Redriver Flows That Give The Wrong Answers for Statistical Simulations

IBIS 6.1 Redriver Flow
BIRD 190 Redriver Flow
Fangyi's BIRD Draft on Additional AMI_Init IR's

BIRD 166 Redriver Flow Gives Right Answers

Walter Katz

IBIS Open Forum

June 23, 2017

SiSoft
We Are Signal Integrity

BIRD 190 Says

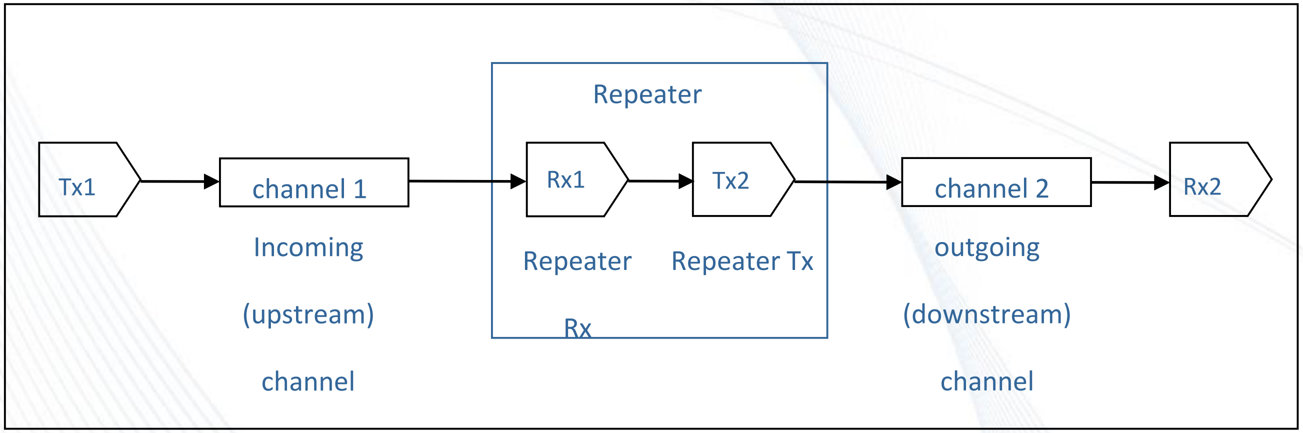
Note: The Rx2 executable model file writer for the downstream channels with Redrivers should keep in mind that the impulse response that is presented to the Rx AMI_Init function does not include the effects of the upstream equalization. Therefore, the Rx AMI_Init function will not be able to perform accurate optimization in the absence of the upstream channel characteristics and/or equalization effects. 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.

Why is BIRD 190 a Problem

- It reinforces the error in the IBIS 6.1 flow that causes wrong results if the downstream Rx has a DFE.
  - An Rx2 with a DFE has a linear (scaling) component and a non-linear (additive) component. If Rx1 (redriver receiver) has gain, then applying the Rx1 Impulse Response to the output of Rx2 will incorrectly scale the Rx2 DFE equalization.
- The “best” solution of the channel requires exploring a very large solution space that can contain between 10^6 to 10^16 simulations.
  - Just multiply the number of CTLE, AGC and DFE tap settings in modern Rx AMI models.

SiSoft™ We Are Signal Integrity

A Simple Example That Proves 166 is Right and 6.1 and 190 are Wrong, Even Without Rx2 Optimization.



- Tx1, Tx2 1V swing, no equalization
- Channel 1 and Channel 2 are ideal, no loss
  - Chanel IR = unit impulse (Dirac Delta with area=1)
- Rx1 just has a gain of 2
- Rx2 just has a 1 tap DFE with fixed -.1 coefficient
- BIRD 190 results are the same as if Rx2 DFE tap 1 was -.2
- Problem with IBIS 6.1 and BIRD 190 is that the output of Rx1 scales the DFE equalization.

Fangyi’s Draft BIRD Does Not Fix the Problem

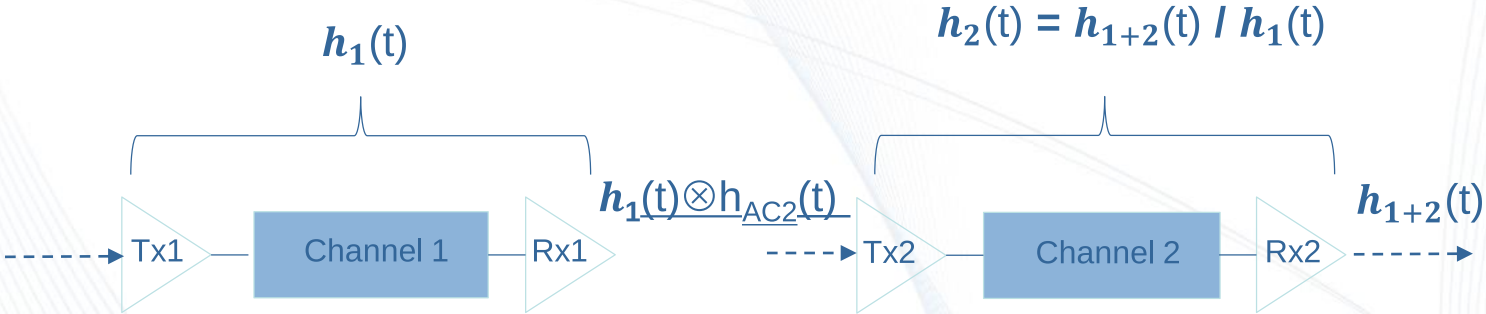
http://www.ibis.org/atm_wip/archive/20170117/fangyirao/AMI%20Simulation%20Reference%20Flow%20Enhancement%20BIRD%20draft%200/AMI_flow_enhance_BIRD_draft_v0.docx

- If all of the AMI models are enhanced to include additional Impulse Responses in their input and outputs, then there are flows that correctly include the upstream equalization for the Rx2 AMI_Init function, and handle time domain simulations when all models do not have GetWave_Exists True.
- If AMI_Init functions are not enhanced, the flows are the same as in 6.1, and therefore continue to give the wrong answer.
- Both statistical and time domain simulations are correct with BIRD 166 if all AMI Models (except Rx2) are Dual Models.
- A Model Maker can make a Dual Model as easily as modifying the AMI_Init function to read and write the additional Impulse Responses required by Fangyi’s BIRD.

Fangyi Claims BIRD166 is Not Correct for Time Domain Flow when Tx2 and Rx2 do not Have GetWave (Problem_in_BIRD166_Flow.pptx)

- Fangyi's claim is specious because he interprets the following in IBIS 6.1 incorrectly (page 143)
  - Step 8a. Redriver: The EDA tool uses the signal waveform at the output end of Rx1's algorithmic model in step 7, regardless whether Rx1's AMI_GetWave exists or not, as the stimulus of Tx2's algorithmic model, regardless 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 spec for channels without Redrivers.
- AMI flow defined in the spec for channels without Redrivers (page 178).
  - 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.
- This was written for simple channels without Redrivers. For Redrivers it should be interpreted for this case by grouping the first models that have GetWave_Exist True and grouping the last models that have GetWave_Exist False. Use the GetWave flow for the first group, and use deconvolution on the last group to determine the last group IR which is applied to the first group GetWave result.
  - not utilize the Tx1/Rx1 AMI_GetWave functionality, by treating the Tx1 and Rx1 AMI models as if there models had GetWave_Exists False.
  - use deconvolution to obtain the impulse response of the Tx2/Rx section. Since the AMI_Init function contains a linear and time invariant algorithm, the Tx2/Rx2 section equalization can be represented as an impulse response. Since the output of the Rx2 AMI_Init function is an impulse response modified by the Tx2/Rx2 section equalization, the impulse response of the Tx2/Rx2 section can be obtained by deconvolving the IR output of Rx2 with the IR output of Rx1.

Better Job of Documenting this Flow First the Statistical Flow



$h_{AC}(t)$

$h_{AC1}(t)$ – Analog channel 1 impulse response

$h_{TE1}(t)$ – Impulse response of Tx1 AMI_Init equalization

$h_{RE1}(t)$ – Impulse response of Rx1 AMI_Init equalization

$h_1(t) = h_{AC1}(t) \otimes h_{TE1}(t) \otimes h_{RE1}(t)$

$h_{AC2}(t)$ – Analog channel 2 impulse response

$h_{TE2}(t)$ – Impulse response of Tx2 AMI_Init equalization

$h_{RE2}(t)$ – Impulse response of Rx2 AMI_Init equalization

$h_{1+2}(t) = h_{AC1}(t) \otimes h_{TE1}(t) \otimes h_{RE1}(t) \otimes h_{AC2}(t) \otimes h_{TE2}(t) \otimes h_{RE2}(t)$

$h_2(t) = h_{1+2}(t) / h_1(t)$

GetWave Flow (All Have GetWave)

- Denote input digital waveform to Tx1 as d(t)
- Denote Rx1 GetWave output waveform as $V_{Rx1GW}(t)$
- Denote Rx2 GetWave output waveform as $V_{out}(t)$

BIRD166 GetWave Flow
Tx1 and Rx1 Have GetWave
Tx2 and Rx2 Do Not Have GetWave

- Denote $h_2(t) = h_{1+2}(t) / h_1(t)$ (from slide 7)
- Denote input digital waveform to Tx1 as d(t)
- Denote Rx1 GetWave output waveform as $V_{Rx1GW}(t)$
- Denote waveform at Rx2 Latch as $V_{out}(t)$

Summary

- IBIS 6.1, BIRD 190 and Fangyi's BIRD give the wrong statistical answer for Redriver channels when Rx2 either does optimization, or Rx2 contains a DFE.
- BIRD 166 gives the correct answer for both statistical and time domain flows.
- Fangyi's BIRD does eliminate deconvolution for both normal and Redriver channels when the model has GetWave_Exists=False.
  - This allows an EDA tool to create a proxy AMI_GetWave
  - The difficulty of writing an AMI_GetWave comparable to writing the code to read and write the additional IRs.

SiSoft™ We Are Signal Integrity