**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**Draft 6**

**ISSUE TITLE:** IBIS-AMI New Reserved Parameters for Dependency Tables

**REQUESTOR:**  Walter Katz, Todd Westerhoff, SiSoft, Adge Hawes, IBM

**DATE SUBMITTED:** *September 24, 2012*

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

Model developers and EDA vendors building IBIS-AMI models using the IBIS 5.0 specification have come across a number of modeling issues that are not addressed in IBIS 5.0. In order to deliver models and EDA tools that meet end-user demands for model accuracy and functionality, EDA vendors have defined "extensions" to add new capabilities to IBIS-AMI models. Unfortunately, EDA vendors have had to use proprietary (and different) syntax to add these capabilities to models, limiting model portability between different EDA tools.

This BIRD proposes new syntax for the .ami parameter file that improves model functionality and accuracy. Including this syntax in the IBIS standard will allow creation of accurate, compliant IBIS-AMI models that are readily portable between commercial EDA simulators.

Dependency Tables are Model\_Specific branches that are defined with format Table, and Usage Dependency\_Table.

Dependency Tables define a relationship between one or more input (independent) parameters and one or more output (dependent) parameters. Note that in this context, "input parameter" and "output parameter" refer to inputs and outputs of the relationship being defined, not to whether the parameters in question are used as inputs or outputs by the algorithmic model itself. Dependency Tables allow complex relationships between related parameters to be programatically defined.

Dependency Tables are not passed to the AMI\_Init function in \*\*AMI\_parameters\_in.

**ANY OTHER BACKGROUND INFORMATION:**

This BIRD is being requested by the following IBIS users and model developers, in conjunction with the authors:

Cisco Systems: Upen Reddy, Doug White

ANALYSIS PATH/DATA THAT LED TO SPECIFICATION

The new Dependency Table defined in this BIRD came from commercial IBIS-AMI model development efforts where new functionality was needed to meet customer expectations for model functionality, accuracy and performance.

## New Usage Dependency\_Table

Dependency Tables are used to specify relationships between different AMI parameters. This is desirable because sometimes the value to be used for an AMI parameter is dependent on the selected value of another AMI parameter. For example, consider the case of an output buffer with a user-selectable output strength setting. The strength selected by the user affects both the output’s voltage swing and impedance.

Let’s assume that the output strength is controlled by a Model\_Specific parameter "Tx\_Strength" that has 8 discrete settings from 0 to 7, with a nominal value of 4:

(Tx\_Strength (Usage In)(Type Integer)(Range 4 0 7)

 (Description "Output buffer strength setting")

)

The Tx analog buffer’s impedance is specified using the equivalent circuit analog model parameters. The output impedance values for each of the 8 output strength settings are:

(Rs (Usage Info)(Type Float)

 (List 45.0 46.0 48.0 50.0 52.0 50.0 48.0 45.0)

 (Description "Tx output impedance")

)

The Tx analog buffer’s output swing is also specified using the equivalent circuit analog model parameters. The output voltage values for each of the 8 output strength settings are:

(Voh (Usage Info)(Type Float)

 (List 0.40 0.42 0.44 0.46 0.48 0.50 0.52 0.54)

 (Description "Tx output voltage")

)

The model user is to be presented with a single control parameter, Tx\_Strength, with the values of Rs and Voh automatically selected by the EDA tool based on the value the user specifies for Tx\_Strength. Dependency Tables specify relationships between parameters to make this possible.

For the example just described, one possible Dependency Table declaration is:

(Tx\_Strength\_Dependency

(Usage Dependency\_Table)

(Description “Rs and Voh dependency on Tx\_Strength”)

(Parameter\_Names Tx\_Strength Rs Voh)

(Column\_Types In Out\_Match Out\_Match)

(Table

(0 45.0 0.40)

 (1 46.0 0.42)

 (2 47.0 0.44)

 (3 50.0 0.46)

 (4 52.0 0.48)

 (5 50.0 0.50)

 (6 48.0 0.52)

 (7 45.0 0.54)

)

)

This tells the EDA tool that a Tx\_Strength setting of

0 corresponds to Rs = 45.0 ohms, Voh = 0.40V

1 corresponds to Rs = 46.0 ohms, Voh = 0.42V, and so on.

Note that Type is not required to specify the column types in the Dependency Table since the Type is automatically determiniuned by the Type of the Parameter assigned to each Column in the Parameter\_Names leaf.

The following rules apply to the use of Dependency Tables:

1. The Parameter names used in the Dependency Table must already be declared in the .ami file, either in the Reserved\_Parameter or Model\_Specific sections
2. The values listed in the table for each Parameter must be a legal Format for that Parameter, and should be a legal value based on the definition of the Parameter in the .ami file.
3. Dependency Tables can only be declared in the Model\_Specific section of the .ami file
4. A Dependency Table defines the relationship between one or more input parameters and one or more output parameters.
5. Two or more Dependency Tables can be linked together (i.e. output variables in one table can be used as input variables in another table) to specify complex relationships
6. Dependency Tables are evaluated in the order they are declared in the .ami file.
7. Dependency Tables are not passed to the AMI\_Init function in \*\*AMI\_parameters\_in.

The following elements define a Dependency Table:

• Usage is Dependency\_Table

• Parameter\_Names Parameter name of each column

• Column\_Types Column usage declaration

• Data rows in Table Format

Parameter\_Names declares the parameters used in the table. Column\_Types declares whether the parameters are inputs or outputs to the relationship being defined. An independent parameter is defined as an input to the table if its column type is In. A dependent parameter is defined as an output from the table if its column type is Out\_Match, Out\_Closest, Out\_Range or Out\_PWL. The independent parameters shall be in the first columns followed by the dependent parameters. Column\_Types must either be all Out\_Match, or any combination of Out\_Closest, Out\_Range or Out\_PWL. If the Column\_Types are a combination of Out\_Closest, Out\_Range or Out\_PWL, the last independent column must be of Type Float, UI, Integer or Tap.

If all of the dependent paramamers have Column\_Type Out\_Match, then the dependent parameters shall have the values which are in the row that matches the values of all of the independent parameters. If there is no such row, then the default value of each dependent parameter shall be used.

(Note that two floating point values “Match” if they are the same with a precision of six decimal places)

If the dependent parameters have Column\_Types Out\_Closest, Out\_Range or Out\_PWL the EDA tool shall first generate a “Subset” of all rows that match all independent parameters except the last independent parameter column. Each row has a numeric “Row Value” defined as the value of the last independent parameter column. Let “Parameter Value” be the value of the parameter in the last independent parameter column.

If the “Parameter Value” matches the “Row Value” of one of the rows in the “Subset” then

The values of the dependent values in that row shall be used.

Else

The “Greatest Lower Bound Row” is the Row in the “Subset” that has the largest “Row Value” that is less than the “Parameter Value”.

The “Least Upper Bound Row” is the Row in the “Subset” that has the smallest “Row Value” that is greater than the “Parameter Value”.

If there is no “Greatest Lower Bound Row” then

The dependent values in the “Least Upper Bound Row” shall be used.

Elseif there is no “Least Upper Bound Row” then

The dependent values in the “Greatest Lower Bound Row” shall be used.

Elseif the dependent parameter is Out\_Closest

DLUB == “Row Value” of the “Least Upper Bound Row” minus “Parameter Value”

DGLB == “Parameter Value” minus “Row Value” of the “Greatest Lower Bound Row”

If DLUB <= DGLB then

The dependent value in the “Least Upper Bound Row” shall be used.

Else

The dependent value in the “Greatest Lower Bound Row” shall be used.

Elseif the dependent parameter is Out\_Range

The dependent value in the “Least Upper Bound Row” shall be used.

Elseif the dependent parameter is Out\_PWL

The dependent value shall be calculated using PieceWise Linear interpolation.

If the the dependent parameter is Fromat List, Increment, or Steps then

The closest allowed values for that parameter shall be used.

Elseif the dependent parameter is Type Integer then

The value shall be rounded to the nearest integer

(Note that Out\_PWL is not allowed for Type String or Boolean)

Example:

(Tx\_Strength (Usage In)(Type Integer)

 (List 0 1 2 3 4 5 6 7

 (Description "Output buffer strength setting")

)

(Rs (Usage Info)(Type Float)

 (List 45.0 46.0 48.0 50.0 52.0 50.0 48.0 45.0)

 (Description "Tx output impedance")

)

(Voh (Usage Info)(Type Float)

 (List 0.40 0.42 0.44 0.46 0.48 0.50 0.52 0.54)

 (Description "Tx output voltage")

)

(Tx\_Strength\_Dependency (Usage Dependency\_Table)

(Description “Rs and Voh dependency on Tx\_Strength”)

(Parameter\_Names Tx\_Strength Rs Voh)

(Column\_Types In Out\_Match Out\_Match)

(Table

(0 45.0 0.40)

(1 46.0 0.42)

(2 47.0 0.44)

(3 50.0 0.46)

(4 52.0 0.48)

(5 50.0 0.50)

(6 48.0 0.52)

(7 45.0 0.54)

)

)

Example:

(Tx\_Strength (Usage In)(Type Integer)(Range 35 0 70)

(Description "Output strength"))

(Rs (Usage Info)(Type Float)(Range 50 45 55)

(Description "Tx output impedance"))

(Voh (Usage Info)(Type Float)(Range .5 .45 .55)

(Description "Tx output voltage"))

(Tstonefile(Usage Info)(Type String)

(Description “On-Die S-Parameter File”)

(List “A.s4p” “B.s4p” “C.s4p” “D.s4p” “E.s4p” “F.s4p”

 “G.s4p”)

(Default “D.s4p”)

(Tx\_Strength\_Dependency (Usage Dependency\_Table)

(Description

 “Rs, Voh and Tstonefile dependency on Tx\_Strength”)

(Parameter\_Names Tx\_Strength Rs Voh Tstonefile)

(Column\_Types In Out\_PWL Out\_Range Out\_Closest)

(Table

 (5 46.0 0.46 “A.s4p“)

 (15 48.0 0.48 “B.s4p“)

 (25 49.0 0.49 “D.s4p“)

 (35 50.0 0.50 “D.s4p“)

 (45 51.0 0.51 “E.s4p“)

 (55 53.0 0.53 “F.s4p“)

 (65 54.0 0.54 “G.s4p“)

)

)

For the following values of Tx\_Strength, the values of the dependent parameters are:

 (0 45.0 0.46 “A.s4p“)

 (5 46.0 0.46 “A.s4p“)

 (6 46.2 0.46 “A.s4p“)

 (10 47.0 0.46 “B.s4p“)

 (14 47.8 0.46 “B.s4p“)

 (70 55.0 0.54 “G.s4p“)

Simulation Reserved Parameters

Simulation Reserved Parameters are not defined in the Reserved\_Parameter section, but may be used as Parameter\_Names in a Dependency Table. Simulation Reserved Parameters may not be Model\_Specific parameters and must have Column\_Types In.

The Simulation Reserved Parameters are:

Corner

The selection of one value is automatically carried out by the EDA tool based on its internal simulation corner setting for IBIS [Model] used to invoke the AMI model

Type is String, and values are “Typ”, “Min” or “Max”

bit\_time

Data Rate, contains the same value as the bit\_time argument of the AMI\_Init function

Type is Float

BAUD

1.0/bit\_time

Type is Float

Model\_Name

The IBIS [Model] Model Name used to invoke the AMI model

Type is String

An example using Simulation Reserved Parameters

(Tstonefile\_Dependency

(Usage Dependency\_Table)

(Parameter\_Names Corner Model\_Name BAUD Tstonefile)

(Column\_Types In In In Out\_Closest)

(Table

 (“Typ” “Tx\_75” 5.0e9 “Typ\_75\_5Gbps.s4p”)

 (“Typ” “Tx\_75” 15.0e9 “Typ\_75\_15Gbps.s4p”)

 (“Typ” “Tx\_50” 5.0e9 “Typ\_50\_5Gbps.s4p“)

 (“Typ” “Tx\_50” 15.0e9 “Typ\_50\_15Gbps.s4p“)

 (“Min” “Tx\_75” 5.0e9 “Min\_75\_5Gbps.s4p“)

 (“Min” “Tx\_75” 15.0e9 “Min\_75\_15Gbps.s4p“)

 (“Min” “Tx\_50” 5.0e9 “Min\_50\_5Gbps.s4p“)

 (“Min” “Tx\_50” 15.0e9 “Min\_50\_15Gbps.s4p“)

 (“Max” “Tx\_75” 5.0e9 “Max\_75\_5Gbps.s4p“)

 (“Max” “Tx\_75” 15.0e9 “Max\_75\_15Gbps.s4p“)

 (“Max” “Tx\_50” 5.0e9 “Max\_50\_5Gbps.s4p“)

 (“Max” “Tx\_50” 15.0e9 “Max\_50\_15Gbps.s4p“)

)

)