

# **GetWave\_Exists Definition (BIRD 120)**

**IBIS-ATM Teleconference  
March 13, 2012**

# Problem statement

- **An AMI model was found with an AMI\_GetWave function in the DLL, but its .ami GetWave\_Exists parameter was “False”**
  - the model maker intended this to be a debugging feature for the model
- **The v5.0 IBIS specification does not have a definition for what the meaning of this parameter is**
  - intuitive reasoning and reading between the lines would interpret the parameter as an indicator for the content of the DLL (that’s obvious, isn’t it?)
  - this is completely unnecessary and redundant, as it is very easy to test for the function “entry points” in a DLL programmatically (does this parameter have another purpose in life?)
  - people would like to be able to turn the GetWave function on/off
  - people would like to have the option of using a modeling template with all AMI functions defined, but not all having actual code “implementation”
- **The v5.1 IBIS specification draft (based on BIRD 120.1) is poorly worded**
  - the flow is described in terms of “If GetWave\_Exists is True/False...”
  - does this mean that the EDA tool should make decisions based on this parameter and not based on the content of the DLL?
  - this can tell the tool to execute a non existent function (obviously we will never do that...)
  - Definition says: “Tells EDA platform whether the AMI\_GetWave is implemented in this model” (what does “implemented” really mean?)

# Goal

- **Find a better wording for the definition of GetWave\_Exists**
  - the specification has to be clear so that model makers and EDA tool vendors don't make their own independent and potentially conflicting assumptions
  - this can save us from having to answer questions about this issue later
- **Find a wording with a minimal change so that this could be handled as an “editorial correction” without a new BIRD**
  - the BIRD approval deadline has passed for IBIS v5.1

# Questions

- **Should the corrected verbiage state that GetWave\_Exists must reflect the content of the DLL, or should it allow for it to be used as “flow control”?**
  - the Definition in the IBIS v5.1 draft says: “Tells EDA platform whether the AMI\_GetWave is implemented in this model”
  - what does “implemented” really mean?
    - **function entry point “exists” in the DLL? (don’t dare to execute it if not)**
    - **meaningful code inside the function exists? (don’t bother executing it if not)**
    - **what if meaningful code exists in the function, but it was intended to be used only for debugging?**
  - are we changing the original meaning of the parameter?
  - if so, shouldn’t its name be different?
  - is a name change justifiable as an editorial change?
  - is a name change breaking existing models?
  - deprecate this parameter and introduce a new parameter with a new meaning?

# Proposals

- **Change the “Definition” and “Usage Rules” in the v5.1 specification:**

*Definition:* Tells EDA platform whether the AMI\_GetWave **function has useful code** ~~is implemented~~ in this **algorithmic** model

*Usage Rules:* Note that if Init\_Returns\_Impulse is set to ‘False’, then **the GetWave function MUST contain useful code in the algorithmic model and consequently** GetWave\_Exists MUST be set to ‘True’.

- **This doesn’t “twist” the meaning of the parameter too much**
  - meaningful code *exists* True/False
  - meaningful code is worth executing True/False
  - if meaningful code doesn’t exist, don’t bother executing it...
  - doesn’t force the EDA tool to execute a non-existent function
  - it tells the model maker that they better implement useful code in AMI\_GetWave if the didn’t in AMI\_Init

# Proposals (cont'd)

- **Change the “Definition” and “Usage Rules” in the v5.1 specification:**

*Definition:* Tells EDA platform whether the AMI\_GetWave **function has useful code** ~~is implemented~~ in this **algorithmic** model

*Usage Rules:* **If an entry point doesn't exist for the AMI\_GetWave function in the algorithmic model, the value of this parameter must be False and the meaning of this parameter is AMI\_GetWave function does not exist. If an entry point exists for the AMI\_GetWave function in the algorithmic model, the value of this parameter may be True or False, instructing the EDA tool whether or not to execute the AMI\_GetWave function. In this case the meaning of this parameter is Use\_AMI\_GetWave\_Function.**

Note that if Init\_Returns\_Impulse is set to 'False', then **the GetWave function MUST contain useful code in the algorithmic model and consequently** GetWave\_Exists **MUST** be set to 'True'.

# Does the flow need clarification?

- | Step 5. If Tx GetWave\_Exists is True **and the GetWave function exists in the Tx algorithmic model**, the output of Step 4 is presented to the Tx model's AMI\_GetWave function and the Tx AMI\_GetWave function is executed. The output of the Tx AMI\_GetWave function is passed on to Step 6.
- | Step 6a. If Tx GetWave\_Exists is True **and the GetWave function exists in the Tx algorithmic model**, and Rx GetWave\_Exists is True **and the GetWave function exists in the Rx algorithmic model**, the output of Step 5 is convolved with the output of Step 1 by the simulation platform and the result is passed on to Step 7.
- | Step 6b. If Tx GetWave\_Exists is False and Rx GetWave\_Exists is True **and the GetWave function exists in the Rx algorithmic model**, the output of Step 4 is convolved with the output of Step 2 by the simulation platform and the result is passed on to Step 7.
- | Step 6c. If Tx GetWave\_Exists is False and Rx GetWave\_Exists is False, the output of Step 4 is convolved with the output of Step 3 by the simulation platform and the result is passed on to Step 8.
- | Step 6d. If Tx GetWave\_Exists is True **and the GetWave function exists in the Tx algorithmic model**, and Rx GetWave\_Exists is False, the output of Step 5 is convolved with the output of Step 1 and the Impulse Response of the Rx filter by the simulation platform and the result is passed on to Step 8. (The Impulse Response of the Rx filter may be obtained by deconvolving the output of Step 3 by the input of Step 3).
- | Note: For the scenario where the Tx AMI\_Init function does NOT include equalization effects (i.e. does not modify the impulse response of the channel), Step 6d is functionally equivalent to simply convolving the output of Step 5 with the output of Step 3.
- | Step 7. If Rx GetWave\_Exists is True **and the GetWave function exists in the Rx algorithmic model**, the output of Step 6 is presented to the Rx model's AMI\_GetWave function and the Rx AMI\_GetWave function is executed. The output of the Rx AMI\_GetWave function is passed on to Step 8.

# Summary

- **We need to make a decision today on this and let the editorial committee know how to edit the text in this area in the IBIS v5.1 draft**