysteresis-free (anhysteretic) magnetization curve is described by a modified Langevin expression. This leads to

$$m_{an} = M\dot{S} \cdot \left(\coth\left(\frac{h_e}{A}\right) - \frac{A}{h_e}\right)$$

$$h_e = h + ALPHA \cdot m_{an}$$

where

m_{an} is the magnetization level, if domain walls could move freely.

h_e is the effective magnetic field.

is the magnetic field.

MS is a model parameter that represents the saturation magnetization.

A is a model parameter that characterizes the shape of the anhysteretic magnetization.

ALPHA is a model parameter that represents the coupling between the magnetic domains.

The above equation generates anhysteretic curves when the model parameter ALPHA has a small value. Otherwise, it generates some elementary forms of hysteresis loops, which is not a desirable result. The slope of the curve at 0 can be calculated by

$$\frac{dm_{an}}{dh} = \frac{1}{3 \cdot \frac{A}{MS} - ALPHA}$$

The slope must be positive, therefore the denominator of the above equation must be positive. HSPICE generates an error message if the slope becomes negative.

The anhysteretic magnetization represents the global energy state of the material if the domain walls could move freely. But the walls are displaced and bend in the material. If the bulk magnetization *m* is expressed as the sum of an irreversible component due to wall displacement and a reversible component due to domain wall bending, then

$$\frac{dm}{dh} = \frac{(m_{an} - m)}{K} + C \cdot \left(\frac{dm_{an}}{dh} - \frac{dm}{dh}\right)$$

or

$$\frac{dm}{dh} = \frac{(m_{an} - m)}{(1 + C) \cdot K} + \frac{C}{1 + C} \cdot \frac{dm_{an}}{dh}$$

By solving the above differential equation, the bulk magnetization *m* is obtained. The flux density b is computed from *m:*

$$b = \mu_0 \cdot (h + m)$$

where μ_0, the permeability of free space, is $4\pi \cdot 10^{-7}$, and the units of h and m are in amp/meter. Then the units of b would be in tesla (Wb/meter^2).

Core Model Statement

General Form

.MODEL mname CORE (LEVEL=1 <keyword = val> ...)

Core Model Parameters

The core model parameters are listed in Table 12-7.

Table 12-7: ***Magnetic Model Parameter Definitions***

Name (Alias)	Units	Default	Description
LEVEL		2	model selector. For the Jiles-Atherton model, set LEVEL=1. LEVEL=2, the default, selects the Pheno model, which is the original HSPICE model
AREA, (AC)	cm^2	1	mean of magnetic core cross section. AC is an alias of AREA
PATH, (LC)	cm	3	mean of magnetic core path length. LC is an alias of PATH
MS	amp/meter	1e6	magnetization saturation
A	amp/meter	1e3	characterizes the shape of the anhysteretic magnetization
ALPHA		1e-3	represents the coupling between the magnetic domains
C		0.2	domain flexing parameter
K	amp/meter	500	domain anisotropy parameter

Magnetic Core Outputs

Table 12-8: Magnetic Core Element Outputs

Output Variable	Description
LX1	magnetic field, *h* (oersted)
LX2	magnetic flux density, *b* (gauss)
LX3	slope of the magnetization curve, $\frac{dm}{dh}$
LX4	bulk magnetization, m (amp/meter)
LX5	slope of the anhysteretic magnetization curve, $\frac{dm_{an}}{dh}$
LX6	anhysteretic magnetization, m_{an} (amp/meter)
LX7	effective magnetic field, h_e (amp/meter)

Jiles-Atherton Model Examples

Example 1 – Effects of Varying the ALPHA, A, and K Parameters

This example demonstrates the effects of the ALPHA, A, and K model parameters on the *b-h* curve.

Table 12-2 shows the *b-h* curves for three values of ALPHA.

Table 12-3 shows the *b-h* curves for three values of A.

Table 12-4 shows the *b-h* curves for three values of K.

HSPICE Input File

```
* Test the Jiles-Atherton model
.options post
* the following analysis studies the effect of
parameter ALPHA.
*.param palpha=0.0 pk=0.0 pc=0.0 pa=26
```

```
*.tran 0.01 1 sweep palpha poi 3 0.0 5.0e-5 1.0e-4
* the following analysis studies the effects of
parameter A.
*.param palpha=0.0 pk=0.0 pc=0.0 pa=26
*.tran 0.01 1 sweep pa poi 3 10 26 50
* the following analysis studies the effects of
parameter K.
.param palpha=0.0 pk=5 pc=1.05 pa=26
.tran 0.01 1.25 $ sweep pk poi 2 5 50
rl 1 2 1
l1 2 0 nt=50
k1 l1 ct
igen 0 1 sin(0 0.1a 1hz 0 )
.model ct core level=1 ms=420k k=pk c=pc a=pa
+ alpha=palpha area=1.17 path=8.49
.probe b=lx2(k1) h=lx1(k1) i(rl) v(1)
.probe dmdh=lx3(k1) m=lx4(k1) man=lx6(k1)
.probe l=lv1(l1)
.alter
.param pk=50
.end
```

Plots of the b-h Curve

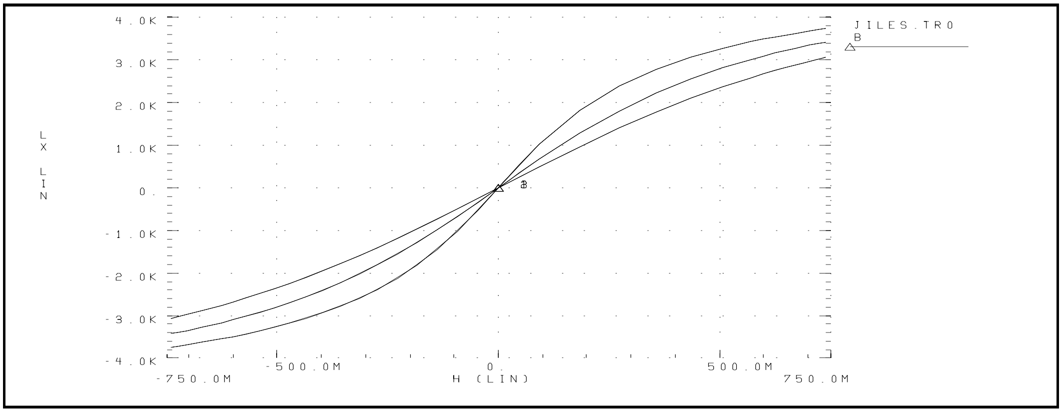


Figure 12-2: Variation of Anhysteretic b-h Curve: the Slope Increases as ALPHA Increases

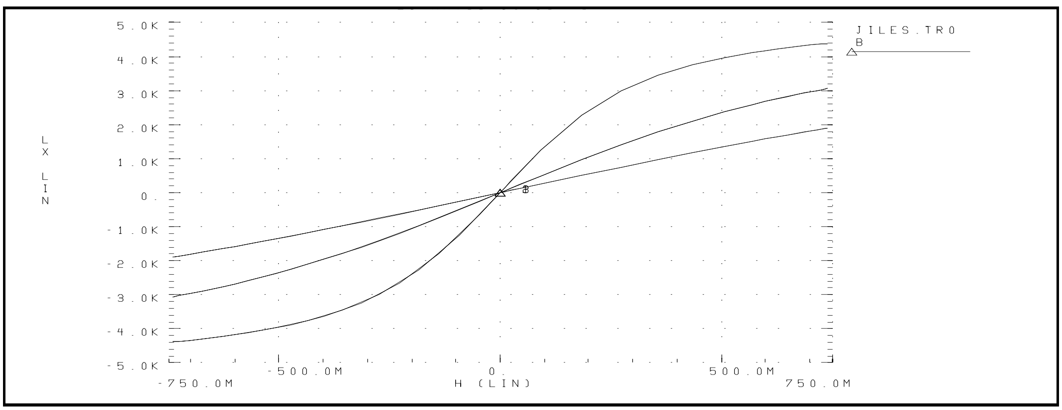


Figure 12-3: Variation of Anhysteretic b-h Curve: the Slope Decreases as A Increases

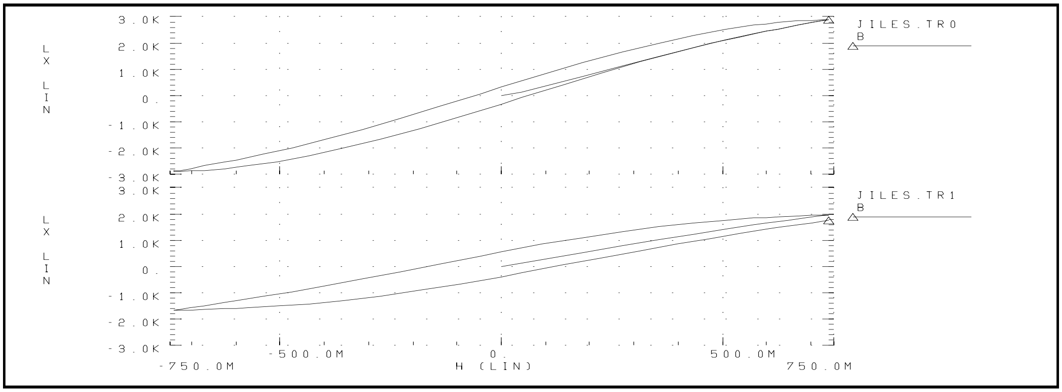


Figure 12-4: Variation of Hysteretic b-h Curve: as K Increases, the Loop Widens and Rotates Clockwise

Example 2 – Discontinuities in Inductance Due to Hysteresis

This example creates multiloop hysteresis *b-h* curves for a magnetic core. Discontinuities in the inductance, which is proportional to the slope of the *b-h* curve, can cause convergence problems. Figure 12-5: demonstrates the effects of hysteresis on the inductance of the core.

HSPICE Input File

```
*file tj2b.sp Multiloop hysteresis test using jiles-
atherton model.
.options post
.tran 0.01 5
rl 1 2 1
l1 2 0 nt=50
k1 l1 ct
igen 0 10 sin(0 0.1a 1hz 0 )
ipls 0 20 pwl(0,0 1m,0.5 1s,0.5
+ 1.001,1.0 2.000,1.0
+ 2.001,1.5 3.000,1.5
+ 3.001,2.0 4.000,2.0
+ 4.001,2.5 5.000,2.5)
```

```
gigen 0 1 cur='v(10)*v(20)'
rpls 0 20 1
rsin 0 10 1

.model ct core level=1 ms=420k k=18 c=1.05 a=26
+ alpha=2e-5 area=1.17 path=8.49

.probe b=lx2(k1) h=lx1(k1) i(rl) v(1)
.probe dmdh=lx3(k1) m=lx4(k1) dmandh=lx5(k1)
+ man=lx6(k1)
.probe l=lv1(l1) heff=lx7(k1)
.end
```

Plots of the Hysteresis Curve and Inductance

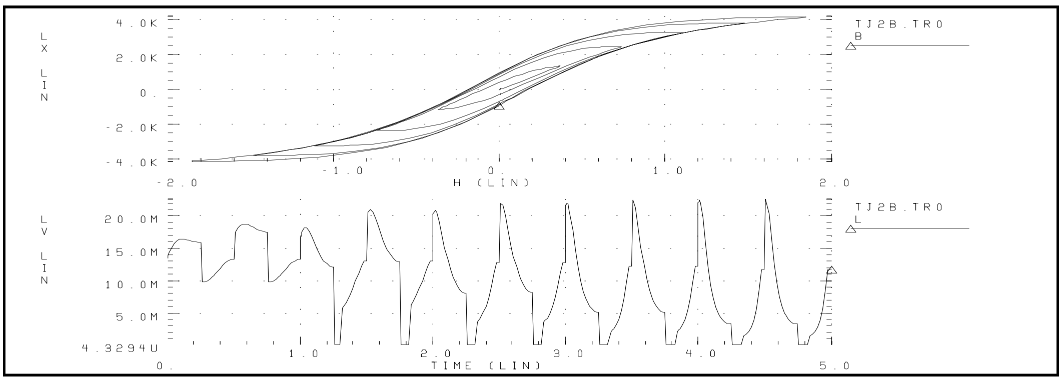


Figure 12-5: Hysteresis Curve and Inductance of a Magnetic Core

Example 3 – Optimization of Parameter Extraction

This example demonstrates the usage of optimization in the parameter extraction of the Jiles-Atherton model. Figure 12-6: shows the plots of the core output before and after optimization.

HSPICE Input File

```
*file tj_opt.sp for Jiles-Atherton model parameter
optimization.
.options post
+ delmax=5m
.param palpha=0.0
.param pms= opt1(150k,100k,500k)
+ pa =opt1(10,5,50)
+ pk=opt1(5,1,50)
+ pc= opt1(1,0,3)
.tran 0.01 1.0
.tran 0.01 1.0 sweep
+ optimize=opt1 results=bsat,br,hc model=optmod
.model optmod opt itropt=40
+ relin=1e-4 relout=1e-6
.meas bsat find par('abs(lx2(k1))') when lx1(k1)=5.0
goal=3.1k
.meas br find par('abs(lx2(k1))') when lx1(k1)=0
td=.25 goal=1k
.meas hc find par('abs(lx1(k1))') when lx2(k1)=0
td=.25 goal=.4
rl 1 2 0.01
l1 2 0 nt=20
k1 l1 ct
igen 0 1 sin(0 2a 1hz 0 )
.model ct core level=1 ms=pms k=pk c=pc a=pa
+ alpha=palpha area=1.17 path=8.49
.probe b=lx2(k1) h=lx1(k1) i(rl) v(1)
.probe dmdh=lx3(k1) m=lx4(k1) dmandh=lx5(k1)
+ man=lx6(k1)
.probe l=lv1(l1) heff=lx7(k1)
.end
```

Analysis Results Listing

```
****** transient analysis tnom= 25.000 temp= 25.000
optimization results
 residual sum of squares = 1.043893E-12
 norm of the gradient = 1.411088E-06
 marquardt scaling parameter = 1.267004E-04
 no. of function evaluations = 30
 no. of iterations = 11
 optimization completed

 norm of gradient < grad= 1.0000E-06 on last
iterations
**** optimized parameters opt1

.param pms = 267.5975k
.param pa = 27.8196
.param pk = 37.2947
.param pc = 316.4197m

*** Measure results
bsat = 3.1000E+03
 br = 9.9999E+02
 hc = 3.9880E-01
```

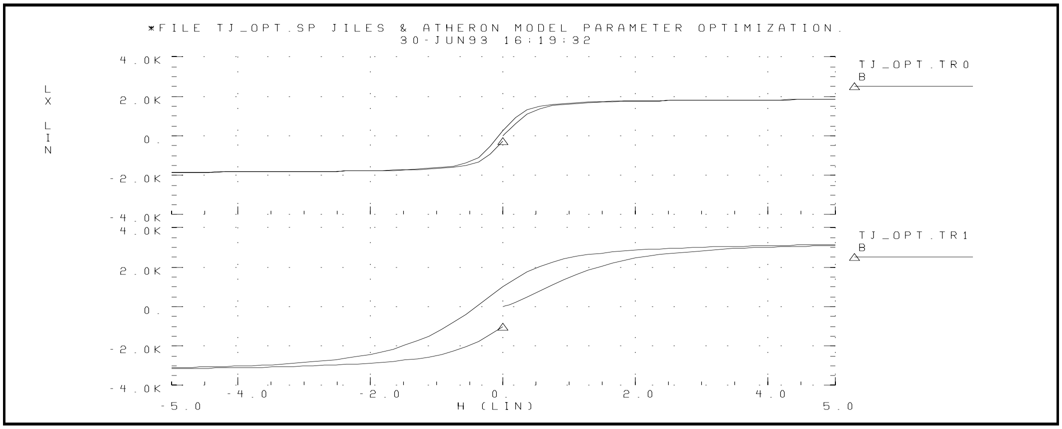


Figure 12-6: Output Curves Before Optimization (top), and After Optimization (bottom)