IEEE802.3ap

10.3125Gbps NRZ Simulation Results Using "StatEye" and "Signal to Interference Model" on Cascaded Channel Components

Shannon Sawyer and Charles Moore
January 24, 2005

Agilent Technologies

Presentation Outline

- Motivation
- Tool Comparisons
- Simulated Channel Specifics
- StatEye Overview and Setup
- StatEye Results
- Signal to Interference Model Overview and Setup
- Signal to Interference Model Results
- Result Correlations
- Conclusions

Motivation

A tool is needed for the committee to create and validate a channel model with various signaling methods and be useful for board vendors to verify channel compliance.

1. Should have:
   - Ability to use S-parameters of IC, package, channel, capacitors, and crosstalk
   - Ability to evaluate loss, impedance, and reflections across frequency span
   - Ability to emulate modern system features i.e. TX emphasis, RX FFE, RX DFE, etc.
   - Ability to provide BER

2. Currently available examples:
   - Open Source: StatEye
   - Commonly Available: HSPICE, ADS, Matlab, Perl
   - Proprietary: IBM's DFECDR tool, Agilent's Signal to Interference Model

Tool Comparisons

StatEye

- Open source Matlab scripts for ISI and jitter modeling of communication link.
- Emulates TX and RX systems against channel S-parameters by building and evaluating a pulse model through PDFs. These PDFs are used to statistically generate BER contours in the shape of an eye.
- Developed by Anthony Sanders, Edoardo Prete, Alex Deas, and Bob Davidov
- Code and forum support available at www.stateye.org

Signal to Interference (SigInt) Model

- Perl scripts that call HSPICE and C functions to evaluate the reflections in end to end channels using a generated pulse response against S-parameters. Output are tap weights, various signal gains, self interference, re-reflected interference, SDD21 margin, and signal / noise relationships.
- Developed by Charles Moore and presented to Ad Hocs. moore_01_1104

Simulated Channel Specifics

Cascaded Channel Components

- Tyco's 7 thru channels from the database each with (qty4) xtalk_rev6.s4p for xtalk. Cases 6&7 have worst stub effect (125mil stubs).
- Intel's 18 thru channels from the database each with NEXT1&3 and FEXT1&2 for xtalk. "T" channels have worst stub effect (175mil stubs).
- Intel's Spec_RL_cap_like.s4p package model (one for each end of channel)
- Agilent's 4_7nf.s4p AC coupling capacitor
- For each of the cases, these 4 channel component's .s4p files were cascaded into a single .s4p as described in sawyer_m2_0105 and run with each simulator.
- TP4-TP5 AC coupling cap vias were not available at simulation time to complete the full channel.
- Simulations were performed with and without xtalk

StatEye Overview

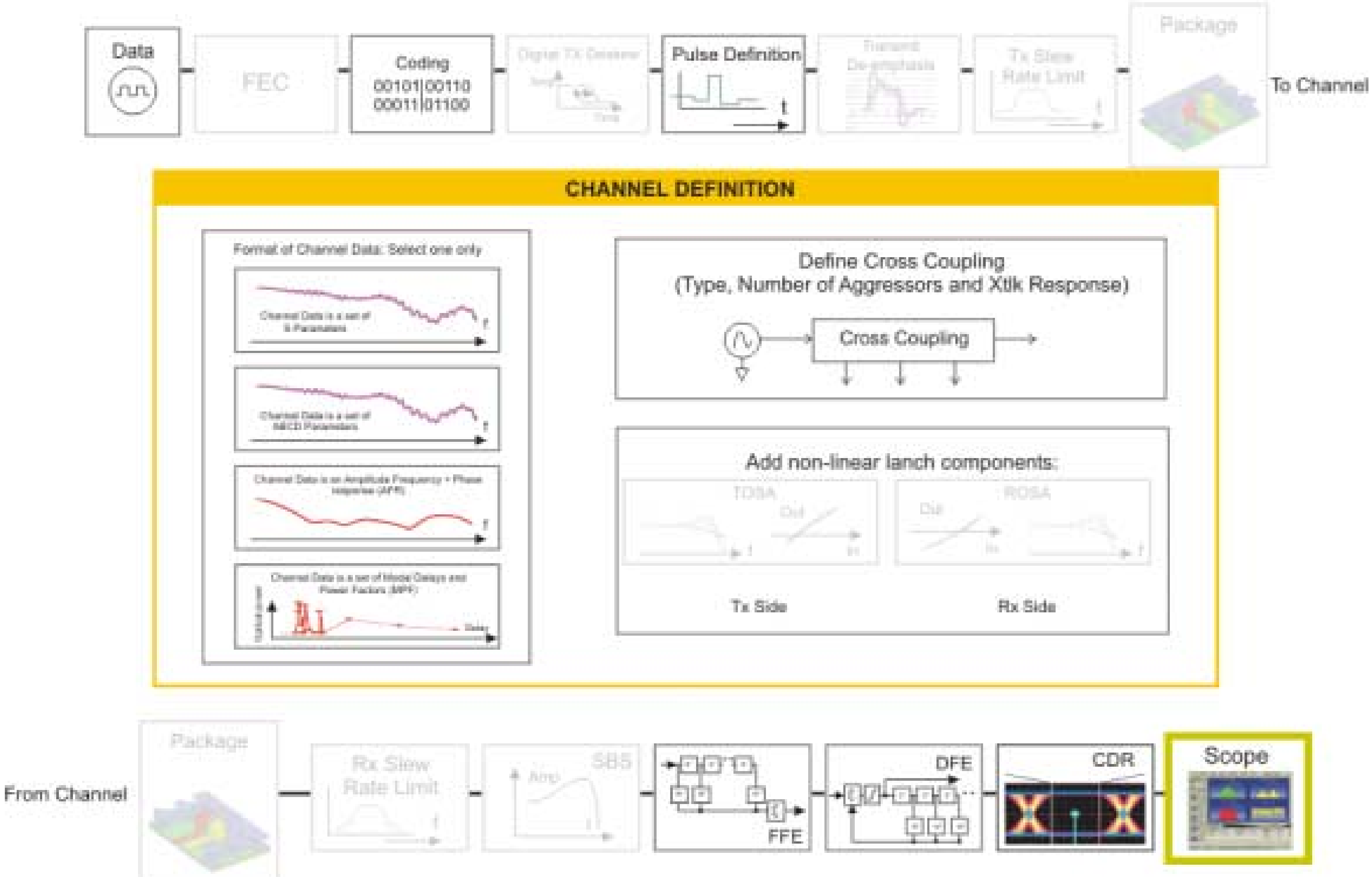
Steps to Generate a StatEye

(Paraphrased from ghiasi_01_09_04)

1. Input S-parameters of channel
2. Generate pulse response from S-parameters
3. DFT of TX pulse, combine with SDD21, then IDFT, sampled at bit rate, for RX pulse at x(0) to generate first conditional vertical PDF
4. Iterate #3 for family of values of x(0+bit time)
5. Generate horizontal PDF from RJ and DJ values.
6. Choose vertical line at sample point, then combine #4 and #5 to generate PDF of vertical line (provides BER height contour)
7. Iterate #6 for family of values of x (provides BER width contour)

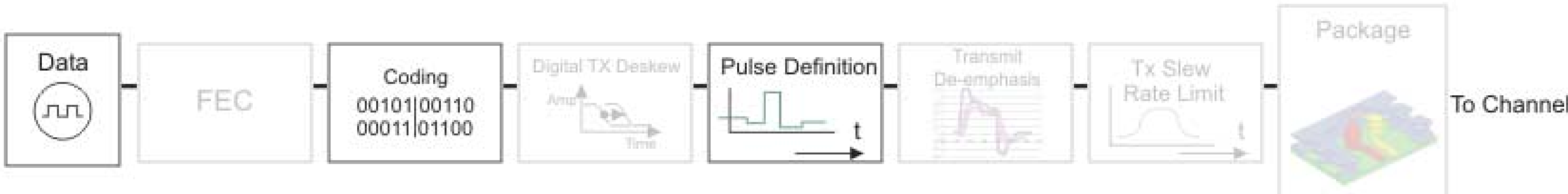
StatEye Overview

Version 3.0E GUI



StatEye Setup

Transmitter



DataGenerator:

- Rate = 10.3125Gbps
- DJ = 0.15UI pp
- RJ = 0.15UI pp
- BER = 1E-15
- PNA = []
- DCD = 0.05
- Pulse
  - Ideal [x]
  - Gaussian []
  - Spectrum []

Coding:

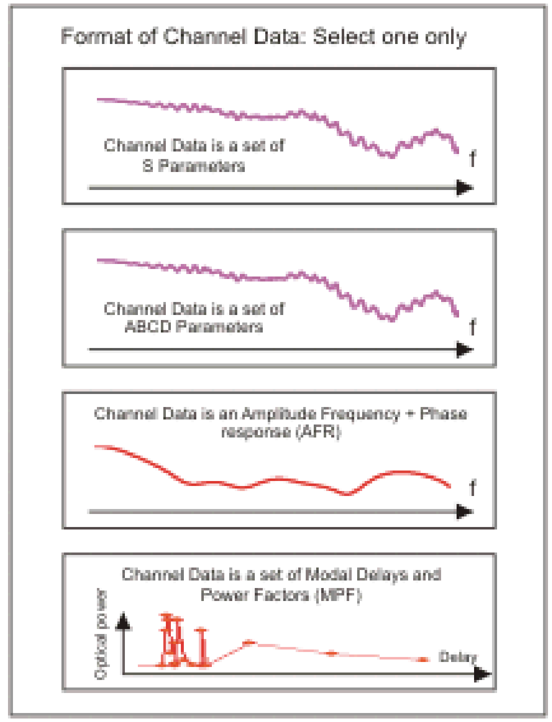
- PAM = PAM2
- Coding Spectrum = Raw NRZ

PulseDefinition:

- txpre = []
- txpost = [-0.1,-0.1]
- vstart = [-0.4, -0.3, -0.2]
- vend = [0,0,0]
- vstep = [0.1, 0.05, 0.025]

StatEye Setup

Channel



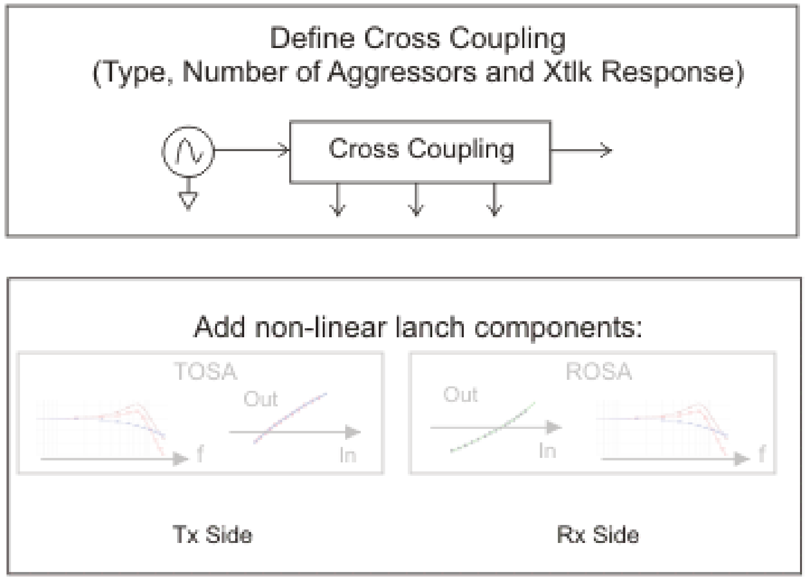
P4_S4P: [x]

4Port S4P Params: cascade.s4p

pkg.s4p → channel.s4p → cap.s4p → pkg.s4p = cascade.s4p

StatEye Setup

CrossTalk



Summation:

- **Linear [x]**
- **RSS []**

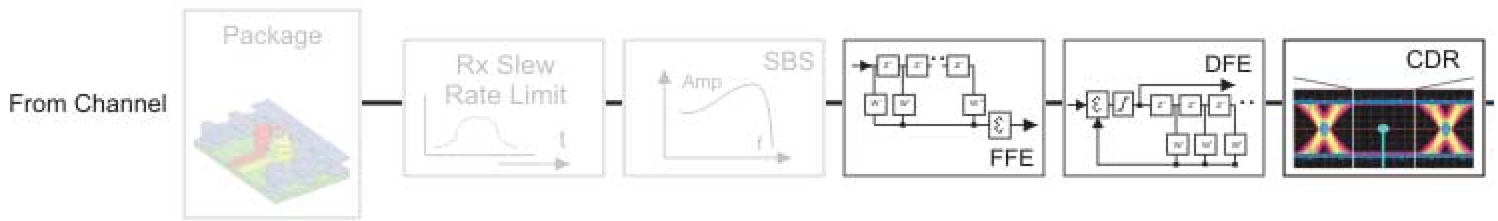
Number of Aggressors: 4

Aggressor: s4p [x]

s4p aggressor data: next.s4p, fext.s4p,....

StatEye Setup

Receiver



FFE: Taps = 4

DFE: Taps = 4

CDR:

- **Single PLL [X]**
- **Dual PLL []**
- **16x Oversampling Digital CDR**
- **Eye enlargement:**
  - **Off [x]**
  - **Acuid []**
  - **Samsung []**
  - **Broadcom []**

StatEye Setup

Output Data

Plot: TX Filter []

Plot: PADs []

Plot: S11, S12, ... []

Plot: ABCD Parameters []

Plot: Transfer Function []

Plot: SDD11, SDD12, ... []

Plot: Measured and Interpolated Data []

Plot: Modal Delays and Power Coupling Coefficients []

Plot: Amplitude Frequency Response []

Plot: TOSA []

Plot ROSA []

Plot: RX Filter []

Plot: Pulse Response [x]

Plot: PNA []

Plot: Statistical Eye [x]

Eye Contour Type: Q []

Eye Contour Type: BER [x]

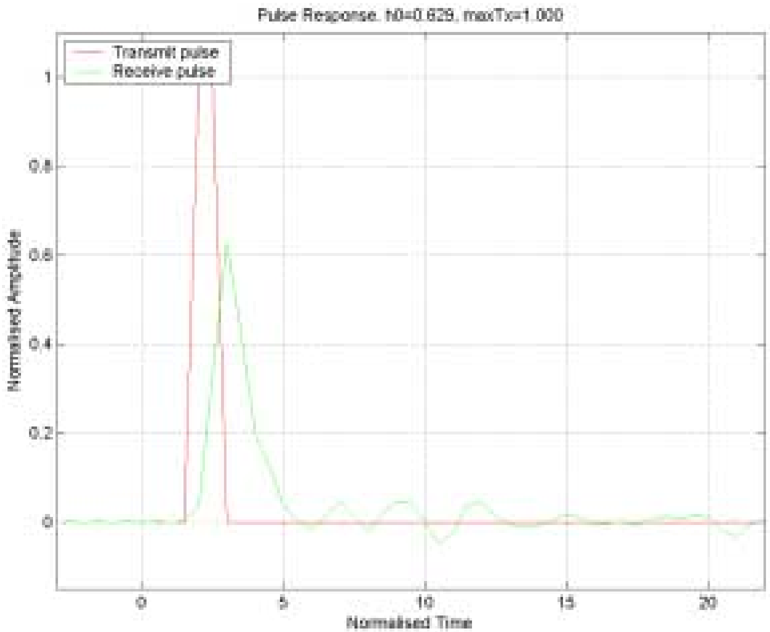
Plot: Bathtub Curve []

BER of min contour: 1E-15

Results File : .. / Results / Example.mat

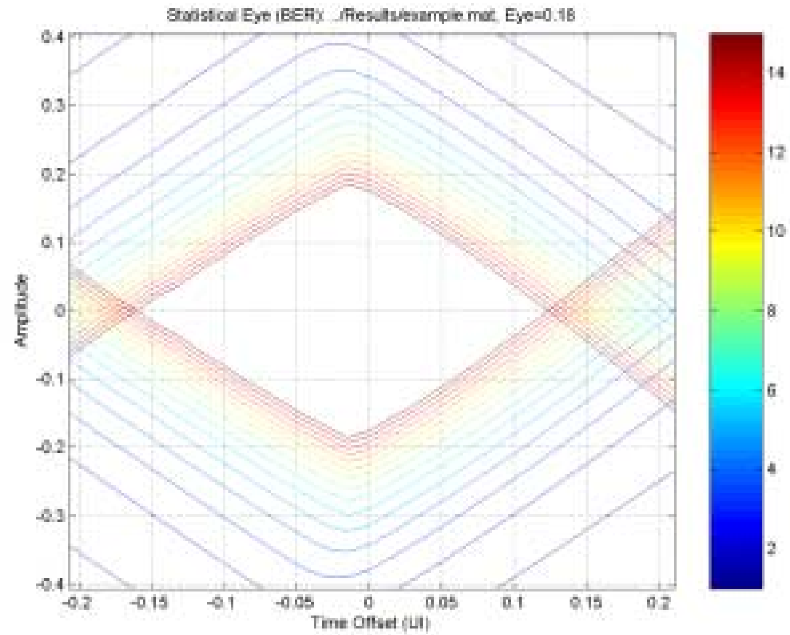
StatEye Example Output

Pulse Response



- TX input pulse (red) and pulse seen at the RX (green)
- TX normalized amplitude
- UI normalized time

StatEye



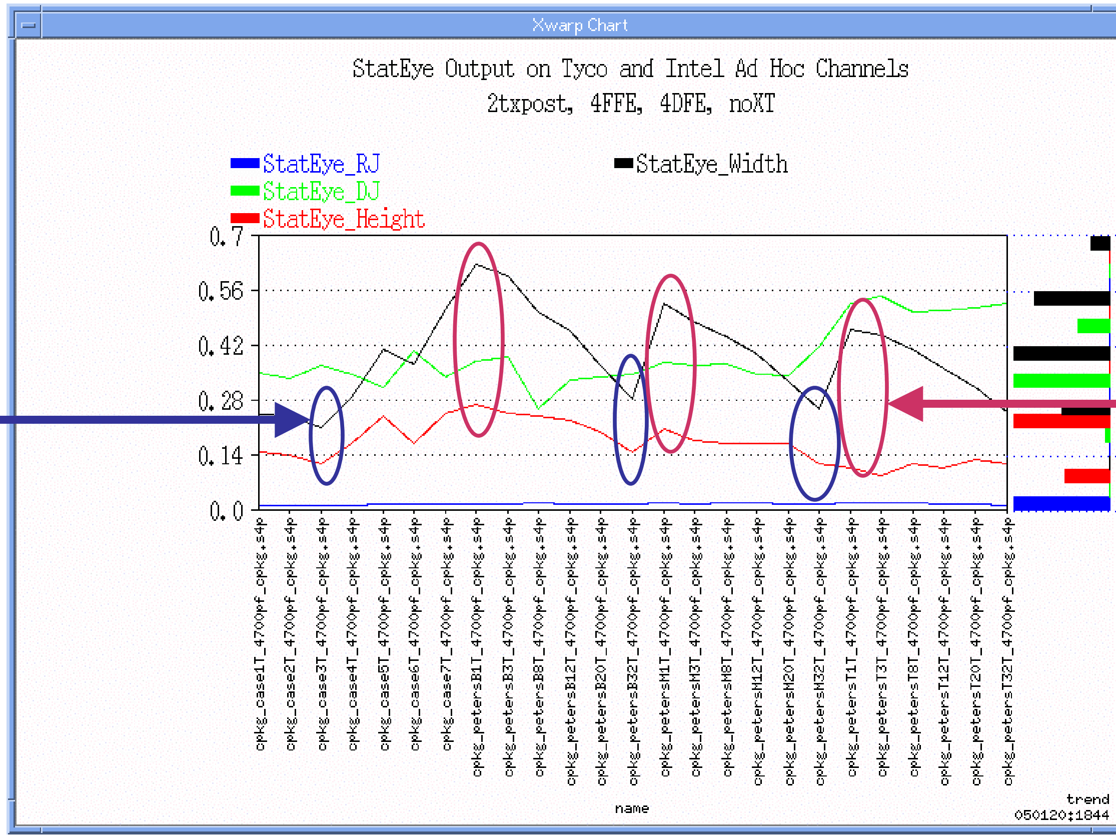
- BER contours (inner contour is 1E-15)
- Min contour amplitude represents output eye in Vsepp normalized to TX pulse amplitude. I.e.. 0.12*1.0V=120mVsepp or 12% of TX input pulse amplitude.

StatEye Simulation Results

EyeHeight, EyeWidth, DJ, RJ vs. Tyco and Intel channels (noXT)

Each channel sim took ~5 minutes (noXT) ~8minutes (XT)

longer channels have smaller EyeWidths and EyeHeights



shorter channels have larger EyeWidths and EyeHeights

EyeWidths and EyeHeights correlate with each other

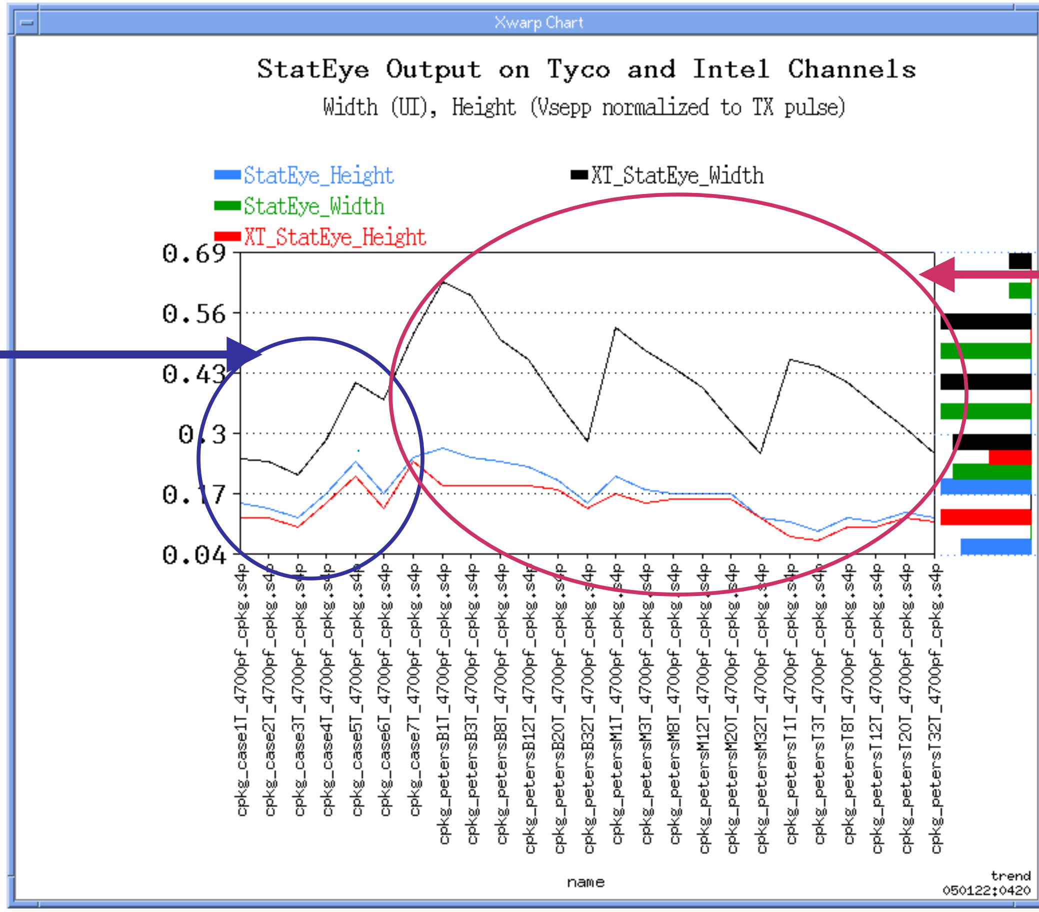
StatEye Simulation Results continued...

EyeHeight, EyeWidth vs. Tyco and Intel channels (XT and noXT)

Tyco channels

EyeHeight
noXT
12%-25%
XT
10%-24%

EyeWidth
noXT
212mUI-515mUI
XT
212mUI-515mUI



Intel channels

EyeHeight
noXT
9%-27%
XT
7%-19%

EyeWidth
noXT
250mUI-629mUI
XT
250mUI-629mUI

Comparison of Tyco Case5&6 and Intel B1,M1,T1 show stub length affects
Intel has more change in EyeHeight with XT, and EyeWidth has no change

Signal Interference Model (SigInt) Overview

Perl and C functions

(Paraphrased from moore_01_1104)

1. Read in S-parameters of channel components
2. Average SDD21 around Nyquist rate to get “AC gain”
3. HSPICE generate 1 bit pulse into ideal termination to get “pulse gain”
4. Optimize TX tap coefficients with channel.
5. Integrate the pulse response tail 2 bit times after cursor to get “self interference”
6. Use

$$\int_{0}^{15G} \left| chS11 \right| \cdot \left| \Gamma tx \right| \cdot \left| chS21 \right| \cdot 2 \cdot \frac{\sin(\pi\ f \cdot \tau)}{\pi\ f}\ df$$

   to determine “re-reflect interference” for TX and a similar integral for RX

7. Linearly sum in worst case interfering patterns for xtalk as in moore_01_0704
8. Output the results and analyze Signal and Noise data to create a channel gain limit or an eye height / width limit.

SigInt Model Setup

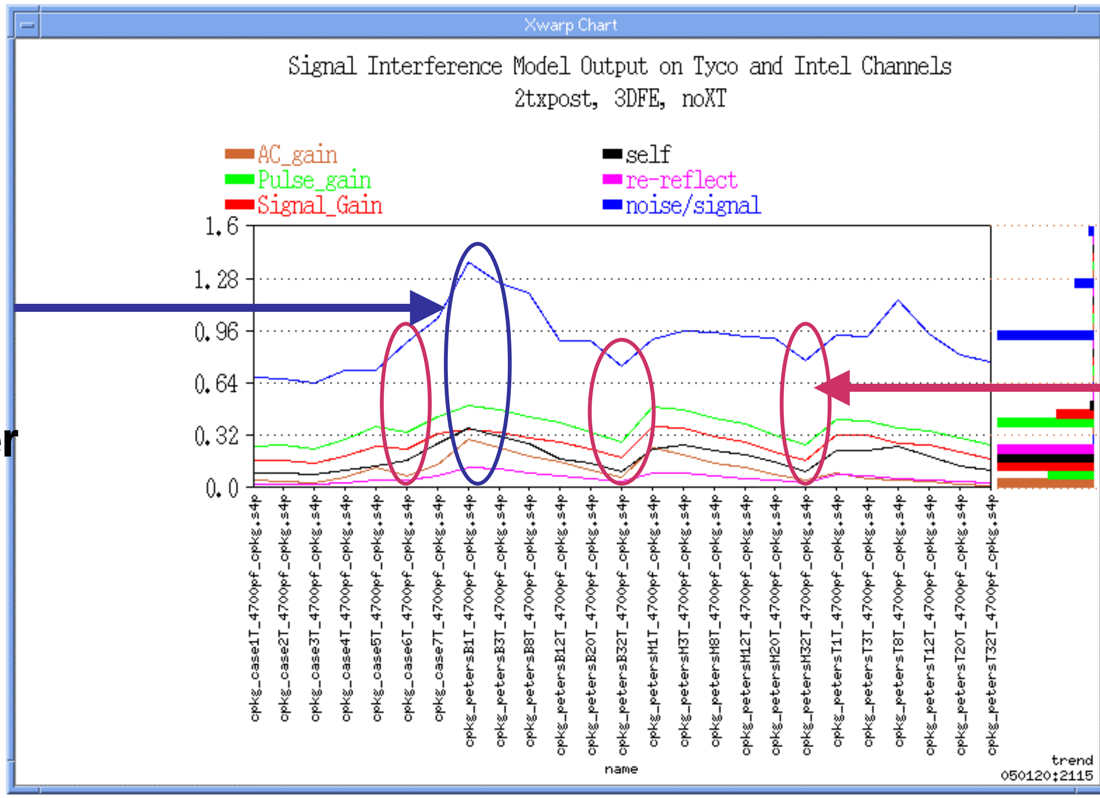
1. NRZ signaling at 10.3125Gbps
2. Cascaded S-parameters of Intel package->Tyco / Intel Channels->coupling capacitor-> Intel package
3. 2 TX post cursor taps, 2 RX DFE taps
4. Tj= 0.4UI
5. No Crosstalk (function detailed in moore_01_0704 needs to be coded in)

SigInt Simulation Results

various gains, self and re-reflected interference, and N / S ratio vs. Tyco and Intel channels (noXT)

Each channel sim took ~30 seconds

Short channels show noise more clearly and pulse amplitude is better



Long channels show low noise due to attenuation and amplitude is diminished

Self interference (pulse response tail ripples) is consistently larger than re-reflect

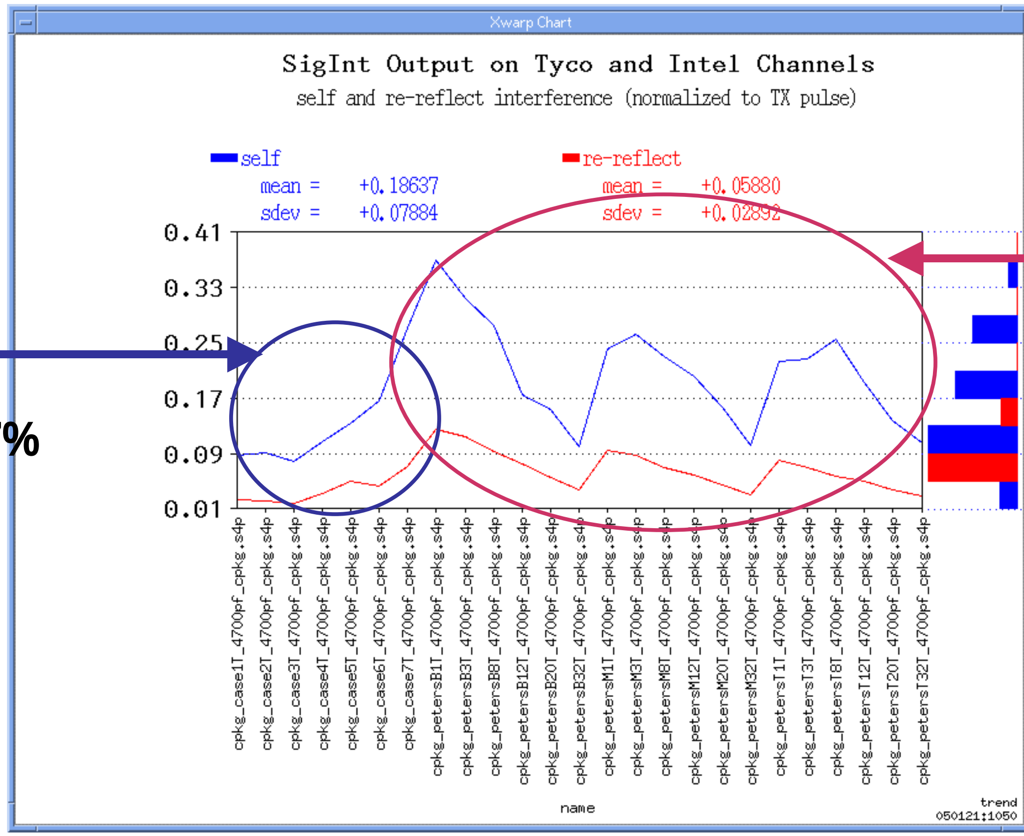
SigInt Simulation Results continued...

self and re-reflected interference vs. Tyco and Intel channels (noXT)

Tyco channels

Self= 8%-27%

ReReflect= 2%-7%



Intel channels

Self= 10%-37%

ReReflect= 3%-13%

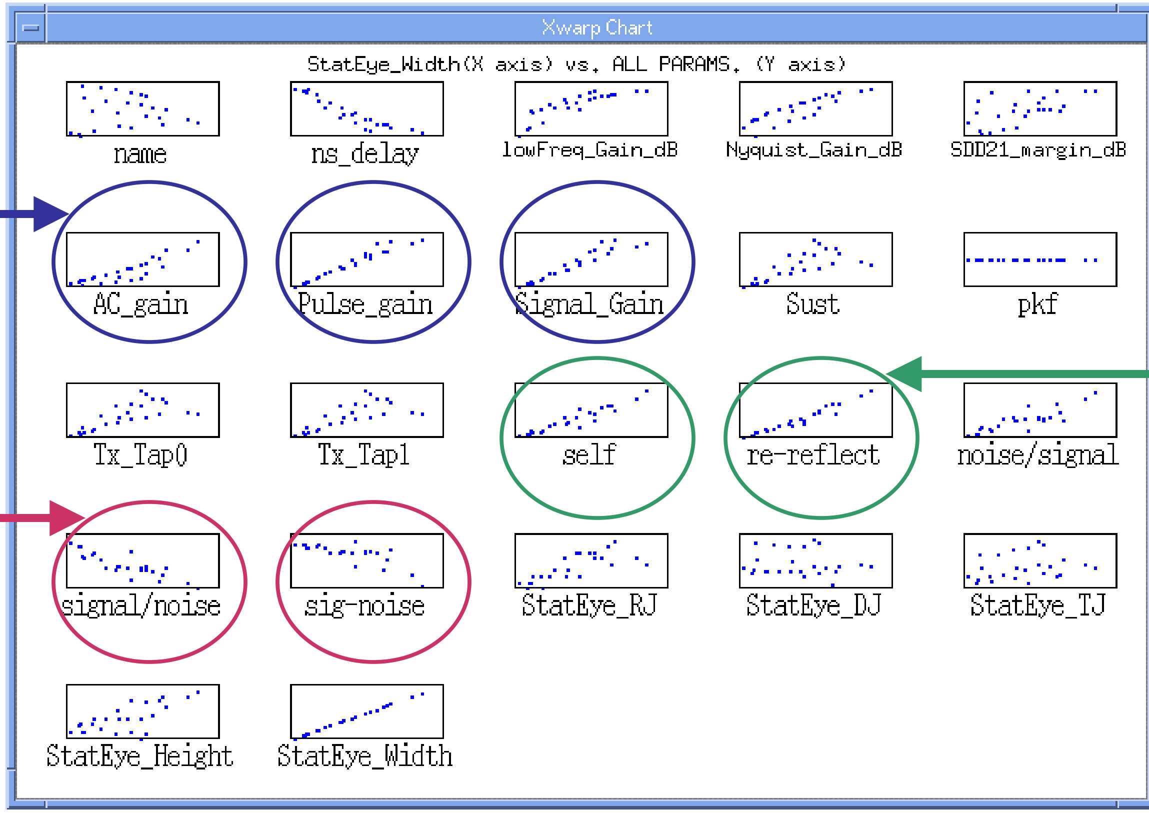
ReReflect appears to be inverted since it diminishes with stub length

Self appears to correlate with low loss channels with pulse response ripples

Correlation Between StatEye and SigInt

StatEye Width against all StatEye and SigInt parameters

gains correlate as expected



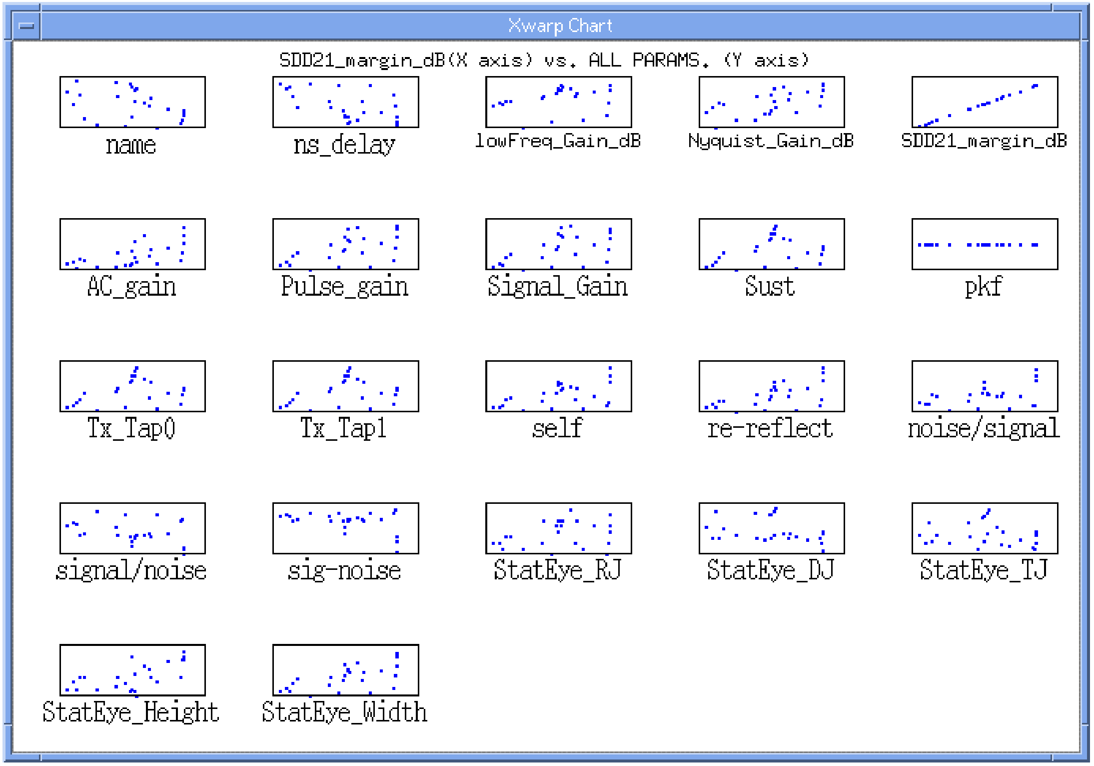
noise characteristics are inverted

ratios are inverted

self and re-reflect interference correlate with StatEyeWidth, but are inverted

Correlation continued...

SDD21_margin (dB delta from Goergen SDD21 limit at Nyquist) against all StatEye and SigInt parameters

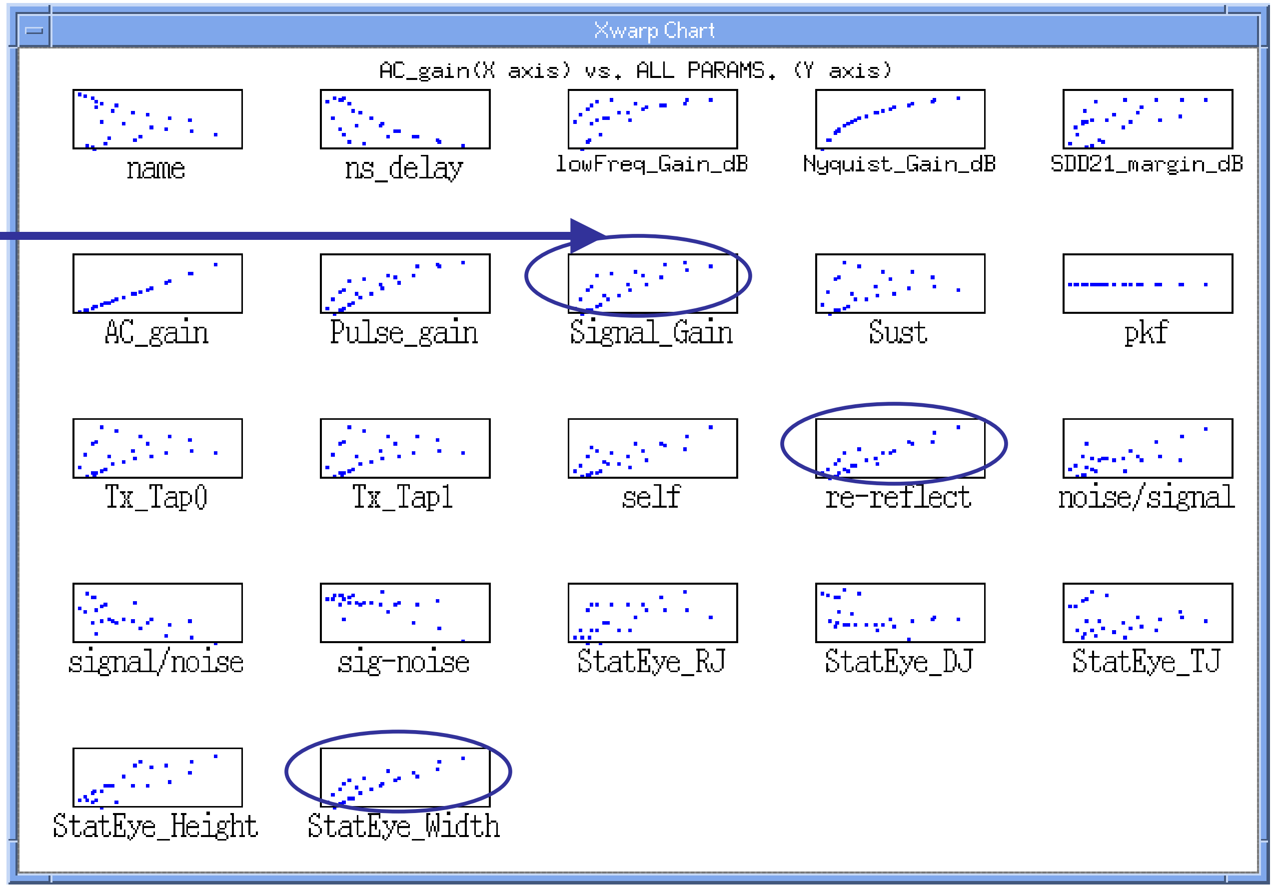


No reasonable correlation to any parameter shows SDD21 is not able to completely account for all necessary 10G channel characteristics

Correlation continued...

AC_gain (at Nyquist of SDD21) against StatEye and SigInt parameters

binomial distribution groups channels with long stubs

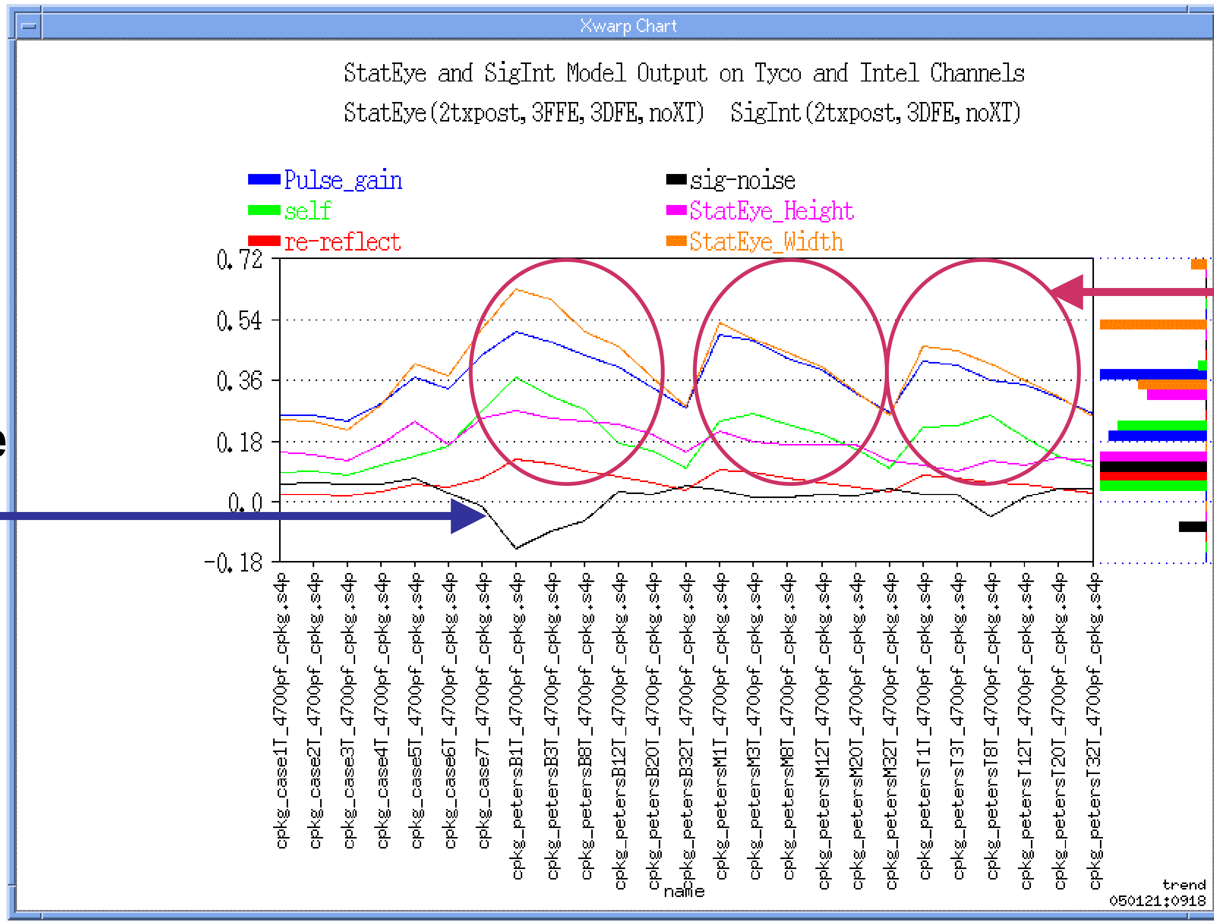


Intel channels with 175mil backplane stubs stand out with either tool

Compared Results

StatEye and SigInt results vs. Tyco and Intel channels

both signal loss and reflection are shown here



sig-noise tracks StatEye except on very low loss (short, clean) channels

self and re-reflect are inverted compared to StatEye Width and Height

Conclusions

- Without crosstalk, all channels with packages and capacitor had at least 9% open StatEyes at a BER of 1E-15 with 2 TX post cursor taps, 4FFE taps, and 4DFE taps.
- With 4 aggressor crosstalk, EyeHeight is reduced 1%-6%.
- StatEye's EyeHeight and EyeWidth are able to discern between reflective stub effects and absorptive loss effects.
- SigInt's "Self" finds channels with bad pulse response ripples.
- SigInt's "AC_gain" separated Intel channels with longest stubs.
- SigInt's "ReReflect" doesn't find stubs, and may need coding changes.
- The StatEye EyeHeight and EyeWidth correlate positively with the SigInt's noise metrics... need to look through SigInt code.

Supporting Slides

StatEye user_parameter.m file used for simulations

```
%{{{{
%************************ THIS IS A GUI GENERATED FILE **********************
% This file of user parameters has been generated by the StatEye GUI.        %
% If you wish to enter parameters manually and not through the GUI then copy  %
% user_parameters_manual_template.m to user_parameters.m, and then run main.m %
%                                                                             %
param.bps = 1.03125e+010;
param.txdj = 0.15;
param.txrj = 0.15;
param.BER = 1e-015;
param.PNA = 'off';
param.DCD = 0.05;
param.tx_ideal = 'on';
param.tx_gaussian = 'off';
param.tx_spectrum = 'off';
param.PAM = 2;
```

Supporting Slides

user_parameter.m continued...

```
param.datacoding = 1;
param.txpre = [];
param.txpost = [-0.1 -0.1];
param.vstart = [-0.4 -0.3 -0.2];
param.vend = [0 0 0];
param.vstep = [0.1 0.05 0.025];
param.txdeemphasis = [];
param.txFilter = 'off';
param.txFilter_restore = 'off';
param.txFilter_type = 'singlepole';
param.txFilterParam = [1 1];
param.txBesselFilterOrder = 4;
param.txcpad = 0;
param.txtermination = 50;
```

Supporting Slides

user_parameter.m continued...

```
param.tosa = 'off';

param.nonlinear_tosa_data = [''];

param.tosa_Sparams_type = 'p2';

param.p2_tosa_data = [''];

param.p4_s2p_tosa_data = [''];

param.p4_s4p_tosa_data = [''];

param.channel = 'p4_s4p';

param.p4_s2p_channel_data = ['../Models/Channels/p4_s2params/thru_TXP_040309/'];

param.p4_s4p_channel_data = ['../../../../IEEE802/802.3ap/channels/bills_channels/cascaded/cpkg_petersB1T_4700pf_cpkg.s4p'];

param.abcd_channel_data = ['../Models/Channels/abcd/thru_2p_TXP_040309.abcd'];

param.afr_data = ['../Models/Channels/afr/Cambridge_rel_1_0/CamMMF1p0f34o23f.txt'];

param.mpf_data = ['../Models/Channels/mpf/Cambridge_rel_1_0/CamMMF1p0f01o17i.txt'];

param.rosa = 'off';

param.nonlinear_rosa_data = [''];

param.rosa_Sparams_type = 'p2';
```

Supporting Slides

user_parameter.m continued...

```
param.p2_rosa_data = [''];

param.p4_s2p_rosa_data = [''];

param.p4_s4p_rosa_data = [''];

param.xtlk_summation = 'linear';

param.noAggressors = 4;

param.stype_aggressor = 'p4_s4p';

param.p4_s2p_aggressor_data = [];

param.p4_s4p_aggressor_data = ['../../../../IEEE802/802.3ap/channels/bills_channels/cascaded/cpkg_petersB1F1_4700pf_cpkg.s4p';
'../../../../IEEE802/802.3ap/channels/bills_channels/cascaded/cpkg_petersB1N1_4700pf_cpkg.s4p';
'../../../../IEEE802/802.3ap/channels/bills_channels/cascaded/cpkg_petersB1F2_4700pf_cpkg.s4p';
'../../../../IEEE802/802.3ap/channels/bills_channels/cascaded/cpkg_petersB1N2_4700pf_cpkg.s4p'];

param.rxcpad = 0;

param.rxtermination = 50;

param.rxFilter = 'off';

param.rxFilter_type = 'singlepole';

param.rxFilterParam = [1 1];

param.rxBesselFilterOrder = 4;

param.ffe_tap = 4;
```

Supporting Slides

user_parameter.m continued...

```
param.dfe_tap = 4;
param.CDRType = 'SinglePLL';
param.eyeenlargement = 'off';
param.plot_txFilter = 'off';
param.plot_PAD = 'off';
param.plot_ini_s_params = 'off';
param.plot_abcd_data = 'off';
param.plot_trans_func = 'off';
param.plot_result_dif_s_params = 'off';
param.plot_interpolated_data = 'off';
param.plot_mpf = 'off';
param.plot_afr = 'off';
param.plot_tosa = 'off';
param.plot_rosa = 'off';
param.plot_rxFilter = 'off';
```

Supporting Slides

user_parameter.m continued...

```
param.plot_pulse_response = 'on';

param.plot_pna = 'off';

param.plot_stat_eye = 'on';

param.plot_stat_eye_type= 'BER';

param.plot_bathtab = 'off';

param.min_eye_contour = 1e-015;

param.resultsfile = ['../Results/example.mat'];

%                                                       %

%*************************** END OF GUI GENERATED FILE ************************

%                                                       %

%}}}}
```