IBIS Interconnect SPICE:
A Proposal for SPICE Standardization Using Commonly-Used Syntax and Documentation

Rev 1.1 March 3, 2009

IBIS Open Forum
Advanced Technology Modeling Task Group
September, 2008

Proposal

The IBIS Open Forum, in order to enable easier data exchange between users of signal/power integrity simulation and physical layout/routing software tools, proposes creation of a generic netlist format, to be called “IBIS Interconnect SPICE.”

This format would be 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 Interconnect SPICE would be the first industry-standard version of SPICE.

Goals and Scope

The syntax of IBIS Interconnect SPICE 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

To these ends, IBIS Interconnect SPICE will include 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
- independent and dependent sources
- string-based node naming
- user-defined comments

IBIS Interconnect SPICE will NOT include or cover:

- model formats or “process cards” for active devices (e.g., diodes, transistors)
- 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 (except as supported through P1735)

Assumptions

IBIS Interconnect SPICE would form a universal starting point for netlist and/or interconnect descriptions. Individual implementations would include the areas related to

simulation engine control, etc. described above and could conceivably define additional elements without conflict with IBIS Interconnect SPICE.

Because the syntax for IBIS Interconnect SPICE represents a subset of at least one common proprietary SPICE implementation, the IBIS Open Forum may solicit donation of existing syntax and accompanying manuals from the appropriate vendor(s).

Precedents for such an approach are listed below.

SystemVerilog:
http://www.eda.org/sv-ec/SV_3.1_Web/Minutes/SV-EC-Minutes-2002-September-4.txt

DEF and OVI:
http://findarticles.com/p/articles/mi_m0EKF/is_n2082_v41/ai_17465901

Verilog, SystemC, SystemVerilog:
http://www.edn.com/index.asp?layout=article&articleid=CA376625

Rules

The syntax rules detailed below provide a rough outline of the format and usage of IBIS Interconnect SPICE basic features. Note that these are edited together from several Internet-available documents outlining the syntax of SPICE variants available to the industry.

Comments and Line Continuation

The first line of a netlist is always a comment, regardless of its first character; comments that are not the first line of the netlist 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>
```

or

```
<IBIS Interconnect SPICE statement> $ <comment following input>
```

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:

- Whitespace
- Comma (,)
- Valid numeric expression

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
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
+ LEVEL=3
```

Using Parameters in Simulation (.PARAM)

Parameters are names associated with numeric values. Algebraic expressions may not (??) include node names in IBIS Interconnect SPICE.

Simple assignment:

```
.PARAM <SimpleParam>=1e-12
```

Algebraic definition:

```
 .PARAM <AlgebraicParam>='SimpleParam*8.2'
```

SimpleParam excludes the output variable.

Subcircuit default:

```
 .SUBCKT <SubName> <ParamDefName>=<Value> str('string')
```

An algebraic parameter (equation) is an algebraic expression of real values, a predefined or user-defined function or circuit or model values. Enclose a complex expression in single quotes to invoke the algebraic processor, *unless* the expression begins with an alphabetic character and contains no spaces. A simple expression consists of a single parameter name.

Built-In Functions and Variables

The following functions will be supported by IBIS Interconnect SPICE:

Simple arithmetic operations: +, -, *, / (**,^)
sin(x)
cos(x)
tan(x)
asin(x)
acos(x)
atan(x)
sinh(x)
cosh(x)
tanh(x)
abs(x)
sqrt(x)
pow(x,y)
pwr(x,y)
log(x)
log10(x)
exp(x)
db(x)
int(x)
nint(x)
sgn(x)
sign(x,y)
min(x,y)
max(x,y)

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 999999999999999 (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 assign the first name, the (.) extension is used to concatenate the circuit path name with the node name. For example:

```
X1.XBIAS.M5
```

Node designations that start with the same number, followed by any letter, are the same node. For example, 1c and 1d are the same node.

The second subcircuit node name is a unique number that is assigned to an input netlist subcircuit. The (.) extension concatenates this number with the internal node name, to form the entire subcircuit's node name (for example, `10.M5`). The output listing file cross-references the node name.

To indicate the ground node, use either the number 0, the name `GND`, or `!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). Floating power supply nodes are terminated with a 1 megohm resistor and a warning message.

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.

Elements

Shunt Element

V*xxx n1 n2* < DC => *0*
(Value must always be 0)
.connect n1 n2

Resistor

R*xxx n1 n2* < R = > *resistance*
Somewhere need a blanket statement that resistance, … is a constant (number, parameter or expression of numbers and parameters)

Capacitor

C*xxx n1 n2* < C = > *capacitance*

Inductor

L*xxx n1 n2* < L = > *inductance*

Mutual Inductor

K*xxx* L*yyy* L*zzz* < K = > *coupling*

T-element (Ideal Transmission Lines)

General form:
T*xxx in refin out refout Z0=val TD=val*

Z0 Characteristic impedance of the transmission line.
Z0 Zo (allow both letter Oh and number zero).
TD Signal delay from a transmission line, in seconds
(Assumes L=1)
? refin == refout

Source Elements (E, F, G, H)

The following capabilities for controlled sources are supported (only E element shown below, not all element support all of these options):

Linear

Exxx n+ n- <VCVS> in+ in- gain

Delay Element

Exxx n+ n- <VCVS> DELAY in+ in- TD=val

Laplace Transform

Voltage Gain H(s):

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

Pole-Zero Function

Voltage Gain H(s):

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

?Frequency Table Format?

All four not available for all four types.

Purgatory

Frequency Response Table

Voltage Gain H(s):

```
Exxx n+ n- FREQ in+ in- f1, a1, f1, ..., fi, ai, f1
```

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})
+ ...
```

.SUBCKT

Defines a subcircuit in a netlist.

Syntax

.SUBCKT subnam n1 <n2 n3 ...> <parnam=val>
.ENDS

Argument Description

subnam Specifies a reference name for the subcircuit model call.
n1 ... Node numbers for external reference; cannot be the ground node (0, gnd, ground, gnd!). Any element nodes that are in the subcircuit, but are not in this list, are strictly local with three exceptions:

- ground node (0, gnd, ground, gnd!).
- parnam A parameter name set to a value. Use only in the subcircuit. To override this value, assign it in the subcircuit call or set a value in a .PARAM statement.
- SubDefaultsList <*SubParam1*>=<*Expression*> [<*SubParam2*>=<*Expression*>...]

Subcircuits

X<*subcircuit_name*> adds an instance of a subcircuit to a netlist. The subcircuit must already have been defined in the netlist by using .SUBCKT

Syntax

X*<subcircuit_name> n1* <*n2 n3* …> *subnam* <*parnam* = *val* >

S-element Syntax

Use the following S-element syntax to show the connections within a circuit:

```
Sxxx nd1 nd2 ... ndN ndRef
+ <MNAME=Smodel_name>
+ <TYPE=[s|y]>
+ <FBASE = base_frequency> <FMAX=maximum_frequency>
```

S Model Syntax

Use the following syntax to describe specific S models:

```
.MODEL Smodel_name S
+ <N=dimension>
+ TSTONEFILE=filename
+ <TYPE=[s|y]>
+ <FBASE=base_frequency> <FMAX=maximum_frequency>
```

```
??? Sxxx nd1+ nd1- nd2+ nd2-    (N=2)

Sxxx nd1+ nd1- nd2+ nd2- 0    (N=4)
```

Input Syntax for the W-element

Syntax:

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

(s – parameter format)

(Geometric model is excluded)

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

Parameter Description

N	Number of conductors (same as in the element card).
L	DC inductance matrix, per unit length
C	DC capacitance matrix, per unit length
Ro	DC resistance matrix, per unit length
Go	DC shunt conductance matrix, per unit length
Rs	Skin effect resistance matrix, per unit length
Gd	Dielectric loss conductance matrix, per unit length

{RLGCFILE format assumed supported but not described here}
(Deprecated?)

Table Model Card Syntax

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

Parameter Description-

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.

{Table Model format includes frequency values for each matrix}

Interpolation

SP model should be simplified for W line models.