Simulation and Scopes

Proposal

P. Pupalaikis

Teledyne LeCroy

07/31/2021

Outline

Outline

Goal of this Presentation

- To present, from this user's perspective, the case for integrating simulation tools with the oscilloscope software
- To provide some examples that promote and facilitate this need.
- Present practical technical details regarding where things stand today.
- Make some recommendations on how to proceed.

Goal of this Presentation

- To present, from this user's perspective, the case for integrating simulation tools with the oscilloscope software
- To provide some examples that promote and facilitate this need.
- Present practical technical details regarding where things stand today.
- Make some recommendations on how to proceed.

Goal of this Presentation

- To present, from this user's perspective, the case for integrating simulation tools with the oscilloscope software
- To provide some examples that promote and facilitate this need.
- Present practical technical details regarding where things stand today.
- Make some recommendations on how to proceed.

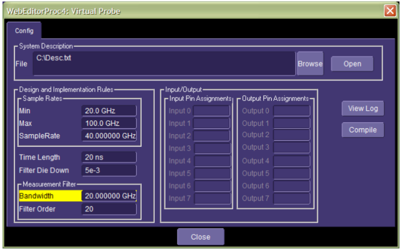
Goal of this Presentation

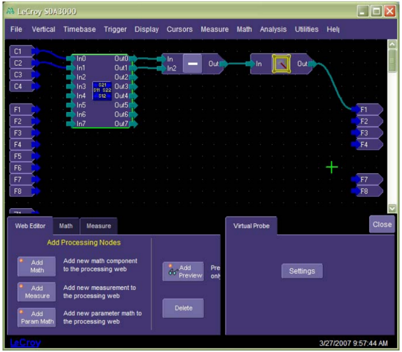
- To present, from this user's perspective, the case for integrating simulation tools with the oscilloscope software
- To provide some examples that promote and facilitate this need.
- Present practical technical details regarding where things stand today.
- Make some recommendations on how to proceed.

Outline

Virtual Probe

Virtual Probe is a component placed in the processing web. It processes a supplied netlist and converts this into the transfer matrices.

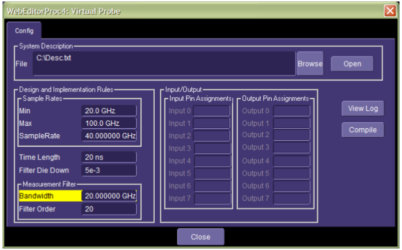


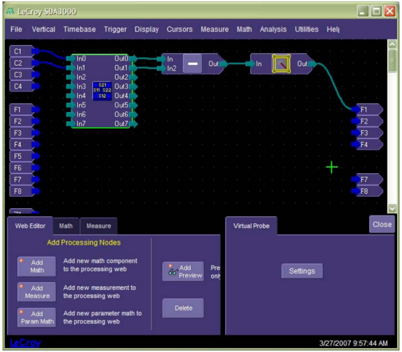


Virtual Probe has always had a need for a schematic driven user interface.

Virtual Probe

Virtual Probe is a component placed in the processing web. It processes a supplied netlist and converts this into the transfer matrices.

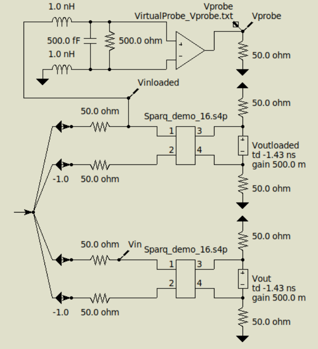




Virtual Probe has always had a need for a schematic driven user interface.

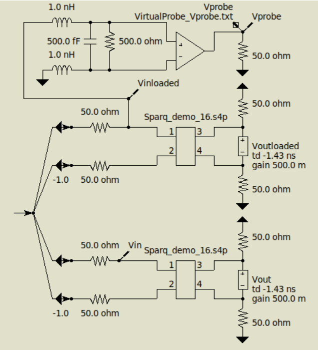
Virtual Probe within *SignalIntegrity*

Schematics to Netlist to Transfer Matrices to Waveforms



Virtual Probe within *SignalIntegrity*

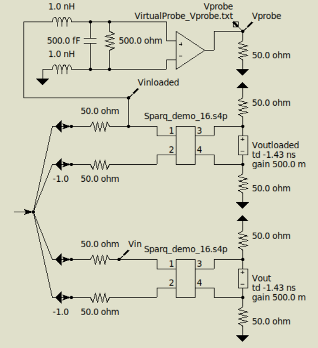
Schematics to Netlist to Transfer Matrices to Waveforms



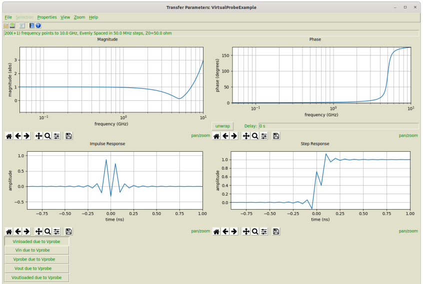

```
device Vout 4 voltagecontrolledvoltagesource 1.0
device Voutloaded 4 voltagecontrolledvoltagesource 1.0
device R8 2 R 50.0
device R9 2 R 50.0
device G5 1 ground
device G6 1 ground
device R6 1 R 50.0
device R7 1 R 50.0
device R10 1 R 50.0
device D3 3 opamp gain 1.0 zi 100.0e6 zo 0
voltageoutput Vinloaded R2 2
connect R2 2 D1 1 L1 1
connect L1 2 R1 2 D3 2 C1 2
connect L2 1 G1 1
connect D3 1 L2 2 C1 1 R1 1
connect R3 2 D1 2
connect Voutloaded 2 R5 1 D1 3
connect D1 4 Voutloaded 1 R4 1
connect G2 1 R2 1
stim m1 G2 1
connect R3 1 G3 1
stim m2 G3 1
meas D3 3
voltageoutput Vprobe D3 3
connect D3 3 R10 1
connect G5 1 R8 1
stim m3 G5 1
connect G6 1 R9 1
stim m4 G6 1
stimdef [[1.0], [-1.0], [1.0], [-1.0]]
```

Virtual Probe within *SignalIntegrity*

Schematics to Netlist to Transfer Matrices to Waveforms

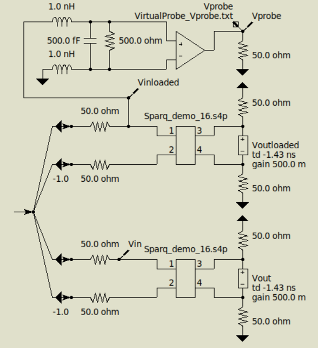



```
device Vout 4 voltagecontrolledvoltagesource 1.0
device Voutloaded 4 voltagecontrolledvoltagesource 1.0
device R8 2 R 50.0
device R9 2 R 50.0
device G5 1 ground
device G6 1 ground
device R6 1 R 50.0
device R7 1 R 50.0
device R10 1 R 50.0
device D3 3 opamp gain 1.0 zi 100.0e6 zo 0
voltageoutput Vinloaded R2 2
connect R2 2 D1 1 L1 1
connect L1 2 R1 2 D3 2 C1 2
connect L2 1 G1 1
connect D3 1 L2 2 C1 1 R1 1
connect R3 2 D1 2
connect Voutloaded 2 R5 1 D1 3
connect D1 4 Voutloaded 1 R4 1
connect G2 1 R2 1
stim m1 G2 1
connect R3 1 G3 1
stim m2 G3 1
meas D3 3
voltageoutput Vprobe D3 3
connect D3 3 R10 1
connect G5 1 R8 1
stim m3 G5 1
connect G6 1 R9 1
stim m4 G6 1
stimdef [[1.0], [-1.0], [1.0], [-1.0]]
```

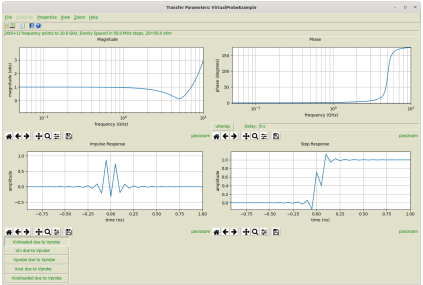


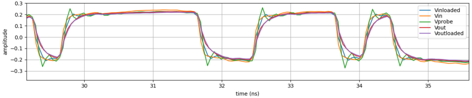
Virtual Probe within *SignalIntegrity*

Schematics to Netlist to Transfer Matrices to Waveforms




```
device Vout 4 voltagecontrolledvoltagesource 1.0
device Voutloaded 4 voltagecontrolledvoltagesource 1.0
device R8 2 R 50.0
device R9 2 R 50.0
device G5 1 ground
device G6 1 ground
device R6 1 R 50.0
device R7 1 R 50.0
device R10 1 R 50.0
device D3 3 opamp gain 1.0 zi 100.0e6 zo 0
voltageoutput Vinloaded R2 2
connect R2 2 D1 1 L1 1
connect L1 2 R1 2 D3 2 C1 2
connect L2 1 G1 1
connect D3 1 L2 2 C1 1 R1 1
connect R3 2 D1 2
connect Voutloaded 2 R5 1 D1 3
connect D1 4 Voutloaded 1 R4 1
connect G2 1 R2 1
stim m1 G2 1
connect R3 1 G3 1
stim m2 G3 1
meas D3 3
voltageoutput Vprobe D3 3
connect D3 3 R10 1
connect G5 1 R8 1
stim m3 G5 1
connect G6 1 R9 1
stim m4 G6 1
stimdef [[1.0], [-1.0], [1.0], [-1.0]]
```





What to do for Virtual Probe Integration

- Have *SignalIntegrity* generate netlists?
  - No. The format is different and the capability of SignalIntegrity insofar as devices, probes, etc. is much more extensive than the scope.
- Have *SignalIntegrity* generate waveforms?
  - No. The processing power of the scope far exceeds that of *SignalIntegrity*.
- Have *SignalIntegrity* analyze waveforms?
  - No. To duplicate the analysis capability of the scope is folly.

Transfer Matrices

SignalIntegrity should generate transfer matrices for the scope. It's fast enough and the transfer matrices used in the scope are identical to those generated in *SignalIntegrity*.

What to do for Virtual Probe Integration

- Have *SignalIntegrity* generate netlists?
  - No. The format is different and the capability of SignalIntegrity insofar as devices, probes, etc. is much more extensive than the scope.
- Have *SignalIntegrity* generate waveforms?
  - No. The processing power of the scope far exceeds that of *SignalIntegrity*.
- Have *SignalIntegrity* analyze waveforms?
  - No. To duplicate the analysis capability of the scope is folly.

Transfer Matrices

SignalIntegrity should generate transfer matrices for the scope. It's fast enough and the transfer matrices used in the scope are identical to those generated in *SignalIntegrity*.

What to do for Virtual Probe Integration

- Have *SignalIntegrity* generate netlists?
  - No. The format is different and the capability of SignalIntegrity insofar as devices, probes, etc. is much more extensive than the scope.
- Have *SignalIntegrity* generate waveforms?
  - No. The processing power of the scope far exceeds that of *SignalIntegrity*.
- Have *SignalIntegrity* analyze waveforms?
  - No. To duplicate the analysis capability of the scope is folly.

Transfer Matrices

SignalIntegrity should generate transfer matrices for the scope. It's fast enough and the transfer matrices used in the scope are identical to those generated in *SignalIntegrity*.

What to do for Virtual Probe Integration

- Have *SignalIntegrity* generate netlists?
  - No. The format is different and the capability of SignalIntegrity insofar as devices, probes, etc. is much more extensive than the scope.
- Have *SignalIntegrity* generate waveforms?
  - No. The processing power of the scope far exceeds that of *SignalIntegrity*.
- Have *SignalIntegrity* analyze waveforms?
  - No. To duplicate the analysis capability of the scope is folly.

Transfer Matrices

SignalIntegrity should generate transfer matrices for the scope. It's fast enough and the transfer matrices used in the scope are identical to those generated in *SignalIntegrity*.

What to do for Virtual Probe Integration

- Have *SignalIntegrity* generate netlists?
  - No. The format is different and the capability of SignalIntegrity insofar as devices, probes, etc. is much more extensive than the scope.
- Have *SignalIntegrity* generate waveforms?
  - No. The processing power of the scope far exceeds that of *SignalIntegrity*.
- Have *SignalIntegrity* analyze waveforms?
  - No. To duplicate the analysis capability of the scope is folly.

Transfer Matrices

SignalIntegrity should generate transfer matrices for the scope. It's fast enough and the transfer matrices used in the scope are identical to those generated in *SignalIntegrity*.

What to do for Virtual Probe Integration

- Have *SignalIntegrity* generate netlists?
  - No. The format is different and the capability of SignalIntegrity insofar as devices, probes, etc. is much more extensive than the scope.
- Have *SignalIntegrity* generate waveforms?
  - No. The processing power of the scope far exceeds that of *SignalIntegrity*.
- Have *SignalIntegrity* analyze waveforms?
  - No. To duplicate the analysis capability of the scope is folly.

Transfer Matrices

SignalIntegrity should generate transfer matrices for the scope. It's fast enough and the transfer matrices used in the scope are identical to those generated in *SignalIntegrity*.

What to do for Virtual Probe Integration

- Have *SignalIntegrity* generate netlists?
  - No. The format is different and the capability of SignalIntegrity insofar as devices, probes, etc. is much more extensive than the scope.
- Have *SignalIntegrity* generate waveforms?
  - No. The processing power of the scope far exceeds that of *SignalIntegrity*.
- Have *SignalIntegrity* analyze waveforms?
  - No. To duplicate the analysis capability of the scope is folly.

Transfer Matrices

SignalIntegrity should generate transfer matrices for the scope. It's fast enough and the transfer matrices used in the scope are identical to those generated in *SignalIntegrity*.

Outline

Transfer Matrices Processing

Transfer matrices are an array of matrices, each matrix representing the input to output frequency response at a given frequency.

$$\begin{pmatrix} H_{[n]_{11}} & H_{[n]_{12}} & \cdots & H_{[n]_{1I}} \\ H_{[n]_{21}} & H_{[n]_{22}} & \cdots & H_{[n]_{2I}} \\ \vdots & \vdots & \ddots & \vdots \\ H_{[n]_{O1}} & H_{[n]_{O2}} & \cdots & H_{[n]_{OI}} \end{pmatrix} \cdot \begin{pmatrix} VI_{[n]_1} \\ VI_{[n]_2} \\ \vdots \\ VI_{[n]_I} \end{pmatrix} = \begin{pmatrix} VO_{[n]_1} \\ VO_{[n]_2} \\ \vdots \\ VO_{[n]_O} \end{pmatrix},$$

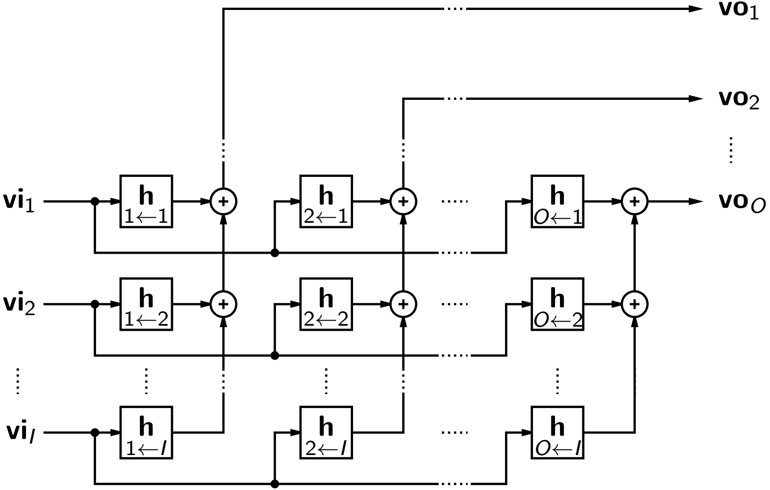
$$VO_{[n]_o} = \sum_{i=1}^{I} VI_{[n]_i} \cdot H_{[n]_{oi}}.$$

Expressed in the time domain[1]:

$$vo_o = \sum_{i=1}^{I} \left(vi_i * h_{oi} \right).$$

[1]The symbol $*$ represents the convolution operation

Transfer Matrices Filter Structure



Transfer Matrices

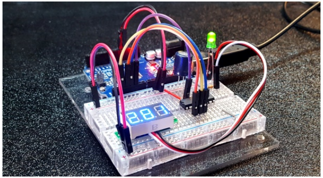
Interesting Facts

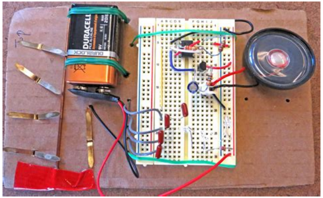
- Virtual Probe (in the scope) generates these from the supplied netlist.
- Virtual Probe caches these in both the time and frequency domain.
- The structure of the transfer matrices is identical for virtual probing and for simulation.

The underlying code in Virtual Probe is already in a position to perform simulations completely within the scope – All that is needed is to supply the transfer matrices. These are easily supplied through the *SignalIntegrity* software.

Outline

Simulation in the Design Environment

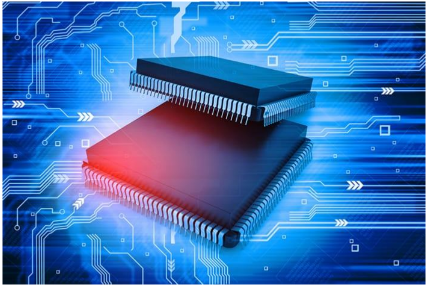




- It used to be common to breadboard circuits. In these cases, test equipment would be employed to debug these prototypes
- In today's world, the design of the prototype involves simulation. Not only is the test equipment cut out of this development phase, but the analysis software employed in high-end equipment, such as high-speed oscilloscopes cannot be used

Oscilloscope software is trapped in the instrument and withheld from simulation.

IC Design



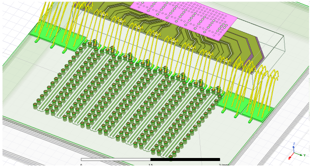
- IC design makes the situation worse – they are only simulated.
- The simulation is long, expensive, and the capability for analyzing the results is particularly poor.

Furthermore, when evaluating design choices and trade-offs, a system level view is desired, necessitating simulation of the constituent elements of the system together to determine final system performance. It is common to refine simulations more and more as the design progresses to get an ever increasingly accurate view of the system.

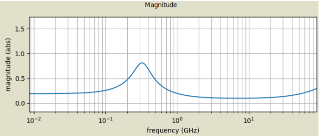
Outline

A PDN Simulation

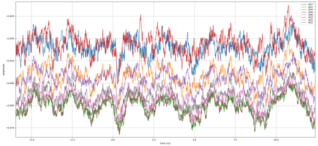
A PDN model is created and simulated in HFSS to determine the impedance vs. frequency, the model is fitted (to identify design points of control, such as wire bond inductance), and a transient simulation shows the supply noise.



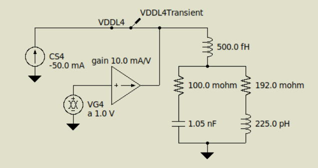
HFSS Model



PDN Impedance



Transient Supply Noise



PDN Approximate Circuit (with Transients)

Missing from the PDN Simulation

What went well:

- HFSS does a great job of producing the simulation (the Z-parameters) of the PDN from geometry and material properties.
- The *SignalIntegrity* software allowed for fitting and transient simulation (waveform generation).

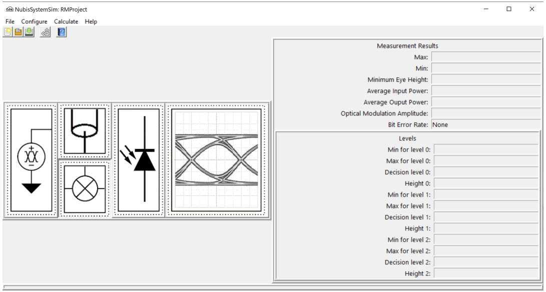
What is missing?

The Analysis!

The user is left with some waveforms that are begging for parameters and measurements.

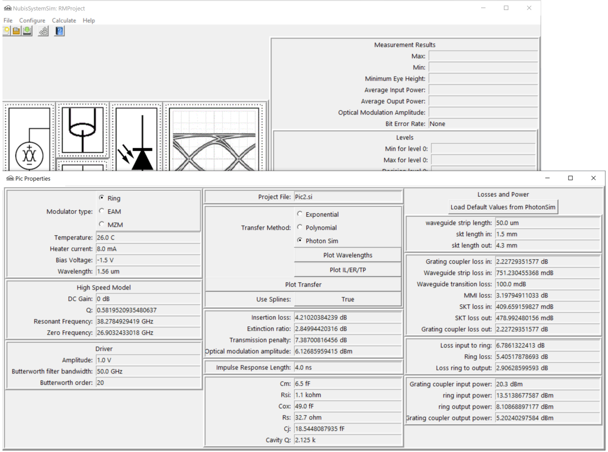
Outline

Nubis System Simulator



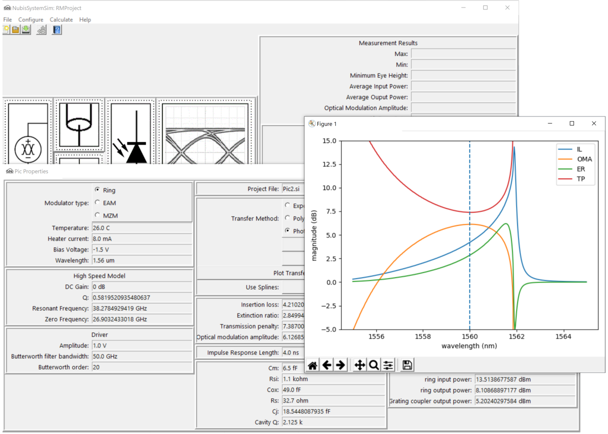
- An end-to-end link simulation environment.
- Allows configuration of various modulator types and controls the simulation environment.
- Enables insight into interactions between electronic and optical components.

Nubis System Simulator



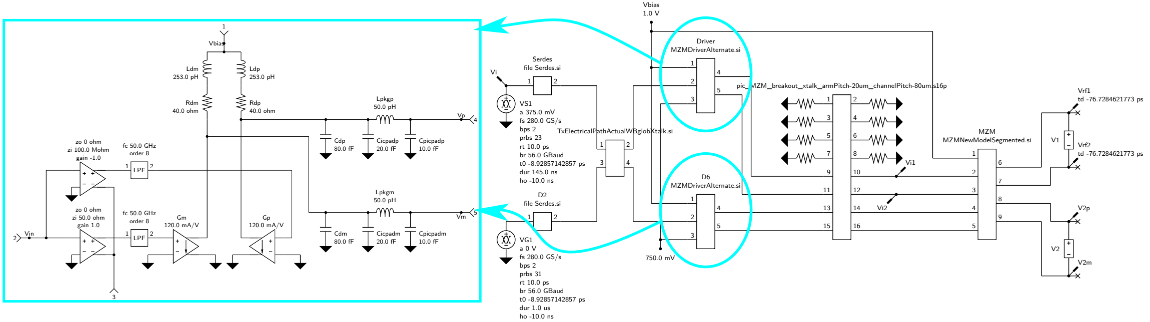
- An end-to-end link simulation environment.
- Allows configuration of various modulator types and controls the simulation environment.
- Enables insight into interactions between electronic and optical components.

Nubis System Simulator



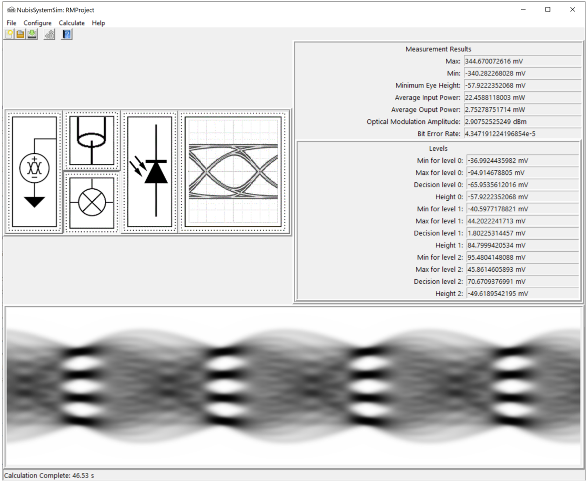
- An end-to-end link simulation environment.
- Allows configuration of various modulator types and controls the simulation environment.
- **Enables insight into interactions between electronic and optical components.**

Transmitter Internals



- The transmitter is a linear, hierarchical model containing s-parameters (simulated in HFSS) and models of the optical modulators.
- A nonlinear transfer function for the electrical to optical modulation is applied to the differential measurements at the end of the schematic.

Eye Diagram Testing



The simulator is used to specify the performance of various sub-blocks and to incorporate increased reality as sub-blocks are designed and separately simulated.

The final goal is to verify the BER as a function of input optical power and resulting optical modulation amplitude.

Outline

Compliance Testing

Some facts:

- The optical link simulator works really great in simulating what we want for the electronics and optical modulators. Nothing more is really desired (except higher speed, frankly).
- The optical link analysis is really painful.
  - We need to pore over and understand standards specification requirements.
  - We must implement the compliance tests in software.

We need to essentially write the same compliance test and serial data analysis tools that would be in an oscilloscope

Compliance Testing

Some facts:

- The optical link simulator works really great in simulating what we want for the electronics and optical modulators. Nothing more is really desired (except higher speed, frankly).
- The optical link analysis is really painful.
  - We need to pore over and understand standards specification requirements.
  - We must implement the compliance tests in software.

We need to essentially write the same compliance test and serial data analysis tools that would be in an oscilloscope

Recommendation

- It's obvious that the same test and analysis tools are needed during the design phase (with simulators) and the prototype phase (with test and measurement instruments).
- It's also a shame that huge amounts of work go into the development of the oscilloscope analysis software that only gets to be used with oscilloscope acquired waveforms.
- There's an opportunity to get more for the oscilloscope software than just using it to sell another box.
- The recommendation would be to integrate this software somehow with simulation tools and make this software available to customers.

Outline

What to do

The end goal should be to integrate the scope software with the various commercial simulation tools. This means:

- Converting or reading waveforms directly produced by these tools in their formats.
- Kicking off simulations from within the oscilloscope.
- Having simulation tools write results directly into oscilloscope memory.
- Integrating specifically QualiPhy with simulators.
- **This should be done for protocol analysis, as well** (although the details would be quite different).

This may (and will preferably) mean collaboration with simulator vendors.

How to start

Start with the simulation software you already own – *SignalIntegrity*.

- *SignalIntegrity* produces transfer matrices just like those used in Virtual Probe.
- You can kill two birds with one stone by allowing *SignalIntegrity* produced transfer matrices to be used directly:
  - you'd get better virtual probe integration and you'd get a simulator that basically works automatically within the scope.
  - The simulator within the scope, while being linear only, would be very fast and powerful.

Commercial Aspects

- *SignalIntegrity* would be bundled with the oscilloscope software and installed simultaneously.
- *SignalIntegrity* would be free, but LeCroy would charge for the option that allows good scope integration.
  - let users, for free, run *SignalIntegrity* in it's slow manner and store waveforms to be read by the scope.
  - Running the fast way through transfer matrices processed by the scope would be controlled by scope options.
  - The compliance tests would be further charges.

A Non-commercial Aspect

LeCroy would be able to integrate this simulator into its testing environment that would allow for much better testing of the analysis capability and of QualiPhy.

Commercial Aspects

- *SignalIntegrity* would be bundled with the oscilloscope software and installed simultaneously.
- *SignalIntegrity* would be free, but LeCroy would charge for the option that allows good scope integration.
  - let users, for free, run *SignalIntegrity* in it's slow manner and store waveforms to be read by the scope.
  - Running the fast way through transfer matrices processed by the scope would be controlled by scope options.
  - The compliance tests would be further charges.

A Non-commercial Aspect

LeCroy would be able to integrate this simulator into its testing environment that would allow for much better testing of the analysis capability and of QualiPhy.

Outline

Summary

1. The analysis software in the scope is needed by engineers during the design and simulation phase as much as when used with acquired waveforms.
2. LeCroy is currently missing out on ways to leverage the oscilloscope software to earn more money.
3. Customers are currently missing out regarding options for analyzing their simulator generated waveforms with much better tools than are currently available.
4. LeCroy already owns simulation software in the form of the open-source *SignalIntegrity* software.
5. A plan should be made and executed upon to integrate simulation tools with the oscilloscope, starting with the *SignalIntegrity* software.

What are we waiting for? Let's get going!

Summary

1. The analysis software in the scope is needed by engineers during the design and simulation phase as much as when used with acquired waveforms.
2. LeCroy is currently missing out on ways to leverage the oscilloscope software to earn more money.
3. Customers are currently missing out regarding options for analyzing their simulator generated waveforms with much better tools than are currently available.
4. LeCroy already owns simulation software in the form of the open-source *SignalIntegrity* software.
5. A plan should be made and executed upon to integrate simulation tools with the oscilloscope, starting with the *SignalIntegrity* software.

What are we waiting for? Let's get going!

Summary

1. The analysis software in the scope is needed by engineers during the design and simulation phase as much as when used with acquired waveforms.
2. LeCroy is currently missing out on ways to leverage the oscilloscope software to earn more money.
3. Customers are currently missing out regarding options for analyzing their simulator generated waveforms with much better tools than are currently available.
4. LeCroy already owns simulation software in the form of the open-source *SignalIntegrity* software.
5. A plan should be made and executed upon to integrate simulation tools with the oscilloscope, starting with the *SignalIntegrity* software.

What are we waiting for? Let's get going!

Summary

1. The analysis software in the scope is needed by engineers during the design and simulation phase as much as when used with acquired waveforms.
2. LeCroy is currently missing out on ways to leverage the oscilloscope software to earn more money.
3. Customers are currently missing out regarding options for analyzing their simulator generated waveforms with much better tools than are currently available.
4. LeCroy already owns simulation software in the form of the open-source *SignalIntegrity* software.
5. A plan should be made and executed upon to integrate simulation tools with the oscilloscope, starting with the *SignalIntegrity* software.

What are we waiting for? Let's get going!

Summary

1. The analysis software in the scope is needed by engineers during the design and simulation phase as much as when used with acquired waveforms.
2. LeCroy is currently missing out on ways to leverage the oscilloscope software to earn more money.
3. Customers are currently missing out regarding options for analyzing their simulator generated waveforms with much better tools than are currently available.
4. LeCroy already owns simulation software in the form of the open-source *SignalIntegrity* software.
5. A plan should be made and executed upon to integrate simulation tools with the oscilloscope, starting with the *SignalIntegrity* software.

What are we waiting for? Let's get going!

Summary

1. The analysis software in the scope is needed by engineers during the design and simulation phase as much as when used with acquired waveforms.
2. LeCroy is currently missing out on ways to leverage the oscilloscope software to earn more money.
3. Customers are currently missing out regarding options for analyzing their simulator generated waveforms with much better tools than are currently available.
4. LeCroy already owns simulation software in the form of the open-source *SignalIntegrity* software.
5. A plan should be made and executed upon to integrate simulation tools with the oscilloscope, starting with the *SignalIntegrity* software.

What are we waiting for? Let's get going!

Thank You!