MATLAB EXPO 2019

Wired Communications Systems Modeling and Analysis

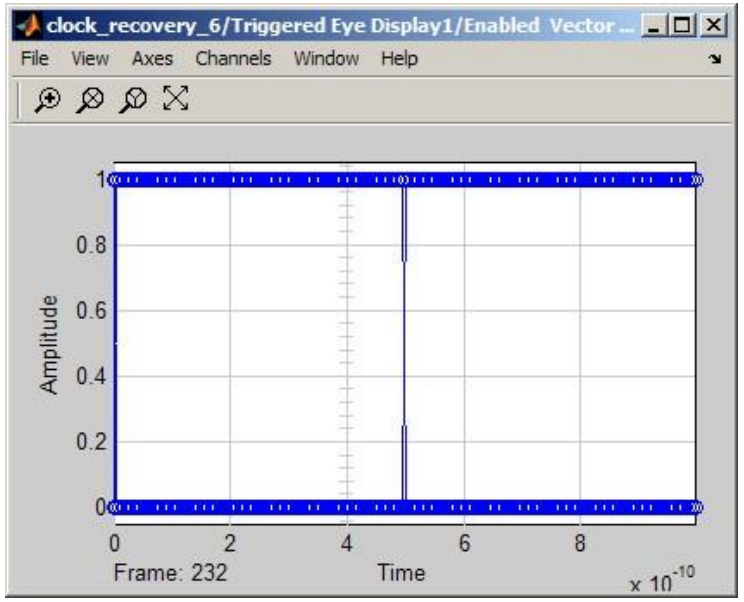
Barry Katz
Development Manager, MathWorks

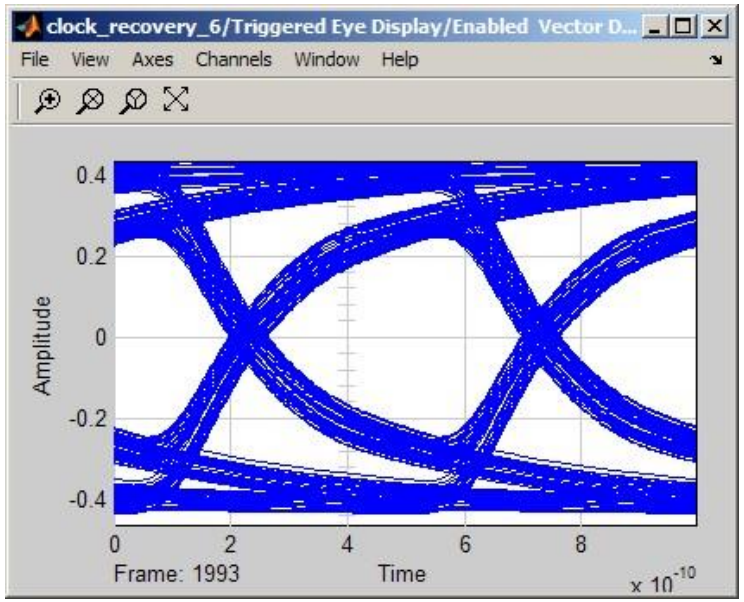
What’s Covered

- Introduction to SerDes Design and Signal Integrity Analysis
- Using SerDes Toolbox for System-Level Design and Analysis
- Automatic Generation of Standard Compliant IBIS-AMI Models
- SerDes Verification Using Channel Simulators

Introduction to SerDes Design and Signal Integrity Analysis

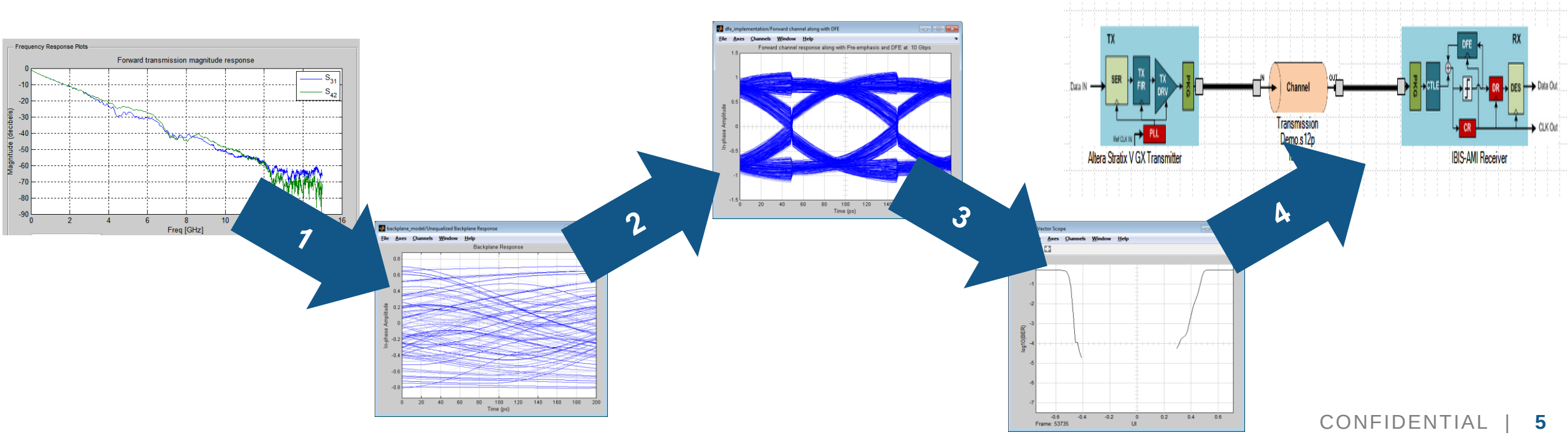
What is Signal Integrity?



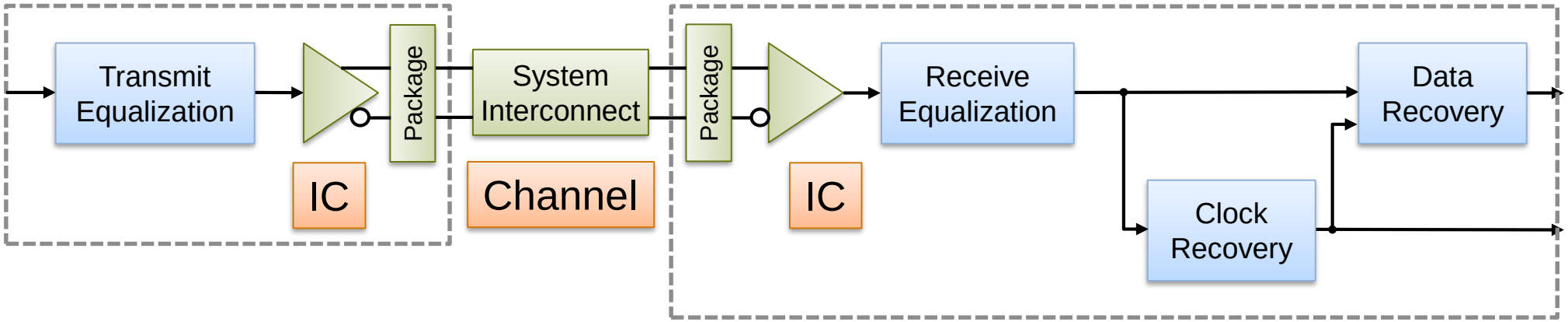


Typical SerDes Design Workflow

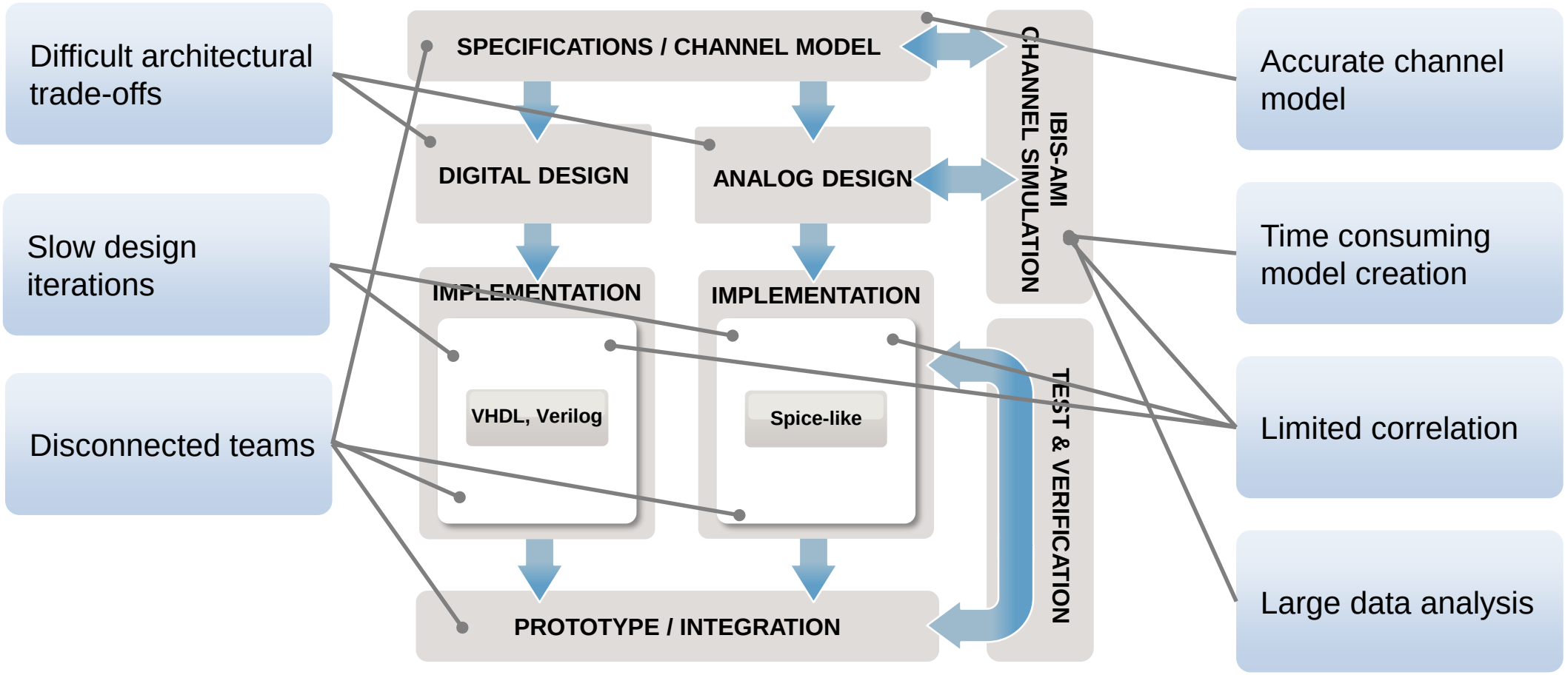
1. Time-domain characterization of the channel
2. Design of analog and digital equalizers
3. Simulation of the system performance in the time domain
4. Hardware implementation and IP verification
5. IBIS-AMI model generation and SerDes system verification



A Typical SerDes System: TX, RX, and Channel



SerDes Design and Verification Challenges

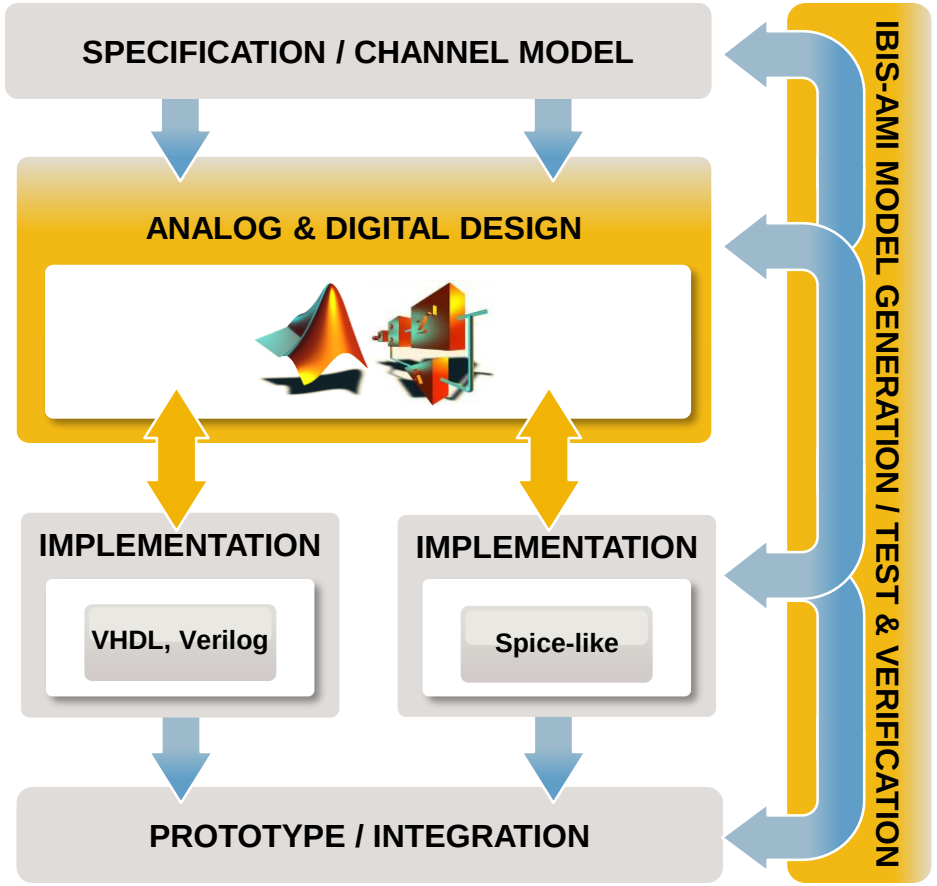


System-Level Design and Analysis Leads to Continuous Verification

Rapid system analysis

Multi-domain simulation

Rapid design iterations

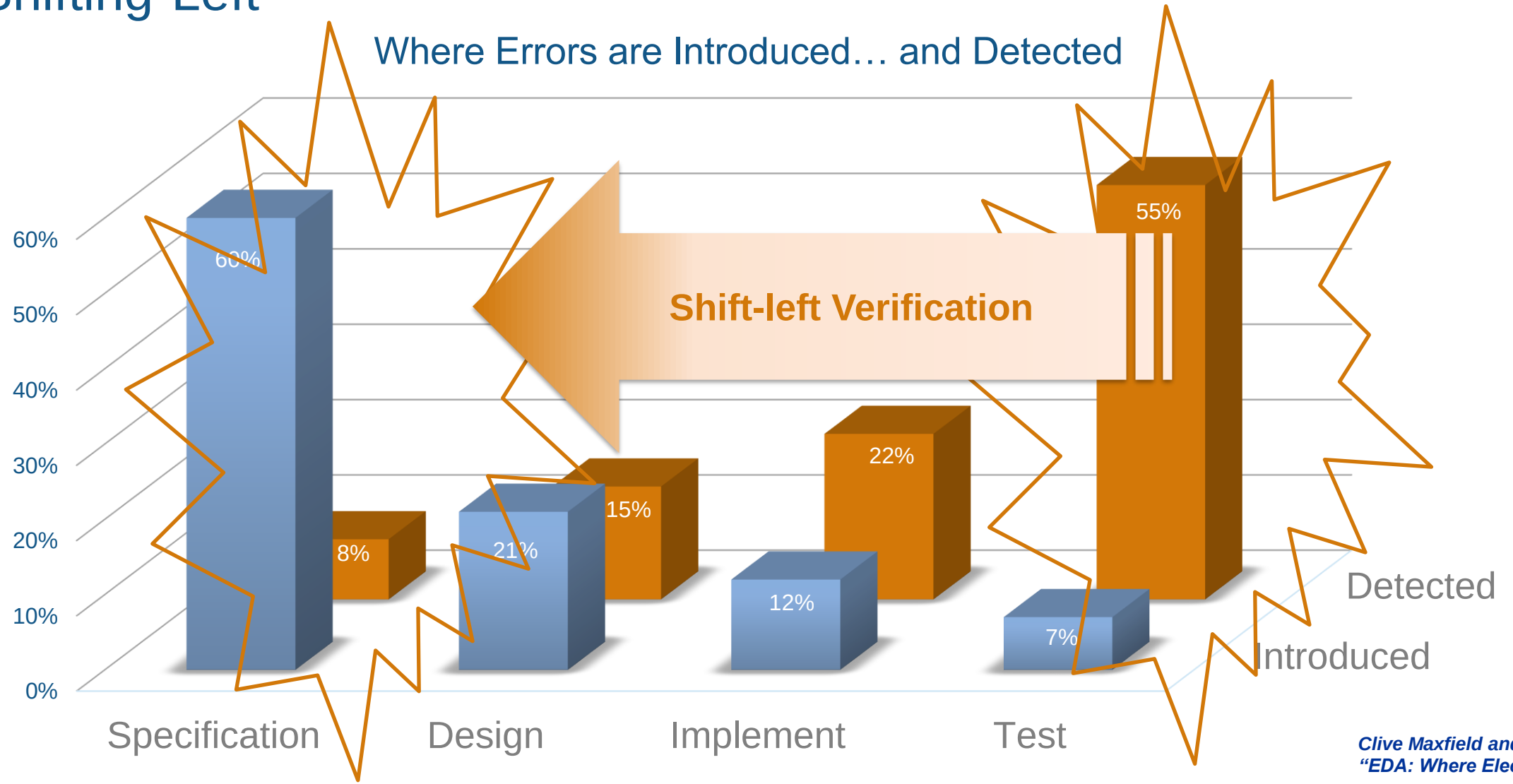


Easier analog modeling

Integrated specification

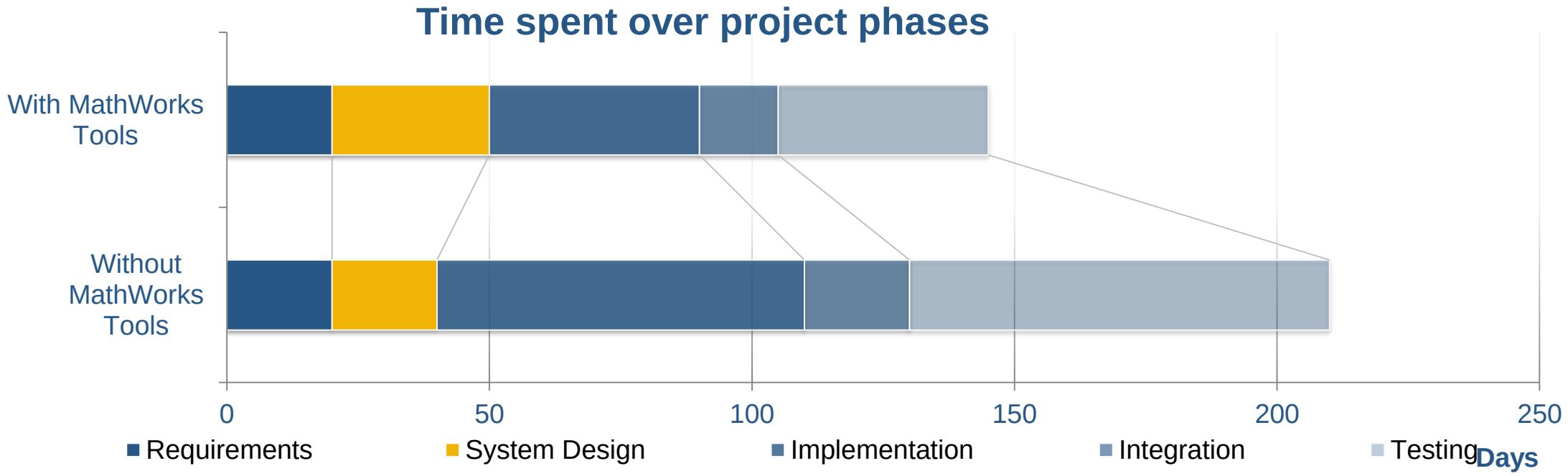
Improved team communication

Shifting-Left



Clive Maxfield and Kuhoo Goyal
"EDA: Where Electronics Begins"

Save 30% of Overall Development Time (and improve quality, reduce re-spins, etc.)



EE Times - Top-down verification guides mixed-signal designs

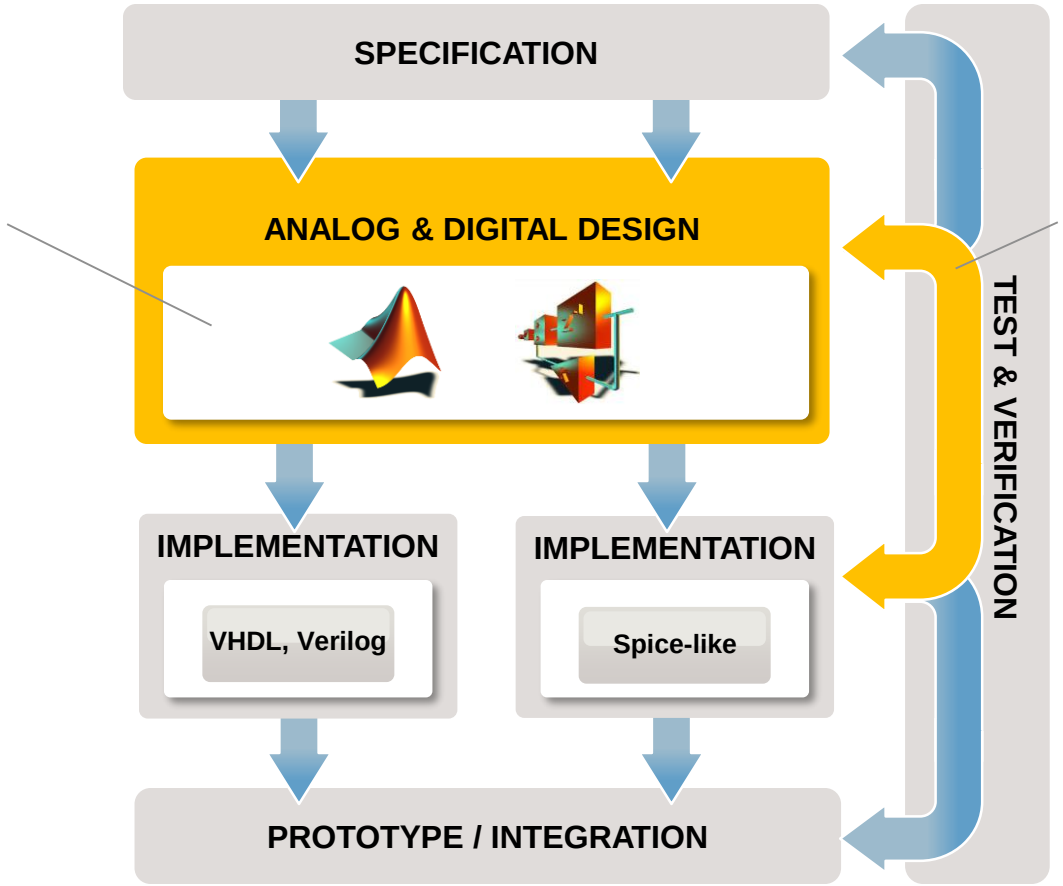
K. Kundert and H. Chang, Partners, Designer's Guide Consulting

*"In order to address these challenges, many design teams are either looking to, or else have already implemented, **a top-down design methodology**. In a top-down approach, the architecture of the chip is defined as a block diagram and simulated and optimized using a system simulator such as **MATLAB or Simulink**. From the high-level simulation, requirements for the individual circuit blocks are derived."*

What's New

R2019a

SerDes Toolbox

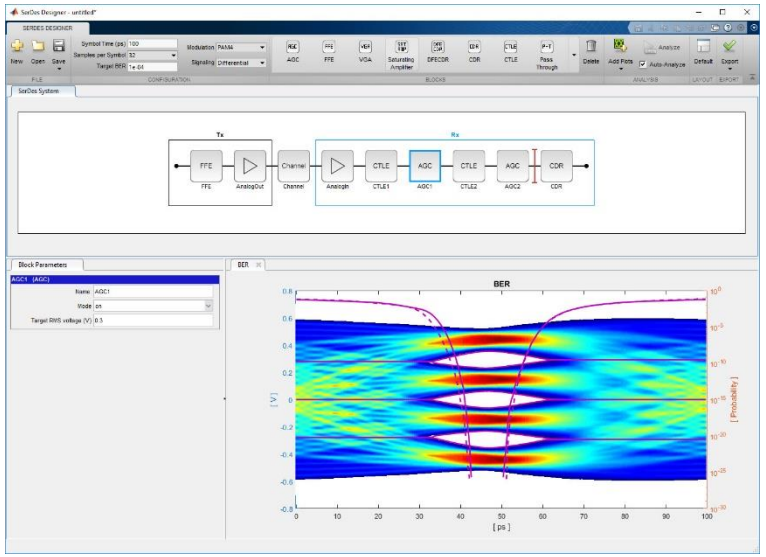


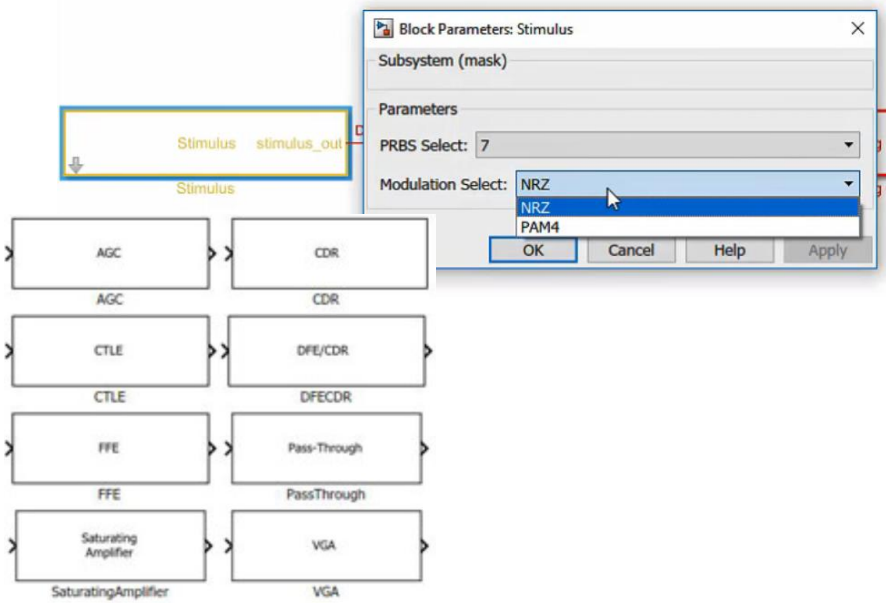
Integration with IC design tools and channel simulators

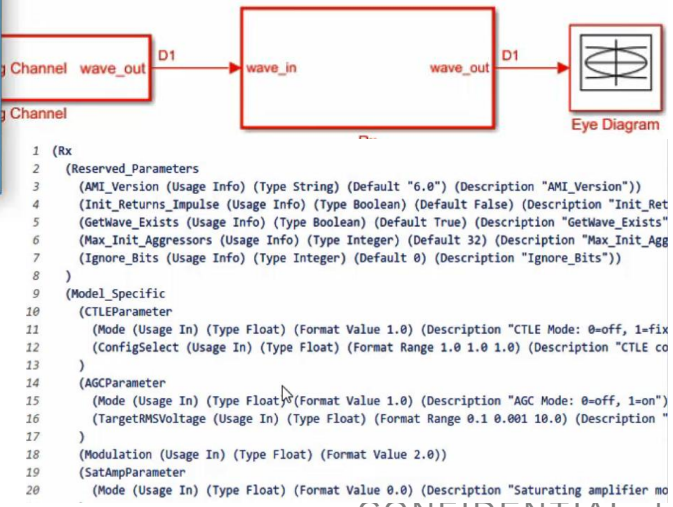
R2019a SerDes Toolbox

Design SerDes systems and generate IBIS-AMI models for high-speed digital interconnects

- Design and analyze transmitters and receivers with the SerDes Designer app
- Develop equalization algorithms with MATLAB System objects and Simulink blocks
  - FFE, DFE, AGC, CDR, CTLE, etc…
- Perform SerDes statistical analysis and time-domain simulation
- Generate dual IBIS-AMI models for 3rd party channel simulators
- Use reference designs for high-speed links such as Ethernet CEI-56G, DDR5, PCI-Gen4, USB3.1

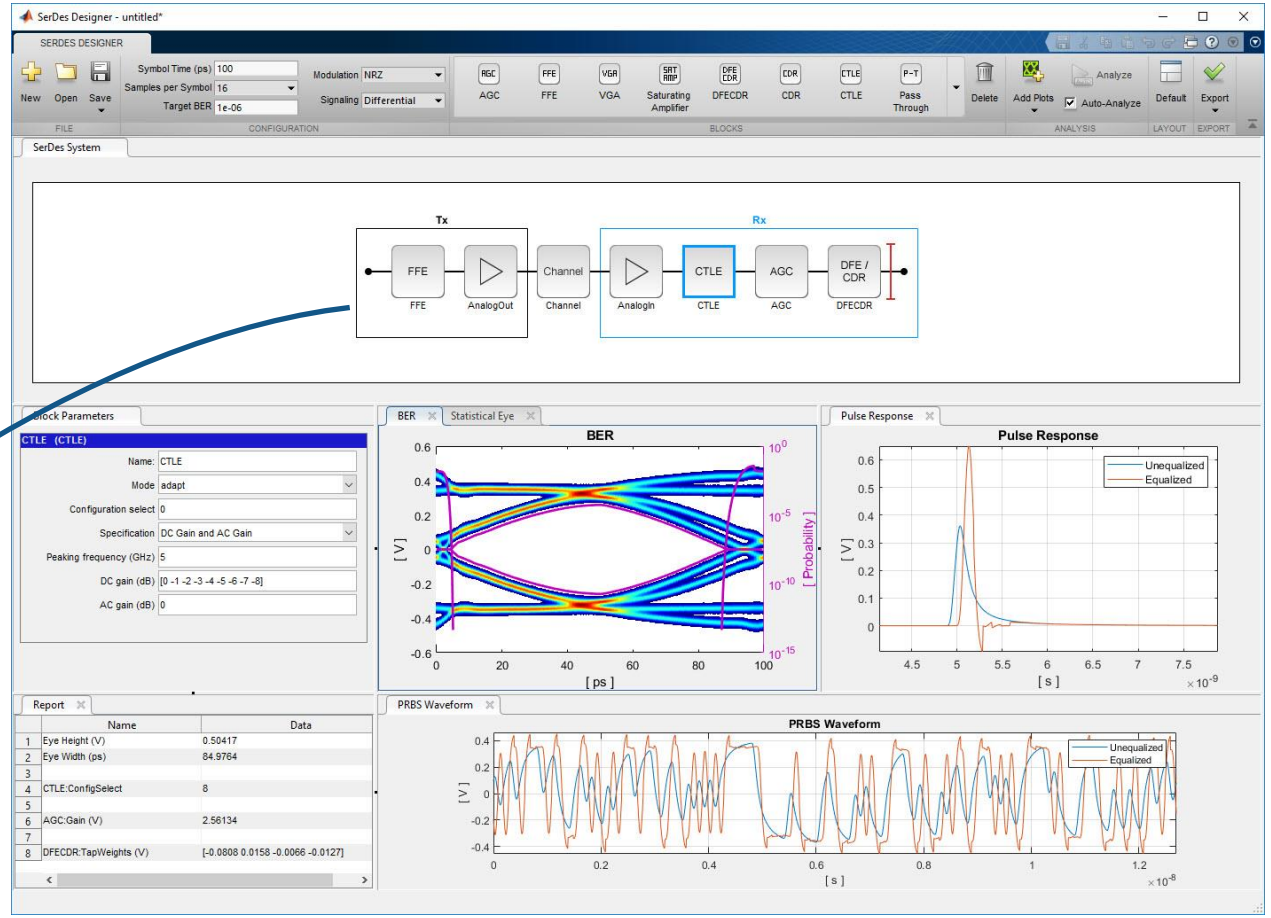


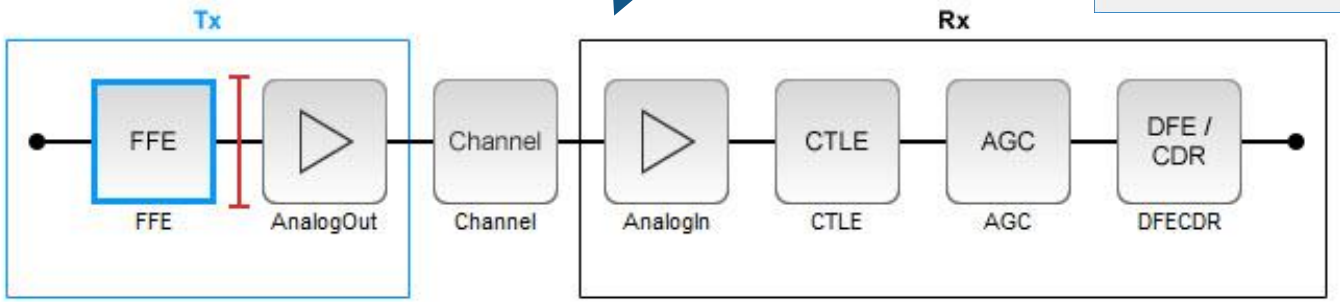




SerDes Designer app: No Need to be a SerDes Expert

- Design and analyze SerDes systems including transmitters and receivers with arbitrary configuration
- Use parameterized building blocks
- Perform statistical analysis: eye diagram, BER, bathtub, pulse response….



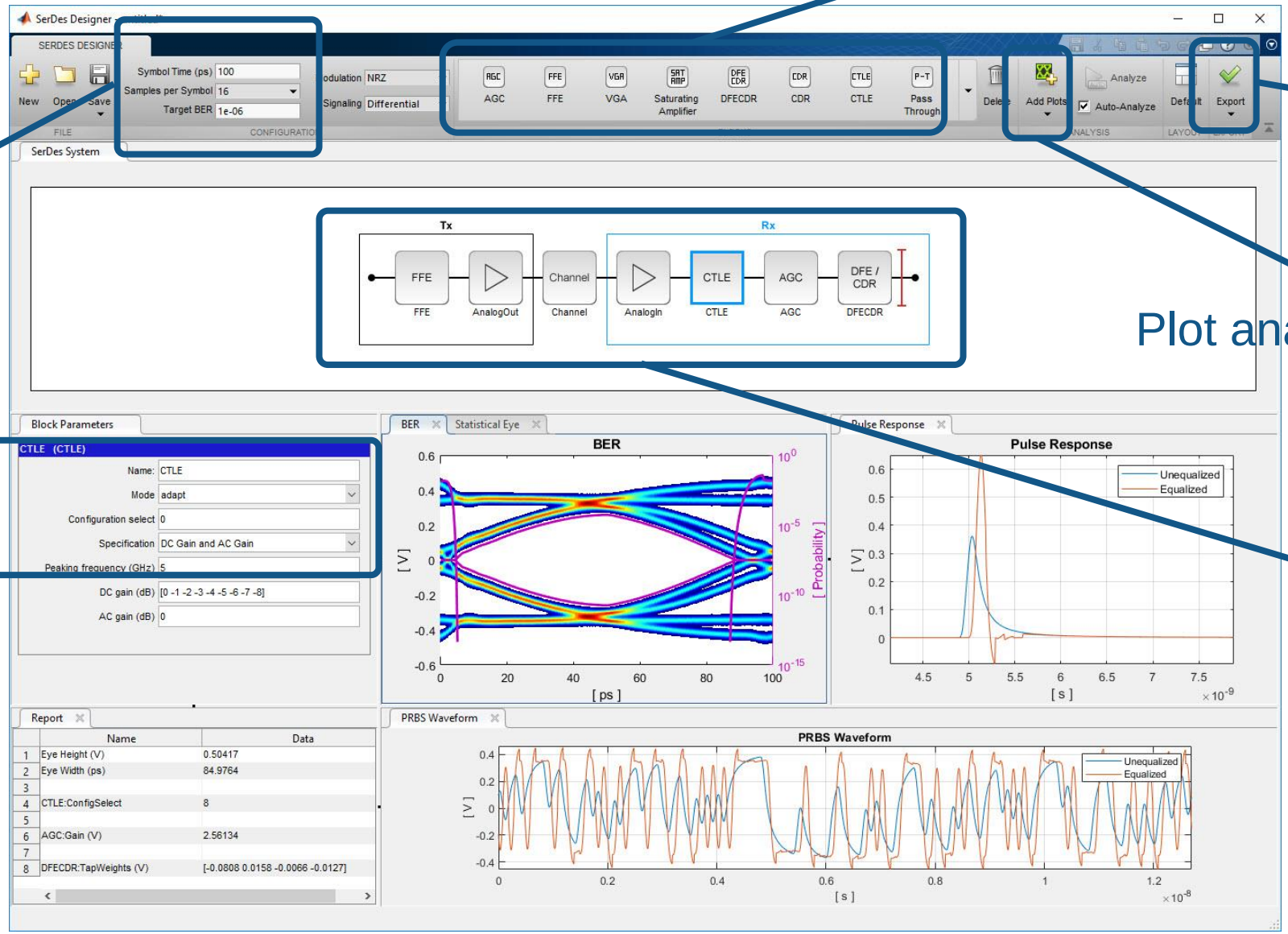


SerDes Design: Where to Start?

Add SerDes components

- Modulation
- Sample rate
- Signaling

Export to:
- MATLAB
- Simulink
- IBIS-AMI



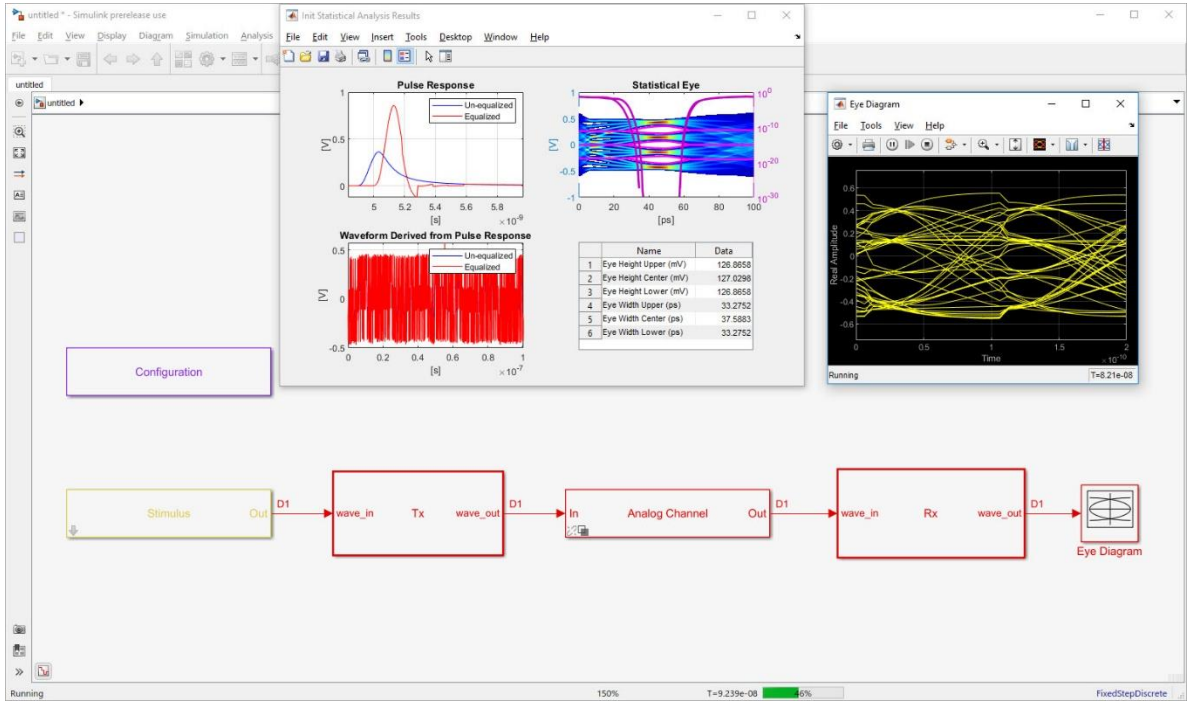
Plot analysis results

Component specifications

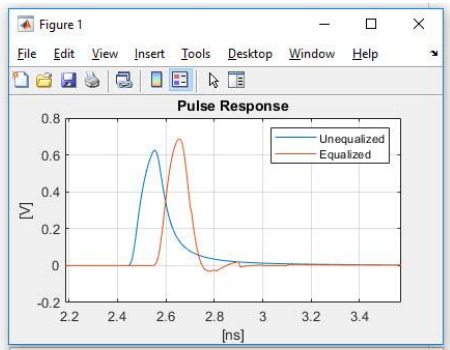
High-speed link

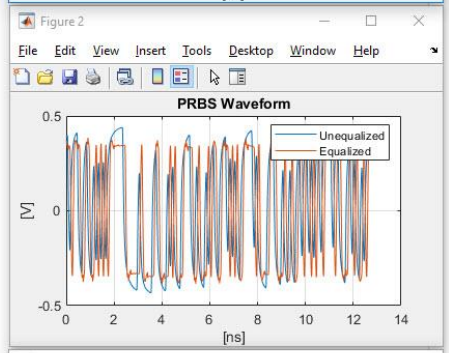
SerDes Top Down Design

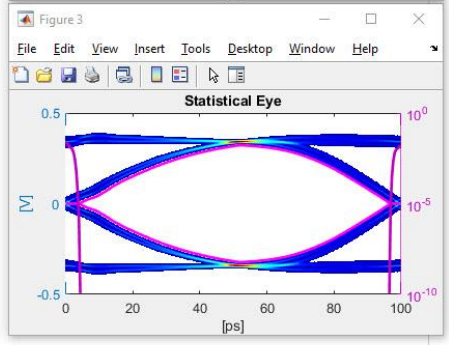
- Create a MATLAB script for automation and design space exploration
- Export to Simulink models for time-domain simulation
- Create dual IBIS-AMI models




```
% ------------------------------------
% MATLAB script to build SerDes System
% ------------------------------------

% Build cell array of Tx blocks:
txBlocks{1} = serdes.FFE;
txBlocks{1}.BlockName = 'FFE';
txBlocks{1}.Mode = 1;
txBlocks{1}.TapWeights = [0 1 0 0 0];
txBlocks{1}.Normalize = true;


% Build cell array of Rx blocks:
rxBlocks{1} = serdes.CTLE;
rxBlocks{1}.BlockName = 'CTLE';
rxBlocks{1}.Mode = 2;
rxBlocks{1}.ConfigSelect = 0;
rxBlocks{1}.Specification = 'DC Gain and Peaking Gain';
rxBlocks{1}.PeakingFrequency = 5000000000;
rxBlocks{1}.DCGain = [0 -1 -2 -3 -4 -5 -6 -7 -8];
rxBlocks{1}.PeakingGain = [0 1 2 3 4 5 6 7 8];

rxBlocks{2} = serdes.AGC;
rxBlocks{2}.BlockName = 'AGC';
rxBlocks{2}.Mode = 1;
rxBlocks{2}.TargetRMSVoltage = 0.3;

rxBlocks{3} = serdes.DFECDR;
rxBlocks{3}.BlockName = 'DFECDR';
rxBlocks{3}.Mode = 2;
rxBlocks{3}.TapWeights = [0 0 0 0];
rxBlocks{3}.MinimumTap = -1;
rxBlocks{3}.MaximumTap = 1;


% Build txModel:
txAnalogModel = AnalogModel( ...
    'R',50, ...
    'C',1.000000e-13);
tx = Transmitter( ...
'Blocks',txBlocks, ...
    'AnalogModel',txAnalogModel, ...
    'RiseTime',1.000000e-11, ...
    'VoltageSwingIdeal',1, ...
    'Name','TX');

% Build rxModel:
rxAnalogModel = AnalogModel( ...
    'R',50, ...
    'C',2.000000e-13);
rx = Receiver( ...
'Blocks',rxBlocks, ...
    'AnalogModel',rxAnalogModel, ...
    'Name','RX');

% Build ChannelData:
channel = ChannelData( ...
    'ChannelLossdB',8, ...
    'ChannelLossFreq',5000000000, ...
    'ChannelDifferentialImpedance',100);

% Build SerDes System:
SymbolTime = 1e-10;
SamplesPerSymbol = 16;
ModulationLevels = 2;
BERtarget = 1e-06;
```

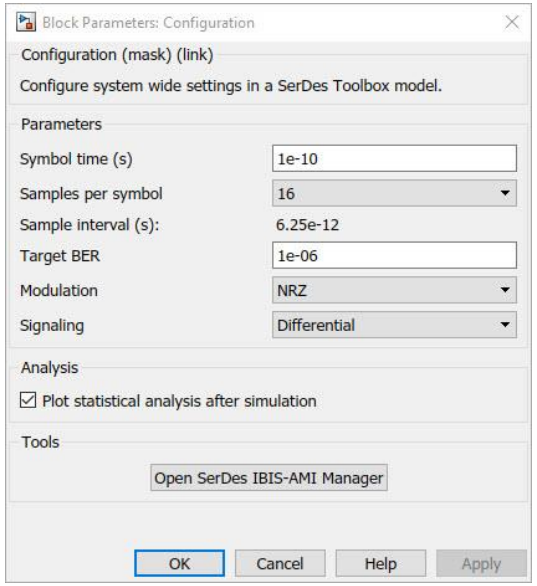


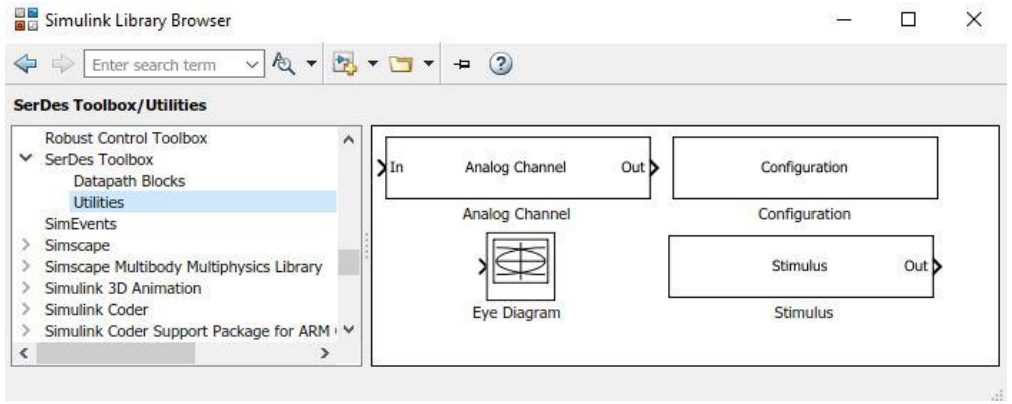


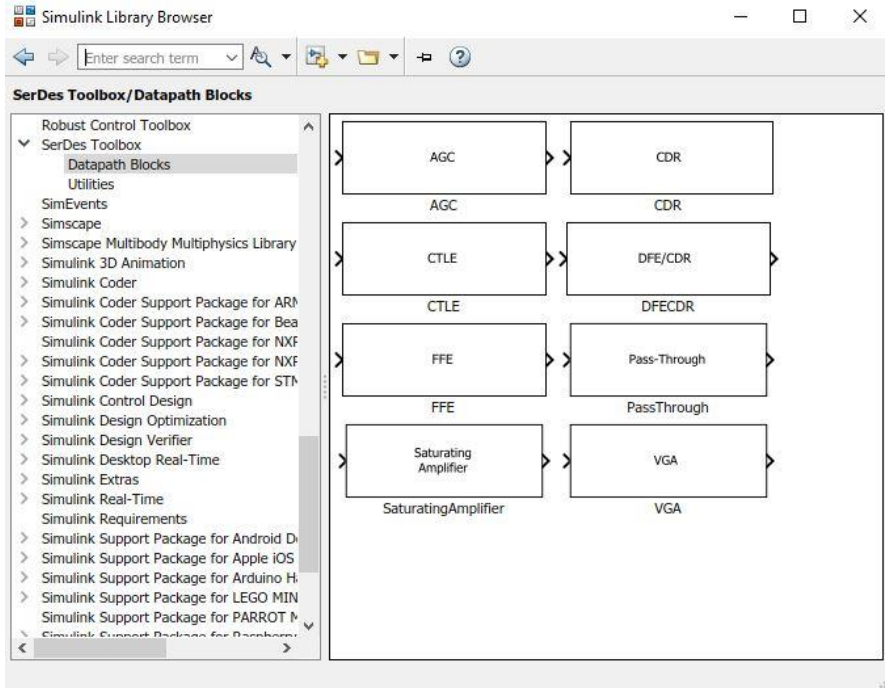


SerDes Toolbox: Simulink Models

- Develop adaptive equalizers using white-box models such as DFE, CTLE, AGC, and CDR
- Use parametrized blocks and algorithms for single-ended and differential signals
- Generate PRBS and custom stimulus patterns supporting PAM4 and NRZ modulation

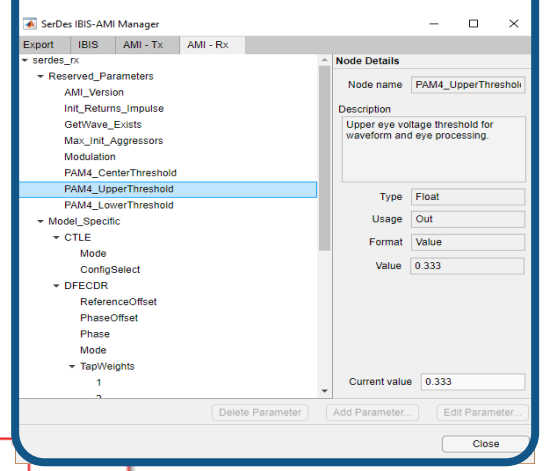


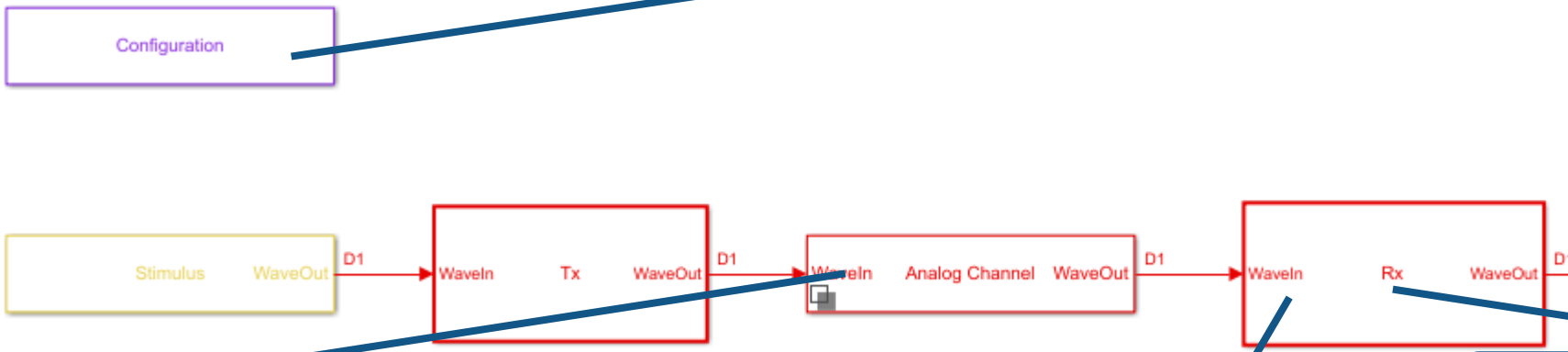




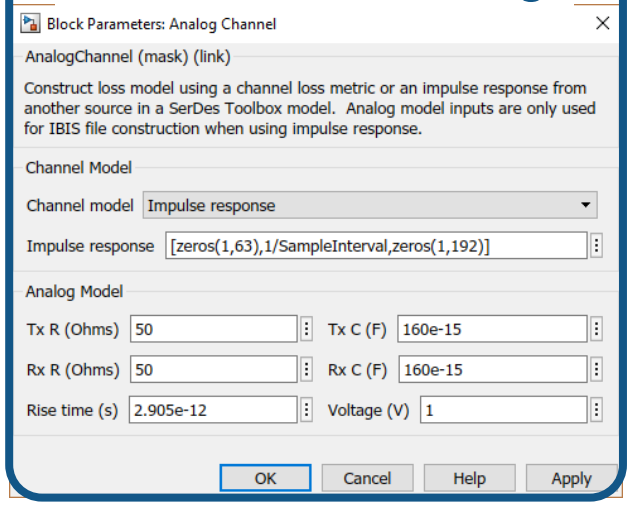
SerDes Simulation and Architecture Exploration

Global Parameters





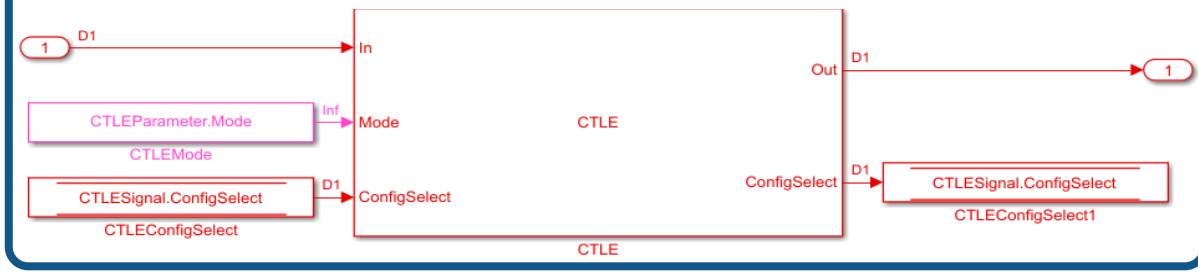
Channel modeling



White-box (customizable) models

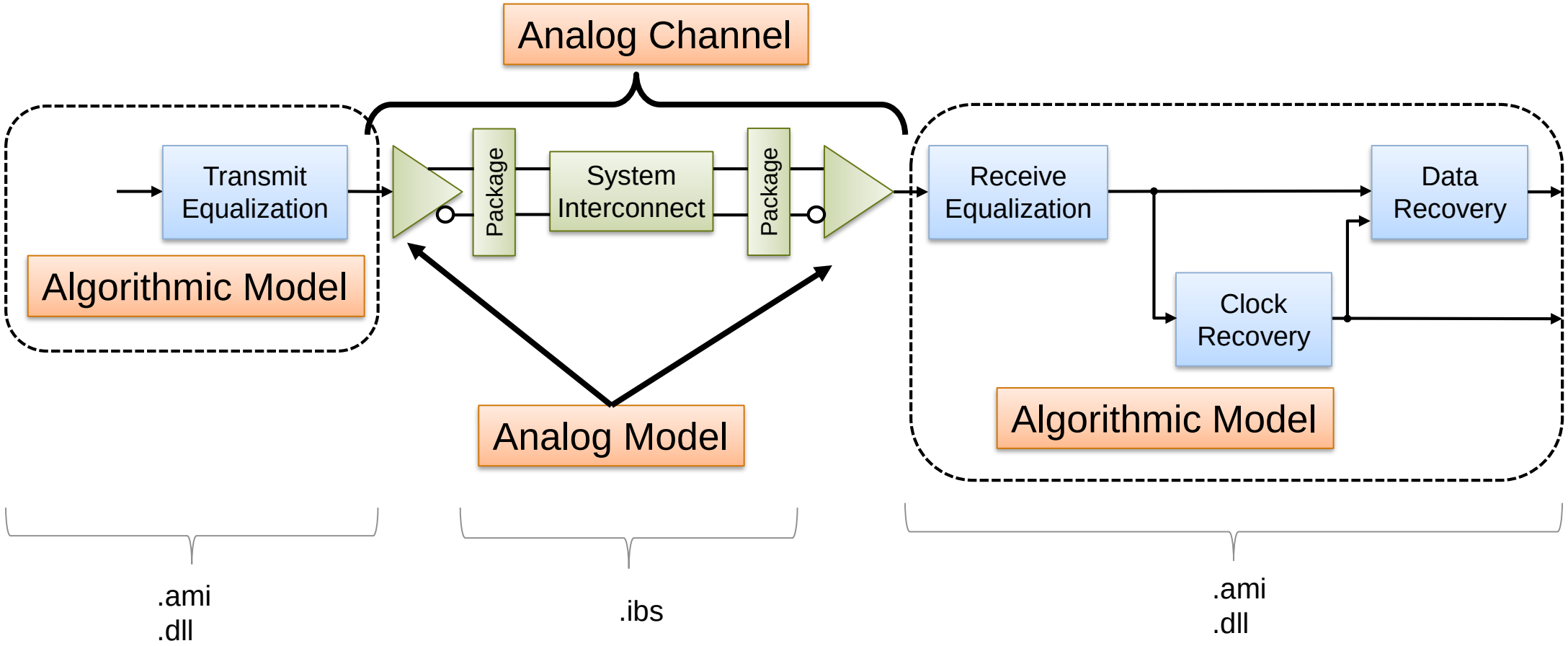
```
%Note that step or time domain adaptation of the CTLE must be
%done in an exterior block.

ixed(obj)
= obj.ConfigSelect;

.privConfigInitialFlag
j.privConfigInitial = obj.ConfigSelect;
j.privConfigInitialFlag = false;

j.privConfigInitial = obj.ConfigSelect;

= obj.privConfigInitial;

oj.FilterCoefficients.np(Config+1)+1;
oj.FilterCoefficients.nz(Config+1)+1;
```

Simulate and customize adaptive equalizers

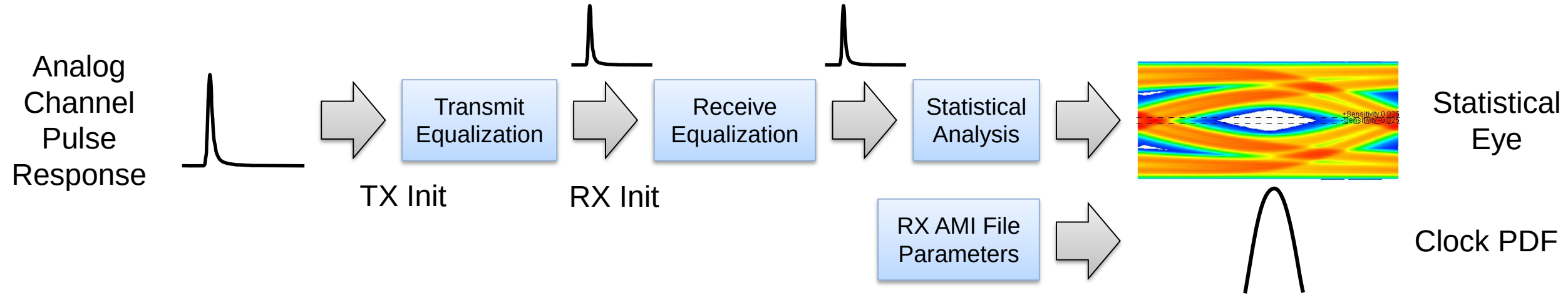


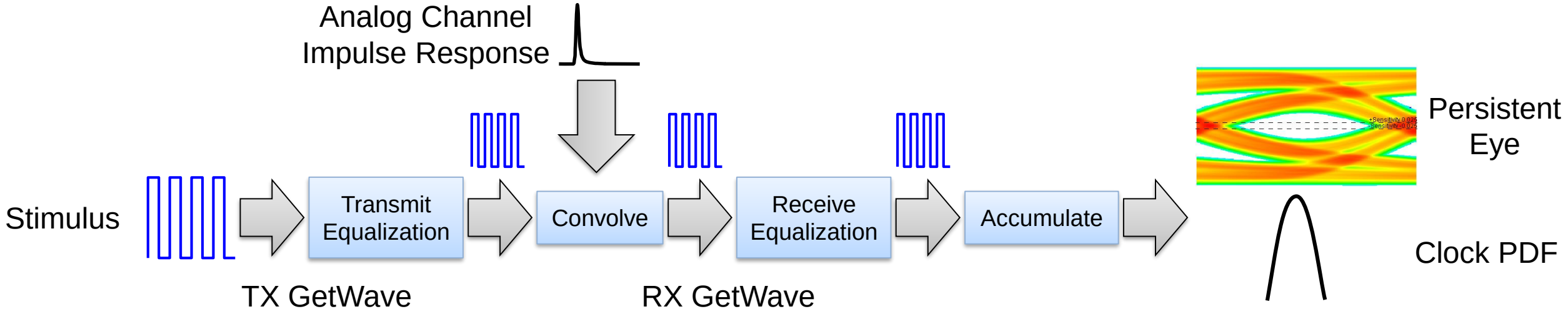
Automatic Generation of Standard Compliant IBIS-AMI Models

IBIS-AMI Terminology



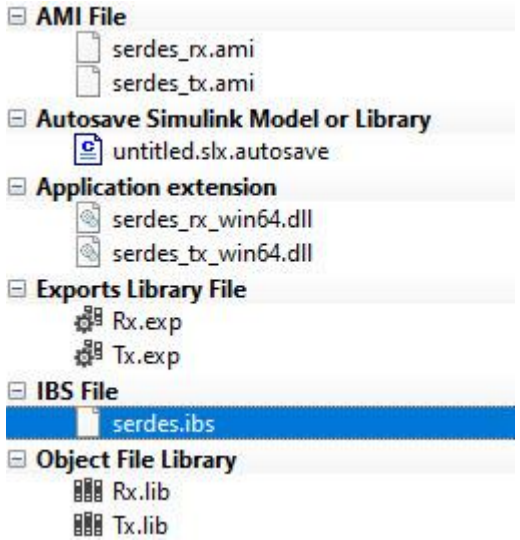
Statistical vs. Time Domain

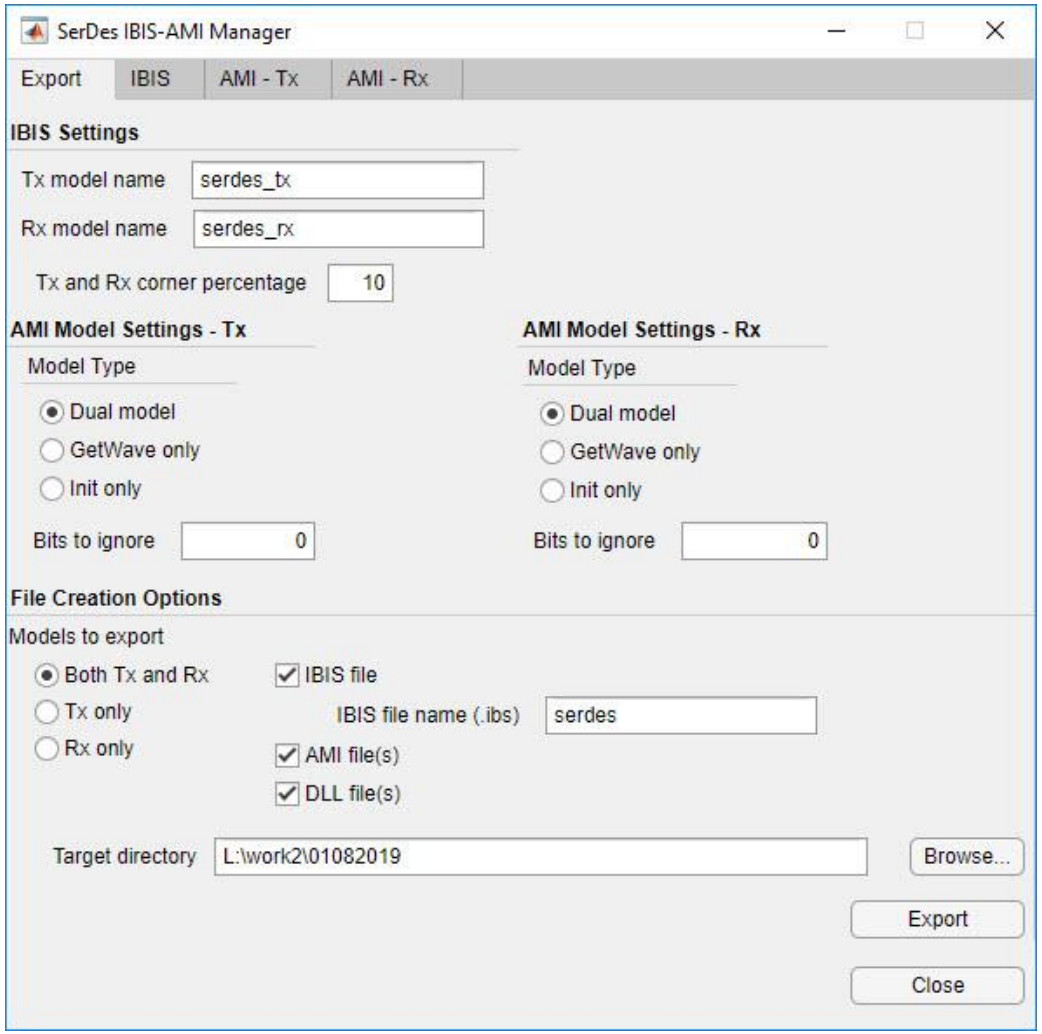




SerDes Toolbox: IBIS-AMI Dual Model Generation

- Generate standard-compliant Init and GetWave IBIS-AMI models
- Generate associated analog IBIS model
- Customize the model interface by managing the IBIS- AMI-parameters

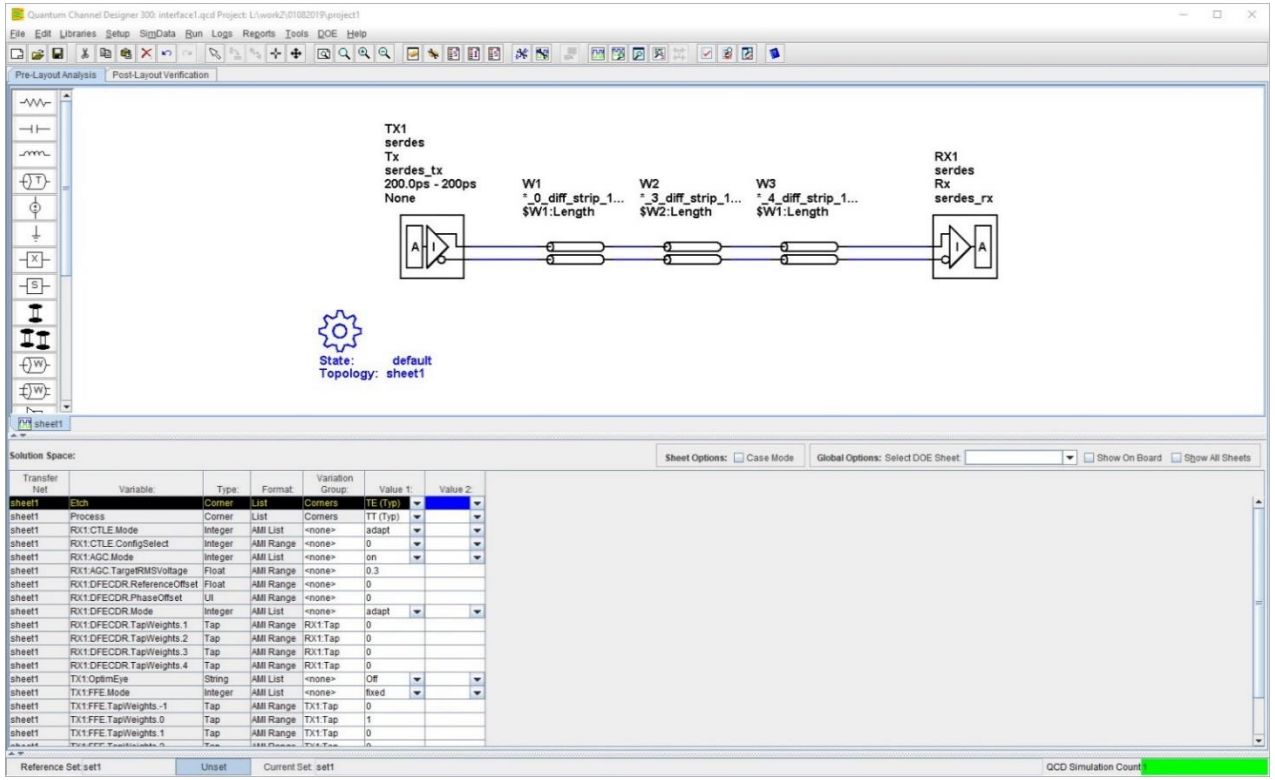




SerDes Verification Using Channel Simulators

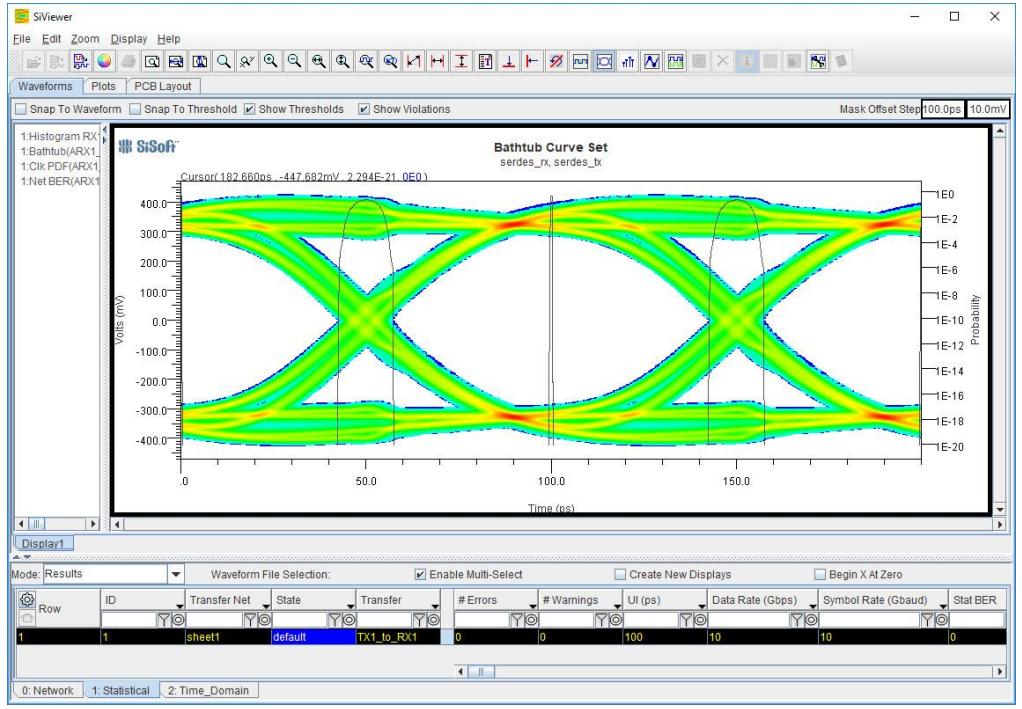
Channel Simulation Using IBIS-AMI Models

- Integrate standard-compliant IBIS-AMI models in 3rd party channel simulators
  - Correlation & regression testing
  - Identification of corner cases over large families of channel models and configurations



Integration with QCD/QSI (SiSoft Link)

- Bidirectional link between SerDes Toolbox and SiSoft QCD/QSI
- Automatically create a QCD/QSI project from SerDes Toolbox
- Back-annotate the channel model, stimuli, and AMI parameter settings into SerDes Toolbox
- Rapidly iterate between system design and channel simulation



Use Simulink and SerDes Toolbox

R2019a

- Algorithmic design, analysis, and system-level simulation of SerDes systems with many trusted functions
- Integrate with 3rd party channel simulators for SerDes verification
  - Generate standard-compliant IBIS-AMI models
- Link with IC design tools to model implementation impairments and reuse testbenches
  - Co-simulation, HDL/SV code generation, and data post-processing

Thank You!
Q&A