**New Reliable AMI Flows**

Walter Katz

The MathWorks

11/17/20

IBIS AMI simulation flows have been problematic because

1. Certain combinations of Dual, Init Only and GetWave Only AMI Models
2. Tx Init models that can optimize themselves base on downstream channels

I would like to propose the following enhancements to the IBIS-AMI standard that when models are written in accordance with these enhancements simulation results will be predictable and accurate.

The first enhancement is no change to the specification at all. It suggests that all AMI models be Dual Models (they have GetWave\_Exists=True and Init\_Returns\_Impulse=True), with the exception at a Terminal Rx or a Retimer Rx can be GetWave Only (GetWave\_Exists=True and Init\_Returns\_Impulse=False). There are other configurations described below.

* Note that there should never be a need for an Init Only model. An Init Only model applies an LTI equalization filter to its input to get its output. For an EDA tool to generate a pseudo GetWave function it would have to do a deconvolution to determine the LTI equalization filter. Since the model make likely knows this LTI equalization filter it should be a simple exercise to create a GetWave function to apply this same LTI equalization filter to the input \*wave.

Things get simplified if Tx models do not adapt based on their input. Only Rx models adapt, this includes Rx CTLE, AGC, DFE, FFE:

* UpstreamIR ֎ txInit(DownstreamIR) = DownstreamIR ֎ txInit(UpstreamIR)

(Note I use the abbreviations Init == AMI\_Init, GW == AMI\_GetWave and Wav == \*wave so that the Init and GetWave flow will all fit on one line.

* Tx1 🡪 Rx1 🡪 Tx2 🡪 Rx2 🡪 Tx3 🡪 Rx3
* H1 H2 H3
* ---------- -------------------------- -------------------------- ------------
* Primary Tx Redriver 1 Redriver 2 Terminal Rx
* Init Channel is H0 Tx1 H1 Rx1 Tx2 H2 Rx2 Tx3 H3 Rx4
	+ H0 is unit IR
	+ Tx1 is primary Tx
	+ H1 is IR of channel between Tx1 and Rx1
	+ Rx1 and Tx2 is redriver 1
	+ H2 is IR of channel between Tx2 and Rx2
	+ Rx2 and Tx3 is redriver 2
	+ H3 is IR of channel between Tx3 and Rx3
	+ Rx3 is terminal Rx
* IR output of Rx3\_Init (Init == AMI\_Init)
	+ Rx3\_Init(H3֎Tx3\_Init (Rx2\_Init (H2֎Tx2\_Init (Rx1\_Init (H1֎(Tx1\_Init(H0)))))))
* Waveform output of Rx3\_GW (GW == AMI\_GetWave) (Wav == \*wave)
	+ Rx3\_GW(H3֎Tx3\_GW (Rx2\_GW (H2֎Tx2\_GW (Rx1\_GW (H1֎(Tx1\_GW(Wav)))))))

The configuration of all models being GetWave\_Exists=True is problematic because although a model may not modify its Init output, it may require an Init input to determine the equalization. The classic case of this is a terminal Rx that determines the CTLE and DFE taps based on the IR input to its AMI\_Init but chooses not to generate an equalized IR output and just generates a GetWave output based on the equalization determined in the Rx AMI\_Init. Some Rx models optimize their CTLE and AGC in Init and then further optimize their DFE in GetWave.

There is another combination of Dual, Init Only and GetWave Only AMI Models that have a clear defined flow that will give consistent and accurate results: All upstream models are Init Only, and the terminal Rx is either Init Only, GetWave Only or Dual.

The Init flow remains:

Rx3\_Init(H3֎Tx3\_Init (Rx2\_Init (H2֎Tx2\_Init (Rx1\_Init (H1֎(Tx1\_Init(H0)))))))

If the terminal Rx (Rx3) is Init\_Only or Dual then the IR output of the Terminal Rx can be used.

When the terminal Rx (Rx3) is Daul or GetWave only then the flow is

Rx3\_GW(Wav֎ H3֎Tx3\_Init (Rx2\_Init (H2֎Tx2\_Init (Rx1\_Init (H1֎(Tx1\_Init(H0))))))

This rule can be generalized a little bit if the initial models in the channel are Init Only, and then the remaining models up to the terminal Rx are Dual Models and the terminal Rx is A Dual or GetWave only model.

A different way of summarizing all the model configuration rules is as follows:

1. All models upstream of the terminal Rx must have Init\_Returns\_Impulse=True
	1. The terminal Rx must always have a valid input IR for it to be able to optimize itself for its GetWave.
2. If any upstream model in the channel has GetWave\_Exists=True, then all models downstream of this model, including the terminal Rx, must also have GetWave\_Exists=True.