# End to end Link Analysis and Optimization with Mid-channel-redriver AMI Models

Mahbubul Bari, Maxim Integrated
Ron Olisar, Maxim Integrated
Hassan Rafat, Maxim Integrated
Fangyi Rao, Agilent Technologies
Sharon Wang, Maxim Integrated
Ming Yan, Agilent Technologies

IBIS Summit at DesignCon
Santa Clara, CA
January 31, 2013

# Outline

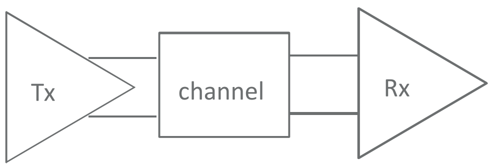
- Link Simulation with Mid-channel-redrivers
- Modeling Redrivers
- Correlation and Simulation Results
- Summary

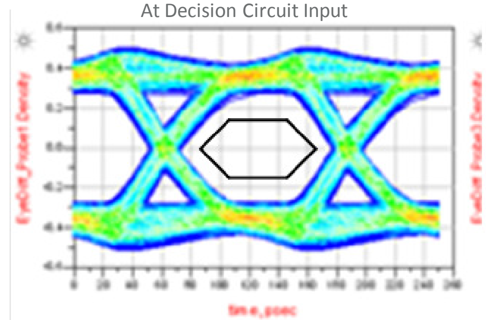
# Outline

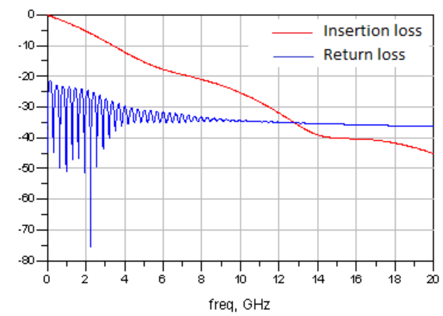
- **Link Simulation with Mid-channel-redrivers**
- Modeling Redrivers
- Correlation and Simulation Results
- Summary

# More Channels require Mid-channel-redrivers

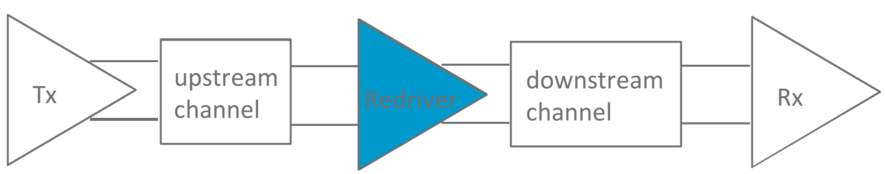
- Problem: Higher Data Rates increase Channel Loss, Reflections, Xtalk
  > Jitter Budget and Receiver Sensitivity are Challenged!




- One Option: Use Mid-channel-redrivers to improve Link Margins
  > Compensate channel loss to improve Jitter Budget
  > Restore signal level to satisfy Receiver Sensitivity



# Link Simulation with Mid-channel-redrivers

- Problem: Non-Linear Redrivers face a simulation roadblock:
  - > Channel simulators such as **STATEYE** & **IBIS-AMI** require linear time-invariant channels, hence non-linear mid-channel Redrivers are not permitted.
  - > **SPICE** simulation can include non-linear Redrivers. But, slow throughput!

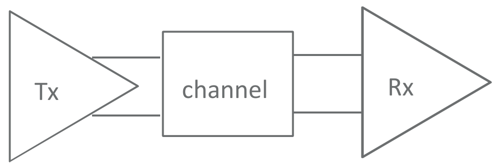
- Need: IBIS-AMI Flow with Mid-channel Non-linear Redrivers
  - > Need end-to-end IBIS-AMI simulation flow that incorporates Redriver models to accurately represent non-linear time-variant and noise behaviors.
  - > Must include all the benefits of IBIS-AMI:
    - High Throughput
    - Interoperability between IC Vendor models
    - Portability of models across EDA vendor tools
    - IP Protection
  - > The goal is improved accuracy of BER estimation and Link Margin prediction

# Outline

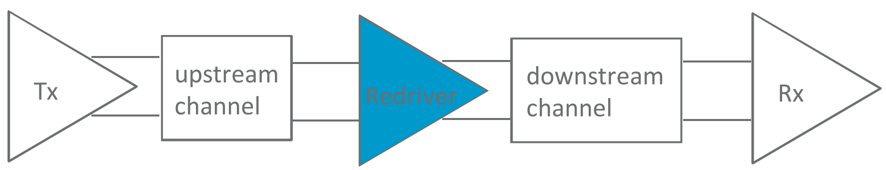
- Link Simulation with Mid-channel-redrivers
- **Modeling Redrivers**
- Correlation and Simulation Results
- Summary

# Redriver Background

Channel without Redriver

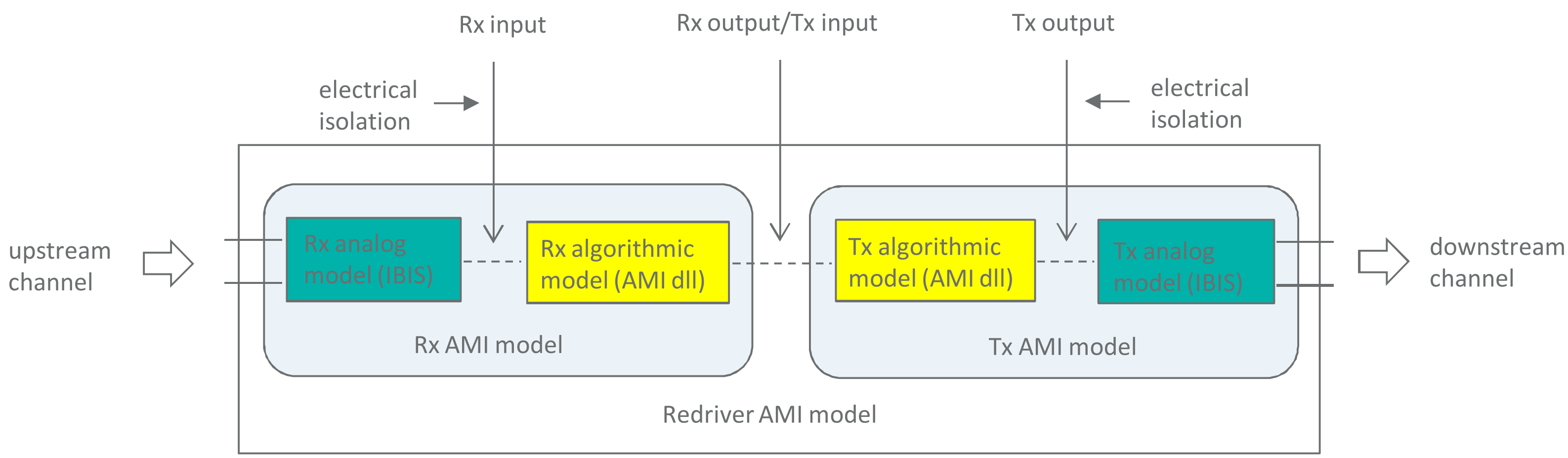



Channel with Redriver




- Redriver is placed in the middle of the channel to compensate channel loss to support high data rate
- It equalizes signals from the upstream channel and retransmits them into the downstream channel
- Redriver output is continuously driven by input. No retiming is performed when it retransmits signal
- The device can be nonlinear and noisy, thus breaks the linear channel assumption in AMI
- Need to extend current standard to include Redriver in AMI simulations

# Redriver AMI Model




- Redriver AMI model consists of two back-to-back regular AMI models that represent receiving and transmitting parts of the device.
- Each half model has its own .ami and .dll (or .so) files.
- Both file pairs are reference in the same .ibs file. The .ami and .dll files of the Rx part are specified under the [Algorithmic Model] keyword in the Rx model section of the .ibs file. The .ami and .dll files of the Tx part are specified under the [Algorithmic Model] keyword in the Tx model section.
- Signal flow is from Rx analog to Rx algorithmic to Tx algorithmic to Tx analog
- Rx analog model represents input termination.
- Looking from Rx analog, the Rx algorithmic block has infinite impedance.
- Tx analog represents output impedance.
- Looking from Tx analog, Tx algorithmic block is an ideal voltage source.

# Redriver AMI Model (Cont'd)

- Rx AMI_Init takes upstream channel impulse matrix as input. Tx AMI_Init takes downstream channel impulse matrix as input.
- Rx algorithmic model's output waveform is the input signal to the Tx algorithmic model.
- Tx analog is continuously driven by Tx algorithmic output waveform instead of digital trigger events as in a regular IBIS output model.
- Tx analog is expected to describe an analog circuit as oppose to the conventional D/A converter.
- If Rx DLL generates clock times, they will be ignored by simulator.
- Jitter parameters in a Redriver model are ignored. Device noise can be modeled in AMI_GetWave.
- Redriver can be cascaded in a channel.

# Example of Redriver IBIS File

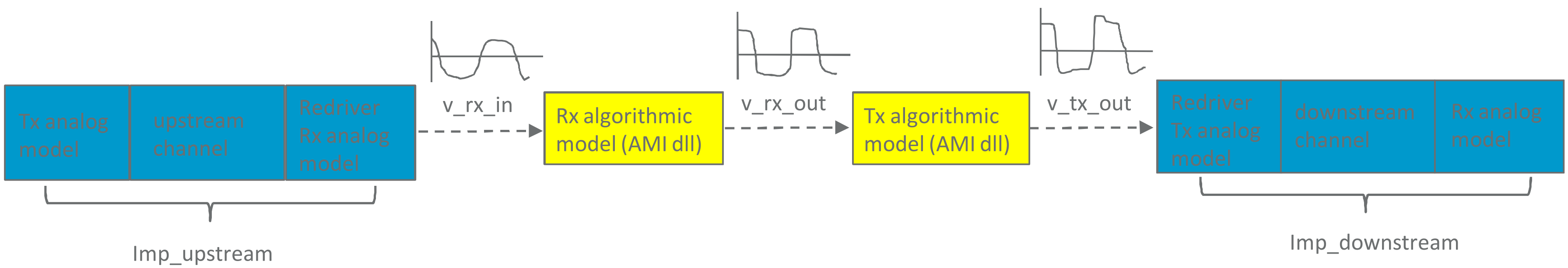
```
…
|***********************************
| Rx Model
|***********************************
[Model] max3997_rx
Model_type Terminator
…
[Algorithmic Model]
Executable Windows_VisualStudio_32      max3997_rx.dll      max3997_rx.ami
Executable linux_gcc_32      max3997_rx.so      max3997_rx.ami
[End Algorithmic Model]

…
[END]

|***********************************
| Tx Model
|***********************************
[Model] max3997_tx
Model_type Output
…
[Algorithmic Model]
Executable Windows_VisualStudio_32      max3997_tx.dll      max3997_tx.ami
Executable linux_gcc_32      max3997_tx.so      max3997_tx.ami
[End Algorithmic Model]

…
[END]
```

# Redriver Simulation Flow




1. v_rx_in = input_to_upstream_channel * Imp_upstream
2. Rx algorithmic model processes v_rx_in and returns v_rx_out
3. Tx algorithmic model processes v_rx_out and returns v_tx_out
4. output_of_downstream_channel = v_tx_out * Imp_downstream

# Redriver Model Example

- EXAMPLE:  10.7G dual-mode linear and limiting equalizer.
  > Allows for easy testing of both solutions and permits the optimum architecture to be chosen for a given topology
- Linear Mode
  > Intended for use with RX ASIC with decision-feedback equalizers (DFEs)

  > The linear Redriver contains gain controls that adjust the EQ peaking and flat gain response of the linear amplifier to compensate loss and maintain the signal in the linear range of the equalizer.
    - Preserves the linearity of the channel characteristics, allowing the DFE to operate on the entire linear channel.
    - Permits extending total channel reach and/or improving signal-to-noise ratio (SNR).
    - Placed in a long channel, this Redriver compensates up to 18dB at 5GHz of typical FR4 loss, thus reduces the effective channel length seen by the DFE receiver.

# Redriver Model Example (Cont'd)

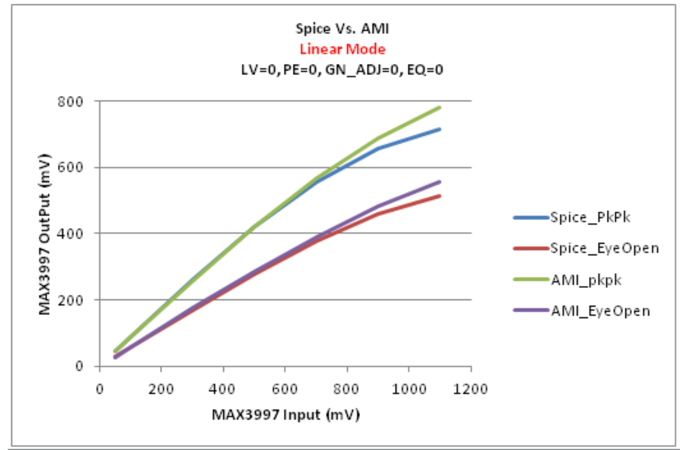
- Limiting Mode
  - > Redriver is intended to be used as a conventional equalizer with a limiting amplifier and Pre-Emphasis driver.
  - > Redriver is able to compensate up to 18dB at 5GHz of typical FR4 loss at the input and drive up to 12dB of typical FR4 loss at the output.

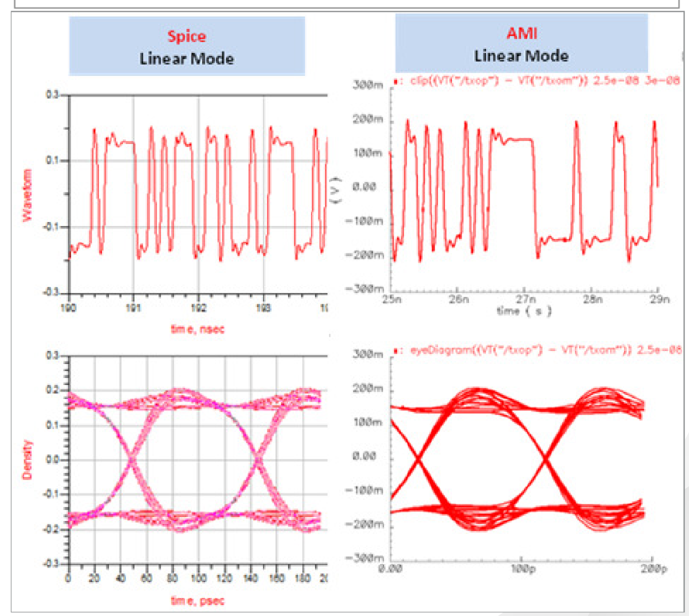
# Outline

- Link Simulation with Mid-channel-redrivers
- Modeling Redrivers
- **Correlation and Simulation Results**
- Summary

# Model Performance and Correlation

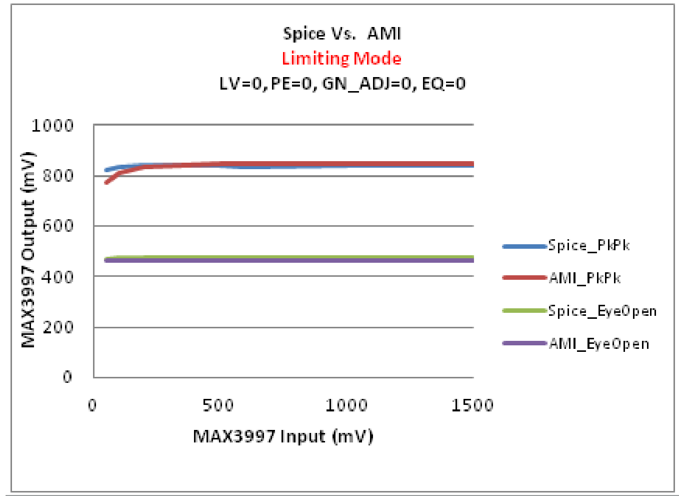
- Identical simulation topology is setup with the original netlist based spice simulation and a topology using IBIS-AMI models
- The discrepancy between IBIS-AMI and spice is less than 5% when the Redriver operates in the linear region
- As the input amplitude into the Redriver approaches the upper end of the operating range, it goes into nonlinear region and the discrepancy is greater than 5% currently.

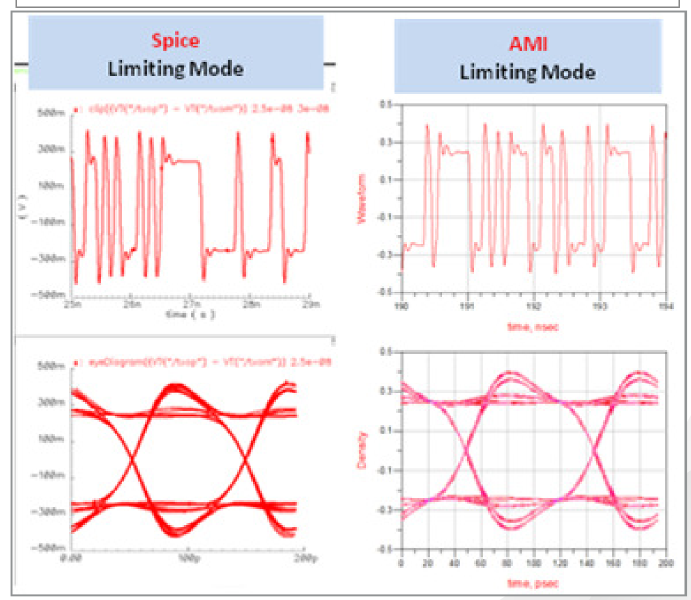


# Model Performance and Correlation (Cont'd)

- For the limiting mode of operation, the discrepancy between IBIS-AMI and spice is shown to be less than 5%.

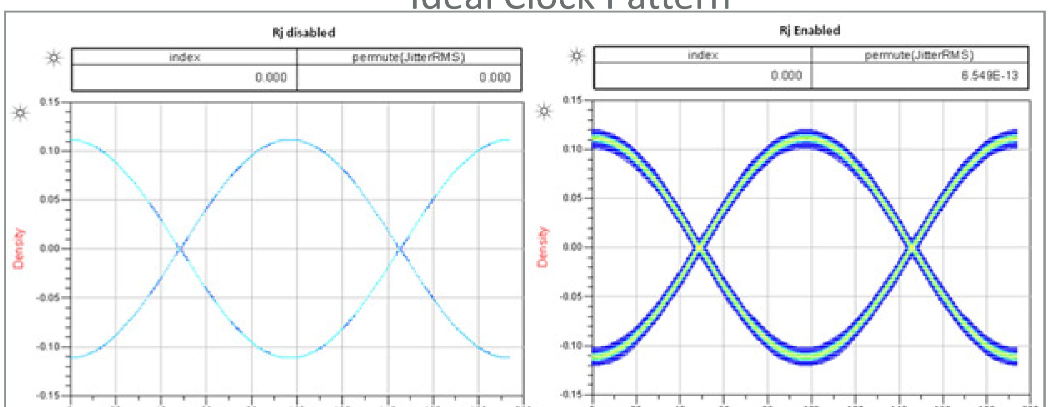


# Random Jitter

- Redriver Self-Generated Noise and Random Jitter
  - > Random jitter is implemented in the model using a Gaussian voltage noise sources before and after the non linear transfer functions

Ideal Clock Pattern

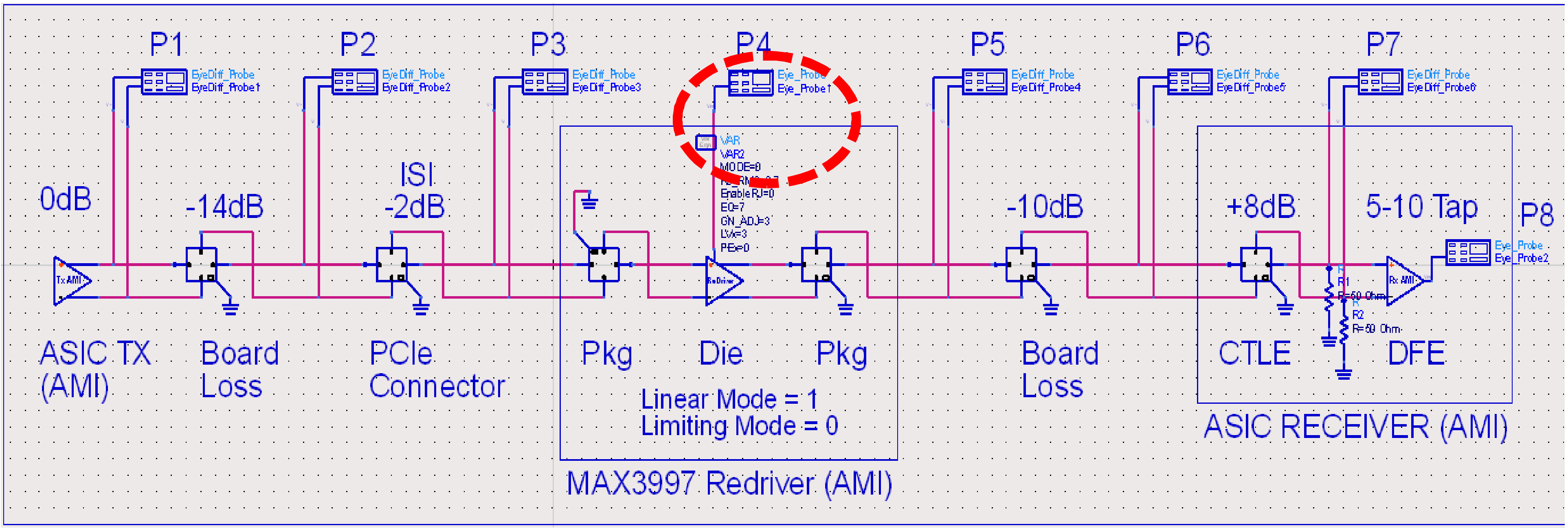



Redriver Rj Disabled

Redriver Rj Enabled

# Example of Channel Simulation with AMI Redriver Model

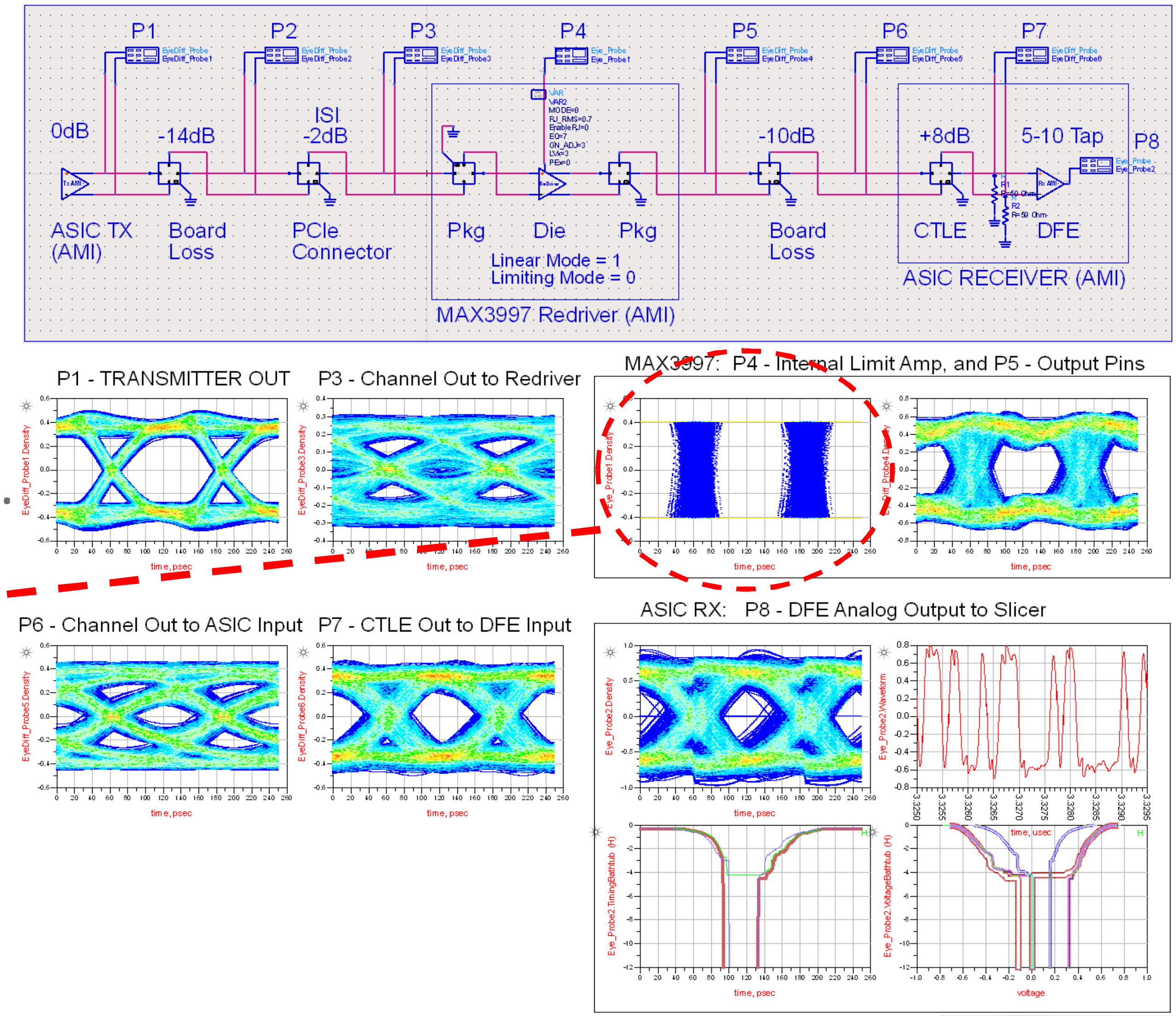
- Non-Linear "AMI" Redriver Model placed in middle of Channel




- Estimate margin to BER to Validate and Compare Redriver Options
- Note the Redriver internal node probe

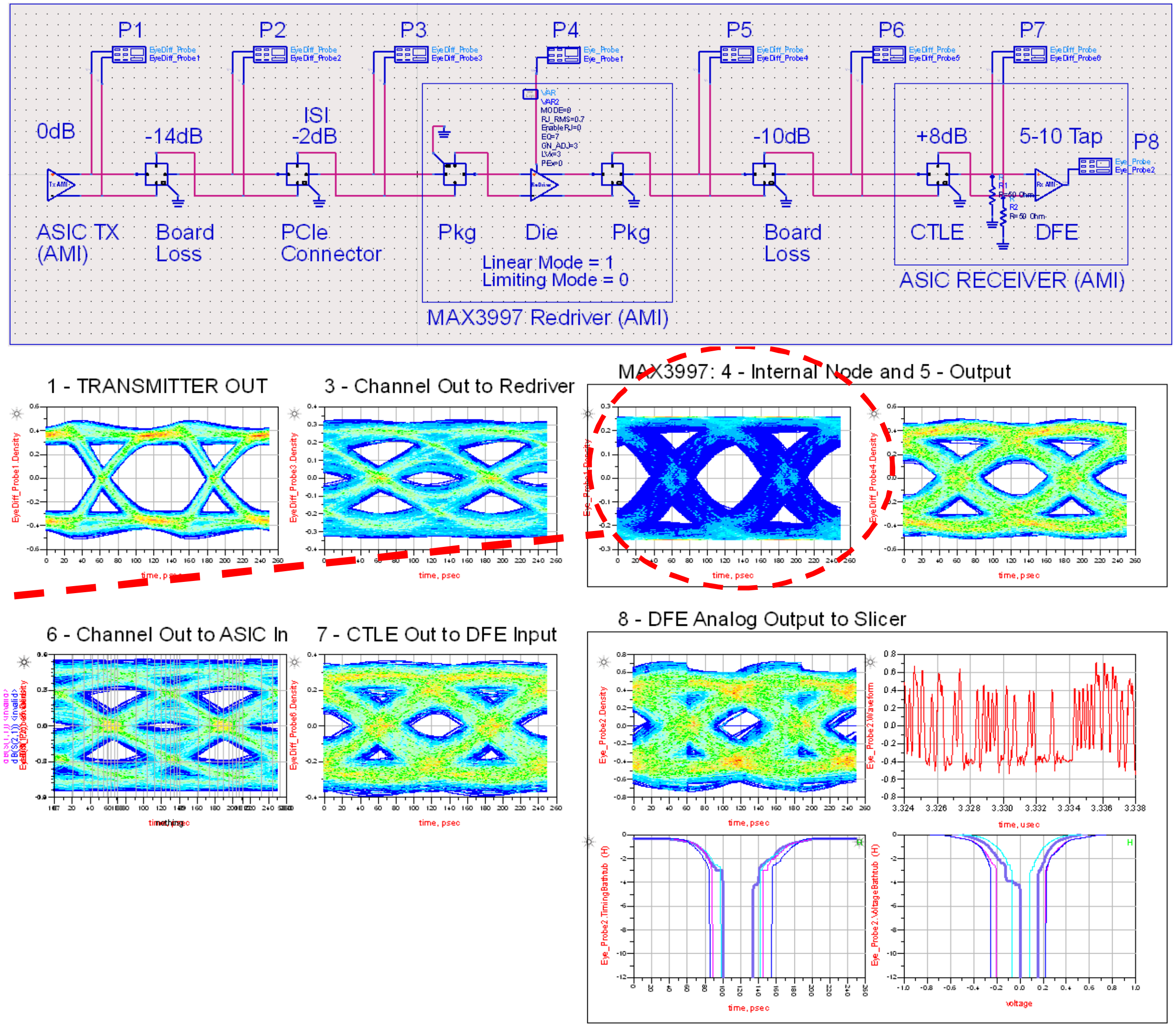
# Example of Channel Simulation with AMI Redriver Model

- Comparing Redriver options
- Simulation of Limiting EQ Redriver mode with 8Gbps Impaired PCIe Channel.
- Redriver internal node probe



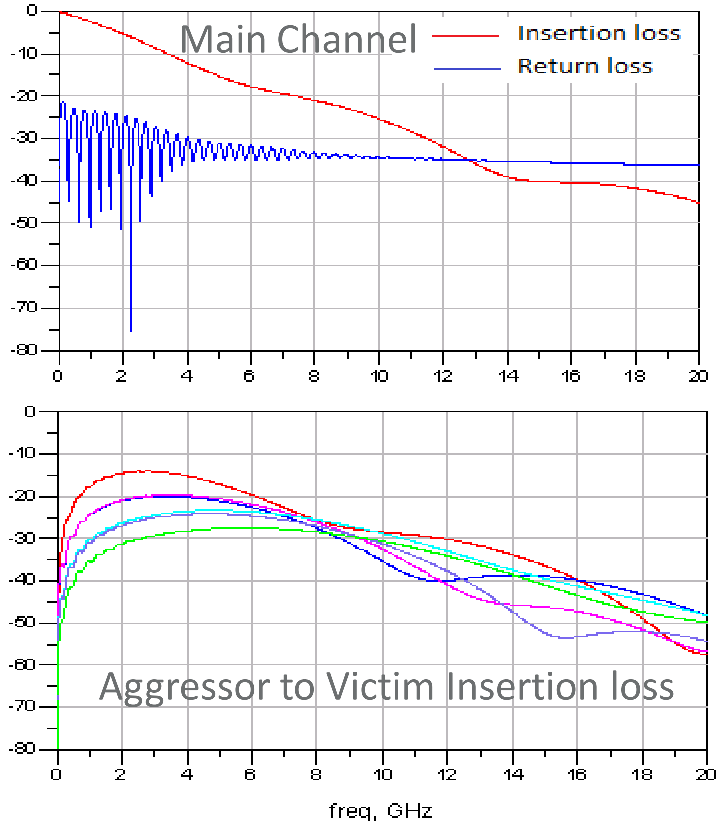
# Example of Channel Simulation with AMI Redriver Model

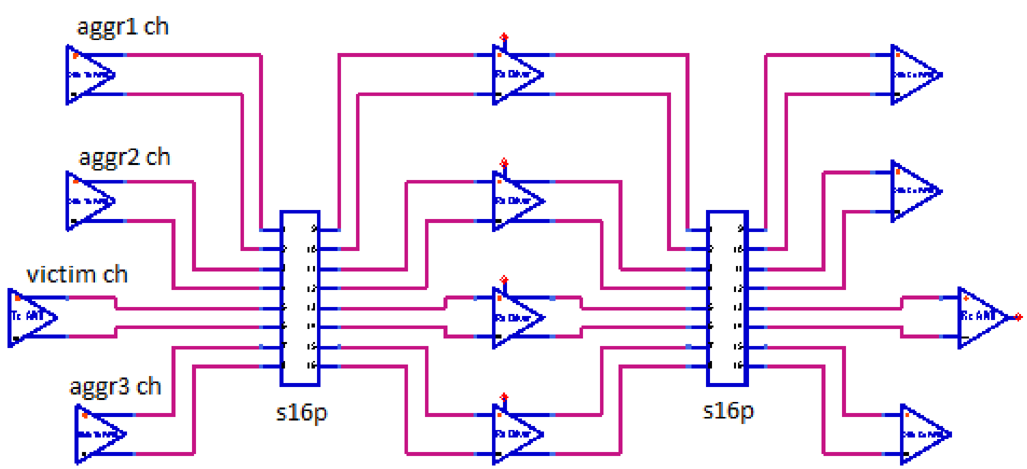
- Comparing Redriver options
- Simulation of <u>Linear</u> EQ Redriver mode with 8Gbps Impaired PCIe Channel.
- Redriver internal node probe



# Channel Crosstalk

- Two identical 16-port S-parameters represent upstream & downstream channels.
- Multiple reflections at each port are accounted for by the full 16-port S-matrix
- To deactivate an aggressor, we set its output amplitude to zero. Non-active aggressor channels stay in simulation so port terminations remain unchanged.




Agilent Technologies



maxim integrated

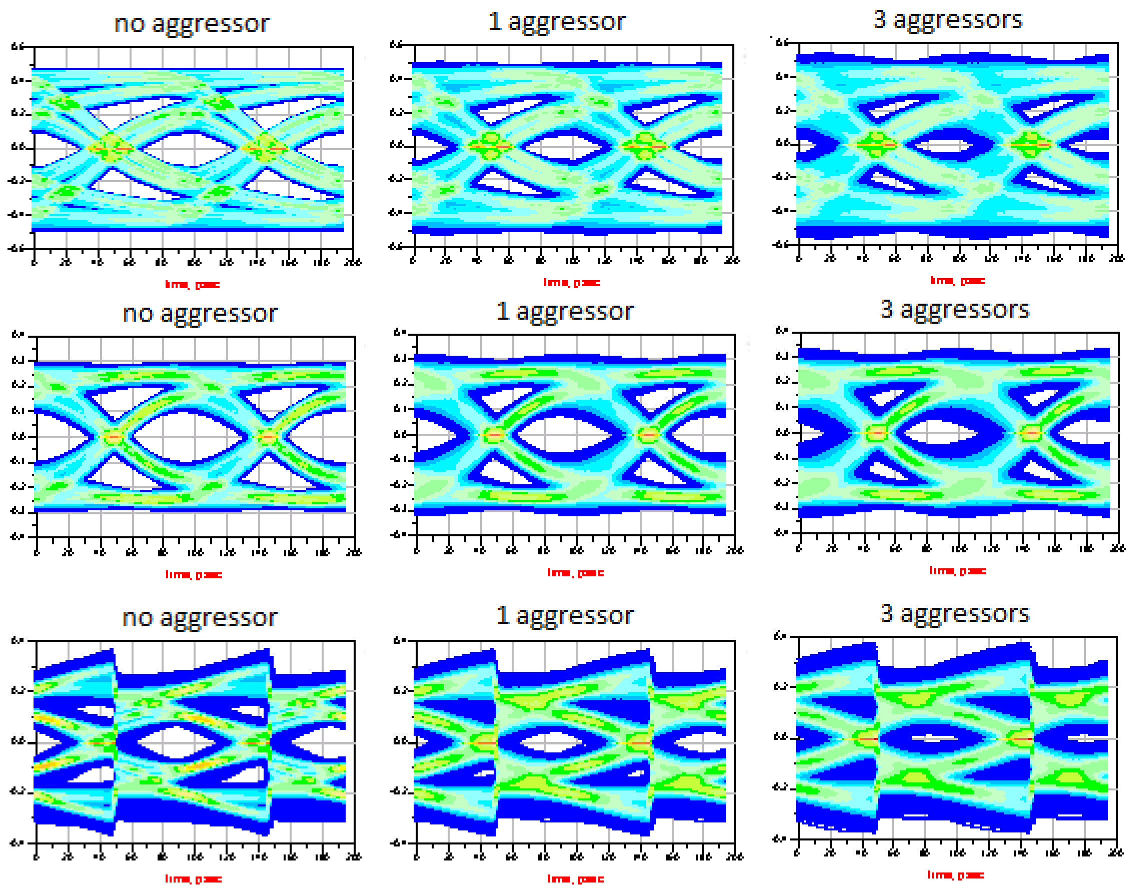
# Channel Crosstalk (Cont'd)

- Each simulation runs one million bits
- To deactivate an aggressor, we set its output amplitude to zero, no other changes.

Victim channel Redriver input eyes

Victim channel Redriver output eyes

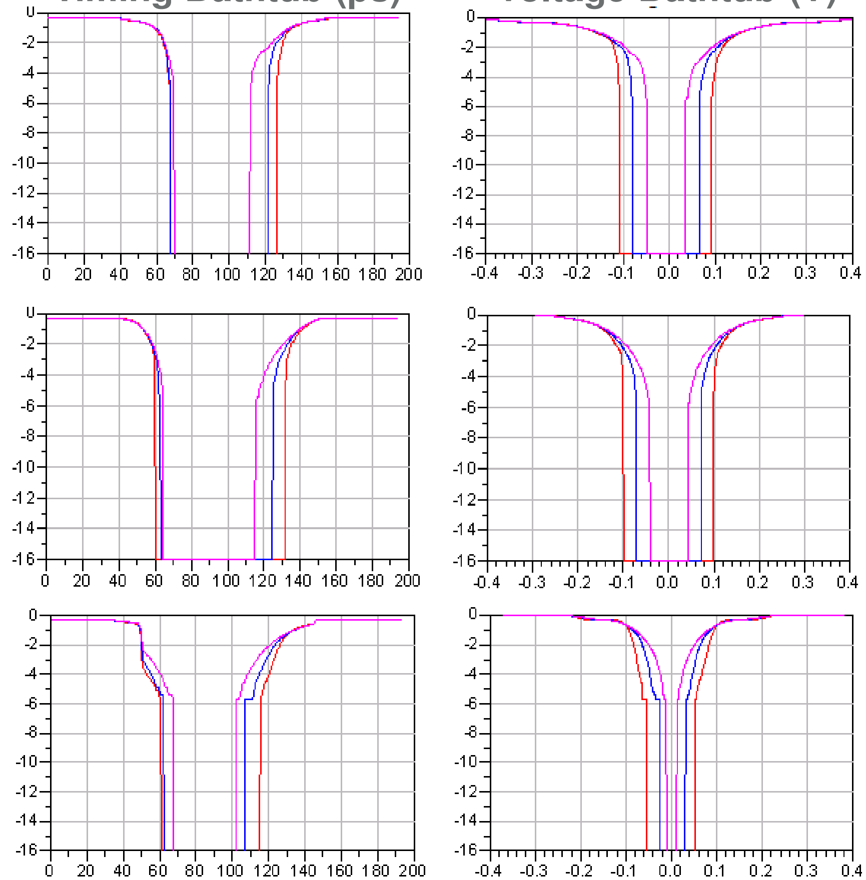
Victim channel Rx output eyes



# Channel Crosstalk (Cont'd)

- Each simulation runs one million bits
- To deactivate an aggressor, we set its output amplitude to zero, no other changes.

Timing Bathtub (ps) Voltage Bathtub (V)




Victim channel Redriver input bathtub curves

Victim channel Redriver output bathtub curves.

Victim channel Rx output bathtub curves.

no aggressor 1 aggressor 3 aggressors

Agilent Technologies

maxim integrated™

# Outline

- Link Simulation with Mid-channel-redrivers
- Modeling Redrivers
- Correlation and Simulation Results
- **Summary**

# Summary

- **Stated** the industry need for an IBIS-AMI methodology to handle mid-channel-redrivers.
- **Presented** an IBIS-AMI Redriver flow and a Redriver model example.
- **Demonstrated** use of Redriver models in link analysis, with impaired channels and crosstalk.
- **Next Up**, our collective task is to make Redriver methodology in IBIS-AMI a reality!