LTspice Component Data

Comparators
Digital
FilterProducts
Misc
Opamps
Optos
PowerProducts
References
SpecialFunctions

Component	Nodes	ID	ID2	File	Alternate File(s)
Special functions		A			
Arbitrary behavioral I source		B	I=	bi.asy	bi2.asy
Arbitrary behavioral V source		B	V=	bv.asy	
Capacitor		C		cap.asy	polcap.asy varactor.asy
Diode		D		diode.asy	LED.asy schottky.asy zener.asy
Voltage dependent voltage		E		e.asy	e2.asy
Current dependent current		F		f.asy	
Voltage dependent current		G		g.asy	g2.asy
Current dependent voltage		H		h.asy	
Independent current source		I		current.asy	load.asy load2.asy
JFET transistor NJFET		J	[njf]	njf.asy	
JFET transistor PJFET		J	[pjf]	pjf.asy	
Mutual inductance		K			(SPICE text directive)
Inductance		L		ind.asy	ind2.asy FerriteBead.asy FerriteBead2.asy
MOSFET transistor NMOS	D g s	M	[nmos]	nmos.asy	nmos4.asy
MOSFET transistor PMOS	D g s	M	[pmos]	pmos.asy	pmos4.asy
MOSFET transistor PMOS	D g s	M	[pchan]	pmos.asy	pmos4.asy
Lossy transmission line		O		ltline.asy	
Bipolar transistor NPN	C B E [S]	Q	[npn]	npn.asy	npn2.asy npn3.asy npn4.asy
Bipolar transistor PNP	C B E [S]	Q	[pnp]	pnp.asy	pnp2.asy pnp4.asy lpnp.asy
Resistor		R		res.asy	res2.asy
Voltage controlled switch		S		sw.asy	
Lossless transmission line		T		tline.asy	
Uniform RC-line		U		misc/urc.asy	misc/urc2.asy
Independent voltage source		V		voltage.asy	
Current controlled switch		W		csw.asy	
Subcircuit		X			
MESFET transistor		Z		mesfet.asy	

Component	Syntax
Special functions	Axx n1 n2 n3 n4 n5 n6 n7 n8 <model> [extra parameters]
Arbitrary behavioral source	Bxx n+ n- <V=... or I=...>
Capacitor	Cxx n+ n- <capacitance> [ic=<val.>] [Rser=<val.>] [Lser=<val.>] [Rpar=<val.>] [Cpar=<val.>] [m=<val.>]
Diode	Dxx A K <model> [area]
Voltage dependent voltage	Exx n+ n- nc+ nc- <gain>
Current dependent current	Fxx n+ n- <Vnam> <gain>
Voltage dependent current	Gxx n+ n- nc+ nc- <transcond.>
Current dependent voltage	Hxx n+ n- <Vnam> <transres.>
Independent current source	Ixx n+ n- <current>
JFET transistor	Jxx D G S <model> [area] [off] [IC=<Vds,Vgs>] [temp=<T>]
Mutual inductance	Kxx L1 L2 L3... <coeff.>
Inductance	Lxx n+ n- <inductance> [ic=<val.>] [Rser=<val.>] [Rpar=<val.>] [Cpar=<val.>] [m=<val.>]
MOSFET transistor	Mxx D G S B <model> [L=<len>] [W=<width>] [AD=<area>] [AS=<area>] [PD=<perim>] [PS=<perim>] [NRD=<value>] [NRS=<value>] [off] [IC=<Vds, Vgs, Vbs> [temp=<T>]
Lossy transmission line	Oxx L+ L- R+ R- <model>
Bipolar transistor	Qxx C B E [S] <model> [area] [off] [IC=Vbe,Vce][temp=<T>]
Resistor	Rxx n1 n2 <value>
Voltage controlled switch	Sxx n1 n2 nc+ nc- <model> [on,off]
Lossless transmission line	Txx L+ L- R+ R- ZO=<value> TD=<value>
Uniform RC-line	Uxx n1 n2 ncommon <model> L=<len> [N=<lumps>]
Independent voltage source	Vxx n+ n- <voltage>
Current controlled switch	Wxx n1 n2 <Vnam> <model> [on,off]
Subcircuit	Xxx n1 n2 n3... <subckt name>
MESFET transistor	Zxx D G S model [area] [off] [IC=<Vds,Vgs>]

Program initialization: create schematic string list (schTSL)

Start new schematic: clear schematic string list, append 1st line “Version 4”, append 2nd line “SHEET 1 <w> <h>” where last two numbers are width and height and are multiples of a 16 point grid (e.g. “960 720”).

Work through each line of netlist checking the first letter to the list of recognized symbols. When a match is found check if a secondary match is necessary. If so find the substring and check for match. If symbol is determined then append lines to schematic string list

“SYMBOL <name> <x> <y> <rotation>” e.g., SYMBOL res 96 80 R0 - (R0 R90, etc.)
“SYMATTR InstName <refdes>” e.g., SYMATTR InstName R1
“SYMATTR Value <value>” e.g., SYMATTR Value 1k

and then parse pin offsets <dx> <dy> in symbol file in LTspice sym folder. Parse corresponding net names from the current netlist line and generate name flags at each pin

“FLAG ,x+dx> <y+dy> <net>” e.g., FLAG 160 80 Vcc

Increment location on schematic: sx:= sx+32, if sx>swidth then sx:=0 and sy:=sy+32; Process next symbol.

When done processing symbols, redo line-by-line netlist search for text and comments and append to string list

“TEXT <x> <y> <orientation> <size> <value>” e.g., TEXT 112 248 Left 2 !.op

When done with text, add wires?