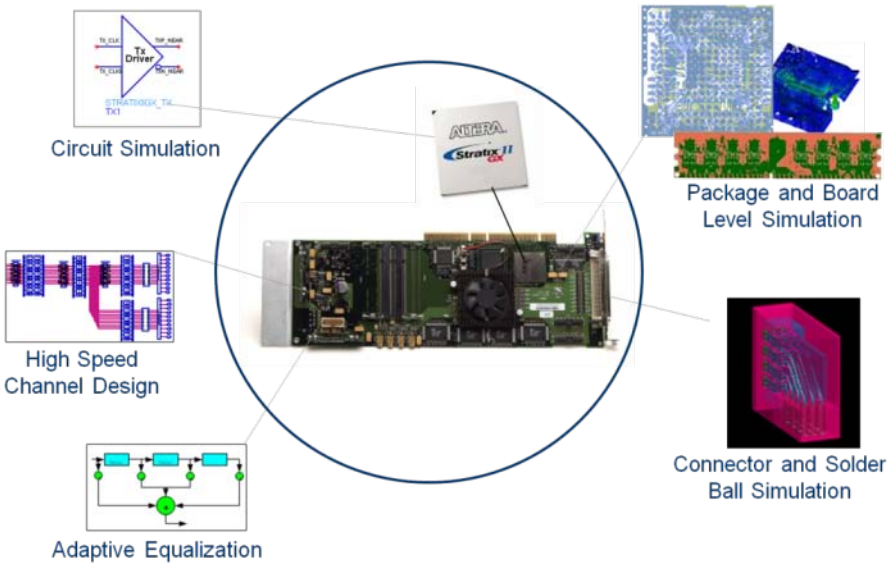
Overcome Signal Integrity Challenges in the Multigigabit/s Era

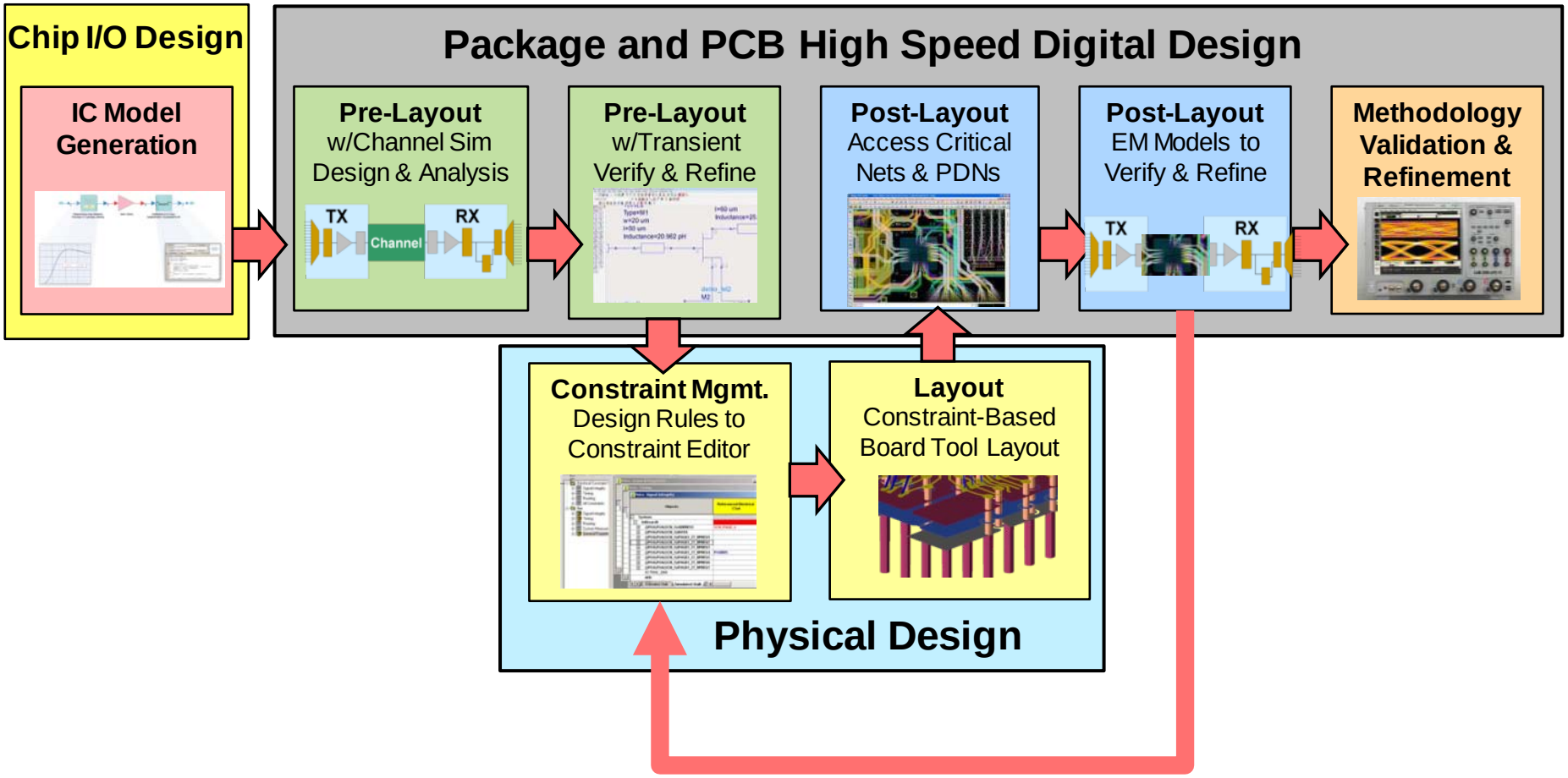


Channel Simulation versus traditional SPICE-like simulation

Agilent EEsof EDA

Dr. Colin Warwick
Product Manager
December 15th, 2011

High Speed Digital Design Flow

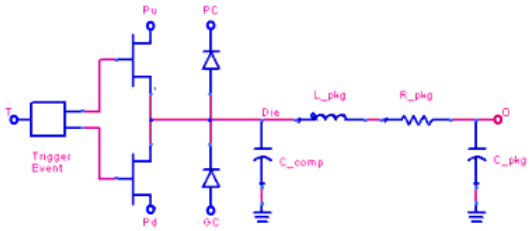


Agilent Technologies

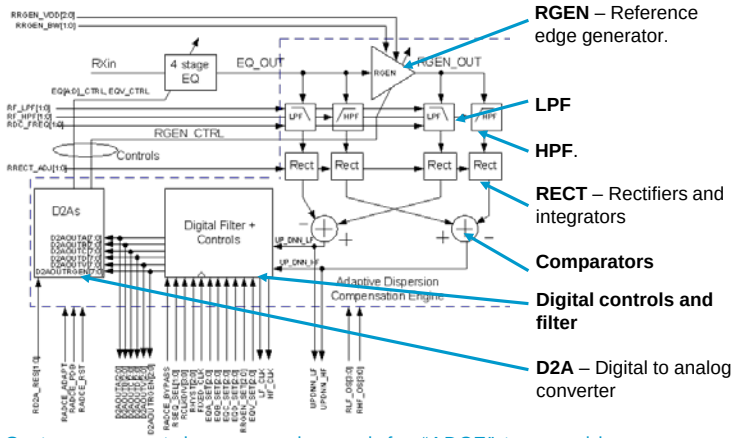
December 15, 2011

Why You Can't Use SPICE For Multigigabit/s SERDES

- Circuitry is too complex
  - 10's k transistors in the logic block
- Design space is too big
  - Tx settings × Channel design × Rx setting = Thousands of simulations



Altera's Adaptive Equalizer



Source: Altera/Agilent joint webcast

IBIS: Input/output Buffer Information Specification

IBIS Open Forum is an organization developing industry standards

http://www.eda-stds.org/ibis/

IBIS 5.0 ratified August 2008 adds an Algorithmic Modeling Interface (AMI) flow as an alternate to the traditional flow.

- SystemVue AMI Modeling Kit for AMI model builder
- Channel Simulator in ADS Transient Convolution supports IBIS 5.0 and proprietary extensions

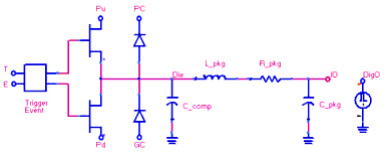
Traditional flow

IBIS Compliant Transient Simulator (SPICE)

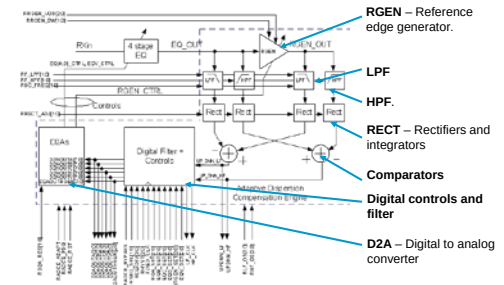
Traditional *.ibs text file

versus

Advanced Design System



Altera's Adaptive Equalizer



AMI flow

IBIS Compliant Channel Simulator

Traditional *.ibs text file plus ref. to…

*.ami header file (text)

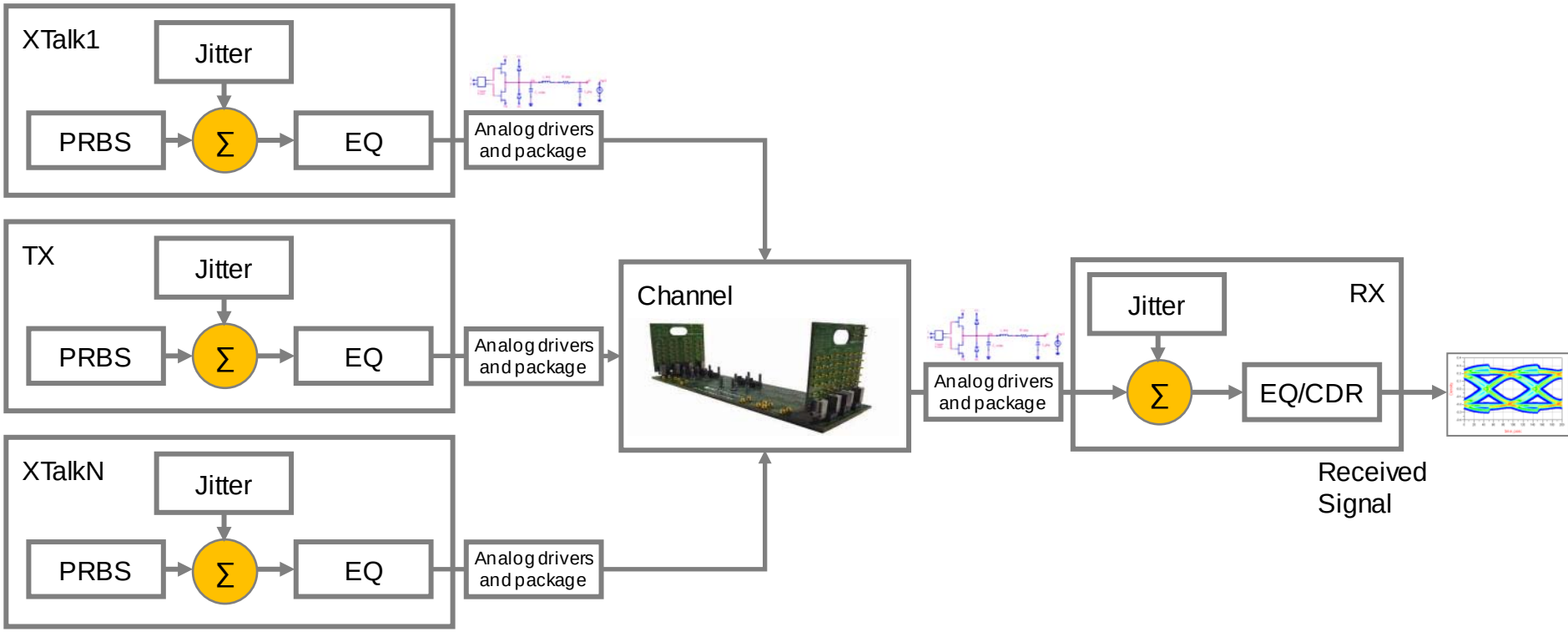
SystemVue 2011.03

Gen. by or C

*.dll or *.so executable

Agilent Technologies

What Do the Two New Files Represent?



*.ami and *.dll files

*.ibs files

Channel model in ADS lumped and distributed components

*.ibs files

Rx *.ami and *.dll files

Agilent Technologies

What Does IBIS AMI Flow Offer?

Interoperability: IC Vendor A←→IC Vendor B

Portability: One IC model runs every compliant EDA tool

Performance: Ultralow BER contours in seconds not days

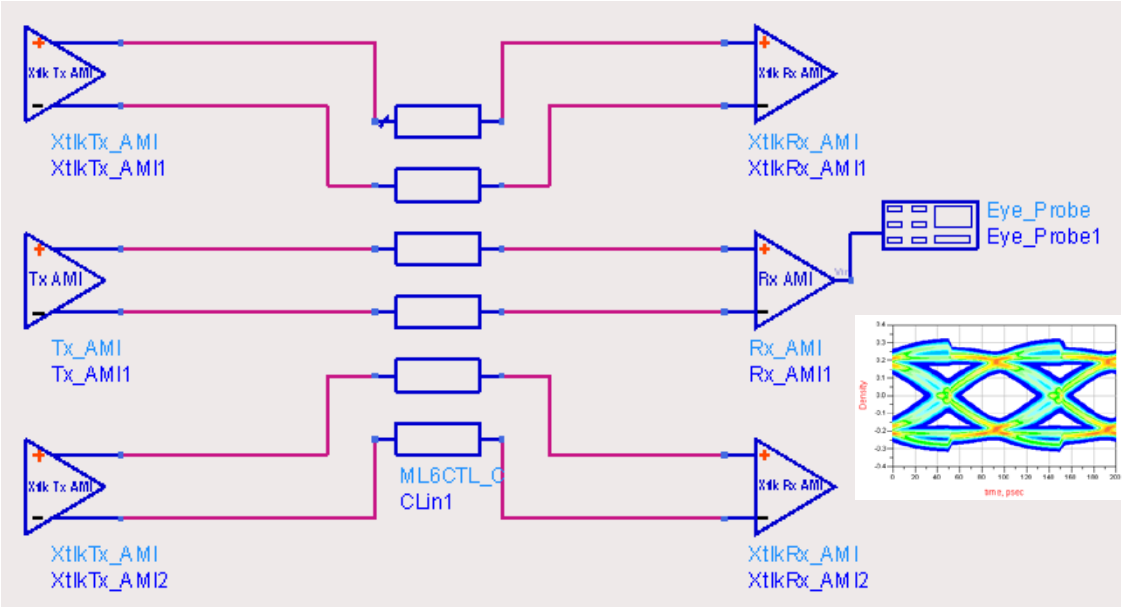
Flexibility: Statistical and bit-by-bit (“time domain”) modes. *.ami header file allows IC vendor to expose arbitrary model-parameters

Optimization: Simulator can sweep model-specific parameter

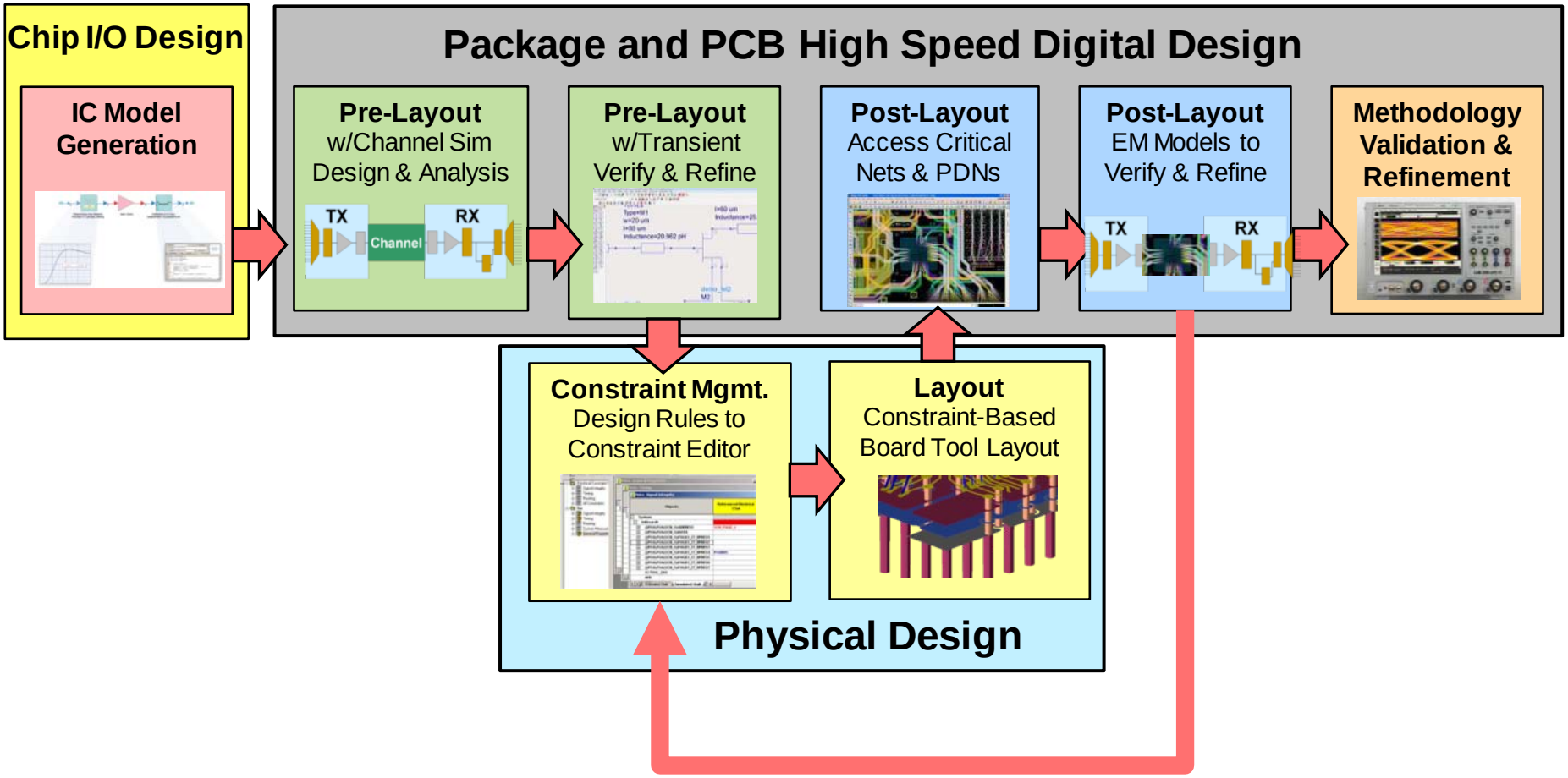
IP Protection: Without the need for non-portable, proprietary encryption keys

…OK, But What's the Catch?

- Must use a channel simulator, not transient simulator (SPICE)
  - …although channel simulator can call other simulators (transient, EM) for generalized channel characterization
- In version 5.0, channel must be linear and time invariant (LTI)
  - …although this restriction is being lifted in a future version to accommodate re-drivers, re-timers etc. (e.g. BIRD 133, ADS 2011.10)
- Non-linear time varying Tx and Rx must be expressed as compiled C code with a specific API
- *.dll or *.so is OS-specific
- Fixed topology
  - …although channel can be made of any arbitrary lumped or distributed elements (if LTI)
- Patching machine code into simulator, not parsing lang.

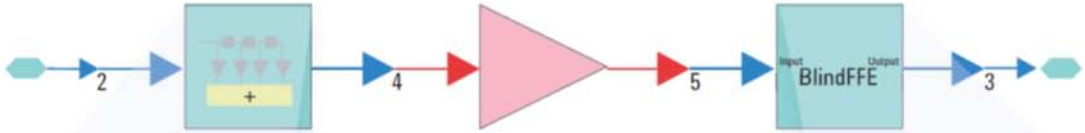


High Speed Digital Design Flow



Agilent Technologies

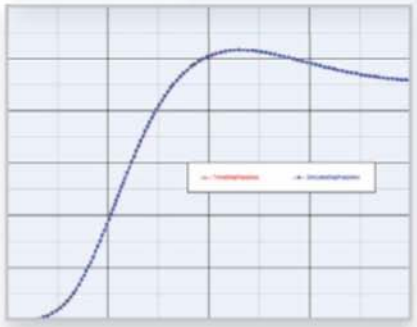
For Model Builders: SystemVue AMI Modeling Kit

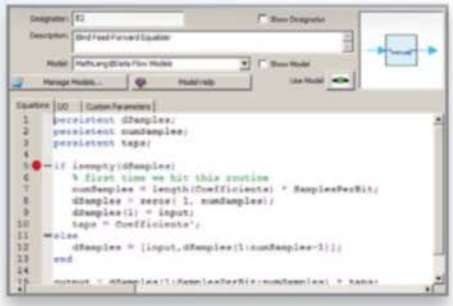


ResponseType=Step Response
TimeStep=1e-12[sample_interval]

Gain=1[Gain]

Coefficients=0;1;0.[Taps]
SamplesPerBit=16[SamplesPer Bit



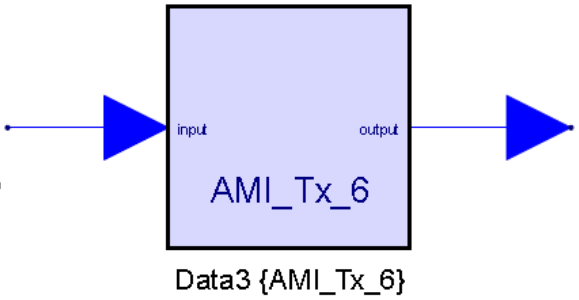


The SerDes model blocks can be specified in several way. The taps of the standard FIR block on the left were tuned so that the step response (blue) matches measured data (red). In contrast, the block on the right was created with custom code.

Agilent Technologies

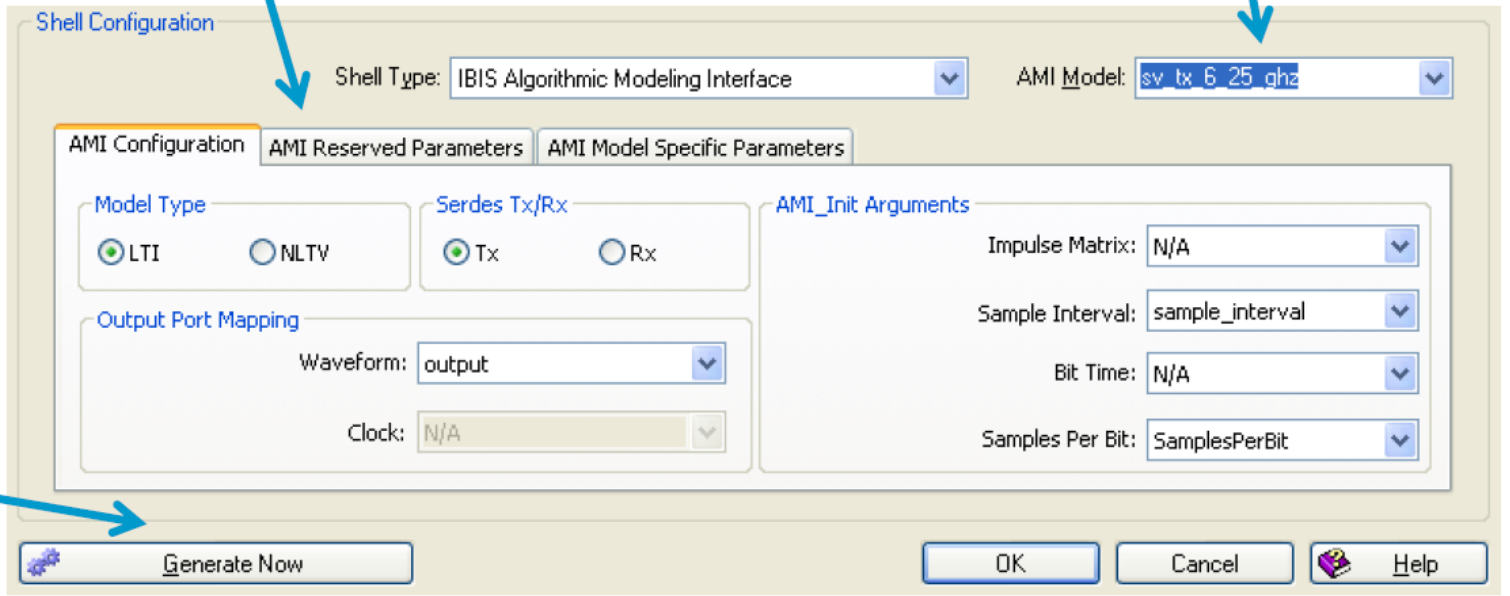
One-click AMI Code-Generation

Step-5: Configure AMI models and generate models with one click.



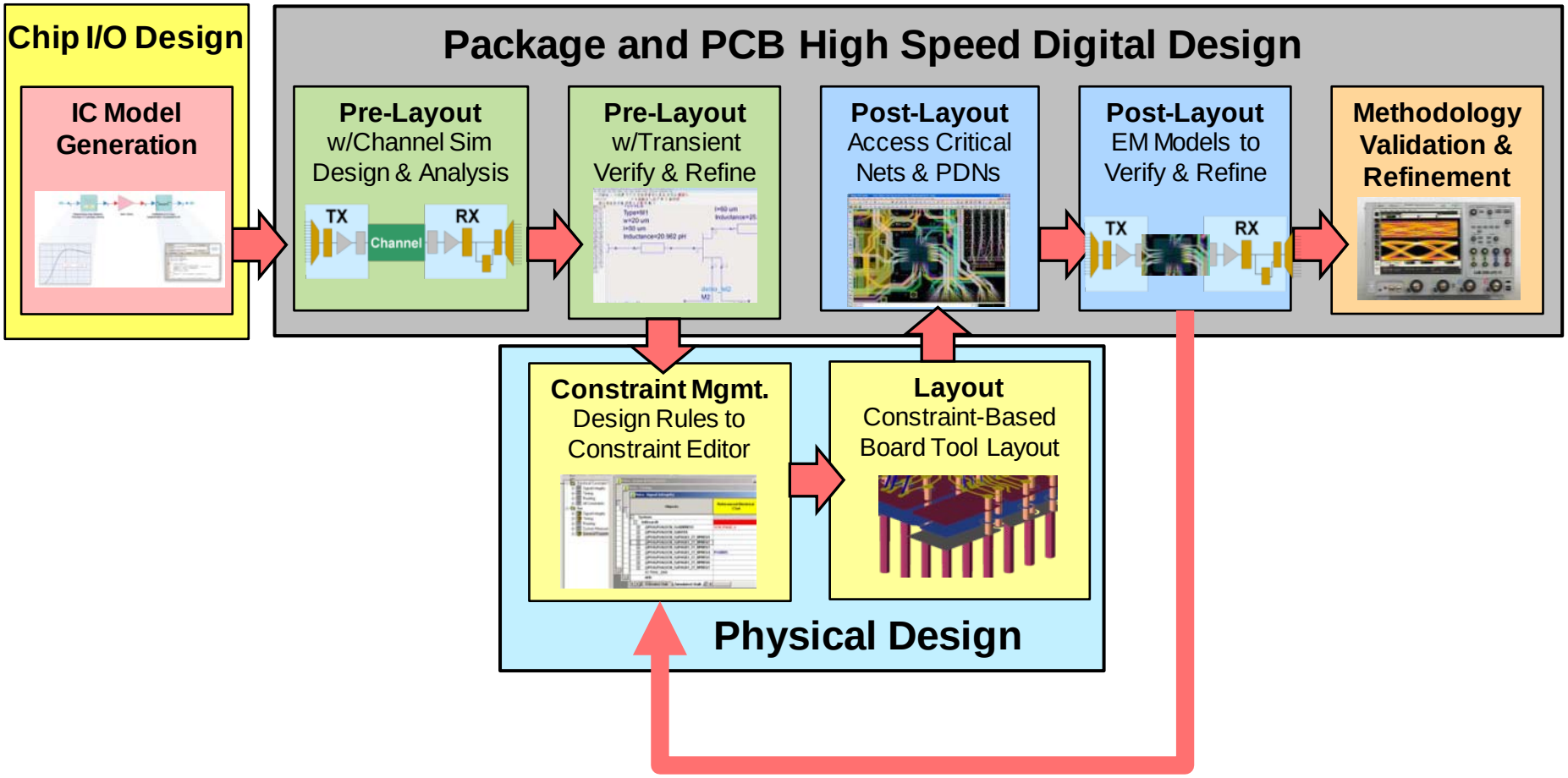
Define Reserved and Model Specific Parameters -> Automatically configure appropriate AMI wrapper

One-click AMI Code-generation



Agilent Technologies

High Speed Digital Design Flow

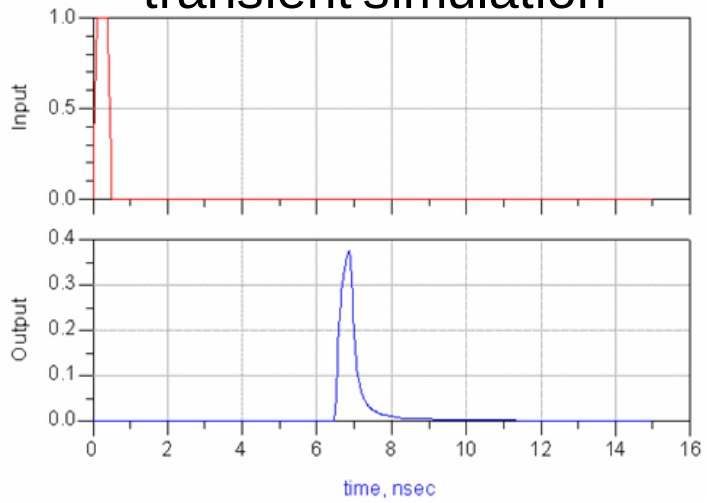


Agilent Technologies

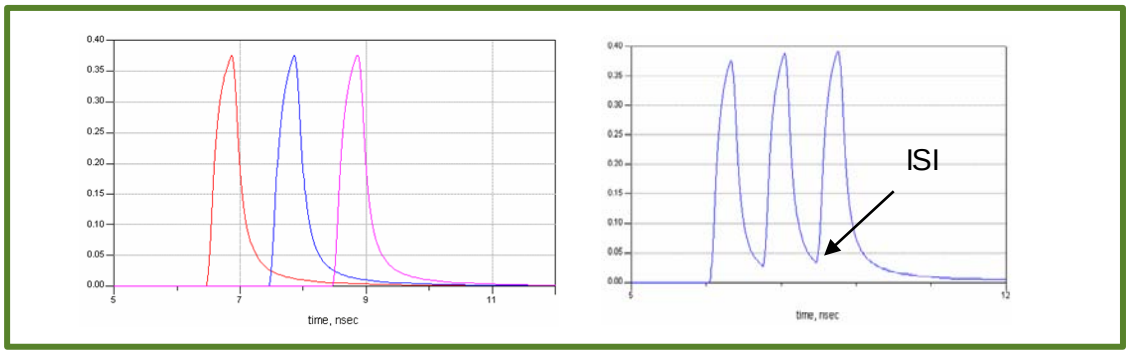
December 15, 2011

For Model Users: Channel Simulator in ADS Transient Convolution Element

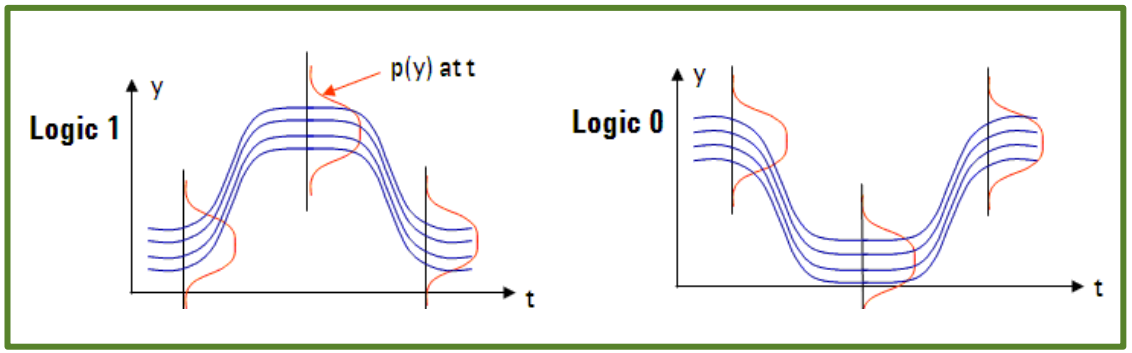
Impulse response is calculated using a short, traditional transient simulation



Bit by bit mode : Superposition of bits



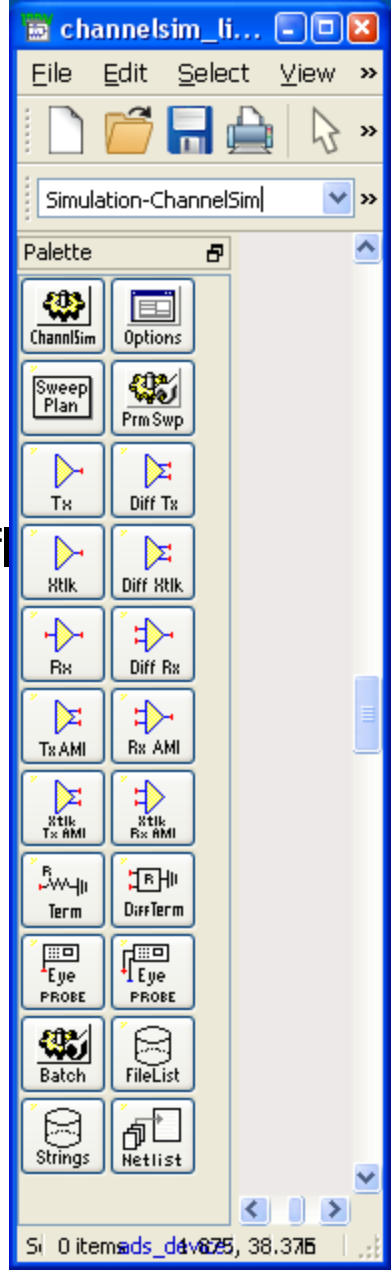
Statistical mode : Statistical techniques



Agilent Technologies

New IBIS-AMI Components in ADS 2011

- IBIS-AMI icons are in the Simulation-ChannelSim palette
- All AMI components are differential
- IBIS pins are autoconnected and hidden
- ADS use model is pin-oriented (same as our traditional IBIS fl
  - select [Component] (IC)
  - then select [Pin]
  - if [Pin] points to [Model Selector]
    - select [Model]
- Only models that contain [Algorithmic Model] are selectable.



Demo

	Transient Simulator	Channel Simulator: Bit-by-bit mode	Channel Simulator: Statistical mode
Method	Modified nodal analysis of Kirchoff's current laws for every time step	Bit-by-bit superposition of step responses	Statistical calculations based on step response
Applicability & Restrictions	Linear and non-linear circuits, traditional IBIS flow	LTI channel, any specific bit pattern, NLTV Tx/Rx e.g. adaptive eq. taps, CDR, IBIS AMI time domain flow	LTI channel, fixed (general) bit pattern, LTI Tx/Rx e.g. fixed eq. taps, IBIS AMI statistical flow
BER floor in one minute simulation	~10^{-3}	~10^{-6}	~10^{-16}

.ibs File

For AMI simulations, the *.ibs file must contain the [Algorithmic Model] keyword:

… (regular IBIS parameters)

[Algorithmic Model]

Executable Windows_VisualStudio_32 IBIS_AMI_Tx.dll IBIS_AMI_Tx.ami

Executable linux_gcc_64 libIBIS_AMI_Tx.so IBIS_AMI_Tx.ami

[End Algorithmic Model]

The information specifies:

platform_compiler_bits (Must match OS, OS bits, & compiler runtime libs)
DLL file name
parameter file name (the *.ami header file)

Header File *.ami:

Exposes a Model-Specific Interface to the EDA Tool

```
(mySampleAMI                                | Name given to the Parameter file
 (Description "Sample AMI File")
 (Reserved_Parameters                       | Required heading
   (Ignore_Bits (Usage Info) (Type Integer) (Default 21)
    (Description "Ignore 21 Bits"))
  (Max_Init_Aggressors (Usage Info) (Type Integer)(Default 25))
  (Init_Returns_Impulse (Usage Info) (Type Boolean)(Default True))
  (GetWave_Exists (Usage Info) (Type Boolean) (Default True))
 )                                          | End Reserved_Parameters
 (Model_Specific                            | Required heading
  (txtaps
   (-2 (Usage Inout)(Type Tap) (Format Range 0 -0.1 0.2)(Default 0.1)
      (Description "Second Precursor Tap"))
   (-1 (Usage Inout)(Type Tap) (Format Range 0 -0.4 0.4)(Default 0.2)
      (Description "First Precursor Tap"))
   (0 (Usage Inout)(Type Tap) (Format Range 1 -1 2)(Default 1)
      (Description "Main Tap"))
   (1 (Usage Inout)(Type Tap) (Format Range 0 -0.4 0.4)(Default2 0.2)
      (Description "First Post cursor Tap"))
   (2 (Usage Inout)(Type Tap) (Format Range 0 -0.1 0.2)(Default 0.1)
      (Description "Second Post cursor Tap"))
  )                                         | End txtaps
  (tx_freq_offset (Format Range 1 0 150) (Type UI) (Default 0))
 )                                          | End Model_Specific
)                                           | End SampleAMI
```

Tree structure

Each leaf defines parameter name, usage, type, format, value, default and description.

Defines (name value) pairs, e.g.:

```
mvSampleAMI
  Reserved_Parameters
    Ignore_Bits   21

mvSampleAMI
  Model_Specific
    txtaps
      -2   0     (note: -2 is a name!)
```

Reserved_Parameters are pre-defined.

Model vendor can define any name and structure for Model_Specific parameters.

Executable Files

Generated by IC vendors. API has three C-calling convention function calls:

AMI_Init (char * params_in, …)

AMI_GetWave (waveform_in, waveform_out, clock_tick_array)

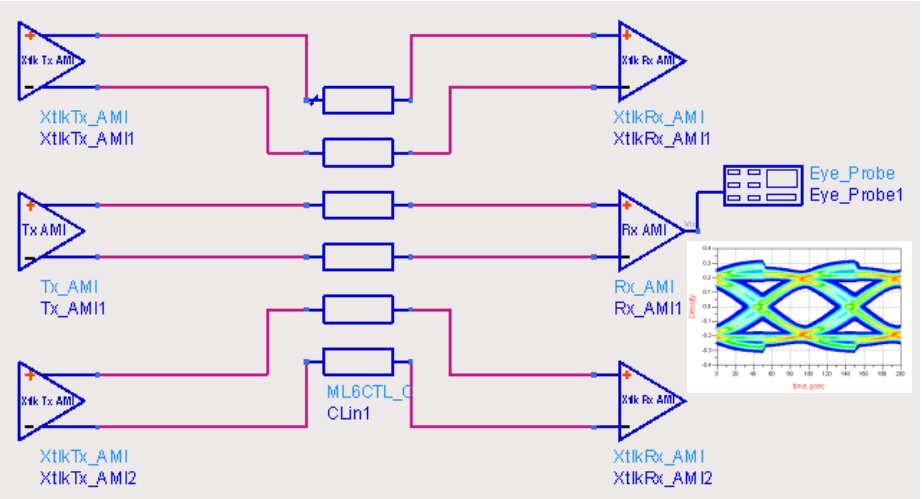
AMI_Close ( )

Note 1: Despite its name, AMI_Init() does much more than initialization

Note 2: Despite its position in between init and close, GetWave() is optional. It is only required if the model is non-linear and/or time varying (NLTV), as explained later.

Channel Simulator AMI schematic has:

- Exactly one Tx AMI
- Exactly one Rx AMI
- Zero or more Xtalk Tx AMI
- Zero of more Xtalk Rx AMI



AMI Crosstalk Transmitter – Aggressor Channel

Load regular AMI transmitter *.ibs file into a crosstalk AMI transmitter symbol

AMI Crosstalk Receiver – Aggressor Channel

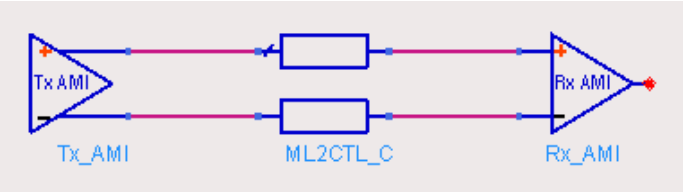
Load regular AMI receiver *.ibs file into a crosstalk AMI transmitter symbol

Agilent Technologies

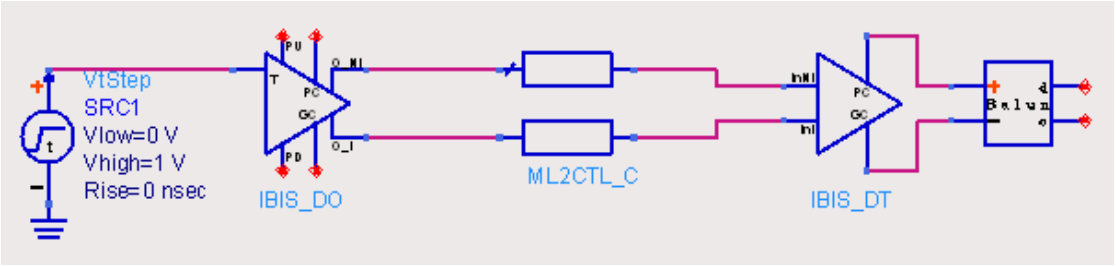
December 15, 2011

Under the Hood

Step 1: Characterization of Analog Channel Impulse Response



Internal schematic: Tx *.ibs analog circuit, physical channel, Rx *.ibs analog circuit and a Balun.



Transient is internally performed with a fixed 1V step stimulus input to Tx.

- Threshold crossing of the step function triggers Tx IBIS model to transmit a rise edge into the channel.

Step response is captured at balun's differential output. When response settles, transient is automatically terminated.

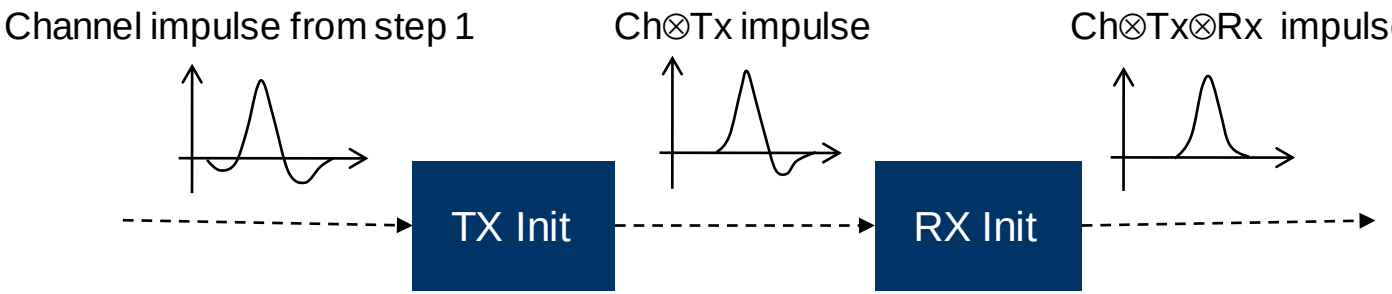
Impulse response is computed from derivative of step response.

Reflection by impedance mismatch, edge swing and shape, and common-differential mode conversion are included in channel impulse response.

- Tx IBIS analog model defines rise/fall edge swing, waveform and Tx impedance.
- Rx IBIS analog model defines Rx load/termination.

Agilent Technologies

Step 2a: Statistical Mode

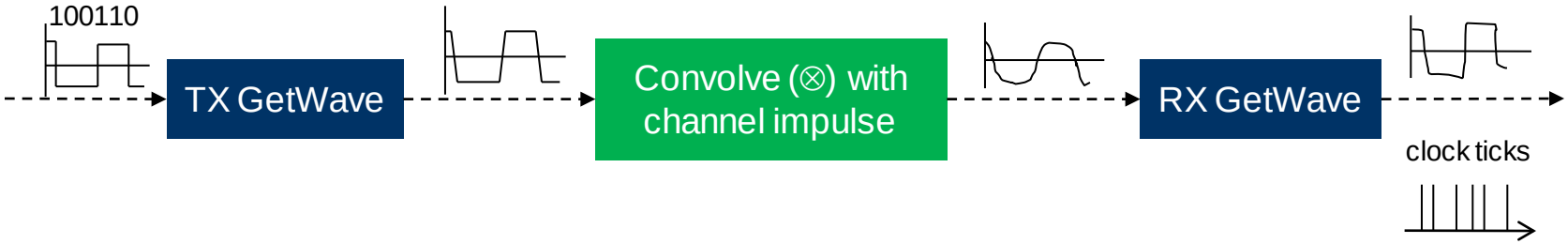


Although AMI_Init performs initialization such as memory allocation it also (optionally) modifies input impulse with an algorithm that returns a new impulse.
- If the end point is LTI, convolution (⊗) completely characterizes the effect of the device in the system.

Both Tx and Rx models must supply impulse modification functions in order to run statistical mode
- The standard allows 'dual-mode' models that have BOTH impulse modification AND AMI_GetWave(). WARNING: Check with your vendor about accuracy of such models in stat mode.

Final impulse response is used to calculate the eye directly
- No bit pattern runs through the model

Step 2b: Bit-by-bit (Time Domain) Simulation



Example case: Tx and Rx have GetWave algorithms

i. Tx_output = Tx_GetWave(bit_pattern(t))

ii. Rx_input = channel_impulse_from_step_1 ⊗ Tx_output

iii. Rx_output = Rx_GetWave(Rx_input)

Rx GetWave optionally returns clock tick values which specify Rx sample times.

Clock ticks are non-uniform in general due to sample time jitter in CDR

Clocks ticks (if any) are passed to ADS Eye Probe as explained later

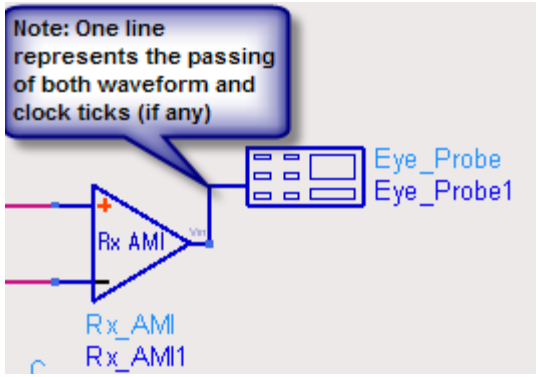
For other cases, we substitute AMI_Init convolution for the GetWave call.

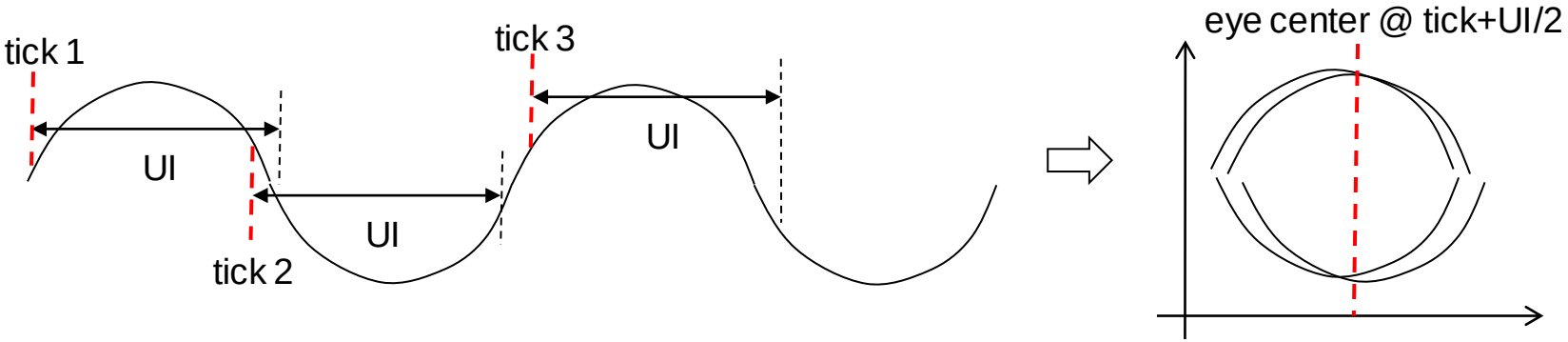
e.g. if Tx is LTI, modifies impulse and has no GetWave, then step i. is convolution using AMI_Init)

Rx Jitter Handling

Either Rx_Clock_PDF or clock ticks represent Rx sample time jitter in CDR. They are used in different situations.

1) When clock ticks **are** available (bit-by-bit mode **and** Rx GetWave returns clock ticks) waveform segments between [tick, tick+UI] are used to construct the eye to capture Rx sample time jitter. Eye is centered at tick+UI/2.





2) When clock ticks **are not** available, uniform sample times are used when constructing the eye. Rx_Clock_PDF is then convolved with eye in post-processing to account for Rx sample time jitter. Situations include

- statistical mode
- bit-by-bit mode but Rx model has no GetWave
- bit-by-bit mode but Rx GetWave does not return clock tick

Evaluate Our Products Today!

agilent.com/find/signal-integrity

pre- and post-layout bundle

pre-layout bundle

ADS Core

Transient Convolution Element

Layout Element

Momentum G2 Element

SystemVue AMI Modeling Kit

EMPro

Agilent Technologies

December 15, 2011

You're invited!

agilent.com/find/eesof-hsd-webcast

Overcome PI Challenges on Perforated Power/Ground Planes

January 19, 2012
7am PT (16:00CET) or 10am PT

Webcast Series

Register for one or all of the live sessions

Agilent Technologies