**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 187.2

**ISSUE TITLE:** Format and Usage Out Clarifications

**REQUESTOR:**  Michael Mirmak, Intel Corporation

**DATE SUBMITTED:** December 13, 2016

**DATE REVISED:** December 16, 2016; January 10, 2016

**DATE ACCEPTED:**

**DEFINITION OF THE ISSUE:**

The text of the Format portion of IBIS 6.1, Section 10.3 is written primarily from the perspective of Usage In or InOut. This should be modified to ensure that the meaning of Usage Out arguments, if permitted, is clear.

**SOLUTION REQUIREMENTS:**

The IBIS specification must meet these requirements:

Table 1: Solution Requirements

|  |  |
| --- | --- |
| Requirement | Notes |
| 1. The definition of parameters such as “Format” must be presented explicitly. |  |
| 1. The data path used for information associated with different Usage assignments must be presented clearly for each parameter, including “Format”. This must include “In”, “InOut”, and “Out” at a minimum, and preferably include “Dep”. |  |

**SUMMARY OF PROPOSED CHANGES:**

For review purposes, the proposed changes are summarized as follows:

Table 2: IBIS Keywords, Subparameters, AMI Reserved\_Parameters, and AMI functions Affected

|  |  |  |
| --- | --- | --- |
| Specification Item | New/Modified/Other | Notes |
| The definition of “Format” is added, along statements on the relationship between Usage “In”, “Out”, “InOut” and “Dep” for “Format”. Finally the correct data flow between various Usage settings, the EDA tool and the algorithmic model is clarified. | Other | The technical definitions and parser operation are unaffected by the proposed text changes. |

**PROPOSED CHANGES:**

*The introductory portion of “Format’, starting on page 186 of IBIS 6.1 Section 10.3 is proposed to be modified**from the following:*

**Format <data\_format>** <data>or **<data\_format>**<data>:

Required, except for the <data\_format> selection of Value as noted below. The word “Format” as part of the Format <data\_format> <data> sequence is optional. Valid entries for the <data\_format> and <data> fields are:

**Value** <value>

Single value data. The model maker may provide any value without any restrictions within the constraints of the Type of the variable. Note that Value and Default (defined below) are mutually exclusive, and shall not be used together for the same parameter.

**Range** <typ value> <min value> <max value>

This defines a continuous range for which the user may select any value greater than or equal to <min value> and less than or equal to <max value> within the constraints of the Type of the variable

**List** <default value> <value> <value> <value> ... <value>

This defines a discrete set of values from which the user may select one value

**List\_Tip** <default\_entry><entry><entry><entry>…<entry>

This is an optional leaf of a parameter with Format **List** and it is followed by a String entry for each entry in the **List**. The number of entries in List\_Tip must be the same as the number of entries in **List**. The nth entry in List\_Tip shall correspond to the nth entry in **List**. Quoted null entries are not permitted. All entries in List\_Tip shall be unique, except that if two entries in **List** are the same, then the corresponding List\_Tip entries must also be the same. List is required for List\_Tip to be entered, and the word Format before List\_Tip as in (Format List\_Tip ,,,) is not allowed.

Example:

(Strength (Usage In) (Type Integer) (Description "Strength of Driver")

(List 0 1 2 3 4) (Default 2)

(List\_Tip "Extra Weak" "Weak" "Nominal" "Strong" "Extra Strong"))

**Corner** <typ value> <slow value> <fast value>

Corner is not allowed with Usage Out parameters. The selection of one value is automatically carried out by the EDA tool based on its internal simulation corner setting

**Increment** <typ> <min> <max> <delta>

where min <= typ <= max and delta is always positive. After expansion, the expanded values of the parameter are typ + N\*delta where N is any positive or negative integer value provided by the EDA tool during the expansion process so that: min <= expanded values <= max

**Steps** <typ> <min> <max> <# steps>

Treat exactly like Increment with <delta> == (<max>-<min>)/<# steps>

**Table** and optional leaf **Labels**

The Format Table states that this parameter consists of one or more columns of data, with each row delimited by parentheses “(“ and “)”. All rows must contain the same number of entries (columns). At least one row shall be included. Default is illegal when Format Table is used.

The column entries shall be of Type Float, UI, Integer, String or Boolean.

Type Tap is illegal under Table. If only one Type is provided, then all Table entries shall be of the specified type.

(Type <type>)

For Table only, Type can also be used to designate the entries for each column. In this case, type entries shall be given for each column in the Table:

(Type <type1> <type2> <type3> ...)

Labels is an optional leaf within Table and it is followed by a String entry for each column in the Table. Quoted null entries are permitted. Labels shall be positioned immediately before the first row in a Table and are of the form:

(Labels <"label1"> <"label2"> <"label3"> ...)

If Table is used for a Reserved Parameter, the rules for the number of columns and their meaning are described in the Reserved\_Parameters section.

The EDA tool and the executable model file shall always transmit the entire contents of a table through the AMI\_parameters\_in or AMI\_parameters\_out string (defined in Section 10.2 and illustrated in the examples below). Only the parameter name and values in the table are included in the parameter string. The values in each row of the table are flattened into a single row of values without the parentheses surrounding each row when producing the parameter string.

For Usage Out and InOut, the number of rows returned by the executable model file may differ from the number of rows documented in the AMI parameter definition file, but a minimum of one row shall be returned. Multiple AMI\_GetWave calls are not required to return the same number of rows. For Usage Out, a one-row Table is required in the AMI parameter definition file to serve as a template for single and multi-row tables. This can be used by the EDA tool to reconstruct a sequence of data values returned by the executable model file into a table with as many rows as needed, and optionally for parameter initialization before being replaced by the actual Table data returned by the executable model file.

*…to (additions shown in red; removals not indicated):*

**Format <data\_format>** <data>or **<data\_format>**<data>:

Format defines the context or arrangement of the data being presented to the the EDA tool. For Usage In and Usage InOut, the EDA tool may accept data provided by the user according to the Format specified in the .ami file. Format is required, except for the <data\_format> selection of Value as noted below. The word “Format” as part of the Format <data\_format> <data> sequence is optional. Unless otherwise noted, Usage Out arguments are effectively ignored by EDA tools. However, Format may determine how Usage Out data is presented to the user by the EDA tool, particularly when data is returned by the executable model (for example, data of Format Table; see “Table” below). Data of Usage Dep, Usage Info or Usage Out shall not be passed to the executable model by the EDA tool, unlike data of Usage In or InOut, which shall always be passed to the executable model by the EDA tool.

Valid entries for the <data\_format> and <data> fields are:

**Value** <value>

Value consists of a single value of data. For Usage In and InOut, the model maker may provide any value without any restrictions within the constraints of the Type of the variable. Note that Value and Default (defined below) are mutually exclusive, and shall not be used together for the same parameter.

**Range** <typ value> <min value> <max value>

This defines a continuous range for which the user may select, for Usage In and InOut, any value greater than or equal to <min value> and less than or equal to <max value> within the constraints of the Type of the variable.

**List** <default value> <value> <value> <value> ... <value>

This defines a discrete set of values from which the user may select, for Usage In and InOut, one value.

**List\_Tip** <default\_entry><entry><entry><entry>…<entry>

This is an optional leaf of a parameter with Format **List** and it is followed by a String entry for each entry in the **List**. The number of entries in List\_Tip shall be the same as the number of entries in **List**. The nth entry in List\_Tip shall correspond to the nth entry in **List**. Quoted null entries are not permitted. All entries in List\_Tip shall be unique, except that if two entries in **List** are the same, then the corresponding List\_Tip entries shall also be the same. List is required for List\_Tip to be entered, and the word Format before List\_Tip as in (Format List\_Tip ,,,) is not allowed.

Example:

(Strength (Usage In) (Type Integer) (Description "Strength of Driver")

(List 0 1 2 3 4) (Default 2)

(List\_Tip "Extra Weak" "Weak" "Nominal" "Strong" "Extra Strong"))

**Corner** <typ value> <slow value> <fast value>

Corner is not allowed with Usage Out parameters. For Usage In and InOut, the selection of one value is automatically carried out by the EDA tool based on its internal simulation corner setting.

**Increment** <typ> <min> <max> <delta>

The Increment Format defines, for Usage In and InOut, a range of discrete integer values which can be swept by the EDA tool using a specified value (“delta”), where min <= typ <= max and delta is always positive. After expansion, the expanded values of the parameter are typ + N\*delta where N is any positive or negative integer value provided by the EDA tool during the expansion process so that: min <= expanded values <= max.

**Steps** <typ> <min> <max> <# steps>

The Steps Format operates exactly like Increment with <delta> == (<max>-<min>)/<# steps>

**Table** and optional leaf **Labels**

The Format Table consists of one or more columns of data, with each row delimited by parentheses “(“ and “)”. All rows shall contain the same number of entries (columns). At least one row shall be included. Default is illegal when Format Table is used.

The column entries shall be of Type Float, UI, Integer, String or Boolean.

Type Tap is illegal under Table. If only one Type is provided, then all Table entries shall be of the specified type.

(Type <type>)

For Table only, Type can also be used to designate the entries for each column. In this case, type entries shall be given for each column in the Table:

(Type <type1> <type2> <type3> ...)

Labels is an optional leaf within Table and it is followed by a String entry for each column in the Table. Quoted null entries are permitted. Labels shall be positioned immediately before the first row in a Table and are of the form:

(Labels <"label1"> <"label2"> <"label3"> ...)

If Table is used for a Reserved Parameter, the rules for the number of columns and their meaning are described in the Reserved\_Parameters section.

The EDA tool and the executable model shall always transmit the entire contents of a table through the AMI\_parameters\_in or AMI\_parameters\_out string (defined in Section 10.2 and illustrated in the examples below). Only the parameter name and values in the table are included in the parameter string. The values in each row of the table are flattened into a single row of values without the parentheses surrounding each row when producing the parameter string.

For Usage Out and InOut, the number of rows returned by the executable model may differ from the number of rows documented in the AMI parameter definition file, but a minimum of one row shall be returned. Multiple AMI\_GetWave calls are not required to return the same number of rows. For Usage Out, a one-row Table is required in the AMI parameter definition file to serve as a template for single and multi-row tables. This can be used by the EDA tool to reconstruct a sequence of data values returned by the executable model into a table with as many rows as needed, and optionally for parameter initialization before being replaced by the actual Table data returned by the executable model.

*The text of the “Combination and Corner Rules” section starting on page 200 of IBIS 6.1 is proposed to be modified from:*

COMBINATION AND CORNER RULES

For Usage Out parameters, ({Format} <data\_format> <data>) may be ignored by the EDA tool, except when <data\_format> is Table where at least a one-row Table is required in <data> to serve as a template for single and multi-row tables.

Formats Value, Corner and List can be of any defined Types whereas Formats Range, Increment and Steps can be of Types Float, UI, Integer and Tap only. Formats Gaussian, Dual-Dirac and DjRj can only be of Types Float and UI. For Format Table, the column entries shall be of Type Float, UI, Integer, String or Boolean. Type Tap is illegal for Format Table. If only one Type is provided, then all Table entries shall be of the specified type. Type can also be used to designate the entries for each column in the table. More information is provided in the definition of the Table format.

Note that modeling and simulating different corner cases is a fundamental concept in IBIS. For each model instance, the EDA tool will make use of either the "Typ", "Min" or "Max" data provided in the .ibs file, according to the user’s simulation setup.

As described in Section 9, "NOTES ON DATA DERIVATION METHOD" of this document, the "Min" and "Max" data for the I-V tables and their corresponding voltage reference keywords, [Ramp] and V-T tables represent the slow and fast behavior of the device, respectively. Following the conservative approach, the "Max" value of C\_comp represents the slow, and the "Min" value of C\_comp represents the fast behavior of the device.

For AMI parameters defined as Format Corner, the EDA tool will pick one of the three supplied values (<typ value>, <slow value>, <fast value>) in the AMI parameter definition file for any given model instance. This selection is governed by the same internal corner variable in the EDA tool that controls the selection of the "Typ", "Min", "Max" model data. <typ value> corresponds to "Typ", <slow value> corresponds to "Min" (slow or weak performance) and <fast value> corresponds to "Max" (fast or strong performance). For AMI parameters, <slow value> does not have to be less than <fast value>.

For AMI parameter Types “Range”, “Increment” and “Steps” <min value>, <max value> does not imply slow and fast corners, and the user may select any value provided by these parameters regardless of what corner is used for the simulation. If the user does not make a selection for parameter types “Range”, “List”, “Increment” and “Steps”, the EDA tool shall automatically use the value defined by Default, if it exists, or the <typ value> otherwise (regardless of what corner is used for the simulation).

When a [Model] that is associated with any of the pins listed under the [Diff Pin] keyword contains the [Algorithmic Model] keyword, the tdelay\_\*\*\* parameters in the fourth, fifth and sixth columns of the [Diff Pin] keyword are ignored in AMI channel characterization simulations, i.e., they are treated as if their value would be zero.

Table 17 summarizes the relationships between the different Format and Data Types for Reserved or Model Specific Parameters.

*…to (additions shown in red; removals not indicated):*

COMBINATION AND CORNER RULES

For Usage Out parameters, ({Format} <data\_format> <data>) may be ignored by the EDA tool where not prohibited, except when <data\_format> is Table. In this case, a Table of at least one row is required in <data> to serve as a template for single and multi-row tables (see “Format” and “Table” descriptions above).

Formats Value, Corner and List can be of any defined Types whereas Formats Range, Increment and Steps can be of Types Float, UI, Integer and Tap only. Formats Gaussian, Dual-Dirac and DjRj can only be of Types Float and UI. For Format Table, the column entries shall be of Type Float, UI, Integer, String or Boolean. Type Tap is illegal for Format Table. If only one Type is provided, then all Table entries shall be of the specified Type. Type can also be used to designate the entries for each column in the table. More information is provided in the definition of the Table format.

Note that modeling and simulating different corner cases is a fundamental concept in IBIS. For each model instance, the EDA tool will make use of either the "Typ", "Min" or "Max" data provided in the .ibs file, according to the user’s simulation setup.

As described in Section 9, "NOTES ON DATA DERIVATION METHOD" of this document, the "Min" and "Max" data for the I-V tables and their corresponding voltage reference keywords, [Ramp] and V-T tables represent the slow and fast behavior of the device, respectively. Following the conservative approach, the "Max" value of C\_comp represents the slow, and the "Min" value of C\_comp represents the fast behavior of the device.

For Usage In and Usage InOut AMI parameters defined as Format Corner, the EDA tool shall pick one of the three supplied values (<typ value>, <slow value>, <fast value>) in the AMI parameter definition file for any given model instance. This selection is governed by the same internal corner variable in the EDA tool that controls the selection of the "Typ", "Min", "Max" model data. <typ value> corresponds to "Typ", <slow value> corresponds to "Min" (slow or weak performance) and <fast value> corresponds to "Max" (fast or strong performance). For AMI parameters, <slow value> does not have to be less than <fast value>.

For AMI parameter Formats “Range”, “Increment” and “Steps” <min value>, <max value> does not imply slow and fast corners, and for Usage In and Usage InOut the user may select any value provided by these parameters regardless of what corner is used for the simulation. If the user does not make a selection for parameter Formats “Range”, “List”, “Increment” and “Steps”, the EDA tool shall automatically use the value defined by Default, if it exists, or the <typ value> otherwise (regardless of what corner is used for the simulation).

When a [Model] that is associated with any of the pins listed under the [Diff Pin] keyword contains the [Algorithmic Model] keyword, the tdelay\_\*\*\* parameters in the fourth, fifth and sixth columns of the [Diff Pin] keyword are ignored in AMI channel characterization simulations, i.e., they are treated as if their value would be zero.

Table 17 summarizes the relationships between the different Format and Data Types for Reserved or Model Specific Parameters.

**BACKGROUND INFORMATION/HISTORY:**

The need for this BIRD was explained in BUG 183 (<http://www.ibis.org/bugs/ibischk/bug183.txt>). The document was reviewed and contains edits from the participants in IBIS-ATM meetings held in November and December, 2016. The document was submitted to the IBIS Open Forum and classified as BIRD 187 on December 13, 2016.

BIRD 187.1 was submitted to highlight overall changes to the text, add Usage In and Usage InOut details for Increment, as well as to add the rest of the (unchanged) text for Table. This unchanged text contains information on the treatment of Table for Usage In and Usage InOut. Finally, minor clarifications are made to the text in “Combination and Corner Rules” regarding Usage In and Usage InOut.

BIRD 187.2 was submitted to incorporate editorial improvements suggested by Curtis Clark and Radek Biernacki, including extraneous uses of the word “Type” and the incorrect phrase “executable model file”.