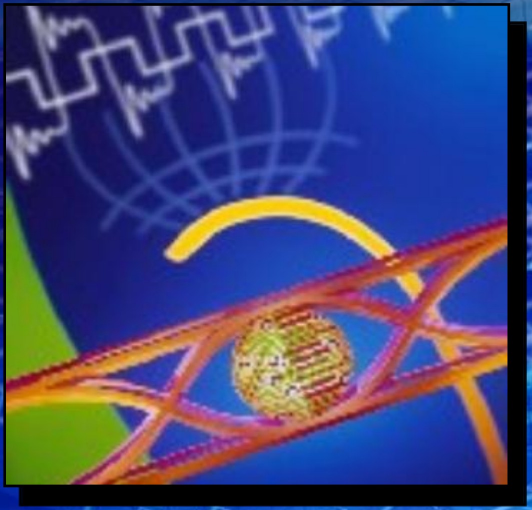
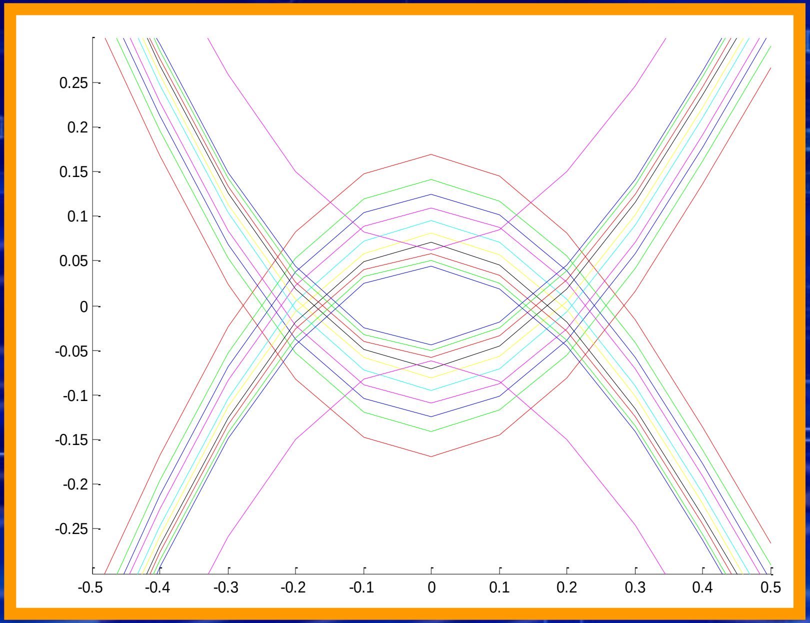
Analog Modeling Discussion

IBIS ATM & Interconnect Task Group Meetings
January 8-9, 2013





Arpad Muranyi

Mentor Graphics®

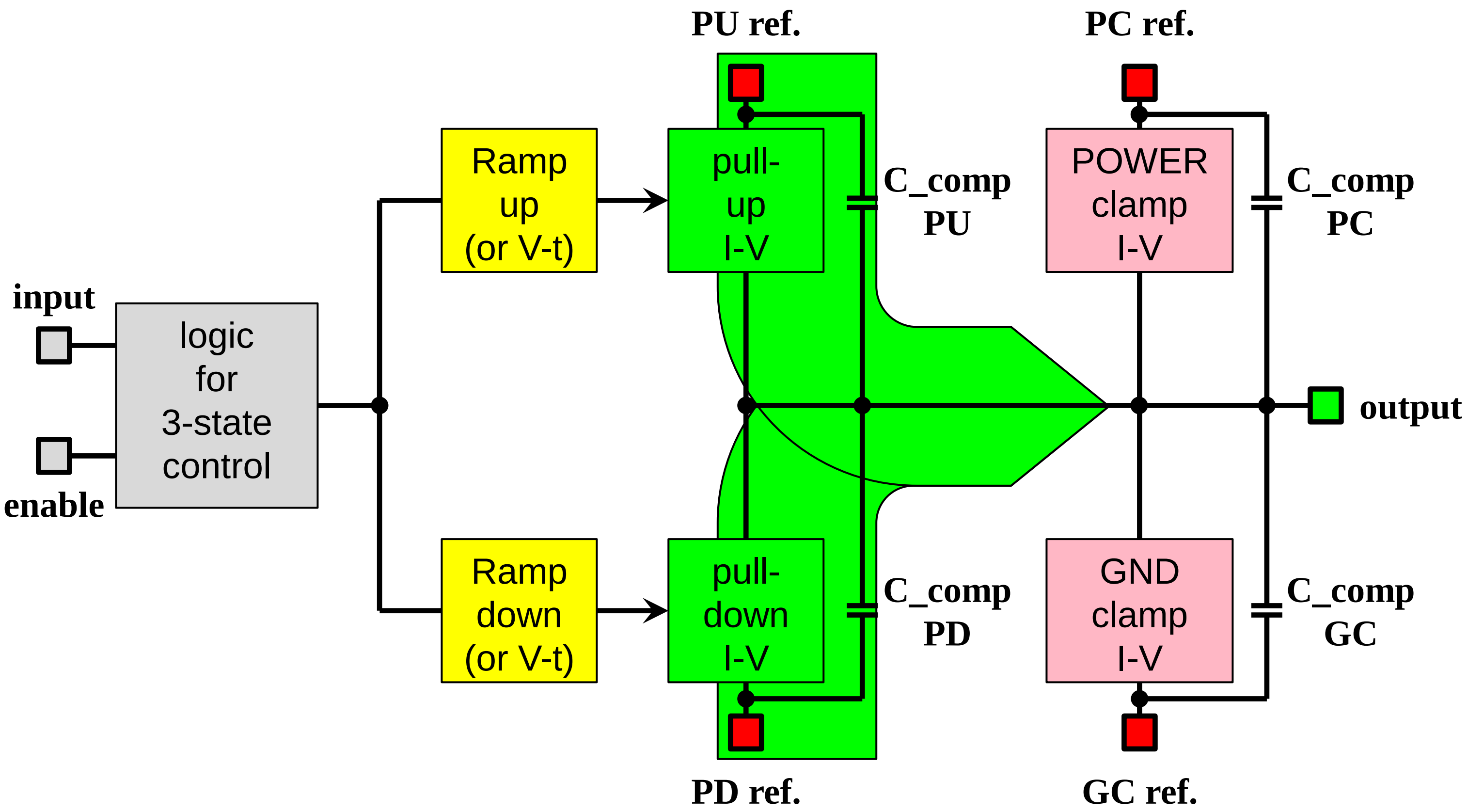
Background

- **Recent discussions in the IBIS-ATM and Interconnect Task Group meetings reached a deadlock**
- **The issue is *NOT* that there are too many BIRDs and we can’t make up are minds**
  - **BIRD 122 and its variants by SiSoft**
  - **BIRD 116, 117, 118, 129 for buffers and BIRD 125 for packages by Mentor**
  - **BIRD 144, 145 by Cadence**
- **There are differences in the interpretation of basic IBIS concepts which need to be addressed and clarified**
  - **does [Model] have an analog or digital input?**
  - **how many states/levels does [Model] have? (logic 1/0, or multiple voltage levels)**
  - **is [Model] a “shunt” (parallel) or a “through” (series) block?**
  - **is the input of [Model] accessible in the AMI flow?**
- **The correct syntax in the analog modeling proposals will depend on the answers to the above questions**

What does an IBIS [Model] do?

- **The [Pullup] or [Pulldown] I-V curves drive the output to one or the other [*** Reference] levels (supply rails)**
  - the output voltage (level) is a function of the [*** Reference] voltages
  - the output voltage (level) is independent of the voltage of the input stimulus
  - except during transitioning, the output state corresponds to the input state
- **An IBIS [Model] has one input and one enable (for I/O)**
  - they are inputs to a control logic to decide which I-V table should be activated
  - neither of these have threshold definitions in IBIS
  - neither of these are accessible through the terminals of [Model] in IBIS
- **The drive strength (output current, or impedance) of [Model] is determined by its I-V curves**
  - ultimately, the output voltage is determined by the voltage divider between the I-V curve's reference voltage, the output current defined by the I-V curve, and the load impedance
- **[Model] is not able to drive multiple voltages (levels), only two states (logic high '1' or logic low '0')**

The “transfer function” of [Model]



The main current/impedance path is between [*** Reference] and the output pad
The input stimulus acts as a selector between the pullup or pulldown path (logic state)
There is no (linear) transfer function between the input (stimulus) and the output pad
The high and low states may have a different impedance

A little detail from [Driver Schedule]

- Quote from the IBIS v5.1 specification, pg. 42 (emphasis added):

“*Description:* Describes the relative model switching sequence for referenced models to produce a multi-staged driver.”

> “The [Driver Schedule] table consists of five columns. The first column contains the model names of other models that exist in the .ibs file. The remaining four columns describe delays: Rise_on_dly, Rise_off_dly, Fall_on_dly, and Fall_off_dly. **The t=0 time of each delay is the *event* when the EDA tool’s internal pulse initiates a rising or falling transition.**
>
> ...

- “Internal pulse” refers to the stimulus of [Model]
- In the absence of the stimulus waveform crossing a predefined threshold, this “event” can only be a digital event

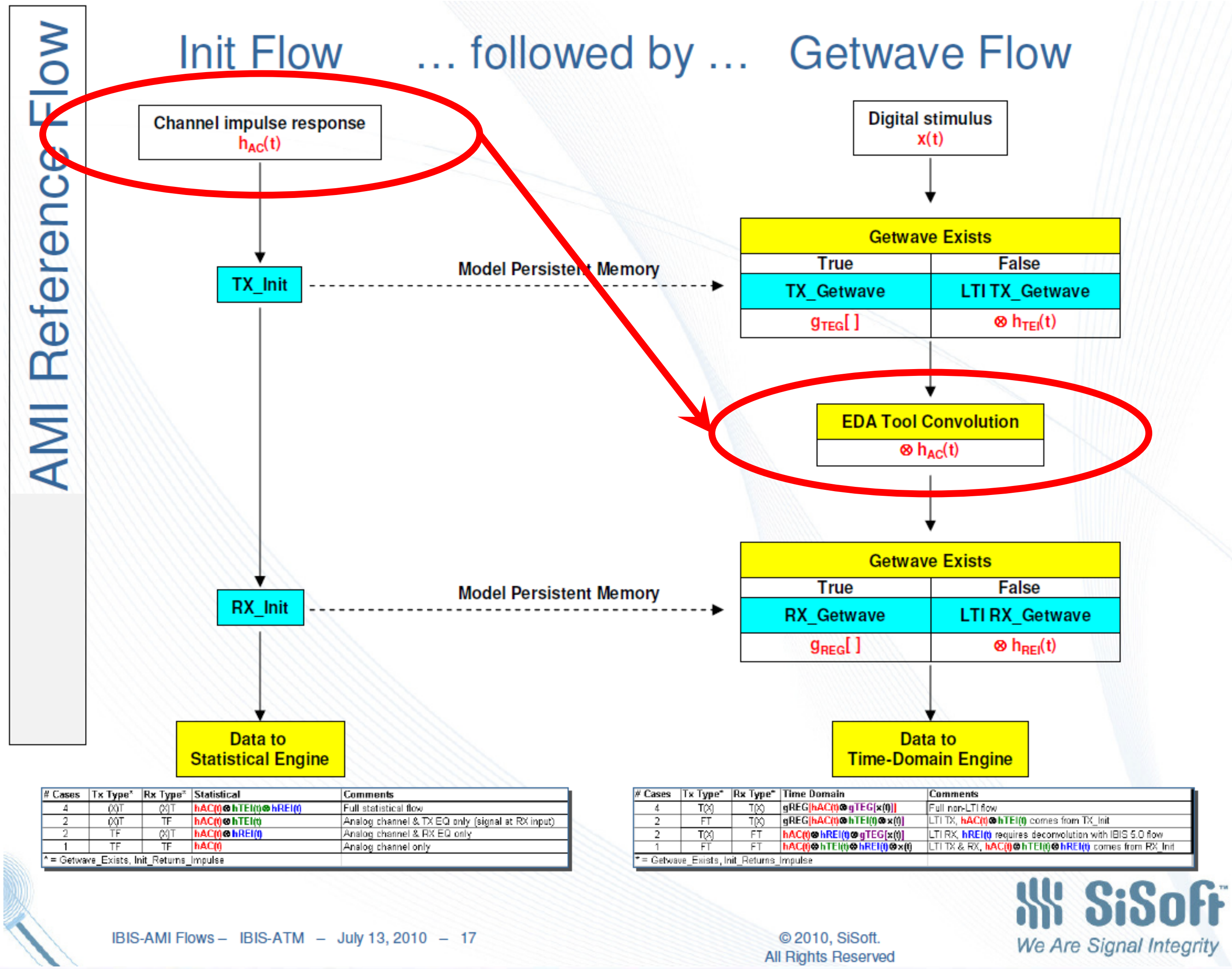
Analog modeling proposals for IBIS-AMI

- **The main idea is to replace the I-V and V-t curve based legacy [Model] with an RC circuit or S-parameter model**
- **The area of disagreements is mostly related to the "core side" of the model**
  - **For Tx: Where is the input, what is the input (digital or analog)**
  - **For Rx: Where is the output, what is the output**
  - **What are the boundary conditions, (ideal sources/loads, or matched Z)**
  - **How do we access the Tx input and Rx output terminals**
  - **etc…**

An additional twist with the AMI flow

- If the analog models are LTI (such as an RC circuit or an S-parameter model), in theory they could be used in the time domain AMI flow “directly”
- However, the v5.1 IBIS specification spells out the following flow (in Section 10.2.3.2, pg. 161)
  - the EDA tool generates an impulse response (IR) of the channel
  - the IR is passed to the AMI_Init function of Tx, then from Tx to Rx
  - the EDA tool passes a digital bit pattern to Tx AMI_GetWave
  - the output of Tx AMI_GetWave is convolved by the EDA tool with the IR of the channel and the result is passed into the Rx AMI_GetWave
  - the output of Rx AMI_GetWave is post processed by the EDA tool and presented to the user in the form of eye diagrams, BER curves, etc…

from: http://www.vhdl.org/pub/ibis/macromodel_wip/archive/20100713/toddwesterhoff/IBIS-AMI%20Flows/Flows_July2010-v2.pdf (slide 17)



# Cases	Tx Type*	Rx Type*	Statistical	Comments
4	(X)T	(X)T	hAC(t)⊗hTEI(t)⊗hREI(t)	Full statistical flow
2	(X)T	TF	hAC(t)⊗hTEI(t)	Analog channel & TX EQ only (signal at RX input)
2	TF	(X)T	hAC(t)⊗hREI(t)	Analog channel & RX EQ only
1	TF	TF	hAC(t)	Analog channel only
* = Getwave_Exists, Init_Returns_Impulse				

# Cases	Tx Type*	Rx Type*	Time Domain	Comments
4	T(X)	T(X)	gREG[hAC(t)⊗gTEG[x(t)]]	Full non-LTI flow
2	FT	T(X)	gREG[hAC(t)⊗hTEI(t)⊗x(t)]	LTI TX, hAC(t)⊗hTEI(t) comes from TX_Init
2	T(X)	FT	hAC(t)⊗hREI(t)⊗gTEG[x(t)]	LTI RX, hREI(t) requires deconvolution with IBIS 5.0 flow
1	FT	FT	hAC(t)⊗hTEI(t)⊗hREI(t)⊗x(t)	LTI TX & RX, hAC(t)⊗hTEI(t)⊗hREI(t) comes from RX_Init
* = Getwave_Exists, Init_Returns_Impulse				

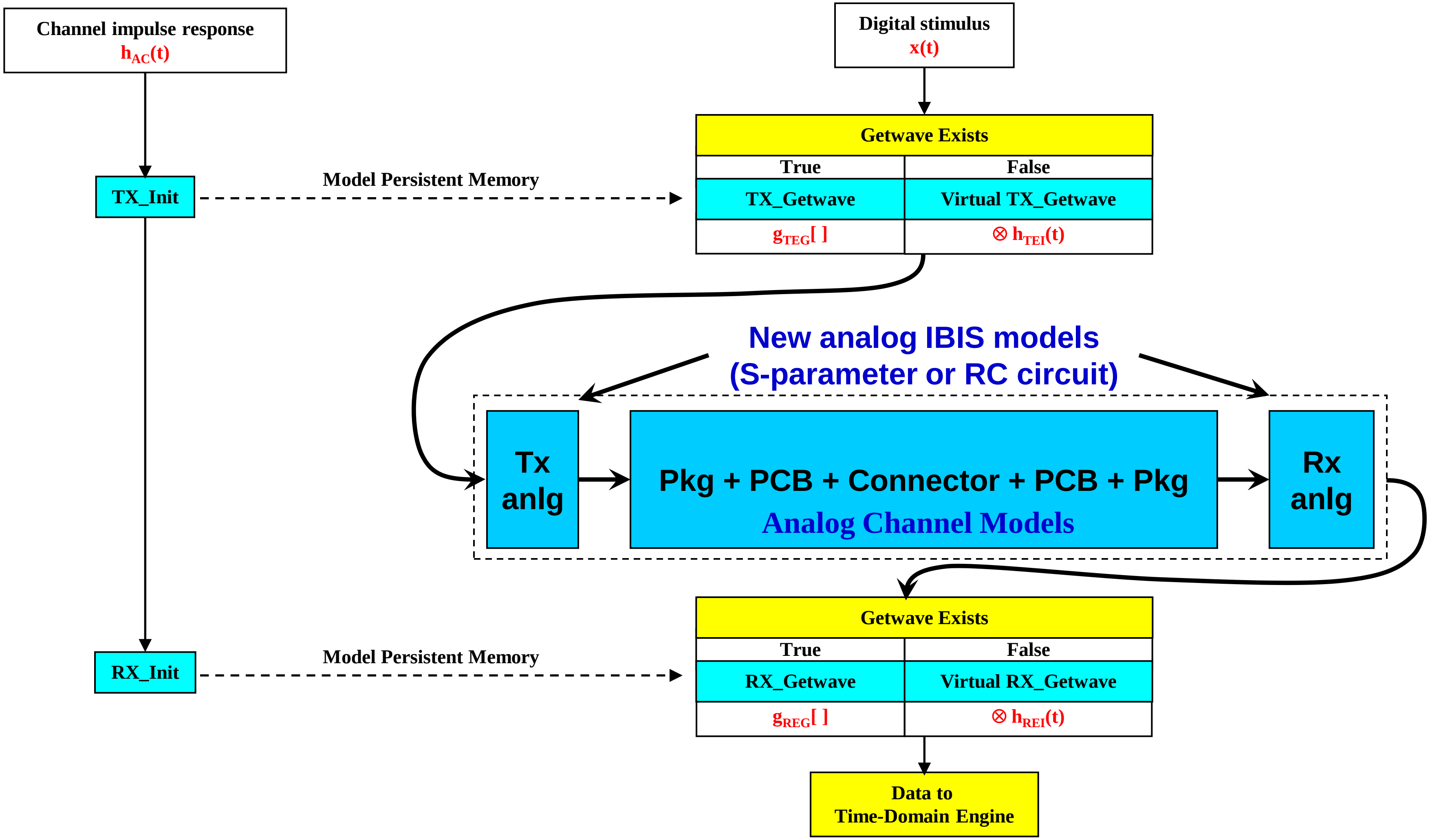
IBIS-AMI Flows – IBIS-ATM – July 13, 2010 – 17

© 2010, SiSoft.
All Rights Reserved

SiSoft™
We Are Signal Integrity

Analog Modeling Discussions

The alternate AMI_GetWave flow



Channel impulse response
$h_{AC}(t)$

Digital stimulus
$x(t)$

TX_Init

Model Persistent Memory

Getwave Exists	
True	False
TX_Getwave	Virtual TX_Getwave
$g_{TEG}[\ ]$	$\otimes\ h_{TEI}(t)$

New analog IBIS models
(S-parameter or RC circuit)

Tx anlg

Pkg + PCB + Connector + PCB + Pkg
Analog Channel Models

Rx anlg

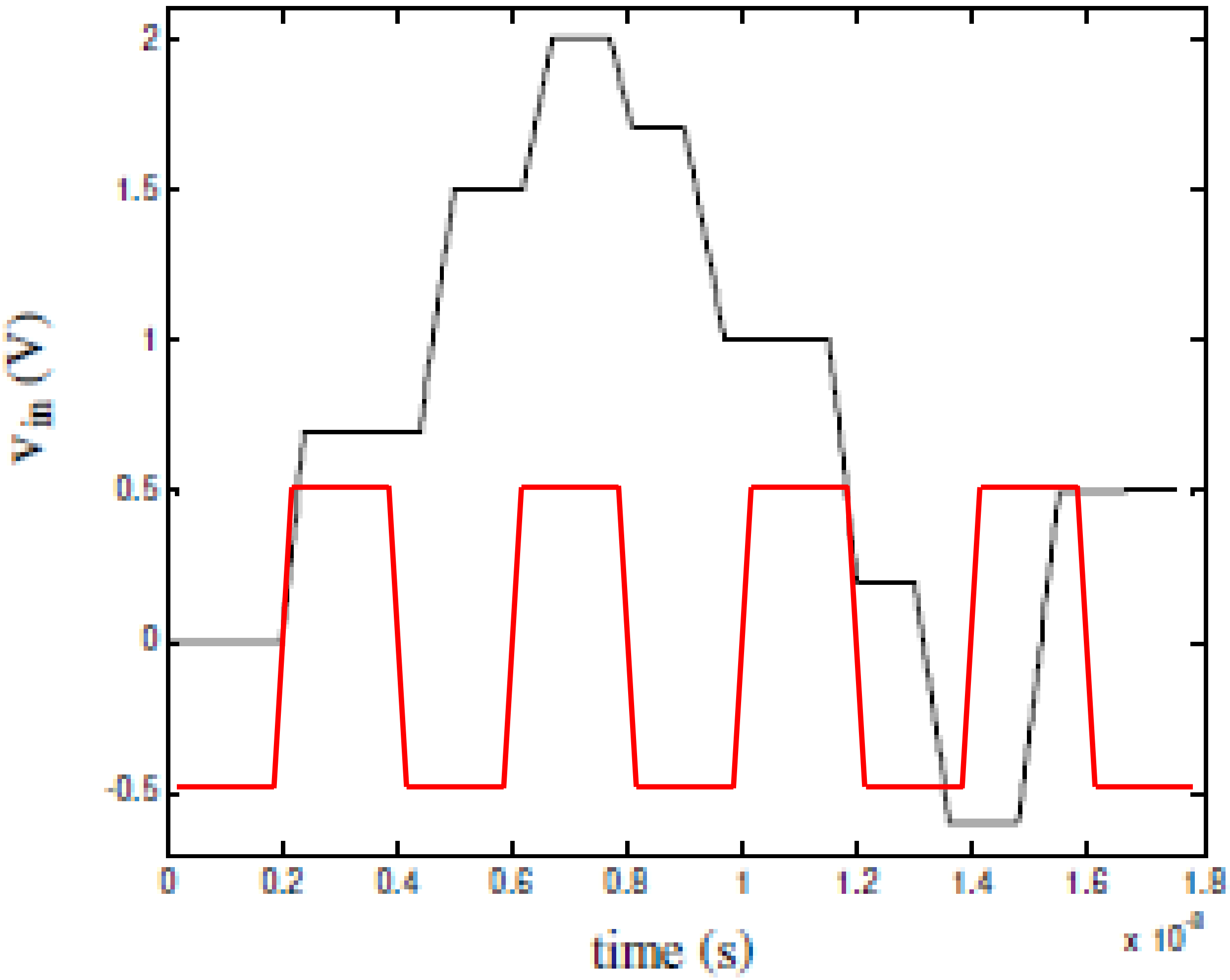
RX_Init

Model Persistent Memory

Getwave Exists	
True	False
RX_Getwave	Virtual RX_Getwave
$g_{REG}[\ ]$	$\otimes\ h_{REI}(t)$

Data to
Time-Domain Engine

The input and output of Tx AMI_GetWave

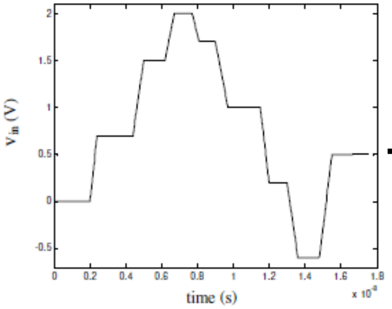


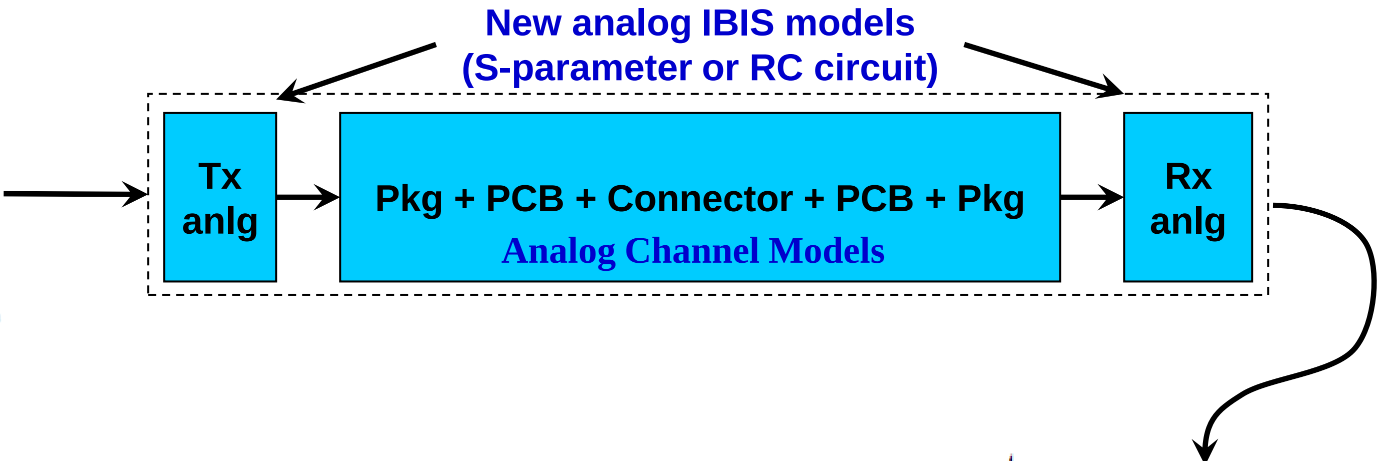
input: "A digital stimulus waveform is 0.5 when the stimulus is "high", -0.5 when the stimulus is "low", and may have a value between -0.5 and 0.5 such that transitions occur when the stimulus crosses 0." (IBIS v5.1 pg. 162)

output: The AMI_GetWave function "scales" the digital input waveform bit by bit by applying the tap coefficients according to the filtering algorithm in the Tx.

This is a "semi-digital" waveform, in which the transitions are ideal, but the voltage levels are meaningful.

The alternate AMI_GetWave flow





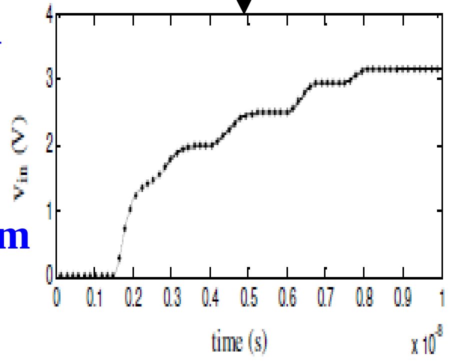
New analog IBIS models
(S-parameter or RC circuit)

Tx anlg

Pkg + PCB + Connector + PCB + Pkg

Analog Channel Models

Rx anlg



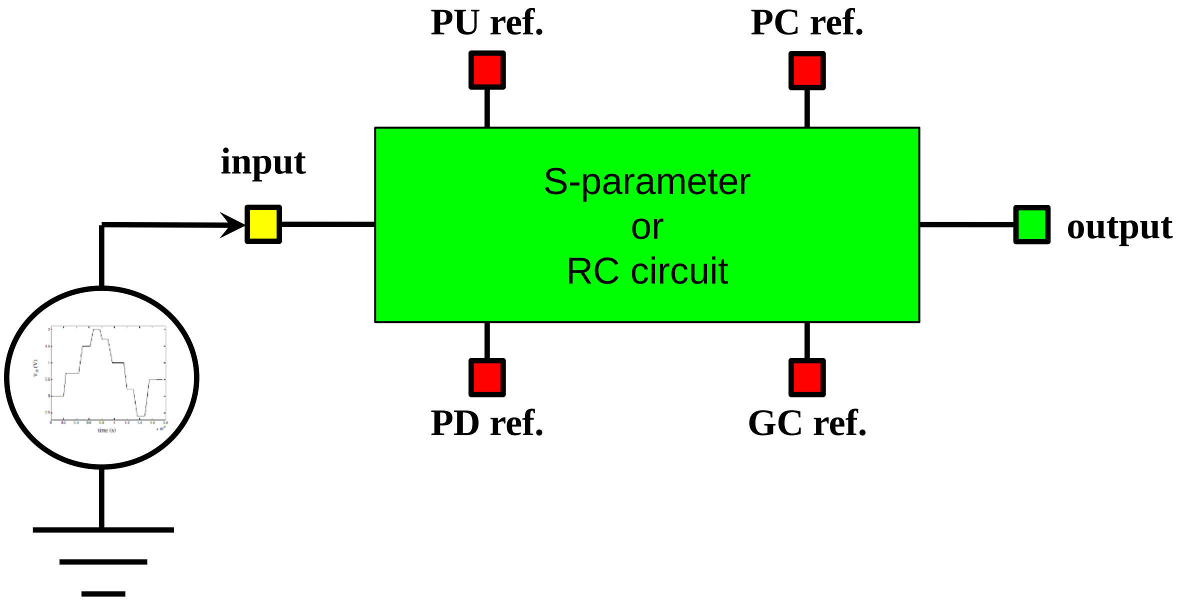
This flow is numerically equivalent to the flow described in the IBIS v5.1 specification

However, in this flow

- **the input of the analog Tx model is an analog (or semi digital) waveform**
- **the amplitude of the Tx [Model]'s output is proportional to its input**
- **the input waveform for Tx and the output waveform for Rx [Model]s are accessible**
- **The output of the analog Rx model is an analog waveform**

None of these requirements are available in IBIS v5.1

The alternate analog model (for Tx)



Here the amplitude variation of the input stimulus represents the filtered digital waveform coming from the output of Tx AMI_GetWave

BIRD 116 proposes a very similar capability for [Model] & [External Model] with IBIS-ISS *except* the input source (D_to_A converter) can have only two levels

The letter of the law in IBIS

- On pg. 161, in Section 10.2.3 the IBIS v5.1 specification says the following:

"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."

- The alternate AMI flow is numerically equivalent to the flow in the specification, but [Model] can't support it
- Pg. 122 in the Introduction of Section 6C says:

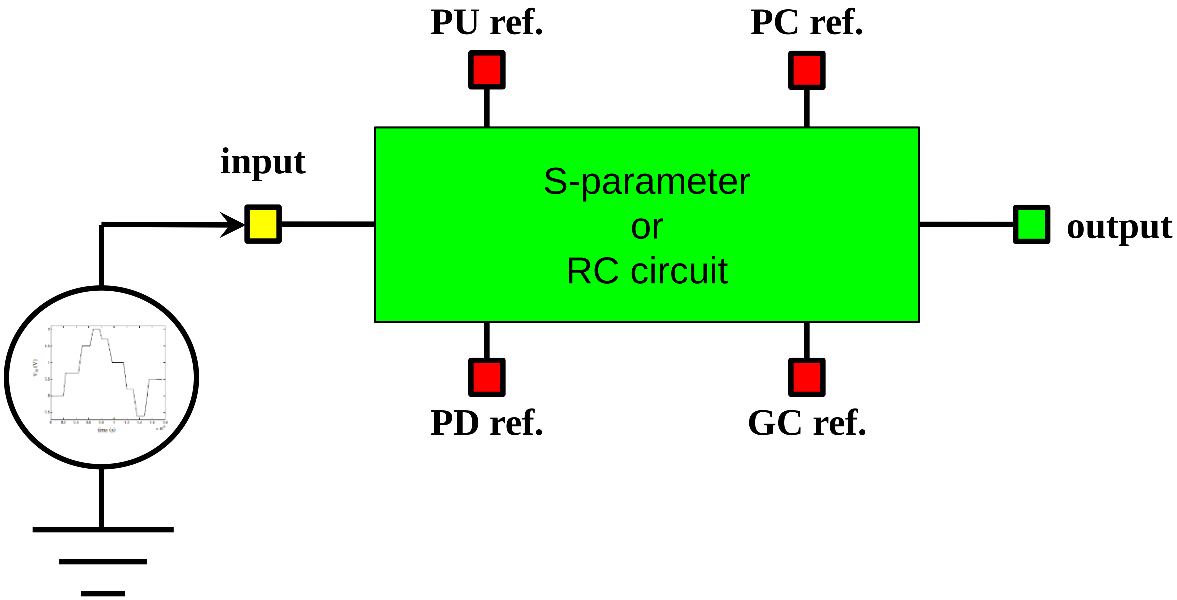
"The "analog" portion of the channel is characterized by means of an impulse response leveraging the IBIS constructs for device models defined in Sections 6, 6A and 6B."

which implies that analog [Model]s were not intended to be used directly in the IBIS-AMI flow as suggested by the alternate AMI flow

How should we change IBIS?

- In order to support the alternate AMI flow with the proposed analog buffer models we could try to modify the [Model] keyword
  - add mechanism to reference RC circuits or Touchstone files (IBIS-ISS)
  - change the definition of the stimulus to allow analog input waveforms
  - provide access (terminal) to the input stimulus of Tx buffers so that the output waveform of AMI_GetWave function (or perhaps any other waveform) cold be used to drive it
  - provide access (terminal) to the output of Rx buffers so that the analog waveform could be used as input to the AMI_GetWave function
- These are substantial changes to [Model] which may make it too complicated and confusing
- There are other complications with using [Model] this way

Supply terminals?



A Thevenin equivalent’s “power” comes from the stimulus source

Is there a need for the [*** Reference] terminals in [Model] for this approach?

If not, these models would not account for power delivery effects

If yes, is it possible to make S-parameter models which load the power terminals but get scaled by the input waveform? (This is non-LTI and probably a moot point for AMI)

Differential or single ended?

- **From pg. 123 of the IBIS v5.1 specification:**

  “The [Algorithmic Model] always processes a single waveform regardless whether the model is single ended or differential. When the model is differential, the waveform passed to the [Algorithmic Model] must be a difference waveform.”

- **However, [Model] is single ended in IBIS!**
  - **differential buffers are modeled using two separate [Model]s**
  - **in normal simulations the RC or S-parameter impedance would have to describe the single ended impedance of the buffer**
  - **in the AMI flow the same blocks would have to contain the differential impedance of the buffer**

- **The same model could not be used both ways**
  - **this could be confusing to the model maker (the number of incorrect models may increase)**
  - **this can confuse the user, not knowing which model is appropriate for which simulation**

Conclusions

- [Model] is a two-state model, not a multi-level model
- The input to [Model] is digital (two states, no levels)
- The transfer function(s) of [Model] is (are) between the power rails and the output pad (not “input to output”)
- The output of [Model] is independent of the input “levels”
- A Touchstone or ISS reference in [Model] will ***NOT*** turn [Model] into an “input to output” (series) model
- [Model] will ***NOT*** work in the AMI_GetWave flow as is
- ***Recommend*** to invent a different “model” keyword for the AMI_GetWave flow (and other “advanced” uses)

Backup slides

Multi-tap buffers (e.g. de-emphasis)

"In serial data transmission, de-emphasis … is to reduce the *level* of all bits except the first one after a transition. That causes the high frequency content due to the transition to be emphasized compared to the low frequency content which is de-emphasized. This is a form of transmitter equalization; it compensates for losses over the channel which are larger at higher frequencies. Well known serial data standards such as PCI Express, SATA and SAS require transmitted signals to use de-emphasis."

http://en.wikipedia.org/wiki/De-emphasis

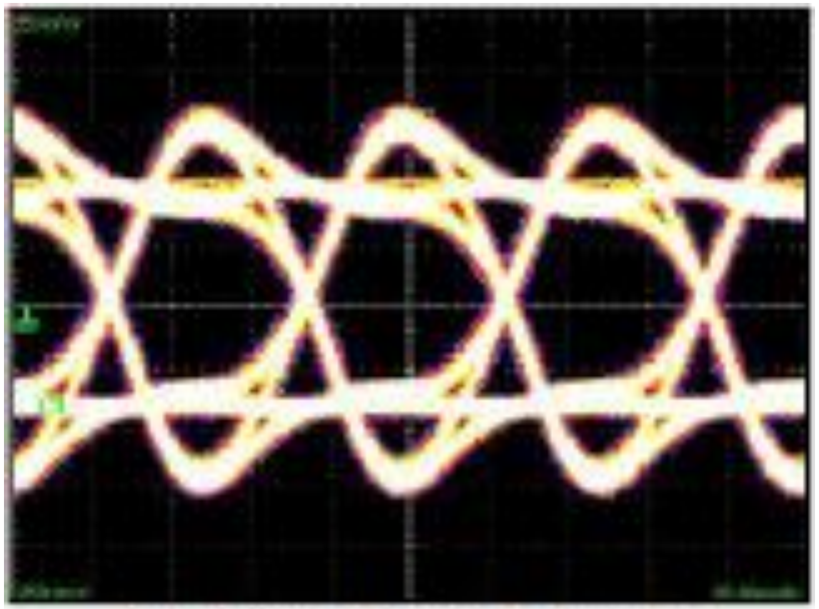
Note:

- **the buffer has a one bit digital input, i.e. two states: 1 or 0**
- **the output amplitude (level) is independent of the input amplitude or state**
- **the output amplitude (level) depends on the bit sequence, i.e. data bit rate**
- **equalizers are usually implemented using delay/summation logic, which drive multiple analog output stages connected in parallel (taps)**
- **de-emphasis or pre-emphasis are synonyms; only the point of reference is different between the two**

from: http://www.picosecond.com/objects/30gb-premphasis-pam.pdf (pg. 1)

Strong bit

Weak bit



High ‘1’ state

Low ‘0’ state

25 Gb/s Data Signal with 2-tap Pre-emphasis

Note:

- the two drive strength levels (strong/weak) are clearly visible
- there is only one eye opening in the vertical direction, representing two states (high/low or 1/0)

Multi-level buffers (e.g. PAM-4)

“Pulse-amplitude modulation, acronym PAM, is a form of signal modulation where the message information is encoded in the amplitude of a series of signal pulses. It is an analog pulse modulation scheme in which the amplitude of train of carrier pulse are varied according to the sample value of the message signal.

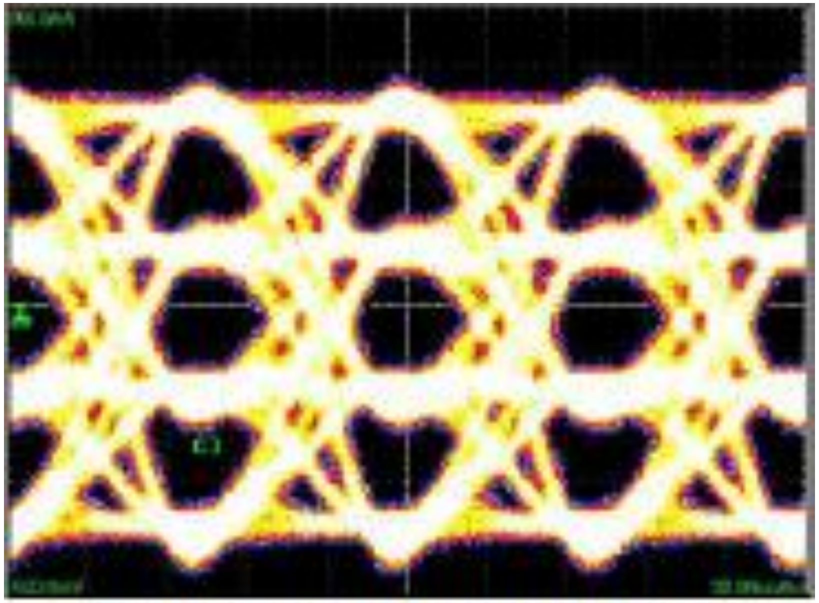
Example: A two-bit modulator (PAM-4) will take *two bits* at a time and will map the signal amplitude to *one of four possible levels*, for example −3 volts, −1 volt, 1 volt, and 3 volts.”

http://en.wikipedia.org/wiki/Pulse-amplitude_modulation

Note:

- **the buffer has a two bit digital input, i.e. four states: 00, 01, 10, 11**
- **each output amplitude (level) corresponds to one of the four states**
- **this is basically a digital to analog conversion**
- **the output amplitude (level) is independent of the bit sequence**

from: http://www.picosecond.com/objects/30gb-premphasis-pam.pdf (pg. 4)



20Gb/s PAM4 eye diagram

state ’11’

state ’10’

state ’01’

state ’00’

Note:

- multiple drive levels represent four different states
- there are multiple eye openings in the vertical direction

Let’s try to use correct terminology

- Emerging technologies often use different words to describe the same thing or the same word to describe different things until someone coins a universally accepted term
  - pre-emphasis, or de-emphasis buffers are really the same thing
  - the above two terms refer to a subset of multi-tap buffers
  - etc…
- To me personally it seems to be incorrect to call multi-tap buffers multi-level buffers
  - multi-tap buffers usually implement digital filters and/or equalizers
  - multi-level buffers are conceptually digital to analog converters

What does [Driver Schedule] do?

- **Quote from the IBIS v5.1 specification, pg. 42:**

“*Description:* Describes the relative model switching sequence for referenced models to produce a multi-staged driver.”

“The [Driver Schedule] table consists of five columns. The first column contains the model names of other models that exist in the .ibs file. The remaining four columns describe delays: Rise_on_dly, Rise_off_dly, Fall_on_dly, and Fall_off_dly. The t=0 time of each delay is the *event* when the EDA tool’s internal pulse initiates a rising or falling transition. All specified delay values must be equal to or greater than 0. There are only five valid combinations in which these delay values can be defined:

1) Rise_on_dly with Fall_on_dly

2) Rise_off_dly with Fall_off_dly

3) Rise_on_dly with Rise_off_dly

4) Fall_on_dly with Fall_off_dly

5) All four delays defined”

Pre/de-emphasis buffers in IBIS

- Using [Model] and [Driver Schedule] one can model pre/de-emphasis buffers in IBIS with legacy keywords
- The two scheduled [Model]s represent the “main” and the “boost” taps (legs) of the buffer
- The “main” buffer is scheduled without delays
- The “boost” buffer is scheduled with a one-bit delay and driven with an inverted stimulus
- Due to the simplicity of the delays in [Driver Schedule] this technique only supports two-tap equalizers, i.e. pre/de-emphasis buffers

from: http://www.eda.org/ibis/summits/jan05/muranyi.pdf (slide 7)

Pre/de-emphasis buffer review

In most of the current two-tap designs the "emphasis stimulus pattern" is a one bit delayed and inverted copy of the "input stimulus pattern"

This is not necessarily true for all pre/de-emphasis buffer designs. The delay may not be a one bit duration in each design, and multi-tap configurations would usually have a more complicated stimulus logic.

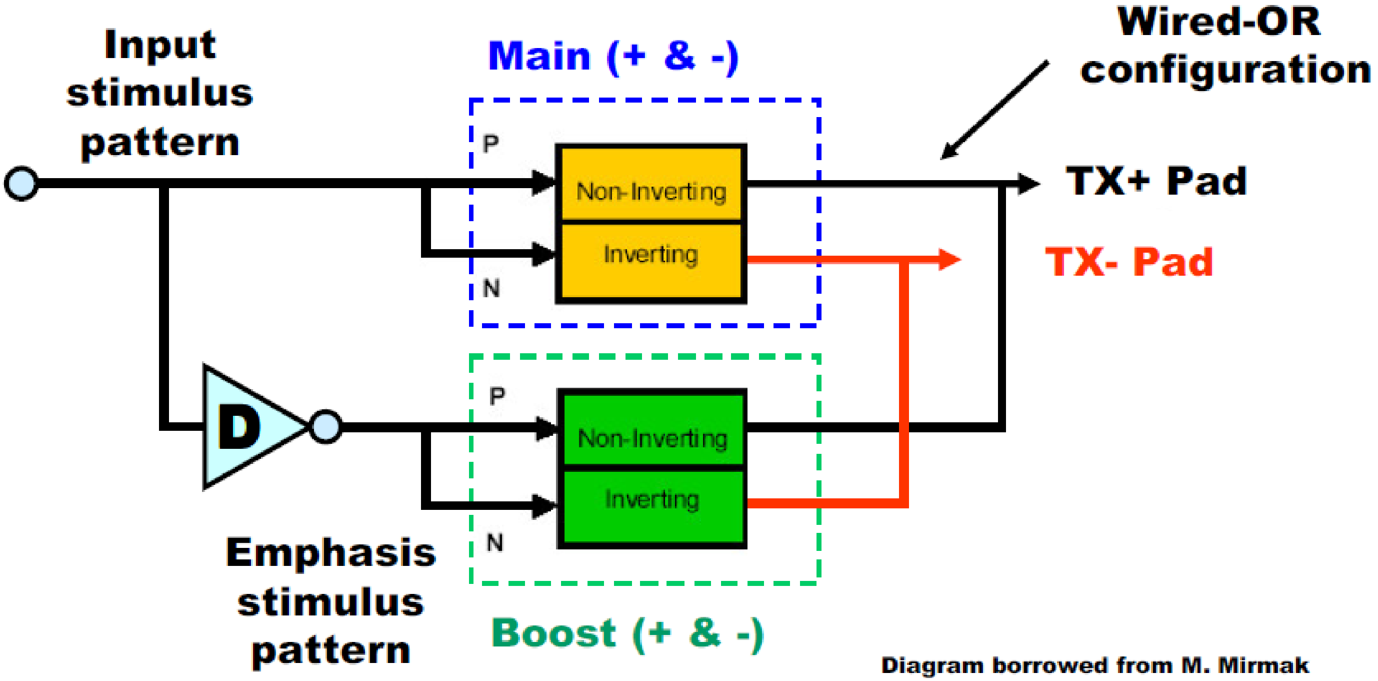
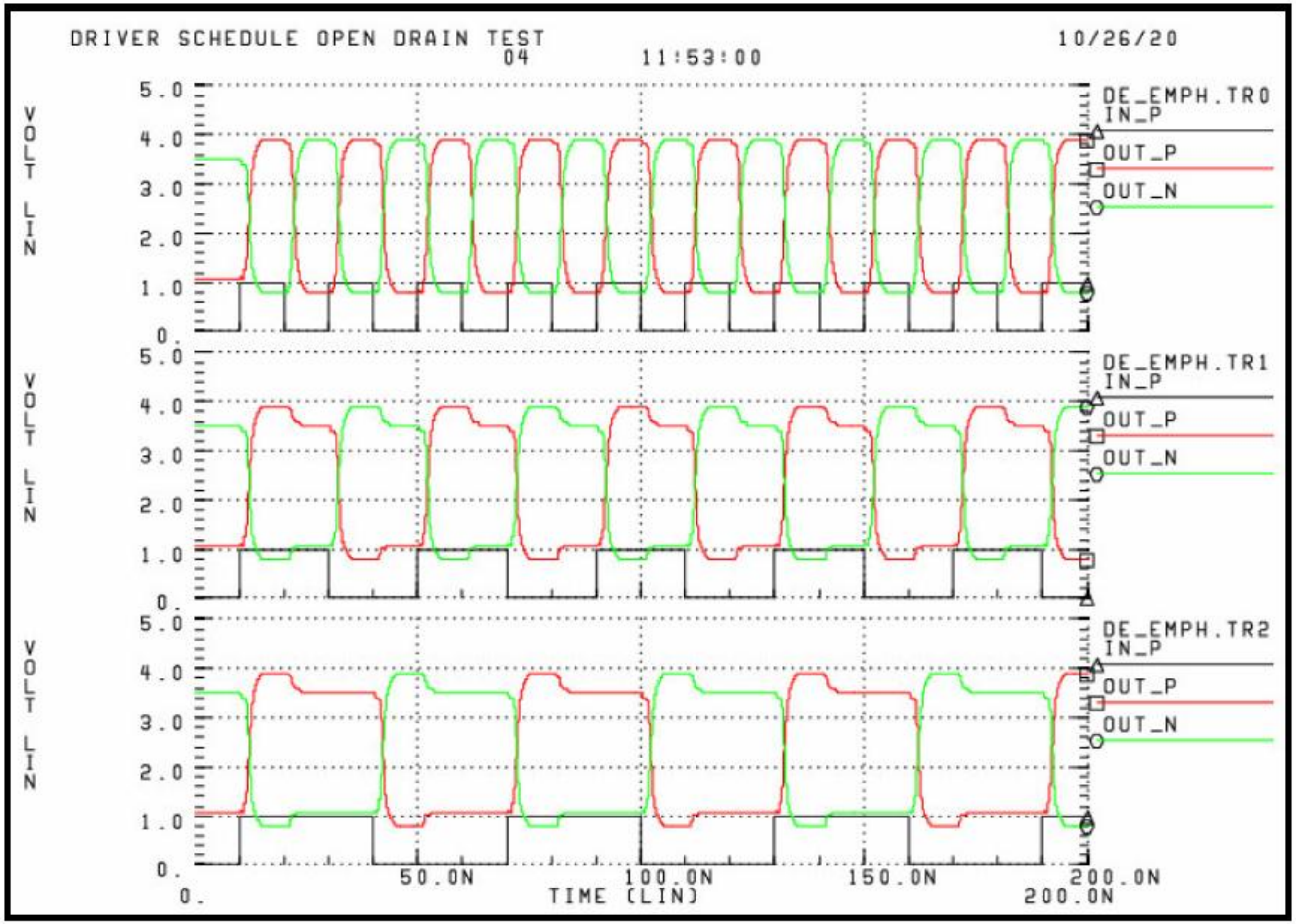


Diagram borrowed from M. Mirmak

from: http://www.eda.org/ibis/summits/jan05/muranyi.pdf (slide 9)

Simulation results with the [Driver Schedule] keyword

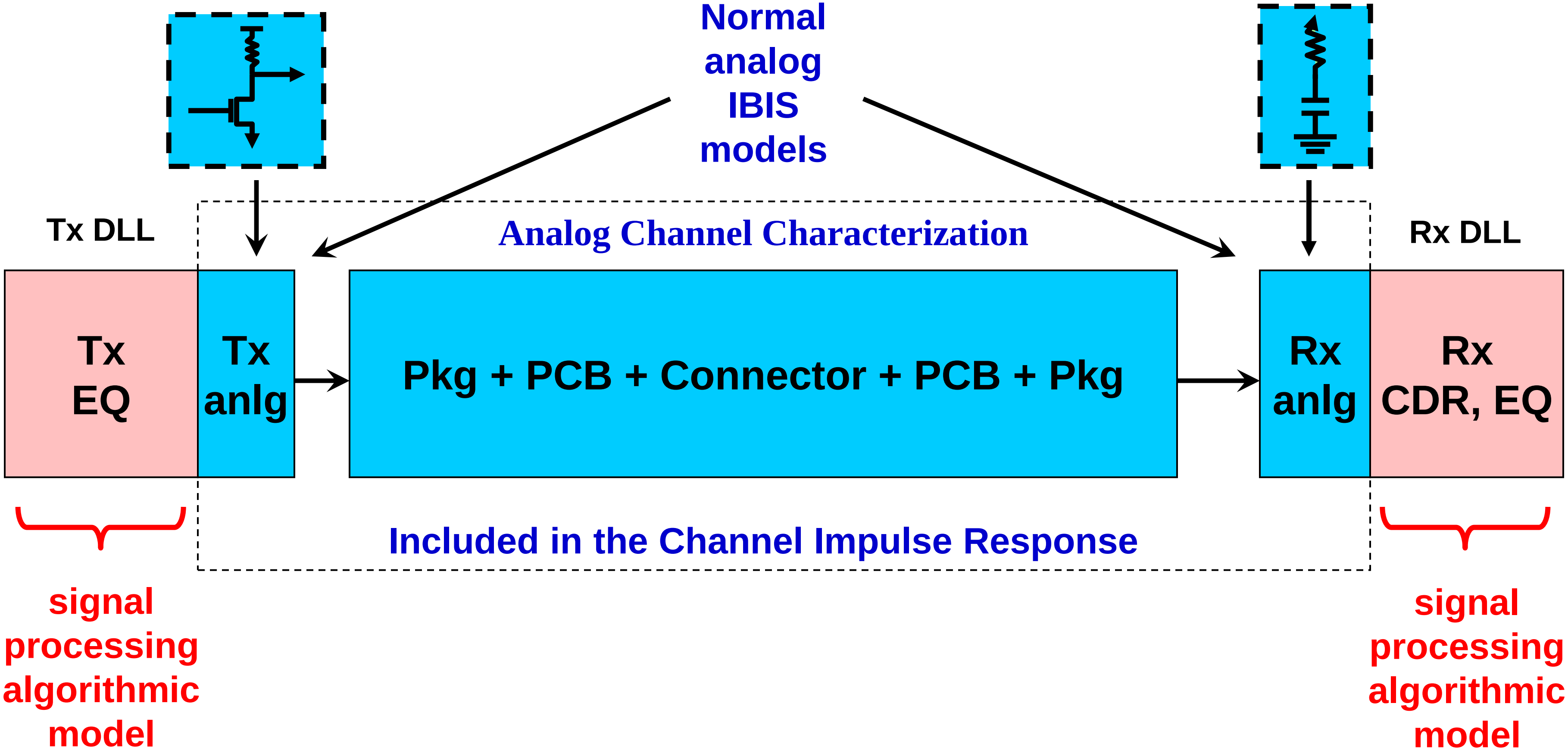


Stimulus: 0101010…(top), 0110011…(middle), 0111000111…(bottom)

Notes on the previous waveforms

- **Each [Model] represents a single tap inside the buffer**
- **The individual [Model]s can still only output two states**
- **The bit pattern dependent output drive strength (“level”) is achieved by connecting multiple [Model]s in parallel and applying a specific stimulus to them**
- **This is consistent with the physical implementation of multi-tap buffers on the die**

Block diagram of the IBIS-AMI approach



Normal analog IBIS models

Tx DLL

Rx DLL

Analog Channel Characterization

Tx EQ

Tx anlg

Pkg + PCB + Connector + PCB + Pkg

Rx anlg

Rx CDR, EQ

Included in the Channel Impulse Response

signal processing algorithmic model

signal processing algorithmic model

Mentor Graphics®

www.mentor.com