**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 165.1\_draft1

**ISSUE TITLE:** Parameter Passing Improvements for [External Circuit]s

**REQUESTOR:**  Arpad Muranyi & John Angulo, Mentor Graphics; Ambrish Varma & Brad Brim, Cadence Design Systems, Inc

**DATE SUBMITTED:** January 9, 2014

**DATE REVISED:** October 26, 2017

**DATE ACCEPTED BY IBIS OPEN FORUM:**

**STATEMENT OF THE ISSUE:**

The IBIS 6.0 specification introduced an elaborate parameter passing mechanism for the [External Model] and [External Circuit] keywords through the “Parameters” and “Converter\_Parameters” subparameters. This mechanism, however, passes the same parameter value(s) to every instance of [External Model] or [External Circuit]. Passing parameters independently to each instance of an [External Model] or [External Circuit] would provide much greater flexibility and efficiency for IBIS models.

This BIRD defines a mechanism through which parameters can be passed independently to each instance of an [External Circuit]. This is achieved by making the “Parameters” and “Converter\_Parameters” subparameters also available under the [Circuit Call] keyword.

While it would be desirable to have a similar capability for [External Model]s as well, this BIRD does not cover the [External Model] keyword. The reason for that is because the [External Model] keyword is located under the [Model] keyword, and as a result it is instantiated by the [Pin] keyword. Extending the [Pin] keyword with a parameter passing syntax would make things too complicated at this time.

**ANALYSIS PATH/DATA THAT LED TO SPECIFICATION:**

**ANY OTHER BACKGROUND INFORMATION:**

BIRD165.1 was updated to use page number references from the v6.1 IBIS specification, and an example was also added. In this version of the BIRD, the green underlined/strike through text indicates changes with respect to the IBIS specification, and the tracking indicates changes with respect to BIRD165.

Add the two new subparameters to [Circuit Call] in the keyword tree on pg. 11.

On pg. 93, modify the following paragraph as indicated:

The [Circuit Call] keyword acts similarly to subcircuit calls in SPICE, instantiating and connecting the various [External Circuit]s and optionally passing parameters to them~~connecting them together~~. Please note that models described by the [External Model] keyword are connected according to the rules and assumptions of the [Model] keyword. [Circuit Call] is not necessary for these cases and must not be used.

Modify pg. 130-131 as follows:

*Keywords:* [Circuit Call], [End Circuit Call]

*Required:* Yes, if any [External Circuit]s are present in a [Component].

*Description:* This keyword is used to instantiate [External Circuit]s and to connect their ports to the die nodes or die pads.

*Sub-Params:* Signal\_pin, Diff\_signal\_pins, Series\_pins, Parameters, Converter\_Parameters, Port\_map

*Usage Rules:* The [Circuit Call] keyword must be followed by the name of an [External Circuit] that exists in the same [Component].

When a [Circuit Call] keyword defines any connections that involve one or more die pads (and consequently pins), the corresponding pins on the [Pin] list must use the reserved word “CIRCUITCALL” in the third column instead of a model name.

Each [External Circuit] must have at least one corresponding [Circuit Call] keyword. Multiple [Circuit Call] keywords may appear under a [Component] using the same [External Circuit] name, if multiple instantiations of an [External Circuit] are needed.

Signal\_pin, Diff\_signal\_pins, or Series\_pins:

The purpose of these subparameters is to identify which [External Circuit] needs to be stimulated in order to obtain a signal on a certain pin. These subparameters must be used only when the [External Circuit] that is referenced by the [Circuit Call] keyword makes use of the stimulus signal of the simulator. Any given [Circuit Call] keyword must contain no more than one instance of only one of these three subparameters. The subparameter is followed by one or two pin names which must be defined by the [Pin] keyword.

Signal\_pin is used when the referenced [External Circuit] has a single analog signal port (I/O) connection to one pin. The subparameter is followed by a pin name that must match one of the pin names under the [Pin] keyword.

Diff\_signal\_pins is used when the referenced [External Circuit] describes a true differential model which has two analog signal port (I/O) connections, each to a separate pin. The subparameter is followed by two pin names, each of which must match one of the pin names under the [Pin] keyword. The first and second pin names correspond to the non-inverting and inverting signals of the differential model, respectively. The two pin names must not be identical.

Series\_pins is used when the referenced [External Circuit] describes a Series or Series\_switch model which has two analog signal port (I/O) connections to two pins. The subparameter is followed by two pin names, each of which must match one of the pin names under the [Pin] keyword. The first and second pin names correspond to the positive and negative ports of the Series or Series\_switch model, respectively. However, the polarity order matters only when the model is polarity sensitive (as with the [Series Current] keyword). The two pin names must not be identical.

Parameters:

Lists names of parameters that may be passed into a specific instance of an external circuit file. The rules for this subparameter are the same as the rules for the corresponding Parameters subparameter under the [External Circuit] keyword, except that if a parameter assignment exists for the same parameter name under both the [Circuit Call] and the [External Circuit] keywords, the parameter assignment under the [Circuit Call] keyword will be in effect.

Converter\_Parameters:

This optional subparameter lists and initializes parameter names to be used as arguments in the A\_to\_D and/or D\_to\_A converter(s) of a specific instance of an [External Circuit]. The rules for this subparameter are the same as the rules for the corresponding Converter\_Parameters subparameter under the [External Circuit] keyword, except that if a parameter assignment exists for the same parameter name under both the [Circuit Call] and the [External Circuit] keywords, the parameter assignment under the [Circuit Call] keyword will be in effect.

Port\_map:

The Port\_map subparameter is used to connect the ports of an [External Circuit] to die nodes or die pads.

Every occurrence of the Port\_map subparameter must begin on a new line and must be followed by two arguments, the first being a port name, and the second being a die node, die pad, or a pin name.

The first argument of Port\_map must contain a port name that matches one of the port names in the corresponding [External Circuit] definition. No port name may be listed more than once within a [Circuit Call] statement. Only those port names need to be listed with the Port\_map subparameter which are connected to a die node or a die pad. This includes reserved and/or user-defined port names.

The second argument of the Port\_map subparameter contains the name of a die node, die pad, or a pin. The names of die nodes, die pads, and pins may appear multiple times as Port\_map subparameter arguments within the same [Circuit Call] statement to signify a common connection between multiple ports, such as common voltage supply.

Please note that a pin name in the second argument does not mean that the connection is made directly to the pin. Since native IBIS does not have a mechanism to declare die pads explicitly, connections to die pads are made through their corresponding pin names (listed under the [Pin] keyword). This convention must only be used with native IBIS package models where a one-to-one path between the die pads and pins is assumed. When a package model other than native IBIS is used with a [Component], the second argument of Port\_map must have a die pad or die node name. These names are matched to the corresponding port name of the non-native package model by name (not by position). In this case, the package model may have an arbitrary circuit topology between the die pads and the pins. A one-to-one mapping is not required.

On pg. 134, change the example as shown here:

|

[Circuit Call] Die\_Interconnect | Instantiates [External Circuit] named

| "Die\_Interconnect"

|

| List of parameters

Parameters sp\_file\_name = paramfile.par(RootName(Model\_Specific(TstoneFile)))

Parameters C1\_value

Parameters R1\_value = paramfile.par(RootName(Model\_Specific(R1)))

Parameters R2\_value = 45

|

| mapping port pad/node

|

Port\_map vcc 10 | Port to implicit pad connection

Port\_map gnd pad\_11 | Port to explicit pad connection

Port\_map io1 1 | Port to implicit pad connection

Port\_map o2 pad\_2a | Port to explicit pad connection

Port\_map vcca1 a | Port to internal node connection

Port\_map vcca2 b | Port to internal node connection

Port\_map int\_ioa c | Port to internal node connection

Port\_map vssa1 d | Port to internal node connection

Port\_map vssa2 e | Port to internal node connection

Port\_map vccb1 f | Port to internal node connection

Port\_map int\_ob g | Port to internal node connection

Port\_map vssb1 h | Port to internal node connection

|

[End Circuit Call]