**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD NUMBER:** 211.1

**ISSUE TITLE:** New Redriver Flow

**REQUESTOR:**  Walter Katz, The MathWorks, Inc

**DATE SUBMITTED:** March 23, 2021

**DATE REVISED:** April 21, 2021

**DATE ACCEPTED:**

**DEFINITION OF THE ISSUE:**

The current Repeater flow in the IBIS specification can be described graphically as follows:

The Channel



The current flow in IBIS 7.0



The current Repeater 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 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 column in the impulse matrix are introduced to support the proposed flow.

The new Reserved Parameter is (Tx\_Impulse\_Input (Value “IBIS7.0” | “Combined” | “Separate” | “DoNotCare”) (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 Impulse Matrix. If Tx\_Impulse\_Input=“IBIS7.0” (default) this column shall contain the impulse response of the Tx’s downstream channel. This is consistent with the current flows in IBIS 7.0.

The following shows the Redriver Flows for “IBIS7.0”, “Combined” and “Separate” respectively.



Note that when Tx\_Impulse\_Input=“IBIS7.0” 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 =“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.
 |  |
| * By default, be compatible with existing Tx model usage.
 |  |

**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=“Separate”.

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

**PROPOSED CHANGES:**

**On page 201, after**

The crosstalk impulse responses may be placed into the impulse response

matrix in any order.

Insert

If Tx\_Impulse\_Input=“Separate” then a new column shall be added to the impulse\_matrix that shall contain the impulse response of the channel downstream of this Tx.

Add the following paragraph to Section 10.2.3.

IBIS Version 7.1 introduces modification to the normal flow and the Redriver simulation flow. The new Reserved Parameter Tx\_Impulse\_Input determines the content of the impulse\_matrix input to the Tx AMI\_Init function and what the EDA tool does with the output of the impulse\_matrix.

1. Modify the through channel column of impulse\_matrix in place by applying its gain and equalization to the first column of the impulse\_matrix
2. Modify the crosstalk channel columns of impulse\_matrix in place by applying its gain and equalization to the aggressor columns
3. The model shall not modify the additional column that contains the downstream channel.

Add the following new parameter:

*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:* Value must be one of the following: “IBIS7.0”, “Combined”, “Separate”, or “DoNotCare”. If Tx\_Impulse\_Input is not present it shall be assumed to be “IBIS7.0”.

*Usage Rules:*

If “IBIS7.0”:

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

If “Combined”:

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

If “Separate”:

Column 1 shall contain the impulse response of the model's 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’ ”

If “DoNotCare”:

Column 1 of the impulse\_matrix can contain any combination of:

Unit impulse response

Impulse response of the model's downstream channel

Cumulative impulse response of all upstream models and channels.

*Other Notes:*

*Example:*

(Tx\_Impulse\_Input (Usage Info) (Type String) (Value “IBIS7.0”)

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

 impulse response of the Tx downstream channel"))

#### Replace the Reference Flows Section 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.

##### Simulation Reference Flow

Step 1. The EDA tool obtains the impulse response 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 2a. If Tx\_Impulse\_Input is not present or is “IBIS7.0” then column 1 of impulse\_matrix shall contain the output of step 1 and Tx’s AMI\_Init function is executed.

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

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

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

Step 3.a If Tx\_Impulse\_Input is not present or is “IBIS7.0” then the output of column 1 of step 2 is presented to the Rx executable model file’s AMI\_Init function and the Rx AMI\_Init function is executed.

Step 3.b If Tx\_Impulse\_Input is “Combined” then the output of column 1 of step 2 is convolved with the output of step 1 and is presented to the Rx executable model file’s AMI\_Init function and the Rx AMI\_Init function is executed.

Step 3.c If Tx\_Impulse\_Input is “DoNotCare” then the output of column 1 of step 2 is convolved with the output of step 1 if the input to column 1 was a unit impulse response and is presented to the Rx executable model file’s AMI\_Init function and the Rx AMI\_Init function is executed.

Step 3.d If Tx\_Impulse\_Input is “Separate” then the output of column 1 of step 2 is presented to the Rx executable model file’s AMI\_Init function and the Rx AMI\_Init function is executed.

Step 4. The EDA tool completes the rest of the simulation/analysis using the impulse response calculated in Step 3 by the Rx executable model file’s AMI\_Init function which is a complete representation of the behavior of a given [Algorithmic Model] combined with the channel. This step is optional if the EDA tool proceeds with the following time domain simulation.

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

1. Deconvolving the output with the input impulse response of the AMI\_Init function.
2. EDA tools add an aggressor column that is initialized to a “unit impulse response”
	1. A “unit impulse response” contains all zeros and except one row shall contain 1.0/sample\_interval
	2. Models that use the crosstalk columns of the impulse\_matrix should ignore any column that contains a “unit impulse response”

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 can 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 can be obtained by deconvolving the output of Step 3 with the input presented to Step 3.

Note: The Rx executable model file 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 file’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 4 is convolved with the Tx filter impulse response and 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 file’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 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 simulation 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.

If just doing a statistical simulation, the flow is terminated after step 4. If doing a time domain simulation, step 4 may be skipped.

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 2a. If Tx1 Tx\_Impulse\_Input is not present or is “IBIS7.0” then column 1 of impulse\_matrix shall contain the output of step 1 and Tx1’s AMI\_Init function is executed.

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

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

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

Step 3.a If Tx1 Tx\_Impulse\_Input is not present or is “IBIS7.0” then the output of column 1 of step 2 is presented to the Rx1 executable model file’s AMI\_Init function and the Rx1 AMI\_Init function is executed.

Step 3.b If Tx1 Tx\_Impulse\_Input is “Combined” then the output of column 1 of step 2 is convolved with the output of step 1 and is presented to the Rx1 executable model file’s AMI\_Init function and the Rx1 AMI\_Init function is executed.

Step 3.c If Tx1 Tx\_Impulse\_Input is “DoNotCare” then the output of column 1 of step 2 is convolved with the output of step 1 if the input to column 1 was a unit impulse response and is presented to the Rx1 executable model file’s AMI\_Init function and the Rx1 AMI\_Init function is executed.

Step 3.d If Tx1 Tx\_Impulse\_Input is “Separate” then the output of column 1 of step 2 is presented to the Rx1 executable model file’s AMI\_Init function and the Rx1 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 Tx\_Impulse\_Input is not present or is “IBIS7.0” 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 Tx\_Impulse\_Input is “Combined” then column 1 of impulse\_matrix shall contain a unit impulse response and Tx2’s AMI\_Init function is executed.

Step 5c. If Tx2 Tx\_Impulse\_Input is “DoNotCare” then column 1 of impulse\_matrix shall contain either a unit impulse response or the output of step 4 and Tx2’s AMI\_Init function is executed.

Step 5d. If Tx2 Tx\_Impulse\_Input is “Separate” then column 1 of impulse\_matrix shall contain an impulse response and column “aggressors+2” shall contain the output of step 4 and Tx2’s AMI\_Init function is executed.

Step 6.a If Tx2 Tx\_Impulse\_Input is not present or is “IBIS7.0” then the output of column 1 of step 4 is presented to the Rx2 executable model file’s AMI\_Init function and the Rx2 AMI\_Init function is executed.

Step 6.b If Tx2 Tx\_Impulse\_Input is “Combined” then the output of column 1 of step 4 is convolved with the output of step 1 and is presented to the Rx2 executable model file’s AMI\_Init function and the Rx2 AMI\_Init function is executed.

Step 6.c If Tx2 Tx\_Impulse\_Input is “DoNotCare” then the output of column 1 of step 4 is convolved with the output of step 1 if the input to column 1 was a unit impulse response and is presented to the Rx1 executable model file’s AMI\_Init function and the Rx2 AMI\_Init function is executed.

Step 6.d If Tx2 Tx\_Impulse\_Input is “Separate” then the output of column 1 of step 5 is presented to the Rx2 executable model file’s AMI\_Init function and the Rx2 AMI\_Init function is executed.

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

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 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, regardless of whether Tx2’s AMI\_GetWave exists or not, and performs simulation on channel 3, which consists of Tx2, physical channel 2, and Rx2, according to the AMI flow defined in the specification for channels without Redriver. 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.

If just doing a statistical simulation, the flow is terminated after step 7. If doing a time domain simulation, step 7 may be skipped.

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**

All models, including the Primary Tx, Redriver Rx, Redriver Tx, and Terminal Rx should specify Init\_Returns\_Impulse=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 2a. If Tx1 Tx\_Impulse\_Input is not present or is “IBIS7.0” then column 1 of impulse\_matrix shall contain the output of step 1 and Tx1’s AMI\_Init function is executed.

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

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

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

Step 3.a If Tx1 Tx\_Impulse\_Input is not present or is “IBIS7.0” then the output of column 1 of step 2 is presented to the Rx1 executable model file’s AMI\_Init function and the Rx1 AMI\_Init function is executed.

Step 3.b If Tx1 Tx\_Impulse\_Input is “Combined” then the output of column 1 of step 2 is convolved with the output of step 1 and is presented to the Rx1 executable model file’s AMI\_Init function and the Rx1 AMI\_Init function is executed.

Step 3.c If Tx1 Tx\_Impulse\_Input is “DoNotCare” then the output of column 1 of step 2 is convolved with the output of step 1 if the input to column 1 was a unit impulse response and is presented to the Rx1 executable model file’s AMI\_Init function and the Rx1 AMI\_Init function is executed.

Step 3.d If Tx1 Tx\_Impulse\_Input is “Separate” then the output of column 1 of step 2 is presented to the Rx1 executable model file’s AMI\_Init function and the Rx1 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 Tx\_Impulse\_Input is not present or is “IBIS7.0” 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 Tx\_Impulse\_Input is “Combined” then column 1 of impulse\_matrix shall contain the output of step 3 and Tx2’s AMI\_Init function is executed.

Step 5c. If Tx2 Tx\_Impulse\_Input is “DoNotCare” then column 1 of impulse\_matrix shall contain a combination of unit impulse response, output of step 1 and/or output of step 3 and Tx2’s AMI\_Init function is executed.

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

Step 6.a If Tx2 Tx\_Impulse\_Input is not present or is “IBIS7.0” then the output of column 1 of step 5 is presented to the Rx2 executable model file’s AMI\_Init function and the Rx2 AMI\_Init function is executed.

Step 6.b If Tx2 Tx\_Impulse\_Input is “Combined” then the output of column 1 of step 5 is convolved with the output of step 4 and is presented to the Rx2 executable model file’s AMI\_Init function and the Rx2 AMI\_Init function is executed.

Step 6.c If Tx2 Tx\_Impulse\_Input is “DoNotCare” then the output of column 1 of step 5 is convolved with the output of step 1 if the output of step 1 was not included in step 5c, and the output of step 3 if step 3 was not included in step5c and the Rx2 AMI\_Init function is executed.

Step 6.d If Tx2 Tx\_Impulse\_Input is “Separate” then the output of column 1 of step 4 is presented to the Rx2 executable model file’s AMI\_Init function and the Rx2 AMI\_Init function is executed.

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

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 channel 2, 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.

If just doing a statistical simulation, the flow is terminated after step 8. If doing a time domain simulation step 8 may be skipped.

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 can 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 respons of the filter’s equalization.

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

1. “IBIS7.0”
	1. This is the default and is compatible with the existing IBIS 7.0 flow with the exception that the ouput of the redriver Rx is included in the impulse response input to the terminal Rx instead of after the terminal Rx.
2. “Combined”
	1. This flow combines the ouput of the redriver Rx with the redriver Tx downstream chanel as the input to the redriver Tx instead of the output of the redriver Tx.
3. “Separate”
	1. In this flow the EDA tool presents two impulse responses to the redriver Tx. The accumulated upstream channel and the redriver Tx downstream channel.
4. “DoNotCare”.
	1. This tells the EDA tool that the equalization applied by any Tx is independent of the impulse response input to the Tx. This gives the EDA tool options on how and where the upstream and donstream channels are included in the simulation flow.