**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 211.3

**ISSUE TITLE:** IBIS AMI Reference Flow Improvements

**REQUESTORS:**  Walter Katz, The MathWorks, Fangyi Rao, Keysight Technologies

**DATE SUBMITTED:** March 23, 2021

**DATE REVISED:** April 21, 2021; June 11, 2021; August 24, 2021

**DATE ACCEPTED:**

**DEFINITION OF THE ISSUE:**

The current Redriver statistical flow in IBIS 7.0 can be described graphically as follows:

The Physical Channel



The current Redriver statistical flow in IBIS 7.0



The current Redriver flow is known to have the 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. Except for the first Redriver Rx’s AMI\_Init function the cumulative upstream impulse response of the Redriver channel is not provided to either the Redriver Tx or the Redriver Rx in AMI\_Init. As a result, the AMI\_Init function cannot perform optimization on the upstream signal.
3. The combination of the Tx GetWave model and the Rx Init-only model leads to deconvolution in time domain simulations. This also occurs in the non-repeater time domain flow.

This BIRD proposes the following new Redriver statistical flow to replace the existing Redriver flow to address these issues. This flow shall apply to Redriver simulations when all the models have Init\_Returns\_Impulse set to True and is independent of the AMI\_Version of the model.



A new Reserved Parameter and new column in the impulse matrix are introduced to support additional flows. The new Reserved Parameter is (Tx\_Impulse\_Input (Value “Downstream” | “Combined” | “Separate” | “Upstream”) (Type String) (Usage Info)). This is an optional parameter for any Tx model. Tx\_Impulse\_Input shall determine the contents of the first column of the input Impulse Matrix of the Tx’s AMI\_Init function. If Tx\_Impulse\_Input is set to “Downstream” (default) this column shall contain the impulse response of the Tx’s Downstream channel. This is the same as the new Redriver flow described above.

The following shows the Redriver statistical flows for Tx2’s Tx\_Impulse\_Input being “Downstream”, “Combined”, “Separate” and “Upstream” respectively.



Note that when Tx\_Impulse\_Input is “Downstream” the output of Tx2 is convolved with the output of Rx1, which ensures that the input to Rx2 will contain its complete upstream impulse response.

One additional column is required for Tx models when Tx\_Impulse\_Input is “Separate”.

**SOLUTION REQUIREMENTS:**

The IBIS specification must meet these requirements:

Table 1: Solution Requirements

|  |  |
| --- | --- |
| Requirement | Notes |
| * Support statistical simulations on Redriver channels whose terminal Rx (including Retimer Rx) has DFE.
 |  |
| * Allow Redriver Tx AMI\_Init to perform optimization on the Downstream signal.
 |  |
| * Allow Redriver Tx AMI\_Init to perform optimization on the Upstream signal and the Downstream signal.
 |  |
| * By default, be compatible with existing Tx model usage.
 |  |
| * Redriver Flow without changes to Terminal Tx and Rx
 |  |
| * A Redriver Tx can be used as a Terminal Tx
 |  |

**SUMMARY OF PROPOSED CHANGES:**

Add new Reserved Parameter Tx\_Impulse\_Input.

Add one column at the end of impulse\_matrix in AMI\_Init when Tx\_Impulse\_Input set to “Separate”.

Modify flows to ensure that terminal Rx model always has the total upstream impulse response.

**PROPOSED CHANGES:**

**In Section 10.2.3, after:**

The crosstalk impulse responses may be placed into the impulse response

matrix in any order.

**Insert:**

If the Reserved Parameter Tx\_Impulse\_Input is “Separate” then a new column shall be added to the impulse\_matrix that shall contain the cumulative impulse response of all upstream models and channels of this Tx.

Note that EDA tools, for AMI models with AMI\_Version 7.2 and later, are allowed to determine the model filter impulse response by adding an aggressor column that contains a “unit impulse response” to determine the filter equalization. A unit impulse response contains all zeros except the first value, which shall equal 1.0/sample\_interval. Any model that uses the contents of the aggressor columns to optimize its equalization should ignore columns that contain a unit impulse response for the purpose of optimizing its equalization. However, the model should still apply equalization and gain to these columns.

Add the following to the end of Section 10.2.3:

The Reserved Parameter Tx\_Impulse\_Input determines the content of the impulse\_matrix input to the Tx AMI\_Init function and what the AMI\_Init function does to the output of the impulse\_matrix as described below.

1. The AMI\_Init function modifies the impulse response of the through channel in the impulse\_matrix in place by applying its gain and equalization to the first column of the impulse\_matrix.
2. The AMI\_Init function modifies the crosstalk channel columns of the impulse\_matrix in place by applying its gain and equalization to the aggressor columns.
3. The content of the input impulse\_matrix is determined by the value of parameter Tx\_Impulse\_Input as described in the parameter definition.

Note when parameter Tx\_Impulse\_Input is not present, or is “Downstream”, then the normal non-repeater flow is unchanged (except an aggressor unit impulse response may now be added to the impulse matrix).

Add the following new parameter in Section 10.4 before Use\_Init\_Output:

*Parameter:* **Tx\_Impulse\_Input**

*Required:* No, and illegal before AMI\_Version 7.2

*Direction:* Tx

*Descriptors:*

Usage: Info

Type:                     String

Format: Value

Default:                 <String\_literal>

Description:*<*string>

*Definition:* This parameter modifies the content of the impulse\_matrix input to AMI\_Init (10.2.3 FUNCTION SIGNATURES, AMI\_Init). Value must be one of the following: “Downstream”, “Combined”, “Separate”, or “Upstream”.

*Usage Rules:*

If “Downstream”:

Column 1 of the impulse\_matrix shall contain the impulse response of the model's direct Downstream channel.

If “Combined”:

Column 1 of the impulse\_matrix shall contain the cumulative impulse response of all upstream models and channels convolved with the Tx direct Downstream channel.

If “Separate”:

Column 1 shall contain the impulse response of the model's direct Downstream channel.

Column ‘aggressors + 2’ shall contain the cumulative impulse response of all upstream models and channels. The model shall not change the output of column ‘aggressors + 2’ (aggressors is the number of aggressors in the impulse\_matrix). For a terminal Tx or Retimer Tx, the upstream impulse response is a unit impulse response.

If “Upstream”:

Column 1 of the impulse\_matrix shall contain the cumulative impulse response of all preceding models and channels. For a terminal Tx or Retimer Tx, the upstream impulse response is a unit impulse response.

*Other Notes:* If Tx\_Impulse\_Input is not present the default shall be “Downstream”.

*Example:*

(Tx\_Impulse\_Input (Usage Info) (Type String) (Value “Downstream”)

(Description "The column 1 of the impulse\_matrix shall contain the

impulse response of the Tx downstream channel"))

In Section 10.2.2 add the following sentences at the end of STATISTICAL SIMULATION REFERENCE FLOW.

Note that in normal (non-repeater) statistical and time-domain simulations the content of the input impulse\_matrix to the Tx’s AMI\_Init is independent of the Tx’s Tx\_Impulse\_Input parameter value because passing different contents of the input impulse\_matrix to the Tx’s AMI\_Init based on the Tx’s Tx\_Impulse\_Input value always yields identical simulation results.

In Section 10.2.2 remove the following paragraphs.

Under certain circumstances, for example when the Rx AMI\_Init function includes an optimization algorithm, the impulse response presented to the Rx AMI\_Init function must include the Tx equalization effects for the optimization to work correctly. However, when the Tx AMI model contains an AMI\_GetWave function that performs a similar or better equalization than the Tx AMI\_Init function, there is a possibility for “double-counting” the equalization effects in the Tx executable model file. To allow for such models to work correctly, the EDA tool can operate in one of several ways, two of which are documented here:

* not utilize the Tx AMI\_GetWave functionality, by treating the Tx AMI model as if the Tx GetWave\_Exists was False.
* use deconvolution to obtain the impulse response of the Rx filter. Since the AMI\_Init function contains a linear and time invariant algorithm, the Rx equalization may be represented as an impulse response. Since the output of the Rx AMI\_Init function (output of step 3) is an impulse response modified by the Rx equalization (e.g., by convolving the input of the Rx AMI\_Init function with the impulse response of the Rx filter), the impulse response of the Rx filter may be obtained by deconvolving the output of step 3 with the input presented to step 3.

In Section 10.2.2 replace the following paragraph

Step 6d. If Tx GetWave\_Exists is True 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 EDA tool and 199 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).

with

Step 6d. If Tx GetWave\_Exists is True and Rx GetWave\_Exists is False, the EDA tool performs one of the two following operations and passes the result to Step 8.

* Not utilize the Tx AMI\_GetWave functionality (the output of Step 5 is discarded), by treating the Tx AMI model as if the Tx GetWave\_Exists was False. Convolve the output of Step 4 with the output of Step 3.
* Convolve the output of Step 5 with the output of Step 1 and a filter that represents the Rx’s gain and equalization, which may be determined using one of the following methods:
1. Deconvolving the output with the input impulse response of the Rx AMI\_Init function.

Note:

1. Since the Rx AMI\_Init function contains a linear and time invariant algorithm, the Rx gain and equalization may be represented by a filter.
2. Since the output of the Rx AMI\_Init function (output of Step 3) is an impulse response modified by the Rx gain and equalization (e.g., by convolving the input of the Rx AMI\_Init function with the impulse response of the Rx filter), the impulse response of the Rx filter may be obtained by deconvolving the output of Step 3 with the input presented to Step 3.
3. The EDA tool may add an aggressor column that is initialized to a “unit impulse response”.

Note:

* 1. If the EDA tool does add an aggressor column that is initialized to a unit impulse response, the tool shall also correspondingly increase the value of the aggressor argument of AMI\_Init by one.
	2. A model that uses the crosstalk columns of the impulse\_matrix to optimize its equalization shall ignore any column that contains a “unit impulse response” for the purpose of optimizing its equalization.
	3. EDA tools should be aware that a pre-AMI Version 7.2 Rx model may optimize its equalization based on the contents of the aggressor columns of the impulse\_matrix and does not ignore unit impulse response columns, rendering method 2 inapplicable.

#### Replace the Repeater Reference Flows Section 10.8.1 With

The physical layout of the Repeater (Retimer or Redriver) link discussed below is illustrated in Figure 41.

Figure – Repeater Link

Repeater

Rx

Tx1

Rx1

Tx2

Rx2

channel 1

channel 2

Repeater

Repeater Tx

Incoming

(upstream)

channel

outgoing

(downstream)

channel

Terminal

Tx

Terminal

Rx

Here Tx1 denotes the Repeater upstream channel (channel 1) Tx AMI model (including analog and algorithmic models), Rx1 denotes the Repeater Rx AMI model (including analog and algorithmic models), Tx2 denotes the Repeater Tx AMI model (including analog and algorithmic models), and Rx2 denotes the Repeater Downstream channel (channel 2) Rx AMI model (including analog and algorithmic models).

**Retimer Statistical Simulation Flow**

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 output of step 1 is presented to the Tx1’s AMI\_Init function, and Tx1’s AMI\_Init function is executed.

Step 3. The output of step 2 is presented to the Rx1’s AMI\_Init function, and the 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 output of step 4 is presented to Tx2’s AMI\_Init function, and Tx2’s AMI\_Init function is executed.

Step 6. The output of step 5 is presented to Rx2’s AMI\_Init function, and Rx2’s AMI\_Init function is executed.

Step 7. The EDA tool completes the rest of the statistical simulation and analysis using the impulse response returned in step 3 by the Rx1’s AMI\_Init function, which is a complete representation of the behavior of Tx1 and Rx1 algorithmic models combined with the upstream channel 1, and the impulse response returned in step 6 by the Rx2’s AMI\_Init function, which is a complete representation of the behavior of Tx2 and Rx2 algorithmic models combined with the downstream channel 2.

**Retimer Time Domain Simulation Flow**

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 output of step 1 is presented to Tx1’s AMI\_Init function, and Tx1’s AMI\_Init function is executed.

Step 3. The output of step 2 is presented to Rx1’s AMI\_Init function, 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 output of step 4 is presented to Tx2’s AMI\_Init function, and Tx2’s AMI\_Init function is executed.

Step 6. The output of step 5 is presented to Rx2’s AMI\_Init function, and Rx2’s AMI\_Init function is executed.

Step 7. The EDA tool performs the time domain 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.

Step 8. The EDA tool samples the output waveform of Retimer Rx1 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, regardless of whether Tx2’s AMI\_GetWave exists or not, and performs the 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. 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.

Steps 7 through 8 can be called once or can be called multiple times to process the full analog waveform. Splitting up the full analog waveform into multiple calls reduces the memory requirements when doing long simulations and allows AMI\_GetWave to return model status every so many bits. Once all blocks of the input waveform have been processed, the EDA tool calls the AMI\_Close function of each algorithmic model in Tx1, Rx1, Tx2, and Rx2.

Since the Retimer output signal is driven by a digital stimulus as described above in step 8, jitter and noise parameters specified in Retimer .ami files are applied according to the specification for channels without Repeaters.

**Redriver Flows**

Both statistical and time domain Redriver simulations require that AMI\_Init functions of Tx1, Rx1, Tx2, and Rx2 are executed first. The following figure shows flows of executing AMI\_Init functions in Redriver statistical and time domain simulations when the Tx2 Tx\_Impulse\_Input is set to “Downstream”, “Combined”, “Separate”, and Upstream, respectively. By setting Tx\_Impulse\_Input to “Upstream”, the Tx2 model maker is declaring that the Tx2 initialization (AMI\_Init) function does not have the ability to adapt itself based on the downstream channel.



If Tx\_Impulse\_Input is not present, the AMI\_Init functions shall be executed in the same manner as when Tx\_Impulse\_Input is set to Downstream.  This rule shall be applied to all AMI models.

After completing all steps of executing Tx1, Rx1, Tx2 and Rx2 AMI\_Init functions, the EDA tool may use results generated in these steps to perform the rest of the statistical or time domain simulation as described below.

**Redriver Statistical Simulation Flow**

To perform statistical simulations, all models, including the terminal Tx, Redriver Rx, Redriver Tx, and terminal Rx shall set Init\_Returns\_Impulse to True.

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

Step 2. The output of step 1 is presented to Tx1’s AMI\_Init function, and Tx1’s AMI\_Init function is executed.

Step 3. The output of step 2 is presented to Rx1’s AMI\_Init function, and Rx1’s AMI\_Init function is executed.

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

Step 5a. If Tx2’s Tx\_Impulse\_Input is not present or is “Downstream” then column 1 of impulse\_matrix shall contain the output of step 4. This impulse\_matrix is presented to Tx2’s AMI\_Init function, and Tx2’s AMI\_Init function is executed.

Step 5b. If Tx2’s Tx\_Impulse\_Input is “Combined” then column 1 of impulse\_matrix shall contain the output of step 3 convolved with the output of step 4. This impulse\_matrix is presented to Tx2’s AMI\_Init function, and Tx2’s AMI\_Init function is executed.

Step 5c. If Tx2’s Tx\_Impulse\_Input is “Separate” then column 1 of impulse\_matrix shall contain the output of step 4 and column “aggressors+2” shall contain the output of step 3. This impulse\_matrix is presented to Tx2’s AMI\_Init function, and Tx2’s AMI\_Init function is executed.

Step 5d. If Tx2’s Tx\_Impulse\_Input is “Upstream” then column 1 of impulse\_matrix shall contain the output of step 3. This impulse\_matrix is presented to Tx2’s AMI\_Init function, and Tx2’s AMI\_Init function is executed.

Step 6a. If Tx2’s Tx\_Impulse\_Input is not present or is “Downstream” then the output of column 1 of step 5 is convolved with the output of step 3, the result is presented to Rx2’s AMI\_Init function, and Rx2’s AMI\_Init function is executed.

Step 6b. If Tx2’s Tx\_Impulse\_Input is “Combined” then the output of column 1 of step 5 is presented to Rx2’s AMI\_Init function, and Rx2’s AMI\_Init function is executed.

Step 6c. If Tx2 Tx\_Impulse\_Input is “Separate” then the output of column 1 of step 5 is convolved with the output of step 3, the result is presented to Rx2’s AMI\_Init function, and Rx2’s AMI\_Init function is executed.

Step 6d. If Tx2 Tx\_Impulse\_Input is “Upstream” then the output of column 1 of step 5 is convolved with the output of step 4, the result is presented to Rx2’s AMI\_Init function, and Rx2 AMI\_Init function is executed.

Step 7. The EDA tool completes the rest of the simulation and analysis using the impulse response returned in step 6 by the Rx2’s AMI\_Init function.

**Redriver Time Domain Simulation Flow**

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

Step 2. The output of step 1 is presented to Tx1’s AMI\_Init function, and Tx1’s AMI\_Init function is executed.

Step 3. The output of step 2 is presented to Rx1’s AMI\_Init function, and Rx1’s AMI\_Init function is executed.

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

Step 5a. If Tx2’s Tx\_Impulse\_Input is not present or is “Downstream” then column 1 of impulse\_matrix shall contain the output of step 4. This impulse\_matrix is presented to Tx2’s AMI\_Init function, and Tx2’s AMI\_Init function is executed.

Step 5b. If Tx2’s Tx\_Impulse\_Input is “Combined” then column 1 of impulse\_matrix shall contain the output of step 3 convolved with the output of step 4. This impulse\_matrix is presented to Tx2’s AMI\_Init function, and Tx2’s AMI\_Init function is executed.

Step 5c. If Tx2’s Tx\_Impulse\_Input is “Separate” then column 1 of impulse\_matrix shall contain the output of step 4 and column “aggressors+2” shall contain the output of step 3. This impulse\_matrix is presented to Tx2’s AMI\_Init function, and Tx2’s AMI\_Init function is executed.

Step 5d. If Tx2’s Tx\_Impulse\_Input is “Upstream” then column 1 of impulse\_matrix shall contain the output of step 3. This impulse\_matrix is presented to Tx2’s AMI\_Init function, and Tx2’s AMI\_Init function is executed.

Step 6a. If Tx2’s Tx\_Impulse\_Input is not present or is “Downstream” then the output of column 1 of step 5 is convolved with the output of step 3, the result is presented to Rx2’s AMI\_Init function, and Rx2’s AMI\_Init function is executed.

Step 6b. If Tx2’s Tx\_Impulse\_Input is “Combined” then the output of column 1 of step 5 is presented to Rx2’s AMI\_Init function, and Rx2’s AMI\_Init function is executed.

Step 6c. If Tx2 Tx\_Impulse\_Input is “Separate” then the output of column 1 of step 5 is convolved with the output of step 3, the result is presented to Rx2’s AMI\_Init function, and Rx2’s AMI\_Init function is executed.

Step 6d. If Tx2 Tx\_Impulse\_Input is “Upstream” then the output of column 1 of step 5 is convolved with the output of step 4, the result is presented to Rx2’s AMI\_Init function, and Rx2’s AMI\_Init function is executed.

Step 7. The EDA tool performs the 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.

Step 8. The EDA tool uses the signal waveform at the output of Rx1’s algorithmic model in step 7 as the stimulus of Tx2’s algorithmic model and performs the 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 Repeaters.

Steps 7 through 8 can be called once or can be called multiple times to process the full analog waveform. Splitting up the full analog waveform into multiple calls reduces the memory requirements when doing long simulations and allows AMI\_GetWave to return model status every so many bits. Once all blocks of the input waveform have been processed, the EDA tool calls the AMI\_Close function of each algorithmic model in Tx1, Rx1, Tx2 and Rx2.

Since the Redriver output signal is driven continuously by the input analog signal and does not have a sampling latch, clock times, if returned by the Rx1 AMI\_GetWave function, jitter parameters, and the Rx\_Noise parameter specified in Redriver .ami files are ignored by the EDA tool.

**BACKGROUND INFORMATION/HISTORY:**

BIRD211.1 includes the following changes:

First change is to remove Reserved Parameter **Init\_Returns\_Equalization**. We agreed that this was not necessary because the EDA tool may always add an aggressor column to the impulse matrix that is initialized to a unit impulse response, and that the output of this column will contains the impulse response of the filter’s equalization.

The second change was to replace **Tx\_Requires\_Downstream\_Channel** with another ReservedParameter **Tx\_Impulse\_Input.** This change allows flexibility to define three flows:

1. “Downstream”
	1. This is the default and is compatible with the existing IBIS 7.0 flow with the exception that the output of the Redriver Rx is included in the impulse response input to the terminal Rx.
2. “Combined”
	1. This flow combines the output of the Redriver Rx with the Redriver Tx Downstream channel as the input to the Redriver Tx.
3. “Separate”
	1. In this flow the EDA tool presents two impulse responses , one of the accumulated upstream channel of the Redriver Tx and the other of the Redriver Tx Downstream channel, to the Redriver Tx.
4. “Upstream”.

In this flow the EDA tool presents the accumulated upstream channel of the Redriver Tx to the Redriver Tx.

BIRD211.2 includes the following changes:

1. When Redriver Tx2 **Tx\_Impulse\_Input** set to “Separate” the impulse matrix column 1 input shall be the direct Downstream channel and column “aggressor +2” shall be the accumulated upstream channel (output of Rx1). The EDA tool shall convolve the column 1 output of the Tx2 impulse matrix with the output of Rx1.
2. **Tx\_Impulse\_Input** ”IBIS7.0”is changed to “Downstream”, with no change of usage
3. **Tx\_Impulse\_Input** ”DoNotCare” is changed to “Upstream”, the impulse matrix column 1 input shall be the cumulative Impulse Response of all upstream channels. This will normally be a Unit Impulse Response if the Tx is a terminal Tx. If the Tx is a Redriver Tx, the impulse matrix column 1 input shall be the column 1 impulse response of output of the Redriver’s Rx.

BIRD211.3 includes the following changes:

1. For normal flows, revert to IBIS 7.0 flows.
2. In the normal time-domain flow, reorganize sections about deconvolution and the handling of the case where Tx has GetWave and Rx does not. Add the section about the method of determining the Rx filter using a unit impulse response aggressor column.
3. For Retimer flows, revert to IBIS 7.0 flows.
4. For channel 1 (upstream) Tx1 and Rx 1 Init steps in Redriver flows, revert to IBIS 7.0 flows.