**BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)**

**BIRD ID#:** 166.3

**ISSUE TITLE:** *Resolving problems with Redriver Init Flow*

**REQUESTER:** *Walter Katz, Signal Integrity Software, Inc.*

*Darshan Shah, F5Networks, Inc.*

**DATE SUBMITTED:** April 2, 2014; April 18, 2017; April 26, 2017; May 11

**ANALYSIS PATH/DATA THAT LED TO SPECIFICATION:**

As currently written, the reference flow for a Redriver makes the incorrect assumption that the downstream Rx equalization is determined from the downstream Tx and the downstream channel. In order for the downstream Rx to properly determine its equalization, the impulse response input to the downstream Rx must also include the impulse response output of the upstream Rx. The current flow assumes that the downstream Tx equalization is determined by the downstream channel. This BIRD does not change that flow, in order so support a time domain flow when the Redriver Tx does not have an AMI\_GetWave.

The proposed revision corrects the Redriver statistical simulation flow by:

* Convolving the upstream equalization with the redriver Tx IR output instead of convolcing the upstream equalization with the downstream Rx output.

Make the changes indicated bellow in this section on pages 243-244:

Repeater

Rx

Tx1

Rx1

Tx2

Rx2

channel 1

channel 2

Repeater

Repeater Tx

Incoming

(upstream)

channel

outgoing

(downstream)

channel

**Figure 40 - Repeater link**

 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).

Step 1. The simulation platform 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 simulation platform 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 6a. Redriver: The simulation platform convolves the impulse response returned by Rx1’s AMI\_Init in step 3 by the output of step 5 is presented to Rx2’s AMI\_Init function and Rx2’s AMI\_Init function is executed.

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

Step 7. The simulation platform 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 8a. Redriver when Tx1, Rx1, Tx2 and Rx2 have AMI\_GetWave: The simulation platform uses the signal waveform at the output end of Rx1’s algorithmic model in step 7 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 spec for channels without Redrivers.

Step 8b. Redriver when Tx1, Rx1, Tx2 or Rx2 do not have AMI\_GetWave: The output of Step 4 according to the AMI flow defined in the specification for channels without Repeaters. is convolved with the output of Step 6a by the EDA tool and the result is passed on to Step 8 according to the AMI flow defined in the specification for channels without Repeaters.

Step 8c. Retimer: The simulation platform 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 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 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\_Reciver\_Sensitivity, the logic level is unchanged from the previous bit. The digital stimulus has values of -½ volt for logic 0 and +½ volt for logic 1.

Step 9. The simulation platform 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 simulation platform. Since the Retimer output signal is driven by a digital stimulus as described above in step 8c, jitter and noise parameters specified in Retimer .ami files are applied according to the specification for channels without Repeaters.

The statistical simulation flow for a Repeater link shown in Fig. 2 is defined below.

Step 1. The simulation platform 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 simulation platform 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 6a. Redriver: The simulation platform convolves the impulse response returned by Rx1’s AMI\_Init in step 3 by the output of step 5 is presented to Rx2’s AMI\_Init function and Rx2’s AMI\_Init function is executed.

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

Step 7a. Redriver: The simulation platform uses the impulse responses returned by Rx2’s AMI\_Init in step 6 to obtained the full channel impulse response and uses it to perform statistical simulation.

Step 7b. Retimer: The simulation platform uses the impulse responses returned by Rx1’s AMI\_Init in step 3 to perform a statistical simulation of channel 1. The simulation platform uses the impulse responses returned by Rx2’s AMI\_Init in step 6 to perform a statistical simulation of channel2.