SerDesDesign.com
IBIS-AMI Modeling Flow
2020/06/09

By John Baprawski

johnb@serdesdesign.com

Web Site

www.serdesdesign.com

IBIS-AMI model development challenges

- IBIS-AMI models and Channel Simulators have become an integral part of SerDes system design.
- SerDes vendors routinely provide IBIS-AMI models for their chips; often before the final chips.
- End users depend on high fidelity IBIS-AMI models and an accurate channel simulator to be able to predict their system behavior in their simulations.
- IBIS-AMI models are highly configurable.
- Model developers can design their models so that the equalization settings, adaptation loop parameters and corner cases can be chosen by the user.
- Key objective: IBIS-AMI models to accurately match the hardware performance in a real system.
- Therein lies the challenge.

IBIS-AMI Modeling Flow - Assumptions

- It is presumed that the customer has their target SerDes IC Tx and Rx circuit designs defined in Spice simulations and may or may not have algorithms defined (in MATLAB or C/C++) for parts of their IC circuit designs.
- IBIS AMI modeling is based on converting Spice circuit models into IBIS AMI models. The typical approach is discussed in the following pages.
- The IBIS-AMI model developer must have a suitable Channel Simulator available.
- The SerDesDesign.com Channel Simulator web based and free.
    - https://www.serdesdesign.com
    - Premium features are available for a low monthly fee.

IBIS-AMI Modeling – Key Decisions

- Key decisions are required to convert Tx/Rx circuits into their IBIS-AMI representations.
- One can split the decisions as follows:

1) Partition the Tx/Rx design into a **signal flow** suitable for IBIS-AMI model representation.
   - This can be quite a challenging task and requires iterative discussion between the model developer and circuit developer to achieve a common understanding.
2) Provide the IBIS-AMI models with parameters that represent the Tx/Rx circuit **control flow**.
   - This task is to define all the states of the circuit to be represented in the model and define a set of parameters associated with each state.
   - Each state is represented with LTI and/or NLTV functionality by the model developer.

- The above decision process is not discussed in detail here.
- Tx/Rx IBIS buffer designs are assumed to properly represent the impedance loading the channel.

Define Operating Conditions

- Define the set of operating conditions for your Spice circuit which is to be converted into an AMI model.
    - For Tx: gain values, pole values, FFE tap values, PVTs, etc.
    - For Rx: include CTLE, CDR and DFE characteristics.
    - Define a set of channels to be used (S4P files)
    - Define Tx and Rx packaging characteristics (S4P files)
    - Define Tx and Rx IBIS buffer characteristics (IBIS RLC circuit or IBIS S4P)
- For example, some customers have required that an Rx AMI model have a CTLE with over 1000 responses selectable by a set of parameter values, either statically before the channel data is processed, or dynamically set while the channel waveform is processed with a feedback algorithm in the Rx model.

Partition and Extract Data

- For the AMI model, define the input and output port names and signal types (int, float, etc.), and parameter properties (names, types, descriptions, etc.).

- Partition the Spice circuit into a Spice signal flow block diagram of LTI and NLTV sections with Tx output and Rx input analog buffer behavior defined.

- The analog buffer behavior is to be modeled in the IBIS analog portion.

- The LTI and NLTV sections are to be modeled in the IBIS AMI portion.

- Extract data from the Spice circuit in the time (preferred) or frequency domains that represent the behavior of individual blocks in the Spice model block diagram.

Model Development

- If modifying an existing IBIS-AMI model provided by SerDesDesign.com that included Training, then modify the pre-defined AMI_Solution and process.

- If creating a new model, use the SerDesDesign.com IBIS-AMI Modeling Kit (see separate presentation)

- Iterate as needed based on validation tests.

- The completed IBIS-AMI model is a portable C/C++ dll (*.so on Linux) that conforms to the IBIS standard.

- There is no need for the IBIS-AMI model developer to be an expert on the IBIS standard, C/C++ coding, or in setting up the compile and linking rules in the Microsoft Visual Studio or Linux tools.

Test the Model

- Set up the Channel simulation test cases for the defined operating conditions.

- Test the IBIS-AMI models in the Channel Simulator.

- Verify against customer supplied reference test simulations from Spice, MATLAB, Simulink, etc.

- Revise the IBIS-AMI model as needed to get the level of fidelity desired.

Optimization and Validation

- For the AMI model:
    - An FFE, CTLE, DFE or Gain model can automatically initialize its taps (gains) based on the channel characteristics.
    - A CTLE, DFE or Gain model can dynamically change its taps (gains) based on waveform feedback as the simulation proceeds.
- Using IBIS-AMI model in the Channel Simulator:
    - Channel simulation results can be compared to expected Spice simulation results and a validation Figure of Merit (FOM) can be derived.
    - The FOM is often based on similar waveforms or eye diagram plots.

IBIS-AMI Model Deliverables

- Every IBIS-AMI model has these deliverables to a user:
  - *.ibs file: Defines the IBIS buffer and references the AMI files.
  - *.ami files: Defines the parameters for the AMI model.
  - *_x64.dll file (Windows 64 bit): AMI model dynamic linked library.
  - *_x64.so file (Linux 64 bit): AMI model shared object.
    - Optional: *.dll for Windows 32 bit
  - Documentation for the IBIS-AMI model.
  - Optional: S4P files for the IBIS model.
  - Optional: S4P files for the packaging characteristics.

Tips for Good Quality AMI Models

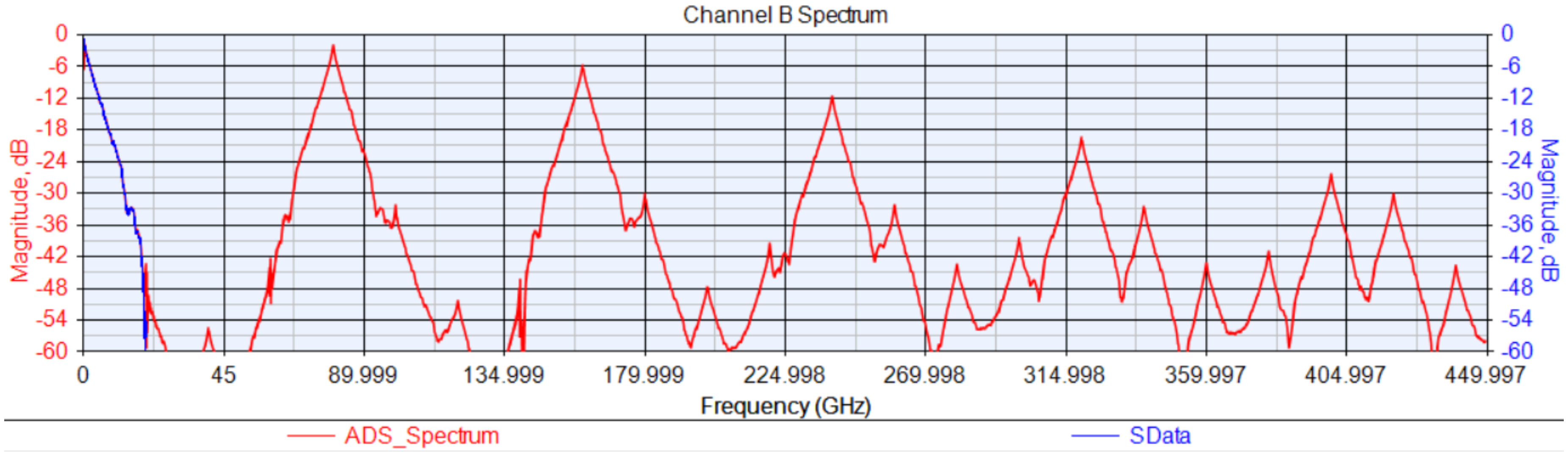
- Begin with an AMI model algorithm concept in mind that is either LTI or NLTV and implement it first with complexity removed and then build in the complexity.
- Design an AMI model with your customer in mind and define the AMI model name and the AMI parameters available at the Channel Simulator level that would be useful to your customer.

Aliasing in the ADS ChannelSim Impulse

- The ADS ChannelSim impulse often gives an accurate representation of the band-limited S-parameters, but may have errors or exhibit high frequency aliasing beyond the S-parameters maximum frequency.
- This may be an issue for some SerDes designs.

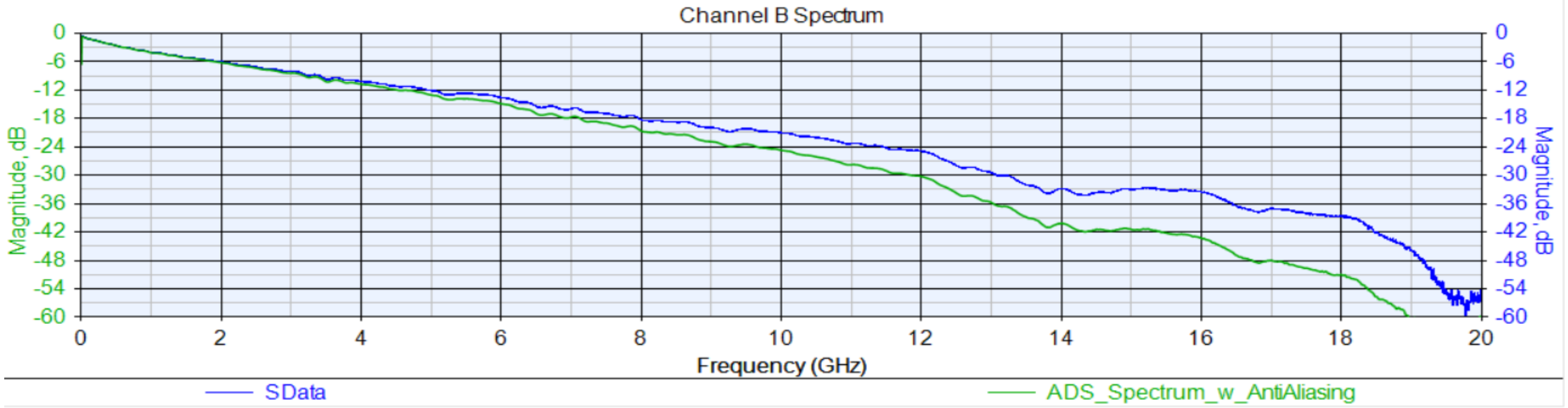
Aliasing in the ADS ChannelSim Impulse

- A typical channel (band limited at 20 GHz) using 28.125 Gbps with 33 dB loss at Nyquist.
- The ChannelSim impulse file was generated: imp_rx_init_out.txt.
- The ADS impulse data was evaluated in time and frequency domains.
- Observe the the S-parameter channel data (blue) and the ADS impulse spectrum (red).
    - The ADS impulse is in red and has high frequency aliasing



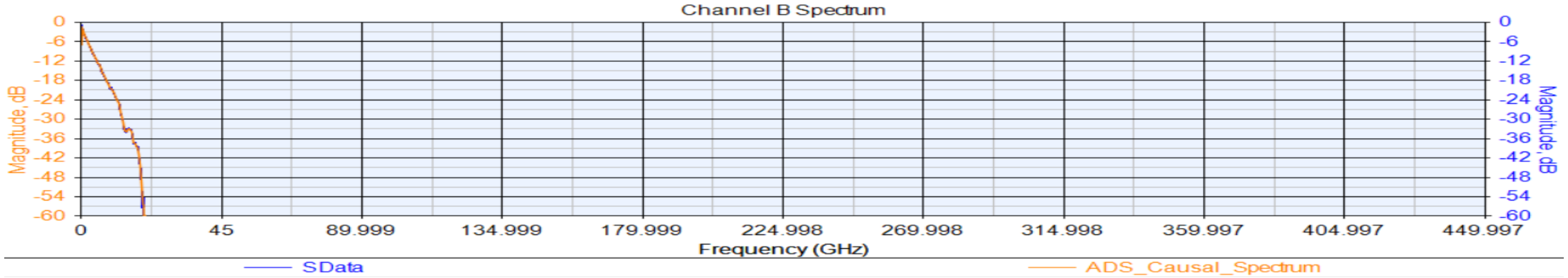
Removing Aliasing in the ADS Impulse

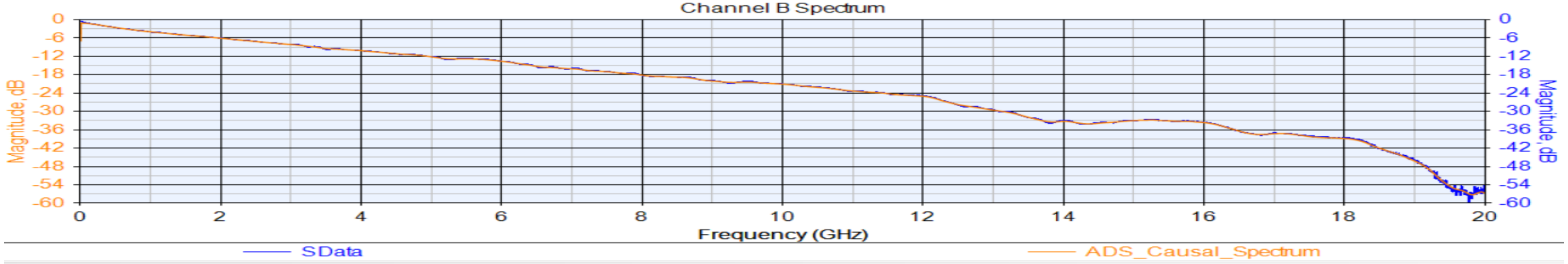
- The ADS ChannelSim can remove the high frequency aliasing.
- Set ‘Anti-aliasing window size in time point’ in the Convolution > Advanced… tab
  - Fc = half width of the filter main lobe = BitRate in this example.
  - M = 2*SamplesPerBit*BitRate/Fc = 64 in this example.
- High frequency is substantially reduced, but at the cost of roll off at Nyquist (14.06125 GHz).
- Compare the original S-Data (blue) and the ADS impulse with anit-aliasing (green).



Another Approach to Improve the ADS Impulse

- Convert the band-limited S-Parameters to Causal S-Parameters
  - Use SerDesDesign.com tool: https://www.serdesdesign.com/home/generate-causal-sparameters/
  - Compare the original S-Data (blue) to the ADS impulse with causal S-parameters (orange).
  - The impulse response using causal S-Parameters tracks the S-Data and has no high frequency aliasing.





ADS ChannelSim Issues

- Changing *.dll, *.ibs, *.ami files
- Problem: Once a ChannelSim schematic is set up with Tx_AMI and Rx_AMI models, changes to any of the associated *.dll, *.ibs or *.ami files are not recognized by the ChannelSim.
- Workaround:
  - For dll file changes: ‘Stop and Release’ the simulator and then copy the new dll to the workspace data directory. The workspace can also be closed and then reopened.
  - For *.ami or *.ibs file changes: Either select another *.ibs file or delete and replace the AMI model.   If the Tx_AMI model is deleted and replaced be sure to reset the Tx_AMI > PRBS > Bit rate.

ADS ChannelSim Issues

- Eye_Probe Density plot fails
- Problem: Sometimes ADS Channel Simulation with the Eye_Probe will display the Eye with poor resolution or wrong eye plot.
- Reason: The Waveform level into the Eye_Probe must be scaled properly in the Eye_Probe if not in the range 0.25 to 4 V peak-to-peak.
- Work around: Set the Eye_Probe 'Amplitude resolution' property
    - On the Eye_Probe Parameters tab, select 'More…', in the 'Histogram Resolution' area:
        - Set the 'Amplitude resolution' value larger if the Waveform peak-to-peak level is larger than 4 v
        - Set the 'Amplitude resolution' value smaller if the Waveform peak-to-peak level is smaller than 0.25 v.

Issues in any Channel Simulator

- Corner Case conditions are different in *.ibs and *.ami files.
- Issue: This is per the IBIS-AMI standard.
    - The *.ibs file corner cases are typical, min and max.
    - The *.ami file corner cases are typical, slow and fast.
- Work around:
    - There is no simple mapping between these two sets of corner cases.
    - The usual mapping is to make these associations:
        - IBIS Typical = AMI Typical
        - IBIS Min = AMI Slow
        - IBIS Max = AMI Fast
    - See detail discussion at this link: https://www.serdesdesign.com/home/web_documents/models/Using_CornerCases.pdf

Appendix

For Further Information

- Various part of the above processes can be achieved with more automation as desired.
- For information, see web site:
  - www.serdesdesign.com
- See Blog posts for various AMI modeling applications.
- This web site provides an easy to use free tools to analyze SerDes models and systems
  - https://www.serdesdesign.com/home/serdes-system-single-channel-tool
  - You must register to use this and other tools

Use Causal S-Parameters

- S-Parameter frequency domain data files are commonly used to define SerDes channel and Tx/Rx packages in a Channel Simulator.
  - S-parameters inherently are band-limited with discrete frequencies and often are non-causal, non-reciprocal, and non-passive.
- All SerDes system channel simulators convert the frequency domain S-parameter characteristics into time domain impulse characteristics.
  - All attempt to correct S-parameters distortions, some less successfully than others.
- The key to obtaining consistent S-Parameter use with any time domain based tool, including any channel simulator, is to convert S-Parameter data into what is called “Causal S-Parameters”.
  - Use the web site www.serdesdesign.com/home/generate-causal-sparameters
- Causal S-Parameters provide portable/accurate eye diagrams/BER plots in any Channel Simulator.
  - Example: https://www.serdesdesign.com/home/serdes-channel-impulse-modeling-with-rambus/

Model Development

- Use SerDesDesign.com for IBIS-AMI model development
- Construct the IBIS-AMI models in the SerDes Systems Tool
  - https://www.serdesdesign.com/home/serdes-system-single-channel-tool
  - Separate presentation available on request.
  - Iterate as needed to achieve desired performance.
- Generate the Tx or Rx IBIS-AMI model from the web site.
  - Cost per model can be as low ass $2K.
- There is no need for the IBIS-AMI model developer to be an expert on the IBIS standard, C/C++ coding, or in setting up the compile and linking rules in the Microsoft Visual Studio or Linux tools.