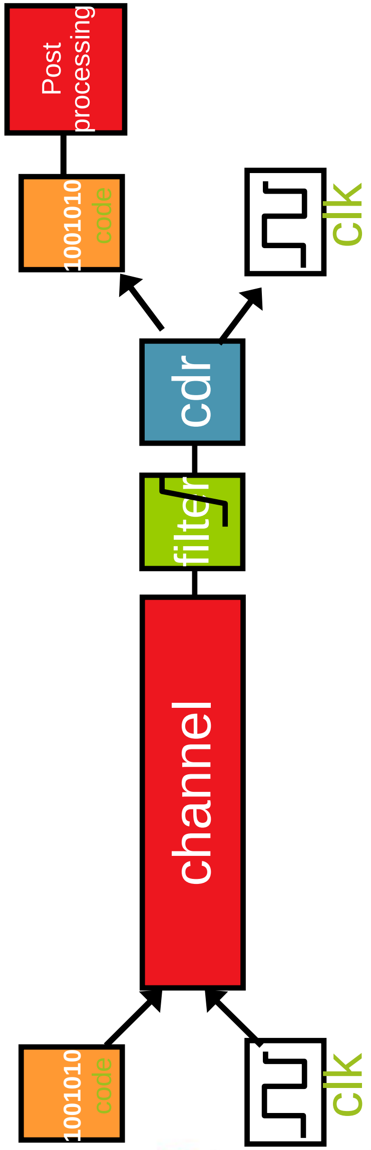
AMI Proposal Background

C. Kumar, Architect
Cadence Design Systems, Inc.

Oct 10, 2006

Advanced Serdes Modeling Challenges

- For 5+Gbps Serdes devices, complex signal processing ***algorithms*** often need to be represented, like:
  - FFE/DFE tap coefficient optimization (with/without crosstalk)
  - CDR algorithms
  - proprietary noise cancellation techniques
  - proprietary post-processing of data
- Architectural level exploration required
  - Algorithms are easy to represent and already exist at design level
  - They are typically modeled in higher level programming languages like C or Matlab
- These algorithms are very difficult to represent with traditional device modeling techniques
  - Long run times even if you can create them
- There is currently no industry-standard way to represent algorithms
  - IP suppliers have developed & distributed their own proprietary tools, increasing their support costs
  - No interoperability for systems company users

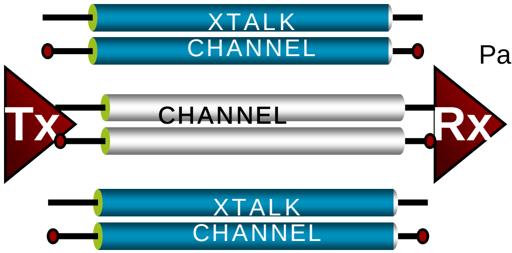


Simple API

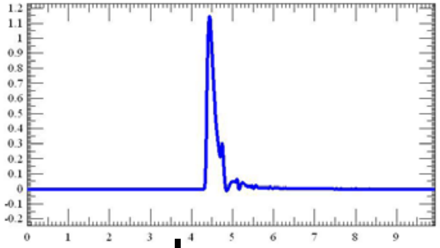
- Init
  - Initialize and optimize channel with Tx / Rx Model
  - This is where the IC DSP decides how to drive the system: e.g., filter coefficients, channel compensation, …
  - Input: Channel Characterization, system and dll specific parameters from config file
    - bit period, sampling intervals, # of forward/backward coefficients, …
  - Output: Modified Channel Characterization, status
- GetWave
  - Modify continuous time domain waveform [CDR, Post Processing]
  - Input: Voltage at Rx input at specific times
  - Output: Modified Voltage, Clock tics, status
- Close
  - Clean up, exit

Parameters passed by the system simulation platform are in red

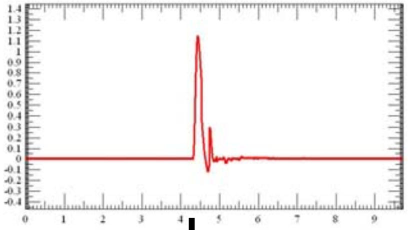
AMI_init



Pass characterization in matrix ‘a’



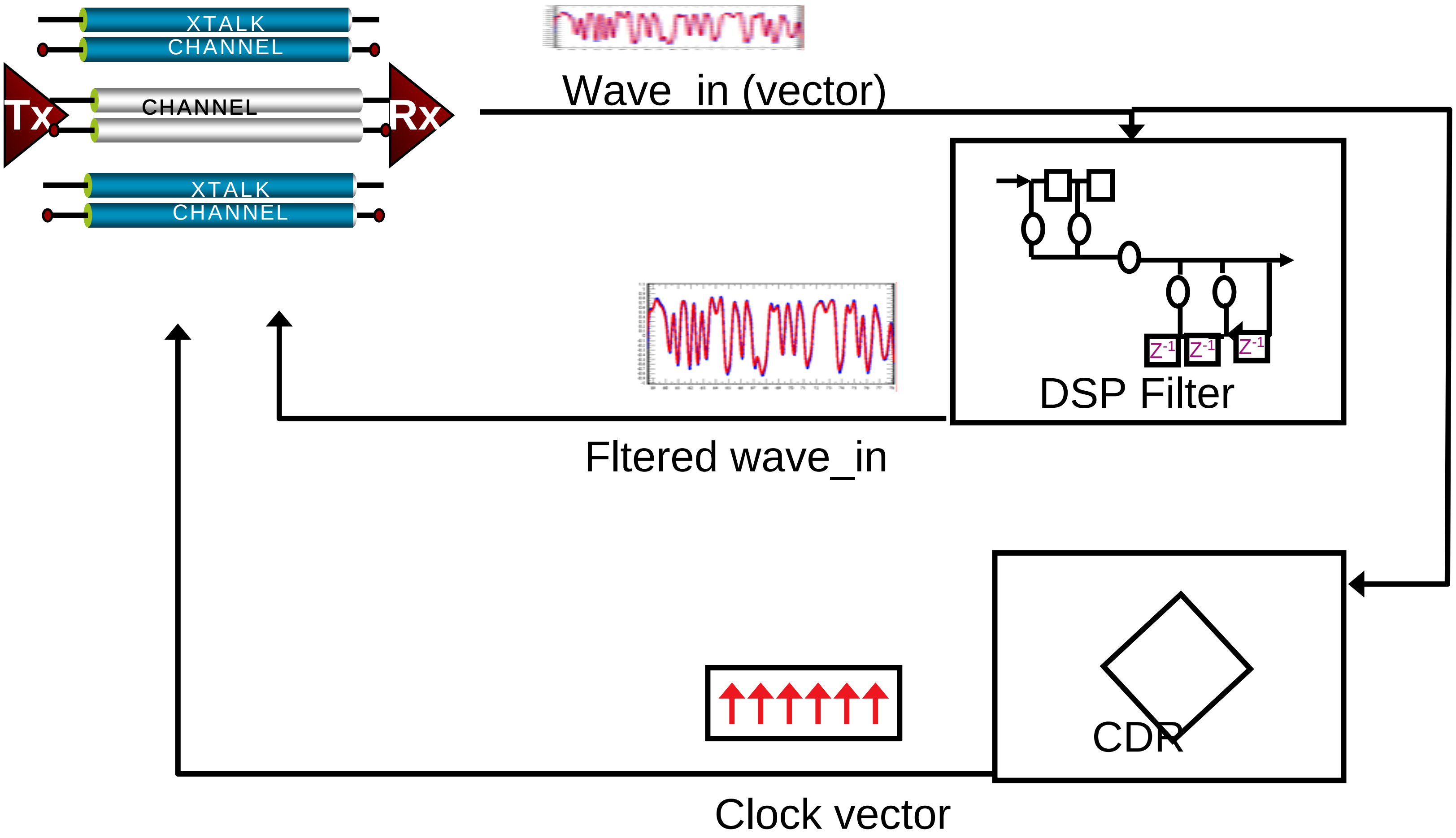
Dsp algorithms , modify characterization



Internal storage

Send modified char back (modify matrix ‘a’)

AMI_getwave



Wave in (vector)

DSP Filter

Fltered wave_in

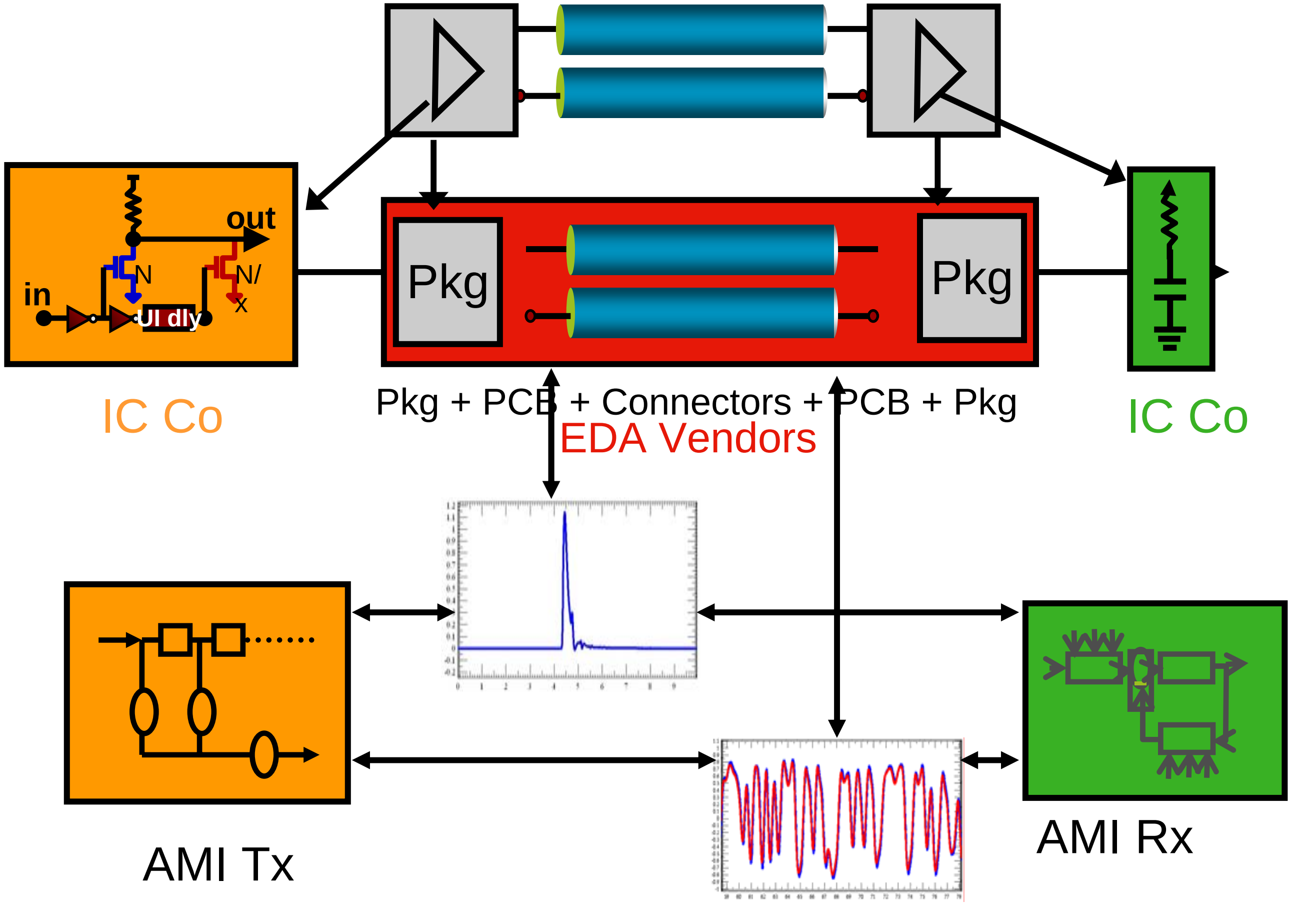
CDR

Clock vector

Proposal is evolutionary

- Leverages existing infrastructure
- Tx front end + channel + rx front end impulse characterization can be done using existing infrastructure

Evolutionary platform



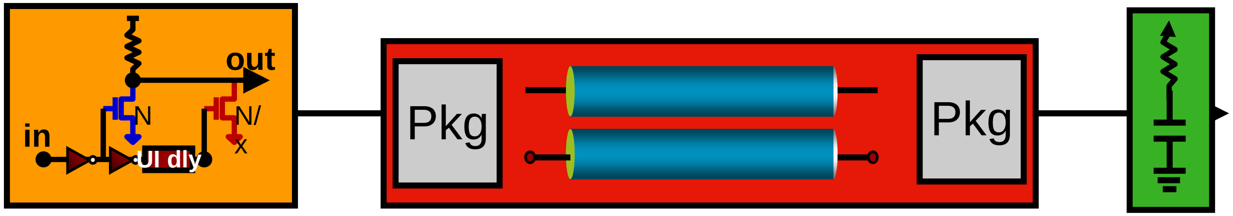
Example – query by Walter Katz – Sisoft

Channel Definition
A channel consists of a differential Tx driver, differential Rx receiver,
and a channel consisting of a 20 inch 100 Ohm differential transmission line.
The channel will be operating at 10Gbps (SymbolTime = 100ps).
The Tx driver is a 50 ohm driver, 1pF, 4 taps, rise and fall time=20ps, vol=0V, voh=1V.
The four taps are controlled by four “registers”,
each can have a value between 0 and 1,
but the sum of the absolute values must be <=1.
The Rx receiver’s electrical model is 1Meg ohm, 1pF.

Questions on this channel
How do I represent the electrical characteristics of the Rx and Tx model in IBIS?
What is the “Channel Characterization”?
What are the Tx and Rx API entry points?
What is the process for determining the 4 tap settings of the Tx?
For a given stimulus, how do I generate a “Voltage at Rx”?
Are time domain waveforms in the form of pairs of (Voltage,Time),
or are they at fixed time points (e.g. SymbolTime/10)?

Case 1: 4 tap Tx completely modeled in ibis

- 4 tap Tx completely described by IBIS
- User fixed tap coefficients
- Rx consists of IBIS load model + algorithm filter model

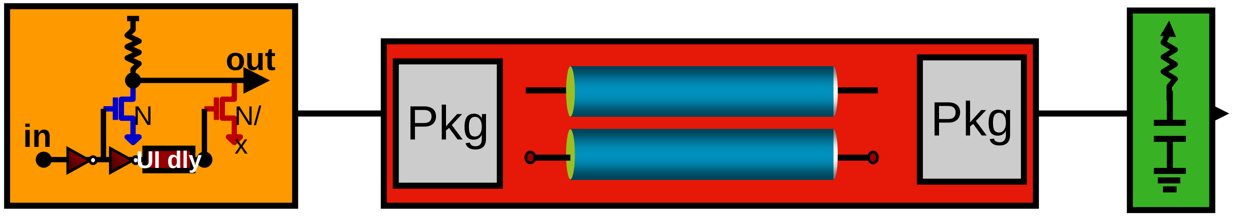


IBIS Tx (4 taps)
(VI/Spice/AMS)

Pkg + PCB + Connectors + PCB + Pkg

IBIS RX
(Load Model Only)

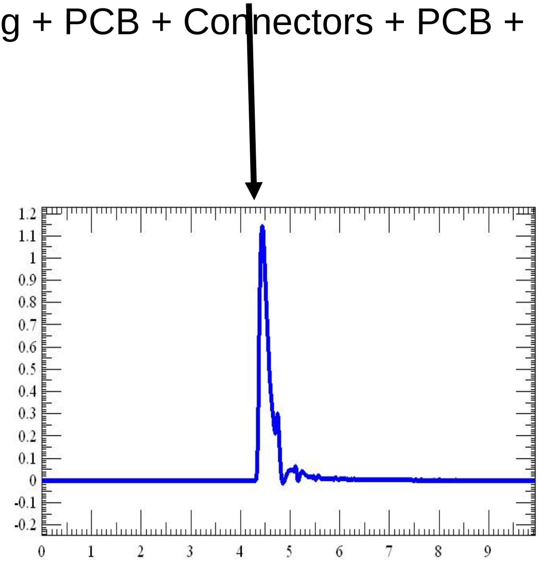
Case 1: AMI_Init – circuit characterization



IBIS Tx (4 taps)
(VI/Spice/AMS)

Pkg + PCB + Connectors + PCB + Pkg

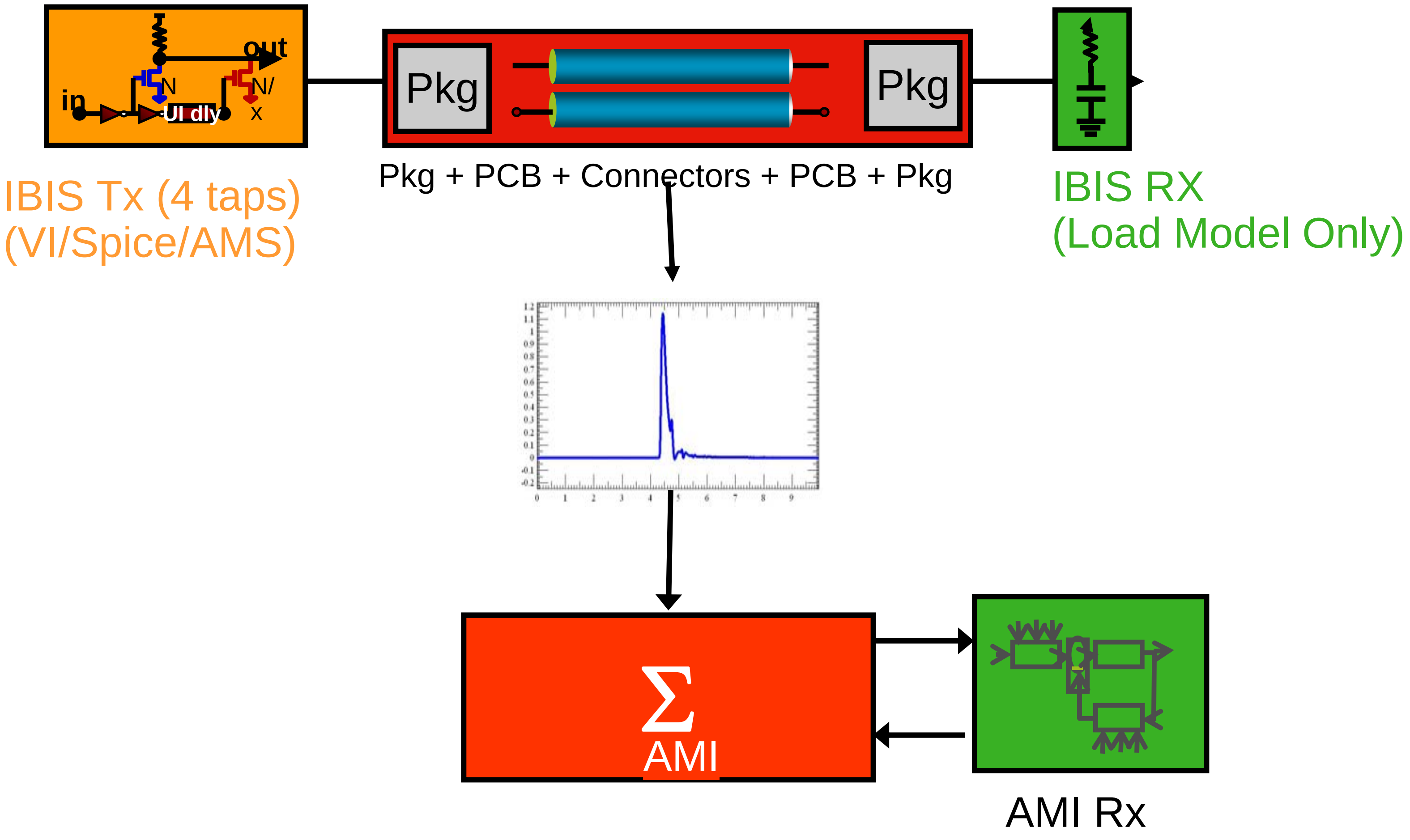
IBIS RX
(Load Model Only)



-Generate complete Tx-Rx path characterization using ibis

-Characterization will include Tx tap settings

Case 1: AMI_Init – AMI initialization



IBIS Tx (4 taps)
(VI/Spice/AMS)

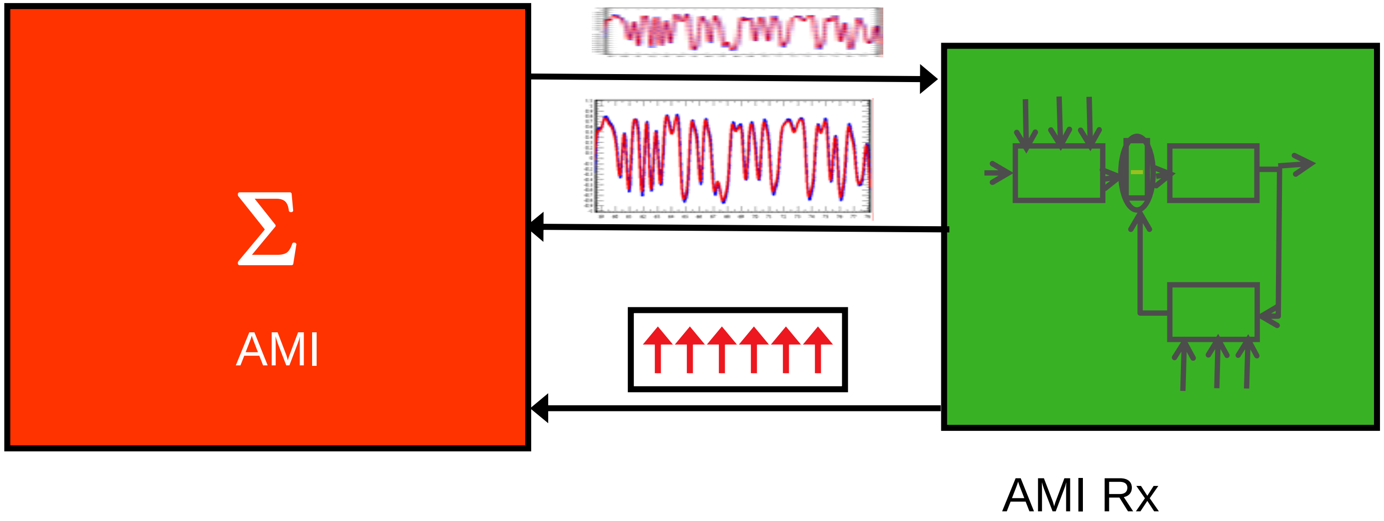
Pkg + PCB + Connectors + PCB + Pkg

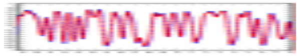
IBIS RX
(Load Model Only)

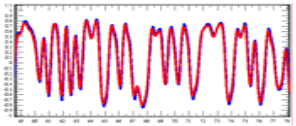
AMI Rx

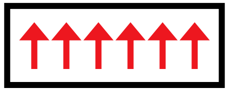
-AMI Rx may or may not modify the characterization

Case 1: AMI_GetWave



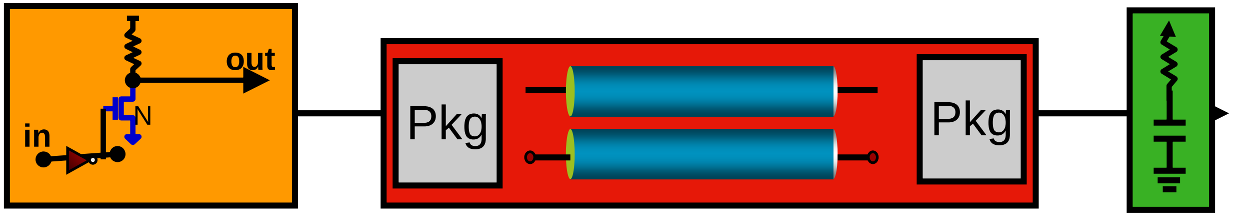

Wave form input


Filtered waveform from AMI Rx


Clocks from AMI Rx

Case 2: 4tap Tx and Rx are algorithm models

- IBIS contains only front end of Tx and Rx
- Both Tx and Rx have algorithm filter models
- Tap/filter coefficients may be automatic and/or user supplied as controlled my parameter inputs to the respective algorithm models

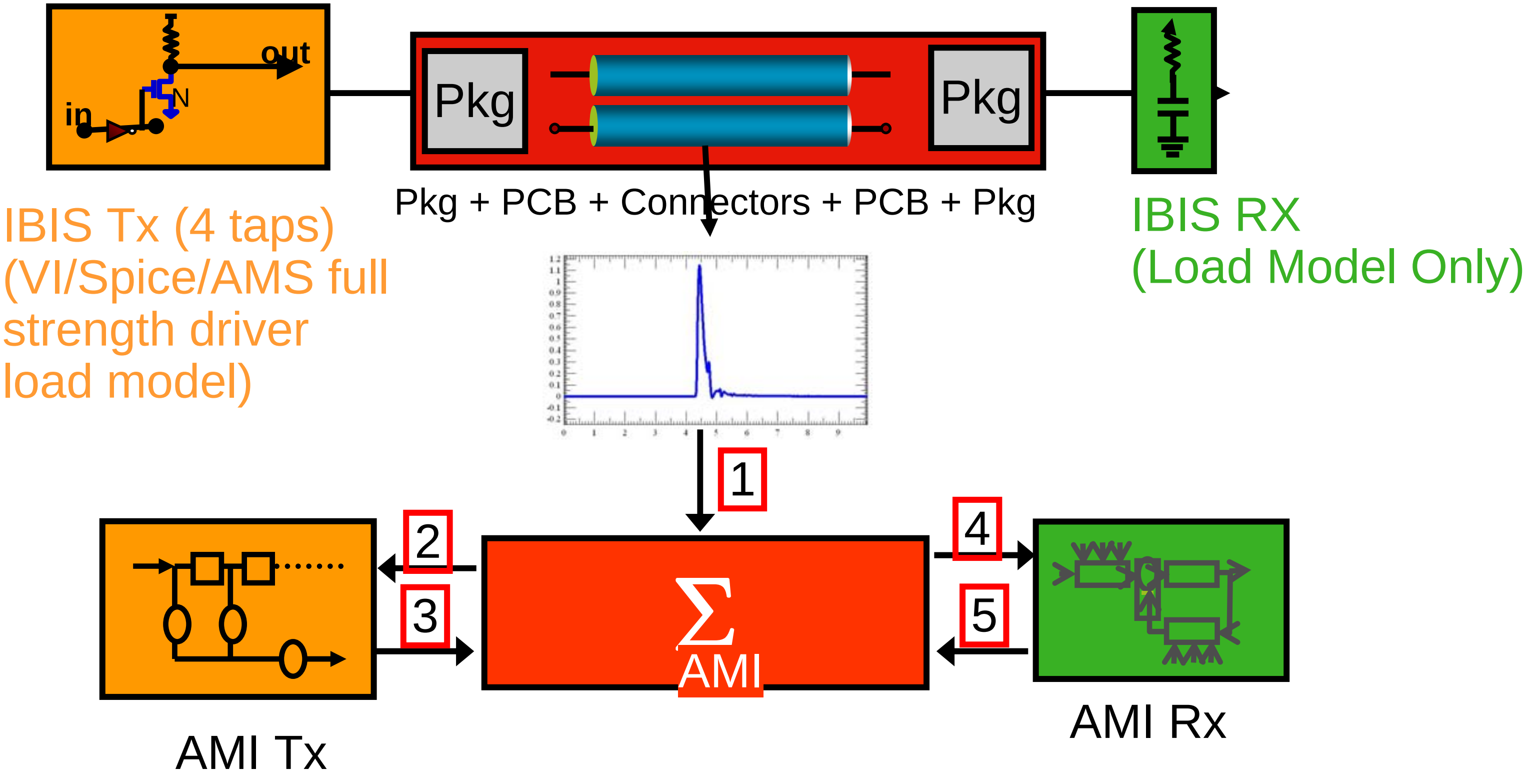


Pkg + PCB + Connectors + PCB + Pkg

IBIS Tx (4 taps) (VI/Spice/AMS full strength driver load model)

IBIS RX (Load Model Only)

Case 2: AMI_Init



Pkg + PCB + Connectors + PCB + Pkg

IBIS Tx (4 taps) (VI/Spice/AMS full strength driver load model)

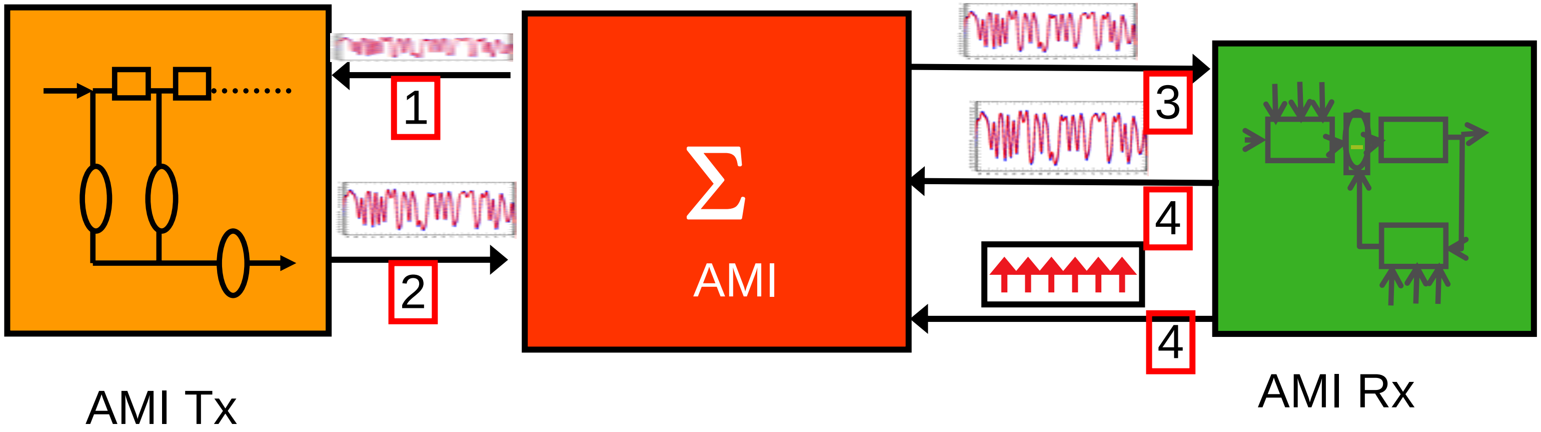
IBIS RX (Load Model Only)

1 Circuit characterization passed to EDA AMI platform

2 3 Characterization sent/received from AMI Tx

4 5 Characterization sent/received from AMI Rx

Case 2: AMI_getwave



1 EDA AMI platform sends wave form to Tx

2 Tx sends back modified waveform

3 EDA AMI platform sends wave form from step 2 to Rx

4 Rx sends back waveform and clock information

Special Case – only AMI_Init utilized

- It is possible that AMI models may use just AMI_Init to modify the characterization
- AMI_getwave my do nothing
- Only linear channel filtering is modeled and there will be no clock and data recovery information
- Such models may be used for channel compliance testing and early architectural exploration