**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** (Draft 1)

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

**REQUESTOR:**  Michael Mirmak, Intel Corporation

**DATE SUBMITTED:** Draft – Nov. 1, 2016

**DATE REVISED:**

**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 always convey its technical syntax requirements, and the context using that syntax, in a clear manner.

**SUMMARY OF PROPOSED CHANGES:**

Only portions of the text of Format in Section 10.3 of IBIS 6.1 are modified, for clarity. The technical definitions and parser operation are unaffected by the proposed changes.

**PROPOSED CHANGES:**

*IBIS 6.1 Section 10.3 is proposed to be modified as shown below in red.*

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

Format defines the context or arrangement of the data being passed to the executable model file by the EDA tool. For Usage In and InOut, the EDA tool may accept data provided by the user according to the Format selected. 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, specific Usage Out arguments or data provided as Format are effectively ignored and will be simply passed by the EDA tool to the executable model file. However, Format may determine how data received by the EDA tool from the executable model file is presented to the user. 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 ~~must~~ 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 ~~must~~ 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 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 ~~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~~ shall contain the same number of entries (columns). At least one row shall be included. Default is illegal when Format Table is used.

**BACKGROUND INFORMATION/HISTORY:**

The need for this BIRD was explained in BUG 183.