UNDERSTANDING IBIS-AMI SIMULATIONS

Track 6, Session TH4

Douglas Burns  Signal Integrity Software, Inc.
dburns@sisoft.com

John Madsen  Signal Integrity Software, Inc.
jmadsen@sisoft.com

Todd Westerhoff  Signal Integrity Software, Inc.
Twesterhof@sisoft.com

Walter Katz  Signal Integrity Software, Inc.
wkatz@sisoft.com
Abstract

IBIS-AMI models support two different simulation methods called Init and Get_Wave which process impulse response and waveform data, respectively. As such, AMI models can be developed that support Init only, Get_Wave only or both simulation methods. Given that a TX and a RX model are implemented with one of three different styles means that there are nine possible simulation flows that must be handled. This paper discusses the consequences of each flow, how the Clock_PDF is used along with the eye characterization to predict BER and bathtub curves, and how assumptions about jitter characteristics impact simulation results.
Author(s) Biography

Douglas Burns manages SiSoft's consulting and support services group. Doug has over 33 years of experience leading teams in designing hardware, performing signal integrity analysis, and implementing ICs for Honeywell, Digital, and Compaq Computer. Doug has a successful track record of bringing products to market and holds seven patents in computer system design. Doug graduated Magna cum Laude from the University of Massachusetts with a B.S.E.E. degree and received an M.S.E.E. degree from Northeastern University.

John Madsen
John Madsen is a principal member of SiSoft’s technical staff and is focused on serial channel design and simulation. Before joining SiSoft, John was a successful designer of high-speed hardware for the networking and server industries at companies such as Calxeda Inc., TippingPoint Technologies, and Alcatel-Lucent. John received the B. Eng and M. Eng degrees from Carleton University, Canada and has been awarded two patents.

Todd Westerhoff has over 35 years of experience in the modeling and analysis of electronic systems, including 18 years of signal integrity experience. Todd is responsible for SiSoft's efforts working with semiconductor vendors to test and develop high quality simulation models. Prior to joining SiSoft, Todd held senior technical and management positions for Cisco and Cadence, worked as an independent consultant, and held product marketing positions at Compact Software, Racal-Redac, FutureNet and HHB-Systems. Todd holds a B.E. degree in electrical engineering from the Stevens Institute of Technology in Hoboken, New Jersey.

Walter Katz, Ph.D
Dr. Katz is a pioneer in the development of constraint driven printed circuit board routers. He developed SciCards, the first commercially successful auto-router. As SiSoft’s Chief Scientist, Dr. Katz has focused on areas of extraction, netlisting, simulation, and data management. Dr. Katz holds a Ph.D. from the University of Rochester and a B.S. from Polytechnic Institute of Brooklyn.
Understanding IBIS-AMI Simulations

This paper attempts to answer some of the common questions that come up when starting out with IBIS-AMI-based simulations and expands on the information in the IBIS specification. Section 1 reviews underlying assumptions, associated IBIS syntax and introduces statistical and time-domain simulation. Section 2 describes different types of IBIS-AMI models and the analysis flows they require. Section 3 describes how IBIS-AMI model outputs are post-processed to produce eye diagrams and predict Bit Error Rates. Section 4 discusses how jitter is introduced into IBIS-AMI simulations.

1 IBIS-AMI Basics

IBIS AMI models represent an alternative to traditional time-domain circuit simulation techniques in that they are capable of simulating many more bits and can therefore be used to verify serial link operation to lower bit error rates. Two fundamental assumptions underlie the IBIS-AMI approach.

- First, we assume that a transmitter or receiver can be broken into two independent parts:
  - An analog part representing a transmitter’s output buffer and termination or a receiver’s input termination
  - An algorithmic, or digital, part modeling the equalization and clock recovery capabilities of the device
- Furthermore, we assume that the analog part of the transmitter and receiver, as well as the channel interconnect, are both linear and time-invariant (LTI) and therefore can be completely characterized by a single impulse response.
  - This path from output buffer through the passive interconnect to the input buffer is referred to as the analog channel, as shown in Figure 1.

Based on these assumptions we can perform simulations that create eye diagrams and report various characteristics about each simulation. Simulations themselves can be performed using two different methods: Statistical simulation and time-domain simulation.
Statistical simulation:

- Assumes that, in addition to the LTI assumption about the analog channel, the algorithmic content of the TX and RX are LTI as well.
  - This allows the end to end channel to be simulated with impulse response techniques.
- Produces an eye diagram showing all combinations of bits capable of driving the channel and generating inter-symbol interference (ISI). (For readers interested in the details of the statistical technique, [3] provides a good overview.) The eye represents the probability of the signal occurring throughout the unit interval (UI). An example is shown in Figure 2.
- It is important to note that a statistical simulation has no bit stream; the eye is constructed based solely on the pulse response of the channel and the principle of superposition.
Time-domain simulation:
- Uses bit-by-bit simulation where the models for algorithmic portions of the TX and RX can be non-LTI.
- Provides models with waveform data stream created by the EDA tool. Models modify waveforms according to their equalization capabilities before passing them back to the EDA tool.
- Constructs an eye diagram by overlaying individual bits of waveform data returned by the model.
- Is based on an actual bit stream, allowing other device features to be modeled, including:
  - Clock recovery
  - Adaptive selection of Decision Feedback Equalizer (DFE) tap weights
  - TX/RX Co-Optimization techniques (Such as back channel)

A common source of confusion is the use of impulse response-based models in time-domain simulations. This occurs because TX models often lack support for bit-by-bit simulations. When this occurs, the EDA tool can use the model’s impulse response in a time-domain simulation through convolution of the impulse response with an input bit stream to generate the desired result.
2 IBIS-AMI Models and Simulation Flows

Let’s consider the important characteristics of AMI models and how these characteristics influence the flow of a simulation.

2.1 IBIS-AMI Models

According to the IBIS-AMI specification [2], “The executable model file of a Serializer-Deserializer (SERDES) transmitter or receiver contains up to three functions:

- **AMI_Init**
  - Required by all algorithmic models
    - The EDA tool **must** call the model’s AMI_Init function regardless of the type of simulation being performed.
    - Model control settings defined by the user are passed to the model during the AMI_Init call
    - Impulse response data is also passed to the model
  - Models static (does not vary from bit to bit) equalization. This is optional (though highly desirable) – models are not required to provide equalization through AMI_Init
  - When models support equalization in AMI_Init, they accepts an impulse response from the EDA tool and return the convolution of that impulse response with the impulse response of the device's equalization

- **AMI_GetWave**
  - Optional model functionality not required by the IBIS spec, but highly desirable for RX models
  - Models dynamic (varies from bit to bit) behavior
  - Accepts an input waveform stream which is modified based on the device’s equalization features before being returned to the EDA tool
  - Models the equalization, clock recovery, and adaptive control loop(s) of the device

- **AMI_Close**
  - Required by all models
As part of the IBIS-AMI specification, each model describes its capabilities to the EDA tool using an associated .ami file. Two required boolean parameters in the Reserved_Parameters section tell the simulator how the model performs equalization:

- **Init_Returns_Impulse**
  - If Init_Returns_Impulse is set to True
    - The model supports statistical simulation
    - The model can also support static (equalization does not vary from bit to bit) time-domain simulation
  - If Init_Returns_Impulse is set to False
    - The model does not support statistical simulation
    - AMI_Init only passes data and control settings into the model
    - Getwave_Exists must be set to True

- **GetWave_Exists**
  - If GetWave_Exists is set to True
    - The model supports time-domain simulation with dynamic (varies from bit to bit) equalization behavior
    - The model may also return clock ticks
    - If the model also returns a modified impulse response, it will be ignored for time-domain simulation
  - When GetWave_Exists is set to False
    - Init_Returns_Impulse must be set to True
    - The model supports time-domain simulation using static equalization returned during AMI_Init
    - The model cannot return clock ticks during time-domain processing. The simulator must use an internal CDR and budget data from the model to create a non-ideal clock distribution.

The three valid combinations of these two parameters are shown below. The simulation flow is determined by the particular combination of TX and RX models.

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Init_Returns_Impulse</th>
<th>GetWave_Exists</th>
<th>Simulation Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init-Only</td>
<td>TRUE</td>
<td>FALSE</td>
<td>Statistical &amp; static EQ time-domain</td>
</tr>
<tr>
<td>GetWave-Only</td>
<td>FALSE</td>
<td>TRUE</td>
<td>Dynamic EQ time-domain only</td>
</tr>
<tr>
<td>Dual</td>
<td>TRUE</td>
<td>TRUE</td>
<td>Statistical &amp; Dynamic EQ time-domain</td>
</tr>
</tbody>
</table>

Table 1 IBIS-AMI Model Types
2.2 Simulation Flows

It is important to note that there is no required relationship between TX and RX model types. For example, if the TX model is Init-Only, there is no requirement that the RX model also be Init-Only. The EDA tool must support all combinations of model types, although we'll see that for some combinations, the results may not be particularly meaningful. Since the TX and RX models are completely independent, there are nine possible combinations of model types for a single serial link. Each combination is described on the following pages. The figures depict the flow through each type of simulation and use the following definitions:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>hAC(t)</td>
<td>Impulse response of analog channel</td>
</tr>
<tr>
<td>hTE(t)</td>
<td>Impulse response of TX equalization</td>
</tr>
<tr>
<td>hRE(t)</td>
<td>Impulse response of RX equalization</td>
</tr>
<tr>
<td>gTE[t]</td>
<td>GetWave-based TX equalization</td>
</tr>
<tr>
<td>gRE[t]</td>
<td>GetWave-based RX equalization</td>
</tr>
</tbody>
</table>

Table 2 Simulation Flow Terminology

- **hAC(t)**
  - Refers to the impulse response of the analog channel and is the product of the network characterization phase of a simulation.
  - Network characterization is required for both statistical and time-domain simulations.

- **hTE(t)**
  - LTI form of TX equalization provided by AMI_Init
  - Static equalization: LTI equalization does not vary from bit to bit in a time-domain simulation

- **hRE(t)**
  - LTI form of RX equalization provided by AMI_Init
  - Static equalization: LTI equalization does not vary from bit to bit in a time-domain simulation

- **gTE[t]**
  - Time-domain, possibly non-LTI, representation of TX equalization provided by AMI_Getwave
  - Dynamic equalization: varies from bit to bit in a time-domain simulation
  - Can model adaptive behavior

- **gRE[t]**
  - Time-domain, possibly non-LTI, representation of RX equalization provided by AMI_Getwave
  - Dynamic equalization: varies from bit to bit in a time-domain simulation
  - Can model adaptive EQ and clock recovery behavior

For clarity, the following figures show the flow of statistical and time-domain simulations as separate paths.
2.2.1 Case 1: TX = Init-Only, RX = Init-Only

- Model Function
  - TX Model: Init_Returns_Impulse = True, GetWave_Exists = False
  - RX Model: Init_Returns_Impulse = True, GetWave_Exists = False

- Results
  - Statistical eye diagrams will include static TX & RX equalization.
  - Time-domain eye diagrams will include static TX & RX equalization.

- General comments
  - The eye diagrams from the two flows should compare well, since the same equalization is used for each. The statistical eyes will typically be smaller since they represent many more bits than the time-domain eyes.
  - Statistical simulation eyes being smaller than time-domain eyes is a general trend that holds across all 9 combinations of TX & RX models.

Figure 3
2.2.2 Case 2: TX = Init-Only, RX = GetWave-Only

- Model Function
  - TX Model: Init_Returns_Impulse = True, GetWave_Exists = False
  - RX Model: Init_Returns_Impulse = False, GetWave_Exists = True

- Results
  - Statistical simulations will only include static TX equalization.
    - No RX equalization will be modeled.
  - Time-domain simulations will include static TX and dynamic RX equalization.

- General comments
  - The eye diagrams from the two flows should be very different.
  - This combination is best suited for Time-domain simulation only.
    - Eye diagrams from statistical simulation should be considered as having probed the Rx die pad, since no RX EQ is present.

![Diagram](image-url)
2.2.3 Case 3: TX = Init-Only, RX = Dual

- **Model Function**
  - TX Model: Init_Returns_Impulse = True,   GetWave_Exists = False
  - RX Model: Init_Returns_Impulse = True,   GetWave_Exists = True

- **Results**
  - Statistical simulations will include static TX & RX equalization.
  - Time-domain simulations will include static TX and dynamic RX equalization.

- **General Comments**
  - This combination of models is suitable for both statistical and time-domain simulation.
  - The eye diagrams of the two flows may be different depending upon how well the static RX equalization compares to its dynamic counterpart.
  - The EDA tool must ensure that the RX equalization is not applied twice in a time-domain simulation. This is conceivable because the EDA tool is required to call the RX AMI_Init function, which returns an impulse response reflecting the RX equalization, and the equalization is applied again by the AMI_GetWave function.
2.2.4 Case 4: TX = GetWave_Only, RX = Init-Only

- Model Function
  - TX Model: Init_Returns_Impulse = False, GetWave_Exists = True
  - RX Model: Init_Returns_Impulse = True, GetWave_Exists = False

- Results
  - Statistical simulations will only include static RX equalization.
    - No TX equalization will be modeled.
  - Time-domain simulations will include dynamic TX and static RX equalization.

- General Comments
  - This combination of models is suitable for time-domain simulation only.
    - Statistical eye diagrams will not be useful, since there is no point in the physical circuit that has RX EQ but no TX EQ.

Figure 6
2.2.5 Case 5: TX = GetWave-Only, RX = GetWave-Only

- Model Function
  - TX Model: Init_Returns_Impulse = False, GetWave_Exists = True
  - RX Model: Init_Returns_Impulse = False, GetWave_Exists = True

- Results
  - Statistical simulations will include no equalization.
    - No TX or RX equalization will be modeled.
  - Time-domain simulations will include dynamic TX & RX equalization.

- General Comments
  - This combination of models is suitable for time-domain simulation only.
    - Statistical eye diagrams will not be useful, as no TX or RX EQ will be present.

![Diagram](Figure 7)
2.2.6 Case 6: TX = GetWave-Only, RX = Dual

- Model Function
  - TX Model: Init_Returns_Impulse = False, GetWave_Exists = True
  - RX Model: Init_Returns_Impulse = True, GetWave_Exists = True

- Results
  - Statistical simulations will include static RX equalization.
    - No TX equalization will be modeled.
  - Time-domain simulations will include dynamic TX & RX equalization.

- General Comments
  - This combination of models is suitable for Time-domain simulation only
    - Statistical Eye diagrams will not be useful, since there is no point in the physical circuit that has RX EQ but no TX EQ
  - The EDA tool must ensure that the RX equalization is only applied by the AML_GetWave function in time-domain simulation.

![Diagram](image)

**Figure 8**
2.2.7 Case 7: TX = Dual, RX = Init-Only

- Model Function
  - TX Model: Init_Returns_Impulse = True, GetWave_Exists = True
  - RX Model: Init_Returns_Impulse = True, GetWave_Exists = False

- Results
  - Statistical simulations will include static TX & RX equalization.
  - Time-domain simulations will use dynamic TX and static RX equalization.

- General Comments
  - This combination of models is suitable for both statistical and time-domain simulation.

- Special Considerations
  - If the RX AMI_Init function includes an optimization algorithm, the impulse response presented to the RX AMI_Init function must include the TX equalization. This is a problem for a time-domain simulation since this can lead to a double-counting of the TX equalization.
  - The IBIS specification suggests that a possible solution to this problem is an extra processing step involving deconvolution to obtain the impulse response of the RX equalization in isolation. This is then convolved with the impulse response of the active channel and the result is used in a standard GetWave-based simulation.
  - EDA tool users should be aware that deconvolution can be numerically unstable and caution is warranted if the EDA tool implements this extra step. The reliability of the results will be based on the numerical craftsmanship of the EDA tool.
2.2.8 Case 8: TX = Dual, RX = GetWave-Only

- Model Function
  - TX Model: Init_Returns_Impulse = True, GetWave_Exists = True
  - RX Model: Init_Returns_Impulse = False, GetWave_Exists = True

- Results
  - Statistical simulations will include static TX equalization.
    - No RX equalization will be modeled.
  - Time-domain simulations will include dynamic TX & RX equalization.

- General Comments
  - This combination of models is suitable for time-domain simulation only.
    - The eye diagrams from statistical simulation should be considered as having probed the Rx die pad, since no RX EQ is present.

![Diagram](image-url)  
Figure 10
2.2.9 Case 9: TX = Dual, RX = Dual

- Model Function
  - TX Model: Init_Returns_Impulse = True, GetWave_Exists = True
  - RX Model: Init_Returns_Impulse = True, GetWave_Exists = True

- Results
  - Statistical simulations will include static TX & RX equalization.
  - Time-domain simulations will use dynamic TX & RX equalization.

- General Comments
  - This is the best case, providing models suitable for both statistical and fully dynamic time-domain simulation.
  - The EDA tool must ensure there is no double-counting of equalization during a time-domain simulation.
  - The eye diagrams of the two flows may be different depending upon how static TX & RX equalization compares to their dynamic counterparts.

![Figure 11](image-url)
2.3 Simulation Flow Summary

The preceding discussions have illustrated the nuances of the different combinations of TX and RX models. The main elements of the nine different flows can be consolidated into the single diagram Figure 12 in order to highlight the following points:

- The statistical flow consists of convolution of the impulse response of the analog channel with the impulse response of the TX equalization and then convolution of this result with the impulse response of the RX equalization.
  - When either the TX equalization or RX equalization are not part of the model, the usefulness of the statistical flow is compromised.

- The time-domain flow consists of application of the TX AMI_GetWave function, followed by a convolution step, followed by application of the RX AMI_GetWave function.
  - The impulse response used in the convolution step depends on the particular combination of TX and RX models.
    - For example, in a time-domain simulation of Case 2, the impulse response corresponding to steps 2 or 3 of the statistical flow is used to represent the TX equalization.

- Case 7 shows one implementation of the extra step required for deconvolution where the Discrete Fourier Transform (DFT) of the impulse response from step 3 of the statistical flow is divided by the DFT of the impulse response from step 2 to give the DFT of the RX equalization only. An inverse DFT transforms the frequency-domain representation of the equalization to an impulse response in the time-domain, to be convolved with the impulse response of the analog channel. This approach is based on the principle that deconvolution in the time-domain is equivalent to division in the frequency domain.
2.4 Simulation Flow Recommendations

After reviewing the simulation flows and their summary as found in Figure 12, the following model combinations provide the best opportunity for rigorous system-level simulation.

- **Dual TX & Dual RX (Case 9)**
  - Supports statistical and fully dynamic time-domain simulations
  - Ideal case that takes maximum advantage of each simulation flow
- **Init-only TX & Dual RX (Case 3)**
  - Supports statistical and static TX / dynamic RX time-domain simulations
  - Most TX devices are modeled acceptably with a static model
  - Second best choice – provides results equivalent to case 9 when TX is adequately represented as a static model
- **Dual TX and Init-only RX (Case 7)**
  - Supports statistical and dynamic TX / static RX time-domain simulations
  - Most RX devices have nonlinear and adaptive equalization capabilities. These behaviors can be approximated with an Init-only model, but only static equalization can be represented. The quality of the prediction is dependent on the numerical craftsmanship in the model algorithms.
- **Init-only TX and Init-only RX (Case 1)**
  - Supports statistical and static TX & RX time-domain simulations
  - Most TX models are modeled acceptably with a static model
  - Most RX devices have nonlinear and adaptive equalization capabilities. These behaviors can be approximated with an Init-only model, but only static equalization can be represented. The quality of the prediction is dependent on the numerical craftsmanship in the model algorithms.
  - Provides results equivalent to case 7 when TX is adequately represented as a static model

Cases 9 and 3 provide the best model choices for both high-performance statistical simulations and accurate time-domain simulations.

Cases 2, 4, 5, 6 and 8 lack support for statistical simulation. Time-domain simulation is the only viable option for these cases even though it can be 100x slower and restrict the number of design simulations that can be performed.
3 Simulation Outputs

Earlier, we noted that the final result of either a statistical or time-domain simulation is an eye diagram. The intent of the first part of this section is to show how an eye diagram is used to generate other common means of evaluating a high-speed, serial channel, in particular, bathtub curves and a bit-error-rate (BER). We follow that with a description of incorporation of jitter and noise in a simulation.

3.1 Bathtub Curves and BER

Figure 3.1 shows a typical eye diagram with the probability distribution function (PDF) of the RX clock in blue and the bathtub curve in black. The red trace is the PDF of the time at which the edge crosses the RX threshold. The eye diagram and possibly the RX recovered clock PDF are the direct results of a simulation while the bathtub curve and final BER are calculated in a post-processing step.

- In a statistical flow, a user or model must explicitly specify the characteristics of the RX clock in terms of different types of jitter.
- A GetWave-based time-domain model can optionally return a vector of times at which the edges of the recovered clock occurred and the EDA tool can use these times to create the RX clock PDF. Otherwise, a user or model must specify the statistical characteristics of the clock PDF.

The bathtub curve represents the rate of errors as an ideal sampling clock is swept across a unit interval (UI). If we think of a virtual BER tester moving the sampling point towards the left edge of the eye, the bathtub curve shows the cumulative probability of an edge occurring to the right of the sampling point, i.e. a bit error. As the sampling point moves farther to the left the probability of error grows. In other words, the bathtub curve is just an integration of the edge PDF from the center of the eye to the sampling point.
The bathtub curve can be combined with the RX clock PDF to produce a final BER, as shown below. The aggregate BER is based on the product of the probability of error with a particular clock position and the probability of that clock position occurring and these values are summed for all possible clock positions.

![Figure 14 Ber Calculation](image)

The BER calculation applies to both statistical and time-domain simulations with the difference that the time-domain calculation is based upon the number of bits simulated and the statistical calculation assumes linearity and a large sample of bits. Using extrapolation, a time-domain simulation can predict BERs lower than the number of bits simulated.
4 Jitter and Noise

Jitter of different types (random, deterministic, sinusoidal, DCD) can be included in a simulation at both the TX and RX devices by considering the effect of the jitter on the relevant clock edges. These are “virtual” clocks in the sense that a simulation doesn't produce physical clock signals that can be plotted; we're only interested in the clock edges since TX clock jitter directly translates to TX data jitter and RX clock jitter directly translates to data sampling (recovered clock) jitter.

In a time-domain simulation, the position of each TX clock edge is varied according to the specified jitter types and characteristics. Random jitter (RJ) is assumed to have a Gaussian distribution, deterministic jitter (DJ) is assumed to have a flat PDF between the minimum and maximum values, and duty cycle distortion (DCD) is divided between successive clock edges. Sinusoidal jitter (SJ) is specified with a magnitude and frequency. The total jitter applied to an edge is the sum of the individual contributions. In a statistical simulation, the individual jitter PDFs are convolved with the unjittered RX eye to produce the final eye diagram.

Jitter contributed by the RX device is modeled in a time-domain simulation by varying the clock_times returned by the RX AMI_GetWave function. Values of random jitter, sinusoidal jitter, and duty cycle distortion are added to each clock_time and the final clock PDF is applied to the data eye according to the simulator's post-processing method. If the AMI_GetWave function doesn't exist or doesn't return clock_times, the RX_Clock_Recovery parameters must be used to model the clock's statistical characteristics including its mean, the standard deviation of any random jitter, the magnitude of sinusoidal jitter, and the magnitude of duty cycle distortion. In a statistical simulation, the clock recovery parameters are always used to represent the RX clock PDF.

Gaussian noise at the input to a receiver's sampling latch is modeled with the Rx_Noise parameter. The noise is directly added to the incoming signal in a time-domain simulation and the noise PDF is convolved with the data eye in the statistical flow.
5 Summary

The consistent theme throughout this paper has been that there are three classes of IBIS-AMI algorithmic models with fundamentally different capabilities and characteristics. As a result, a user must understand the nature of the models available for a simulation and confirm that their combination is capable of achieving the desired goal. This task may seem daunting, particularly because of the large number of valid model combinations. Hopefully, this is no longer the case and new users can now enjoy the distinct advantages of the IBIS-AMI approach to serial channel simulation.

6 References