Simulating High-Speed Serial Channels with IBIS-AMI Models

Bob Sullivan, Michael Rose, Jason Boh

Application Note

Introduction

The Input/Output Buffer Information Specification (IBIS) has been an essential component of the electrical simulation toolbox for nearly two decades. Many design engineers are familiar with the application of IBIS models and for the most part, the models have provided an accurate, easy-to-use alternative to SPICE-based transistor models. In fact, most IBIS models are simple behavioral translations of a vendor's SPICE buffer model. However, as serial interface bit rates increase, critical limitations with IBIS models have become more acute.

With the current (version 5.0) of the IBIS specification, an important algorithmic modeling component has been added to the conventional behavioral analog IBIS model. Several earlier efforts were made to add a mixed-signal model capability with limited success. IBIS-Algorithmic Modeling Interface (AMI) represents an important milestone in the IBIS mixed-signal evolution.

This paper reviews some of the benefits and limitations of using IBIS models and introduces the new AMI extensions to the latest IBIS version 5.0 specification. Additionally, it illustrates how to perform several simulations of a typical backplane system using the Advanced Design System 2011 (ADS2011) toolset.

Agilent Technologies

IBIS Historical Perspective

The first version of IBIS was released in 1993 through the IBIS Open Forum. There are many reasons for the popularity of IBIS models; namely, that they are widely available, standardized (ANSI/EIA-656 and GEIA-STD-0001), and the specification is controlled by an open forum with a membership of top tier EDA and silicon vendors and equipment manufacturers. Since IBIS models are behavioral, simulation times are usually very fast and do not suffer from convergence issues. Unlike un-encrypted SPICE transistor-level circuit models, IBIS models do not expose the intellectual property (IP) of the silicon vendor or foundry. In addition, transistor-level models are generally encrypted for a single specific EDA tool. In contrast, IBIS models are portable allowing them to run on any EDA tool that supports the standard. IC vendors are not burdened with generating and supporting one model for each EDA tool. And, integration support for IBIS is excellent. Moreover, there are a number of free utilities for viewing, translating and parsing IBIS models.

IBIS models can support most IO buffer types and signaling standards through a table of I-V and V-t curves for both transistors in the complimentary pair of a buffer. Rising and falling edges are characterized in separate tables. The I-V curves provide the steady state characteristics, while the V-t curves modify the buffer's behavior for transient conditions. Together, these tables capture the fundamental buffer characteristics including nonlinearity behaviors. As shown in *Figure 1*, the basic IBIS electrical model is able to characterize the output buffer including the clamp diodes, the die capacitance and the lumped package parasitics.

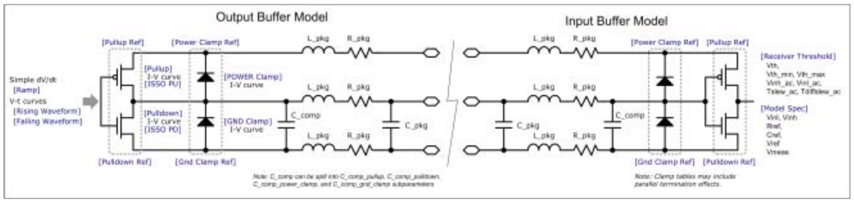


Figure 1. A basic IBIS input and output buffer model schematic.

The power pins also include package parasitics that help model simultaneous switch noise. Additional keywords can be used to characterize behaviors such as slew-rate with the [Ramp] keyword and the [Vmeas], [Vref], [Rref], & [Cref] keywords for modeling Tco loading characteristics.

IBIS models do have some inherent limitations. For instance, they do not include any internal timing information for calculating input to output propagation delays. Also, the simple lumped element package model (shown in *Figure 1* as L_pkg, R_pkg, and C_pkg) only includes the self impedance and not mutual impedance. Later versions of IBIS had a mechanism for including an external ".pkg" file for defining RLGC impedance matrices. Some silicon vendors provide package models in this format, although broadband S-parameter package models are typically used for high-speed serial analysis. Generally speaking, early IBIS models were unable to simulate crosstalk in IC packages. Later versions of IBIS provided a simple mechanism for including the effects of simultaneous switching output (SSO) and ground bounce through the [Pin Mapping] function used to associate specific signal and power pins. Starting with IBIS 4.0, the C_Comp value can be split into separate components for Pullup, Pulldown, POWER Clamp, and GND Clamp, which improves power integrity simulation capabilities.

Another set of limitations involves the C_comp element shown in *Figure 1*. This capacitor is intended to represent the buffer's die capacitance. However, die capacitance has both a frequency and voltage dependence that cannot be accurately represented with a single, fixed C_comp value. Also, there is a problem with the C_comp implementation that involves the way differential buffers are constructed with two single-ended buffers associated through the [Diff Pin] keyword. While the C_comp component can reasonably model the common-mode capacitance in differential configurations, it does not model the differential mode capacitance that can lead to AC errors. Another common type of error involving C_comp occurs if the loading capacitance is not considered in the [ramp] V-t curves.

Another limitation became apparent as interest grew in adding pre-emphasis to IBIS transmitter models. The C_comp value is usually extracted and treated as an external capacitor element in EDA tools and therefore cannot be adjusted on-the-fly when a model uses IBIS [Driver Scheduling] to switch between the normal and boosted buffer outputs.

Introducing IBIS-AMI

As the complexity of high-speed serial channels continues to evolve, a new simulation methodology has emerged to manage the ballooning simulation run times. With bit times dropping down to the picoseconds range, classic time-domain transient simulations that iteratively solve a set of simultaneous equations are no longer practical due to the extremely long simulation time and the expanding transistor count associated with the complex digital equalization blocks. IBIS-AMI addresses this problem by supporting both time-domain superposition (quasi-analytic or bit-by-bit mode) and statistical mode. These relatively new simulation techniques offer orders-of-magnitude improvement in simulation run time while preserving the accuracy of transient convolution simulation. IBIS-AMI still performs a transient convolution of the channel's differential mode impulse response initially. This step is accomplished in a much shorter period (on the order of 20 to 30 unit intervals, depending on the settling time of the analog channel). From that point on, solutions are provided by signal processing functions that execute much faster than transient convolution solvers. These methods are contrasted in greater detail later in this paper and in several references cited at the end.

At the same time, designers face a new challenge characterizing the complex digital signal processing functions found in multi-Gbps transceivers such as equalization and clock data recovery (CDR). Very high bit rates typically require sophisticated equalization techniques to cancel out a channel's attenuation and dispersion losses and must be accounted for if the simulation is to have meaningful results.

The arrangement of a typical high-speed serial interface is shown in *Figure 2*. On the transmit side, the data stream is serialized, encoded and fed to the DSP. Likewise, at the receiver, the stream is de-serialized and decoded before being buffered. DSP signal processing filtering blocks are used to implement functions such as Feed-Forward Equalization (FFE) and Decision Feedback Equalization (DFE), Clock Data Recovery (CDR), and bit slicing. The analog sections contain the behavioral buffer descriptions, as well other important analog functions such as package parasitics, and for some transceivers, a linear equalizer stage. The transceiver's analog buffers along with the physical channel form the "analog channel model," which is characterized together with the impulse response transient simulation.

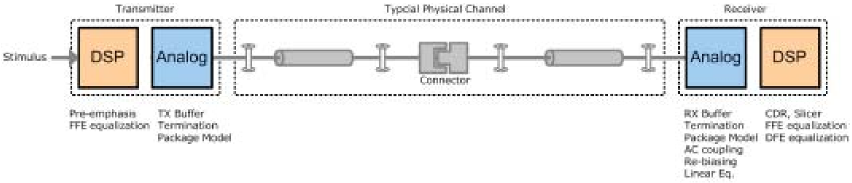


Figure 2. IBIS-AMI model partitioning.

SerDes designers have often used various combinations of custom and off-the-shelf mixed-signal modeling tools such as MATLAB, Verilog-AMS and VHDL-AMS to co-simulate the digital and analog sections of their transceivers. IC vendors are understandably reluctant to release the algorithmic code used to implement the digital pieces since it contains valuable IP. A system engineer interested in the overall channel behavior would have to model the transceivers behaviorally using generic macro models or create custom algorithmic functions in a tool such as MATLAB or Agilent's Ptolemy. Without specific knowledge of a particular transceiver design, this task is difficult at best. Several SerDes vendors, such as IBM with its HSSCDR MATLAB-based simulator, provide models embedded and distributed with its proprietary simulator tool. Unfortunately though, this tool is not interoperable between IC or EDA tool vendors.

The IBIS 5.0 AMI models the functions typically implemented in the DSP block at a behavioral level. Unlike other mixed-signal modeling formats though, the IP is hidden and protected within a compiled executable that is called by the EDA tool through a standardized interface. The algorithmic code is provided as an executable DLL Dynamic Link Library (DLL) file for Windows-based PCs or an Shared Object (SO) file for Linux systems.

IBIS-AMI Simulation Requirements

The simulation methods used in IBIS-AMI 5.0 depend on the assumption that the analog channel is both linear and time invariant. The Linear and Time Invariant (LTI) premise allows accurate and efficient conversion between the channel's impulse response and frequency response through the fast Fourier transform (FFT). Likewise, the input to output transfer function may be derived from the impulse response through the convolution process $y(t) = x(t) * h(t)$. The transmit bit stream is conditioned by the convolved impulse response to predict the signal at the receiver. The resultant Rx waveform is then used to develop the eye opening contours using superposition.

While the passive interconnect elements within a channel are typically LTI, the IBIS Tx buffer tables can have nonlinear characteristics. CMOS buffers often have time variations in impedance as well. Tx equalization tap settings may also impact the buffer's linearity. Even though IBIS specifies the LTI requirement, it cannot be simply assumed and testing is needed to gain confidence in the simulation results. Methods for handling nonlinear, time-variant (NLTV) buffer behavior are currently being discussed in the IBIS Advanced Technology Modeling Task Group (e.g., for the purpose of modeling mid-channel repeater chips like re-drivers and re-timers, and optoelectronic links).

Simulation Flow

To understand the capabilities and limitations of IBIS-AMI models, it is important to understand how IBIS-AMI models interact with the simulation tool. IBIS-AMI supports two primary simulation flow methodologies: one for time-domain simulation using a superposition (or “bit-by-bit”) technique and another for statistic simulation. Although the two methods have similar performance and accuracy, they also have unique limitations. For instance, the algorithmic model may support NLTV equalization behaviors for time-domain simulation, whereas statistic simulations require LTI equalization modeling. Also, in some EDA tools, jitter modeling is implemented differently for statistical simulation. For time-domain simulations, Tx jitter is added to the stimulus waveform. For statistical simulation, Tx jitter may be post-processed at the receiver output by some EDA platforms.

(Note that ADS applies the same Tx jitter treatment in time domain and statistical modes. Adding Tx jitter in post-processing does not account for channel jitter amplification).

To prevent interactions, each element shown in *Figure 3* below is joined by ideal electrical interfaces (outputs have zero impedance and inputs have infinite impedance).

IBIS-AMI defines several standardized interfaces between the EDA tool and the algorithmic model for passing impulse response and waveform data. The initial version 5.0 release of IBIS-AMI had several critical problems and included unnecessarily complicated modeling scenarios. BIRD 120 addresses these issues with a refined simulation flow strategy. Only the BIRD 120 flow will be discussed. Many EDA tool vendors have already implemented the new flow, including Agilent Technologies with its ADS tool.

As mentioned previously, there are two fundamental simulation flows supported by IBIS-AMI: a statistical simulation flow for models with LTI equalization algorithms; and a time-domain flow which permits nonlinear or time-variant equalization characteristics. In either case, IBIS-AMI simulations begin by characterizing the channel’s impulse response in the time domain. This is typically accomplished by generating a Heaviside step function at the transmitter’s analog buffer and converting the response at the receiver’s analog buffer by calculating the impulse response using the first-order derivative of the step response. With the impulse response of the analog channel, noted as $h_{AC}(t)$ in IBIS-AMI terminology, an IBIS-AMI simulation processes the effects of the models’ filtering functions (equalization) quite differently for time-domain or statistical methods.

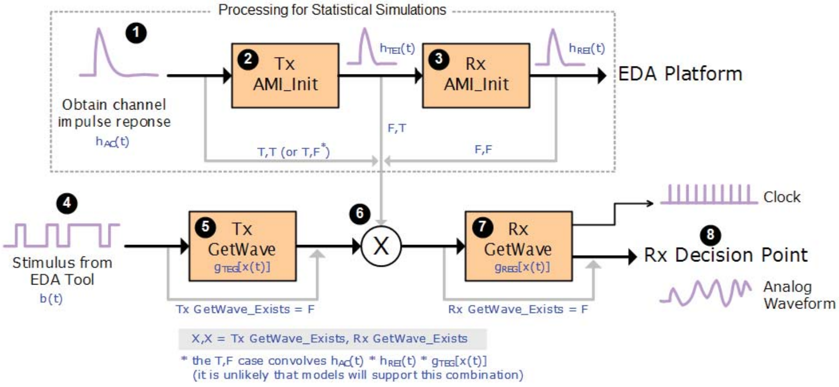


Figure 3. The IBIS-AMI statistical and time-domain reference flow.

For statistical simulation processing, just the block shown in Figure 3 steps 1 to 3 is used. The analog channel impulse response is generated in step 1 and passed to the Tx AMI_Init() and Rx AMI_Init() functions. Typically, the calls apply signal processing to the impulse response and output the modified response. These processing functions are noted in Figure 3 as $h_{TEI}(t)$ and $h_{REI}(t)$. (If the Init_Returns_Impulse setting is false, the call will pass the input response through without modification). Finally, the EDA tool processes the filtered output from the Rx AMI_Init() call using its standard statistical simulation process.

The model interaction scenarios for time domain processing are more elaborate using various combinations of the AMI_Init() and AMI_GetWave() functions. As discussed earlier, time-domain simulations allow the possibility of modeling NLTV equalization behaviors. In a time-domain simulation, specific bit stimulus patterns are applied and the filtered analog (and clock tick) waveforms are output. The BIRD 120 time-domain reference flow is represented in Figure 3, steps 1 to 8.

As with the statistical simulation reference flow, the time-domain process starts by generating the analog channel response which may be combined with some number of crosstalk aggressors into an impulse matrix and passed to the Tx AMI_Init() function. Referring to Figure 3, equalization can be applied in either the AMI_Init() or AMI_GetWave() calls, although applying the filtering in AMI_GetWave() is obviously preferred since it can support NLTV algorithms. After the AMI_Init() functions are executed, the stimulus waveform is applied by the EDA tool in steps 4 and 5 to the Tx AMI_GetWave() function. If the models' Tx GetWave_Exists is false, the bitstream b(t) is passed through unchanged to step 6 where it can be convolved with the filtering performed in the AMI_Init() calls (referred by IBIS-AMI as $h_{TEI}(t)$ or $h_{REI}(t)$ for Tx or Rx respectively). If Tx GetWave_Exists is true, Tx equalization is applied in the Tx AMI_GetWave() function (note that $h_{TEI}(t)$ filtering from the Tx AMI_Init() call will not be used so that the equalization is not double-counted).

If the Rx GetWave_Exists parameter is true, Rx equalization is applied in step 7. The analog waveform is then output to the EDA platform for additional processing and viewing (this is known as the "Rx Decision Point" in IBIS-AMI terminology). Also, if the Rx GetWave_Exists parameter is true and the function supports it, clock ticks from the CDR section of the model can be output to the tool. For instance, ADS uses the clock tick output to center the eye density and coutour plots for accurate eye margin measurements. Please note that the flow shown in Figure 3 is contingent on Tx and Rx Init_Returns_Impulse parameters being true.

Although the various flow scenarios may appear confusing at first glance, knowing how a model declares the Init_Returns_Impulse and GetWave_Exists parameters allows the simulation engineer to quickly understand the type of simulation to run, how and where filtering is applied, and what to expect for output.

As discussed above, IBIS-AMI provides a standard mechanism to communicate a model's capability through several IBIS [Reserved_Parameters] in the model file. They are defined as either true or false:

- **Init_Returns_Impulse** – indicates that the model AMI_Init() function can return a filtered response (shown as steps 2 and 3 in the reference flow in Figure 3)
- **GetWave_Exists** – indicates that the model supports the AMI_GetWave() function (shown as steps 5 and 6 in the reference flow in Figure 3). Note that if GetWave_Exists is false, then Init_Returns_Impulse must be true.
- **Use_Init_Output** – NO LONGER SUPPORTED beyond version 5.0 (BIRD 120).

There are two basic processing combinations: AMI_Init() only, and both AMI_Init() and AMI_GetWave(). Since the Tx and Rx models are independent, there are 4 possible processing scenarios that a typical EDA tool may support (based on the assumption that Init_Returns_Impulse is always true and that GetWave_Exists can be either true or false for both Rx and Tx). Note that ADS does not support the unusual case where Tx GetWave_Exists is true and Rx GetWave_Exists is false.

IBIS-AMI File Structure

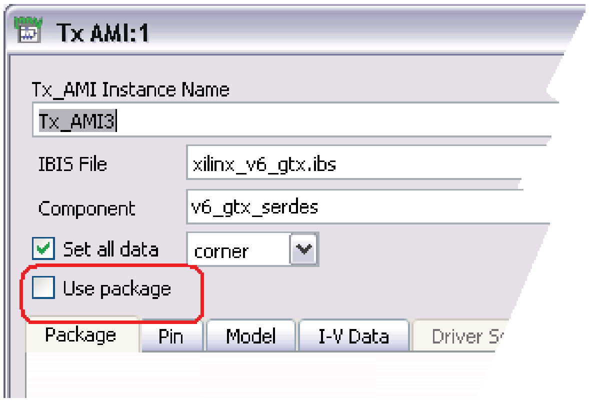
The file structure used for IBIS-AMI models is very simple. The familiar ASCII-text <model>.IBS file contains several new keywords for referencing the algorithmic executables (refer to Sections 6c and 10 in the IBIS v5.0 specification). The pointer to the algorithmic shared library file, the AMI parameter file and the operating system-specific Platform_Compiler_Bits declaration is given under the keyword [Algorithmic_Model]. The Platform_Compiler_Bits defines the operating system and compiler for the shared library file (typically a .DLL format for Windows OS). The entry also defines whether the OS is 32 or 64 bits. Although the IBIS parser will check the OS version, having an incompatible version in the executable may cause problems with some EDA tools (remember to double-check compatibility). IBIS supports multiple operating systems for the executable shared library. All supported versions of a given model will be listed under the [Algorithmic_Model] keyword. A single top-level <model>.IBS file references both Tx and Rx algorithmic files.

The IBIS algorithmic parameter file is a simple ASCII-text file with an AMI extension. The parameter file contains two main sections. The first section lists [Reserved_Parameters] that define the model's standardized capabilities, such as the Init_Returns_Impulse and GetWave_Exists declarations. The file can be edited to add parameters such as Tx_Jitter or Rx_Clock_PDF definitions. The second, optional, section is listed under the keyword [Model_Specific] and is used to pass simulation parameters to the executable for controlling model specific settings such as equalization, CDR and signal swing. The usage rules of the parameters listed under these keywords are controlled by the arguments: in, out, inout, and info. All IBIS files are assumed to be located in the same directory so that the EDA tool can resolve the location pointers.

Constructing a Simulation Topology

In the following sections, the steps involved for simulating a typical backplane system are described. The methods for adding and configuring IBIS-AMI models are shown. The model settings are parameterized so that the equalization settings can be optimized with a batch process.

The IBIS package model limitations discussed earlier are even more severe at gigabit data rates. Instead of using the IBIS package parasitics or even IBIS .PKG models, most simulation users will opt for external S-parameter models provided by the IC vendor, integrating them into the passive channel model. Do not forget to disable the IBIS model package parasitic entries. Some EDA tools such as ADS include a selection for disabling the package entries without having to edit the .IBS file (as shown in *Figure 4*). In fact, with the Virtex6 IBIS-AMI models used in the following simulations, the package parasitics values are nulled by default.



Tx AMI:1

Tx_AMI Instance Name: Tx_AMI3

IBIS File: xilinx_v6_gtx.ibs

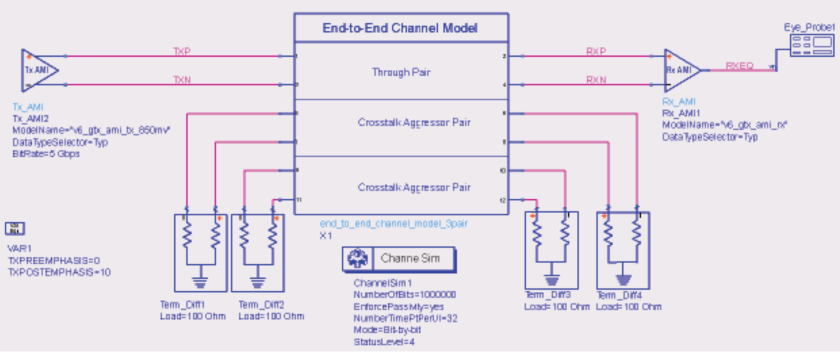
Component: v6_gtx_serdes

☑ Set all data: corner

□ Use package

Figure 4. Disabling the IBIS package parasitics.

A typical backplane channel topology was developed for the following simulation examples. The connector and vias were created in a full-wave electromagnetic (EM) solver and saved as 12-port Touchstone models covering a frequency range of DC to 20 GHz. All 3 pairs are fully coupled to model crosstalk effects. The traces are implemented using ADS MultiLine elements, which model frequency dependent dielectric dispersion loss. The ADS Multiline models also account for resistive loss from conductor skin effect and surface roughness.



Tx_AMI
Tx_AMI2
ModelName="v6_gtx_ami_tx_850mV"
DataTypeSelector=Typ
BitRate=5 Gbps

Rx_AMI
Rx_AMI1
ModelName="v6_gtx_ami_rx"
DataTypeSelector=Typ

VAR1
TXPREEMPHASIS=0
TXPOSTEMPHASIS=10

Term_Diff1
Load=100 Ohm

Term_Diff2
Load=100 Ohm

ChannelSim1
NumberOfBits=1000000
EnforcePassMty=yes
NumberTimePtPerUI=32
Mode=Bit-by-bit
StatusLevel=4

Term_Diff3
Load=100 Ohm

Term_Diff4
Load=100 Ohm

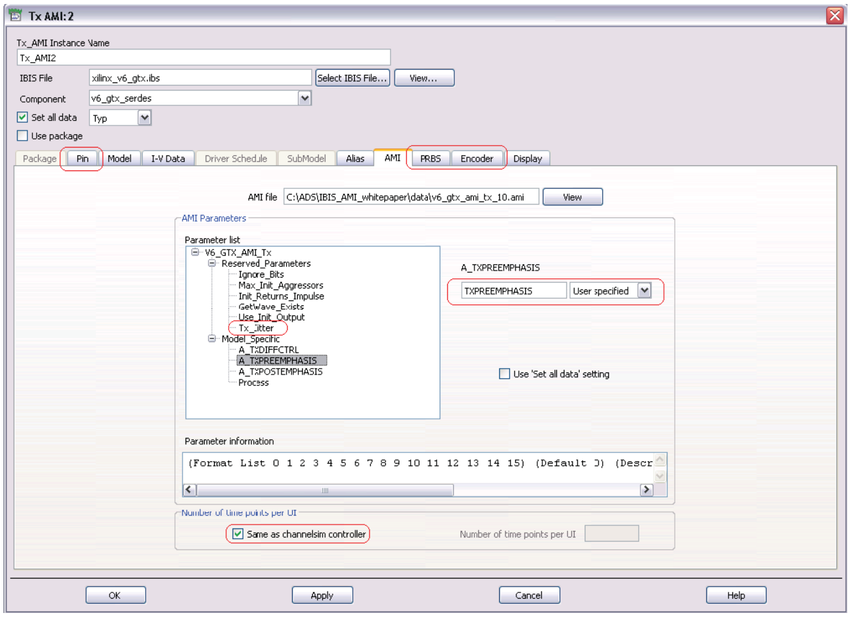
Figure 5. Demonstration channel topology.

Configuring the IBIS-AMI Models

Adding the IBIS-AMI models in ADS is straightforward. In the test case, Virtex 6 GTX IBIS-AMI transceiver models are used. All model files, including the DLLs are placed in the data folder for the project. The IBIS-AMI Tx and Rx symbols are placed as shown in the topology diagram in *Figure 5*. The Tx model is configured for a given output swing with the [Model Selector] keyword located under the "Pin" tab. This method can be somewhat confusing. Since there is a separate <model>.AMI file for each output setting, modifying the file manually (for instance, to add a jitter statement) requires that you either have to remember which file has been selected or add the statement to all Tx model files. In the Virtex 6 GTX model, each output swing model includes unique V-I table and [ramp] entries.

The AMI tab shown in *Figure 6* provides both the standardized list of supported model attributes under the IBIS [Reserved_Parameters] heading and a list of unique items for this particular model under the IBIS [Model_Specific] keyword. In the example, the Tx_Jitter statement was added manually to the model using the syntax outlined in the IBIS v5.0 specification. The Tx equalization parameters were assigned to user variables for sweeping the settings using the ADS Batch Simulation controller. The TXDIFFCTRL setting should match the Tx model selection so that the AMI_Getwave() calls will be in sync with the V-I, V-t tables settings declared in the main <model>.IBS file. The PRBS and Encoder tabs shown in *Figure 6* provide a means for configuring the bit stream stimulus for time-domain simulations. In this case, the stimulus was configured as an 8b/10b encoded, 8-bit PRBS pattern. A user-configured bit sequence or bit file can also be used.

Also note the checkbox for setting the time step for time-domain analysis. Smaller time steps will model electrically finer features with better accuracy, but will increase simulation time. In this example, the setting was made to match the configuration in the Channel Simulation controller's convolution setup.


Tx AMI:2

Tx_AMI Instance Name: Tx_AMI2

IBIS File: xilinx_v6_gtx.ibs

Component: v6_gtx_serdes

☑ Set all data: Typ

☐ Use package

AMI file: C:\ADS\IBIS_AMI_whitepaper\data\v6_gtx_ami_tx_10.ami

AMI Parameters

Parameter list

```
⊟ V6_GTX_AMI_Tx
   ⊟ Reserved_Parameters
      Ignore_Bits
      Max_Init_Aggressors
      Init_Returns_Impulse
      GetWave_Exists
      Use_Init_Output
      Tx_Jitter
   ⊟ Model_Specific
      A_TXDIFFCTRL
      A_TXPREEMPHASIS
      A_TXPOSTEMPHASIS
      Process
```

A_TXPREEMPHASIS: TXPREEMPHASIS — User specified

☐ Use 'Set all data' setting

Parameter information

```
(Format List 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15) (Default 3) (Descr
```

Number of time points per UI

☑ Same as channelsim controller

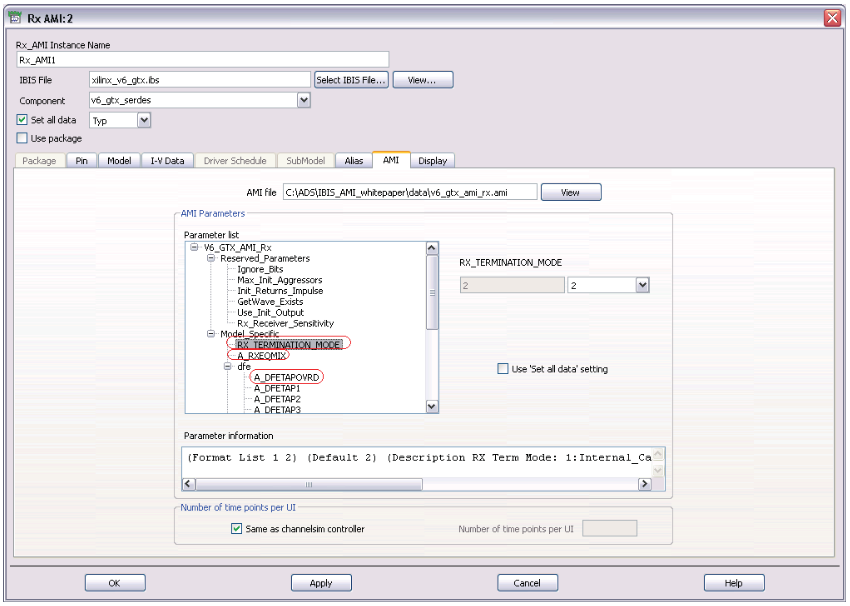
Number of time points per UI

Figure 6. The ADS IBIS-AMI Tx setup.

Setting up the IBIS-AMI Rx model (shown in *Figure 7*) is similar to the process used for the Tx model. It is important to take the time to fully understand the configuration options. For the Virtex 6 GTX model used in this example, Xilinx provides detailed transceiver configuration information in application note UG366. This transceiver includes a continuous time linear equalization stage and a manually-configured or auto-adaptive DFE equalizer function. In this example, the channel isn't very lossy, so we do not need (and prefer not to use) either continuous time linear equalizer (CTLE) or Decision Feedback Equalizers (DFE). The appropriate overrides are set with the RXEQMIX and DFETAPOVRD parameters. The termination option must also be set appropriately to match the application.

Applying the Rx model illustrates a fundamental advantage of using IBIS-AMI models for channel simulation. With IBIS-AMI, the simulation results will generally include the effect of any internal gain stages in the receiver. Historically, a receiver model just characterized the loading effects. Consequently, simulations that measured the eye opening at the input to a receiver did not always represent the actual eye opening seen at the internal digitizer. Often the analog front end details are hidden from the user making it impossible to account for their effects. Previously, when using ADS behavioral transceivers configured to emulate the GTX transceivers, CTLE had to be modeled externally by extracting the poles and zeros by curve-fitting to the published frequency response plots.

IBIS-AMI models have the ability to model the bit sampling point since the CDR can be included in the Rx model. As a result, the effect of Rx jitter (such as sinusoidal jitter and periodic jitter) through the CDR's phase-locked-loop (PLL) can be modeled. ADS automatically adjusts the effective sampling point on the bit stream output of the Rx AMI_Getwave**()** call.



Rx_AMI Instance Name: Rx_AMI1

IBIS File: xilinx_v6_gtx.ibs

Component: v6_gtx_serdes

☑ Set all data: Typ

☐ Use package

AMI file: C:\ADS\IBIS_AMI_whitepaper\data\v6_gtx_ami_rx.ami

Parameter list

```
V6_GTX_AMI_Rx
  Reserved_Parameters
    Ignore_Bits
    Max_Init_Aggressors
    Init_Returns_Impulse
    GetWave_Exists
    Use_Init_Output
    Rx_Receiver_Sensitivity
  Model_Specific
    RX_TERMINATION_MODE
    A_RXEQMIX
    dfe
      A_DFETAPOVRD
      A_DFETAP1
      A_DFETAP2
      A_DFETAP3
```

RX_TERMINATION_MODE: 2 | 2

☐ Use 'Set all data' setting

```
(Format List 1 2) (Default 2) (Description RX Term Mode: 1:Internal_Ca
```

☑ Same as channelsim controller

Number of time points per UI

Figure 7. The ADS IBIS-AMI Rx setup dialog.

For the example topology, the channel is initially configured without crosstalk aggressors. One percent of random jitter is added to the transmitter model. The topology is swept with several Tx equalizer and output swing settings using the statistical simulation mode. The eye opening is evaluated against the desired bit error rate goal. In this example, it looks like only a moderate amount of post-cursor transmit equalization is needed. With -2.5 dB of post-cursor de-emphasis, the results shown in *Figure 8* indicate an excellent margin at a BER of 10e-12.

(Note that the eye density plot also includes the BER 10e-12 contour and a mask representing the Virtex 6 GTX receiver threshold requirements.)

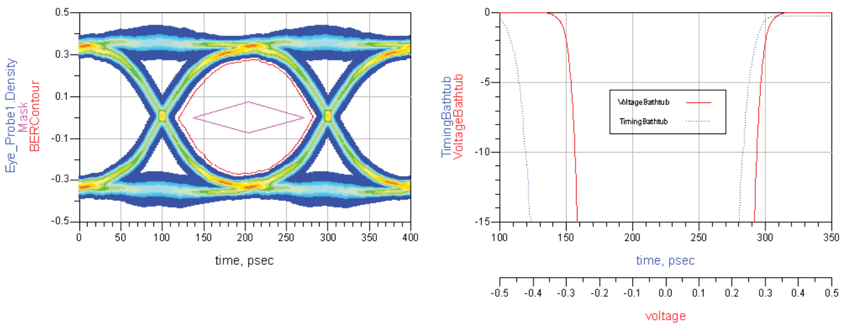


Figure 8. Eye density and bathtub plots with -2.5 dB of Tx equalization.

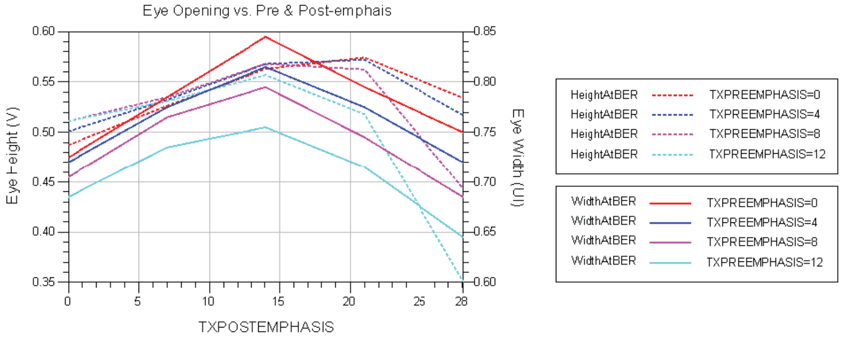


Figure 9. Sweeping Tx equalization to optimize the eye opening.

In *Figure 9*, the vertical and horizontal eye opening are plotted for 20 combinations of transmitter equalization configurations (please refer to the Xilinx UG366 transceiver guide to see how the TXPOSTEMPHASIS index corresponds to the magnitude of de-emphasis expressed in dB). The best results are obtained with a post-cursor de-emphasis index of 14 (-2.5 dB) and 0 dB of pre-cursor de-emphasis. These settings will be used for the following simulations.

It is also evident from the Eye Density and Contour plot in *Figure 8* that the output swing is much higher than needed for the relatively low loss channel used in the simulation example. Subsequent simulations will reduce the output swing. The lower output helps reduce crosstalk and electromagnetic interference (EMI) noise levels.

Adding Crosstalk Sources

The IBIS-AMI crosstalk transmitters shown in *Figure 10* are equivalent to the standard IBIS-AMI transmitter model except that the relative phase and bit pattern may be adjusted independently. For the simulation examples, the worst-case phase difference of zero degrees is used (e.g., synchronous crosstalk). Otherwise, the configuration settings match the through-channel transmitter. The crosstalk transmitters are arranged as near-end aggressors which were found to generate higher crosstalk noise than other configurations. The driver differential output swing was reduced to 665 mV based on the high margins seen in the previous simulation. As before, the post-cursor de-emphasis is set to -2.5 dB and .01 UI of random jitter is added to the transmitter.

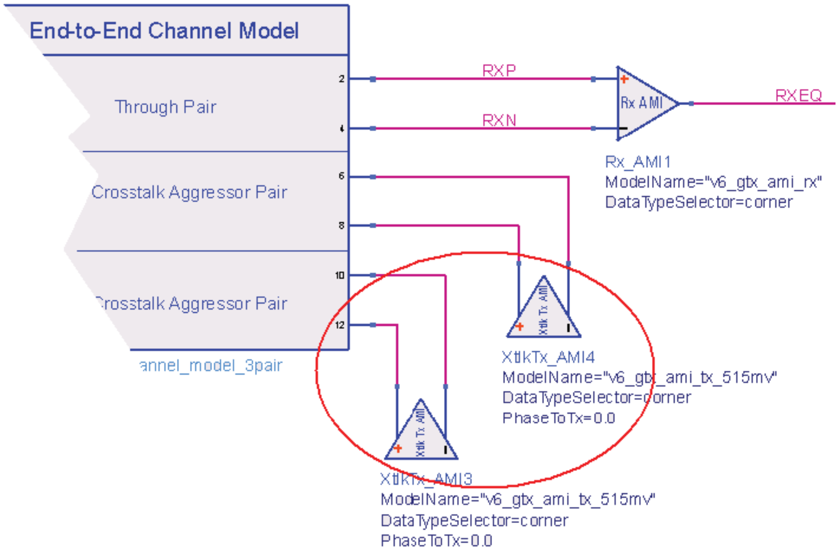


Figure 10. The passive terminations are swapped out with IBIS-AMI NEXT crosstalk transmitters.

The simulation results with the added crosstalk drivers are shown in *Figure 11*. A measured signal-to-noise ratio of over 16:1 suggests that the output swing is still a bit generous. A horizontal eye opening of 0.805 UI at the target bit error rate of 10e-12 is shown in the bathtub plot.

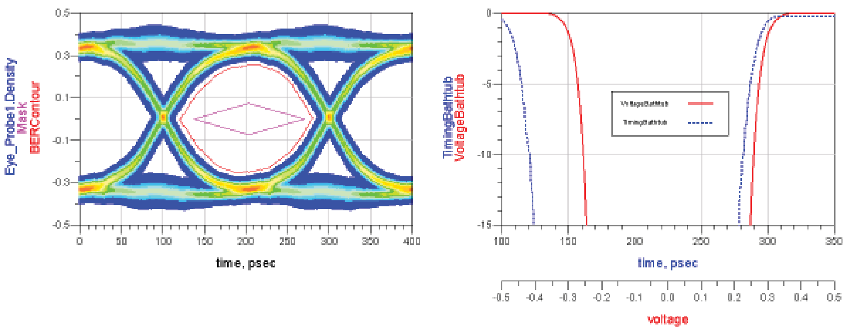
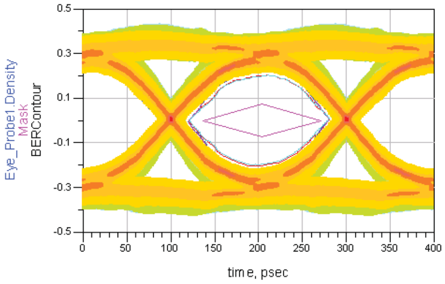


Figure 11. An eye opening with 2 NEXT crosstalk aggressors.

To ensure adequate margins over a variety of operating conditions, the topology is simulated with the effects of process, voltage and temperature (PVT) variations added (*Figure 12*). IBIS models generally provide minimum (slow-weak), maximum (fast-strong) and typical PVT characteristics. Ideally, the output buffer model will have independent V-I and V-t tables for all three cases. In addition, the C_comp value in the model should have separate values. If the receiver model includes entries for Vth_min and Vth_max under the [receiver threshold] keyword, the logic thresholds will vary as well.



Corner	Eye Height @ BER 10e-12	Eye Width @ BER 10e-12	SNR
min	0.402	0.805	10.238
max	0.470	0.805	13.619
fast	0.478	0.795	14.205
slow	0.396	0.795	9.862

Figure 12. An eye opening over minimum, maximum, fast, and slow corners.

These example simulations demonstrate the ease of setting up an IBIS-AMI channel simulation and the efficiency that comes with fast simulation run times for optimizing a particular topology. IBIS-AMI models enable modeling of the receiver input stages so that accurate margins through the receiver's analog front end can be evaluated.

Future IBIS Improvements

Several important improvements in modeling accuracy and capability are being considered for the upcoming version 5.1 release of the IBIS specification. These improvements include:

- BIRD 116, IBIS-ISS (Interconnect SPICE Sub-circuits). Adds support for commonly used SPICE constructs that are used to model interconnect elements such as traces and connectors. The SPICE elements supported will include ideal and lossy transmission line models (U- and W-element), S-parameter models, SPICE sources (E, G, etc.), etc. Circuit descriptions will be embedded in standard SPICE .subckt element wrappers referenced from [external model] or [external circuit] keywords. BIRD 125 makes use of the IBIS-ISS methods for adding package models under the [Define Package Model] keyword. Perhaps IC vendors will start integrating broadband package models in their IBIS-AMI device models.
- BIRD 120, IBIS-AMI Flow Correction. Fixes version 5.0 algorithmic flow inconsistencies. Adds clarity regarding LTI and non-LTI support. Also, the Use_Init_Output option under the [Reserved_Parameter] keyword has been eliminated, simplifying the number of possible model support options.
- BIRD 122. Broadband analog modeling of the analog buffers. Improves on the capability offered under the [model] keyword for characterizing the IBIS model analog buffer. Either a broadband Touchstone model or an equivalent circuit (R-C) can be referenced under the [Reserved_Parameters] keyword. At the time of this writing, adoption of BIRD 122 has been rejected and a new version will be re-submitted for consideration.
- BIRD 123. New jitter, noise and clock modeling parameters. This BIRD greatly expands the jitter modeling characterization of both Tx and Rx devices. The version 5.0 parameter Tx_Jitter is broken out into the components Tx_RJ, Tx_SJ (with Tx_SJ_Frequency). Also, Tx_DCD format has been redefined. The version 5.0 parameter Rx_Clock_PDF is replaced with the parameters Rx_Clock_Recovery_RJ, Rx_Clock_Recovery_SJ, and Rx_Clock_Recovery_DCD. Additionally, another useful parameter Rx_Clock_Recovery_Mean is defined. It represents the fixed offset between the recovered clock and the median eye sample point.

Another initiative is currently being discussed in the IBIS Advanced Technology Modeling Task Group that will define the parameters and methods for optimizing Tx equalization based on receiver adaption to a training pattern. Numerous standards, such as 10GBase-KR and PCIe Gen 3 use this method for transmitter equalization. The extension will define the parameters for the training pattern, the back-channel protocol, tap coefficient format, etc.

Conclusions

IBIS version 5.0 represents an important milestone in the specification's long successful history. It embraces the latest simulation methodologies, supports algorithmic equalization and CDR modeling, and adds the ability to model cross-talk and jitter. Simulation time for large complex channels is computationally efficient, allowing parameters such as equalization coefficients to be optimized quickly. It appears likely that major IC vendors will quickly gravitate to this standard since it is designed to be interoperable between EDA toolsets and it protects their valuable IP within a compiled executable.

As shown in the example simulation, IBIS-AMI models can be easily integrated into a system simulation workflow. Even so, the simulation engineer has to carefully study the capability and limitations of a particular model set. It is not plug-and-go. For those willing to take the time to fully understand the models (and the transceivers that they represent), the accuracy and efficiency of system simulation will be enhanced. Knowing the limitations of a particular IBIS-AMI model and performing diligent checking is critical to accurate simulation, especially at higher bit rates or with very lossy or resonant channel elements. Be aware that there can be a considerable variation in accuracy in the way nonlinear buffers are modeled.

Several critical improvements are coming. BIRDs 120 and 123 are significant improvements. Additionally, there has been a lot of activity around adding or improving the Power Delivery Network (PDN) and SSO modeling capability. Although not specifically discussed in this paper, IBIS version 5 has already implemented some of these improvements. For more information see the version 5.0 Gate Modulation Effects table support [ISSO_PU, ISSO_PD] and the [Composite_Current] keyword from BIRD95, which permits the definition of pre-driver current.

IBIS Resources

- IBIS Open Forum:
  http://www.eigroup.org/ibis/specs.htm
- IBIS Advanced Technology Task Group:
  http://www.eda.org/pub/ibis/macromodel_wip/
- IBIS Quality Task Group:
  http://www.vhdl.org/pub/ibis/quality_wip/
- IBIS 5.0 Specification:
  http://eda.org/pub/ibis/ver5.0/
- IBIS Cookbook (v4.0):
  http://www.eda.org/ibis/cookbook/cookbook-v4.pdf
- IBIS Golden Parser:
  http://www.eda.org/ibis/ibischk5/
- IBIS active BIRDs:
  http://www.eda.org/pub/ibis/birds/

References

1. Anthony Sanders, Mike Resso and John D. Ambrosia. Channel Compliance Testing Utilizing Novel Statistical Eye Methodology, DesignCon 2004.
2. Chad Morgan, Tyco Electronics. Validation of Quasi-Analytical and Statistical Simulation Techniques for Multi-Gigabit Interconnect Channels.
3. IBIS (I/O Buffer Information Specification), Version 5.0, August 29, 2008.
4. Predicting BER with IBIS-AMI, DesignCon, Feb. 4, 2010.
5. R. Mellitz, M Tsuk, T. Donisi, and S. Pytel, Strategies for coping with non-linear and time variant behavior for high speed serial buffer modeling, DesignCon 2008.
6. Walter Katz, Mike Steinberger, and Todd Westerhoff. IBIS-AMI Terminology Overview, 2009 IBIS Summit.
7. Virtex 6 FPGA GTX Transceivers User Guide, UG366 (v2.5), Jan. 17, 2011.
8. IBIS Modeling Cookbook for IBIS Version 4.0, The IBIS Open Forum, Sept. 15, 2005.
9. Rigorous Modeling of Transmit Jitter for Accurate and Efficient Statistical Eye Simulation, DesignCon 2010.

Author Biographies

Bob Sullivan is the CTO for Engineered Packaging at Curtiss-Wright Controls Electronic Systems and is responsible for keeping abreast of industry technology trends, setting technical direction for the product line, and defining technical approaches to solve challenging problems for Curtiss-Wright's key customers. He is active on OpenVPX, VITA/VSO and PICMG technical standards committees, he chairs the VITA 68 VPX Compliance Channel working group, and he recently chaired the OpenVPX Development Chassis team. Mr. Sullivan has more than 30 years' experience in the design of high performance instrumentation and systems, holds a number of patents in the design of high performance systems, and has authored various technical papers and magazine articles.

Michael Rose is a Principal Engineer specializing in Signal Integrity analysis at Curtiss-Wright Controls Electronic Systems. Michael has been involved in the design and development of analog and digital equipment for more than 30 years, developing custom analog and digital devices including power devices and systems, embedded microprocessor boards, network processor and line cards, system management & protection devices.

Jason Boh is an applications engineer for Agilent EEsof EDA in the greater Boston area, where he is responsible for new product sales, customer training, and support. Jason holds a Master of Science in Electrical Engineering degree from the University of South Florida, where he participated in the Wireless and Microwave Information Systems (WAMI) program. His past experience includes design and fabrication of amplifiers, receivers, and other RF and microwave circuits using PCB, GaAs, and SiGe technologies. Jason also has expertise in high frequency test and measurement, high speed digital signal integrity simulation, electromagnetic simulation, and device modeling.

For more information, visit:
High Speed Digital Design and Simulation
http://www.agilent.com/find/signal-integrity-analysis

Signal Integrity Blog
http://www.agilent.com/find/signal-integrity

✉ **Agilent Email Updates**

www.agilent.com/find/emailupdates
Get the latest information on the products and applications you select.

www.agilent.com

For more information on Agilent Technologies' products, applications or services, please contact your local Agilent office. The complete list is available at:

www.agilent.com/find/contactus

Americas	
Canada	(877) 894 4414
Brazil	(11) 4197 3500
Mexico	01800 5064 800
United States	(800) 829 4444

Asia Pacific	
Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	1 800 112 929
Japan	0120 (421) 345
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Other AP Countries	(65) 375 8100

Europe & Middle East	
Belgium	32 (0) 2 404 93 40
Denmark	45 70 13 15 15
Finland	358 (0) 10 855 2100
France	0825 010 700* *0.125 €/minute
Germany	49 (0) 7031 464 6333
Ireland	1890 924 204
Israel	972-3-9288-504/544
Italy	39 02 92 60 8484
Netherlands	31 (0) 20 547 2111
Spain	34 (91) 631 3300
Sweden	0200-88 22 55
United Kingdom	44 (0) 131 452 0200

For other unlisted countries:
www.agilent.com/find/contactus

Revised: June 8, 2011

Product specifications and descriptions in this document subject to change without notice.

Published in USA, November 15, 2011
5990-9111EN

Agilent Technologies