**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 163

**ISSUE TITLE:** Instantiating and Connecting [External Circuit] Package Models with [Circuit Call]

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

**DATE SUBMITTED:** January 9, 2014

**DATE REVISED:**

**DATE ACCEPTED:** Rejected July 20, 2018

**STATEMENT OF THE ISSUE:**

This BIRD was written to replace BIRD 145 with the expectation that BIRD 145 will be rejected when this BIRD is approved. A major shortcoming of BIRD 145 is that it requires a [Model Call] statement for each instance of every single [Model] in an IBIS file. Since [Model Call] itself consists of multiple lines, this would significantly increase the size of an IBIS file.

In addition, BIRD 145 allows for arbitrary connectivity between [Pin]s and [Model]s (including splits and/or joins in the signal path) which may pose significant challenges to some EDA tools which are architected based on the one [Model] per [Pin] hierarchy of the existing IBIS specification.

This BIRD proposes a new syntax in the Port\_map subparameter of [Circuit Call] to allow [Model]s to be cascaded with [External Circuit]s while keeping the instantiation mechanism of [Model]s on the [Pin] keyword. This makes on-die (and/or package) modeling possible with [External Circuit]s and legacy [Model]s without the need for [Model Call] while keeping the 1:1 relationship between [Pin]s and [Model]s. Note that while this BIRD preserves the existing 1:1 [Pin] to [Model] relationship of IBIS, the power supply connectivity has no such restrictions.

This BIRD is closely related to and dependent on BIRD 164, which proposes to make the [External Circuit] keyword available for package modeling purposes.

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

**ANY OTHER BACKGROUND INFORMATION:**

## Multi-Lingual Model Extensions

INTRODUCTION

The SPICE, IBIS-ISS, VHDL-AMS and Verilog-AMS languages are supported by IBIS. This chapter describes how models written in these languages can be referenced and used by .ibs files.

Table 10 shows the keywords used by the language extensions within the IBIS framework.

Table 11 – Language Extension Keywords

| **Keyword** | **Description** |
| --- | --- |
| [External Circuit]  [End External Circuit] | References enhanced descriptions of structures on the die, including digital and/or analog, active and/or passive circuits |
| [External Model]  [End External Model] | Same as [External Circuit], except limited to the connection format and usage of the [Model] keyword, with one additional feature added: support for true differential buffers |
| [Node Declarations]  [End Node Declarations] | Lists on-die connection points related to the [Circuit Call] keyword |
| [Circuit Call]  [End Circuit Call] | Instantiates [External Circuit]s and connects them to each other and/or die pads |

The placement of these keywords within the hierarchy of IBIS is shown below:

├── **[Component]**

│ │

│ ├── **[Node Declarations]**

│ │ └── **[End Node Declarations]**

│ │

│ ├── **[Circuit Call]**

│ │ └── **[End Circuit Call]**

│ │

│

├── **[Model]**

│ │

│ ├── **[External Model]**

│ │ └── **[End External Model]**

│

├── **[External Circuit]**

│ └── **[End External Circuit]**

Languages Supported:

.ibs files can reference other files which are written using the SPICE, IBIS-ISS, VHDL-AMS, or Verilog-AMS languages. In this document, these languages are defined as follows:

“SPICE” refers to SPICE 3, Version 3F5 developed by the University of California at Berkeley, California. Many vendor-specific EDA tools are compatible with most or all of this version.

"IBIS-ISS" refers to the "IBIS Interconnect SPICE Subcircuits Specification (IBIS-ISS)", developed by the members of the IBIS Open Forum.

“VHDL-AMS” refers to “IEEE Standard VHDL Analog and Mixed-Signal Extensions”, approved March 18, 1999 by the IEEE-SA Standards Board and designated IEEE Std. 1076.1-1999, or later.

“Verilog-AMS” refers to the Analog and Mixed-Signal Extensions to Verilog-HDL as documented in the Verilog-AMS Language Reference, Version 2.0, or later. This document is maintained by Accellera (formerly Open Verilog International), an independent organization. Verilog-AMS is a superset that includes Verilog-A and the Verilog Hardware Description Language IEEE 1364-2001, or later.

“VHDL-A(MS)” refers to the analog subset of VHDL-AMS described above.

“Verilog-A(MS)” refers to the analog subset of Verilog-AMS described above.

In addition, the “IEEE Standard Multivalue Logic System for VHDL Model Interoperability (Std\_logic\_1164)”, designated IEEE Std. 1164-1993 or later, is required to promote common digital data types for .ibs files referencing VHDL-AMS. Also, the Accellera Verilog-AMS Language Reference Manual Version 2.2 or later, is required to promote common digital data types for .ibs files referencing Verilog-AMS.

Note that, for the purposes of this section, keywords, subparameters and other data used without reference to the external languages just described are referred to collectively as “native” IBIS.

Overview:

The four keyword pairs discussed in this chapter can be separated into two groups based on their functionalities. The [External Model], [End External Model], [External Circuit], and [End External Circuit] keywords are used as pointers to the models described by one of the external languages. The [Node Declarations], [End Node Declarations], [Circuit Call], and [End Circuit Call] keywords are used to describe how [External Circuit]s are connected to each other, [Models], die pads and component pins.

The [External Model] and [External Circuit] keywords are very similar in that they both support the same external languages, and they can both be used to describe passive and/or active circuitry. The key difference between the two keywords is that [External Model] can only be placed under the [Model] keyword, while [External Circuit] can only be placed outside the [Model] keyword, as illustrated in the portion of the keyword hierarchy, shown above.

The intent behind [External Model] is to provide an upgrade path from native IBIS [Model]s to the external languages (one exception to this is the support for true differential buffers). Thus, the [External Model] keyword can be used to replace the usual I-V and V-T tables, C\_comp, C\_comp\_pullup, C\_comp\_pulldown, C\_comp\_power\_clamp, C\_comp\_gnd\_clamp subparameters, [Ramp], [Driver Schedule], [Submodel] keywords, etc. of a [Model] by any modeling technique that the external languages allow. For [External Model]s, the connectivity, test load and specification parameters (such as Vinh and Vinl) are preserved from the [Model] keyword and the simulator is expected to carry out the same type of connections and measurements as is usually done with the [Model] keyword. The only difference is that the model itself is described by an external language.

In the case of the [External Circuit], however, one can model a circuit having any number of ports (see definitions below). For example, the ports may include impedance or buffer strength selection controls in addition to the usual signal and supply connections. The connectivity of an [External Circuit] is defined by the [Node Declarations] and [Circuit Call] keywords. Currently, the test loads and measurement parameters for an [External Circuit] can only be defined inside the model description itself. The results of measurements can be reported to the user or tool via other means.

The [Circuit Call] keyword acts similarly to subcircuit calls in SPICE, instantiating the various [External Circuit]s and connecting them together and/or to [Model]s, die pads and component pins. For example, one [Circuit Call] may instantiate an [External Circuit] which contains a package model, while another [Circuit Call] may instantiate an [External Circuit] which contains an on-die power distribution network. Each [Circuit Call] defines how the ports of the [External Circuit] it instantiates connect to other [External Circuit] instances or to component pins, die pads or buffers. If none of the [Circuit Call]s of a component make connections to a particular [Model]’s terminal(s), that [Model] will be connected to the package at unnamed, but implicitly assumed nodes corresponding to pads on the die. A [Circuit Call] may connect the ports of an [External Circuit] to buffers by naming these [Model] terminals explicitly. This allows the use of a languge supported by [External Circuit] to model frequency dependent loss, coupling and other electrical behavior in an IBIS component that uses the [Model] keyword to describe its buffers.

Definitions:

For the purposes of this document, several general terms are defined below.

circuit - any arbitrary collection of active or passive electrical elements treated as a unit

node - any electrical connection point; (may be digital or analog; may be a connection internal to a circuit or between circuits). A node may be a connection point anywhere on the die, including die pads and die bumps which connect the die to the package.

pad, bump - a special case of a node. A pad or bump connects a buffer or other circuitry to a package; also called die pad or die bump.

port - access point in an [External Model] or [External Circuit] definition for digital or analog signals

pseudo-differential circuits - combination of two single-ended circuits which drive and/or receive complementary signals, but where no internal current relationship exists between them

true differential circuits - circuits where a current relationship exists between two outputs or inputs which drive or receive complementary signals

General Assumptions:

Ports under [Model]s:

The use of ports under native IBIS must be understood before the multi-lingual extensions can be correctly applied. The [Model] keyword assumes, but does not explicitly require, naming ports on circuits. These ports are automatically connected by IBIS-compliant tools without action by the user. For example, the [Voltage Reference] keyword implies the existence of power supply rails which are connected to the power supply ports of the circuit described by the [Model] keyword.

For multi-lingual modeling, ports must be explicitly named in the [External Model] or [External Circuit]; the ports are no longer assumed by EDA tools. To preserve compatibility with the assumptions of [Model], a list of pre-defined port names has been created where the ports are reserved with fixed functionality. These reserved ports are defined in Table 12.

Table 12 – Port Names in Multi-Lingual Modeling

| **Port** | **Name** | **Description** |
| --- | --- | --- |
| 1 | D\_drive | Digital input to a model unit |
| 2 | D\_enable | Digital enable for a model unit |
| 3 | D\_receive | Digital receive port of a model unit, based on data on A\_signal (and/or A\_signal\_pos and A\_signal\_neg) |
| 4 | A\_puref | Voltage reference port for pullup structure |
| 5 | A\_pcref | Voltage reference port for power clamp structure |
| 6 | A\_pdref | Voltage reference port for pulldown structure |
| 7 | A\_gcref | Voltage reference port for ground clamp structure |
| 8 | A\_signal | I/O signal port for a model unit |
| 9 | A\_extref | External reference voltage port |
| 10 | D\_switch | Digital input for control of a series switch model |
| 11 | A\_gnd | Global reference voltage port |
| 12 | A\_pos | Non-inverting port for series or series switch models |
| 13 | A\_neg | Inverting port for series or series switch models |
| 14 | A\_signal\_pos | Non-inverting port of a differential model |
| 15 | A\_signal\_neg | Inverting port of a differential model |

The first letter of the port name designates it as either digital (“D”) or analog (“A”). Reserved ports 1 through 13 are assumed or implied under the native IBIS [Model] keyword. Again, for multi-lingual models, these ports must be explicitly assigned by the user in the model if their functions are to be used. A\_gnd is a universal reference node, similar to SPICE ideal node “0.” Ports 14 and 15 are only available under [External Model] for support of true differential buffers.

Under the [Model] description, power and ground reference ports are created and connected by IBIS-compliant tools as defined by the [Power Clamp Reference], [GND Clamp Reference], [Pullup Reference], [Pulldown Reference] and/or [Voltage Range] keywords. The A\_signal port is connected to the die pad, to drive or receive an analog signal.

Ports under [External Model]s:

The [External Model] keyword may only appear under the [Model] keyword and it may only use the same ports as assumed with the native IBIS [Model] keyword. However, [External Model] requires that reserved ports be explicitly declared in the referenced language(s)

For [External Model], reserved analog ports are connected the same way as the terminals of the [Model] keyword. These ports are connected to the component pins through [Package Model]s or [Pin] parasitics or to [External Circuit]s when the [Circuit Call] keyword contains references to [Model] terminal names. Digital ports under [External Model] would connect to other internal digital circuitry.

Two standard [Model] structures—an I/O buffer and a Series Switch—are shown, with their associated port names, in Figure 19 and Figure 20.



2. - Port Names for I/O Buffer



1. - Port Names for Series Switch

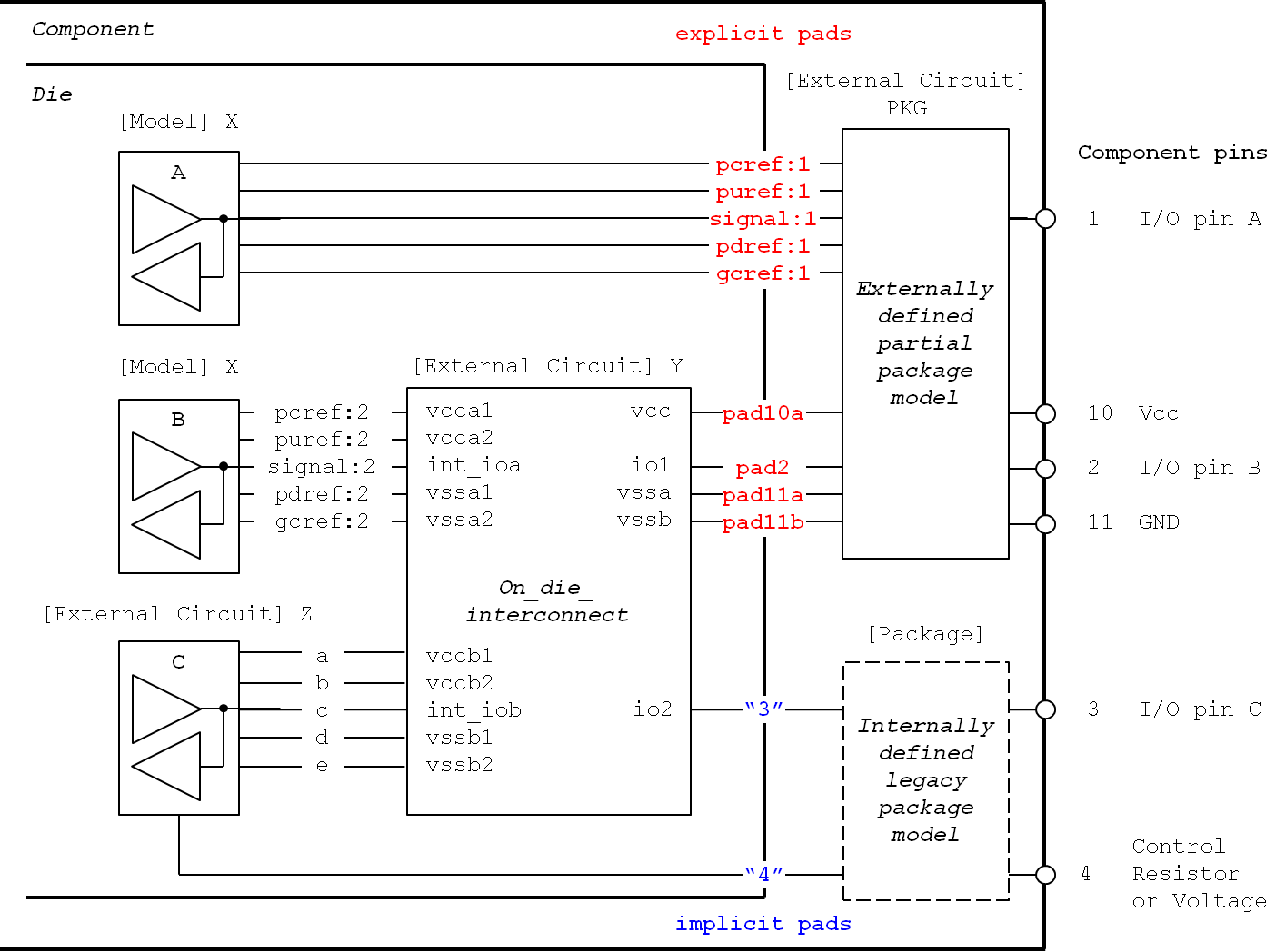
Ports under [External Circuit]s:

The [External Circuit] keyword allows the user to define any number of ports and port functions on a circuit. The [Circuit Call] keyword instantiates [External Circuit]s and connects their ports to specific die nodes (which may include any die pads or die bumps) and/or component pins. Note that, if reserved digital port names are used with an [External Circuit], those ports will be connected automatically as defined in the port list above (under [External Circuit], reserved analog port names do not retain particular meanings).

Figure 21 illustrates the use of [External Circuit]. Buffer “A” and “B” are two instances of [Model] “X”. Similarly, Buffer “C” is an instance of [External Circuit] “Z”. The instances of [Model] “X” are created by the [Pin] keyword, and the instance of buffer “C” is created through a [Circuit Call]. [External Circuit] “Y” defines an on-die interconnect circuit. Nodes “a” through “e” are specific instances of the ports defined for [External Circuit] “Z”. These ports become the internal nodes of the die and must be explicitly declared with the [Node Declarations] keyword. The “On-die Interconnect” [Circuit Call] creates an instance of the [External Circuit] “Y” and connects the instance with die pads, die nodes and [Model] instance terminals. The “C” [Circuit Call] connects the individual ports of the buffer instance to the “On\_die\_interconnect” [External Circuit] through die nodes “a” - “e”.

Note that the “Analog Buffer Control” signal of buffer “C” is connected directly to the implicit pad for pin 4. This connection is also made through an entry under the [Circuit Call] keyword.

[External Circuit] “PKG” defines a package interconnect circuit between die pads, [Model] instance terminals and component pins. The [Circuit Call] instantiating this [External Circuit] makes the connections between its ports and the appropriate die pads, [Model] instance terminals and component pins. Even though buffers “A” and “B” are instances of [Model] “X” and are declared by the [Pin] keyword, their connections to the package model in [External Circuit] “PKG” are defined by the [Circuit Call] instantiating “PKG”.



1. - Example Showing [External Circuit] Ports

The [Model], [External Model] and [External Circuit] keywords (with [Circuit Call]s and [Node Declarations] as appropriate) may be combined together in the same .ibs file or even within the same [Component] description.

Port types and states:

The intent of native IBIS is to model the circuit block between the region where analog signals are of interest, and the digital logic domain internal to the component. For the purposes of this discussion, the IBIS circuit block is called a “model unit” in Figure 22 and Figure 23 and the document text below.

The multi-lingual modeling extensions maintain and expand this approach, assuming that both digital signals and/or analog signals can move to and from the model unit. All VHDL-AMS and Verilog-AMS models, therefore, must have digital ports and analog ports. In certain cases, digital ports may not be required, as in the case of interconnects; see [External Circuit] below. Routines to convert signals from one format to the other are the responsibility of the model author.

Digital ports under AMS languages must follow certain constraints on type and state. In VHDL-AMS models, analog ports must have type “electrical”. Digital ports must have type “std\_logic” as defined in IEEE Standard Multivalue Logic System for VHDL Model Interoperability (Std\_logic\_1164), or later. In Verilog-AMS models, analog ports must be of discipline “electrical” or a subdiscipline thereof. Digital ports must be of discipline “logic” as defined in the Accellera Verilog-AMS Language Reference Manual Version 2.2, or later and be constrained to states as defined in IEEE Std. 1164-1993, or later.

The digital ports delivering signals to the AMS model, D\_drive, D\_enable, and D\_switch, must be limited to the ‘1’ or ‘0’ states for VHDL-AMS, or, equivalently, to the 1 or 0 states for Verilog-AMS. The D\_receive digital port may only have the ‘1’, ‘0’, or ‘X’ states in VHDL-AMS, or, equivalently, the 1, 0, or X states in Verilog-AMS. All digital ports other than the foregoing predefined ports may use any of the logic states allowed by IEEE Std. 1164-1993, or later.

SPICE, IBIS-ISS, VHDL-A(MS), Verilog-A(MS) versus VHDL-AMS and VERILOG-AMS:

SPICE, IBIS-ISS, VHDL-A(MS), Verilog-A(MS) cannot process digital signals. All SPICE, IBIS-ISS, VHDL-A(MS), Verilog-A(MS) input and output signals must be in analog format. Consequently, IBIS multi-lingual models using SPICE, IBIS-ISS, VHDL-A(MS) or Verilog-A(MS) require analog-to-digital (A\_to\_D) and/or digital-to-analog (D\_to\_A) converters to be provided by the EDA tool. The converter subparameters are declared by the user, as part of the [External Model] or [External Circuit] syntax, with user-defined names for the ports which connect the converters to the analog ports of the SPICE, IBIS-ISS, VHDL-A(MS), or Verilog-A(MS) model. The details behind these declarations are explained in the keyword definitions below.

The electrical output characteristics of D\_to\_A converters are equivalent to ideal voltage sources having a zero ohm output impedance, and the electrical input characteristics of A\_to\_D converters are equivalent to ideal voltage probes, having an infinite input impedance.

To summarize, Verilog-AMS and VHDL-AMS contain all the capability needed to ensure that a model unit consists of only digital ports and/or analog ports. SPICE, IBIS-ISS, VHDL-A(MS) and Verilog-A(MS), however, need extra data conversion, provided by the EDA tool, to ensure that any digital signals can be correctly processed.



1. - AMS Model Unit, Using an I/O Buffer as an Example



1. - An Analog-Only Model Unit, Using an I/O Buffer as an Example

KEYWORD DEFINITIONS

*Keywords:* [External Model], [End External Model]

*Required:* No

*Description:* Used to reference an external file written in one of the supported languages containing an arbitrary circuit definition, but having ports that are compatible with the [Model] keyword, or having ports that are compatible with the [Model] keyword plus an additional signal port for true differential buffers.

*Sub-Params:* Language, Corner, Parameters, Converter\_Parameters, Ports, D\_to\_A, A\_to\_D

*Usage Rules:* The [External Model] keyword must be positioned within a [Model] section and it may only appear once for each [Model] keyword in a .ibs file. It is not permitted under the [Submodel] keyword.

[Circuit Call] may not be used to connect an [External Model].

A native IBIS [Model]’s data may be incomplete if the [Model] correctly references an [External Model]. Any native IBIS keywords that are used in such a case must contain syntactically correct data and subparameters according to native IBIS rules. In all cases, [Model]s which reference [External Model]s must include the following keywords and subparameters:

Model\_type

Vinh, Vinl (as appropriate to Model\_type)

[Voltage Range] and/or [Pullup Reference], [Pulldown Reference], [POWER Clamp Reference], [GND Clamp Reference], [External Reference]

[Ramp]

In models without the [External Model] keyword, data for [Ramp] should be measured using a load that conforms to the recommendations in Section 9, "NOTES ON DATA DERIVATION METHOD". However, when used within the scope of [External Model], the [Ramp] keyword is intended strictly to provide EDA tools with a quick first-order estimate of driver switching characteristics. When using [External Model], therefore, data for [Ramp] may be measured using a different load, if it results in data that better represent the driver’s behavior in standard operation. Also in this case, the R\_load subparameter is optional, regardless of its value, and will be ignored by EDA simulators. For example, the 20% to 80% voltage and time intervals for a differential buffer may be measured using the typical differential operating load appropriate to that buffer’s technology. Note that voltage and time intervals must always be recorded explicitly rather than as a reduced fraction, in accordance with [Ramp] usage rules.

The following keywords and subparameters may be omitted, regardless of Model\_type, from a [Model] using [External Model]:

C\_comp, C\_comp\_pullup, C\_comp\_pulldown, C\_comp\_power\_clamp, C\_comp\_gnd\_clamp

[Pulldown], [Pullup], [POWER Clamp], [GND Clamp]

Subparameter Definitions:

Language:

Accepts “SPICE”, “IBIS-ISS”, “VHDL-AMS”, “Verilog-AMS”, “VHDL-A(MS)” or “Verilog-A(MS)” as arguments. The Language subparameter is required and must appear only once.

Corner:

Three entries follow the Corner subparameter on each line:

corner\_name file\_name circuit\_name

The corner\_name entry is “Typ”, “Min”, or “Max”. The file\_name entry points to the referenced file in the same directory as the .ibs file.

Up to three Corner lines are permitted. A “Typ” line is required. If “Min” and/or “Max” data is missing, the tool may use “Typ” data in its place. However, the tool should notify the user of this action.

Models instantiated by corner\_name "Min" describe slow, weak performance, and models instantiated by corner\_name "Max" describe fast, strong performance.

The circuit\_name entry provides the name of the circuit to be simulated within the referenced file. For SPICE and IBIS-ISS files, this is normally a “.subckt” name. For VHDL-AMS files, this is normally an “entity(architecture)” name pair. For Verilog-AMS files, this is normally a “module” name.

No character limits, case-sensitivity limits or extension conventions are required or enforced for file\_name and circuit\_name entries. However, the total number of characters in each Corner line must comply with the rules in Section 3. Furthermore, lower-case file\_name entries are recommended to avoid possible conflicts with file naming conventions under different operating systems. Case differences between otherwise identical file\_name entries or circuit\_name entries should be avoided. External languages may not support case-sensitive distinctions.

Parameters:

Lists names of parameters that can be passed into an external model file. Each Parameters entry must match a name or keyword in the external file or language. The list of Parameters may span several lines by using the word Parameters at the start of each line. The Parameters subparameter is optional, and the external model must operate with default settings without any Parameters assignments.

Parameter passing is not supported in SPICE. VHDL-AMS and VHDL-A(MS) parameters are supported using “generic” names, and Verilog-AMS and Verilog-A(MS) parameters are supported using “parameter” names. IBIS-ISS parameters are supported for all IBIS-ISS parameters which are defined on the subcircuit definition line.

Parameters are locally scoped under each [External Model] keyword, i.e., the same parameter under two different [External Model] will have independent values.

The parameter(s) listed under the Parameters subparameter may optionally be followed by an equal sign and a numeric, Boolean or string literal or a reference to a parameter name which is located in a parameter tree. The reference must begin with a file name, followed by an open parentheses and a the tree root name, a new open parentheses for any branch names (including the Reserved\_Parameters or Model\_Specific branch names if present in the tree) and the parameter name, and a matching set of closing parentheses. The file reference may point to any file which contains one or more parameter trees. The files referenced must be located in the same directory as the .ibs file containing the reference. The file names of parameter files must follow the rules for file names given in Section 3, “GENERAL SYNTAX RULES AND GUIDELINES”. Parameter files may only contain parameter trees using the tree syntax described in IBIS in Section 10.3 with the following exceptions and additions:

When the extension of the external parameter’s file name ends with “.ami”:

a) only Usage In or Usage Info are allowed for parameters which are to be passed into models instantiated by the [External Model] or the [External Circuit] keywords

When the extension of the external parameter’s file name does not end with “.ami”:

a) the parameter tree must not contain the Reserved\_Parameters branch but must contain the Model\_Specific branch

b) only Usage Info is allowed

Note that in the case when a parameter is located in an .ami file and it is of Usage In, the parameter value will be passed into the AMI executable model but this does not mean that the same parameter couldn’t be used by other model(s) which are instantiated through [External Model] or [External Circuit].

Multiple parameters may only be listed on a single line if no value assignments are made. When the Parameters line includes a parameter value assignment, each parameter must be listed on a new line. String literals must be enclosed in double quotes.

The EDA tool may provide additional means to the user to assign values to Parameters. This may include the option to override the values provided in the .ibs file, to allow the user to make selections for multi-valued parameters in the parameter tree, or to provide values for uninitialized Parameters.

Converter\_Parameters:

This optional subparameter lists and initializes parameter names to be used as arguments for the A\_to\_D and/or D\_to\_A converter(s) of the [External Model] keyword under which it appears. The list of Converter\_Parameters may span several lines by using the word Converter\_Parameters at the start of each line. Any A\_to\_D or D\_to\_A argument which is entered as a parameter must be declared and initialized with the Converter\_Parameters subparameter.

Converter\_Parameters are locally scoped under each [External Model] keyword, i.e., the same converter parameter under two different [External Model]s will have independent values.

The Converter\_Parameters subparameter must contain one parameter name per line, which must be followed by an equal sign and a constant numeric literal or a reference to a parameter name which is located in a parameter tree. The reference must begin with a file name, followed by an open parentheses and a the tree root name, a new open parentheses for any branch names (including the Reserved\_Parameters or Model\_Specific branch names if present in the tree) and the parameter name, and a matching set of closing parentheses. The file reference may point to any file which contains one or more parameter trees. The files referenced must be located in the same directory as the .ibs file containing the reference. The file names of parameter files must follow the rules for file names given in Section 3, “GENERAL SYNTAX RULES AND GUIDELINES”. Parameter files may only contain parameter trees using the tree syntax described in IBIS in Section 10.3 with the following exceptions and additions:

When the extension of the external parameter’s file name ends with “.ami”:

a) only Usage In or Usage Info are allowed for parameters which are to be passed into models instantiated by the [External Model] or the [External Circuit] keywords

When the extension of the external parameter’s file name does not end with “.ami”:

a) the parameter tree must not contain the Reserved\_Parameters branch but must contain the Model\_Specific branch

b) only Usage Info is allowed

Note that in the case when a parameter is located in an .ami file and it is of Usage In, the parameter value will be passed into the AMI executable model but this does not mean that the same parameter couldn’t be used by other model(s) which are instantiated through [External Model] or [External Circuit].

The EDA tool may provide additional means to the user to make assignments to Converter\_Parameters. This may include the option to override the values provided in the .ibs file, or to allow the user to make selections for multi-valued parameters in the parameter tree.

Ports:

Ports are interfaces to the [External Model] which are available to the user and tool at the IBIS level. They are used to connect the [External Model] to die nodes. The Ports parameter is used to identify the ports of the [External Model] to the simulation tool. The port assignment is by position and the port names do not have to match exactly the names inside the external file. The list of port names may span several lines if the word Ports is used at the start of each line.

Model units under [External Model] may only use reserved ports. The reserved, pre-defined port names are listed in the General Assumptions heading above. As noted earlier, digital and analog reserved port functions will be assumed by the tool and connections made accordingly. All the ports appropriate to the particular Model\_type subparameter entry must be explicitly listed (see below). Note that the user may connect SPICE, IBIS-ISS, Verilog-A(MS) and VHDL-A(MS) models to A\_to\_D and D\_to\_A converters using custom names for analog ports within the model unit, as long as the digital ports of the converters use the digital reserved port names.

The rules for pad connections with [External Model] are identical to those for [Model]. The [Pin Mapping] keyword may be used with [External Model]s but is not required. If used, the [External Model] specific voltage supply ports—A\_puref, A\_pdref, A\_gcref, A\_pcref, and A\_extref—are connected as defined under the [Pin Mapping] keyword. In all cases, the voltage levels connected on the reserved supply ports are defined by the [Power Clamp Reference], [GND Clamp Reference], [Pullup Reference], [Pulldown Reference], and/or [Voltage Range] keywords, as in the case of [Model].

Digital-to-Analog/Analog-to-Digital Conversions:

These subparameters define all digital-to-analog and analog-to-digital converters needed to properly connect digital signals with the analog ports of referenced external SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) models. These subparameters must be used when [External Model] references a file written in the SPICE, IBIS-ISS, Verilog-A(MS), or VHDL-A(MS) languages. They are not permitted with Verilog-AMS or VHDL-AMS external files.

D\_to\_A:

As assumed in [Model], some interface ports of [External Model] circuits expect digital input signals. As SPICE, IBIS-ISS, Verilog-A(MS), or VHDL-A(MS) models understand only analog signals, some conversion from digital to analog format is required. For example, input logical states such as “0” or “1”, implied in [Model], must be converted to actual input voltage stimuli, such as a voltage ramp, for SPICE simulation.

The D\_to\_A subparameter provides information for converting a digital stimulus, such as “0” or “1”, into an analog voltage ramp (a digital “X” input is ignored by D\_to\_A converters). Each digital port which carries data for conversion to analog format must have its own D\_to\_A line.

The D\_to\_A subparameter is followed by eight or optionally nine arguments:

d\_port port1 port2 vlow vhigh trise tfall corner\_name polarity

The d\_port entry holds the name of the digital port. This entry is used for the reserved port names D\_drive, D\_enable, and D\_switch. The port1 and port2 entries hold the SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) analog input port names across which voltages are specified. These entries are used for the user-defined port names, together with another port name, used as a reference.

Normally port1 accepts an input signal and port2 is the reference for port1. However, for an opposite polarity stimulus, port1 could be connected to a reference port and port2 could serve as the input. In some situations, such as in the case of a true differential buffer model, it might be desirable to provide two D\_to\_A converters, one to drive the Non-Inverting input and the other one to drive the Inverting input. In this case the D\_to\_A converters may be defined with the polarity argument, one with the value Non-Inverting and the other with the value Inverting.

The vlow and vhigh entries accept analog voltage values which must correspond to the digital off and on states, where the vhigh value must be greater than the vlow value. When polarity is Non-Inverting, vlow corresponds to the digital off state '0', vhigh corresponds to the digital on state '1', trise corresponds to the analog edge rate going from the digital off to on state, and tfall corresponds to the analog edge rate going from the digital on to off state. When polarity is Inverting, the analog behavior corresponds to the opposite digital states. For example, a 3.3 V ground-referenced buffer would list vlow as 0 V and vhigh as 3.3 V. For a Non-Inverting D\_to\_A converter, a rising edge in D\_drive would result in a transition from 0 V to 3.3 V, and for an Inverting D\_to\_A converter, a rising edge in D\_drive would result in a transition from 3.3 V to 0 V. The trise and tfall entries are times, must be positive, and define input ramp rise and fall times between 0 and 100 percent.

The vlow, vhigh, trise and tfall arguments may be defined by parameter names, which must be declared and initialized by one or more Converter\_Parameters subparameter.

The corner\_name entry holds the name of the external model corner being referenced, as listed under the Corner subparameter.

The last argument, polarity, is optional. If present, its value must be "Inverting" or "Non-Inverting". If the argument is not present, "Non-Inverting" is in effect. The polarity argument may only be used with D\_to\_A converters which are connected to the d\_port name D\_drive. If the polarity argument is used, two D\_to\_A converter lines are required, one defined as Non-Inverting and another defined as Inverting.

At least one D\_to\_A line must be present, corresponding to the “Typ” corner model, for each digital line to be converted. Additional D\_to\_A lines for other corners may be omitted. In this case, the typical corner D\_to\_A entries will apply to all model corners and the “Typ” corner\_name entry may be omitted if the polarity argument is not present. When the polarity argument is present, the corner\_name argument must also be present.

A\_to\_D:

The A\_to\_D subparameter is used to generate a digital state (“0”, “1”, or “X”) based on analog voltages generated by the SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) model or analog voltages present at the pad/pin. This allows an analog signal from the external SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) circuit or pad/pin to be read as a digital signal by the simulation tool.

The A\_to\_D subparameter is followed by six arguments:

d\_port port1 port2 vlow vhigh corner\_name

The d\_port entry lists the reserved port name D\_receive. As with D\_to\_A, the port1 entry would normally contain the reserved name A\_signal (see below) or a user-defined port name, while port2 may list any other analog reserved port name, used as a reference. The voltage measurements are taken in this example from the port1 entry with respect to the port2 entry. These ports must also be named by the Ports subparameter.

The vlow and vhigh entries list the low and high analog threshold voltage values. The reported digital state on D\_receive will be “0” if the measured voltage is lower than the vlow value, “1” if above the vhigh value, and “X” otherwise.

The vlow and vhigh arguments may be defined by parameter names, which must be declared and initialized by one or more Converter\_Parameters subparameter.

The corner\_name entry holds the name of the external model corner being referenced, as listed under the Corner subparameter.

At least one A\_to\_D line must be supplied corresponding to the “Typ” corner model. Other A\_to\_D lines for other corners may be omitted. In this case, the typical corner A\_to\_D entries will apply to all model corners.

IMPORTANT: measurements for receivers in IBIS are normally assumed to be conducted at the die pads/pins. In such cases, the electrical input model data comprises a “load” which affects the waveform seen at the pads. However, for models measure the analog input response at the die pads or inside the circuit (this does not preclude tools from reporting digital D\_receive and/or analog port responses in addition to at-pad A\_signal response). If at-pad measurements are desired, the A\_signal port would be named in the A\_to\_D line under port1. The A\_to\_D converter then effectively acts “in parallel” with the load of the circuit. If internal measurements are desired (e.g., if the user wishes to view the signal after processing by the receiver), the user-defined signal port would be named in the A\_to\_D line under port1. The A\_to\_D converter is effectively “in series” with the receiver model. The vhigh and vlow parameters should be adjusted as appropriate to the measurement point of interest.

Note that, while the port assignments and SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) model must be provided by the user, the D\_to\_A and A\_to\_D converters will be provided automatically by the tool (the converter parameters must still be declared by the user). There is no need for the user to develop external SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) code specifically for these functions.

A conceptual diagram of the port connections of a SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) [External Model] is shown in Figure 24. The example illustrates an I/O buffer. Note that the drawing implies that the D\_receive state changes in response to the analog signal my\_receive, not A\_signal:



1. - Example of an [External Model] I/O Buffer Using SPICE,  
   Verilog-A(MS), or VHDL-A(MS)

Pseudo-Differential Buffers:

Pseudo-differential buffers may be described using a pair of [External Model]s which may or may not be identical. Each of the analog I/O signal ports (usually A\_signal) is connected to a specific pad through the [Pin] list in the usual fashion, and the two ports are linked together as a differential pair through the [Diff Pin] keyword.

The reserved signal name A\_signal is required for the I/O signal ports of [External Model]s connected to pads used in a pseudo-differential configuration.

Users should note that, in pseudo-differential buffers, only one formal signal port is used to stimulate the two [External Model] digital inputs (D\_drive). One of these inputs will reflect the timing and polarity of the formal signal port named by the user, while the other input is inverted and (potentially) delayed with respect to the formal port as defined under the [Diff Pin] keyword. THIS SECOND PORT IS AUTOMATICALLY CREATED BY THE SIMULATION TOOL. Users do not have to create special structures to invert or delay the driven digital signal. Simulation tools will correctly implement the two input ports once the [Diff Pin] keyword has been detected in the .ibs file. This approach is identical to that used in native IBIS.

The D\_to\_A adapters used for SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) files can be set up to control ports on pseudo-differential buffers. If SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) is used as an external language, the [Diff Pin] vdiff subparameter overrides the contents of vlow and vhigh under A\_to\_D.

IMPORTANT: For pseudo-differential buffers under [External Model], the analog input response may only be measured at the die pads. The [Diff Pin] parameter is required, and controls both the polarity and the differential thresholds used to determine the D\_receive port response (the D\_receive port will follow the state of the non-inverting pin/pad as referenced to the inverting pin/pad). For SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) models, the A\_to\_D line must name the A\_signal port under either port1 or port2, as with a single-ended buffer. The A\_to\_D converter then effectively acts “in parallel” with the load of the buffer circuit. The vhigh and vlow parameters will be overridden by the [Diff Pin] vdiff declarations.

The port relationships are shown in Figure 25.



1. -Example SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) Implementation

Figure 26 illustrates the same concepts with a \*-AMS model. Note that the state of D\_receive is determined by the tool automatically by observing the A\_signal ports. The outputs of the actual receiver circuits in the \*-AMS models are not used for determining D\_receive.



1. - Example \*-AMS Implementation

Two additional differential timing test loads are available:

Rref\_diff, Cref\_diff

These subparameters are also available under the [Model Spec] keyword for typical, minimum, and maximum corners.

These timing test loads require both sides of the differential model to be operated. They can be used with the existing timing test loads Rref, Cref, and Vref. The existing timing test loads and Vmeas are used if Rref\_diff and Cref\_diff are NOT given.

True Differential Models:

True differential buffers may be described using [External Model]. In a true differential [External Model], the differential I/O ports which connect to die pads use the reserved names A\_signal\_pos and A\_signal\_neg, as shown in Figure 27.



1. - Port Names for True Differential I/O Buffer

IMPORTANT: All true differential models under [External Model] assume single-ended digital port connections (D\_drive, D\_enable, D\_receive).

The [Diff Pin] keyword is still required within the same [Component] definition when [External Model] describes a true differential buffer. The [Model] names or [Model Selector] names referenced by the pair of pins listed in an entry of the [Diff Pin] MUST be the same.

The D\_to\_A or A\_to\_D adapters used for SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) files may be set up to control or respond to true differential ports. An example is shown in Figure 28.



1. - Example SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) Implementation of a True Differential Buffer

If at-pad or at-pin measurement using a SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) [External Model] is desired, the vlow and vhigh entries under the A\_to\_D subparameter must be consistent with the values of the [Diff Pin] vdiff subparameter entry (the vlow value must match -vdiff, and the vhigh value must match +vdiff). The logic states produced by the A\_to\_D conversion follow the same rules as for single-ended buffers, listed above. An example is shown at the end of this section.

IMPORTANT: For true-differential buffers under [External Model], the user can choose whether to measure the analog input response at the die pads or internal to the circuit (this does not preclude tools from reporting digital D\_receive and/or analog responses in addition to at-pad A\_signal response). If at-pad measurements for a SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) model are desired, the A\_signal\_pos port would be named in the A\_to\_D line under port1 and A\_signal\_neg under port2. The A\_to\_D converter then effectively acts “in parallel” with the load of the buffer circuit. If internal measurements are desired (e.g., if the user wishes to view the signal after processing by the input buffer), the user-defined analog signal port would be named in the A\_to\_D line under port1. The A\_to\_D converter is “in series” with the receiver buffer model. The vhigh and vlow parameters should be adjusted appropriate to the measurement point of interest, so long as they as they are consistent with the [Diff Pin] vdiff declarations.

Note that the thresholds refer to the state of the non-inverting signal, using the inverting signal as a reference. Therefore, the output signal is considered high when, for example, the non-inverting input is +200 mV above the inverting input. Similarly, the output signal is considered low when the same non-inverting input is -200 mV “above” the inverting input.

EDA tools will report the state of the D\_receive port for true differential \*-AMS [External Model]s according to the AMS code written by the model author; the use of [Diff Pin] does not affect the reporting of D\_receive in this case. EDA tools are free to additionally report the state of the I/O pads according to the [Diff Pin] vdiff subparameter.

For SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) and \*-AMS true differential [External Model]s, the EDA tool must not override or change the model author’s connection of the D\_receive port.

Four additional Model\_type arguments are available under the [Model] keyword. One of these must be used when an [External Model] describes a true differential model:

I/O\_diff, Output\_diff, 3-state\_diff, Input\_diff

Two additional differential timing test loads are available:

Rref\_diff, Cref\_diff

These subparameters are also available under the [Model Spec] keyword for the typical, minimum, and maximum corner cases.

These timing test loads require that both the inverting and non-inverting ports of the differential model refer to valid buffer model data (not terminations, supply rails, etc.). The differential test loads may also be combined with the single-ended timing test loads Rref, Cref, and Vref. Note that the single-ended timing test loads plus Vmeas are used if Rref\_diff and Cref\_diff are NOT supplied.

Series and Series Switch Models:

Native IBIS did not define the transition characteristics of digital switch controls. Switches were assumed to either be on or off during a simulation and I-V characteristics could be defined for either or both states. The [External Model] format allows users to control the state of a switch through the D\_switch port. As with other digital ports, the use of SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) in an [External Model] requires the user to declare D\_to\_A ports, to convert the D\_switch signal to an analog input to the SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) model (whether the port’s state may actually change during a simulation is determined by the EDA tool used).

Series and Series\_switch devices both are described under the [External Model] keyword using the reserved port names A\_pos and A\_neg. Note that the [Series Pin Mapping] keyword must be present and correctly used elsewhere in the file, in order to properly set the logic state of the switch. The A\_pos port is defined in the first entry of the [Series Pin Mapping] keyword, and the A\_neg port is defined in the pin2 entry. For series switches, the [Series Switch Groups] keyword is required.

Ports required for various Model\_types:

As [External Model] makes use of the [Model] keyword’s Model\_type subparameter, not all digital and analog reserved ports may be needed for all Model\_types. Table 12 and Table 13 below define which reserved port names are required for various Model\_types.

Table 13 – Required Port Names for Single-ended Model\_type Assignments

| **Model\_type** | **D\_drive** | **D\_enable** | **D\_receive** | **A\_signal** | **D\_switch** | **A\_pos** | **A\_neg** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I/O\* | X | X | X | X |  |  |  |
| 3-state\* | X | X |  | X |  |  |  |
| Output\*, Open\* | X |  |  | X |  |  |  |
| Input |  |  | X | X |  |  |  |
| Terminator |  |  |  | X |  |  |  |
| Series |  |  |  |  |  | X | X |
| Series\_switch |  |  |  |  | X | X | X |

Table 14 – Required Port Names for Differential Model\_type Assignments

| **Model\_type** | **D\_drive** | **D\_enable** | **D\_receive** | **A\_signal\_pos** | **A\_signal\_neg** |
| --- | --- | --- | --- | --- | --- |
| I/O\_diff | X | X | X | X | X |
| 3-state\_diff | X | X |  | X | X |
| Output\_diff | X |  |  | X | X |
| Input\_diff |  |  | X | X | X |

*Examples:*

Example [External Model] using SPICE:

[Model] ExBufferSPICE

Model\_type I/O

Vinh = 2.0

Vinl = 0.8

|

| Other model subparameters are optional

|

| typ min max

[Voltage Range] 3.3 3.0 3.6

|

[Ramp]

dV/dt\_r 1.57/0.36n 1.44/0.57n 1.73/0.28n

dV/dt\_f 1.57/0.35n 1.46/0.44n 1.68/0.28n

|

[External Model]

Language SPICE

|

| Corner corner\_name file\_name circuit\_name (.subckt name)

Corner Typ buffer\_typ.spi buffer\_io\_typ

Corner Min buffer\_min.spi buffer\_io\_min

Corner Max buffer\_max.spi buffer\_io\_max

|

| Parameters - Not supported in SPICE

|

| Ports List of port names (in same order as in SPICE)

Ports A\_signal my\_drive my\_enable my\_receive my\_ref

Ports A\_puref A\_pdref A\_pcref A\_gcref A\_extref

|

| D\_to\_A d\_port port1 port2 vlow vhigh trise tfall corner\_name

D\_to\_A D\_drive my\_drive my\_ref 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_enable my\_enable A\_gcref 0.0 3.3 0.5n 0.3n Typ

|

| A\_to\_D d\_port port1 port2 vlow vhigh corner\_name

A\_to\_D D\_receive my\_receive my\_ref 0.8 2.0 Typ

|

| Note: A\_signal might also be used instead of a user-defined interface port

| for measurements taken at the die pads

|

[End External Model]

Example [External Model] using IBIS-ISS:

[Model] ExBufferISS

Model\_type I/O

Vinh = 2.0

Vinl = 0.8

|

| Other model subparameters are optional

|

| typ min max

[Voltage Range] 3.3 3.0 3.6

|

[Ramp]

dV/dt\_r 1.57/0.36n 1.44/0.57n 1.73/0.28n

dV/dt\_f 1.57/0.35n 1.46/0.44n 1.68/0.28n

|

[External Model]

Language IBIS-ISS

|

| Corner corner\_name file\_name circuit\_name (.subckt name)

Corner Typ buffer\_typ.spi buffer\_io\_typ

Corner Min buffer\_min.spi buffer\_io\_min

Corner Max buffer\_max.spi buffer\_io\_max

|

| List of parameters

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

Parameters C1\_value

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

|

| List of converter parameters

Converter\_Parameters MyVlow = 0.0

Converter\_Parameters MyVHigh = 3.3

Converter\_Parameters MyVinl = paramfile.par(TreeRootName(Model\_Specific(Vinl)))

Converter\_Parameters MyVinh = paramfile.par(TreeRootName(Model\_Specific(Vinh)))

Converter\_Parameters MyTrise = paramfile.par(TreeRootName(Model\_Specific(Trf)))

Converter\_Parameters MyTfall = paramfile.par(TreeRootName(Model\_Specific(Trf)))

|

| Ports List of port names (in same order as in ISS)

Ports A\_signal my\_drive my\_enable my\_receive my\_ref

Ports A\_puref A\_pdref A\_pcref A\_gcref A\_extref

|

| D\_to\_A d\_port port1 port2 vlow vhigh trise tfall corner\_name

D\_to\_A D\_drive my\_drive my\_ref MyVlow MyVhigh MyTfall MyTrise Typ

D\_to\_A D\_enable my\_enable A\_gcref 0.0 3.3 0.5n 0.3n Typ

|

| A\_to\_D d\_port port1 port2 vlow vhigh corner\_name

A\_to\_D D\_receive my\_receive my\_ref MyVinl MyVinh Typ

|

| Note: A\_signal might also be used instead of a user-defined interface port

| for measurements taken at the die pads

|

[End External Model]

Example [External Model] using VHDL-AMS:

[Model] ExBufferVHDL

Model\_type I/O

Vinh = 2.0

Vinl = 0.8

|

| Other model subparameters are optional

|

| typ min max

[Voltage Range] 3.3 3.0 3.6

|

[Ramp]

dV/dt\_r 1.57/0.36n 1.44/0.57n 1.73/0.28n

dV/dt\_f 1.57/0.35n 1.46/0.44n 1.68/0.28n

|

[External Model]

Language VHDL-AMS

|

| Corner corner\_name file\_name entity(architecture)

Corner Typ buffer\_typ.vhd buffer(buffer\_io\_typ)

Corner Min buffer\_min.vhd buffer(buffer\_io\_min)

Corner Max buffer\_max.vhd buffer(buffer\_io\_max)

|

| Parameters List of parameters

Parameters delay rate

Parameters preemphasis

| Ports List of port names (in same order as in VHDL-AMS)

Ports A\_signal A\_puref A\_pdref A\_pcref A\_gcref

Ports D\_drive D\_enable D\_receive

|

[End External Model]

Example [External Model] using Verilog-AMS:

[Model] ExBufferVerilog

Model\_type I/O

Vinh = 2.0

Vinl = 0.8

|

| Other model subparameters are optional

|

| typ min max

[Voltage Range] 3.3 3.0 3.6

|

[Ramp]

dV/dt\_r 1.57/0.36n 1.44/0.57n 1.73/0.28n

dV/dt\_f 1.57/0.35n 1.46/0.44n 1.68/0.28n

|

[External Model]

Language Verilog-AMS

|

| Corner corner\_name file\_name circuit\_name (module)

Corner Typ buffer\_typ.v buffer\_io\_typ

Corner Min buffer\_min.v buffer\_io\_min

Corner Max buffer\_max.v buffer\_io\_max

|

| Parameters List of parameters

Parameters delay rate

Parameters preemphasis

|

| Ports List of port names (in same order as in Verilog-AMS)

Ports A\_signal A\_puref A\_pdref A\_pcref A\_gcref

Ports D\_drive D\_enable D\_receive

|

[End External Model]

Example [External Model] using VHDL-A(MS):

[Model] ExBufferVHDL\_analog

Model\_type I/O

Vinh = 2.0

Vinl = 0.8

|

| Other model subparameters are optional

|

| typ min max

[Voltage Range] 3.3 3.0 3.6

|

[Ramp]

dV/dt\_r 1.57/0.36n 1.44/0.57n 1.73/0.28n

dV/dt\_f 1.57/0.35n 1.46/0.44n 1.68/0.28n

|

[External Model]

Language VHDL-A(MS)

|

| Corner corner\_name file\_name circuit\_name entity(architecture)

Corner Typ buffer\_typ.vhd buffer(buffer\_io\_typ)

Corner Min buffer\_min.vhd buffer(buffer\_io\_min)

Corner Max buffer\_max.vhd buffer(buffer\_io\_max)

|

| Parameters List of parameters

Parameters delay rate

Parameters preemphasis

|

| Ports List of port names (in same order as in VHDL-A(MS))

Ports A\_signal my\_drive my\_enable my\_receive my\_ref

Ports A\_puref A\_pdref A\_pcref A\_gcref A\_extref

|

| D\_to\_A d\_port port1 port2 vlow vhigh trise tfall corner\_name

D\_to\_A D\_drive my\_drive my\_ref 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_enable my\_enable A\_gcref 0.0 3.3 0.5n 0.3n Typ

|

| A\_to\_D d\_port port1 port2 vlow vhigh corner\_name

A\_to\_D D\_receive my\_receive my\_ref 0.8 2.0 Typ

|

| Note: A\_signal might also be used instead of a user-defined interface port

| for measurements taken at the die pads

Example [External Model] using Verilog-A(MS):

[Model] ExBufferVerilog\_analog

Model\_type I/O

Vinh = 2.0

Vinl = 0.8

|

| Other model subparameters are optional

|

| typ min max

[Voltage Range] 3.3 3.0 3.6

|

[Ramp]

dV/dt\_r 1.57/0.36n 1.44/0.57n 1.73/0.28n

dV/dt\_f 1.57/0.35n 1.46/0.44n 1.68/0.28n

|

[External Model]

Language Verilog-A(MS)

|

| Corner corner\_name file\_name circuit\_name (module)

Corner Typ buffer\_typ.va buffer\_io\_typ

Corner Min buffer\_min.va buffer\_io\_min

Corner Max buffer\_max.va buffer\_io\_max

| Parameters List of parameters

Parameters delay rate

Parameters preemphasis

|

| Ports List of port names (in same order as in Verilog-A(MS))

Ports A\_signal my\_drive my\_enable my\_receive my\_ref

Ports A\_puref A\_pdref A\_pcref A\_gcref A\_extref

|

| D\_to\_A d\_port port1 port2 vlow vhigh trise tfall corner\_name

D\_to\_A D\_drive my\_drive my\_ref 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_enable my\_enable A\_gcref 0.0 3.3 0.5n 0.3n Typ

|

| A\_to\_D d\_port port1 port2 vlow vhigh corner\_name

A\_to\_D D\_receive my\_receive my\_ref 0.8 2.0 Typ

|

| Note: A\_signal might also be used instead of a user-defined interface port

| for measurements taken at the die pads

|

[End External Model]

Example of True Differential [External Model] using SPICE:

[Model] Ext\_SPICE\_Diff\_Buff

Model\_type I/O\_diff

Rref\_diff = 100

|

| Other model subparameters are optional

|

| typ min max

[Voltage Range] 3.3 3.0 3.6

|

[Ramp]

dV/dt\_r 1.57/0.36n 1.44/0.57n 1.73/0.28n

dV/dt\_f 1.57/0.35n 1.46/0.44n 1.68/0.28n

|

[External Model]

Language SPICE

|

| Corner corner\_name file\_name circuit\_name (.subckt name)

Corner Typ diffio.spi diff\_io\_typ

Corner Min diffio.spi diff\_io\_min

Corner Max diffio.spi diff\_io\_max

|

| Ports List of port names (in same order as in SPICE)

Ports A\_signal\_pos A\_signal\_neg my\_receive my\_drive my\_enable

Ports A\_puref A\_pdref A\_pcref A\_gcref A\_extref my\_ref A\_gnd

|

| D\_to\_A d\_port port1 port2 vlow vhigh trise tfall corner\_name

D\_to\_A D\_drive my\_drive my\_ref 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_drive my\_drive my\_ref 0.0 3.0 0.6n 0.3n Min

D\_to\_A D\_drive my\_drive my\_ref 0.0 3.6 0.4n 0.3n Max

D\_to\_A D\_enable my\_enable my\_ref 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_enable my\_enable my\_ref 0.0 3.0 0.6n 0.3n Min

D\_to\_A D\_enable my\_enable my\_ref 0.0 3.6 0.4n 0.3n Max

|

| A\_to\_D d\_port port1 port2 vlow vhigh corner\_name

A\_to\_D D\_receive A\_signal\_pos A\_signal\_neg -200m 200m Typ

A\_to\_D D\_receive A\_signal\_pos A\_signal\_neg -200m 200m Min

A\_to\_D D\_receive A\_signal\_pos A\_signal\_neg -200m 200m Max

|

[End External Model]

Example of True Differential [External Model] using IBIS-ISS:

[Model] Ext\_ISS\_Diff\_Buff

Model\_type I/O\_diff

Rref\_diff = 100

|

| Other model subparameters are optional

|

| typ min max

[Voltage Range] 3.3 3.0 3.6

|

[Ramp]

dV/dt\_r 1.57/0.36n 1.44/0.57n 1.73/0.28n

dV/dt\_f 1.57/0.35n 1.46/0.44n 1.68/0.28n

|

[External Model]

Language IBIS-ISS

|

| Corner corner\_name file\_name circuit\_name (.subckt name)

Corner Typ diffio.spi diff\_io\_typ

Corner Min diffio.spi diff\_io\_min

Corner Max diffio.spi diff\_io\_max

|

| List of parameters

Parameters sp\_file\_name

Parameters c\_diff r\_diff

|

|

| List of converter parameters

Converter\_Parameters MyVlow = 0.0

Converter\_Parameters MyVHigh = 3.3

|

| Ports List of port names (in same order as in IBIS-ISS)

Ports A\_signal\_pos A\_signal\_neg my\_receive my\_driveP my\_driveN my\_enable

Ports A\_puref A\_pdref A\_pcref A\_gcref A\_extref my\_ref A\_gnd

|

| D\_to\_A d\_port port1 port2 vlow vhigh trise tfall corner\_name polarity

D\_to\_A D\_drive my\_driveP my\_ref MyVlow MyVHigh 0.5n 0.3n Typ Non-Inverting

D\_to\_A D\_drive my\_driveN my\_ref MyVlow MyVHigh 0.5n 0.3n Typ Inverting

D\_to\_A D\_enable my\_enable my\_ref 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_enable my\_enable my\_ref 0.0 3.0 0.6n 0.3n Min

D\_to\_A D\_enable my\_enable my\_ref 0.0 3.6 0.4n 0.3n Max

|

| A\_to\_D d\_port port1 port2 vlow vhigh corner\_name

A\_to\_D D\_receive A\_signal\_pos A\_signal\_neg -200m 200m Typ

A\_to\_D D\_receive A\_signal\_pos A\_signal\_neg -200m 200m Min

A\_to\_D D\_receive A\_signal\_pos A\_signal\_neg -200m 200m Max

|

[End External Model]

Example of True Differential [External Model] using VHDL-AMS:

[Model] Ext\_VHDL\_Diff\_Buff

Model\_type I/O\_diff

Rref\_diff = 100

|

| typ min max

[Voltage Range] 3.3 3.0 3.6

|

[Ramp]

dV/dt\_r 1.57/0.36n 1.44/0.57n 1.73/0.28n

dV/dt\_f 1.57/0.35n 1.46/0.44n 1.68/0.28n

|

| Other model subparameters are optional

|

[External Model]

Language VHDL-AMS

|

| Corner corner\_name file\_name entity(architecture)

Corner Typ diffio\_typ.vhd buffer(diff\_io\_typ)

Corner Min diffio\_min.vhd buffer(diff\_io\_min)

Corner Max diffio\_max.vhd buffer(diff\_io\_max)

|

| Parameters List of parameters

Parameters delay rate

Parameters preemphasis

|

| Ports List of port names (in same order as in VHDL-AMS)

Ports A\_signal\_pos A\_signal\_neg D\_receive D\_drive D\_enable

Ports A\_puref A\_pdref A\_pcref A\_gcref

|

[End External Model]

Example of Pseudo-Differential [External Model] using SPICE:

| Note that [Pin] and [Diff Pin] declarations are shown for clarity

|

|

[Pin] signal\_name model\_name R\_pin L\_pin C\_pin

1 Example\_pos Ext\_SPICE\_PDiff\_Buff

2 Example\_neg Ext\_SPICE\_PDiff\_Buff

|

| ...

|

[Diff Pin] inv\_pin vdiff tdelay\_typ tdelay\_min tdelay\_max

1 2 200mV 0ns 0ns 0ns

|

| ...

|

[Model] Ext\_SPICE\_PDiff\_Buff

Model\_type I/O

|

| Other model subparameters are optional

|

| typ min max

[Voltage Range] 3.3 3.0 3.6

|

[Ramp]

dV/dt\_r 1.57/0.36n 1.44/0.57n 1.73/0.28n

dV/dt\_f 1.57/0.35n 1.46/0.44n 1.68/0.28n

|

[External Model]

Language SPICE

|

| Corner corner\_name file\_name circuit\_name (.subckt name)

Corner Typ diffio.spi diff\_io\_typ

Corner Min diffio.spi diff\_io\_min

Corner Max diffio.spi diff\_io\_max

|

| Ports List of port names (in same order as in SPICE)

Ports A\_signal my\_drive my\_enable my\_ref

Ports A\_puref A\_pdref A\_pcref A\_gcref A\_gnd A\_extref

|

| D\_to\_A d\_port port1 port2 vlow vhigh trise tfall corner\_name

D\_to\_A D\_drive my\_drive my\_ref 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_drive my\_drive my\_ref 0.0 3.0 0.6n 0.3n Min

D\_to\_A D\_drive my\_drive my\_ref 0.0 3.6 0.4n 0.3n Max

D\_to\_A D\_enable my\_enable A\_pcref 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_enable my\_enable A\_pcref 0.0 3.0 0.6n 0.3n Min

D\_to\_A D\_enable my\_enable A\_pcref 0.0 3.6 0.4n 0.3n Max

|

| A\_to\_D d\_port port1 port2 vlow vhigh corner\_name

A\_to\_D D\_receive A\_signal my\_ref 0.8 2.0 Typ

A\_to\_D D\_receive A\_signal my\_ref 0.8 2.0 Min

A\_to\_D D\_receive A\_signal my\_ref 0.8 2.0 Max

|

| This example shows the evaluation of the received signals at the die

| pads. [Diff Pin] defines the interpretation of the A\_to\_D output

| polarity and levels and overrides the A\_to\_D settings shown above.

|

[End External Model]

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

*Required:* No

*Description:* Used to reference an external file containing an arbitrary circuit description using one of the supported languages.

*Sub-Params:* Language, Corner, Parameters, Converter\_Parameters, Ports, D\_to\_A, A\_to\_D

*Usage Rules:* Each [External Circuit] keyword must be followed by a unique name that differs from any name used for any [Model] or [Submodel] keyword.

The [External Circuit] keyword may appear multiple times. It is not scoped by any other keyword.

Each instance of an [External Circuit] is referenced by one or more [Circuit Call] keywords discussed later. (The [Circuit Call] keyword cannot be used to reference a [Model] keyword.)

The [External Circuit] keyword and contents may be placed anywhere in the file, outside of any [Component] keyword group or [Model] keyword group, in a manner similar to that of the [Model] keyword.

Subparameter Definitions:

Language:

Accepts “SPICE”, “IBIS-ISS”, “VHDL-AMS”, “Verilog-AMS”, “VHDL-A(MS)” or “Verilog-A(MS)” as arguments. The Language subparameter is required and must appear only once.

Corner:

Three entries follow the Corner subparameter on each line:

corner\_name file\_name circuit\_name

The corner\_name entry is “Typ”, “Min”, or “Max”. The file\_name entry points to the referenced file in the same directory as the .ibs file.

Up to three Corner lines are permitted. A “Typ” line is required. If “Min” and/or “Max” data is missing, the tool may use “Typ” data in its place. However, the tool should notify the user of this action.

The circuit\_name entry provides the name of the circuit to be simulated within the referenced file. For SPICE and IBIS-ISS files, this is normally a “.subckt” name. For VHDL-AMS files, this is normally an “entity(architecture)” name pair. For Verilog-AMS files, this is normally a “module” name.

No character limits, case-sensitivity limits or extension conventions are required or enforced for file\_name and circuit\_name entries. However, the total number of characters in each Corner line must comply with Section 3. Furthermore, lower-case file\_name entries are recommended to avoid possible conflicts with file naming conventions under different operating systems. Case differences between otherwise identical file\_name entries or circuit\_name entries should be avoided. External languages may not support case-sensitive distinctions.

Parameters:

Lists names of parameters that may be passed into an external circuit file. Each Parameters entry must match a name or keyword in the external file or language. The list of Parameters can span several lines by using the word Parameters at the start of each line. The Parameters subparameter is optional, and the external circuit must operate with default settings without any Parameters assignments.

Parameter passing is not supported in SPICE. VHDL-AMS and VHDL-A(MS) parameters are supported using “generic” names, and Verilog-AMS and Verilog-A(MS) parameters are supported using “parameter” names. IBIS-ISS parameters are supported for all IBIS-ISS parameters which are defined on the subcircuit definition line.

Parameters are locally scoped under each [External Circuit] keyword, i.e., the same parameter under two different [External Circuit] will have independent values.

The parameter(s) listed under the Parameters subparameter may optionally be followed by an equal sign and a numeric, Boolean or string literal or a reference to a parameter name which is located in a parameter tree. The reference must begin with a file name, followed by an open parentheses and a the tree root name, a new open parentheses for any branch names (including the Reserved\_Parameters or Model\_Specific branch names if present in the tree) and the parameter name, and a matching set of closing parentheses. The file reference may point to any file which contains one or more parameter trees. The files referenced must be located in the same directory as the .ibs file containing the reference. The file names of parameter files must follow the rules for file names given in Section 3, “GENERAL SYNTAX RULES AND GUIDELINES”. Parameter files may only contain parameter trees using the tree syntax described in IBIS in Section 10.3 with the following exceptions and additions:

When the extension of the external parameter’s file name ends with “.ami”:

a) only Usage In or Usage Info are allowed for parameters which are to be passed into models instantiated by the [External Model] or the [External Circuit] keywords

When the extension of the external parameter’s file name does not end with “.ami”:

a) the parameter tree must not contain the Reserved\_Parameters branch but must contain the Model\_Specific branch

b) only Usage Info is allowed

Note that in the case when a parameter is located in an .ami file and it is of Usage In, the parameter value will be passed into the AMI executable model but this does not mean that the same parameter couldn’t be used by other model(s) which are instantiated through [External Model] or [External Circuit].

Multiple parameters may only be listed on a single line if no value assignments are made. When the Parameters line includes a parameter value assignment, each parameter must be listed on a new line. String literals must be enclosed in double quotes.

The EDA tool may provide additional means to the user to assign values to Parameters. This may include the option to override the values provided in the .ibs file, to allow the user to make selections for multi-valued parameters in the parameter tree, or to provide values for uninitialized Parameters.

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 the [External Circuit] keyword under which it appears. The list of Converter\_Parameters may span several lines by using the word Converter\_Parameters at the start of each line. Any A\_to\_D or D\_to\_A argument which is entered as a parameter must be declared and initialized with the Converter\_Parameters subparameter.

Converter\_Parameters are locally scoped under each [External Circuit] keyword, i.e., the same converter parameter under two different [External Circuit]s will have independent values.

The Converter\_Parameters subparameter must contain one parameter name per line, which must be followed by an equal sign and a constant numeric literal or a reference to a parameter name which is located in a parameter tree. The reference must begin with a file name, followed by an open parentheses and a the tree root name, a new open parentheses for any branch names (including the Reserved\_Parameters or Model\_Specific branch names if present in the tree) and the parameter name, and a matching set of closing parentheses. The file reference may point to any file which contains one or more parameter trees. The files referenced must be located in the same directory as the .ibs file containing the reference. The file names of parameter files must follow the rules for file names given in Section 3, “GENERAL SYNTAX RULES AND GUIDELINES”. Parameter files may only contain parameter trees using the tree syntax described in IBIS in Section 10.3 with the following exceptions and additions:

When the extension of the external parameter’s file name ends with “.ami”:

a) only Usage In or Usage Info are allowed for parameters which are to be passed into models instantiated by the [External Model] or the [External Circuit] keywords

When the extension of the external parameter’s file name does not end with “.ami”:

a) the parameter tree must not contain the Reserved\_Parameters branch but must contain the Model\_Specific branch

b) only Usage Info is allowed

Note that in the case when a parameter is located in an .ami file and it is of Usage In, the parameter value will be passed into the AMI executable model but this does not mean that the same parameter couldn’t be used by other model(s) which are instantiated through [External Model] or [External Circuit].

The EDA tool may provide additional means to the user to make assignments to Converter\_Parameters. This may include the option to override the values provided in the .ibs file, or to allow the user to make selections for multi-valued parameters in the parameter tree.

Ports:

Ports are interfaces to the [External Circuit] which are available to the user and tool at the IBIS level. They are used to connect the [External Circuit] to die nodes (including die pads and die bumps) and component pins. The Ports parameter is used to identify the ports of the [External Circuit] to the simulation tool. The port assignment is by position and the port names do not have to match exactly the names inside the external file. The list of port names may span several lines if the word Ports is used at the start of each line.

The Ports parameter is used to identify the ports of the [External Circuit] to the simulation tool. The port assignment is by position and the port names do not have to match exactly the port names in the external file. The list of port names may span several lines if the word Ports is used at the start of each line.

[External Circuit] allows any number of ports to be defined, with any names which comply with Section 3 format requirements. Reserved port names may be used, but ONLY DIGITAL PORTS will have the pre-defined functions listed in the General Assumptions heading above. User-defined and reserved port names may be combined within the same [External Circuit].

The [Pin Mapping] keyword cannot be used with [External Circuit] in the same [Component] description.

Digital-to-Analog/Analog-to-Digital Conversions:

These subparameters define all digital-to-analog and analog-to-digital converters needed to properly connect digital signals with the analog ports of referenced external SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) models. These subparameters must be used when [External Circuit] references a file written in the SPICE, IBIS-ISS,Verilog-A(MS) or VHDL-A(MS) language. They are not permitted with Verilog-AMS or VHDL-AMS external files.

D\_to\_A:

As assumed in [Model] and [External Model], some interface ports of [External Circuit]s expect digital input signals. As SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) models understand only analog signals, some conversion from digital to analog format is required. For example, input logical states such as “0” or “1” must be converted to actual input voltage stimuli, such as a voltage ramp, for SPICE simulation.

The D\_to\_A subparameter provides information for converting a digital stimulus, such as “0” or “1”, into an analog voltage ramp (a digital “X” input is ignored by D\_to\_A converters). Each digital port which carries data for conversion to analog format must have its own D\_to\_A declaration.

The D\_to\_A subparameter is followed by eight or optionally nine arguments:

d\_port port1 port2 vlow vhigh trise tfall corner\_name polarity

The d\_port entry holds the name of the digital port. This entry may contain user-defined port names or the reserved port names D\_drive, D\_enable, and D\_switch. he port1 and port2 entries hold the SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) analog input port names across which voltages are specified. These entries contain user-defined port names. One of these port entries must name a reference for the other port (for example, A\_gnd).

Normally, port1 accepts an input signal and port2 is the reference for port1. However, for an opposite polarity stimulus, port1 could be connected to a voltage reference and port2 could serve as the input. In some situations, such as in the case of a true differential buffer model, it might be desirable to provide two D\_to\_A converters, one to drive the Non-Inverting input and the other one to drive the Inverting input. In this case the D\_to\_A converters may be defined with the polarity argument, one with the value Non-Inverting and the other with the value Inverting.

The vlow and vhigh entries accept voltage values which correspond to fully-off and fully-on states, where the vhigh value must be greater than the vlow value. When polarity is Non-Inverting, vlow corresponds to the digital off state '0', vhigh corresponds to the digital on state '1', trise corresponds to the analog edge rate going from the digital off to on state, and tfall corresponds to the analog edge rate going from the digital on to off state. When polarity is Inverting, the analog behavior corresponds to the opposite digital states. For example, a 3.3 V ground-referenced buffer would list vlow as 0 V and vhigh as 3.3 V. For a Non-Inverting D\_to\_A converter, a rising edge in D\_drive would result in a transition from 0 V to 3.3 V, and for an Inverting D\_to\_A converter, a rising edge in D\_drive would result in a transition from 3.3 V to 0 V. The trise and tfall entries are times, must be positive and define input ramp rise and fall times between 0 and 100 percent.

The vlow, vhigh, trise and tfall arguments may be defined by parameter names, which must be declared and initialized by one or more Converter\_Parameters subparameter.

The corner\_name entry holds the name of the external circuit corner being referenced, as listed under the Corner subparameter.

The last argument, polarity, is optional. If present, its value must be "Inverting" or "Non-Inverting". If the argument is not present, "Non-Inverting" is in effect. The polarity argument may only be used with D\_to\_A converters which are connected to the d\_port name D\_drive. If the polarity argument is used, two D\_to\_A converter lines are required, one defined as Non-Inverting and another defined as Inverting. Any number of D\_to\_A subparameter lines is allowed, so long as each contains a unique port name entry and at least one unique port1 or port2 entry (i.e., several D\_to\_A declarations may use the same reference node under port1 or port2). At least one D\_to\_A line must be present, corresponding to the “Typ” corner model, for each digital line to be converted. Additional D\_to\_A lines for other corners may be omitted. In this case, the typical corner D\_to\_A entries will apply to all model corners and the “Typ” corner\_name entry may be omitted if the polarity argument is not present. When the polarity argument is present, the corner\_name argument must also be present.

A\_to\_D:

The A\_to\_D subparameter is used to generate a digital state (“0”, “1”, or “X”) based on analog voltages from the SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) model or from the pad/pin. This allows an analog signal from the external SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) circuit to be read as a digital signal by the simulation tool.

The A\_to\_D subparameter is followed by six arguments:

d\_port port1 port2 vlow vhigh corner\_name

The d\_port entry lists port names to be used for digital signals going. As with D\_to\_A, the port1 entry would contain a user-defined analog signal. Port2 would list another port name to be used as a reference. The voltage measurements are taken from the port1 entry with respect to the port2 entry. These ports must also be named by the Ports subparameter.

The vlow and vhigh entries list the low and high analog threshold voltage values. The reported digital state on D\_receive will be “0” if the measured voltage is lower than the vlow value, “1” if above the vhigh value, and “X” otherwise.

The vlow and vhigh arguments may be defined by parameter names, which must be declared and initialized by one or more Converter\_Parameters subparameter.

The corner\_name entry holds the name of the external model corner being referenced, as listed under the Corner subparameter.

Any number of A\_to\_D subparameter lines is allowed, so long as each line contains at least one column entry which is distinct from the column entries of all other lines. In practice, this means that A\_to\_D subparameter lines describing different corners will have identical port names. Other kinds of variations described through A\_to\_D subparameter lines should use unique port names. For example, a user may wish to create additional A\_to\_D converters for individual analog signals to monitor common mode behaviors on differential buffers.

At least one A\_to\_D line must be supplied corresponding to the “Typ” corner model. Other A\_to\_D lines for other corners may be omitted. In this case, the typical corner D\_to\_A entries will apply to all model corners.

IMPORTANT: measurements for receivers in IBIS may be conducted at the die pads or the pins. In such cases, the electrical input model data comprises a “load” which affects the waveform seen. However, for [External Circuit]s, the user may choose whether to measure the analog input response in the usual fashion or internal to the circuit (this does not preclude tools from reporting digital D\_receive and/or analog responses in addition to normal A\_signal response). If native IBIS measurements are desired, the A\_signal port would be named in the A\_to\_D line under port1. The A\_to\_D converter then effectively acts “in parallel” with the load of the circuit. If internal measurements are desired (e.g., if the user wishes to view the signal after processing by the receiver), the user-defined analog signal port would be named in the A\_to\_D line under port1. The A\_to\_D converter is effectively “in series” with the receiver model. The vhigh and vlow parameters should be adjusted appropriate to the measurement point of interest.

Note that, while the port assignments and SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) model data must be provided by the user, the D\_to\_A and A\_to\_D converters will be provided automatically by the tool. There is no need for the user to develop external SPICE, IBIS-ISS, Verilog-A(MS) or VHDL-A(MS) code specifically for these functions.

The [Diff Pin] keyword is NOT required for true differential [External Circuit] descriptions.

Pseudo-differential buffers are not supported under [External Circuit]. Use the existing [Model] and [External Model] keywords to describe these structures.

Note that the EDA tool is responsible for determining the specific measurement points for reporting timing and signal quality for [External Circuit]s.

In all other respects, [External Circuit] behaves exactly as [External Model].

*Examples:*

Example of Model B as an [External Circuit] using SPICE:

[External Circuit] BUFF-SPICE

Language SPICE

|

| Corner corner\_name file\_name circuit\_name (.subckt name)

Corner Typ buffer\_typ.spi bufferb\_io\_typ

Corner Min buffer\_min.spi bufferb\_io\_min

Corner Max buffer\_max.spi bufferb\_io\_max

|

| Parameters - Not supported in SPICE

|

| Ports List of port names (in same order as in SPICE)

Ports A\_signal int\_in int\_en int\_out A\_control

Ports A\_puref A\_pdref A\_pcref A\_gcref

|

| D\_to\_A d\_port port1 port2 vlow vhigh trise tfall corner\_name

D\_to\_A D\_drive int\_in my\_gcref 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_drive int\_in my\_gcref 0.0 3.0 0.6n 0.3n Min

D\_to\_A D\_drive int\_in my\_gcref 0.0 3.6 0.4n 0.3n Max

D\_to\_A D\_enable int\_en my\_gnd 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_enable int\_en my\_gnd 0.0 3.0 0.6n 0.3n Min

D\_to\_A D\_enable int\_en my\_gnd 0.0 3.6 0.4n 0.3n Max

|

| A\_to\_D d\_port port1 port2 vlow vhigh corner\_name

A\_to\_D D\_receive int\_out my\_gcref 0.8 2.0 Typ

A\_to\_D D\_receive int\_out my\_gcref 0.8 2.0 Min

A\_to\_D D\_receive int\_out my\_gcref 0.8 2.0 Max

|

| Note, the A\_signal port might also be used and int\_out not defined in

| a modified .subckt.

|

[End External Circuit]

Example [External Circuit] using IBIS-ISS:

[External Circuit] BUFF-ISS

Language IBIS-ISS

|

| Corner corner\_name file\_name circuit\_name (.subckt name)

Corner Typ buffer\_typ.spi bufferb\_io\_typ

Corner Min buffer\_min.spi bufferb\_io\_min

Corner Max buffer\_max.spi bufferb\_io\_max

|

| List of parameters

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

Parameters C1\_value

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

|

Converter\_Parameters MyVlow = 0.0

Converter\_Parameters MyVHigh = 3.3

Converter\_Parameters MyVinl = paramfile.par(TreeRootName(Model\_Specific(Vinl)))

Converter\_Parameters MyVinh = paramfile.par(TreeRootName(Model\_Specific(Vinh)))

Converter\_Parameters MyTrise = paramfile.par(TreeRootName(Model\_Specific(Trf)))

Converter\_Parameters MyTfall = paramfile.par(TreeRootName(Model\_Specific(Trf)))

|

| Ports List of port names (in same order as in ISS)

Ports A\_signal int\_in int\_en int\_out A\_control

Ports A\_puref A\_pdref A\_pcref A\_gcref

|

| D\_to\_A d\_port port1 port2 vlow vhigh trise tfall corner\_name

D\_to\_A D\_drive int\_in my\_gcref MyVlow MyVhigh MyTfall MyTrise Typ

D\_to\_A D\_enable int\_en my\_gnd 0.0 3.3 0.5n 0.3n Typ

D\_to\_A D\_enable int\_en my\_gnd 0.0 3.0 0.6n 0.3n Min

D\_to\_A D\_enable int\_en my\_gnd 0.0 3.6 0.4n 0.3n Max

|

| A\_to\_D d\_port port1 port2 vlow vhigh corner\_name

A\_to\_D D\_receive int\_out my\_gcref MyVinl MyVinh Typ

|

| Note, the A\_signal port might also be used and int\_out not defined in

| a modified .subckt.

|

[End External Circuit]

Example [External Circuit] using VHDL-AMS:

[External Circuit] BUFF-VHDL

Language VHDL-AMS

|

| Corner corner\_name file\_name entity(architecture)

Corner Typ buffer\_typ.vhd bufferb(buffer\_io\_typ)

Corner Min buffer\_min.vhd bufferb(buffer\_io\_min)

Corner Max buffer\_max.vhd bufferb(buffer\_io\_max)

|

| Parameters List of parameters

Parameters delay rate

Parameters preemphasis

|

| Ports List of port names (in same order as in VHDL-AMS)

Ports A\_signal A\_puref A\_pdref A\_pcref A\_gcref A\_control

Ports D\_drive D\_enable D\_receive

|

[End External Circuit]

Example [External Circuit] using Verilog-AMS:

[External Circuit] BUFF-VERILOG

Language Verilog-AMS

|

| Corner corner\_name file\_name circuit\_name (module)

Corner Typ buffer\_typ.v bufferb\_io\_typ

Corner Min buffer\_min.v bufferb\_io\_min

Corner Max buffer\_max.v bufferb\_io\_max

|

| Parameters List of parameters

Parameters delay rate

Parameters preemphasis

|

| Ports List of port names (in same order as in Verilog-AMS)

Ports A\_signal A\_puref A\_pdref A\_pcref A\_gcref A\_control

Ports D\_drive D\_enable D\_receive

|

[End External Circuit]

Example [External Circuit] using SPICE:

| Interconnect Structure as an [External Circuit]

|

|

[External Circuit] BUS\_SPI

Language SPICE

|

| Corner corner\_name file\_name circuit\_name (.subckt name)

Corner Typ bus\_typ.spi Bus\_typ

Corner Min bus\_min.spi Bus\_min

Corner Max bus\_max.spi Bus\_max

|

| Parameters - Not supported in SPICE

|

| Ports are in same order as defined in SPICE

Ports vcc gnd io1 io2

Ports int\_ioa vcca1 vcca2 vssa1 vssa2

Ports int\_iob vccb1 vccb2 vssb1 vssb2

|

| No A\_to\_D or D\_to\_A required, as no digital ports are used

|

[End External Circuit]

Example [External Circuit] using IBIS-ISS:

| Interconnect Structure as an [External Circuit]

|

|

[External Circuit] BUS\_SPI

Language IBIS-ISS

|

| Corner corner\_name file\_name circuit\_name (.subckt name)

Corner Typ bus\_typ.spi Bus\_typ

Corner Min bus\_min.spi Bus\_min

Corner Max bus\_max.spi Bus\_max

|

| List of parameters

Parameters sp\_file\_name

Parameters C1\_value R1\_value

|

| Ports are in same order as defined in IBIS-ISS

Ports vcc gnd io1 io2

Ports int\_ioa vcca1 vcca2 vssa1 vssa2

Ports int\_iob vccb1 vccb2 vssb1 vssb2

|

| No A\_to\_D or D\_to\_A required, as no digital ports are used

|

[End External Circuit]

Example [External Circuit] using VHDL-AMS:

[External Circuit] BUS\_VHD

Language VHDL-AMS

|

| Corner corner\_name file\_name entity(architecture)

Corner Typ bus.vhd Bus(Bus\_typ)

Corner Min bus.vhd Bus(Bus\_min)

Corner Max bus.vhd Bus(Bus\_max)

|

| Parameters List of parameters

Parameters r1 l1

Parameters r2 l2 temp

|

| Ports are in the same order as defined in VHDL-AMS

Ports vcc gnd io1 io2

Ports int\_ioa vcca1 vcca2 vssa1 vssa2

Ports int\_iob vccb1 vccb2 vssb1 vssb2

Example [External Circuit] using Verilog-AMS:

[External Circuit] BUS\_V

Language Verilog-AMS

|

| Corner corner\_name file\_name circuit\_name (module)

Corner Typ bus.v Bus\_typ

Corner Min bus.v Bus\_min

Corner Max bus.v Bus\_max

|

| Parameters List of parameters

Parameters r1 l1

Parameters r2 l2 temp

|

| Ports are in the same order as defined in Verilog-AMS

Ports vcc gnd io1 io2

Ports int\_ioa vcca1 vcca2 vssa1 vssa2

Ports int\_iob vccb1 vccb2 vssb1 vssb2

|

[End External Circuit]

The scope of the following keywords is limited to the [Component] keyword. They apply to the specific set of pin numbers and internal nodes only within that [Component].

*Keywords*: [Node Declarations], [End Node Declarations]

*Required:* Yes, if any [Circuit Call] in the [Component] references internal die nodes (including die pads or die bumps), except terminal names of any [Model] instances.

*Description:* Provides a list of internal die nodes, die pads, and/or die bumps for a [Component] to make unambiguous interconnection descriptions possible.

*Usage Rules*: All die node and die pad names that appear under any [Circuit Call] keyword within the same [Component], except terminal names of any [Model] instances, must be listed under the [Node Declarations] keyword. See the [Circuit Call] keyword for the list of reseved [Model] terminal names for making connections to the terminals of buffers described by the [Model] keyword.

If used, the [Node Declarations] keyword must appear before any [Circuit Call] keyword(s) under the [Component] keyword. Only one [Node Declarations] keyword is permitted for each [Component] keyword. Since the [Node Declarations] keyword is part of the [Component] keyword, all internal node, die pad, or die bump references apply only to that [Component] (i.e., they are local).

The internal die node, die pad, and/or die bump names within [Node Declarations] must be distinct from any pin names listed by the [Pin] keyword and from any reserved [Model] terminal name connections Each node and/or pad name must be separated by at least one white space. The list may span several lines and is terminated by the [End Node Declarations] keyword.

The names of die nodes and die pads can be composed of any combination of the legal characters outlined in Section 3.

*Example:*

[Node Declarations] | Must appear before any [Circuit Call] keyword

|

| Die nodes:

a b c | List of die nodes

|

| Die pads:

pad\_3a pad\_3b pad\_4 | List of die pads

|

[End Node Declarations]

*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 (including die pads or die bumps) and/or component pins.

*Sub-Params:* Signal\_pin, Diff\_signal\_pins, Series\_pins, 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 instantiates an [External Circuit] containing a buffer model, 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.

Port\_map:

The Port\_map subparameter is used to connect the ports of an [External Circuit] to die nodes (including die pads or die bumps) and/or component pins.

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.

The second argument of the Port\_map subparameter contains the name of a die node (including die pads and die bumps), [Model] instance terminal name, or a pin name. A pin name must match one of the pin names listed under the [Pin] keyword. A die node (including die pad and die bump) name, except terminal names of any [Model] instances, must match a name listed under the [Node Declarations] keyword. These die node (including die pad and die bump) names and pin names 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. Any of the [Model] instance terminal names may appear only once under a [Circuit Call], and may appear in only one [Circuit Call] keyword within a [Component]. Model instance names are reserved names and are formulated as follows:

signal:<PinName>

puref:<PinName>

pcref:<PinName>

pdref:<PinName>

gcref:<PinName>

extref:<PinName>

pos:<PinName>

neg:<PinName>

signal\_pos:<PinName>

signal\_neg:<PinName>

where <PinName> is the name of any pin listed under the [Pin] keyword of the [Component]. These nodes are counterparts to the reserved analog ports listed in Table 12. When the second argument of the Port\_map subparameter contains one of these model terminal names, the EDA tool will connect the appropriate terminal of the [Model] defined by the [Pin] keyword for the given <PinName> to the port of the [External Circuit] given by the first argument of the Port\_map subparameter. This replaces connections the EDA tool would otherwise make from a [Model] signal terminal to package models defined by the [Pin] or [Package Model] keywords, or from [Model] power supply reference terminals to power supply busses defined by the [Pin Mapping] keyword. [Circuit call] thus allows one or more [External Circuit]s to be placed between the component pins and the terminals of buffers described by the [Model] keyword, with or without an [External Model] under the [Model] keyword. Multiple [External Circuit]s may be connected in series between the component pins and buffer terminals by declaring die nodes, die pads, or die bumps with the [Node Declarations] keyword where [External Circuit]s may be joined.

Please note that pin names in the second argument may refer to actual pins defined in the [Pin] keyword or implicit pad or bump names, depending on the following conditions. When the [Circuit Call] instantiates an [External Ciruit] which contains the subparameter “Package\_Model”, the [External Ciruit] describes a package model located between the pin and the die of the component. In this case, a pin name in the second column of Port\_map refers to an actual pin declared in the [Pin] keyword of the component and the die-side connections of an [External Circuit] are all made to explicitly named die nodes (including die pads or die bumps) declared by the [Node Declarations] keyword or [Model] instance terminals. The package model may have an arbitrary circuit topology between the pins and die pads, die bumps or [Model] instance terminals, as long as there is a one to one mapping between the signal terminal of a [Model] instance and a pin. The supply terminals of [Model] instances may have arbitrary connectivity to die nodes (including die pads and die bumps) and/or component pins. When the [Circuit Call] instantiates an [External Circuit] which does not contain the subparameter “Package\_Model”, the [External Circuit] describes an interconnect located on the die. In this case, a pin name in the second column of Port\_map refers to an implicit die pad or die bump since native IBIS does not have a mechanism to declare die pads or die bumps explicitly. Connections to die pads or die bumps are made through their corresponding pin names (listed under the [Pin] keyword) with native IBIS package models where a one-to-one path between the die pads and pins is assumed.

QUESTIONS:

(done) (done) (this is OK because these connections are overridden by the [Circuit Call] connections.Reserved [Model] terminal names should be listed in the Overview section, perhaps in a new Table 13

Why are we allowing only one [Circuit Call] to refer to a model instance terminal name within a Component?

*Examples:*

NOTE REGARDING THIS EXAMPLE:

For the examples below please refer to Figure 29 and the example provided for the [Node Declarations] keyword.



1. - Reference Example for [Node Declarations] Keyword

[Circuit Call] A | Instantiates [External Circuit] named "A"

|

Signal\_pin 1

|

| mapping port pad/node

|

Port\_map A\_mypcr 10 | Port to implicit pad connection

Port\_map A\_mypur 10 | Port to implicit pad connection

Port\_map A\_mysig 1 | Port to implicit pad connection

Port\_map A\_mypdr 11 | Port to implicit pad connection

Port\_map A\_mygcr 11 | Port to implicit pad connection

Port\_map D\_mydrv nd1 | Port to internal node connection

|

[End Circuit Call]

|

|

|

[Circuit Call] C | Instantiates [External Circuit] named "C"

|

Signal\_pin 3

|

| mapping port pad/node

|

Port\_map A\_mypur a | Port to internal node connection

Port\_map A\_mysig b | Port to internal node connection

Port\_map A\_mypdr c | Port to internal node connection

Port\_map A\_mycnt pad\_3b | Port to explicit pad connection

|

[End Circuit Call]

|

|

[Circuit Call] D | Instantiates [External Circuit] named "D"

|

Signal\_pin 4

|

| mapping port pad/node

|

Port\_map A\_mypcr 10 | Port to implicit pad connection

Port\_map A\_mysig pad\_4 | Port to explicit pad connection

Port\_map A\_mygcr 11 | Port to implicit pad connection

Port\_map D\_receive nd1 | Port to internal node connection

|

[End Circuit Call]

|

|

[Circuit Call] On\_die\_interconnect | Instantiates [External Circuit] named

| "On\_die\_interconnect"

|

| mapping port pad/node

|

Port\_map vcc 10 | Port to implicit pad connection

2

Port\_map gnd 11 | Port to implicit pad connection

Port\_map o2 pad\_3a | Port to explicit pad connection

Port\_map vcca1 pcref:2 | Port connection to [Model] instance “B”

Port\_map vcca2 puref:2 | Port connection to [Model] instance “B”

Port\_map int\_ioa signal:2 | Port connection to [Model] instance “B”

Port\_map vssa1 pdref:2 | Port connection to [Model] instance “B”

Port\_map vssa2 gcref:2 | Port connection to [Model] instance “B”

Port\_map vccb1 a | Port to internal node connection

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

Port\_map vssb1 c | Port to internal node connection

|

[End Circuit Call]

|

|

[Circuit Call] PKG | Instantiates [External Circuit] named "PKG"

|

| mapping port pad/node

|

Port\_map gnd 11 | Port to pin connection

Port\_map vcc 10 | Port to pin connection

Port\_map pin3 3 | Port to pin connection

Port\_map pin4 4 | Port to pin connection

Port\_map pin5 5 | Port to pin connection

Port\_map pad3a pad\_3a | Port to explicit pad connection

Port\_map pad3b pad\_3b | Port to explicit pad connection

Port\_map pad4 pad\_4 | Port to explicit pad connection

Port\_map pc5 pcref:5 | Port connection to [Model] instance “E”

Port\_map pu5 puref:5 | Port connection to [Model] instance “E”

Port\_map pad5 signal:5 | Port connection to [Model] instance “E”

Port\_map pd5 pdref:5 | Port connection to [Model] instance “E”

Port\_map gc5 gcref:5 | Port connection to [Model] instance “E”

Port\_map ref5 extref:5 | Port connection to [Model] instance “E”

|

[End Circuit Call]