IBIS Interconnect SPICE Subcircuits Specification (IBIS-ISS)

Draft 0.2 3
August 11, 2009June 14, 2010

Style Definition: Heading 1: Page break before

Formatted: Centered

Formatted: Centered

Formatted: Centered

Contents

1. Overview 5
2. Goals and Scope 6
3. Best Practices 7
4. Conventions 9
5. Input Structure and Data Entry 12
   1. Input File Guidelines 12
   2. Statements and Tokens 12
   3. Special Characters 14
   4. First Character 18
   5. Delimiters 19
   6. Instance Names 19
   7. Numbers 20
   8. Parameters and Expressions 21
   9. Node Name (or Node Identifier) Conventions 23
   10. Line Continuations 24
6. Parameters 26
7. File Includes 35
8. Comments 51
9. Subcircuit Definitions 53
   1. Subckt Scoping Rules 53
10. Subcircuit Definition Ending Statements 54
11. Elements 56
   1. Subcircuits 56
   2. Linear Resistor 57
   3. Linear Capacitor 57
   4. Voltage Shunt 57
   5. Mutual Inductor 58
   6. Linear Inductor 59
   7. T-element (Ideal Transmission Line) 59
   8. W-element (Coupled Transmission Line) 61
      Format 1: RLGC Model 62
      Format 2: Frequency-Dependent Tabular Specification 64

9. S-element ... 67
10. E-element (Voltage-Controlled Voltage Source) ... 71
Linear ... 71
Laplace Transform ... 71
Pole-Zero Function ... 71
Foster Pole-Residue Form ... 72
11. F-element (Current-Controlled Current Source) ... 74
12. G-element (Voltage-Controlled Current Source) ... 76
Linear ... 76
Laplace Transform ... 76
Pole-Zero Function ... 76
Foster Pole-Residue Form ... 77
13. H-element (Current-Controlled Voltage Source) ... 80
~~1. Overview ... 3~~
~~2. Goals and Scope ... 3~~
~~3. Best Practices ... 4~~
~~4. Conventions ... 5~~
~~5. Input Structure and Data Entry ... 6~~
~~6. Parameters ... 17~~
~~7. File Includes ... 26~~
~~8. Comments and Line Continuation ... 28~~
~~9. Elements ... 29~~
~~1. Voltage-Controlled Voltage Source (VCVS) ... 44~~

Formatted: Heading 1

1. Overview

The IBIS Open Forum, in order to enable easier data exchange between users of signal/power integrity simulation and physical layout/routing software tools, is issuing a generic netlist format, to be called "IBIS Interconnect SPICE Subcircuits" (IBIS-ISS).

This format would be is similar in structure and major functions to the SPICE (Simulation Program with Integrated Circuit Emphasis) nodal syntax developed at the University of California at Berkeley and since implemented in various forms by individual software tool vendors. If approved, IBIS-ISS would beis the first industry-wide standard versionattempt to standardize of SPICE Subcircuitss subcircuit representation.

This version of IBIS-ISS is based on a subset of HSPICE ®, used with permission from Synopsys, Inc. HSPICE is a registered trademark of Synopsys, Inc.

2. Goals and Scope

The syntax of IBIS-ISS is intended ~~for use~~ to:

- describe interconnect structures (such as PCB traces, connectors, cables, etc.) electrically, for analysis in a signal integrity and/or power integrity context
- describe the arrangement or topology of interconnect structures, as they relate to each other and to active devices in a system

Formatted: List Paragraph, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Space Before: Auto, After: Auto

To these ends, IBIS-ISS ~~will include~~ includes support for:

- elementary circuit elements (resistors, capacitors, inductors)
- ~~abstraction through modular, user-defined subcircuit definitions~~
- transmission line elements (lossless and lossy)
- frequency-domain network parameters (e.g., S-parameters)
- parameter/variable passing to elements and subcircuits
- dependent sources
- string-based node naming
- user-defined comments
- abstraction through modular, user-defined subcircuit definitions

Formatted: List Paragraph, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Normal, No bullets or numbering

Formatted: List Paragraph, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Font color: Black

Formatted: Space Before: Auto, After: Auto

IBIS-ISS ~~will~~ does NOT include or cover:

- descriptions of complete netlists intended for input "as-is" to simulation tools
- model formats or "process cards" for active devices (e.g., diodes, transistors)
- independent sources
- controls or options for any simulation engine (e.g., precision, algorithm selection)
- simulation or analysis types (e.g., DC, transient)
- sweep or run control (e.g., Monte Carlo)
- geometrical descriptions for field solvers
- support for other kinds of data extraction/export (e.g., S-parameter generation)
- measurement, printing or probing
- encryption support

Formatted: Space Before: Auto, After: Auto, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Indent: First line: 0"

~~1.~~3. **Best Practices**

~~2.~~

~~3. Scaling~~

~~4.~~ Scaling of interconnect subcircuits may give different results between different simulators and should be avoided.

~~5. Global Parameters~~

~~6.~~ Global parameters may give different results between different simulators and should be avoided.

~~Exponent Range~~

~~7.~~ Exponent range ~~should~~ shall be limited to between e-60 and e+60.

~~8. Numeric Scale Factors~~

~~9. Berkeley Spice~~For maximum compatibility, IBIS-ISS does not support the “X” (Meg) scale factor. ~~and should be avoided~~.

~~10. Name Fields~~

A name field ~~should~~ shall begin with [a-z] or [A-Z] ~~], the~~ The remaining characters ~~should~~ of a name field shall be limited to

- [a-z], [A-Z], [0-9], ~!@#%&_<>?[]|:;

~~11.~~

~~Parameter Names~~

Parameter nNames ~~should~~shall begin with [a-z] or [A-Z], and the remaining characters ~~should~~shall be limited to

- [a-z] or [A-Z], [0-9], ! # $ % [ ] _

While ~~A~~ a Parameter ~~should~~ may be defined in ~~only~~ more than one .param statement within a subckt, this practice is best avoided.~~.~~

~~Node Names~~

~~12.~~ Node names shall~~ould~~ either be all numeric [0-9], or be a Name Field.

Formatted: Font: 20 pt, Bold
Formatted: Normal
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: Font: 12 pt
Formatted: List Paragraph, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"
Formatted: Font: 12 pt, Font color: Black
Formatted: Font: 12 pt
Formatted: Normal
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt
Formatted: Font: 12 pt, Not Bold
Formatted: Indent: Left: 0"
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt, Not Bold
Formatted ... [1]
Formatted: Font: 12 pt
Formatted: List Paragraph, Indent: Left: 0"
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt
Formatted: Font: 12 pt, Not Bold
Formatted: Font: 12 pt
Formatted: Normal, Indent: Left: 0"
Formatted: Font: 12 pt
Formatted: Font: 12 pt

Formatted: Font: 12 pt, Bold

13.4. Conventions

The following typographical conventions are used in IBIS-ISS documentation. Note that these may be combined (e.g., Courier font in bold type).

Convention	Description
`Courier`	Indicates command statement syntax.
Italic	Indicates a user-defined value, such as object_namewhere a specific text string will replace the italics shown in an actual IBIS-ISS file (e.g., R*xxxx* is a generic representation of a resistor element name, such as Rname).
Bold	Indicates user inputverbatim text—text you type verbatim—in syntax and examples.
[ ]	Denotes optional parameters, such as: write_file [-f filename]tokens
...	Indicates that parameters tokens can of the same type may be repeated added as many times as necessary as appropriate to the element structure: pin1 pin2 ... pinNpin
|	Indicates a choice among defined alternatives, such as low | medium | high
+	Indicates a continuation of a command linestatement across input lines. Note that continuation may only be used between tokens and shall not split any single token across lines.
/	Indicates levels of directory structure.
Edit > Copy	Indicates a path to a menu command, such as opening the Edit menu and choosing Copy.
Control-c	Indicates a keyboard combination, such as holding down the Control key and pressing c.

Formatted: Font: (Default) Courier New, 11 pt, Not Bold, Font color: Black

Formatted: Font: (Default) Courier New

Formatted: Font: Helvetica, 11 pt, Not Bold, Italic, Font color: Black

Formatted: Font: Italic

Formatted: Font: (Default) Courier New, 11 pt, Not Bold, Font color: Black

Formatted: Font: (Default) Courier New, 11 pt, Not Bold, Font color: Black

Formatted: Font: Helvetica, 11 pt, Font color: Black

Formatted: Font: Bold

Formatted: Centered

~~1.~~

Formatted: Left, Right: 1.88", No bullets or numbering, Tab stops: Not at 0.5"

~~1.~~5. Input Structure~~Netlist~~ and Data Entry

This section d~~D~~escribes the input ~~netlist~~ file and ~~methods of entering~~structures for representing input data.

1. Input ~~Netlist~~ File Guidelines

An input file co~~ntains the following:~~nsists of a collection of statements describing a portion of a complete circuit. This input file is intended for inclusion in a larger netlist or description of a complete circuit, to be used by a simulation tool.

~~1. Design netlist (subcircuits, and so on).~~

An input filename ~~can~~may be up to 1024 characters long. The input ~~netlist~~ file shall be in ASCII format (insert IEEE or ANSI definition here). The input file ~~cannot~~ shall not be in a binary, packed or compressed format.

~~**Statements in the input netlist file can be in any order.**~~

~~Table 10~~

~~**Netlist input processing is case insensitive, except for file names and their paths.**~~

2. ~~Input~~Statements and Tokens ~~Line Format~~

A statement in IBIS-ISS is a text string consisting of tokens and delimiters. An IBIS-ISS file may contain multiple statements (the number of statements is not limited by the IBIS-ISS definition, but may be limited by the computer architecture and/ or operating system used to process the file).

Any individual input statement may be up to 1024 characters long.

Statements in an input file may appear in any order.

Formatted: Heading 1, No bullets or numbering, Tab stops: Not at 0.5"

Formatted: Font: 14 pt, Bold

Formatted: Heading 2

Formatted: Font: (Default) Arial, 12 pt

Formatted: Font: (Default) Arial

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

Formatted: Font: (Default) Arial

Formatted: Font: (Default) Arial

Formatted: Heading 2

Formatted: Heading 2, No bullets or numbering

Formatted: Heading 2

Formatted: Indent: Left: 0.69", No bullets or numbering, Tab stops: Not at 0.5"

Formatted: Bullet, Indent: Left: 0.69", No bullets or numbering, Tab stops: Not at 0.5"

Formatted: Body, No bullets or numbering, Tab stops: Not at 0.5"

Formatted: Font: (Default) Arial

Any valid string of characters between two token delimiters is a token.

For the purpose of this specification, statements are grouped into the following types:

- Element instances
- Parameter definitions
- File includes
- Subcircuit definitions
- Model definitions
- Comments
- Subcircuit ending statements

Subcircuit ending statements, subcircuit definitions, model definitions, parameter definitions and file includes all begin with the dot (.) character.

The specific syntax of the above statement types are described in the sections below.

1. The input reader can accept an input token, such as:
    1. a statement name.
    2. a node name.
    3. a parameter name or value.

    Any valid string of characters between two token delimiters is a token.
4. An input statement, or equation can be up to 1024 characters long.
5. IBIS-ISS ignores differences between upper and lower case in input statementslines, except in quoted filenames.
6. To continue a statement on the next across multiple lines, enter athe plus (+) sign shall be used as the first non-numeric, non-blank character in the nextof each continued line. The + sign shall be used only between tokens and token delimiters and never to split tokens.
7. To indicate "to the power of" in your netlist, use two asterisks (**). For example, 2**5 represents two to the fifth power (2^5)
8. To continue all IBIS-ISS statements, including quoted stringsTokens with extended length (such as paths and algebraicsexpressions) may span multiple lines, using use a single whitespace character followed by a backslash (–\) or a double backslash (\\) at the end of the line containing the token to be continued on the following line. that you want to continue.
9. A single backslash preserves white space.

Formatted: Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5", Tab stops: Not at 0.5"

Formatted: No bullets or numbering, Tab stops: Not at 0.5"

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

- Parameters are used in two contexts.
    - Parameters in parameter definition statements are strings, defining names to be used as variables which are assigned specific values by the statement. These values may be numeric, strings defining an expression or equation, or strings matching parameters defined elsewhere.
    - Parameters may also appear in element instances, model definitions and subcircuit definitions. These parameters may be user-defined or may use names pre-defined by the syntax of the element.

Formatted: No bullets or numbering, Tab stops: Not at 0.5"

10.■ Parameter names must begin with an alphabetic character, but thereafter canmay contain numbers and/or some special characters:.

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

11. cCurly braces ({ }),), and/or are interpreted as square brackets ([ ]).

Formatted: Indent: Left: 1.29", No bullets or numbering, Tab stops: Not at 0.5"

12.• Names Parameter names are input tokens. Token delimiters must precede and follow names.

Formatted: Indent: Left: 1", Hanging: 0.29", Bulleted + Level: 1 + Aligned at: 1" + Indent at: 1", Tab stops: Not at 0.5"

1.• Names Parameter names canmay be up to 1024 characters long and are not case-sensitive.

2. Do not use any of the time keywords as a parameter name or node name in your netlist.

3.• The following symbols are reserved operator keywords:

( ) = " ‘

Do not use tThese symbols shall not be used as part of any parameter or node name that you define.

3. Special Characters

Formatted: Heading 2

The following table lists the special characters that canmay be used as part of node names, element parameter names, and element instance names. For detailed discussion, see the appropriate sections in this chapter.

Table 10 **Note:**

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

To avoid unexpected results or error messages, do not use the following mathematical characters in a parameter name in IBIS-ISS: * - + ^ and /.

1.Table 4 *IBIS-ISS / Netlist Special Characters*

Formatted: Indent: Left: 0.69", Hanging: 0.81", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

Special Character “Legal anywhere”=first character or any position in name “Included only”=any position except first character		**Node Name**	**Instance Name (cannot be the first character; element key letter only)**	**Parameter Name (cannot be the first character, element key letter only)**	**Delimiters**
~	tilde	Legal anywhere	Included only	Included only	n/a
!	exclamation point	Legal anywhere	Included only	Included only	n/a
@	at sign	Legal anywhere	included only	Included only	n/a
#	pound sign	Legal anywhere	Included only	Included only	n/a
$	dollar sign	Included only (avoid if after a number in node name)	Included only	Included only	In-line comment character
%	percent	Legal anywhere	Included only	included only,	n/a
^	caret	Legal anywhere	Included only	included only (avoid usage),	“To the power of”, i.e., 2^5, two raised to the fifth power
&	ampersand	Legal anywhere	Included only	Included only	n/a

*****	asterisk	included only (avoid using * in node names),	Included only	included only (avoid using in parameter names),	Comment in both IBIS-ISS Wildcard character. Double asterisk (**) is “To the power of”.
( )	parentheses	Illegal	Illegal	Illegal	Token delimiter
-	minus	included only	Included only	Illegal	n/a
_	underscore	Legal anywhere	Included only	Included only	n/a
+	plus sign	included only	Included only	included only (avoid usage); Illegal	Continues previous line, except for quoted strings (expressions, paths, algebraics)
=	equals	Illegal	Illegal	optional in .PARAM statements	Token delimiter
< >	less/more than	Legal anywhere	Included only	Included only	n/a
?	question mark	Legal anywhere	Included only	Included only	Wildcard in character in both IBIS-ISS
/	forward slash	Legal anywhere	Included only	Illegal	n/a

{ }	curly braces	included only, converts { } to [ ]	Included only	Included only	Auto-converts to square brackets ([ ])
[ ]	square brackets	Included only	Included only	Included only	n/a
****	backslash (requires a whitespace before to use as a continuation)	included only	Included only	Illegal	Continuation character for quoted strings (preserves whitespace)
****	double backslash (requires a whitespace before to use as a continuation)	included only	Illegal	Illegal	Continuation character for quoted strings (preserves whitespace)
|	pipe	Legal anywhere	Included only	Included only	n/a
,	comma	Illegal	Illegal	Illegal	Token delimiter
.	period	Illegal	Included only	Included only	~~Netlist~~ Statement ~~keyword~~identifier, (i.e., .PARAMETER, etc.).
:	colon	Included only	Included only	Included only	Delimiter for element attributes

;	semi-colon	Included only	Included only	Included only	n/a
" "	double-quotes	Illegal	Illegal	Illegal	Expression and filename delimiter
' '	single quotes	Illegal	Illegal	Illegal	Expression and filename delimiter
	Blank (whitespace)	Use before \ or \\ line continuations			Token delimiter

4. First Character

Formatted: Heading 2

The first character in every line specifies how IBIS-ISS interprets the remainder of the line.

~~1.~~Table 5 *First Character Descriptions*

Formatted: Indent: Left: 0.69", Hanging: 0.81", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

Line	If the First Character is...	Indicates
Subsequent lines of netlist, and all lines of included files	. (period)	~~Netlist keyword~~Statement identifier. For example, .PARAM
	c, C, e, E, f, F, g, G, h, H, , k, K, l, L, r, R, s, S, v, V, w, W	Element instantiation
	* (asterisk)	Comment line

+ (plus)	Continues previous line

5. Delimiters

~~1.~~ ~~An input t~~Tokens ~~is any~~are ~~item~~ strings in the input file ~~that IBIS-ISS recognizes~~separated by delimiters. Input token delimiters are: tab, blank, comma (,), equal sign (=), and parentheses ( ).

~~2.~~ In addition, ~~Single~~ single (‘) or double quotes (“) delimit tokens used as expressions ~~-~~and filenames.

~~3.~~

~~4.~~

6. Instance Names

The names of element instances begin with the element key letter, except for subcircuit instances, whose instance names begin with X. ~~(Subcircuits are sometimes called macros or modules.)~~ Instance names ~~can~~may be up to 1024 characters long.

~~1.~~Table 6 *Element Identifiers*

Key Letter (First Char)	Element	Example Line
C	Capacitor	`Cbypass 1 0 10pf`
E	Voltage-controlled voltage source	`Ea 1 2 3 4 K`
F	Current-controlled current source	`Fsub n1 n2 vin 2.0`
G	Voltage-controlled current source	`G12 4 0 3 0 10`
H	Current-controlled voltage source	`H3 4 5 Vout 2.0`
K	Linear mutual inductor (general form)	`K1 L1 L2 1`
L	Linear inductor	`LX a b 1e-9`

Formatted: Heading 2

Formatted: Indent: Left: 0.69", No bullets or numbering, Tab stops: Not at 0.5"

Formatted: Indent: Left: 0.69", First line: 0.01", No bullets or numbering, Tab stops: Not at 0.5" + 1"

Formatted: Bullet, No bullets or numbering, Tab stops: Not at 0.5"

Formatted: Heading 2

Formatted: Indent: Left: 0.69", Hanging: 0.81", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

R	Resistor	`R10 21 10 1000`
S	S-parameter element	`S1 nd1 nd2 s_model2`
V	Voltage source	`V1 8 0 DC=0`
W	Transmission Line	`W1 in1 0 out1 0 N=1 L=1`
T	Transmission Line	`Txxx in 0 out 0 Z0=50 TD=30n`
X	Subcircuit instance~~call~~	`X1 2 4 17 31 MULTI WN=100 LN=5`

~~1.~~

7. Numbers

~~You can enter n~~Numbers may be entered as integer, floating point, floating point with an integer exponent, or integer or floating point with one of the scale factors listed below.

~~1.~~Table 7 *Scale Factors*

Scale Factor	Prefix	Symbol	Multiplying Factor
T	tera	T	1e+12
G	giga	G	1e+9
MEG or X	mega	M	1e+6
K	kilo	k	1e+3
MIL	n/a	none	25.4e-6

Formatted: Font: (Default) Courier New, 11 pt

Formatted: Body

Formatted: Font: (Default) Courier New, 11 pt

Formatted: Font: (Default) Courier New, 11 pt

Formatted: Font: (Default) Courier New, 11 pt

Formatted: Font: (Default) Courier New, 11 pt, Character scale: 100%

Formatted: Indent: Left: 1", No bullets or numbering, Tab stops: Not at 0.5"

Formatted: Heading 2

Formatted: Indent: Left: 0.69", Hanging: 0.81", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

M	milli	m	1e-3
U	micro	µ	1e-6
N	nano	n	1e-9
P	pico	p	1e-12
F	femto	f	1e-15
A	atto	a	1e-18

~~Table 10~~ **Note:**

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

Scale factor A is not a scale factor in a character string that contains amps. For example, IBIS-ISS interprets the 20amps string as 20e-18mps (20^{-18}amps), but it correctly interprets 20amps as 20 amperes of current, not as 20e-18mps (20^{-18}amps).

- ~~1.~~ Numbers ~~can~~may use exponential format or engineering key letter format, but not both (1e-12 or 1p, but not 1e-6u).

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

- ~~2.~~ To designate exponents, use D or E.
- ~~3.~~ Trailing alphabetic characters are interpreted as units comments.
- ~~4.~~ Units comments are not checked.

8. Parameters and Expressions

Formatted: Heading 2

- ~~5.~~ Parameter names use IBIS-ISS name syntax rules, except that names must begin with an alphabetic character. The other characters must be either a number, or one of these characters:

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

```
! # $ % [ ] _
```

- ~~6.~~ If ~~you create~~ multiple definitions are given for the same parameter, IBIS-ISS uses the last parameter definition even if that definition occurs later in the input than a reference to the parameter.

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

- ~~7.~~ ~~You must define a~~A parameter must be defined before ~~you use~~ that parameter is used ~~to~~ in a ~~define~~ definition for another parameter.
- ~~8. When you select design parameter names, be careful to avoid conflicts with parameterized libraries.~~
- ~~9.~~ To delimit expressions, use single quotes.

10. Expressions cannot exceed 1024 characters.

1. For improved readability, use a double slash (\\) at end of a line, to continue the line.

Using Subcircuits

Reusable cells are the key to saving labor in any CAD system. This also applies to circuit simulation, in IBIS-ISS

2. To create and simulate a reusable circuit, construct it as a subcircuit.

3. Use parameters to expand the utility of a subcircuit.

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

Formatted: Heading 2

9. Node Name (or Node Identifier) Conventions

Nodes are the points of connection between elements in the input netlist. Either names or numbers may be used to designate nodes. Node numbers may be from 1 to 9999999999999999 (1 to 1e16-1); node number 0 is always ground. Letters that follow numbers in node names are ignored.

When the node name begins with a letter or a valid special character, the node name may contain a maximum of 1024 characters.

Subcircuit Node Names
Two subcircuit node names are assigned in this format.

To indicate the ground node, use either the number 0, the name `GND`, or `!GND, or GROUND, GND!`. Every node shall have at least two connections, except for transmission line nodes (unterminated transmission lines are permitted) and MOSFET substrate nodes (which have two internal connections).

Element, Instance, and Subcircuit Naming Conventions
Instances and subcircuits are elements and as such, follow the naming conventions for elements.

Element names begin with a letter designating the element type, followed by up to 1023 alphanumeric characters. Element type letters are R for resistor, C for capacitor and so on.

10. Line Continuations

Line continuations require a plus sign (+) as the first character in the line that follows. Here is an example of comments and line continuation in a netlist file:

```
.ABC Title Line
* on this line, because the first line is always a comment)
* This is a comment line
.subckt example n1 n2 $ this is an example of an inline comment
* This is a comment line and the following line is a
continuation
+ n3 n4
```

Formatted: Heading 2

~~1.~~
1.

Formatted: Left, Right: 0.83", No bullets or numbering, Tab stops: Not at 0.5"

~~1.~~6. Parameters

~~Describes how to use parameters within IBIS-ISS subckts.~~

Parameters are similar to the variables used in most programming languages. Parameters hold ~~a~~ values ~~that you~~ assigned when ~~you create your~~the circuit design is created or that are calculated ~~the simulation calculates~~ based on circuit solution values. Parameters can store static values for a variety of quantities (resistance, source voltage, rise time, and so on).

Formatted: Heading 1, No bullets or numbering, Tab stops: Not at 0.5"

Formatted: Normal

Using Parameters in Simulation (.PARAM)

Defining Parameters

Parameters in IBIS-ISS are ~~names~~ strings that ~~you~~ are associated with numeric values. ~~You can use any of the methods described below to define p~~Parameters may be defined using the methods shown below.

~~1.~~Table 9 .*PARAM Statement Syntax*

Formatted: Indent: Left: 0.69", Hanging: 0.81", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

~~Parameter~~Token	Description
Simple assignment	.PARAM <SimpleParam>=1e-12
Algebraic definition	.PARAM <AlgebraicParam>='SimpleParam*8.2' SimpleParam excludes the output variable.
Subcircuit default	.SUBCKT <SubName> <ParamDefName>=<Value> .

A parameter definition in IBIS-ISS always uses the last value found in the input netlist (subject to local versus global parameter rules). The definitions below assign a value of 3 to the *DupParam* parameter.

```
.PARAM DupParam=1
...
```

```
.PARAM DupParam=3
```

IBIS-ISS assigns `3` as the value for all instances of `DupParam`, including instances that are earlier in the input than the `.PARAM DupParam=3` statement.

All parameter values in IBIS-ISS are IEEE double floating point numbers. The parameter resolution order is:

~~1.~~1. Resolve all literal assignments.

~~1.~~2. Resolve all expressions.

~~1.~~3. Resolve all function calls.

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

~~**Error! Reference source not found.**~~ shows the parameter passing order.

Field Code Changed

Field Code Changed

~~1.~~Table 10 *Parameter Passing Order*

Formatted: Indent: Left: 0.69", Hanging: 0.81", Bulleted + Level: 1 + Aligned at: 1" + Indent at: 1", Tab stops: Not at 0.5"

.PARAM statement ()	.SUBCKT call (instance)
.SUBCKT call (instance)	.SUBCKT definition (symbol)
.SUBCKT definition (symbol)	.PARAM statement ()

Assigning Parameters

~~You can assign t~~The following types of values may be assigned to parameters:

~~1.~~■ Constant real number

~~2.~~■ Algebraic expression of real values

~~3.~~■ Predefined function

~~4.~~■ Circuit value

~~5.~~■ Model value

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

To invoke the algebraic processor, enclose a complex expression in single quotes. A simple expression consists of one parameter name.

The parameter keeps the assigned value, unless a later definition changes its value.

Inline Parameter Assignments

To define circuit values, using a direct algebraic evaluation:

```
r1 n1 0 R='1k/sqrt(HERTZ)' $ Resistance for frequency
```

Formatted: Indent: Left: 0"

Formatted: Example, Space After: 12 pt, Tab stops: 1.14", Left + 1.6", Left + 2.04", Left + 2.49", Left + 2.94", Left + 3.39", Left + 3.85", Left + 4.29", Left + 4.74", Left + 5.19", Left

Using Algebraic Expressions

In IBIS-ISS, an algebraic expression, with quoted strings, can replace any parameter in the netlist.

Some uses of algebraic expressions are:

~~6.~~■ Parameters:

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

```
.PARAM x='y+3'
```

~~7.~~■ Algebra in elements:

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

```
R1 1 0 r='ABS(v(1)/i(m1))+10'
```

In addition to using quotations, ~~you must define t~~the expression inside the `PAR( )` statement must be defined ~~for~~ to enable output. The continuation character for quoted parameter strings, in IBIS-ISS, is a double backslash (\\). (Outside of quoted strings, the single backslash (\) is the continuation character.)

Built-In Functions and Variables

In addition to simple arithmetic operations (+, -, *, /), ~~you can use t~~the built-in functions ~~listed below~~ and ~~the~~ variables listed below may be used in IBIS-ISS expressions.

~~1.~~Table 11 *IBIS-ISS Built-in Functions*

Formatted: Indent: Left: 0.69", Hanging: 0.81", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

IBIS-ISS Form	Function	Class	Description
sin(x)	sine	trig	Returns the sine of x (radians)

cos(x)	cosine	trig	Returns the cosine of x (radians)
tan(x)	tangent	trig	Returns the tangent of x (radians)
asin(x)	arc sine	trig	Returns the inverse sine of x (radians)
acos(x)	arc cosine	trig	Returns the inverse cosine of x (radians)
atan(x)	arc tangent	trig	Returns the inverse tangent of x (radians)
sinh(x)	hyperbolic sine	trig	Returns the hyperbolic sine of x (radians)
cosh(x)	hyperbolic cosine	trig	Returns the hyperbolic cosine of x (radians)
tanh(x)	hyperbolic tangent	trig	Returns the hyperbolic tangent of x (radians)
abs(x)	absolute value	math	Returns the absolute value of x: |x|
sqrt(x)	square root	math	Returns the square root of the absolute value of x: sqrt(-x)=-sqrt(|x|)
pow(x,y)	absolute power	math	Returns the value of x raised to the integer part of y: $x^{\text{(integer part of y)}}$
pwr(x,y)	signed power	math	Returns the absolute value of x, raised to the y power, with the sign of x: (sign of x)$\|x\|^y$

x**y	power		If x<0, returns the value of x raised to the integer part of y. If x=0, returns 0. If x>0, returns the value of x raised to the y power.
log(x)	natural logarithm	math	Returns the natural logarithm of the absolute value of x, with the sign of x: (sign of x)log(|x|)
log10(x)	base 10 logarithm	math	Returns the base 10 logarithm of the absolute value of x, with the sign of x: (sign of x)$\log_{10}$(|x|)
exp(x)	exponential	math	Returns e, raised to the power x: e^x
db(x)	decibels	math	Returns the base 10 logarithm of the absolute value of x, multiplied by 20, with the sign of x: (sign of x)20$\log_{10}$(|x|)
int(x)	integer	math	Returns the integer portion of x. The fractional portion of the number is lost.
nint(x)	integer	math	Rounds x up or down, to the nearest integer.
sgn(x)	return sign	math	Returns -1 if x is less than 0. Returns 0 if x is equal to 0. Returns 1 if x is greater than 0
sign(x,y)	transfer sign	math	Returns the absolute value of x, with the sign of y: (sign of y)|x|
def(x)	parameter defined	control	Returns 1 if parameter x is defined. Returns 0 if parameter x is not defined.
min(x,y)	smaller of two args	control	Returns the numeric minimum of x and y

max(x,y)	larger of two args	control	Returns the numeric maximum of x and y
[cond] ?x : y	ternary operator		Returns *x* if *cond* is not zero. Otherwise, returns *y*. .param z='condition ? x:y'
<	relational operator (less than)		Returns 1 if the left operand is less than the right operand. Otherwise, returns 0. .para x=y<z (y less than z)
<=	relational operator (less than or equal)		Returns 1 if the left operand is less than or equal to the right operand. Otherwise, returns 0. .para x=y<=z (y less than or equal to z)
>	relational operator (greater than)		Returns 1 if the left operand is greater than the right operand. Otherwise, returns 0. .para x=y>z (y greater than z)
>=	relational operator (greater than or equal)		Returns 1 if the left operand is greater than or equal to the right operand. Otherwise, returns 0. .para x=y>=z (y greater than or equal to z)
==	equality		Returns 1 if the operands are equal. Otherwise, returns 0. .para x=y==z (y equal to z)
!=	inequality		Returns 1 if the operands are not equal. Otherwise, returns 0. .para x=y!=z (y not equal to z)
&&	Logical AND		Returns 1 if neither operand is zero. Otherwise, returns 0. .para x=y&&z (y AND z)

||	Logical OR	Returns 1 if either or both operands are not zero. Returns 0 only if both operands are zero. .para x=y||z (y OR z)

~~1.~~Table 12 *IBIS-ISS Special Variables*

Formatted: Indent: Left: 0.69", Hanging: 0.81", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

IBIS-ISS Form	Function	Class	Description
time	current simulation time	control	Uses parameters to define the current simulation time, during transient analysis.
temper	current circuit temperature	control	Uses parameters to define the current simulation temperature, during transient/temperature analysis.
hertz	current simulation frequency	control	Uses parameters to define the frequency, during AC analysis.

Parameter Scoping and Passing

If ~~you use~~ parameters are used to define values in sub-circuits, ~~you need to create~~ fewer similar cells should be used, to provide enough functionality in the resulting ~~your~~ library. ~~You can pass c~~Circuit parameters may be passed into hierarchical designs, and ~~assign~~ different values may be assigned to the same parameter within individual cells, when ~~you run~~ simulations are run.

A parameter is defined either by a .parameter statement (local to that subcircuit), or ~~can~~may be passed into a subcircuit, or ~~can~~may be defined on a .subckt definition line.

(Some details need to be clarified on this)

```
.param x=0
.subckt def
.param x=1
x1 1 2 abc x=2
.subckt abc 1 2 x=3
.param x=3
r1 1 2 R=x
.ends abc
.ends def
.end
```

Formatted: Font: (Default) Courier New, 11 pt

Formatted: Indent: Left: 0.69", Space After: 0 pt, Line spacing: single

~~How you handle hierarchical parameters depends on how you construct and analyze your cells. You can construct a design in which information flows from the top of the design, down into the lowest hierarchical levels.~~ The specific details of any particular hierarchy are left to the choice of the user.

~~2. To construct a library of small cells that are individually controlled from within, set *local* parameters and build up to the block level.~~

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

This section describes the scope of parameter names, and how IBIS-ISS resolves naming conflicts between levels of hierarchy.

Library Integrity (Needs careful discussion)

Integrity is a fundamental requirement for any symbol library. Library integrity can be as simple as a consistent, intuitive name scheme, or as complex as libraries with built-in range checking.

Library integrity might be poor ~~if you use~~ libraries from different vendors are used in a single circuit design. Because names of circuit parameters are not standardized between vendors, two components can include the same parameter name for different functions. For example, one vendor might build a library that uses the name `Tau` as a parameter to control one or more subcircuits in their library. Another vendor might use `Tau` to control a different aspect of their library. If ~~you set~~ a global parameter named `Tau` is used to control one library, ~~you also modify~~ the behavior of the second library may be unintentionally affected.~~, which might not be the intent~~. This is why Best Practices recommends that Global Parameters be avoided.

7.File Includes

Formatted: Heading 1

Formatted: Automatically adjust right indent when grid is defined, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Subcircuits

`X`<*subcircuit_name*> adds an instance of a subcircuit to your netlist. You must already have defined that subcircuit in your netlist by using a `.SUBCKT` command.

Formatted: Head1

Syntax

X<subcircuit_name> n1 <n2 n3 …> subnam

<parnam = val &> <M = val> <S=val> <DTEMP=val>

Formatted: Head1, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Argument	Definition
X<*subcircuit_name*>	Subcircuit element name. Must begin with an X, followed by up to 15 alphanumeric characters.
n1 …	Node names for external reference.
subnam	Subcircuit model reference name.
Parnam	A parameter name set to a value (val) for use only in the subcircuit. It overrides a parameter value in the

Formatted: Head1

Formatted: Head1

Formatted: Head1

Formatted: Head1

Formatted: Head1

~~subcircuit definition, but is overridden by a value set in a .PARAM statement.~~

~~14.~~

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Heading 1

Formatted: Heading 1, Automatically adjust right indent when grid is defined, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Font: Arial

~~Subckt scoping rules~~

Formatted: Font: Arial, 20 pt, Character scale: 100%

Formatted: Heading 1, Indent: Left: 0", Automatically adjust right indent when grid is defined, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Font: Arial, 20 pt

Formatted: Font: Arial

~~A .subckt or .model definition must occur in the subckt in which the subckt or model is referenced, or in a calling subckt at any level above.~~

Formatted: Font: Arial, Character scale: 100%

Formatted: Font: Arial

Formatted: Heading 1, Automatically adjust right indent when grid is defined, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Font color: Black

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

~~15. .INCLUDE~~

Includes another file's contents ~~netlist~~ in the current file ~~as a subcircuit of the current netlist~~.

Syntax

```
.INCLUDE 'file_path file_name'
.inc 'file_path file_name'
```

Arguments

Argument	Description
file_path	Path name of a file for computer operating systems that support tree-structured directories. An include file can contain nested .INCLUDE calls to itself or to another include file. If ~~you use~~ a relative path is used in a nested .INCLUDE call, the path starts from the directory of the parent .INCLUDE file, not from the current working directory. If the path starts from the current working directory, IBIS-ISS ~~can~~may also find the .INCLUDE file, but prints a warning.
file_name	Name of a file to include in the data file. The file path, plus the file name, ~~can~~may be up to 16 characters long. ~~You can use a~~Any ~~valid file name~~ name valid ~~for~~ under the computer's operating system may be used.

Description

Use this command to include another netlist in the current netlist. ~~You can include a~~A netlist may be used as a subcircuit in one or more other netlists. ~~You must enclose t~~The file path and file name shall be enclosed in single or double quotation marks. ~~Otherwise, an error message is generated.~~

~~Example~~

```
.INCLUDE `/myhome/subcircuits/diode_circuit´
```

Formatted: Font: Helvetica, 12 pt, Font color: Black

Formatted: Font: Bold

Formatted: Font: (Default) Courier New, 12 pt, Not Bold, Font color: Black

Formatted: Font: (Default) Courier New

Formatted: Font: (Default) Courier New, 12 pt, Not Bold, Font color: Black

Formatted: Font: (Default) Courier New

Formatted: Font: Helvetica, 11 pt, Not Bold, Italic, Font color: Black

Formatted: Font: Italic

Formatted: Font: Helvetica, 11 pt, Not Bold, Italic, Font color: Black

Formatted: Font: Italic

Formatted: Indent: Left: 0"

Formatted: Font: 12 pt, Not Bold

Formatted: Code, Automatically adjust right indent when grid is defined, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Node Name (or Node Identifier) Conventions

Nodes are the points of connection between elements in the input netlist. Either names or numbers may be used to designate nodes. Node numbers can be from 1 to 9999999999999999 (1 to 1e16-1); node number 0 is always ground. Letters that follow numbers in node names are ignored.

When the node name begins with a letter or a valid special character, the node name can contain a maximum of 1024 characters.

Subcircuit Node Names
Two subcircuit node names are assigned in this format.

To indicate the ground node, use either the number 0, the name GND, or !GND, or GROUND, GND!. Every node should have at least two connections, except for transmission line nodes (unterminated transmission lines are permitted) and MOSFET substrate nodes (which have two internal connections).

Element, Instance, and Subcircuit Naming Conventions
Instances and subcircuits are elements and as such, follow the naming conventions for elements.

Element names begin with a letter designating the element type, followed by up to 1023 alphanumeric characters. Element type letters are R for resistor, C for capacitor and so on.

Formatted: Centered

Formatted: Automatically adjust right indent when grid is defined, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

8. Comments ~~and Line Continuation~~

Comments require an asterisk (*) as the first character in a line or a dollar sign ($) directly in front of the comment anywhere on the line. For example:

```
* <comment_on_a_line_by_itself>
```

Formatted: Font: Italic

or

```
<–IBIS-ISS statement> $ <comment following input>
```

Formatted: Font: Italic

Formatted: Font: Italic

Comment statements may appear anywhere in the circuit description. The dollar sign ($) must be used for comments that do *not* begin in the first character position on a line (for example, for comments that follow simulator input on the same line). If it is not the first nonblank character, then the dollar sign must be preceded by either:

~~Note:~~ Whitespace
~~Note:~~ Comma (,)
~~Note:~~ Valid numeric expression

Formatted: Automatically adjust right indent when grid is defined, No bullets or numbering, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Automatically adjust right indent when grid is defined, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

The dollar sign may also be used within node or element names. For example:

```
* RF=1K GAIN SHOULD BE 100
$ ~~MAY THE FORCE BE WITH MY CIRCUIT~~CIRCUIT EXAMPLE
VIN 1 0 PL 0 0 5V 5NS $ 10v 50ns
R12 1 0 1MEG $ FEED BACK
.PARAM a=1w$comment a=1, w treated as a space and ignored
.PARAM a=1k$comment a=1e3, k is a scale factor
```

A dollar sign is the preferred way to indicate comments, because of the flexibility of its placement within the code.

~~Line continuations require a plus sign (+) as the first character in the line that follows. Here is an example of comments and line continuation in a netlist file:~~

```
~~.ABC Title Line~~
~~* on this line, because the first line is always a comment)~~
~~* This is a comment line~~
~~.MODEL n1 NMOS $ this is an example of an inline comment~~
~~* This is a comment line and the following line is a continuation~~
```

Formatted: Font: Courier New

Model Definitions (.MODEL Statements)

Model definitions are used to specify the electrical parameters for W-element and S-element instances. They can be considered a special form of subcircuit definition, in which the defined subcircuit is only available to W- and S-elements.

The specific syntax for W-element and S-element .MODEL definitions are detailed below, as part of the W-element and S-element portions of the IBIS-ISS specification. Note that .MODEL statements are hierarchically at the same level as element instances.

Formatted: Body

Formatted: Character scale: 0%

9. Subcircuit Definitions

Syntax

```
.subckt name n1 n2…
statement
statement
statement
…
.ends
```

1. Subckt Scoping Rules

A .subckt or .model definition must occur in the subckt in which the subckt or model is referenced, or in a calling subckt at any level above.

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

10. Subcircuit Definition Ending Statements

Subcircuit definitions must be ended with the .ends token. See Subcircuit Definitions above for syntax and examples.

Formatted: Normal, Indent: Left: 0.5"

Formatted: Font: Arial

Formatted: Automatically adjust right indent when grid is defined, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

11. + LEVEL=3

Formatted: Font: (Default) Courier New

Formatted: Heading 1, Left, Automatically adjust right indent when grid is defined, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Elements

1. Subcircuits

Formatted: Heading 2, Numbered + Level: 1 + Numbering Style: 1, 2, 3, ... + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Indent: Left: 0", First line: 0"

Using Subcircuits

Reusable cells are the key to saving labor in any CAD system. This also applies to circuit simulation, in IBIS-ISS

- To create and simulate a reusable circuit, construct it as a subcircuit.
- Use parameters to expand the utility of a subcircuit.

`X`*<subcircuit_name>* creates an instance of a subcircuit. . The subcircuit must have already been defined elsewhere in the IBIS-ISS file using a `.SUBCKT` command.

Syntax

```
Xxxxx n1 [n2 n3 …] subnam
[parnam = val] [M = val]
```

Argument	Definition
X*<subcircuit_name>*	Subcircuit element name. Must begin with an X, followed by up to 15 alphanumeric characters.
n1 …	Node names for external reference.
subnam	Subcircuit model reference name.
Parnam	A parameter name set to a value (val) for use only in the subcircuit. It overrides a parameter value in the subcircuit definition, but is overridden by a value set in a `.PARAM` statement.
M	Multiplier

2. Linear Resistor~~s~~

Syntax

```
Rxxx node1 node2 [R =] value
```

The value of a linear resistor ~~can~~may be a constant, or an expression of parameters.

~~Parameter~~Token	Description
Rxxx	Name of a resistor
node1 and node2	Names or numbers of the connecting nodes
value	resistance value, in ohms

3. Linear Capacitor~~s~~

Syntax

```
Cxxx node1 node2 [C=]value
```

The value of a linear capacitor ~~can~~may be a constant, or an expression of parameters.

~~Parameter~~Token	Description
C*xxx*	Name of a capacitor. Must begin with C, followed by up to 1023 alphanumeric characters.
node1 ~~and~~ node2	Names ~~or numbers~~ of connecting nodes.
value	C~~c~~apacitance value, in f~~F~~arads.

4. Voltage Shunt

~~This~~ A voltage shunt creates a short between ~~nodes node1 and node2~~two nodes.

Formatted: Heading 2

Formatted: Font: Helvetica, Font color: Black

Formatted: Font: Helvetica

Formatted: Font: Helvetica-Bold, Font color: Black

Formatted: Heading 2, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Formatted: Font: Helvetica, 11 pt, Italic, Font color: Black

Formatted: Font: Helvetica, 11 pt, Italic, Font color: Black

Formatted: Font: Helvetica, 11 pt, Italic, Font color: Black

Formatted: Font: Italic

Formatted: Font: Helvetica, 11 pt, Italic, Font color: Black

Formatted: Font: Italic

Formatted: Font: Helvetica

Formatted: Font: Helvetica-Bold, Font color: Black

Formatted: Heading 2, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Syntax

```
Vxxx node1 node2 [DC=]0
```

Token	Description
Vxxx	Voltage shunt element name. Must begin with K, followed by up to 1023 alphanumeric characters.
node1, node 2	Nodes between which the shunt is placed.
DC=0	The zero value is required and sets the voltage between the nodes at zero volts. The text “DC=” is optional.

~~This creates a short between nodes node1 and node2~~

5. Mutual Inductor~~s~~

A mutual inductor describes inductive coupling between two defined inductors.

Syntax

~~General form:~~

```
Kxxx Lyyy Lzzz [K=]coupling
```

~~Parameter~~Token	Description
Kxxx	Mutual inductor element name. Must begin with K, followed by up to 1023 alphanumeric characters.
Lyyy	Name of the first of two coupled inductors. This inductor must be defined elsewhere in the file.
Lzzz	Name of the second of two coupled inductors. This inductor must be defined elsewhere in the file.

Formatted: Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Formatted: Font: Italic

Formatted: Font: Courier, Italic, Font color: Black

Formatted: Font: Italic

Formatted: Font: Helvetica, Font color: Black

Formatted: Font: Helvetica

Formatted: Heading 2

Formatted: Syntax, Tab stops: 1.14", Left + 1.6", Left + 2.04", Left + 2.5", Left + 2.94", Left + 3.39", Left + 3.85", Left + 4.29", Left + 4.75", Left + 5.19", Left

Formatted: Font: Helvetica

~~K=~~*coupling*	Coefficient of mutual coupling. ~~K~~ This is a unitless number, with magnitude > 0. If ~~K~~ the coupling coefficient is negative, the direction of coupling reverses. This is equivalent to reversing the polarity of either of the coupled inductors. Use the K=~~*coupling*~~ *xxx* syntax when defining the coupling coefficient using a parameter ~~value~~ name or an equation~~, and t~~. The ~~keyword~~ pre-defined parameter “k=” ~~can~~may be omitted.

6. Linear Inductor~~s~~

Syntax

```
Lxxx node1 node2 [L =] inductance
```

~~Parameter~~Token	Description
Lxxx	Name of an inductor.
node1, ~~*and*~~ *node2*	Names or numbers of the connecting nodes.
inductance	inductance value, in h~~H~~enries.

7. T-element (Ideal Transmission Line~~s~~)

Syntax

~~General form:~~

```
Txxx in refin out refout Z0=val TD=val [L=val]
+ [IC=v1, i1, v2, i2]
```

Formatted: Font: Helvetica, 11 pt, Italic, Font color: Black
Formatted: Font: Helvetica, 11 pt, Italic, Font color: Black
Formatted: Heading 2
Formatted: Character scale: 0%
Formatted: Body
Formatted: Font: 11 pt, Not Bold, Italic
Formatted: Font: 11 pt, Not Bold, Italic
Formatted: Font: Italic
Formatted: Font: 11 pt, Not Bold, Italic
Formatted: Font: Italic
Formatted: Indent: Left: 0.5"
Formatted: Heading 2, Indent: Left: 0", Automatically adjust right indent when grid is defined, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers
Formatted: Indent: Left: 0.5"
Formatted: Font: (Default) Courier New
Formatted: Font: (Default) Courier New, Italic
Formatted: Font: (Default) Courier New
Formatted: Font: (Default) Courier New, Italic
Formatted: Font: (Default) Courier New
Formatted: Font: (Default) Courier New, Italic
Formatted: Font: (Default) Courier New
Formatted: Font: (Default) Courier New, Italic
Formatted: Font: (Default) Courier New
Formatted: Font: (Default) Courier New, Italic
Formatted: Font: (Default) Courier New, Italic
Formatted: Font: (Default) Courier New, Italic
Formatted: Font: (Default) Courier New, Italic

~~Parameter~~Token	Description
Txxx	Lossless transmission line element name. Must begin with T, followed by up to 1023 alphanumeric characters.
In	Signal input node.
Refin	Ground reference for the input signal.
Out	Signal output node.
Refout	Ground reference for the output signal.
Z0	Characteristic impedance of the transmission line (Ohms).
TD	Propagation time delay of the transmission line (in seconds). If physical length (L) is specified, then units for TD are considered in seconds per meter.
L	Physical length of the transmission line, in units of meters. Default=1.

Formatted: Indent: Left: 0.5"

Formatted: Heading 2, No bullets or numbering

Table 7 8. W-element (Modeling of Coupled Transmission Line)s

Describes how to use basic transmission line simulation equations and an optional method for computing the parameters of transmission line equations.

The W-element is a versatile transmission line model that you can apply to efficiently and accurately simulate transmission lines, ranging frommay be used to describe a variety of transmission line structures, from -a simple lossless line to complex frequency-dependent lossy-coupled lines.

Input Syntax for the W-element

Syntax:

```
Wxxx i1 i2 ... iN iR o1 o2 ... oN oR N=val L=val
+ [RLGCMODEL=name | TABLEMODEL=name ]
```

ParameterToken	Description
N	Number of signal conductors (excluding the reference conductor).
i1...iN	Node names for the near-end signal-conductor terminal
iR	Node name for the near-end reference-conductor terminal.
o1... oN	Node names for the far-end signal-conductor terminal
oR	Node name for the far-end reference-conductor terminal.
L	Length of the transmission line.
RLGCMODEL	Name of the RLGC model.

TABLEMODEL Name of the frequency-dependent tabular model

The W-element supports t~~hese~~wo formats to specify transmission line properties:

- ~~Table 5~~ FormatModel 1: RLGC ~~Model~~ specification

  - ~~3.~~ Internally specified in a `.MODEL` statement.
  - ~~4.~~ Externally specified in a different file.

- ~~Table 5~~ Format~~Model~~ 2~~4~~: Frequency-dependent tabular specification~~model.~~

P~~Normally, you can specify p~~arameters in the W-element ~~card~~ element declaration may be declared in any order. Specify the number of signal conductors, N, after the list of nodes. ~~You can intermix t~~The nodes and parameters in the W-element ~~card~~element declaration may be interspersed.

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

~~Input Model~~Format 1: ~~W-element,~~ RLGC Model

Formatted: Heading 3

Equations and Parameters on page 96 (NOTE: Do we want to include these explanations) describes the inputs of the W-element per unit length matrices: R_o (DC resistance), L, G, C, R_s (skin effect), and G_d (dielectric loss)

The W-element does not limit any of the following parameters:

- ~~Table 5~~ Number of coupled conductors.
- ~~Table 5~~ Shape of the matrices.
- ~~Table 5~~ Line loss.
- ~~Table 5~~ Length or amount of frequency dependence.

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

The RLGC text file contains frequency-dependent RLGC matrices per unit length. The W-element also handles frequency-independent RLGC, and lossless (LC) lines. It does not support RC lines.

Because RLGC matrices are symmetrical, the RLGC model specifies only the lower triangular parts of the matrices. The syntax of the RLGC model for the W-element is:

```
.MODEL name W MODELTYPE=RLGC N=val
+ Lo=matrix_entries
+ Co=matrix_entries [Ro=matrix_entries Go=matrix_entries]
+ Rs=matrix_entries wp=val Gd=matrix_entries Rognd=val
+ Rsgnd=val Lgnd=val
```

Formatted: Font: Italic

~~Parameter~~ Token	Description
N	Number of conductors (same as in the element card).
L	DC inductance matrix, per unit length $\left[\frac{H}{m}\right]$.
C	DC capacitance matrix, per unit length $\left[\frac{F}{m}\right]$.
Ro	DC resistance matrix, per unit length $\left[\frac{\Omega}{m}\right]$.
Go	DC shunt conductance matrix, per unit length $\left[\frac{S}{m}\right]$.
Rs	Skin effect resistance matrix, per unit length $\left[\frac{\Omega}{m\sqrt{Hz}}\right]$.
Gd	Dielectric loss conductance matrix, per unit length $\left[\frac{S}{m \cdot Hz}\right]$.
wp	Angular frequency of the polarization constant [radian/sec] (see Introduction to the Complex Dielectric Loss Model on page 99)). When the wp value is specified, the unit of Gd becomes [S/m].
Lgnd	DC inductance value, per unit length for grounds $\left[\frac{H}{m}\right]$ (reference line).
Rognd	DC resistance value, per unit length for ground $\left[\frac{\Omega}{m}\right]$.

Rsgnd

Skin effect resistance value, per unit length for ground $\left[\frac{\Omega}{m\sqrt{Hz}}\right]$.

The following input netlist file shows RLGC input for the W-element:

```
* W-Element example, four-conductor line
W1 N=3 1 3 5 0 2 4 6 0 RLGCMODEL=example_rlc l=0.97


* RLGC matrices for a four-conductor lossy
.MODEL example_rlc W MODELTYPE=RLGC N=3
+ Lo=
+ 2.311e-6
+ 4.14e-7 2.988e-6
+ 8.42e-8 5.27e-7 2.813e-6
+ Co=
+ 2.392e-11
+ -5.41e-12 2.123e-11
+ -1.08e-12 -5.72e-12 2.447e-11
+ Ro=
+ 42.5
+ 0 41.0 + 0 0 33.5
+ Go= + 0.000609
+ -0.0001419 0.000599
+ -0.00002323 -0.00009 0.000502
+ Rs=
+ 0.00135
+ 0 0.001303
+ 0 0 0.001064
+ Gd=
+ 5.242e-13
+ -1.221e-13 5.164e-13
+ -1.999e-14 -7.747e-14 4.321e-13
```

Using RLGC Matrices

RLGC matrices in the RLGC model of the W-element are in the Maxwellian format

~~Input Model~~Format 24: Frequency-Dependent Tabular Specification~~Model~~

Formatted: Heading 3

~~You can use t~~The tabular RLGC model may be used as an extension of the analytical RLGC model to model any arbitrary frequency-dependent behavior of transmission lines (this model does not support RC lines).

~~You can use this extension of t~~The W-element syntax ~~to specify a~~ supports table~~ models~~s of data (use a `.MODEL` statement of type w). To accomplish this, the `.MODEL` statement refers to .MODEL statements where the “type” is SP (described in Small-Signal Parameter Data Frequency Table Model (SP Model) on page 77), which contain the actual table data for the RLGC matrices.

~~Table 10~~ **Note:**

To ensure accuracy, the W-element tabular model requires the following:

~~Table 5~~ ▪ R and G tables require zero frequency points.

~~Table 5~~ ▪ L and C tables require infinity frequency points as well as zero frequency points.

To specify a zero frequency point, ~~you may use~~ the pre-defined DC ~~keyword~~ parameter may be used. Alternatively, ~~or~~ the f~~=0 data entry~~ parameter ~~-~~in the DATA field of the SP model may be set to a value of 0. To specify an infinity frequency point, use the INFINITY ~~keyword~~ token~~of the SP model~~.

See also, Small-Signal Parameter Data Frequency Table Model (SP Model) on page 77.

~~Notation Used~~

~~Table 5 Lower-case variable: Scalar quantity~~

~~Table 5 Upper-case variable: Matrix quantity~~

~~Table 5 All upper-case words: Keyword~~

~~Table 5 Parentheses and commas: Optional~~

~~Table~~W-element Model Definition ~~Card~~ Syntax

```
.MODEL name W MODELTYPE=TABLE [FITGC=0|1] N=val
+ LMODEL=l_freq_model CMODEL=c_freq_model
+ [RMODEL=r_freq_model GMODEL=g_freq_model]
```

~~Parameter~~Token	Description
FITCG	~~Keyword~~ Pre-defined parameter token for the W~~-Model (w/~~ -element with MODELTYPE=TABLE~~)~~. A value of 1~~=~~ instructs the tool to run a causality check on the data. ~~,~~A value of 0~~=~~ turns any causality checking off (default)
N	Number of signal conductors (excluding the reference conductor).

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

LMODEL	SP model name for the inductance matrix array.
CMODEL	SP model name for the capacitance matrix array.
RLMODEL	SP model name for the resistance matrix array. By default, it is zero.
GMODEL	SP model name for the conductance matrix array. By default, it is zero.

Formatted: Heading 2

9. S-element Syntax

Use the following S-element syntax to show the connections within a circuit:An S-element is a frequency-domain set of network data, described using scattering parameters.

Syntax

```
Sxxx nd1 nd2 ... ndN [ndRef]
+ [MNAME=Smodel_name]
+ [FBASE = base_frequency] [FMAX=maximum_frequency]
```

ParameterToken	Description
nd1 nd2...ndN	Nodes of an S-element Three kinds of definitions are present: ■ With no reference node ndRef, the default reference node is GND. Each node ndi (i=1~N) and GND construct one of the N ports of the S-element. ■ With one reference node, ndRef is defined. Each node ndi (i=1~N) and the ndRef construct one of the N ports of the S-element. ■ With an N reference node, each port has its own reference node. TYou can write the node definition may be written more clearlyin a clearer way as: nd1+ nd1- nd2+ nd2- ... ndN+ ndN- Each pair of the nodes (ndi+ and ndi-, i=1~N) constructs one of the N ports of the S-element.
ndRef	Reference node
MNAME	Name of the S model; Note that string parameters are supported in calling an MNAME.

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Indent: Left: 0", Hanging: 0.19", Bulleted + Level: 1 + Aligned at: 0" + Indent at: 0", Tab stops: Not at 0.5"

FBASE	Base frequency to use for transient analysis. This value becomes the base frequency point for Inverse Fast Fourier Transformation (IFFT).

~~3.~~ ▪ If ~~you do not set~~ this value is not set, the base frequency is a reciprocal value of the transient period.

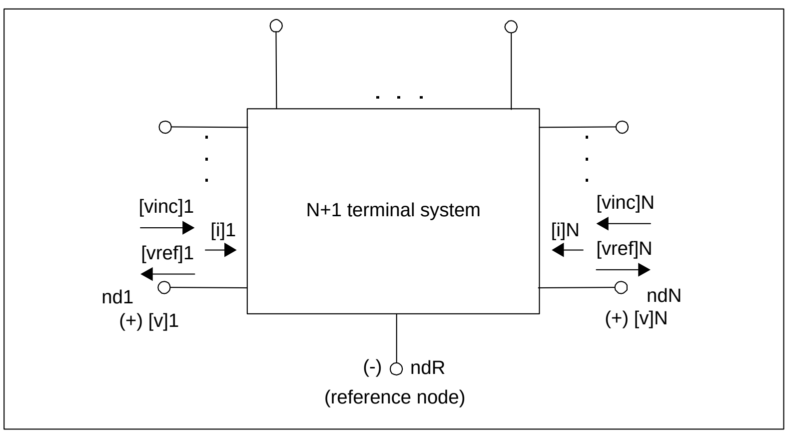
~~4.~~ ▪ If you set a frequency that is smaller than the reciprocal value of the transient, then transient analysis performs circular convolution, and uses the reciprocal value of FBASE as its base period.

Formatted: Indent: Left: 0", Hanging: 0.19", Bulleted + Level: 1 + Aligned at: 0" + Indent at: 0", Tab stops: Not at 0.5"

FMAX	Maximum frequency use in transient analysis. Used as the maximum frequency point for Inverse Fast Fourier Transformation (IFFT).

The nodes of the S-element must come first. You can specify all the optional parameters in both the
S-element and S model statements, except for `MNAME` argument.

~~You can enter t~~The optional arguments may be entered in any order, and the parameters specified in the element statement have a higher priority.



~~1.~~Figure 27 *Terminal Node Notation*

Formatted: Indent: Left: 0.69", Hanging: 0.9", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69", Tab stops: Not at 0.5"

Node Example

The following example illustrates the *nd1 nd2...ndN—no reference*, *single reference*, and multi-reference parameters.

```
**S-parameter example
```

```
* no reference
S_no_ref n1 n2 mname=s_model

* single reference
S_one_ref n1 n3 gnd mname=s_model

*multi-reference
S_multi_ref n1 gnd n4 gnd mname=s_model
```

The S-element must have a call to one of the supported S-parameter file formats (IBIS-ISS gets the number of ports from the S-parameter file ~~You can also explicitly specify~~ The number of ports, 'n', may be specified explicitly as N=n ~~where 'n' is the number of ports.~~.

~~Table 5~~ ■ For n terminals, the S-element assumes no reference node.

~~Table 5~~ ■ For n+1 terminals, the S-element assumes one reference node.

~~Table 5~~ ■ For 2n terminals, the S-element assumes signal nodes and n reference nodes. Each pair of nodes is a signal and a reference node.

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

S Model Syntax

Use the following syntax to describe specific S models:

```
.MODEL Smodel_name S [N=dimension]
+ [TSTONEFILE=filename

+ [FBASE=base_frequency] [FMAX=maximum_frequency]
```

~~Parameter~~Token	Description
Smodel_name	Name of the S model.
S	Specifies that the model type is an S model.
N	S model dimension, which is equal to the terminal number of an S-element and excludes the reference node.

TSTONEFILE	Specifies the name of a Touchstone file. Data contains frequency-dependent array of matrixes. Touchstone files must follow the .s#p file extension rule, where # represents the dimension of the network. Note that string parameters are supported for TSTONEFILE Example: .subckt sparam n1 n2 tsfile=str('ss_ts.s2p') S1 n1 n2 0 mname=s_model .model s_model S TSTONEFILE=str(tsfile) .ends x1 A B sparam tsfile=str('ss_ts.s2p') … For details, see *Touchstone® File Format Specification* by the EIA/IBIS Open Forum (http://www.eda.org).
FBASE	Base frequency used for transient analysis. IBIS-ISS uses this value as the base frequency point for Fast Inverse Fourier Transformation (IFFT). ~~2.~~ ▪ If FBASE is not set, IBIS-ISS uses a reciprocal of the transient period as the base frequency. ~~3.~~ ▪ If FBASE is set smaller than the reciprocal value of transient period, transient analysis performs circular convolution by using the reciprocal value of FBASE as a base period.
FMAX	Maximum frequency for transient analysis. Used as the maximum frequency point for Inverse Fast Fourier Transform (IFFT).

Formatted: Indent: Left: 0", Hanging: 0.19", Bulleted + Level: 1 + Aligned at: 0" + Indent at: 0", Tab stops: Not at 0.5"

The, `TSTONEFILE` parameters describe the frequency-varying behavior of a network.

10. E-element (Voltage-Controlled Voltage Source ~~(VCVS~~)

This section explains the E-element syntax and parameters.

Formatted: Heading 2

Linear

```
Exxx n+ n- [VCVS] in+ in- gain
```

Formatted: Font: Italic

Formatted: Font: Italic

For a description of these parameters, see table VCVS Parameters.

Laplace Transform

Formatted: Heading 3

Voltage Gain H(s):

```
Exxx n+ n- LAPLACE in+ in-    k0, k1, ..., kn / d0,
    d1, ..., dm
```

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

For a description of these parameters, see table VCVS Parameters.

H(s) is a rational function~~, in the following form:You can use~~, with parameters used ~~parameters~~ to define the values of all coefficients (k_0, k_1, ..., d_0, d_1, ...).

Pole-Zero Function

Formatted: Heading 3

Voltage Gain H(s):

```
Exxx n+ n- POLE in+ in- a az1, fz1, ..., azn, fzn / b,
+ ap1, fp1, ..., apm, fpm
```

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

For a description of these parameters, see table VCVS Parameters.

The following equation defines H(s) in terms of poles and zeros:

$$H(s) = \frac{a \cdot (s + \alpha_{z1} - j2\pi f_{z1}) \ldots (s + \alpha_{zn} - j2\pi f_{zn})(s + \alpha_{zn} + j2\pi f_{zn})}{b \cdot (s + \alpha_{p1} - j2\pi f_{p1}) \ldots (s + \alpha_{pm} - j2\pi f_{pm})(s + \alpha_{pm} + j2\pi f_{pm})}$$

The complex poles or zeros are in conjugate pairs. The element description specifies only one of them, and the program includes the conjugate. ~~You can use p~~Parameters may be used to specify the a, b, α, and f values.

Example

```
Elow_pass out 0 POLE in 0 1.0 / 1.0, 1.0,0.0 0.5,0.1379
```

Formatted: Code, Space After: 0 pt, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.49" + 2.94" + 3.39" + 3.85" + 4.29" + 4.74" + 5.19"

The `Elow_pass` statement describes a low-pass filter, with the transfer function:

$$H(s) = \frac{1.0}{1.0 \cdot (s+1)(s+0.5+j2\pi \cdot 0.1379)(s+0.5-(j2\pi \cdot 0.1379))}$$

~~Table 5~~

~~A~~

Foster Pole-Residue Form

Gain E(s) form

```
Exxx n+ n- FOSTER in+ in- k0 k1
+ (Re{A1}, Im{A1})/ (Re{p1}, Im{p1})
+ (Re{A2}, Im{A2})/ (Re{p2}, Im{p2})
+ (Re{A3}, Im{A3})/ (Re{p3}, Im{p3})
+ ...
```

For a description of these parameters, see table VCVS Parameters.

In the above syntax, parenthesis , commas, and slashes are separators—they have the same meaning as a space. A pole-residue pair is represented by four numbers (real and imaginary part of the residue, then real and imaginary part of the pole).

You must make sure that Re[pi]<0; otherwise, the simulations will certainly diverge. Also, it is a good idea to assure passivity of the model (for an N-port admittance matrix Y, Re{Y} should be positive-definite), or the simulation is likely to diverge).

~~Table 10~~ **Note:**

For real poles, half the residue value is entered because it is applied twice. In the above example, the first pole-residue pair is real, but is written as "A1/(s-p1)+A1/(s-p1)"; therefore, 0.0004 is entered rather than 0.0008.

Formatted: Indent: Left: 1", No bullets or numbering

Formatted: BulletPrev

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

Table VCVS Parameters.

E-element Parameters

The E-element parameters are described in the following list.

~~Parameter~~Token	Description
Exxx	Voltage-controlled element name. Must begin with E, followed by up to 1023 alphanumeric characters.
gain	Voltage gain.
in +/-	Positive or negative controlling nodes. Specify one pair for each dimension.
K	Ideal transformer turn ratio: $V(in+,in-) = k \cdot V(n+,n-)$ or, number of gates input.
n+/-	Positive or negative node of a controlled element.
VCVS	~~Keyword~~ Pre-defined token for a voltage-controlled voltage source. VCVS is a reserved word; do not use it as a node name.

11. ~~Current-Dependent Current Sources — ~~F-element (Current-Controlled Current Source)~~s~~

Formatted: Heading 2

This section explains the F-element syntax and parameters.

~~Table 10~~ **Note:**

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

G-elements with algebraic expressions ~~make F-elements obsolete~~may be used to duplicate the functions of an F-element~~. You can still use F-elements for backward-compatibility with existing designs~~.

~~Current-Controlled Current Source (CCCS) Syntax~~

Formatted: NotePara

~~Linear~~Syntax

```
Fxxx n+ n- [<CCCS]> vn1 gain
```

Formatted: Font: Italic

Formatted: Font: Italic

F-element Parameters

The F-element parameters are described in the following list.

~~Parameter~~Token	Description
CCCS	~~Keyword~~ Pre-defined token for current-controlled current source. CCCS is a IBIS-ISS reserved ~~key~~word; do not use it as a node name.
Fxxx	Element name of the current-controlled current source. Must begin with F, followed by up to 1023 alphanumeric characters.
gain	Current gain.
n+/-	Connecting nodes for a positive or negative controlled source.
vn1 …	Names of voltage sources, through which the controlling current flows. Specify one name for each dimension.
x1,...	Controlling current, through the *vn1* source. Specify the *x* values in increasing order.

Formatted: Font: (Default) Helvetica

y1,...	Corresponding output current values of *x*.

Formatted: Heading 2

12. ~~Voltage-Dependent Current Sources —~~ G-element (Voltage-Controlled Current Source)~~s~~

This section explains G-element syntax statements, and their parameters.

~~Gxxx n+ n- <VCCS| > in+ in- ...~~

Formatted: Heading 3, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.5" + 2.94" + 3.39" + 3.85" + 4.29" + 4.75" + 5.19"

Formatted: Heading 3

~~Voltage-Controlled Current Source (VCCS)~~

Linear

```
Gxxx n+ n- [<VCCS]> in+ in- transconductance
```

Formatted: Font: Italic

Formatted: Font: Italic

For a description of the G-element parameters, see Table VCCS Parameters.

Formatted: Indent: Left: 0"

Laplace Transform

Formatted: Heading 3

Transconductance H(s):

```
Gxxx n+ n- LAPLACE in+ in-   k0, k1, ..., kn / d0,
    d1, ..., dm
```

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

H(s) is a rational function, in the following form:

$$H(s) = \frac{k_0 + k_1 s + \ldots + k_n s^n}{d_0 + d_1 s + \ldots + d_m s^m}$$

~~You can use p~~Parameters may be used to define the values of all coefficients (k_0, k_1, ..., d_0, d_1, ...).

Pole-Zero Function

Formatted: Heading 3

Transconductance H(s):

```
Gxxx n+ n- POLE in+ in- a az1, fz1, ..., azn, fzn / b,
+ ap1, fp1, ..., apm, fpm
```

Formatted: Font: Italic

Formatted: Font: Italic

The following equation defines H(s) in terms of poles and zeros:

$$H(s) = \frac{a \cdot (s + \alpha_{z1} - j2\pi f_{z1})\ldots(s + \alpha_{zn} - j2\pi f_{zn})(s + \alpha_{zn} + j2\pi f_{zn})}{b \cdot (s + \alpha_{p1} - j2\pi f_{p1})\ldots(s + \alpha_{pm} - j2\pi f_{pm})(s + \alpha_{pm} + j2\pi f_{pm})}$$

The complex poles or zeros are in conjugate pairs. The element description specifies only one of them, and the program includes the conjugate. You can use parameters to specify the a, b, α, and f values.

For a description of the G-element parameters, see table VCVS Parameters.

Example

```
Ghigh_pass 0 out POLE in 0 1.0 0.0,0.0 / 1.0 0.001,0.0
```

Formatted: Code, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.49" + 2.94" + 3.39" + 3.85" + 4.29" + 4.74" + 5.19"

The `Ghigh_pass` statement describes a high-pass filter, with the transfer function:

$$H(s) = \frac{1.0 \cdot (s + 0.0 + j \cdot 0.0)}{1.0 \cdot (s + 0.001 + j \cdot 0.0)}$$

Foster Pole-Residue Form

Formatted: Heading 3

Transconductance G(s) form

```
Gxxx n+ n- FOSTER in+ in- k0 k1
+ (Re{A1}, Im{A1})/ (Re{p1}, Im{p1})
+ (Re{A2}, Im{A2})/ (Re{p2}, Im{p2})
+ (Re{A3}, Im{A3})/ (Re{p3}, Im{p3})
+ ...
```

In the above syntax, parenthesis , commas, and slashes are separators—they have the same meaning as a space. A pole-residue pair is represented by four numbers (real and imaginary part of the residue, then real and imaginary part of the pole).

You must make sure that Re[pi]<0; otherwise, the simulations will certainly diverge. Also, it is a good idea to assure passivity of the model (for an N-port admittance matrix Y, Re{Y} should be positive-definite), or the simulation is likely to diverge).

For a description of the G-element parameters, see table VCVS Parameters.

Example

To represent a G(s) in the form,

$$G(s) = 0.001 + 1 \times 10^{-12}s + \frac{0.0008}{s + 1 \times 10^{10}} + \frac{(0.001 - j0.006)}{s - (-1 \times 10^{8} + j1.8 \times 10^{10})} + \frac{(0.001 + j0.006)}{s - (-1 \times 10^{8} - j1.8 \times 10^{10})}$$

~~You would input:~~The IBIS-ISS syntax would be

```
G1 1 0 FOSTER 2 0 0.001 1e-12
+(0.0004, 0)/(-1e10, 0) (0.001, -0.006)/(-1e8, 1.8e10)
```

Formatted: Code, Tab stops: Not at 1.14" + 1.6" + 2.04" + 2.49" + 2.94" + 3.39" + 3.85" + 4.29" + 4.74" + 5.19"

~~Table 10~~ **Note:**

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

For real poles, half the residue value is entered because it is applied twice. In the above example, the first pole-residue pair is real, but is written as “A1/(s-p1)+A1/(s-p1)”; therefore, 0.0004 is entered rather than 0.0008.

~~Table VCCS Parameters.~~

G-element Parameters

The G-element parameters described in the following list.

~~Parameter~~Token	Description
Gxxx	Name of the voltage-controlled element. Must begin with G, followed by up to 1023 alphanumeric characters.
in +/-	Positive or negative controlling nodes. Specify one pair for each dimension.
n+/-	Positive or negative node of the controlled element.
transconductance	Voltage-to-current conversion factor.
VCCS	~~Keyword~~ Pre-defined token for the voltage-controlled current source. VCCS is a reserved IBIS-ISS ~~key~~word; do not use it as a node name.
x1,...	Controlling voltage, across the *in*+ and *in*- nodes. Specify the *x* values in increasing order.
y1,...	Corresponding element values of *x*.

Formatted: Indent: Left: 0"

13. Current-Dependent Voltage Sources — H-element (Current-Controlled Voltage Source)s

This section explains H-element syntax statements, and defines their parameters.

Table 10 **Note:**

- The E-elements with algebraic expressions s makemay be used to duplicate the function of the H-elements obsolete. You can still use H-elements for backward compatibility with existing designs.

Current-Controlled Voltage Source (CCVS)

LinearSyntax

```
Hxxx n+ n- <[CCVS>] vn1 transresistance
```

ParameterToken	Description
CCVS	Pre-defined token Keyword for the current-controlled voltage source. CCVS is a IBIS-ISS reserved keyword; do not use it as a node name.
Hxxx	Element name of current-controlled voltage source. Must start with H, followed by up to 1023 alphanumeric characters.
n+/-	Connecting nodes for positive or negative controlled source.
transresistance	Current-to-voltage conversion factor.
vn1 …	Names of voltage sources, through which controlling current flows. You must specify one name for each dimension.
x1,...	Controlling current, through the *vn1* source. Specify the *x* values in increasing order.
y1,...	Corresponding output voltage values of *x*.

Formatted: Heading 2

Formatted: Indent: Left: 0.69", Hanging: 0.31", Bulleted + Level: 1 + Aligned at: 0.69" + Indent at: 0.69"

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Indent: Left: 0"

Page 7: [1] Formatted mmirmak

6/27/2010 3:44:00 AM

List Paragraph, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"