**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 210

**ISSUE TITLE:** New Redriver AMI Flow

**REQUESTOR:**  Fangyi Rao, Keysight Technologies

**DATE SUBMITTED:** February 19, 2021

**DATE REVISED:**

**DATE ACCEPTED:** Rejected April 22, 2022

**DEFINITION OF THE ISSUE:**

The current Repeater flow is known to have following issues.

1. The cumulative upstream impulse response of the Redriver channel is not provided to the terminal Rx (including Retimer Rx) in AMI\_Init. As a result, when the terminal Rx has DFE, the end-to-end cumulative impulse response of the Redriver channel needed in statistical simulations is not available.
2. The cumulative upstream impulse response of the Redriver channel is not provided to either Tx or Rx in AMI\_Init. As a result, the AMI\_Init function cannot perform optimization on the upstream signal.
3. The combination of Tx GetWave model and Rx Init-only model leads to deconvolution in time domain simulations.

This BIRD proposes a new repeater flow, in which certain requirements are imposed on models, to address these issues. A new reserved parameter and new data in the impulse matrix are introduced to support the proposed flow.

**SOLUTION REQUIREMENTS:**

The IBIS specification must meet these requirements:

Table 1: Solution Requirements

|  |  |
| --- | --- |
| Requirement | Notes |
| 1. Support statistical simulations on Redriver channels whose terminal Rx (including Retimer Rx) has DFE.
 |  |
| 1. Allow AMI\_Init to perform optimization on the upstream signal.
 |  |
| 1. Eliminate the need for deconvolution in simulations.
 |  |

**SUMMARY OF PROPOSED CHANGES:**

Add two columns at the end of impulse\_matrix in AMI\_Init.

Add new Boolean Reserved Parameter Use\_v7p1\_AMI\_Flow.

In the new flow all models are required to support parameter Use\_v7p1\_AMI\_Flow in the AMI\_Init function. In the new flow’s time domain simulations terminal Rx (including Retimer Rx) models with DFE are required to have the AMI\_GetWave function.

**PROPOSED CHANGES:**

Add the following paragraph to Section 10.2.3.

Version 7.1 introduces an alternative Redriver simulation flow. In this flow (indicated by Use\_v7p1\_AMI\_Flow=True), the EDA tool is responsible to add two additional columns at the end of the input impulse\_matrix and to fill the first additional column with the cumulative upstream impulse response from the terminal Tx (including Retimer Tx) to the algorithmic model’s input node. The second additional column in the input impulse\_matrix is a placeholder. The model’s AMI\_Init function is responsible to:

1. Modify the through channel column of impulse\_matrix in place by applying its gain and equalization (excluding DFE if it exists) to the column
2. Modify the crosstalk channel columns of impulse\_matrix in place by applying its gain and equalization (excluding DFE if it exists) to the columns
3. Modify the first additional column of impulse\_matrix in place by applying its gain and equalization (including DFE if it exists) to the column
4. Filled the second additional column of impulse\_matrix in place with the filter impulse response of its gain and equalization (excluding DFE if it exists).

Add the following new parameter.

*Parameter:* **Use\_v7p1\_AMI\_Flow**

*Required:* No, and illegal before AMI\_Version 7.1

*Direction:* Tx,Rx

*Descriptors:*

Usage: In

Type:                     Boolean

Format: List

Default:                 <Boolean\_literal>

Description:*<*string>

*Definition:* The List must contain both True and False. The EDA tool is responsible to set the value and pass it to the model in the AMI\_Init function call to inform the model whether the new flow or the old flow is being executed. The EDA tool can set the value to either True or False in the AMI\_Init function call.

*Usage Rules:* A model that specifies this parameter must also specify Init\_Returns\_Impulse=True. When Use\_v7p1\_AMI\_Flow=False, the model’s AMI\_Init function behaves as specified in the 7.0 Specification. When Use\_v7p1\_AMI\_Flow=True, the model’s AMI\_Init function modifies impulse\_matrix as specified in Section 10.2.3. If this parameter is not present in the AMI\_parameters\_in string, the model assumes that Use\_v7p1\_AMI\_Flow=False.

*Other Notes:*

*Example:*

(Use\_v7p1\_AMI\_Flow (Usage In) (Type Boolean) (Value True)

(Description "The model supports the new Redriver flow"))

Add the following paragraphs to Section 10.8.1 before the paragraph that starts with “Since the Redriver output signal is driven continuously by …”.

**New Redriver Time Domain Flow**

For Redriver channels where all models specify parameter Use\_v7p1\_AMI\_Flow and the terminal Rx has AMI\_GetWave if the model has DFE, an alternative Redriver time domain simulation flow is defined below.

Step 1. The EDA tool obtains the impulse response of the upstream analog channel, which represents the combined impulse response of Tx1’s analog model, physical channel 1, and Rx1’s analog model.

Step 2. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Tx1. The output of step 1 and the value of Use\_v7p1\_AMI\_Flow are presented to Tx1’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Tx1’s AMI\_Init function is executed. Note that because Tx1 is a terminal Tx, the cumulative upstream impulse response in the input impulse\_matrix can be a unit impulse response, whose first value equals 1/sample\_interval and the others equal zero. The EDA tool may use the filter impulse response in the last column of the output impulse\_matrix to construct an AMI\_GetWave function for Tx1 if it does not have one.

Step 3. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Rx1. The output of step 2 and the value of Use\_v7p1\_AMI\_Flow are presented to Rx1’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Rx1’s AMI\_Init function is executed. The EDA tool may use the filter impulse response in the last column of the output impulse\_matrix to construct an AMI\_GetWave function for Rx1 if it does not have one.

Step 4. The EDA tool obtains the impulse response of the downstream analog channel, which represents the combined impulse response of Tx2’s analog model, physical channel 2, and Rx2’s analog model.

Step 5. The EDA tool uses the output of step 3 to construct the cumulative upstream impulse response for Tx2.

Step 6. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Tx2. The output of steps 4 and 5 and the value of Use\_v7p1\_AMI\_Flow are presented to Tx2’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Tx2’s AMI\_Init function is executed. The EDA tool may use the filter impulse response in the last column of the output impulse\_matrix to construct an AMI\_GetWave function for Tx2 if it does not have one.

Step 7. The EDA tool uses the output of steps 4 and 6 to construct the cumulative upstream impulse response for Rx2.

Step 8. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Rx2. The output of steps 6 and 7 and the value of Use\_v7p1\_AMI\_Flow are presented to Rx2’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Rx2’s AMI\_Init function is executed. The EDA tool may use the filter impulse response in the last column of the output impulse\_matrix to construct an AMI\_GetWave function for Rx2 if it does not have one.

Step 9. The EDA tool performs simulation on the upstream channel, which consists of Tx1, physical channel 1, and Rx1, according to the AMI flow defined in the specification for channels without Repeaters. Note that deconvolution can be avoided by using the AMI\_GetWave function constructed by the EDA tool for Init-only Rx1 in step 3.

Step 10. The EDA tool uses the signal waveform at the output end of Rx1’s algorithmic model in step 9 as the stimulus of Tx2’s algorithmic model and performs simulation on the downstream channel, which consists of Tx2, physical channel 2 and Rx2, according to the AMI flow defined in the specification for channels without Repeater. Note that deconvolution can be avoided by using the AMI\_GetWave function constructed by the EDA tool for Init-only Rx2 in step 8.

Step 11. The EDA tool calls the AMI\_Close function of each algorithmic model in Tx1, Rx1, Tx2 and Rx2.

**New Retimer Time Domain Flow**

For Retimer channels where all models specify parameter Use\_v7p1\_AMI\_Flow and each terminal Rx (including Retimer Rx) has AMI\_GetWave if the model has DFE, an alternative Retimer time domain simulation flow is defined below.

Step 1. The EDA tool obtains the impulse response of the upstream analog channel, which represents the combined impulse response of Tx1’s analog model, physical channel 1, and Rx1’s analog model.

Step 2. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Tx1. The output of step 1 and the value of Use\_v7p1\_AMI\_Flow are presented to Tx1’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Tx1’s AMI\_Init function is executed. Note that because Tx1 is a terminal Tx, the cumulative upstream impulse response in the input impulse\_matrix can be a unit impulse response, whose first value equals 1/sample\_interval and the others equal zero. The EDA tool may use the filter impulse response in the last column of the output impulse\_matrix to construct an AMI\_GetWave function for Tx1 if it does not have one.

Step 3. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Rx1. The output of step 2 and the value of Use\_v7p1\_AMI\_Flow are presented to Rx1’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Rx1’s AMI\_Init function is executed. The EDA tool may use the filter impulse response in the last column of the output impulse\_matrix to construct an AMI\_GetWave function for Rx1 if it does not have one.

Step 4. The EDA tool obtains the impulse response of the downstream analog channel, which represents the combined impulse response of Tx2’s analog model, physical channel 2, and Rx2’s analog model.

Step 5. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Tx2. The output of step 4 and the value of Use\_v7p1\_AMI\_Flow are presented to Tx2’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Tx2’s AMI\_Init function is executed. Note that because Tx2 is a terminal Tx, the cumulative upstream impulse response in the input impulse\_matrix can be a unit impulse response, whose first value equals 1/sample\_interval and the others equal zero. The EDA tool may use the filter impulse response in the last column of the output impulse\_matrix to construct an AMI\_GetWave function for Tx2 if it does not have one.

Step 6. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Rx2. The output of step 5 and the value of Use\_v7p1\_AMI\_Flow are presented to Rx2’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Rx2’s AMI\_Init function is executed. The EDA tool may use the filter impulse response in the last column of the output impulse\_matrix to construct an AMI\_GetWave function for Rx2 if it does not have one.

Step 7. The EDA tool performs simulation on the upstream channel, which consists of Tx1, physical channel 1, and Rx1, according to the AMI flow defined in the specification for channels without Repeaters. Note that deconvolution can be avoided by using the AMI\_GetWave function constructed by the EDA tool for Init-only Rx1 in step 3.

Step 8. The EDA tool samples the output waveform of Retimer Rx AMI\_GetWave at ½ UI after each clock tick returned by the function, generates a digital stimulus as the input to Tx2’s algorithmic model, and performs simulation on the downstream channel, which consists of Tx2, physical channel 2 and Rx2, according to the AMI flow defined in the specification for channels without Repeater. Note that deconvolution can be avoided by using the AMI\_GetWave function constructed by the EDA tool for Init-only Rx2 in step 6. The logic level of the digital stimulus is 1 if sampled value >= Rx1’s Rx\_Receiver\_Sensitivity and 0 if sampled value <= −Rx1’s Rx\_Receiver\_Sensitivity. If –Rx1’s Rx\_Receiver\_Sensitivity < sampled value < Rx1’s Rx\_Receiver\_Sensitivity, the logic level is unchanged from the previous bit. The digital stimulus shall have values of -½ volt for logic 0 and +½ volt for logic 1.

Step 9. The EDA tool calls the AMI\_Close function of each algorithmic model in Tx1, Rx1, Tx2 and Rx2.

Add the following paragraphs to Section 10.8.1 before the paragraph that starts with “IBIS does not prohibit the use of multiple Repeaters …”.

**New Redriver Statistical Flow**

For Redriver channels where all models specify parameter Use\_v7p1\_AMI\_Flow, an alternative Redriver statistical simulation flow is defined below.

Step 1. The EDA tool obtains the impulse response of the upstream analog channel, which represents the combined impulse response of Tx1’s analog model, physical channel 1, and Rx1’s analog model.

Step 2. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Tx1. The output of step 1 and the value of Use\_v7p1\_AMI\_Flow are presented to Tx1’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Tx1’s AMI\_Init function is executed. Note that because Tx1 is a terminal Tx, the cumulative upstream impulse response in the input impulse\_matrix can be a unit impulse response, whose first value equals 1/sample\_interval and the others equal zero.

Step 3. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Rx1. The output of step 2 and the value of Use\_v7p1\_AMI\_Flow are presented to Rx1’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Rx1’s AMI\_Init function is executed.

Step 4. The EDA tool obtains the impulse response of the downstream analog channel, which represents the combined impulse response of Tx2’s analog model, physical channel 2, and Rx2’s analog model.

Step 5. The EDA tool uses the output of step 3 to construct the cumulative upstream impulse response for Tx2.

Step 6. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Tx2. The output of steps 4 and 5 and the value of Use\_v7p1\_AMI\_Flow are presented to Tx2’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Tx2’s AMI\_Init function is executed.

Step 7. The EDA tool uses the output of steps 4 and 6 to construct the cumulative upstream impulse response for Rx2.

Step 8. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Rx2. The output of steps 6 and 7 and the value of Use\_v7p1\_AMI\_Flow are presented to Rx2’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Rx2’s AMI\_Init function is executed.

Step 9. The EDA tool uses the output of step 8 to perform statistical simulations.

Note that the input and output impulse\_matrix of each AMI\_Init call in this alternative Redriver channel statistical flow are identical respectively to those of the corresponding AMI\_Init call in the alternative Redriver channel time domain flow.

**New Retimer Statistical Flow**

For Retimer channels where all models specify parameter Use\_v7p1\_AMI\_Flow, an alternative Retimer statistical simulation flow is defined below.

Step 1. The EDA tool obtains the impulse response of the upstream analog channel, which represents the combined impulse response of Tx1’s analog model, physical channel 1, and Rx1’s analog model.

Step 2. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Tx1. The output of step 1 and the value of Use\_v7p1\_AMI\_Flow are presented to Tx1’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Tx1’s AMI\_Init function is executed. Note that because Tx1 is a terminal Tx, the cumulative upstream impulse response in the input impulse\_matrix can be a unit impulse response, whose first value equals 1/sample\_interval and the others equal zero.

Step 3. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Rx1. The output of step 2 and the value of Use\_v7p1\_AMI\_Flow are presented to Rx1’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Rx1’s AMI\_Init function is executed.

Step 4. The EDA tool obtains the impulse response of the downstream analog channel, which represents the combined impulse response of Tx2’s analog model, physical channel 2, and Rx2’s analog model.

Step 5. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Tx2. The output of step 4 and the value of Use\_v7p1\_AMI\_Flow are presented to Tx2’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Tx2’s AMI\_Init function is executed. Note that because Tx2 is a terminal Tx, the cumulative upstream impulse response in the input impulse\_matrix can be a unit impulse response, whose first value equals 1/sample\_interval and the others equal zero.

Step 6. The EDA tool sets Use\_v7p1\_AMI\_Flow to True for Rx2. The output of step 5 and the value of Use\_v7p1\_AMI\_Flow are presented to Rx2’s AMI\_Init function as specified in Section 10.2.3 for Use\_v7p1\_AMI\_Flow=True, and Rx2’s AMI\_Init function is executed.

Step 7. The EDA tool uses the impulse response returned by Rx1’s AMI\_Init in step 3 to perform a statistical simulation of channel 1. The EDA tool uses the impulse response returned by Rx2’s AMI\_Init in step 6 to perform a statistical simulation of channel 2.

Note that the input and output impulse\_matrix of each AMI\_Init call in this alternative Retimer channel statistical flow are identical respectively to those of the corresponding AMI\_Init call in the alternative Retimer channel time domain flow.

**BACKGROUND INFORMATION/HISTORY:**