**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 211.2

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

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

**DATE SUBMITTED:** March 23, 2021

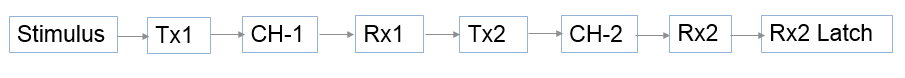
**DATE REVISED:** April 21, 2021; June 11, 2021

**DATE ACCEPTED:**

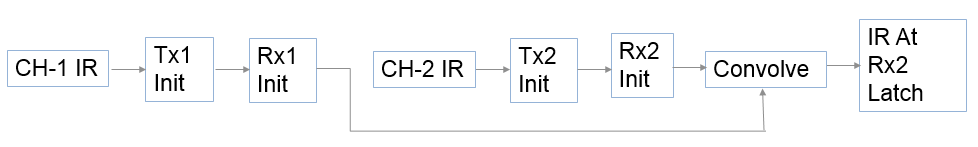
**DEFINITION OF THE ISSUE:**

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

The Physical Channel



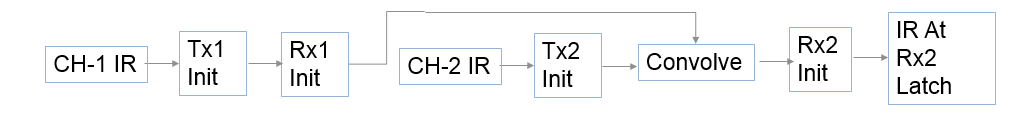
The current Redriver statistical flow in IBIS 7.0



The current Redriver statistical flow is known to have the following issues:

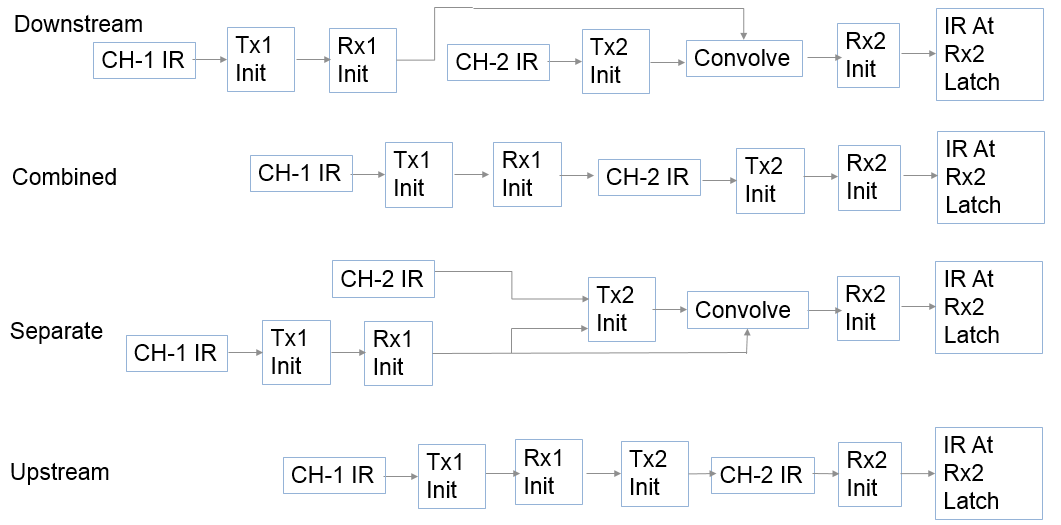
1. The cumulative upstream impulse response of the Redriver channel is not provided to the terminal Rx (including Retimer Rx) in AMI\_Init. As a result, when the terminal Rx has DFE, the end-to-end cumulative impulse response of the Redriver channel needed in statistical simulations is not available.
2. The cumulative upstream impulse response of the Redriver channel is not provided to either redriver Tx or redriver 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 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 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 total upstream impulse response.

**PROPOSED CHANGES:**

**On page 201 in Section 10.2.3, after:**

The crosstalk impulse responses may be placed into the impulse response

matrix in any order.

**Insert:**

If 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 7.1 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. Models that use the contents of the aggressor columns to determine the model’s equalization should ignore, when determining model equalization, columns that contain a unit impulse response. However, the model should still apply equalization and gain to these columns.

Add the following to 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”.

1. AMI\_Init function modifies the through channel (column 1) of impulse\_matrix in place by applying its gain and equalization to the first column of the impulse\_matrix
2. AMI\_Init function modifies the crosstalk channel columns of impulse\_matrix in place by applying its gain and equalization to the aggressor columns
3. Column 1 of the impulse response input is determined by the value of Tx\_Impulse\_Input. If Tx\_Impulse\_Input is “Separate”, an additional column that contains a second through channel impulse response is added to the impulse matrix by the EDA tool. The model does not modify this additional column.

Note when Tx\_Impulse\_Input parameter 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 on page 226:

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

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

*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. Model shall not change the output of column ‘aggressors + 2’ (aggressors is the number of aggressors in the impulse\_matrix).

If “Upstream”:

Column 1 of the impulse\_matrix shall contain the cumulative impulse response of all preceding models and channels.

*Other Notes:* If Tx\_Impulse\_Input is not present its value 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"))

#### Replace the Reference Flows In Section 10.2.2 With

#### Reference Flows

The next section defines a reference simulation flow for statistical and time domain system analysis simulations. Other methods of calling models and processing results may be employed, but the final simulation waveforms are expected to match the waveforms produced by this reference simulation flow.

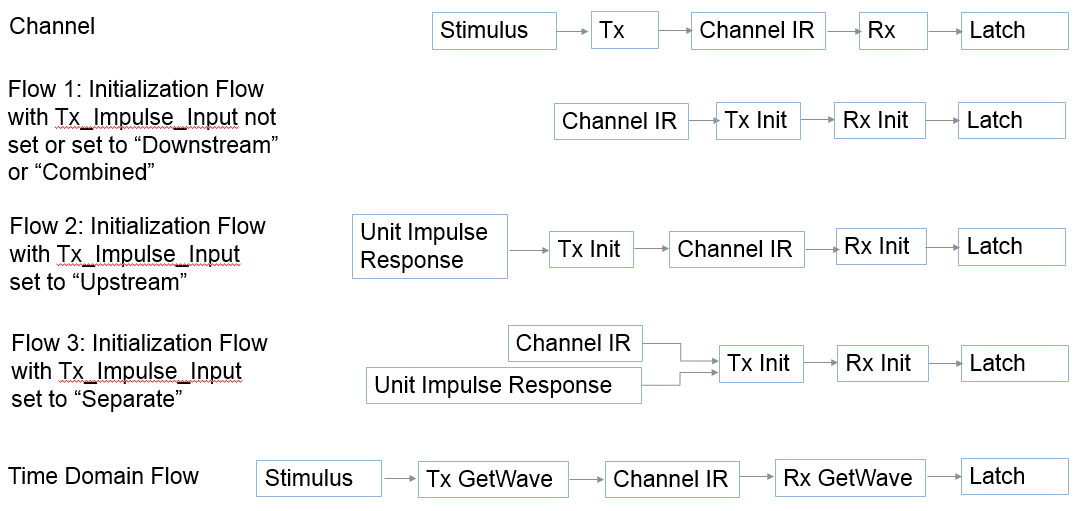
A system simulation usually involves a transmitter (Tx) and a receiver (Rx) model with a passive channel placed between them. In the following figure, the passive channel is represented by the “Channel IR” blocks. There is an initialization flow (calls to AMI\_Init) and a time domain flow (calls to AMI\_GetWave). The EDA tool may perform just the initialization flow and use the results of the initialization flow to do statistical analysis. The EDA tool may perform both an initialization flow and a time domain flow. When doing the latter, the EDA tool may analyze the results of the time domain flow and optionally analyze the results of the initialization flow to do statistical analysis. The EDA tool cannot do a time domain flow without doing an initialization flow first. Note that doing statistical analysis following the initialization flow requires that all AMI\_Init functions do have Init\_Returns\_Impulse set to True. An Initialization Flow can be followed by a statistical analysis. This is called a statistical flow. Time domain simulations require that an initialization flow be performed first. The following figure shows three normal (non-repeater) initialization flows.

Flow 1: Initialization Flow with Tx\_Impulse\_Input is not present or set to “Downstream” or “Combined”

Flow 2: Initialization Flow with Tx\_Impulse\_Input set to “Upstream”

Flow 3: Initialization Flow with Tx\_Impulse\_Input set to “Separate”

By setting Tx\_Impulse\_Input set to “Upstream”, the model maker is declaring that the Tx initialization (AMI\_Init) function does not have the ability to adapt itself based on the downstream channel.



##### Normal (Non-Repeater) Simulation Reference Flow

Step 1. The EDA tool obtains the impulse response (“Channel IR”) of the analog channel. This represents the combined impulse response of the transmitter’s analog output, the channel and the receiver’s analog front end. The transmitter’s output or receiver’s input characteristics must not include any filtering effects, for example equalization, in this impulse response, although it may include any parasitics which are included in the Tx or Rx analog model.

Step 2ab. If Tx\_Impulse\_Input is not present or is “Downstream” or “Combined” then column 1 of impulse\_matrix shall contain the output of step 1 and Tx’s AMI\_Init function is executed.

Step 2c. If Tx\_Impulse\_Input is “Separate” then column 1 of impulse\_matrix shall contain the output of step 1 and column “aggressors+2” shall contain a unit impulse response and Tx’s AMI\_Init function is executed.

Step 2d. If Tx\_Impulse\_Input is “Upstream” then column 1 of impulse\_matrix shall contain a unit impulse response and Tx’s AMI\_Init function is executed.

Step 3abc. If Tx\_Impulse\_Input is not present or is “Downstream”, “Combined” or “Separate” then the output of column 1 of step 2 is presented to the Rx executable model’s AMI\_Init function and the Rx AMI\_Init function is executed.

Step 3d. If Tx\_Impulse\_Input is “Upstream” then the EDA tool shall convolve the output of step 1 with the output of step 2d and present the result to the Rx executable model’s AMI\_Init function and the Rx AMI\_Init function is executed.

Step 4. (This step is optional if the EDA tool proceeds with the following time domain simulation.) The EDA tool completes the rest of the simulation/analysis using the impulse response returned in step 3 by the Rx executable model’s AMI\_Init function which is a complete representation of the behavior of Tx and Rx algorithmic models combined with the channel. If only doing a statistical simulation, the flow is terminated after step 4.

The time domain reference flow assumes that if GetWave\_Exists is False, the EDA tool may emulate the AMI\_GetWave function by convolving the “GetWave Input” with a filter that represents the model’s equalization, which may be determined using one of the following methods:

1. Deconvolving the output with the input impulse response of the AMI\_Init function.
2. EDA tools may add an aggressor column that is initialized to a “unit impulse response”
   1. A “unit impulse response” contains all zeros and except the first value shall equal 1.0/sample\_interval
   2. Models that use the crosstalk columns of the impulse\_matrix to determine the equalization should ignore when determining the model’s equalization any column that contains a “unit impulse response”
   3. Note that pre–AMI Version 7.1 Rx models may optimize their equalization based on the contents of the aggressor columns of the impulse\_matrix. EDA tools should use deconvolution if the model is Init Only to emulate a GetWave functionality.

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. To allow for such models to work correctly, the EDA tool may 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.

Note: The Rx executable model writer should keep in mind that it is not guaranteed that the impulse response that is presented to the Rx AMI\_Init function will always include the effects of the Tx filter. Therefore, the Rx AMI\_Init function may not be able to perform accurate optimization under all circumstances. For this reason, the parameters of the Rx AMI\_Init function should always default to valid values or have a mechanism to accept user-defined coefficients and allow the user to turn off any automatic optimization routines to ensure successful simulations.

Step 5. The EDA tool produces a digital stimulus waveform. A digital stimulus waveform is 0.5 when the stimulus is "high", -0.5 when the stimulus is "low", and may have a value between -0.5 and 0.5 such that transitions occur when the stimulus crosses 0.

Step 6a. If Tx GetWave\_Exists is True, the output of step 5 is presented to the Tx executable 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 7.

Step 6b. If Tx GetWave\_Exists is False, the output of step 5 is convolved with the Tx filter impulse response and the result is passed on to step 7.

Step 7. The output of step 6 is convolved with the output of step 1 by the EDA tool and the result is passed on to step 8.

Step 8a. If Rx GetWave\_Exists is True, the output of step 7 is presented to the Rx executable 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 9.

Step 8b. If Rx GetWave\_Exists is False, the output of step 7 is convolved with the Rx filter impulse response and is passed on to step 9.

Step 9. The output of step 8 becomes the simulation waveform output at the Rx decision point. Step 8a optionally may also return clock ticks, which may be post-processed by the EDA tool or presented to the user as-is.

Steps 5 through 9 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, Tx AMI\_Close and Rx AMI\_Close are called to perform any final processing and release allocated memory.

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

The time domain simulation flow for a Repeater link shown in Figure 41 is defined below.

Figure – Repeater Link

Repeater

Rx

Tx1

Rx1

Tx2

Rx2

channel 1

channel 2

Repeater

Repeater Tx

Incoming

(upstream)

channel

outgoing

(downstream)

channel

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

**Retimer 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 2ab. If Tx1’s Tx\_Impulse\_Input is not present or is “Downstream” or “Combined” then column 1 of impulse\_matrix shall contain the output of step 1 and Tx1’s AMI\_Init function is executed.

Step 2c. If Tx1’s Tx\_Impulse\_Input is “Separate” then column 1 of impulse\_matrix shall contain the output of step 1 and column “aggressors+2” shall contain a unit impulse response and Tx1’s AMI\_Init function is executed.

Step 2d. If Tx1’s Tx\_Impulse\_Input is “Upstream” then column 1 of impulse\_matrix shall contain a unit impulse response and Tx1’s AMI\_Init function is executed.

Step 3.abc If Tx1’s Tx\_Impulse\_Input is not present or is “Downstream”, “Combined” or “Separate” then the output of column 1 of step 2 is presented to the Rx1 executable model’s AMI\_Init function and the Rx1’s AMI\_Init function is executed.

Step 3.d If Tx1’s Tx\_Impulse\_Input is “Upstream” then the EDA tool shall convolve the output of step 1 with the output of step 2d and present the result to the Rx1’s executable model’s AMI\_Init function and the 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 5ab. If Tx2’s Tx\_Impulse\_Input is not present or is “Downstream” or “Combined” then column 1 of impulse\_matrix shall contain the output of step 4 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 a unit impulse response and Tx2’s Tx’s AMI\_Init function is expected.

Step 5d. If Tx2’s Tx\_Impulse\_Input is “Upstream” then column 1 of impulse\_matrix shall contain a unit impulse and Tx2’s AMI\_Init function is executed.

Step 6abc. If Tx2’s Tx\_Impulse\_Input is not present or is “Downstream”, “Combined” or “Separate” then the output of column 1 of step 5 is presented to the Rx2’s executable model’s AMI\_Init function and the Rx2’s AMI\_Init function is executed.

Step 6d. If Tx\_Impulse\_Input is “Upstream” then the EDA tool shall convolve the output of step 4 with the output of step 5d and present the result to the Rx2’s executable model’s AMI\_Init function and the Rx2’s AMI\_Init function is executed.

Step 7. The EDA tool completes the rest of the statistical simulation/analysis using the impulse response returned in step 3 by the Rx1’s executable model’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 executable model’s AMI\_Init function which is a complete representation of the behavior of Tx2 and Rx2 algorithmic models combined with the downstream channel 2. This step is optional if the EDA tool proceeds with the following time domain simulation. If not doing a time domain simulation, the flow is terminated after step 7.

Step 8. The EDA tool performs 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 9. 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 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 8 through 9 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 9, jitter and noise parameters specified in Retimer .ami files are applied according to the specification for channels without Repeaters.

**Redriver Flow**

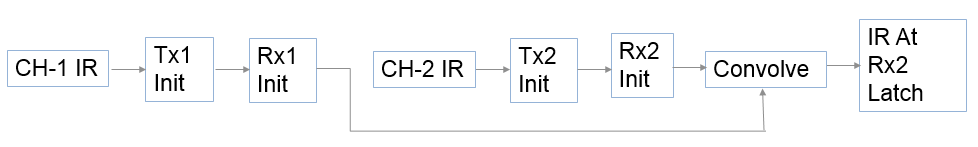
To perform statistical simulations, all models, including the Primary Tx, Redriver Rx, Redriver Tx, and Terminal Rx shall set Init\_Returns\_Impulse to True. Note that if a model’s AMI\_Version is pre 7.1 then it shall follow the flow below which is different than the Redriver flow in IBIS 7.0 Redriver flow.

If Tx\_Impulse\_Input is not present, the Redriver flow shall be the same as the flow when Tx\_Impulse\_Input is Downstream.

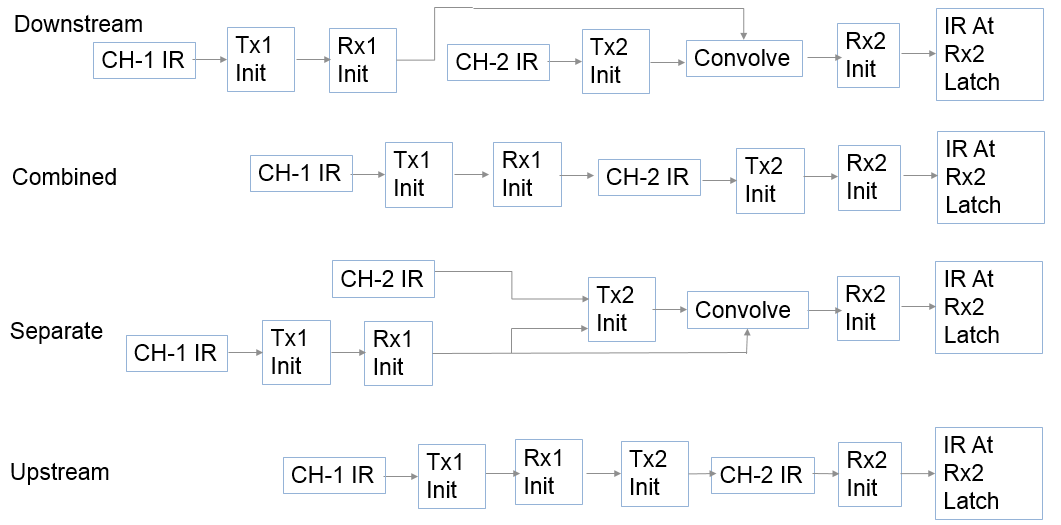
Note that this change effects only the statical flow when all the AMI\_Init model have Init\_Returns\_Impulse set to True.

The flow for models prior to AMI\_Version 7.1 is replaced with the same flow as AMI\_Version 7.1 models with Tx\_Impulse\_Input set to Downstream. In this changed flow the output of step 3 below is convolved with the output of step 5 which is then presented to the Terminal Rx2.

Original Redriver initialization flow replaced by Downstream flow below:



The following figures are the initialization flows when the Tx2 Tx\_Impulse\_Init is set to “Downstream”, “Combined”, “Separate” and Upstream. In all of these cases the Tx1 Tx\_Impulse\_Init is set to “Downstream”.



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 2ab. If Tx1’s Tx\_Impulse\_Input is not present or is “Downstream” or is “Combined” then column 1 of impulse\_matrix shall contain the output of step 1 and Tx1’s AMI\_Init function is executed.

Step 2c. If Tx1’s Tx\_Impulse\_Input is “Separate” then column 1 of impulse\_matrix shall contain the output of step 1 and column “aggressors+2” shall contain a unit impulse response and Tx1’s AMI\_Init function is executed.

Step 2d. If Tx1’s Tx\_Impulse\_Input is “Upstream” then column 1 of impulse\_matrix shall contain a unit impulse response and Tx1’s AMI\_Init function is executed.

Step 3abc. If Tx1’s Tx\_Impulse\_Input is not present or is “Downstream”, “Combined” or “Separate” then the output of column 1 of step 2 is presented to the Rx1’s executable model’s AMI\_Init function and the Rx1’s AMI\_Init function is executed.

Step 3d. If Tx1’s Tx\_Impulse\_Input is “Upstream” then the EDA tool shall convolve the output of step 1 with the output of step 2d and present the result to the Rx1’s executable model’s AMI\_Init function and the 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 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 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 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 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 and the result is presented to the Rx2’s executable model’s AMI\_Init function and the 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 the Rx2’s executable model’s AMI\_Init function and the 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 and the result is presented to the Rx2’s executable model’s AMI\_Init function and the 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 and the Rx2 AMI\_Init function is executed.

Step 7. (This step is optional if the EDA tool proceeds with the following time domain simulation.) The EDA tool completes the rest of the simulation/analysis using the impulse response returned in step 6 by the Rx2’s executable model’s AMI\_Init function. If only doing a statistical simulation, the flow is terminated after step 7.

Step 8. 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.

Step 9. The EDA tool uses the signal waveform at the output end of Rx1’s algorithmic model in step 8 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 Repeaters.

Steps 8 through 9 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 a Redriver model, 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.