IBIS-AMI Correlation and BIRD Update

SiSoft

IBIS-ATM Working Group

4/1/08

SiSoft™

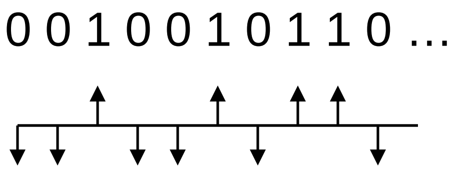
Signal Integrity Software, Inc.

Overview

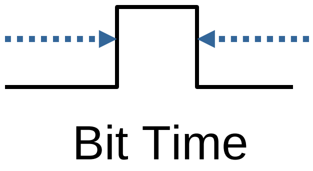
- DesignCon IBIS Summit presentation demonstrated interoperability and performance
  - SiSoft & Cadence IBIS-AMI models running together in SiSoft's test simulator
- Recent effort looked at simulation results
  - Run the same TX model in SiSoft & Cadence test simulators
  - Correlate both sets of results back to original SPICE simulations
- Identified areas of IBIS-AMI spec that needed clarification

Terminology

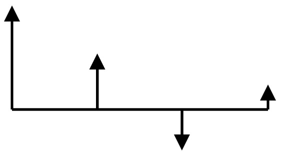
- Bit stream b(t)
  - Sum of delta functions



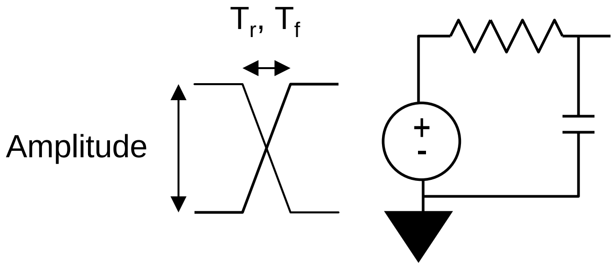
- Data symbol p(t)
  - Single bit width pulse



- Transmitter equalization $h_{TE}(t)$
  - Sum of weighted delta functions
    - Coefficients & delays

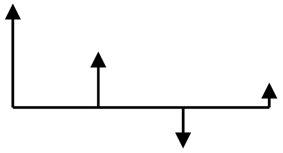


- Transmitter characteristic $h_{TX}(t)$
  - Rise/fall time
  - Voltage swing
  - Drive impedance
  - Capacitance

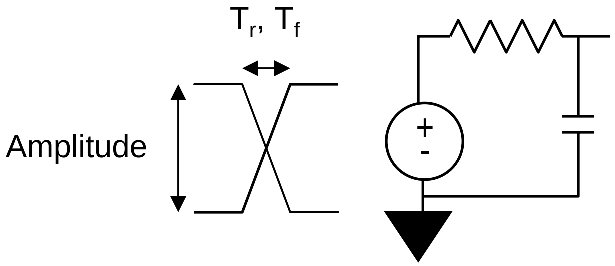


Terminology (Cont'd)

- Receiver equalization $h_{RE}(t)$
  - Sum of weighted delta functions
    - Coefficients & delays



- Receiver characteristic $h_{RX}(t)$
  - Rise/fall time
  - Voltage swing
  - Drive impedance
  - Capacitance



Channel Math

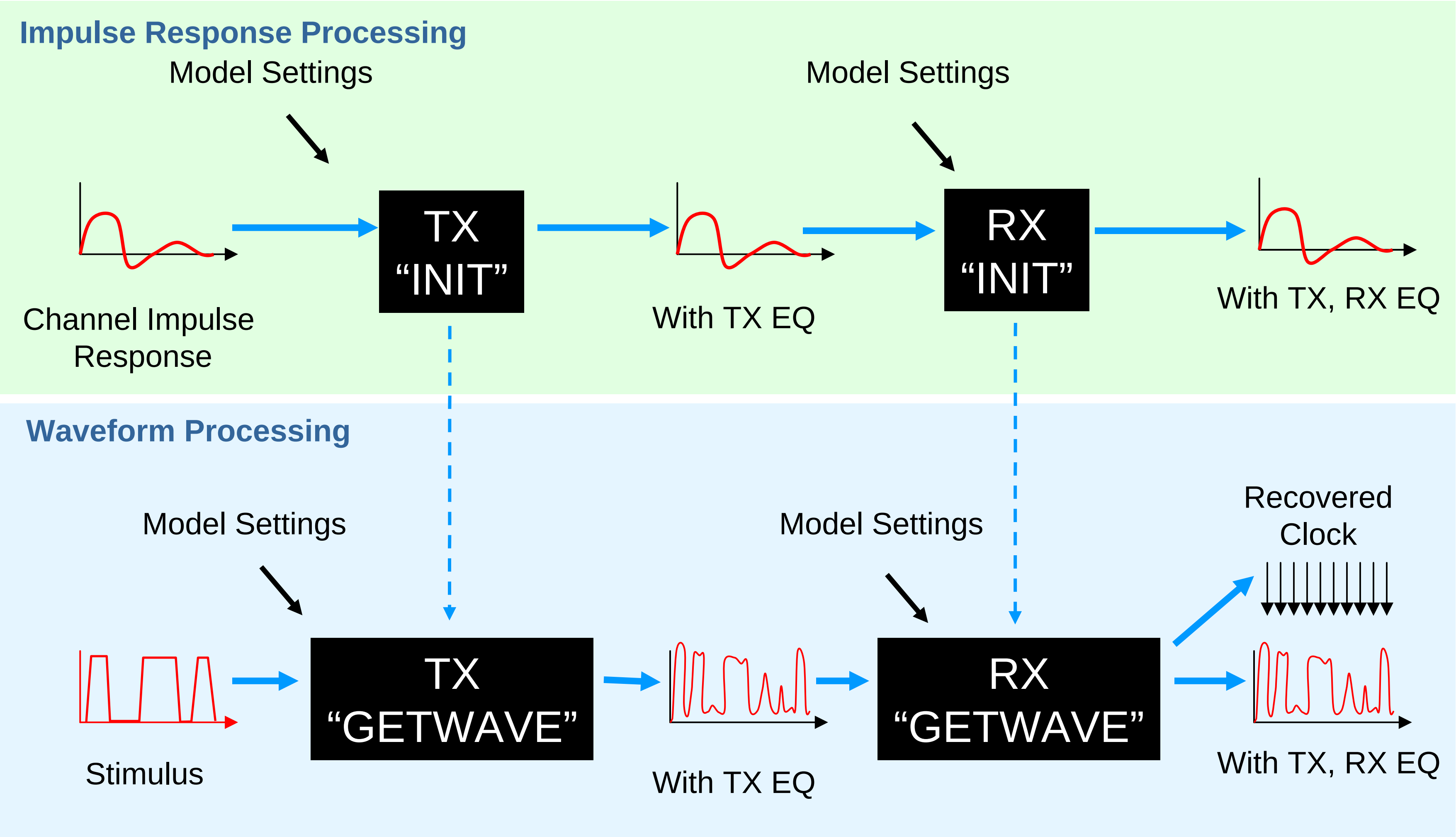
- Channel impulse response
  $h_{CR}(t) = h_{TX}(t) \otimes h(t) \otimes h_{RX}(t)$

- Equalized channel impulse response
  $= h_{TE}(t) \otimes h_{CR}(t) \otimes h_{RE}(t)$

- Waveform @ RX pad
  $= p(t) \otimes b(t) \otimes h_{TE}(t) \otimes h_{CR}(t)$

- Waveform @ RX sampler
  $= p(t) \otimes b(t) \otimes h_{TE}(t) \otimes h_{CR}(t) \otimes h_{RE}(t)$

IBIS-ATM Algorithmic Models



IBIS_AMI Specific Terminology

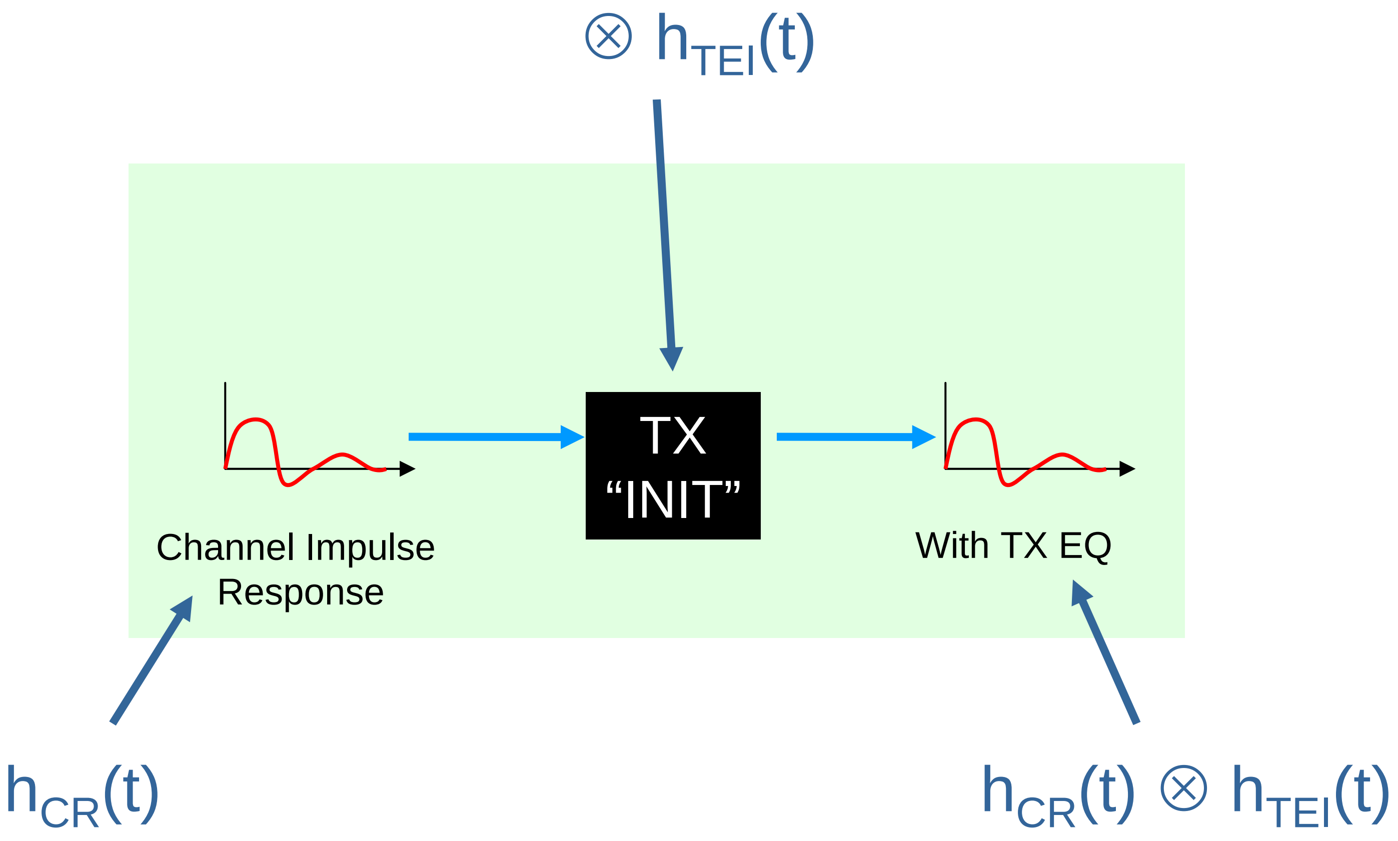
- TX Equalization

$h_{TEI}(t)$	Filtering applied by TX AMI_Init
$h_{TEG}(t)$	Filtering applied by TX AMI_Getwave

- RX Equalization

$h_{REI}(t)$	Filtering applied by RX AMI_Init
$h_{REG}(t)$	Filtering applied by RX AMI_Getwave

TX AMI_Init Processing



$\otimes\ h_{TEI}(t)$

Channel Impulse Response

TX "INIT"

With TX EQ

$h_{CR}(t)$

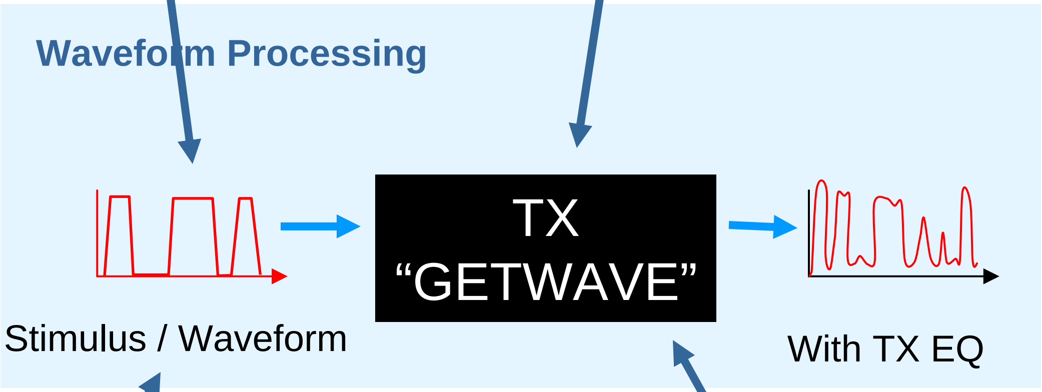
$h_{CR}(t) \otimes h_{TEI}(t)$

TX AMI_Getwave Processing

Note: for this specific model
$h_{TEG}(t) = h_{CR}(t) \otimes h_{TEI}(t)$

SiSoft

$b(t) \otimes p(t)$ $\otimes h_{TEG}(t)$



Waveform Processing

Stimulus / Waveform

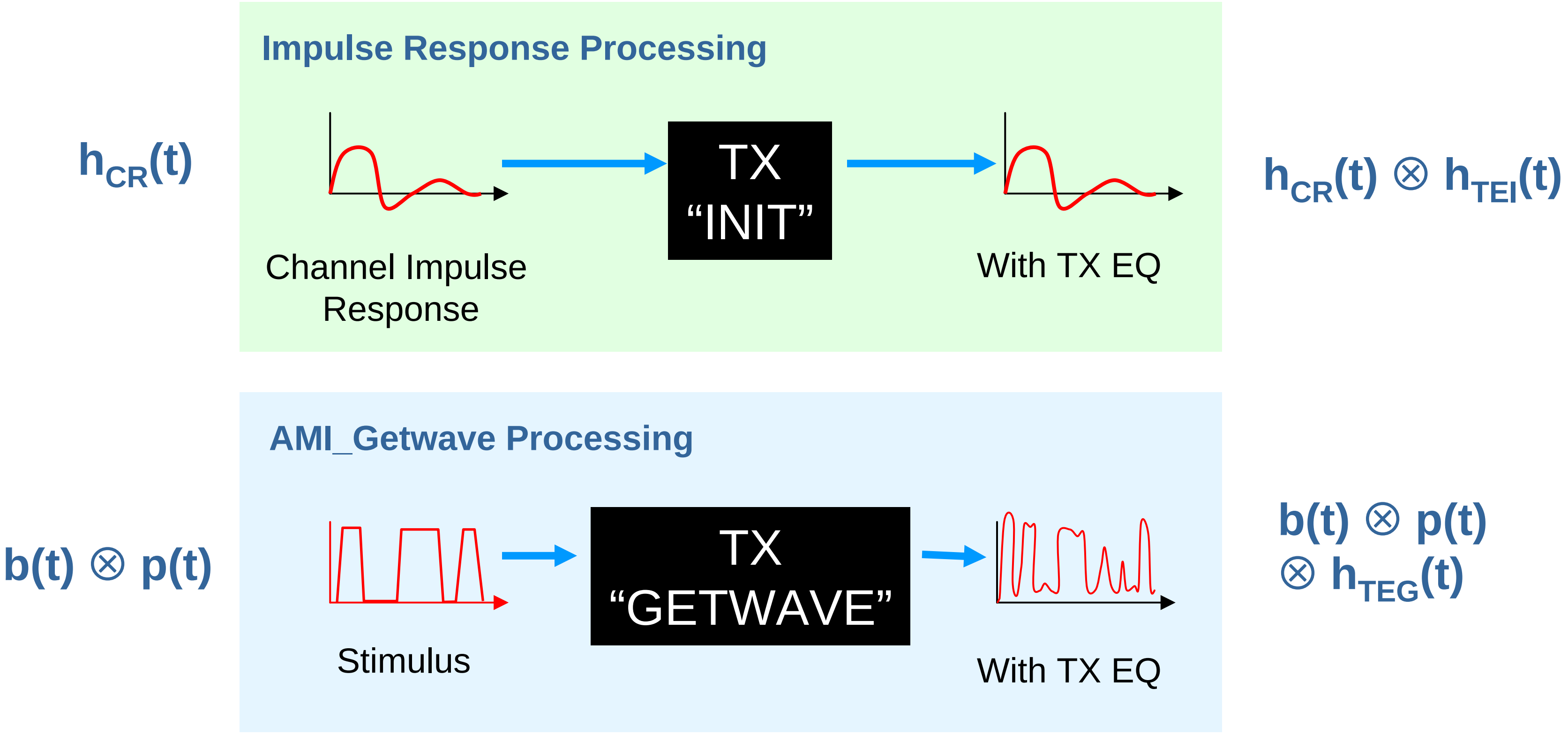
TX
“GETWAVE”

With TX EQ

Cadence

$b(t) \otimes p(t) \otimes h_{CR}(t) \otimes h_{TEI}(t)$ $\otimes h_{TEG}(t)$

Analysis Flow - SiSoft



Impulse Response Processing

$h_{CR}(t)$

Channel Impulse Response

TX "INIT"

With TX EQ

$h_{CR}(t) \otimes h_{TEI}(t)$

AMI_Getwave Processing

$b(t) \otimes p(t)$

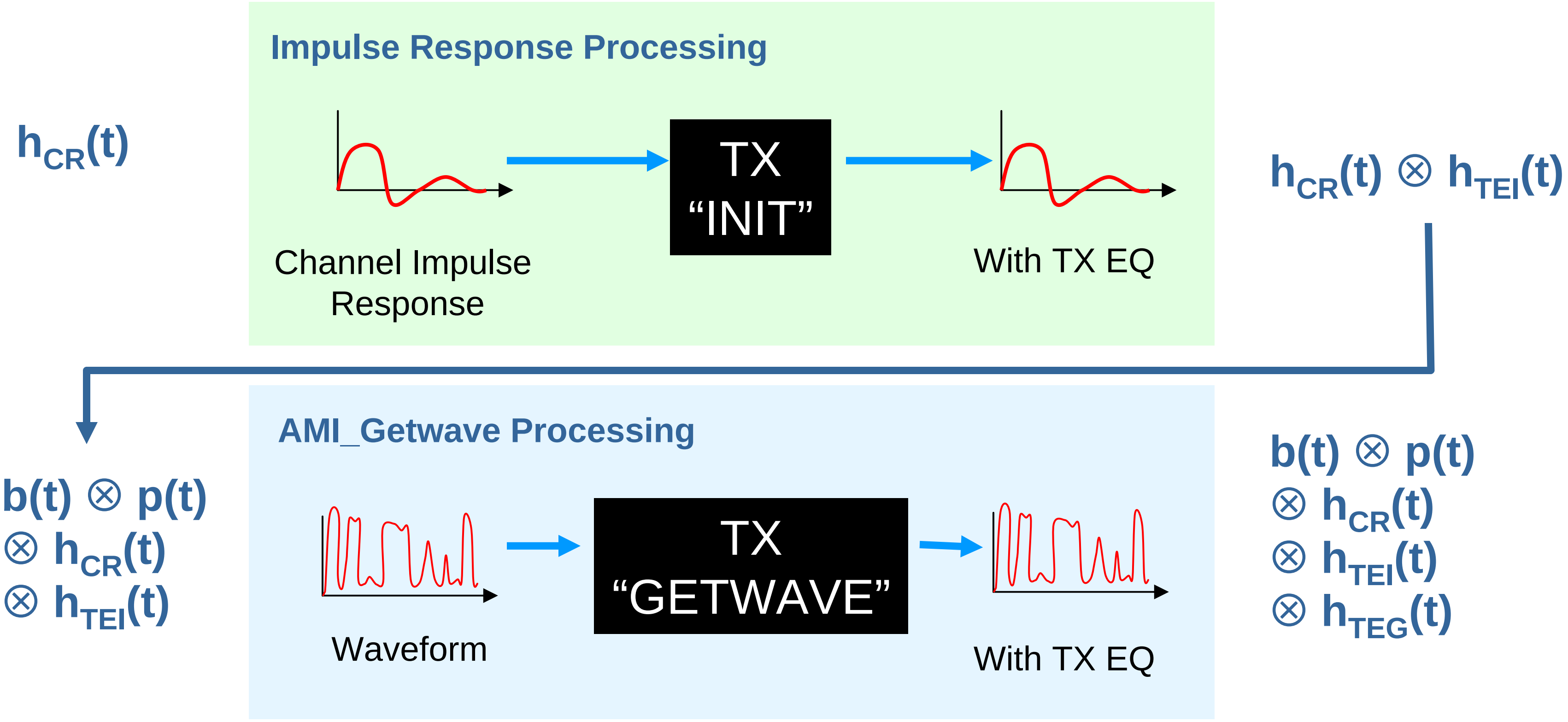
Stimulus

TX "GETWAVE"

With TX EQ

$b(t) \otimes p(t) \otimes h_{TEG}(t)$

Analysis Flow - Cadence



Impulse Response Processing

$h_{CR}(t)$

Channel Impulse Response

TX “INIT”

With TX EQ

$h_{CR}(t) \otimes h_{TEI}(t)$

AMI_Getwave Processing

$b(t) \otimes p(t)$
$\otimes\, h_{CR}(t)$
$\otimes\, h_{TEI}(t)$

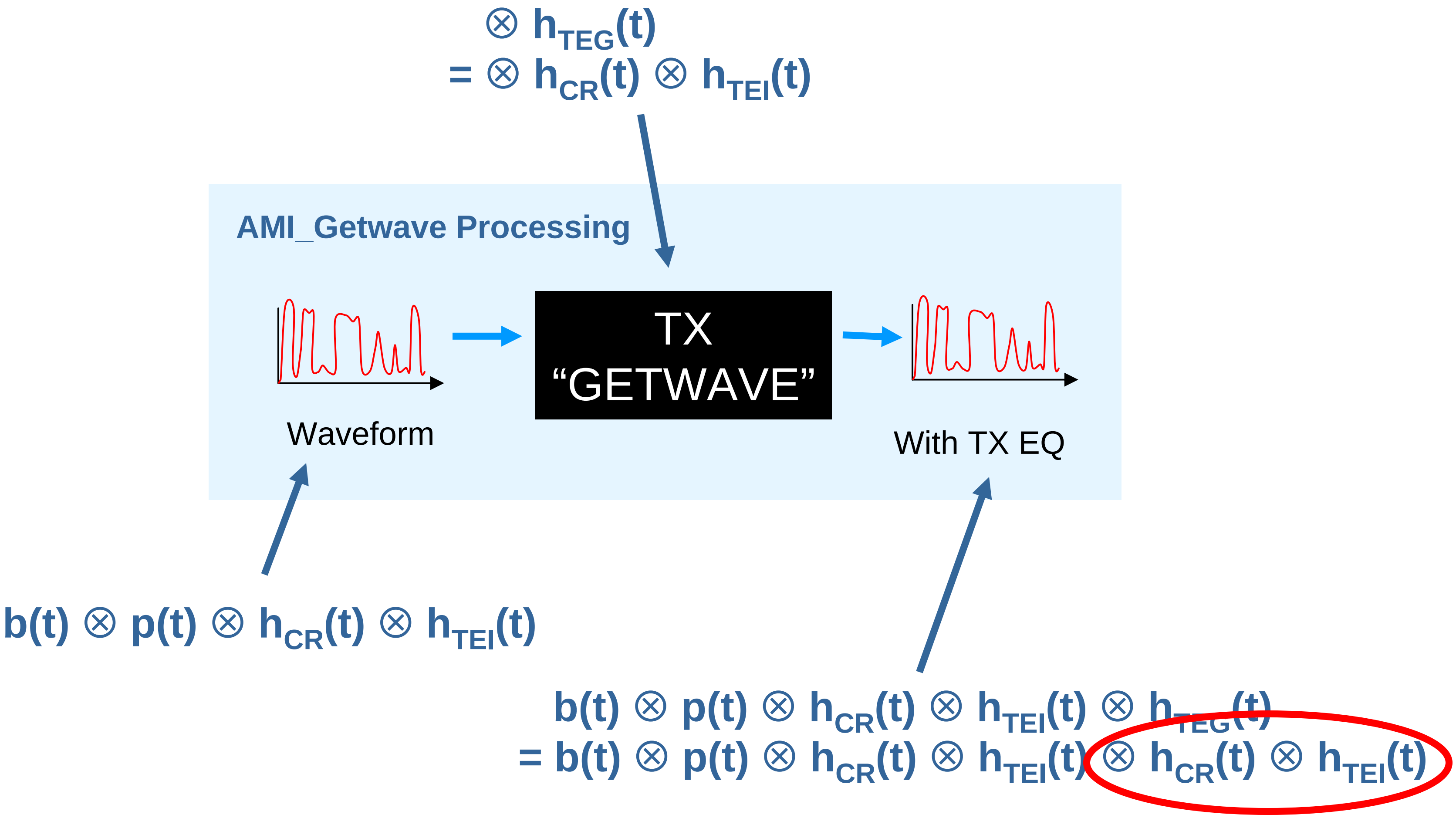
Waveform

TX “GETWAVE”

With TX EQ

$b(t) \otimes p(t)$
$\otimes\, h_{CR}(t)$
$\otimes\, h_{TEI}(t)$
$\otimes\, h_{TEG}(t)$

SiSoft TX Model in Cadence Tester



$\otimes\ h_{TEG}(t)$
$= \otimes\ h_{CR}(t) \otimes h_{TEI}(t)$

AMI_Getwave Processing

TX
“GETWAVE”

Waveform

With TX EQ

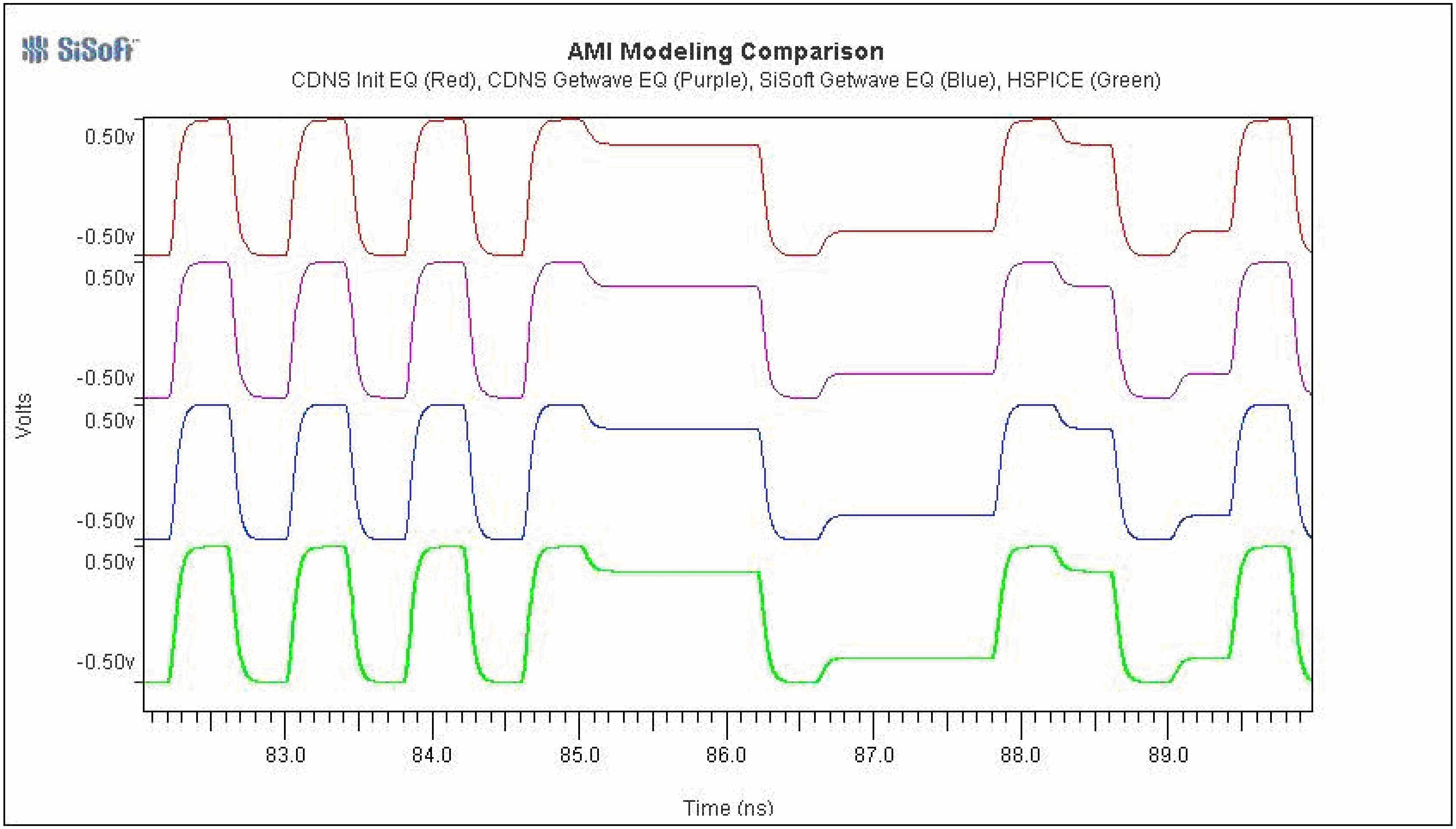
$b(t) \otimes p(t) \otimes h_{CR}(t) \otimes h_{TEI}(t)$

$b(t) \otimes p(t) \otimes h_{CR}(t) \otimes h_{TEI}(t) \otimes h_{TEG}(t)$
$= b(t) \otimes p(t) \otimes h_{CR}(t) \otimes h_{TEI}(t) \otimes h_{CR}(t) \otimes h_{TEI}(t)$

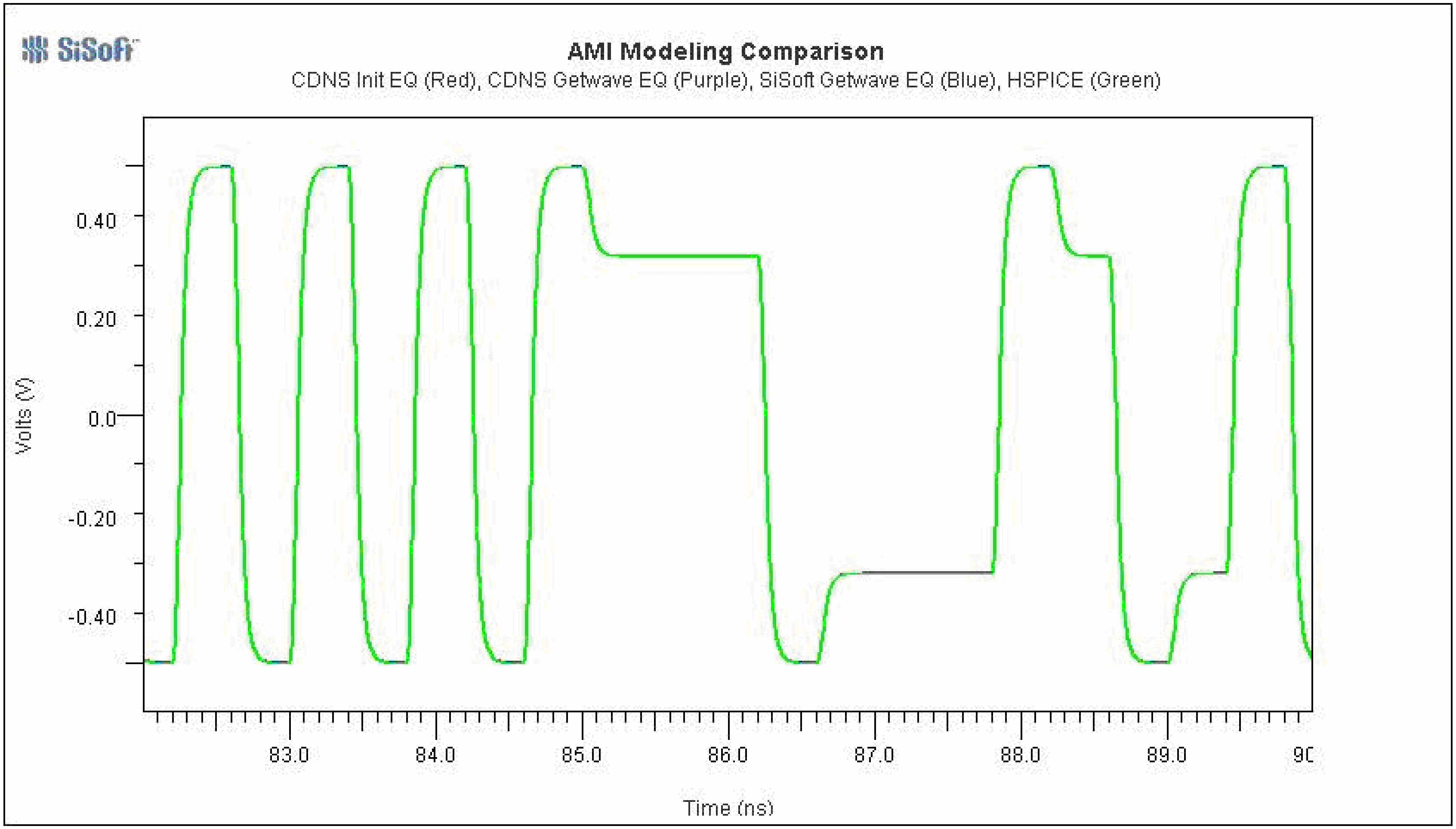
Simulation Results

- SiSoft TX model modified to work in current Cadence test environment
  - Allowed testing of proposed flows & correlation
- Flows tested
  - Cadence toolkit, EQ in AMI_Init
    - $h_{CR}(t) \otimes h_{TEI}(t) \otimes b(t) \otimes p(t)$
  - Cadence toolkit, EQ in AMI_Getwave
    - $b(t) \otimes p(t) \otimes h_{CR}(t) \otimes h_{TEG}(t)$
  - SiSoft toolkit, EQ in AMI_Getwave
    - $b(t) \otimes p(t) \otimes h_{CR}(t) \otimes h_{TEG}(t)$
- Compared to HSPICE baseline results

Simulation Results



Simulation Results



When Models Return Both AMI_Init and AMI_Getwave …

- What do the AMI_Init and AMI_Getwave calls in a model represent?
  - Different models for the entire device's behavior?
    - (e.g. an approximate vs. detailed model)
    - Implies that AMI_Init and AMI_Getwave results are independent
  - Models for different stages of the devices behavior?
    - (e.g. front end filtering in AMI_Init, DFE Equalization in AMI_Getwave)
    - Impiles that AMI_Init and AMI_Getwave calls must be chained together to model the full device behavior

Proposal

- Drop “digital input” to TX AMI_Getwave call
  - Purpose of TX AMI_Getwave call is to apply $h_{TEG}(t)$, which should only represent the device’s filtering behavior
- Allow AMI model to specify how AMI_Init and AMI_Getwave outputs are handled
  - New Reserved_Parameter “Use_Init_Output”
    - True: chain the model calls together
    - False: treat model calls as independent

Updated IBIS-AMI Text

```
Current Version
---------------
|
| 3.2.2.1 wave
| ============
|
| A vector of a time domain waveform, sampled uniformly at an interval
| specified by the 'sample_interval' specified during the init call.  The
| wave is both input and output.  The EDA platform provides the wave.  The
| algorithmic model is expected to modify the waveform in place.
|
| Depending on the EDA platform and the analysis/simulation method chosen,
| the input waveform could include many components.  For example, the input
| waveform could include:
|

--------------------------------------------------------------------------------
New Version
---------------
|
| 3.2.2.1 wave
| ============
|
| A vector of a time domain waveform, sampled uniformly at an interval
| specified by the 'sample_interval' specified during the init call. The
| wave is both input and output. The EDA platform provides the wave.
| The algorithmic model is expected to modify the waveform in place by
| applying a filtering behavior, for example, an equalization function,
| being modeled in the AMI_Getwave call.
|
| Depending on the EDA platform and the analysis/simulation method chosen,
| the input waveform could include many components.  For example, the input
| waveform could include:
|
==========================================================
```

```
Current Version
---------------
|        Reserved Parameters:
|
|        Init_Returns_Impulse, GetWave_Exists, Max_Init_Aggressors and
|        Ignore_Bits
--------------------------------------------------------------------------------
New Version
---------------
|        Reserved Parameters:
|
|        Init_Returns_Impulse, Use_Init_Output, GetWave_Exists,
|        Max_Init_Aggressors and Ignore_Bits
================================================================================
New Addition
------------
Add after Init_Returns_Impulse

|                                              Use_Init_Output:
|
|                                              Use_Init_Output is of usage Info and type Boolean.  When Use_Init_Output
|                                              is set to "True", the effects of the AMI_Init and AMI_Getwave calls are
|            chained together by convolving the impulse response returned by AMI_Init
|            with the input waveform, which is then presented to the AMI_Getwave call.
|
|            If the Reserved Parameter, Use_Init_Output, is set to "False", EDA tools will
|            use the original (unfiltered) impulse response of the channel.
|            The algorithmic model is expected to modify the waveform in place.
|
|                                              The default value for this parameter is "True", instructing the EDA tool
|                                              to use the output impulse response from the AMI_Init function
|                                              when creating the input wave for the AMI_Getwave function.
|
|            If Use_Init_Output is False, GetWave_Exists must be True.

================================================================================
Minor changes WRT missing words etc.

Current Version
---------------
| 3.1.2.6 AMI_parameters (_in and _out)
| ====================================
|
| Memory for AMI_parameters_in is allocated and de-allocate by the EDA.  The
| memory pointed to by AMI_parameters_out is allocated and by the model.

New Version
-----------
| 3.1.2.6 AMI_parameters (_in and _out)
| ====================================
|
| Memory for AMI_parameters_in is allocated and de-allocated by the EDA platform. The
| memory pointed to by AMI_parameters_out is allocated and de-allocated by the model.
```