IBIS-AMI: Assumptions, Terminology & Analytical Flows

Walter Katz (wkatz@sisoft.com)
Mike Steinberger (msteinb@sisoft.com)
Todd Westerhoff (twesterh@sisoft.com)
SiSoft

DesignCon IBIS Summit
February 3, 2017
Santa Clara, California

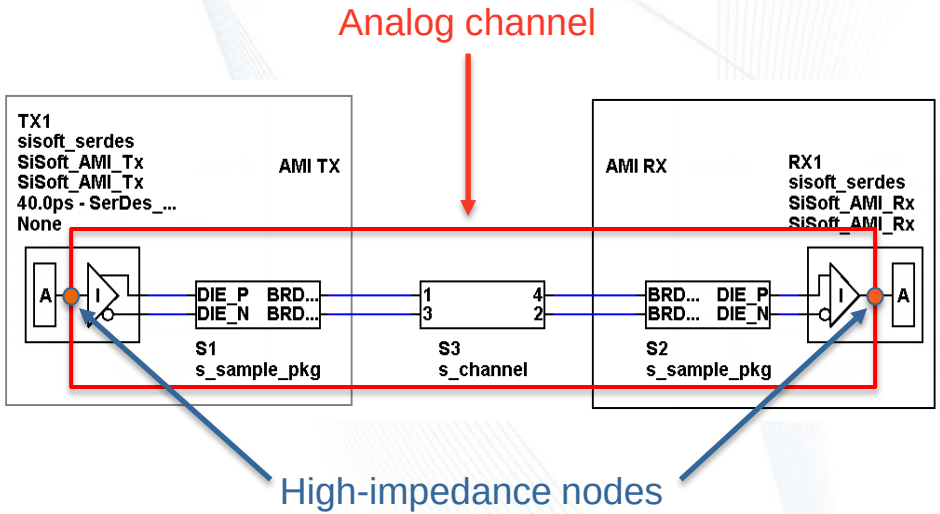
SiSoft™
We Are Signal Integrity

Agenda

- IBIS-AMI Assumptions & Terminology
- IBIS-AMI Model Components
- Analysis Stages & Simulation Types
- Algorithmic Model Types
- Static and Dynamic Equalization
- IBIS-AMI Simulation Cases & Reference Flows
- Interpreting Simulation Results
- Recovered Clock Processing
- Jitter
- Summary

IBIS-AMI Assumptions

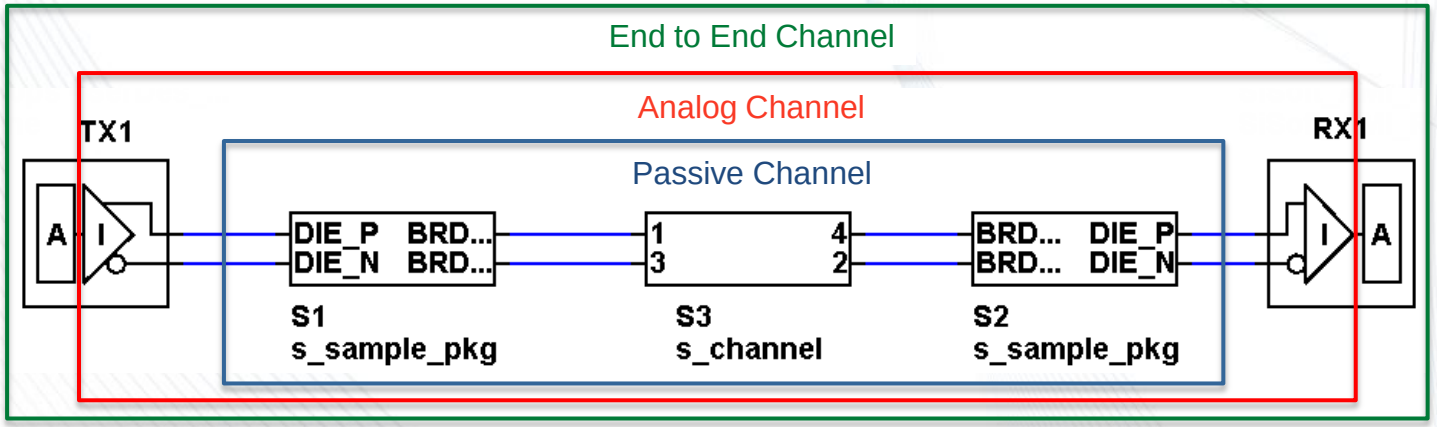
- SerDes channels can be broken into two parts for analysis:
  - Analog (electrical) and Algorithmic
- TX analog input & RX analog output have “high-impedance” connection to analog channel
- Analog channel can be considered linear and time-invariant (LTI)



http://ibis.org/ver6.1/ver6_1.pdf, page 170

SiSoft™ We Are Signal Integrity

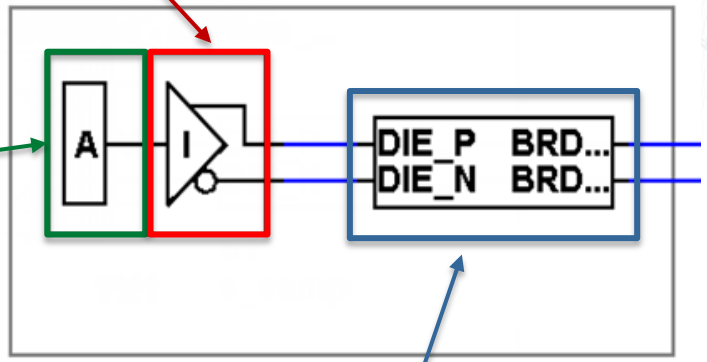
IBIS-AMI Channel Terminology



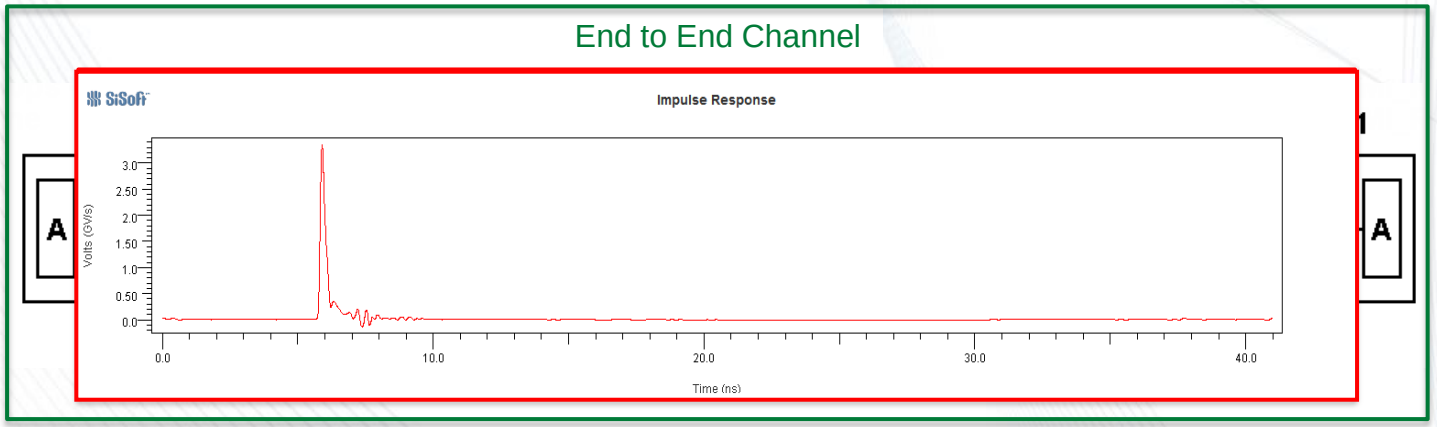
- Circuit simulation techniques are used for the analog channel
- Signal processing techniques are used for the end to end channel

IBIS-AMI Model Components

- Analog model
  - [Model] keyword in .ibs file
  - Tabular V/I & V/T data
  - Assumed to be LTI
- Algorithmic model
  - [Algorithmic Model] keyword in .ibs file
  - .ami file describes capabilities & inputs
- Package model
  - Can be described in .ibs file
  - Often supplied separately as .sNp file



Analysis Stages

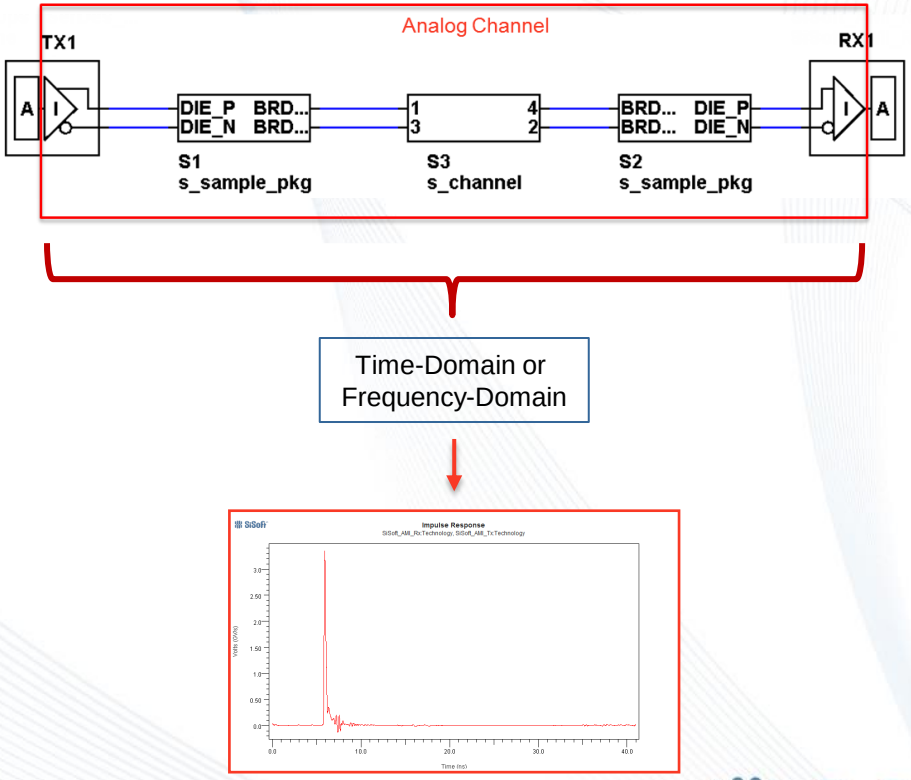


- Network Characterization (Circuit Simulation)
- Channel Simulation (Signal Processing)
  - Statistical Simulation
  - Time-Domain Simulation

SiSoft™ We Are Signal Integrity

Network Characterization

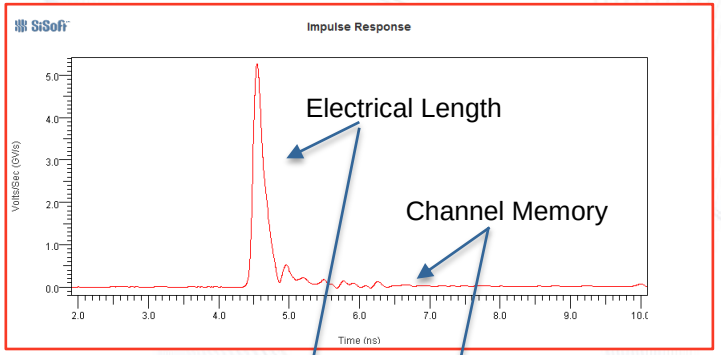
- Inputs:
  - Analog sections of .ibs file
  - Passive topology elements
- Analysis Method:
  - Not specified by IBIS
  - Time-domain (step response)
  - Frequency-domain (transfer function)
- Outputs:
  - Impulse response
  - Fixed time steps
  - Long enough for signal to settle



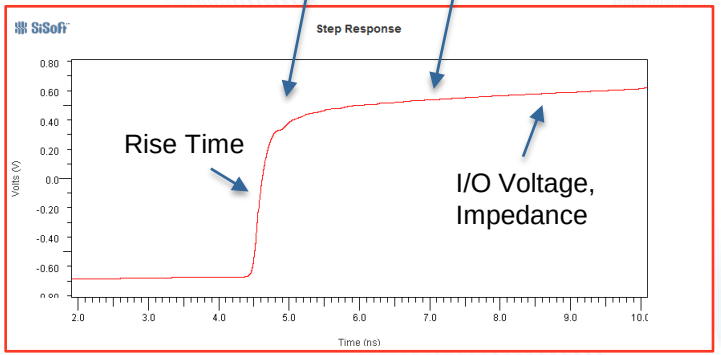
SiSoft™ We Are Signal Integrity

About the Impulse Response …

- Only the impulse response goes forward from Network Characterization
  - If the impulse response is bad, running Channel Simulation ***is a waste of time***
- Verify impulse response before running channel analysis
- Step response is easier to interpret
  - Voltage levels
  - Rise time
  - Network delay
  - Reflections and settling behavior
- Remember – impulse response does not include TX or RX equalization



Impulse Response

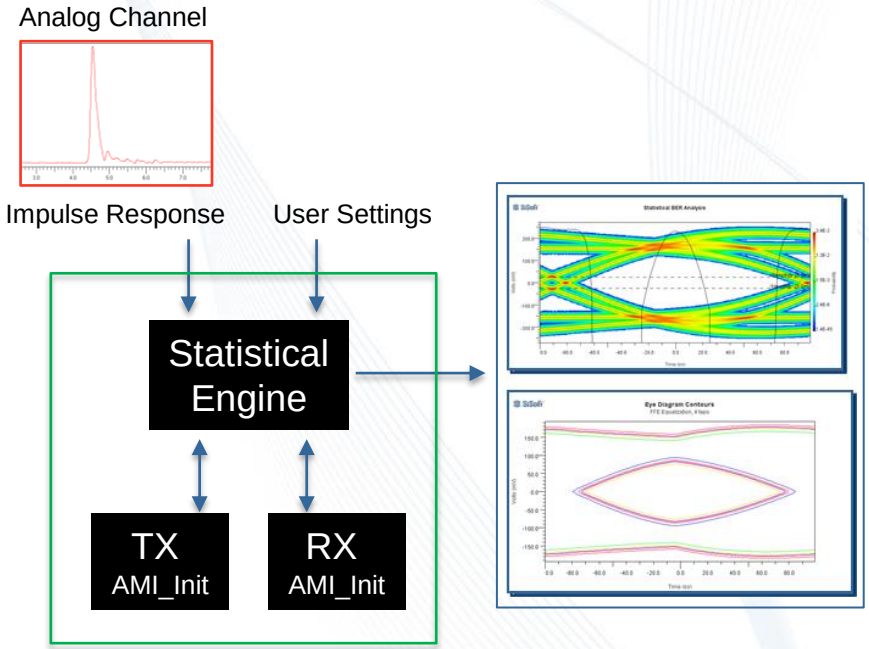


Step Response

SiSoft™ We Are Signal Integrity

Statistical Simulation

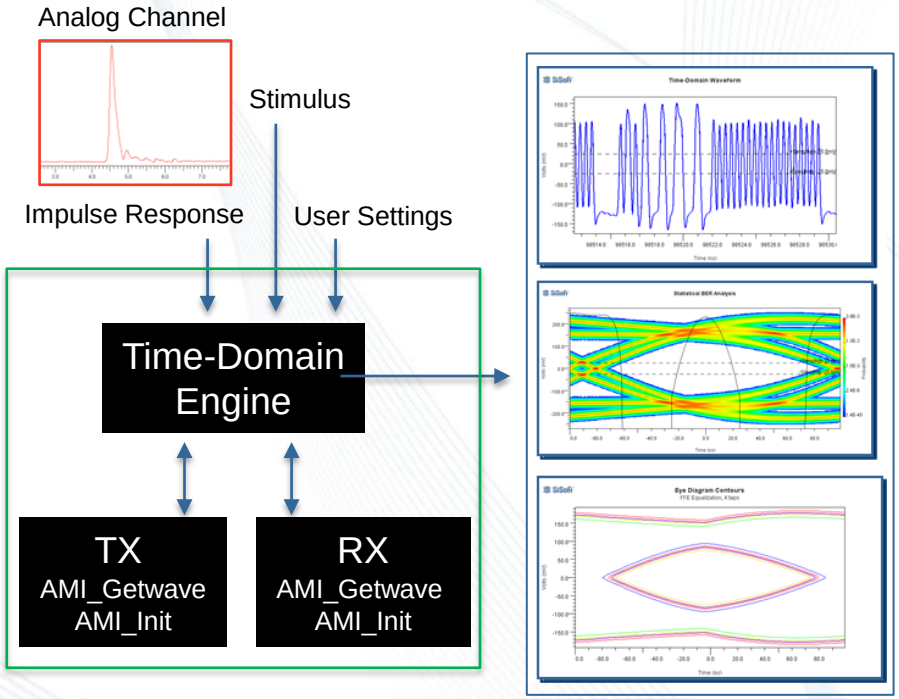
- Inputs:
  - Analog channel impulse response
  - User selections for model parameters
  - Algorithmic models
    (AMI_Init / impulse response processing)
- Analysis Method:
  - Convolution engine (pulse response)
- Outputs:
  - Not specified by IBIS
  - Statistical eye diagrams
  - Eye height / width measurements
  - Eye contours @ probabilities
  - Equalized / unequalized responses



SiSoft™ We Are Signal Integrity

Time-Domain Simulation

- Inputs:
  - Impulse responses from prior steps
  - User-defined input stimulus
  - Algorithmic models (AMI_Getwave / waveform processing)
- Analysis Method:
  - Waveform processing & convolution
- Outputs:
  - Not specified by IBIS
  - Persistent eye diagrams
  - Eye height / width measurements
  - Eye contours @ probabilities
  - Equalized / unequalized responses



SiSoft™ We Are Signal Integrity

Algorithmic Model Types

3 types of algorithmic models exist:

1. Impulse response (Init) only
   - Init_Returns_Impulse = TRUE
   - Getwave_Exists = FALSE
2. Waveform (Getwave) only
   - Init_Returns_Impulse = FALSE
   - Getwave_Exists = TRUE
3. Dual
   - Init_Returns_Impulse = TRUE
   - Getwave_Exists = TRUE

```
(IBIS_AMI_Tx
   (Description "Generic transmitter model published by SiSoft")

   (Reserved_Parameters
      (Ignore_Bits (Usage Info) (Type Integer) (Default 4)
         (Description "Ignore four bits to fill up tapped delay line."))
      (Max_Init_Aggressors (Usage Info) (Type Integer) (Default 25)
         (Description "Number of aggressors is actually unlimited."))
      (Init_Returns_Impulse (Usage Info) (Type Boolean) (Default True)
         (Description "Both impulse and parameters_out returned."))
      (GetWave_Exists (Usage Info) (Type Boolean) (Default True)
         (Description "GetWave is well and truly provided in the module."))
) | End Reserved_Parameters

   (Model_Specific
      (tap_filter (Description "Array of transmit de-emphasis tap weights")
         (-1 (Usage InOut)(Format Range 0.0 -1.0 1.0)(Type Tap)(Default 0)
               (Description "Pre-cursor tap weight"))
         (0  (Usage InOut)(Format Range 1.0 -1.0 1.0)(Type Tap)(Default 1)
               (Description "Main tap weight"))
         (1  (Usage InOut)(Format Range 0.0 -1.0 1.0)(Type Tap)(Default 0)
               (Description "First post-cursor tap weight"))
         (2  (Usage InOut)(Format Range 0.0 -1.0 1.0)(Type Tap)(Default 0)
               (Description "Second post-cursor tap weight"))
      ) | End tap_filter
      (tx_swing (Usage In)(Format Range 0.8 0.3 1.0)(Type Float)(Default 0.8)
               (Description "Peak differential output voltage"))
   ) | End Model_Specific

) | End IBIS_AMI_Tx
```

Reserved_Parameters
Provide information on model function and control analysis flow
[Info for the simulator]

Model_Specific
Parameters used by this specific model, legal and default values
[Info for providing inputs to the model]

.AMI file

The 9 AMI Simulation Cases

- The method an AMI simulator uses to create Time-Domain results is based on the types of TX and RX algorithmic models involved.

{ Init-Only, Getwave-Only, Dual } x { Init-Only, Getwave-Only, Dual } = 9 Cases

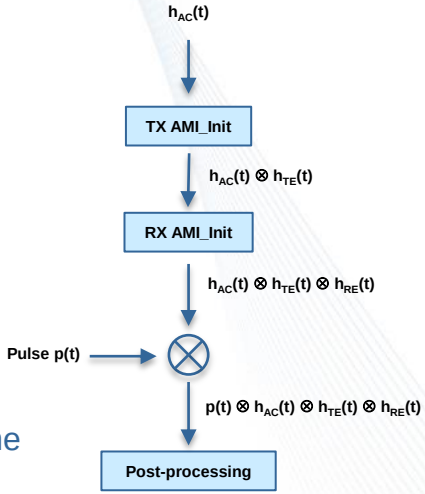
Case #	TX			RX		
	Getwave Exists	**Init_Returns_Impulse**	**Meaning**	**Getwave Exists**	**Init_Returns_Impulse**	**Meaning**
1	FALSE	TRUE	Init Model Only	FALSE	TRUE	Init Model Only
2	FALSE	TRUE	Init Model Only	TRUE	FALSE	Getwave Model Only
3	FALSE	TRUE	Init Model Only	TRUE	TRUE	Dual Model
4	TRUE	FALSE	Getwave Model Only	FALSE	TRUE	Init Model Only
5	TRUE	FALSE	Getwave Model Only	TRUE	FALSE	Getwave Model Only
6	TRUE	FALSE	Getwave Model Only	TRUE	TRUE	Dual Model
7	TRUE	TRUE	Dual Model	FALSE	TRUE	Init Model Only
8	TRUE	TRUE	Dual Model	TRUE	FALSE	Getwave Model Only
9	TRUE	TRUE	Dual Model	TRUE	TRUE	Dual Model

IBIS-AMI Simulation Terminology

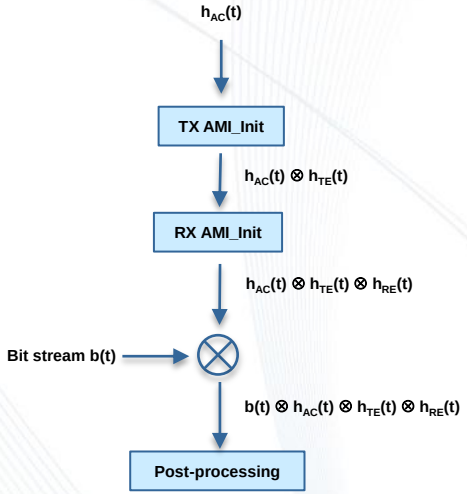
- $h_{AC}(t)$ – Analog channel impulse response
- $p(t)$ – Unit pulse at target data rate
- $b(t)$ – Data bit stream suitable for convolution processing
- $h_{TE}(t)$ – Impulse response of TX AMI_Init equalization
- $h_{RE}(t)$ – Impulse response of RX AMI_Init equalization
- $g_{TE}[x(t)]$ – Waveform output of TX Getwave processing
- $g_{RE}[x(t)]$ – Waveform output of RX Getwave processing

Case 1: TX: Init-only ➔ RX: Init-only

- Statistical Simulation
  - Includes TX & RX equalization
- Time-Domain simulation
  - Neither model supports waveform processing
  - Static TX & static RX equalization
  - Any clock recovery is performed by simulator
- Implications
  - Statistical & Time-Domain equalization is the same
  - Little benefit in running Time-Domain, unless
    - Low density encoding (e.g. 8b10b) is used
    - Need to isolate specific stimulus pattern
  - Time-Domain simulations (if run) can be fairly short, as there are no RX control loops to settle



Statistical Flow
Static TX EQ, Static RX EQ

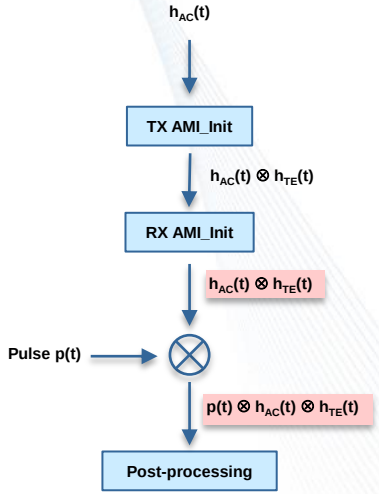


Time-Domain Flow
Static TX EQ, Static RX EQ

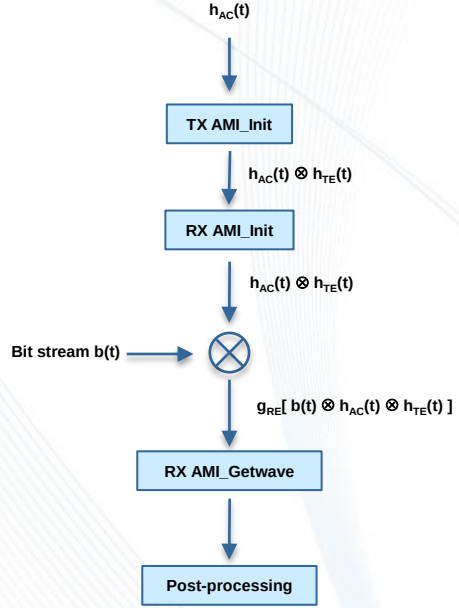
SiSoft™ We Are Signal Integrity

Case 2: TX: Init-only ➔ RX: Getwave-only

- Statistical Simulation
  - Includes TX equalization only
- Time-Domain simulation
  - Static TX & Dynamic RX equalization
  - Clock recovery (usually) performed by RX model
- Implications
  - Statistical results are missing RX equalization
    - Consider results as “signal at RX die pad”
  - Statistical & Time-Domain TX behavior is the same
  - Time-domain simulations must allow RX feedback loops to settle before data accumulation begins



Statistical Flow
Static TX EQ, **No RX EQ**

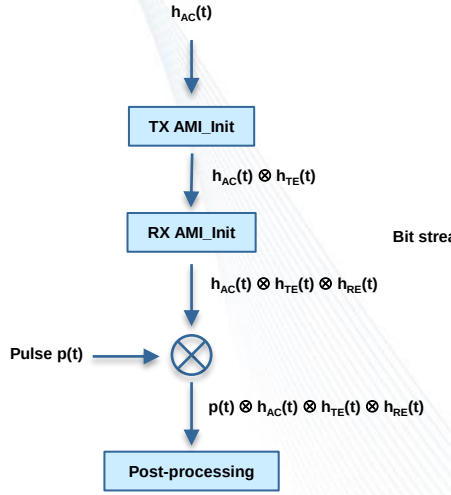


Time-Domain Flow
Static TX EQ, Dynamic RX EQ

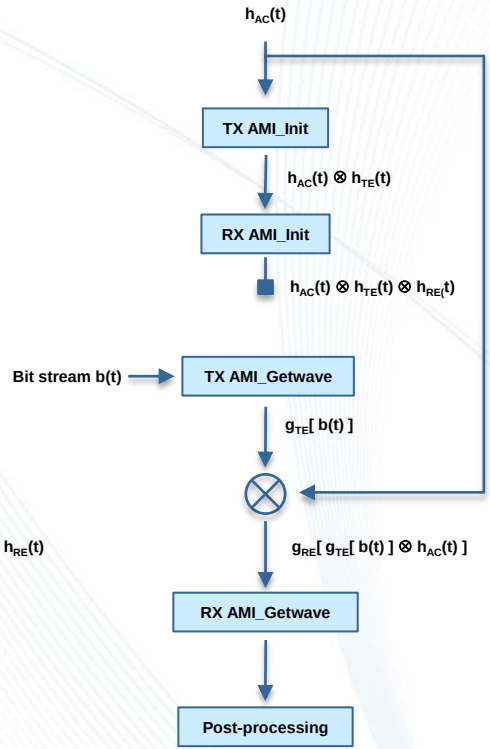
SiSoft™
We Are Signal Integrity

Case 9: TX: Dual ➔ RX: Dual

- Statistical Simulation
  - Includes TX & RX equalization
- Time-Domain simulation
  - Dynamic TX & Dynamic RX equalization
  - Clock recovery (usually) performed by RX model
- Implications
  - Best possible case: fully supports Statistical & Time-Domain simulation
  - Time-domain simulations must allow RX feedback loops to settle before data accumulation begins
  - Statistical simulations can be used for exploration, Time-Domain simulations for validation
  - Should compare Statistical & Time-Domain results to determine how well Statistical results predict their Time-Domain counterparts



Statistical Flow
Static TX EQ, Static RX EQ



Time-Domain Flow
Dynamic TX EQ, Dynamic RX EQ

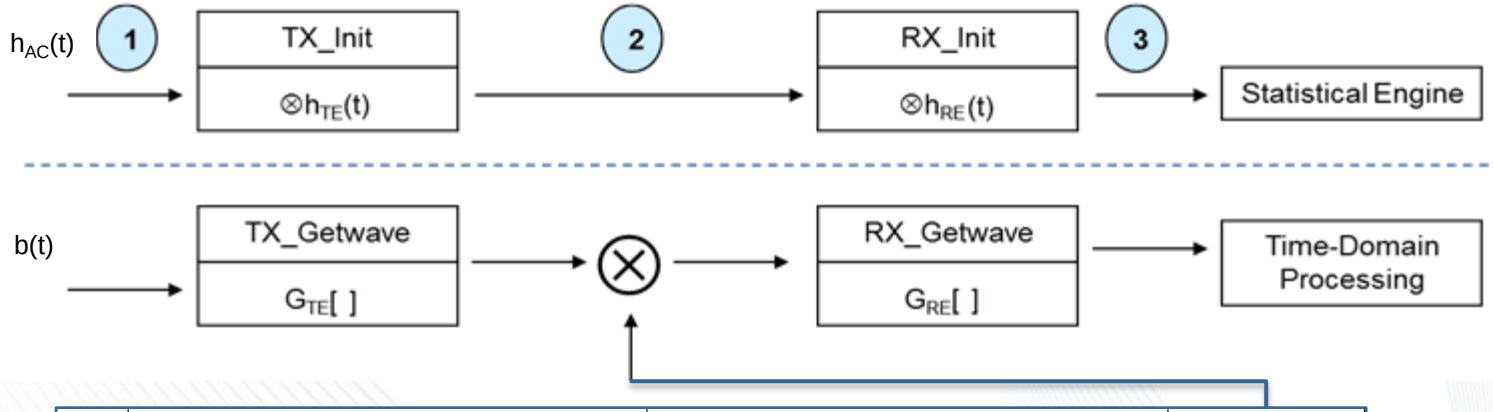
SiSoft™
We Are Signal Integrity

AMI Equations for 9 TX/RX Cases

Case #	Tx Type*	Rx Type*	Statistical	Time Domain
1	FT	FT	hAC(t)⊗hTE(t)⊗hRE(t)	hAC(t)⊗hTE(t)⊗hRE(t)⊗x(t)
2	FT	TF	hAC(t)⊗hTE(t)	gREG[hAC(t)⊗hTE(t)⊗x(t)]
3	FT	TT	hAC(t)⊗hTE(t)⊗hRE(t)	gREG[hAC(t)⊗hTE(t)⊗x(t)]
4	TF	FT	hAC(t)⊗hRE(t)	hAC(t)⊗hRE(t)⊗gTE[x(t)]
5	TF	TF	hAC(t)	gREG[hAC(t)⊗gTE[x(t)]]
6	TF	TT	hAC(t)⊗hRE(t)	gREG[hAC(t)⊗gTE[x(t)]]
7	TT	FT	hAC(t)⊗hTE(t)⊗hRE(t)	hAC(t)⊗hRE(t)⊗gTE[x(t)]
8	TT	TF	hAC(t)⊗hTE(t)	gREG[hAC(t)⊗gTE[x(t)]]
9	TT	TT	hAC(t)⊗hTE(t)⊗hRE(t)	gREG[hAC(t)⊗gTE[x(t)]]
* = Getwave_Exists, Init_Returns_Impulse				

- Allows us to efficiently & unambiguously define what simulation results should be

IBIS-AMI Reference Flow



Case #	TX			RX			Convolution Input
	Getwave Exists	**Init_Returns_Impulse**	**Meaning**	**Getwave Exists**	**Init_Returns_Impulse**	**Meaning**	
1	FALSE	TRUE	Init-Only	FALSE	TRUE	Init-Only	3
2	FALSE	TRUE	Init-Only	TRUE	FALSE	Getwave-Only	1 or 2
3	FALSE	TRUE	Init-Only	TRUE	TRUE	Dual	2
4	TRUE	FALSE	Getwave-Only	FALSE	TRUE	Init-Only	3
5	TRUE	FALSE	Getwave-Only	TRUE	FALSE	Getwave-Only	1,2,or 3
6	TRUE	FALSE	Getwave-Only	TRUE	TRUE	Dual	1
7	TRUE	TRUE	Dual	FALSE	TRUE	Init-Only	iFFT(FFT(3)/FFT(2))
8	TRUE	TRUE	Dual	TRUE	FALSE	Getwave-Only	1
9	TRUE	TRUE	Dual	TRUE	TRUE	Dual	1

Static and Dynamic Equalization

- Static equalization
  - Impulse response processing (Init)
  - Happens once - does not vary from bit to bit
  - Treated as LTI by simulation engine
  - Can be used to generate Statistical and Time-Domain results
- Dynamic equalization
  - Waveform processing (Getwave)
  - Can vary from bit to bit
  - Includes equalization and clock recovery
  - Only used to generate Time-Domain results

Model Type	Equalization
Init-Only	Static
Getwave-Only	Dynamic
Dual	Static & Dynamic

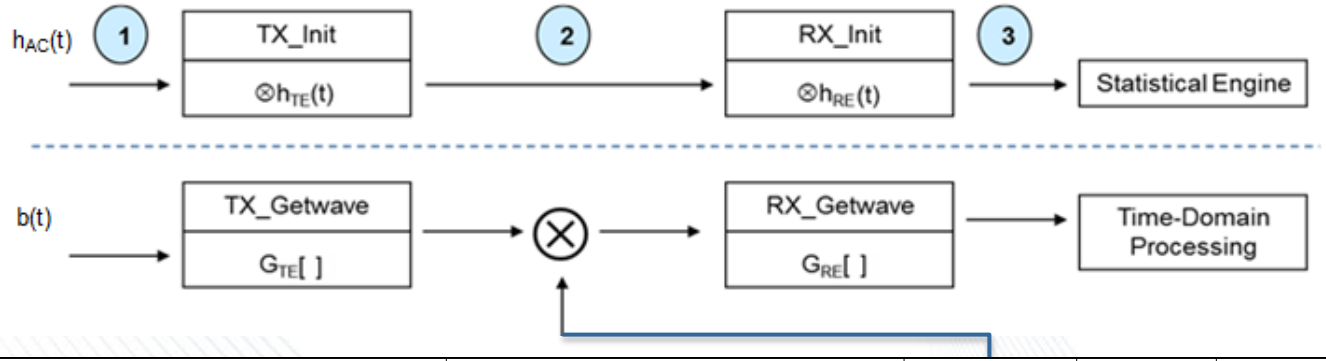
Interpreting Simulation Results

Case #	TX			RX			Statistical	Time Domain
	Getwave Exists	**Init_Returns_Impulse**	**Meaning**	**Getwave Exists**	**Init_Returns_Impulse**	**Meaning**		
1	FALSE	TRUE	Init-Only	FALSE	TRUE	Init-Only	OK	Static TX EQ, Static RX Eq
2	FALSE	TRUE	Init-Only	TRUE	FALSE	Getwave-Only	No RX EQ	Static TX EQ, Dynamic RX Eq
3	FALSE	TRUE	Init-Only	TRUE	TRUE	Dual	OK	Static TX EQ, Dynamic RX Eq
4	TRUE	FALSE	Getwave-Only	FALSE	TRUE	Init-Only	No TX EQ	Dynamic TX EQ, Static RX EQ
5	TRUE	FALSE	Getwave-Only	TRUE	FALSE	Getwave-Only	No TX or RX EQ	Dynamic TX EQ, Dynamic RX EQ
6	TRUE	FALSE	Getwave-Only	TRUE	TRUE	Dual	No TX EQ	Dynamic TX EQ, Dynamic RX EQ
7	TRUE	TRUE	Dual	FALSE	TRUE	Init-Only	OK	Dynamic TX EQ, Static RX EQ
8	TRUE	TRUE	Dual	TRUE	FALSE	Getwave-Only	No RX EQ	Dynamic TX EQ, Dynamic RX EQ
9	TRUE	TRUE	Dual	TRUE	TRUE	Dual	OK	Dynamic TX EQ, Dynamic RX EQ

	Correct equalization of TX and RX modeled
	Correct equalization of TX and RX modeled :Assuming no adaptation in TX
	Assumes Static RX EQ is a good representation of the RX: No Adaptation
	Assumes Static RX EQ is a good representation of the RX: No Adaptation, advanced simulator required
	Equalization data is missing

- Statistical simulations can be missing TX and/or RX equalization, depending on case
  - Some partial statistical results are useful, others are not
- Time-Domain simulations ALWAYS include TX & RX equalization
  - Equalization can be either static or dynamic, depending on the case
- Case 9 fully supports both Statistical & Time-Domain simulation

The Whole Enchilada



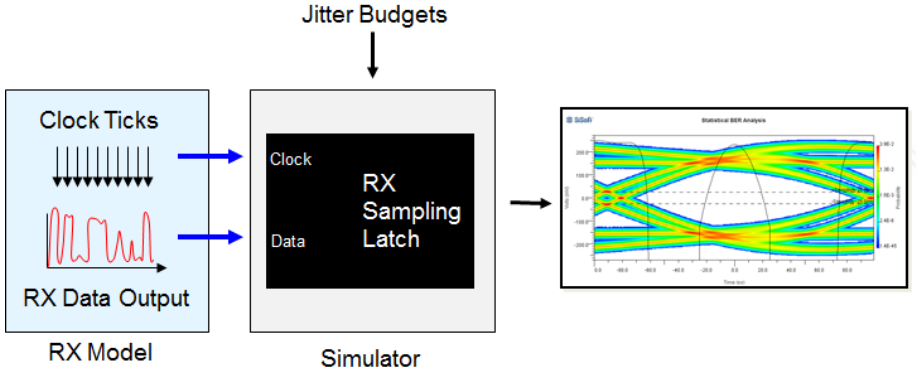
Case #	TX			RX			Convolution Input	Statistical	Time Domain
	Getwave Exists	**Init_Returns_Impulse**	**Meaning**	**Getwave Exists**	**Init_Returns_Impulse**	**Meaning**			
1	FALSE	TRUE	Init-Only	FALSE	TRUE	Init-Only	3	OK	Static TX EQ, Static RX Eq
2	FALSE	TRUE	Init-Only	TRUE	FALSE	Getwave-Only	1 or 2	No RX EQ	Static TX EQ, Dynamic RX Eq
3	FALSE	TRUE	Init-Only	TRUE	TRUE	Dual	2	OK	Static TX EQ, Dynamic RX Eq
4	TRUE	FALSE	Getwave-Only	FALSE	TRUE	Init-Only	3	No TX EQ	Dynamic TX EQ, Static RX EQ
5	TRUE	FALSE	Getwave-Only	TRUE	FALSE	Getwave-Only	1,2,or 3	No TX or RX EQ	Dynamic TX EQ, Dynamic RX EQ
6	TRUE	FALSE	Getwave-Only	TRUE	TRUE	Dual	1	No TX EQ	Dynamic TX EQ, Dynamic RX EQ
7	TRUE	TRUE	Dual	FALSE	TRUE	Init-Only	iFFT(FFT(3)/FFT(2))	OK	Dynamic TX EQ, Static RX EQ
8	TRUE	TRUE	Dual	TRUE	FALSE	Getwave-Only	1	No RX EQ	Dynamic TX EQ, Dynamic RX EQ
9	TRUE	TRUE	Dual	TRUE	TRUE	Dual	1	OK	Dynamic TX EQ, Dynamic RX EQ

	Correct equalization of TX and RX modeled
	Correct equalization of TX and RX modeled :Assuming no adaptation in TX
	Assumes Static RX EQ is a good representation of the RX: No Adaptation
	Assumes Static RX EQ is a good representation of the RX: No Adaptation, Requires advanced math capabilities in Simulator
	Equalization data is missing

SiSoft™ We Are Signal Integrity

Recovered Clock Processing

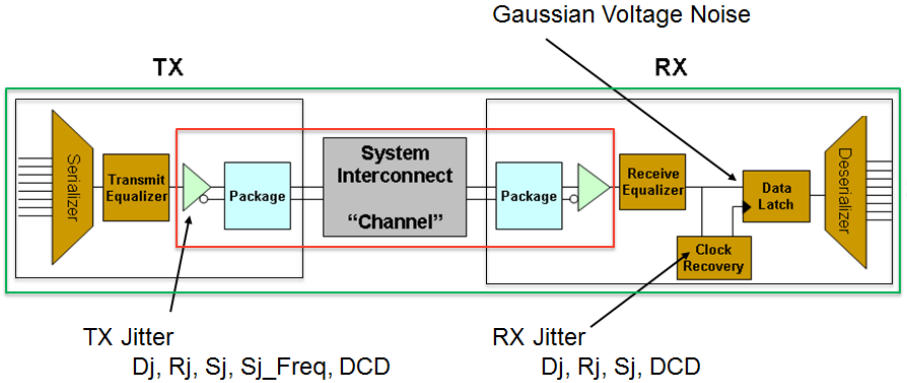
- AMI models can return both equalized signal and recovered clock behavior
- Understanding where an eye is sampled is critical to analyzing system margin
- RX models are not required to return clock behavior
  - Simulators must handle these situations appropriately
- RX jitter budgets are combined with clock behavior returned from the algorithmic model



SiSoft™ We Are Signal Integrity

Jitter and Noise

- IBIS 6.1 provides multiple TX & RX impairments
- TX jitter directly modulates the TX output
  - There is no jitter transfer function; transfer is 100%
- RX jitter affects recovered clock behavior
  - Simulators combine jitter data with clock information returned by the model
- RX noise affects sampling latch data input



Summary

- IBIS-AMI simulation is a multi-stage process that includes pre- and post-processing steps
- IBIS-AMI simulations combine Circuit Simulation & Signal Processing
- Each Algorithmic model can be 1 of 3 types
- Simulators must support 9 different simulation flows
  - The significance of Statistical results varies by flow
  - Time-Domain results have differing TX & RX equalization by flow
- Understanding how simulators work and how to isolate issues is key

Next Steps / Questions

- Is this information complete and correct?
- Should we incorporate these terms into IBIS?
- Would this terminology speed the process of proposing and discussing new BIRDs (e.g. Repeater flows)?

Thanks!

SiSoft™
We Are Signal Integrity