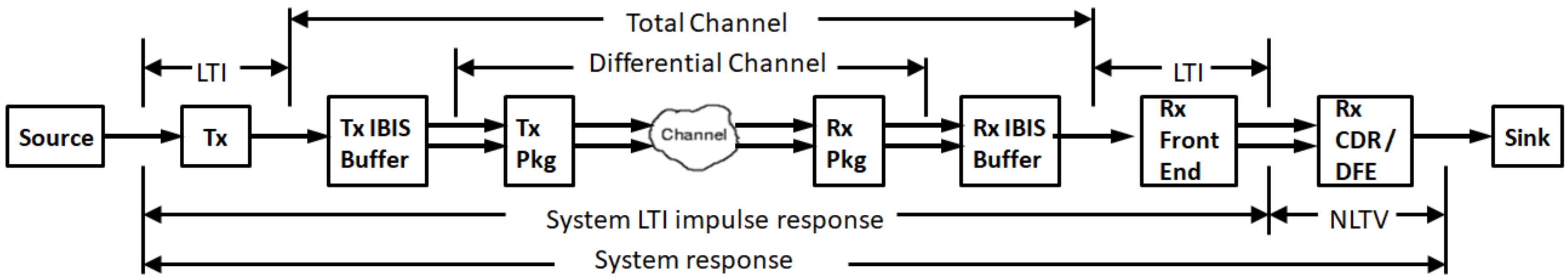
Subject: Portable IBIS-AMI Models

Author: John Baprawski (JB); SerDesDesign.com

Date: April 16, 2021

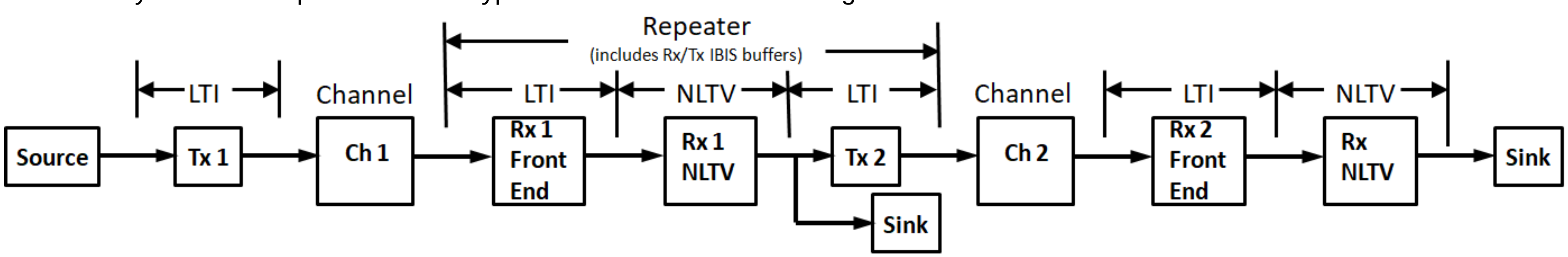
This paper discusses features on the web site: **https://www.serdesdesign.com**

A SerDes system for a single channel has the typical structure shown in this figure.



See details in **About the SerDes System Tool...**

A SerDes system for a repeater has the typical structure shown in this figure.



For a repeater (redriver or retimer) type SerDes system, see details in **About the SerDes Repeater Tool...**

SerDesDesign.com supports generating and exporting portable Linear and Time Invariant (LTI) and Non-Linear and/or Time Variant (NLTV) Transmit (Tx) and Receive (Rx) IBIS-AMI models:

- Tx LTI IBIS-AMI model: Noted as Tx + Tx IBIS Buffer in the above figures
- Rx LTI IBIS-AMI model: Noted as Rx Front End + Rx IBIS Buffer in the above figures
- Rx NLTV IBIS-AMI model: Noted as Rx Front End + Rx CDR/DFE + Rx IBIS Buffer in the above figures
- Repeater IBIS-AMI model: Noted as Repeater in the above figures.
- For reuse in SerDesDesign.com tools or any other Channel Simulator

This document discusses:

- Template based IBIS-AMI models
- Generating IBIS-AMI models
- Ordering IBIS-AMI models including pricing
- Terms and Conditions for delivered IBIS-AMI models
- Settable parameters for generated IBIS-AMI models

Template Based IBIS-AMI models

IBIS-AMI models can be generated with a successful analysis of a user defined SerDes system using one of the SerDesDesign.com SerDes System Channel Simulator tools.

- SerDes System Tool: https://www.serdesdesign.com/home/serdes-system-tool/
- SerDes Repeater Tool: https://www.serdesdesign.com/home/serdes-repeater-tool/
- SerDes EOE Repeater Tool: https://www.serdesdesign.com/home/serdes-eoe-repeater-tool/

Free registration is required to use these tools.

Use our template-based models in our channel simulator tools. Within our Channel Simulator you can specify your Tx and/or Rx models and test them within the context of a SerDes channel. When you are satisfied with your design and its system performance, you can order the model you created by setting 'GenerateModels' for your next Run. See the following sections on generating and ordering your models.

Our template-based IBIS-AMI LTI Tx models include:

- Tx IBIS buffer definition based on the IBIS standard up to version 7.0.
- Feed Forward Equalizer (FFE) where the user has knowledge of the FFE taps and filtering.
- FEE based on black box measurements where the user only has knowledge of the FFE stimulus response characteristics.
- FFE with registers where the user has knowledge of the digital register architecture of their FFE.
- Continuous time Linear Equalizer (CTLE) where the user has knowledge of the CTLE in terms of 1) peaking characteristics, 2) poles and zeros, 3) time domain step responses, or 4) frequency domain spectrums.

Our template-based IBIS-AMI LTI Rx models include:

- Rx IBIS buffer definition based on the IBIS standard up to version 7.0.
- CTLE with 1 section where the user has knowledge of the CTLE in terms of 1) peaking characteristics, 2) poles and zeros, 3) time domain step responses, or 4) frequency domain spectrums.
- CTLE with 2, 3, or 4 sections where the user has knowledge of the CTLE in terms time domain step responses.
- FFE where the user has knowledge of the FFE taps and filtering.
- FEE based on black box measurements where the user only has knowledge of the FFE stimulus response characteristics.

Our template-based IBIS-AMI NLTV Rx models include:

- Nonlinearity model based on a tanh() function, Rapp function, table data or list of tables.
- Clock and Data Recovery (CDR) model based on user defined Jitter Transfer Function (JTF) characteristic.
- CDR with Decision Feedback Equalizer (DFE) model based on user define DFE tap characteristics.

Our template-based IBIS-AMI Repeater models include:

- Any combination of Rx IBIS buffer, Rx LTI, Rx Nonlinearity, Rx NLTV models and Tx LTI, Tx IBIS buffer models.

Our template-based IBIS-AMI EOE Repeater models include:

- IBIS-AMI Repeater supplemented with mid channel optical transmitter, optical channel and optical receiver.

All of our template based models may include your specification jitter and corner cases.

Generating IBIS-AMI models

IBIS-AMI models can be generated with a successful analysis of a user defined SerDes system using one of the SerDesDesign.com SerDes System Channel Simulator tools.

- SerDes System Tool: https://www.serdesdesign.com/home/serdes-system-tool/
- SerDes Repeater Tool: https://www.serdesdesign.com/home/serdes-repeater-tool/
- SerDes EOE Repeater Tool: https://www.serdesdesign.com/home/serdes-eoe-repeater-tool/

Free registration is required to use these tools.

For discussion purposes, use of the SerDes System Tool is discussed.

After a user has successfully created their SerDes system design and after they have simulation results that are acceptable to them, then IBIS-AMI models are generated by setting parameter GenerateModels > 0 in section 7. Setup Analysis.

The IBIS-AMI models are generated at the end of the successful analysis after the section 8. Run Analysis 'Run' button is selected.

- When GenerateModels = 1, then the Tx IBIS-AMI model is generated.
- When GenerateModels = 2, then the Rx IBIS-AMI model is generated.
- When GenerateModels = 3, then both the Tx and the Rx IBIS-AMI model is generated.
- When GenerateModels = 4, then the Repeater IBIS-AMI model is generated (requires use of the Repeater or EOE Repeater tool).

After the successful analysis (after the Run Analysis 'Run' button is selected and with GenerateModels > 1), the displayed log file shows the IBIS-AMI model generation message after the line:

> Exiting Channel Analysis with success; run time = <xxx> sec.

Within the model generation message is a model reference number of the form: xxx_yyy where xxx is the assigned user number and yyy is the unique assigned part number. This model reference number, along with the model generation message is used to order your IBIS-AMI models.

For each IBIS-AMI model, the files generated are:

- <model name>.ami
- <model name>.ibs
- <model name>.dll (for Windows 64 bit)
- <model name>.so (for Linux 64 bit)
- <model name>_Definition.pdf; Definition and use of the IBIS-AMI models.
- <model name>_Verification.pdf; Verification test report for the IBIS-AMI models.

When the IBIS-AMI model is ordered, the user can specify the <model name> in place of this default name. See the next section on ordering the model.

Free example IBIS-AMI models generated using SerDesDesign.com

- **https://www.serdesdesign.com/home/web_documents/models/Pass_Through_Tx_and_Rx_IBIS_AMI_Models.zip**
    - This example is for a simple pass thru Tx and Rx IBIS-AMI model.
    - Included in this zip file are all the deliverables listed above.
- **https://www.serdesdesign.com/home/web_documents/models/Tx_FFE_IBIS_AMI_Model.zip**
    - This example is for a Tx feed forward equalizer (FFE) with one pre-cursor and one post-cursor.
    - Included in the Tx AMI model is the ability to specify that the FFE taps be automatically optimized for the channel it is being used with.
    - Included in this zip file are all the deliverables listed above.
- **https://www.serdesdesign.com/home/web_documents/models/Rx_CTLE_802_3_83E_IBIS_AMI_Model.zip**
    - This example is for a Rx continuous time linear equalizer (CTLE) with one 9 states that provide 1 to 9 dB peaking in 1 dB steps as defined in the IEEE standard:
        - IEEE Standard for Ethernet Section Six Annex 83E, Table 83E–2—Reference CTLE coefficients
    - Included in the Rx AMI model is:
        - the ability to specify that the CTLE dB peaking be automatically optimized for the channel it is being used with
        - the ability to enable an Automatic Gain Control (AGC) to set the output of the Rx model to a specified level.
    - Included in this zip file are all the deliverables listed above.

Ordering IBIS-AMI models including pricing

After the successful analysis (after the Run Analysis 'Run' button is selected and with GenerateModels > 1), the displayed log file shows the IBIS-AMI model generation message after the line:

Exiting Channel Analysis with success; run time = <xxx> sec.
==

Within the model generation message is a model reference number of the form: xxx_yyy where xxx is the assigned user number and yyy is the Linux 10 digit date/time stamp. This model reference number, along with the model generation message is used to order your IBIS-AMI models.

Copy and paste the entire model generation message (which includes the model reference number) into an email addressed to:

admin@serdesdesign.com

This email becomes your Statement of Work (SOW).

Use your email address as used to log into SerDesDesign.com. Your IBIS-AMI models will be delivered to that email address.

Within your email defined alternate names for the model name and the parameter names if you do not want to use the default names.

Example list of Preferred Names:

Default Name	Preferred Name
SerDes_ChannelTest	HSD_28G_TX
Tx_CornerCase	CornerCase
Tx_AdaptForChannel	AutoInitFFE
Tx_PreCursorValue	Cm1
Tx_PostCursorValue	C1

OS required:

Operating System	Required
Windows 64 bit	Yes

Linux 64 bit	Yes

Each IBIS-AMI model includes these files:
- <model name>.ami
- <model name>.ibs
- <model name>.dll (for Windows 64 bit)
- <model name>.so (for Linux 64 bit)

Each IBIS-AMI model order includes these files:
- <tx model name and/or rx model name>_IBIS_AMI_Model_Definition.pdf; Definition and use of the IBIS-AMI models.
- <tx model name and/or rx model name>_SerDes_System_Verification.pdf; Verification test report for the IBIS-AMI models.

Each IBIS-AMI model can be used in any Channel Simulator that complies with the IBIS-AMI standard version 6.0 (or above) for NRZ data streams; version 6.1 (or above) for PAM4 data streams or repeaters; version 7.0 (or above) for S-parameter based IBIS buffers.

Deposit your funds using PayPal and your PayPal account or your company credit card on the SerDesDesign.com Store page: **https://www.serdesdesign.com/home/store/**

See the Store web page for pricing.

Here is example pricing an IBIS-AMI model:
- Deposit $2000 USD for the first OS (Windows 64 bit or Linux 64 bit)
- Deposit additional $1000 for the second OS (Windows 64 bit or Linux 64 bit)
- Deposit additional $1000 for support for 3 corner cases
- For example: for Tx and Rx models, no corners, for both Windows 64 bit and Linux 64 bit, the amount required is $6000 USD.

Your deposit becomes your purchase order (PO).

In addition to your models (an IBIS-AMI model order is a pre-requisite), you can order a Web based IBIS-AMI modeling training course with which an IBIS-AMI Modeling Kit and the source code for your models will be delivered to you. The Web based training enables you to further edit and modify the model, model name, model parameters and model data. This includes your ability to change (if applicable) the step response data files for all corner cases associated with this model. This training and your own model building does not require you to purchase anything else: no channel simulator, no C++ code development environment, no other tools. It does require that you obtain and install specific free public domain software. With this training and the training deliverables, you can make as many modifications to your models by yourself as needed.

If a portable Linux 64 bit build machine is needed that is fully loaded with build software and your software IBIS-AMI models and ready to use for your further modification and build purposes, a Linux build machine (works with your PC) can be ordered as part of the IBIS-AMI Modeling Training for an additional cast.

See details on IBIS-AMI Modeling Training at this link: **IBIS_AMI_Modeling_Training...**

The terms and conditions for your purchase are listed below.

Terms and Conditions for delivered IBIS-AMI models

See terms and conditions for IBIS-AMI Modeling are at this link: **Terms & Conditions | Privacy Policy**

Settable Parameters for Generated IBIS-AMI models

Generated IBIS-AMI models have settable parameters based on which template-based models were used. The settable parameters are listed here for each of the template-based models.

For detail discussion on model parameters, see the model documentation page that is accessed when the HELP button is selected on the model Pop-Up Web dialog box.

If jitter was enabled for the Tx model, the Tx Reserved_Parameters may include entries for jitter (Tx_Rj, Tx_Dj, Tx_DCD, Tx_Sj, Tx_Sj_Frequency).

If jitter was enabled for the Rx model, the Rx Reserved_Parameters may include entries for jitter (Rx_Rj, Rx_Dj, Rx_DCD, Rx_Sj).

Settable parameters for template-based IBIS-AMI LTI Tx models:

- Tx IBIS buffer definition based on the IBIS standard up to version 7.0.

    - No settable parameters. The model is as defined from setup.
- TransmitterType=1: Feed Forward Equalizer (FFE); settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
OutputGain	Additional output scalar gain	Real		
EnableAdaptForChannel	Auto adapt FFE during initialization to equalize the channel	Integer	[0,1]	0 = No; 1 = Yes
PreCursorValues	Comma separated list of precursor values; 1st is closest to the main tap	Real array		Used when AdaptForChannel=0
PostCursorValues	Comma separated list of post cursor values; 1st is closest to the main tap	Real array		Used when AdaptForChannel=0

- TransmitterType=2: FEE based on black box measurements; settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
OutputGain	Additional output scalar gain	Real		
EnableAdaptForChannel	Auto adapt during initialization to equalize the channel	Integer	[0,1]	0 = No; 1 = Yes
PostCursor1	PostCursor1 index	Integer	[0,NumPostCursor1-1]	
PostInverted	Set to 1 to invert the postcursor data tap gain value	Integer	[0,1]	
Drive	Drive index	Integer	[0,NumDrive-1]	

PreCursor1	PreCursor1 index	Integer	[0,NumPreCursor1-1]	Used when EnablePreCursor=1
PreInverted	Set to 1 to invert the postcursor data tap gain value	Integer	[0,1]	Used when EnablePreCursor=1
CornerCase	Corner case	Integer	[0,2]	Used when EnableCorners=1 0 = Typ; 1 = Slow; 2 = Fast

- TransmitterType=3: FFE with registers; settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
EnableAdaptForChannel	Auto adapt DRVTRIM, DATA_INV and DEL during initialization to equalize the channel	Integer	[0,1]	0 = No; 1 = Yes
OutputGain	Additional output scalar gain	Real		
DRV	Integer representing number of 8 register segments that are active	Integer	[0,7]	The rightmost DRV+1 integers are used
DRVTRIM	String of 8 integers representing the level for the associated DRV segment	String	[0,7]	The rightmost DRV+1 integers

				are used
DATA_INV	String of 8 integers where 1 inverts the value associated with DRVTRIM	String	[0,1]	The rightmost DRV+1 integers are used
DEL	String of 8 integers where 1 assigns segment to pre-cursor, 2 assigns main cursor, 3 assigns post-cursor	String	[1,3]	The rightmost DRV+1 integers are used
NL_GainSplit	Gain applied before the nonlinear model with 1/Gain applied after	Real	>0	
CornerCase	Corner case	Integer	[0,2]	0 = Typ; 1 = Slow; 2 = Fast

- TrasnmitterType=4: Continuous time Linear Equalizer (CTLE); settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
EnableAdaptForChannel	Auto adapt Data1Index during initialization to equalize the channel	Integer	[0,1]	0 = No; 1 = Yes
OutputGain	Additional output scalar gain	Real		
Data1Index	Data set index	Integer	[1,NumData]	Used when AdaptForChannel=0

CornerCase	Corner case	Integer	[0,2]	Used when EnableModelType= 3 or 4 and EnableCorners=1 0 = Typ; 1 = Slow; 2 = Fast

Settable parameters for template-based IBIS-AMI LTI Rx models:

- Rx IBIS buffer definition based on the IBIS standard up to version 7.0.
    - No settable parameters. The model is as defined from setup.
- ReceiverFEType=1, 2, 3 or 4: CTLE with 1 section; settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
InputGain	Additional input scalar gain	Real		
EnableAutoGainControl	Auto adapt output gain during initialization to achieve the AGC_Level	Integer	[0,1]	0 = No; 1 = Yes
AGC_Level	Target eye sampling instant peak to peak level	Real	>0	
EnableAdaptForChannel	Auto adapt Data1Index during initialization to equalize the channel	Integer	[0,1]	0 = No; 1 = Yes
Data1Index	Data 1 set index	Integer	[1,NumData1]	Used when AdaptForChannel=0

Data2Index	Data 2 set index	Integer	[1,NumData2]	Used when AdaptForChannel=0 and ReceiverFEType>=2
Data3Index	Data 3 set index	Integer	[1,NumData3]	Used when AdaptForChannel=0 and ReceiverFEType>=3
Data4Index	Data 4 set index	Integer	[1,NumData4]	Used when AdaptForChannel=0 and ReceiverFEType=4
CornerCase	Corner case	Integer	[0,2]	Used when EnableModelType = 3 or 4 and EnableCorners=1. 0 = Typ; 1 = Slow; 2 = Fast

- ReceiverFEType=5: FFE based on tap values; settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
InputGain	Additional input scalar gain	Real		
EnableAutoGainControl	Auto adapt output gain during initialization to achieve the AGC_Level	Integer	[0,1]	0 = No; 1 = Yes
AGC_Level	Target eye sampling instant peak to peak level	Real	>0	
EnableAdaptForChannel	Auto adapt FFE during initialization to equalize the channel	Integer	[0,1]	0 = No; 1 = Yes
PreCursorValues	Comma separated list of precursor values; 1st is	Real		Used when AdaptForChannel=0

	closest to the main tap	array		
PostCursorValues	Comma separated list of post cursor values; 1st is closest to the main tap	Real array		Used when AdaptForChannel=0

- ReceiverFEType=6: FEE based on black box measurements; settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
InputGain	Additional input scalar gain	Real		
EnableAutoGainControl	Auto adapt output gain during initialization to achieve the AGC_Level	Integer	[0,1]	0 = No; 1 = Yes
AGC_Level	Target eye sampling instant peak to peak level	Real	>0	
EnableAdaptForChannel	Auto adapt during initialization to equalize the channel	Integer	[0,1]	0 = No; 1 = Yes
PostCursor1	PostCursor1 index	Integer	[0,NumPostCursor1-1]	
PostInverted	Set to 1 to invert the postcursor data tap gain value	Integer	[0,1]	
Drive	Drive index	Integer	[0,NumDrive-1]	
PreCursor1	PreCursor1 index	Integer	[0,NumPreCursor1-1]	Used when EnablePreCursor=1

PreInverted	Set to 1 to invert the postcursor data tap gain value	Integer	[0,1]	Used when EnablePreCursor=1
CornerCase	Corner case	Integer	[0,2]	Used when EnableCorners=1 0 = Typ; 1 = Slow; 2 = Fast

Settable parameters for template-based IBIS-AMI NLTV Rx models:

- ReceiverNLType=1: Nonlinearity model; settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
InputGain	Input scalar gain	Real	>0	
OutputGain	Output scalar gain	Real	>0	
GainSplit	Gain applied before the nonlinearity with 1/GainSplit applied after the nonlinearity	Real	>0	
VinVoutIndex	Integer that selectes one of NumVinVout sets of nonlinearity data in the table	Integer	[1,NumVinVout]	Used when SyncNLwithRxFE = 0
CornerCase	Corner case	Integer	[0,2]	Used when NonlinearityType=3 or 4 OR LPFType=2 or 3

				0 = Typ; 1 = Slow; 2 = Fast
StepRespIndex	Integer that selectes one of NumStepResp sets of nonlinearity LPF step resp data in the table	Integer	[1,NumStepResp]	Used when SyncNLwithRxFE = 0

- ReceiverNLTVType=1: Clock and Data Recovery (CDR) model; settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
CDR_OJTS_Fc	CDR observed jitter transfer specification approximate corner frequency in Hz	Real	>0	
CDR_SamplerUIDelay	CDR sampler delay in UI units	Real	[-0.5, 0.5]	0 = Auto set

- ReceiverNLTVType=2: CDR with Decision Feedback Equalizer (DFE) model; settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
CDR_OJTS_Fc	CDR observed jitter transfer specification approximate corner frequency in Hz	Real	>0	
CDR_SamplerUIDelay	CDR sampler delay in UI units	Real	[-0.5, 0.5]	0 = Auto set
DFE_AutoInitTaps	Automatically initialize DFE taps	Integer	[0,1]	0 = No; 1 = Yes

DFE_TapValues	Comma seperated list of initial DFE_NumTaps DFE tap values or codes	Real array		Used when EnableAutoInitTaps=0
EnableDFE_Adapt	Enable DFE tap adaptive equalization	Integer	[0,1]	0 = No; 1 = Yes
DFE_Alpha	DFE taps update factor	Real	>0	
EnableCDR_ConstClock	Override DFE input clock after DFE_Ignore_Bits with constant clock phases	Integer	[0,1]	0 = No; 1 = Yes
DFE_OutputUIDelay	DFE output delay in UI units	Real	[-0.5, 0.5]	

Settable parameters for template-based IBIS-AMI Repeater models:

- Any combination of Rx IBIS buffer, Rx LTI, Rx Nonlinearity, Rx NLTV models and Tx LTI, Tx IBIS buffer models.
- The settable IBIS-AMI model parameters are based on the above tables.

Settable parameters for template-based IBIS-AMI EOE Repeater models:

- IBIS-AMI Repeater supplemented with mid channel optical transmitter, optical channel and optical receiver.
- The settable IBIS-AMI model parameters are based on the above tables plus these additional parameters for the Optical link.
- OpticalTransmitter: VCSEL with driver model; settable IBIS-AMI model parameters.

Name	**Description**	**Type**	**Limits**	**Comment**
AvgCurIndex	Average current index	Integer	[1,NumAvgValues]	
ModCurIndex	Modulation current index	Integer	[1,NumModValues]	

EnableAWGN	Enable VCSEL output AWGN	Integer	[0,1]	0 = No; 1 = Yes
Vn_mVrms	AWGN noise level in mVrms	Real	[0,10]	
NBW_Ratio	AWGN noise bandwidth in SymbolRate units	Real	[0.5, 2.0]	

- OpticalChannel: Optical fiber model; settable IBIS-AMI model parameters.

Name	Description	Type	Limits
FiberLoss	Optical cable loss per unit length in dB/km	Real	[0,10]
FiberLength	Optical cable length in meters	Real	[0,300]
FiberConnectorLoss	Optical cable connector loss in dB	Real	[0,3]
FiberCouplingLoss	Optical cable coupling loss in dB	Real	>=0
FiberLengthBW	Optical cable length bandwidth product GHz*m	Real	[5,5000]

- OpticalReceiver: Optical PIN/TIA model; settable IBIS-AMI model parameters.

Name	Description	Type	Limits	Comment
Responsivity	Photo diode responsivity in mA/mW	Real	>0	
Capacitance	Photo diode capacitance in F	Real	>=0	

Resistance	Trans impedance gain in V/A	Real	>0	
EnableNRZ_Level	Enable restore to NRZ level	Integer	[0,1]	0 = No; 1 = Yes auto set 2 = Yes manually set
GainSplit	Gain applied before the nonlinearity with 1/GainSplit applied after the nonlinearity	Real	>0	
CornerCase	Corner case	Integer	[0,2]	Used when EnableCorners=1 0 = Typ; 1 = Slow; 2 = Fast