The use of Optimization in Signal Integrity Performance Centric High Speed Digital Design Flow

Asian IBIS Summit, Shanghai China
November 4, 2009

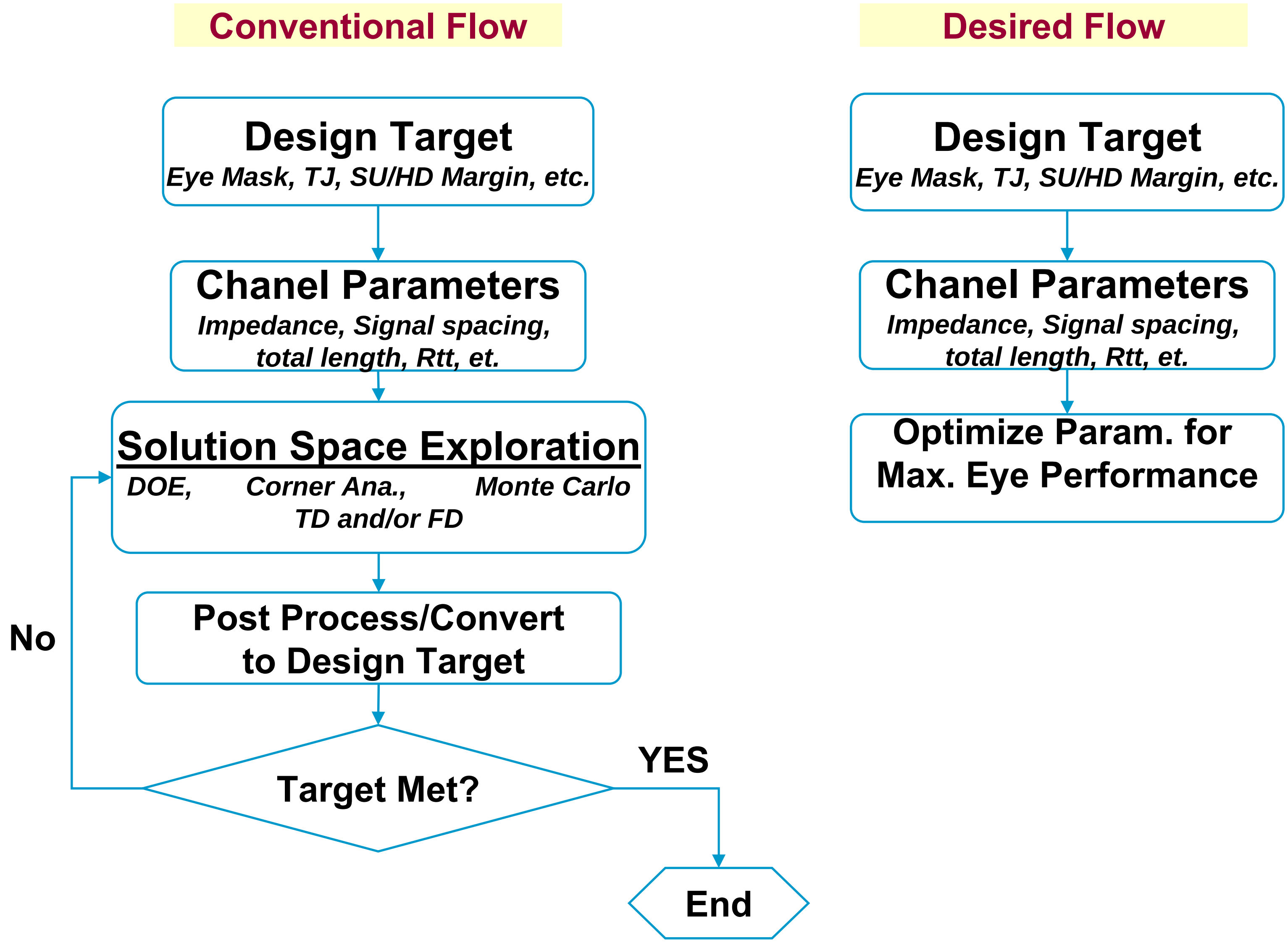
Brahim Bensalem, Intel Corporation
Lihua Wang, Agilent Technologies, EEsof Division
Sanjeev Gupta, Agilent Technologies, EEsof Division
Xuliang Yuan, Agilent Technologies, EEsof Division

Email contact: sanjeev_gupta@agilent.com

Agenda

- Channel Complexity and Gaps in Current Design Flow
- Advantage of End to End Eye Centric Design Flow
- Why an Eye Diagram ? And Eye Diagram Measurements
- Optimization of DDR2 Channel.
- Introduction to Batch Mode Simulation
- DDR Compliance

Gap in Channel Design Flow

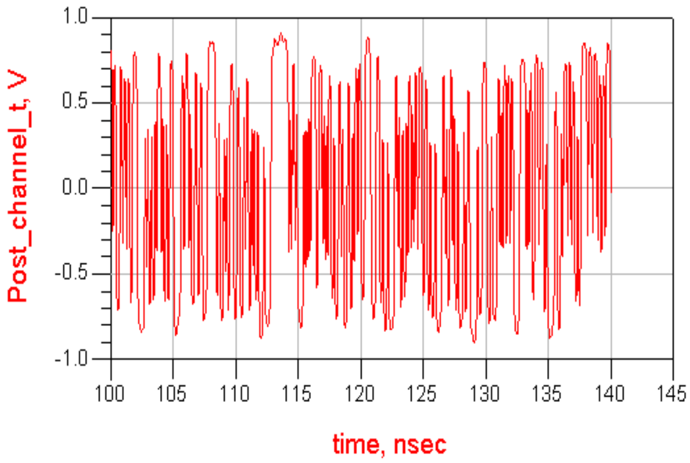


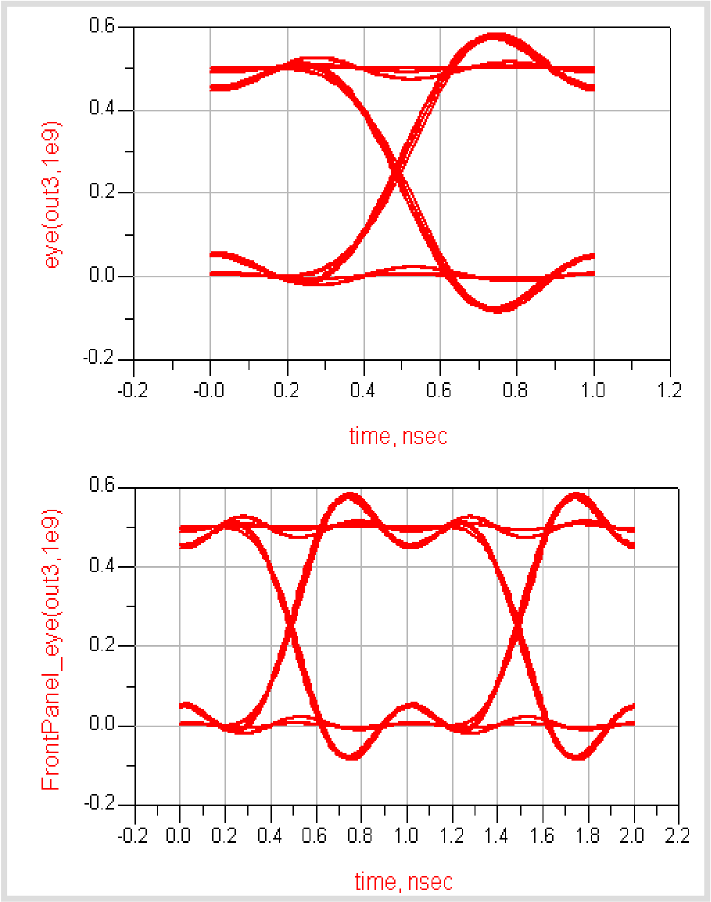
Advantage of Eye Centric Design Flow

- Substantial gain in Channel Design Time (More than 3 weeks in our DDR2 case)
- Design is more robust since channel is optimized toward end to end channel performance
- Reduces risk of marginal design or opportunity loss due to over design

Different Eye Diagrams Functions

Why Eye Diagram is Important?





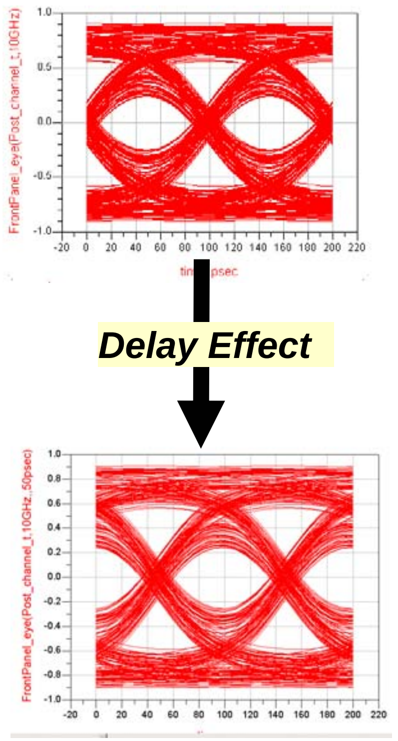
intel

Agilent Technologies

Characterizing an Eye Diagram

Most commonly used eye diagram measurements

- Eye level 1 & level 0
- Eye rise/ fall time
- Eye opening
- Eye width
- Eye height
- Eye amplitude
- Peak to peak & RMS jitter

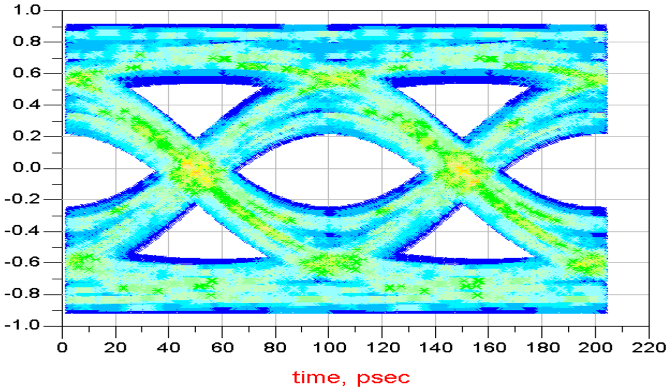


Most of the measurements are statistical in nature

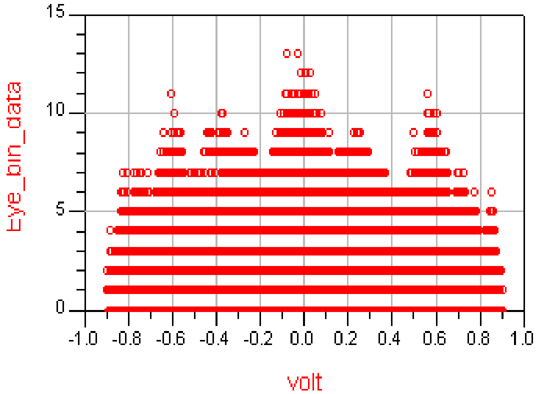
Slice and Dice

Eye Binning

Eye Diagram

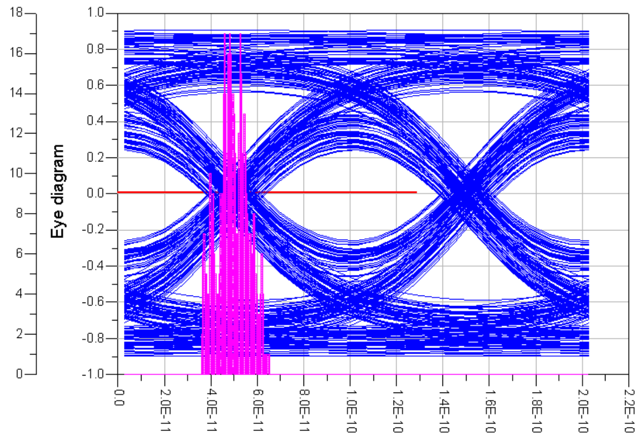


Binning Data

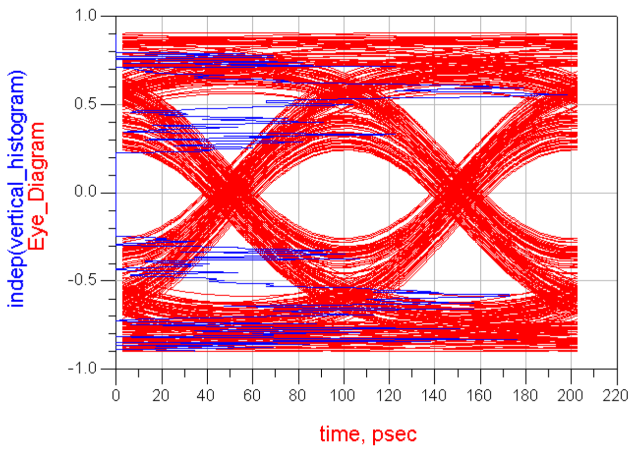


Histogram Plots

Histogram can be created for any portion of eye diagram



Histogram across timing axis provide peak to peak jitter

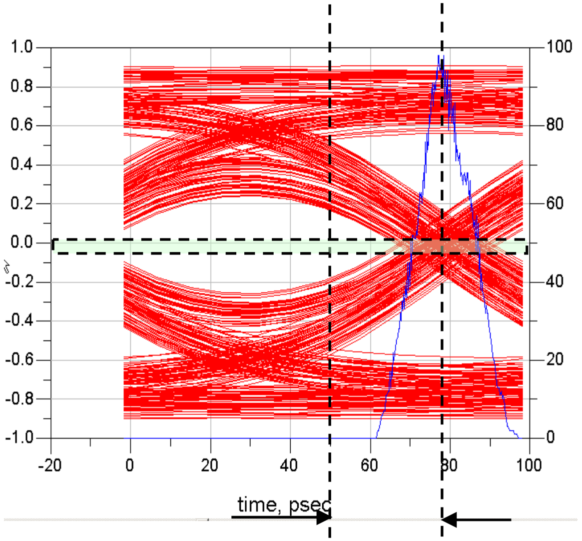


Histogram across amplitude axis provide distribution around level one and zero

Eye Delay

Why delay calculation is required

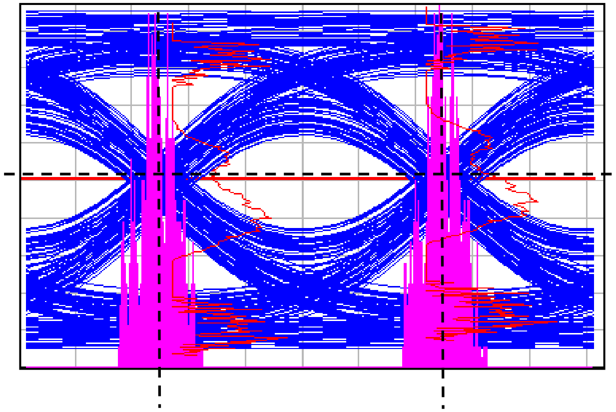
- Delay calculation is required for automated eye parameter measurements
- Binning the eye diagram makes this calculation easy



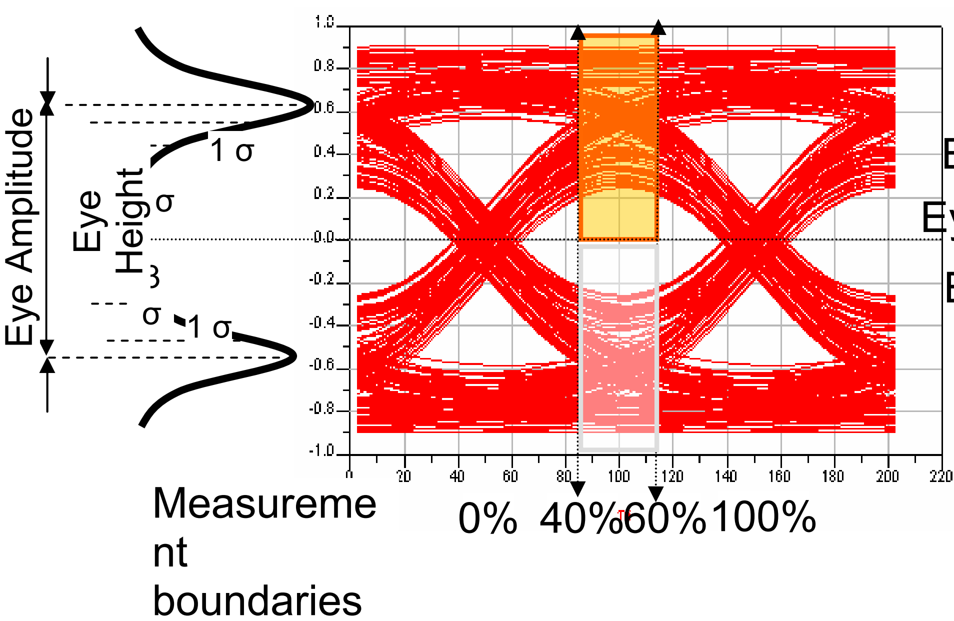
Automated Eye Crossing Detection

- Mean value of horizontal histogram provide crossing time value

- Mean value of amplitude histogram provides crossing amplitude value



Measurements of Eye Level One/Zero



Eye Amplitude = Level One – Level Zero

Eye Height = (Eye level one- 3σ)- (Eye level zero+3σ

$$\text{Eye S/N} = \frac{(\text{Eye level one} - \text{Eye level zero})}{1\sigma_{\text{level one}} + 1\sigma_{\text{level zero}}}$$

intel

Agilent Technologies

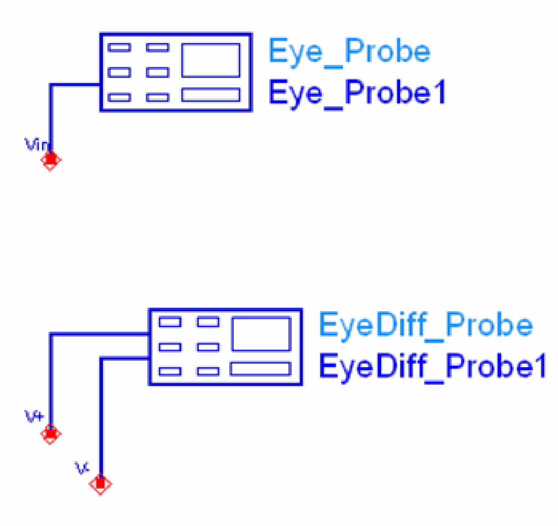
Eye Measurements

```
MeasEqn
GetEye
sc_single_eye=eye(Diff0,Rate)
sc_get_delay=FrontPanel_eye_delay(Diff0,Rate,"NRZ")
sc_get_eye=eye_binning(FrontPanel_eye(Diff0,Rate,1,sc_get_delay),451,321)

MeasEqn
Levels
sc_get_max_voltage=max(Diff0)
sc_get_min_voltage=min(Diff0)
sc_get_avg_voltage=(sc_get_max_voltage-sc_get_min_voltage)
sc_level_40=0.2*sc_get_avg_voltage
sc_level_60=0.8*sc_get_avg_voltage

MeasEqn
Waveform
sc_amplitude_histogram_data=FrontPanel_eye_amplitude_histogram(sc_get_eye)
sc_waveform_topbase=FrontPanel_wave_topbase(sc_amplitude_histogram_data,"NRZ")
sc_waveform_top=sc_waveform_topbase[0]
sc_waveform_base=sc_waveform_topbase[4]
sc_eye_crossing=FrontPanel_eye_crossings(sc_get_eye,sc_waveform_topbase[7],sc_waveform_topbase[2],"NRZ")
sc_eye_top_base=FrontPanel_eye_topbase(sc_get_eye,sc_eye_crossing,40,60,"NRZ")

MeasEqn
Eye_Measurements
sc_frontpanel_eye=eye_density(FrontPanel_eye(Diff0,Rate,1,sc_get_delay),451,321)
sc_eye_Amplitude=sc_eye_Level_One-sc_eye_Level_Zero
sc_eye_Height=sc_eye_top_base[9]-sc_eye_top_base[8]
sc_eye_Width=FrontPanel_eye_width(sc_get_eye,sc_eye_crossing,Rate)
```

Simplified Eye Measurements

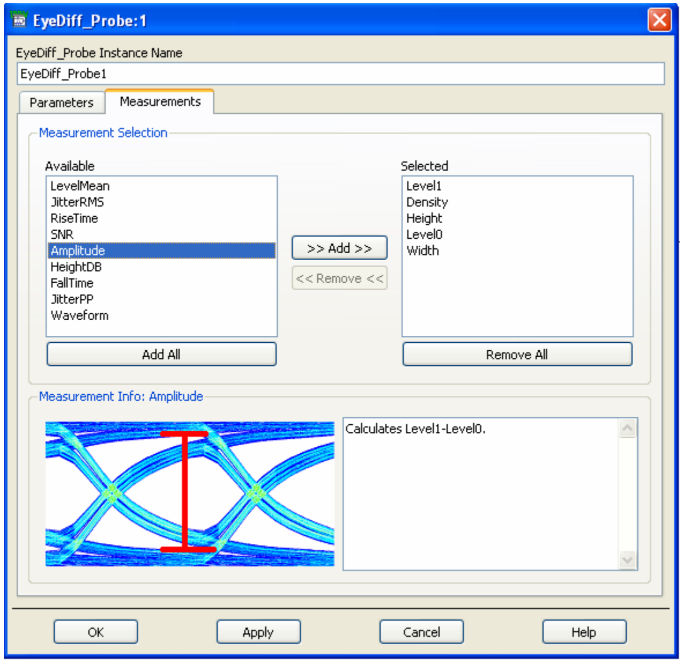
Fast and Easy

Concept of Eye Probe

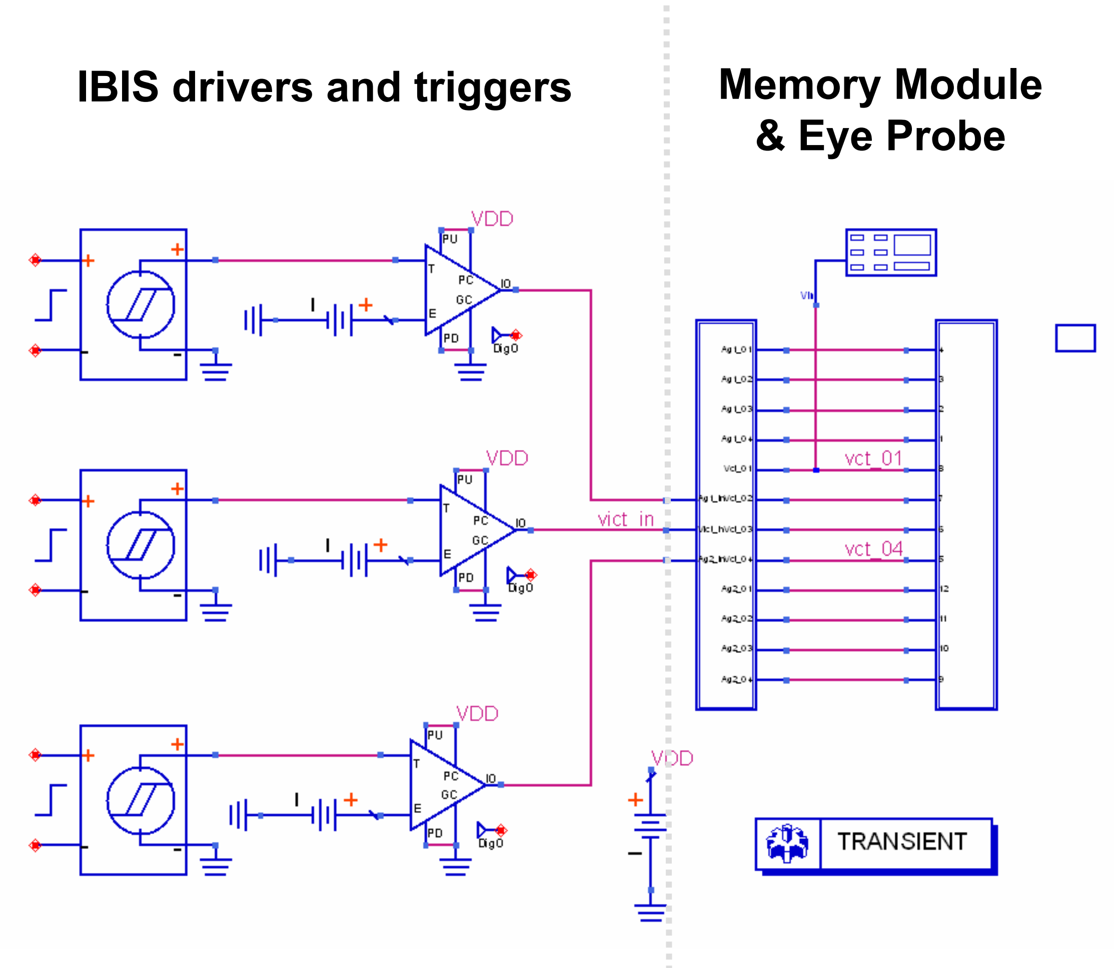


Any number of Eye probes could be used in a design

Measurements which needs to be performed

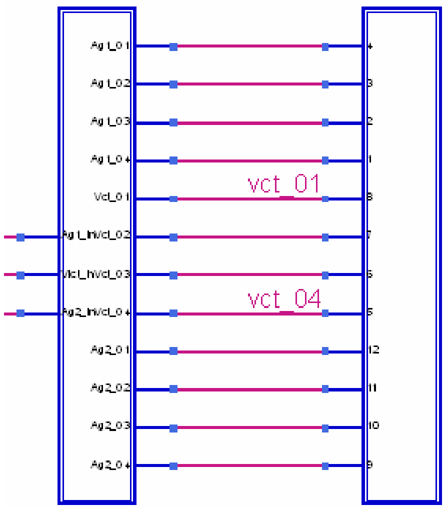


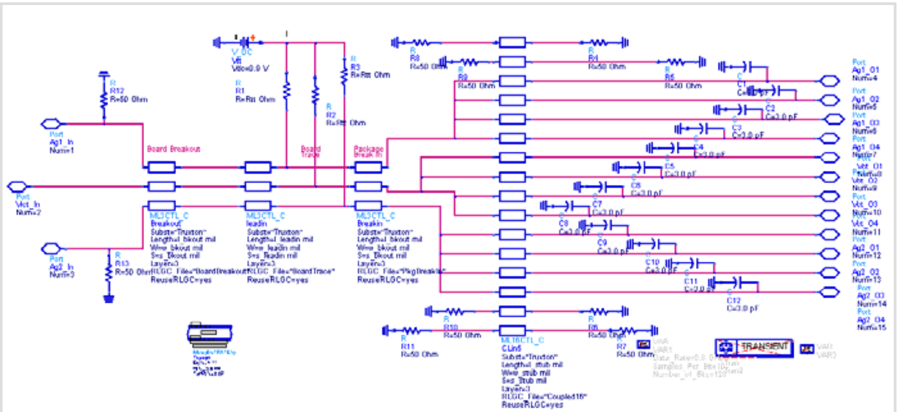
DDR2 Simulation

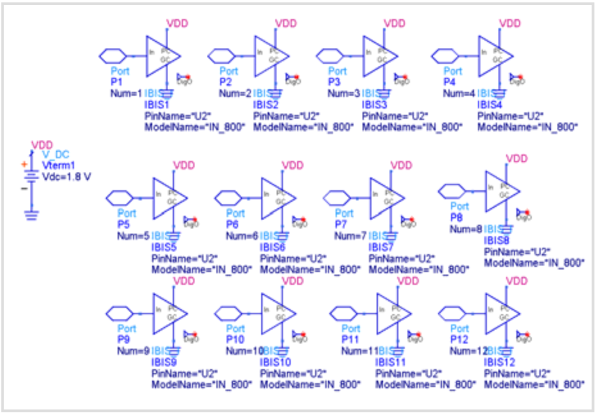


Memory Bus Simulation Details

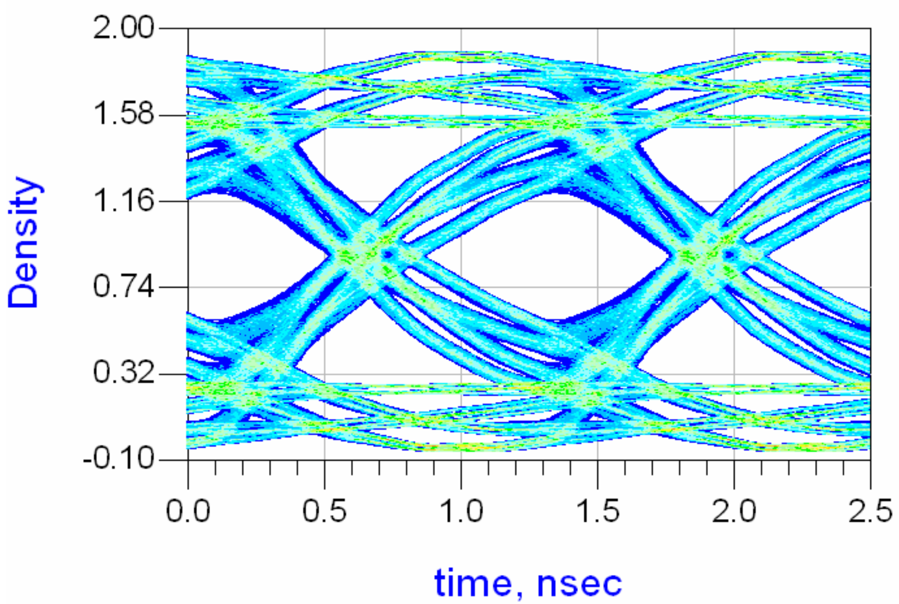
- X8 Un-buffered Memory Down Channel
- 8 SDRAM Devices per signal
- Signal Group : CMD/ADD







Memory Performance at DRAM



permute(Height)
0.472

permute(Width)
9.050E-10

How to improve channel performance?

- Introduction to Optimization
- Time domain Optimization and why it is difficult
- Optimization of DDR2 Channel

Introduction to Optimization

Modify your Designs Automatically to Achieve Required Performance

Why Optimization?

- Parameter sweep often doesn't lead to an optimized designs
- Parameter sweeps requires usually large number of simulation when number of variables are large

The use of Optimizers in a design process

- Automatically change design parameter to meet design goals
- Categorized by their error function formulation
- Coarse design stages: Random optimizer, Random Minimax optimizer and Simulated Annealing optimizer
- Fine design stages: Gradient optimizer, Gradient Minimax optimizer, Quasi-Newton optimizer and Minimax optimizer

Issues with Time Domain Optimization

– Time domain optimization goals are often difficult to define

– Changes in any reactive component during optimization will effect channel delay and optimization goals may no longer be applicable

DDR2 Channel Design

X8 Un-buffered Memory Down Channel

8 SDRAM Devices

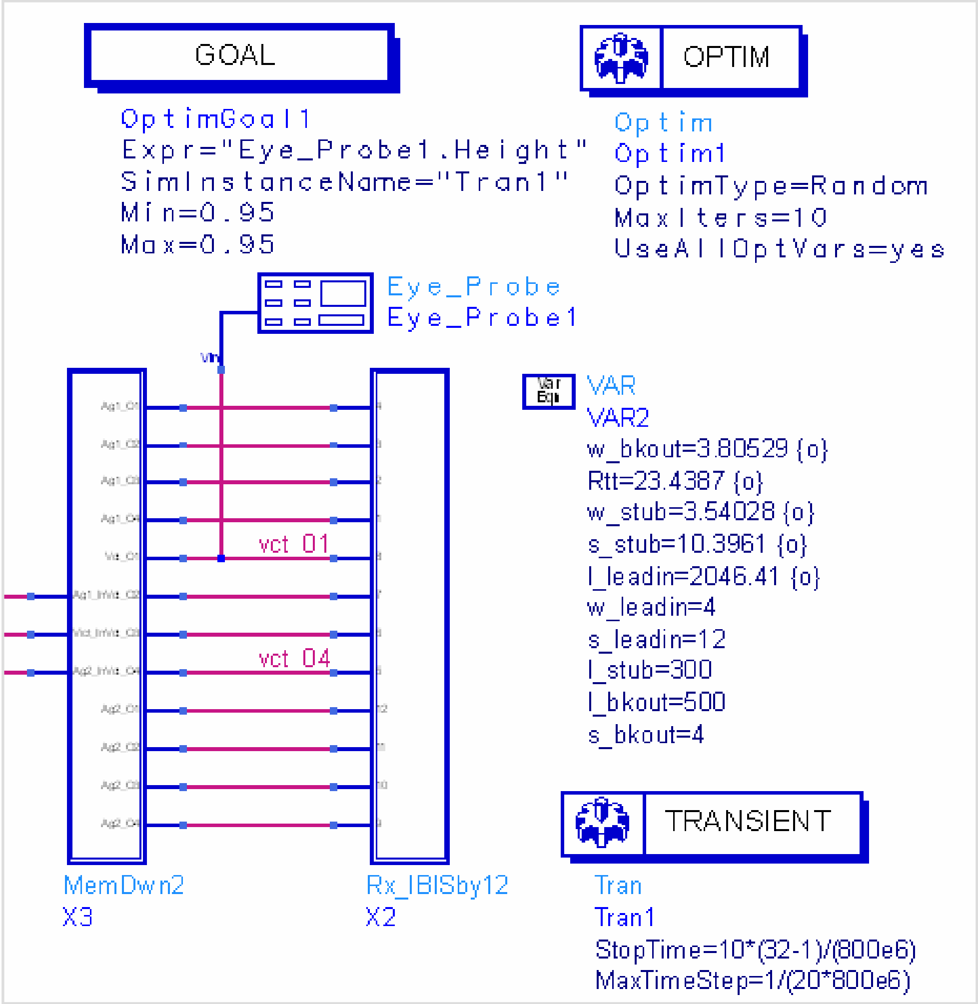
Signal Group to Optimize: CMD/ADD

Design Parameters:

Leadin	Escape	W leadin	S Breakout	Trace Spacing	Rtt	L Brkout
2-4 in	0.3-0.8 in	3.5-5.5 mils	3-5 mils	8-15 mils	20-100 Ω	0.3-0.8in

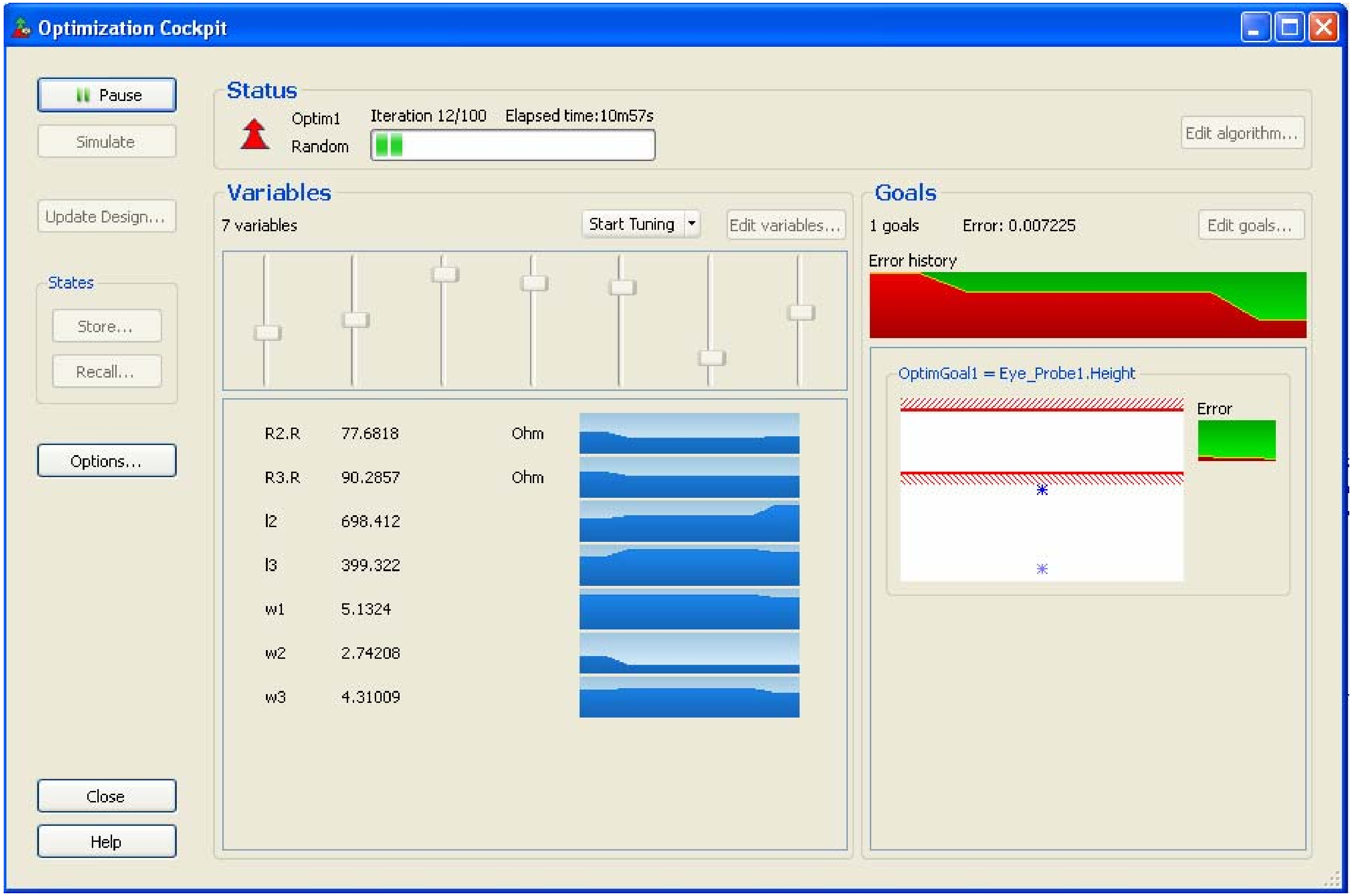
Eye Diagram Optimization

- Which all eye measurements can be optimized
- How many nodes can be optimized simultaneously
- Can optimization be performed easily



Optimization Cockpit

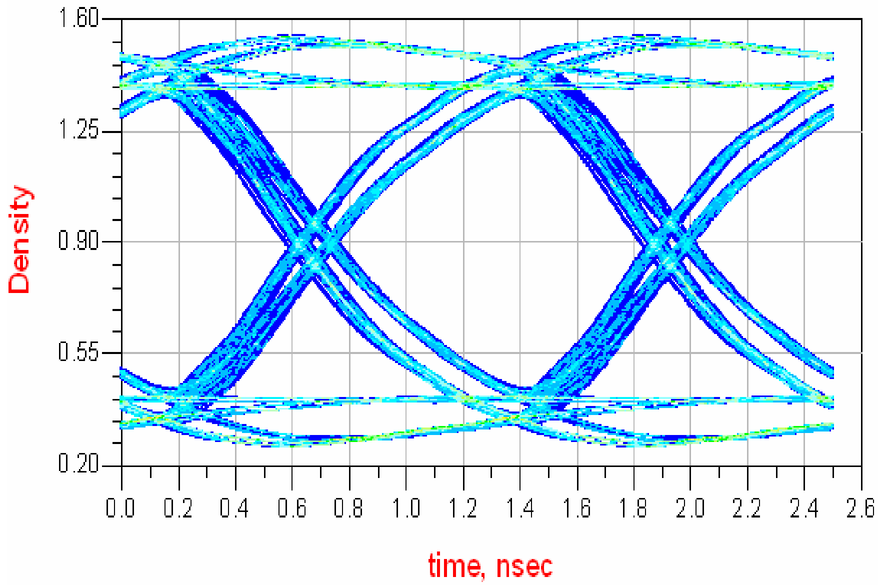
What are the benefits of such a cockpit?



Agilent Technologies

Optimized Eye Diagram Performance

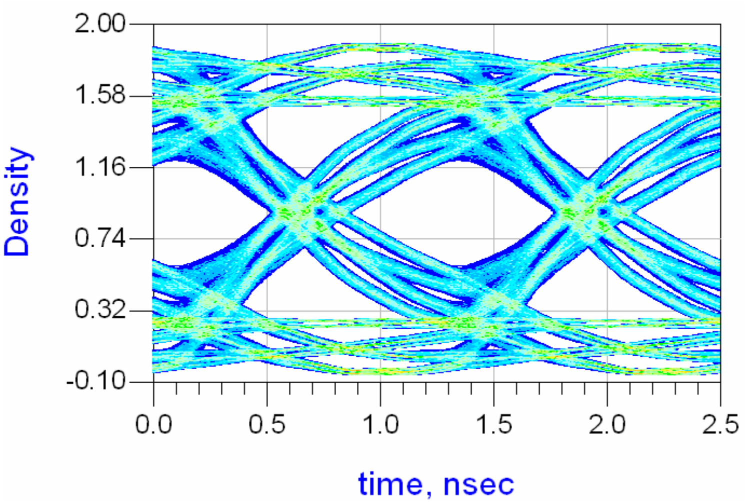
Eye Diagram after Channel Optimization



Optimizer Type	**: Random**
Number of iteration	**: 20**
Optimization time	**: 25 minutes**

Performance Comparison

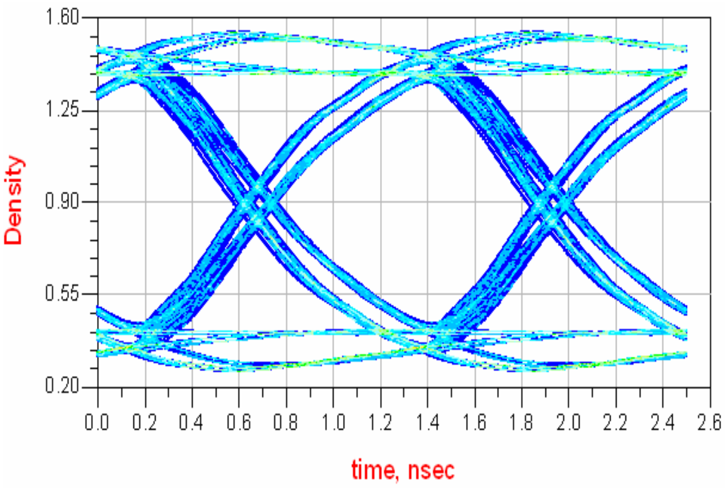
Before



permute(Height)
0.472

permute(Width)
9.050E-10

After

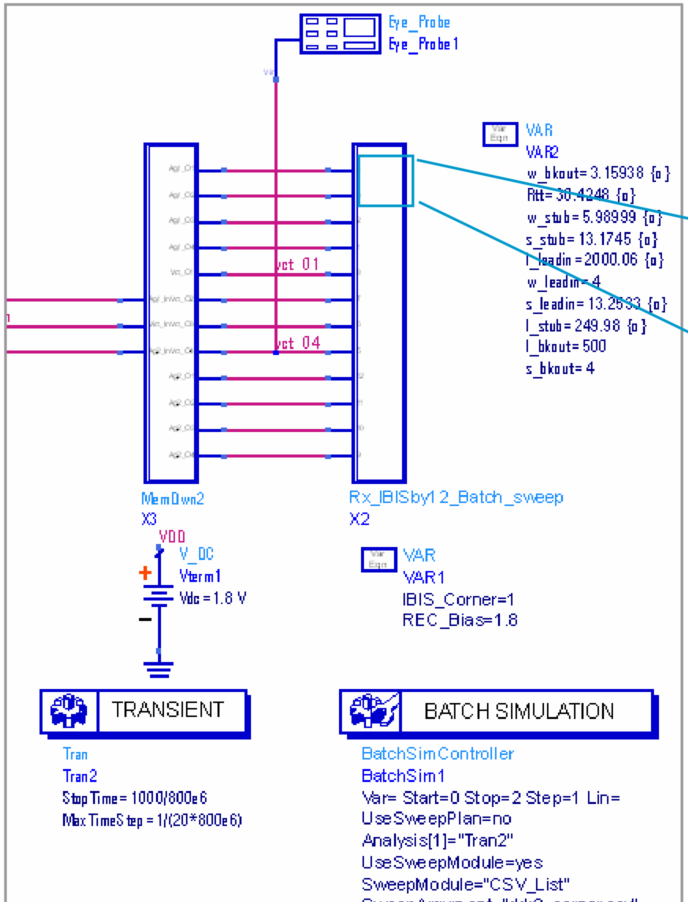


permute(Height)
0.700

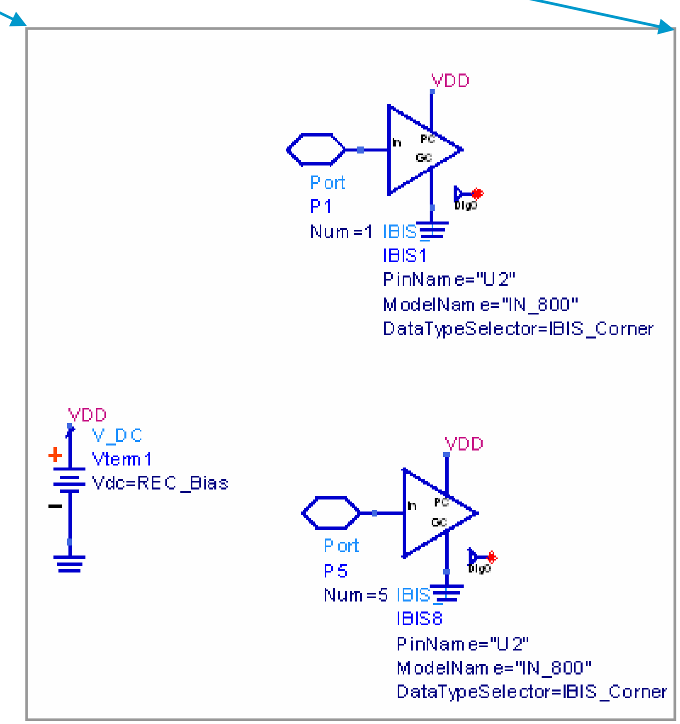
permute(Width)
1.078E-9

Sweeping Corner Case

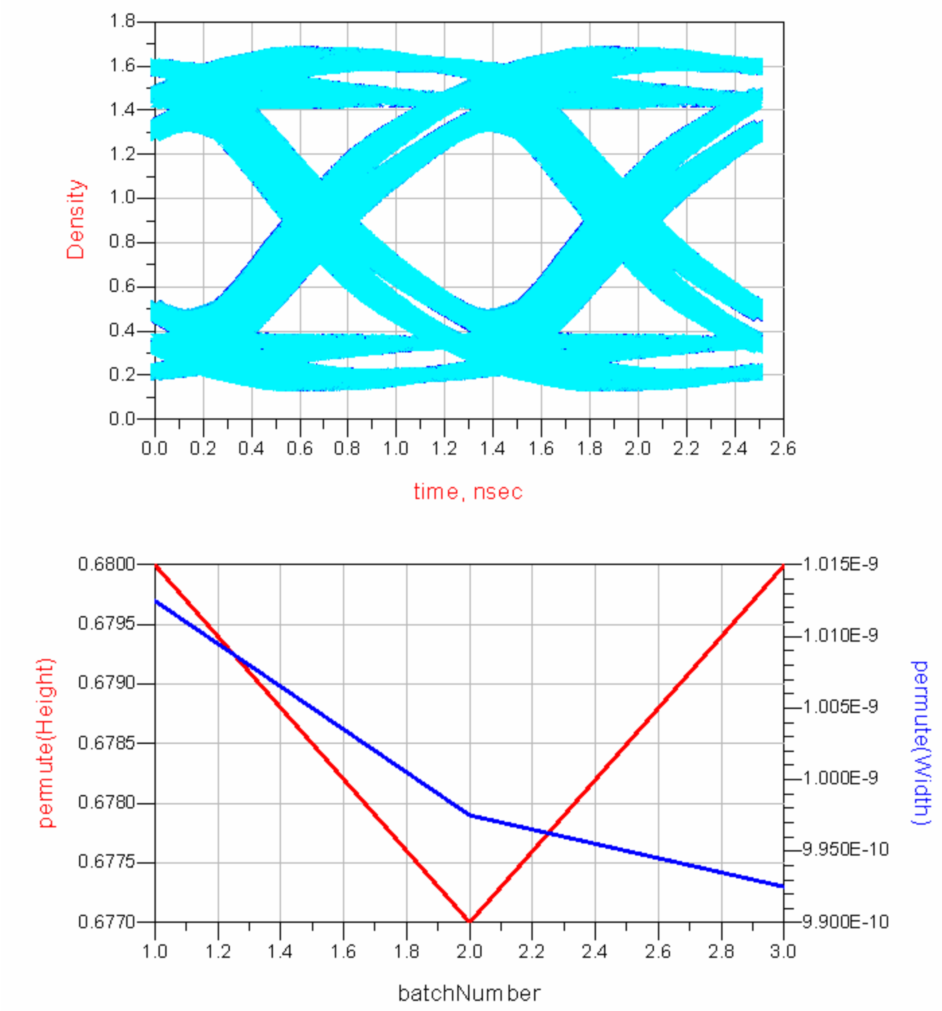
Batch simulation is used to sweep corner cases



	A	B
1	IBIS_Corner	REC_Bias
2	1	1.8
3	2	1.7
4	3	1.9



Simulation Results



Here only Rx corner cases was swept.

Batch mode allows simulation of any combination of Tx/Rx corner cases and IBIS models

intel

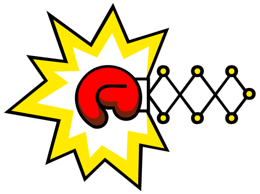
Agilent Technologies

Time Domain Optimization Discussed

- Works well even if the flight time delay is changed due to change in the reactive element value.
- Automatically calculates delay required for eye positioning
- Automatically detects eye crossing point and 40-60% region
- Optimize eye diagram performance
- Corner case simulation performance was determined

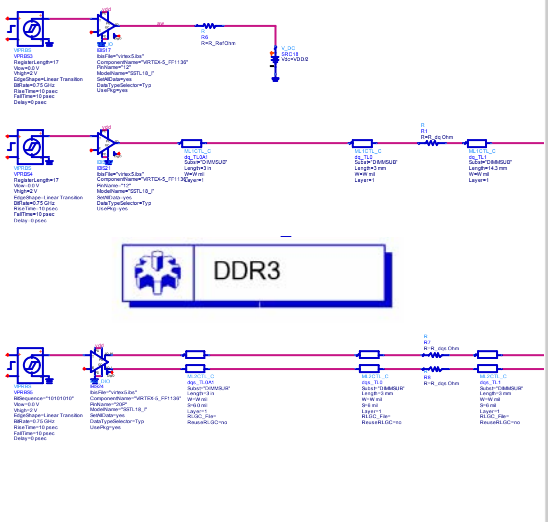
Any eye diagram parameter such as eye opening factor, eye height, peak to peak jitter, rise time … can be used as an optimization goal.

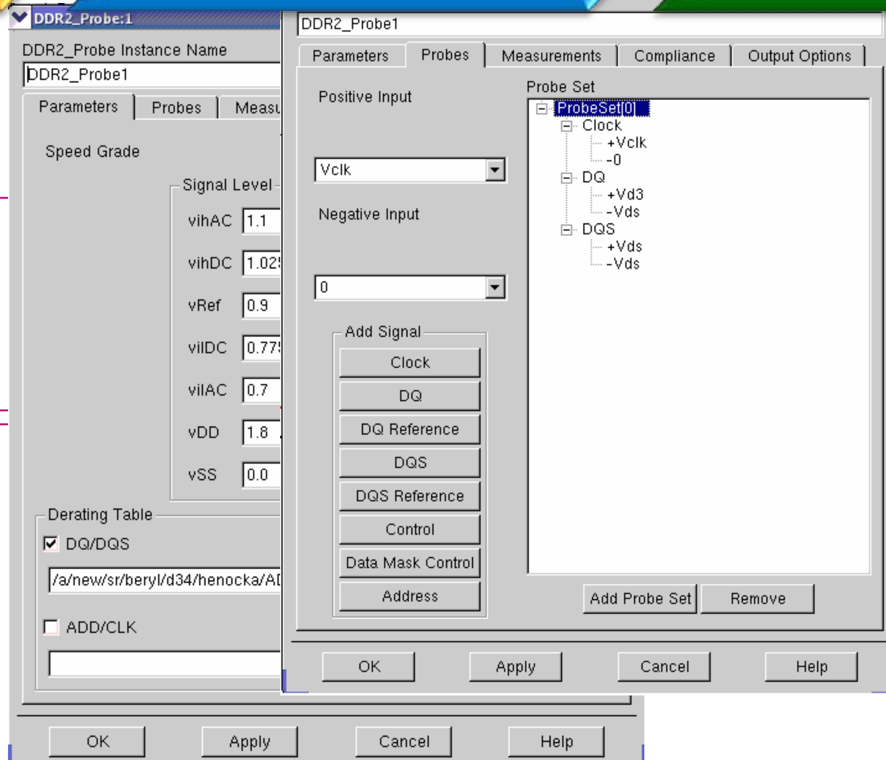
Will make your design work without running 1000's of parameter sweep



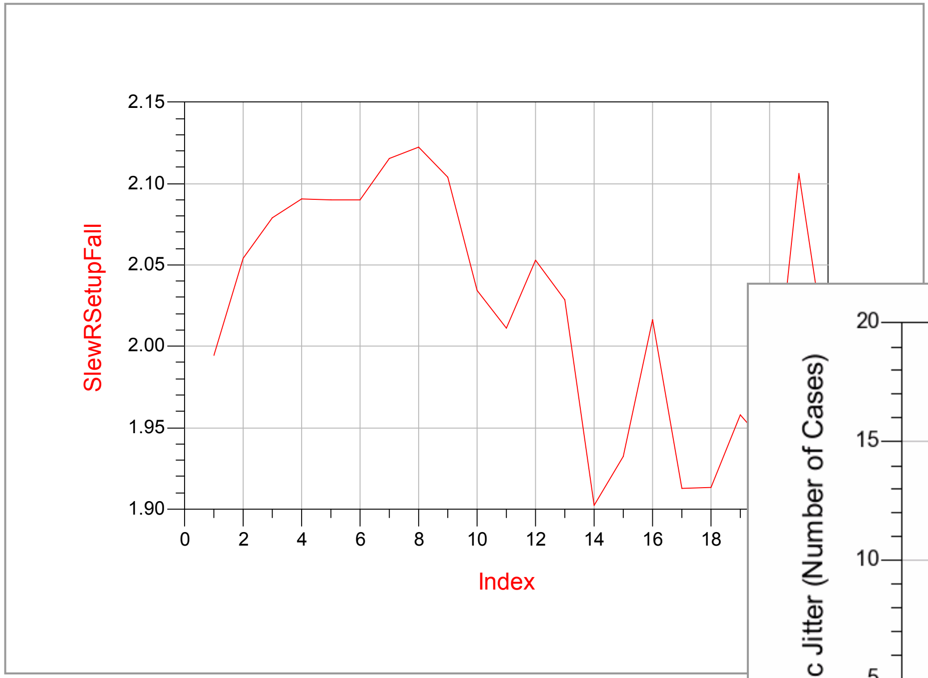
DDR Mesurements

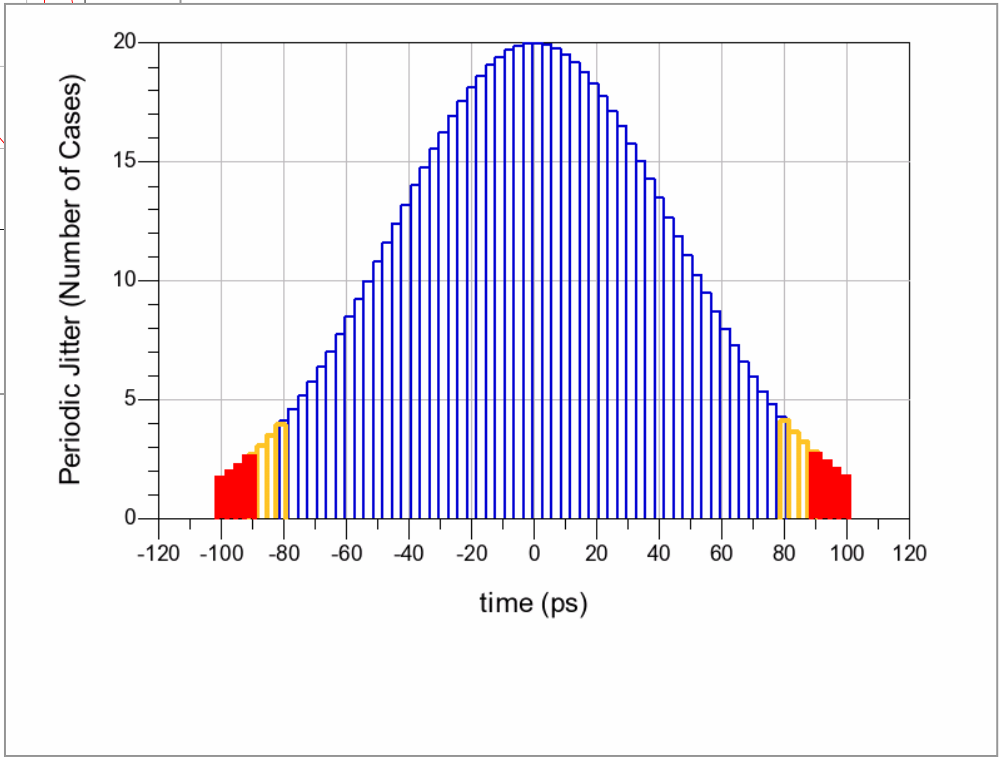
Memory Compliance Toolkits





Memory Compliance Toolkits-Features





Conclusion

- Time Domain optimization of eye diagram provides a powerful methodology to improve high speed memory design and to extract even fraction of the psec of timing margin buried in interconnects
- Substantial reduction in time needed to design and optimize of memory platform design is made (from weeks to hours)
- Guaranteed maximal channel robustness
- Minimize Over/Under design risks

Opportunities for future development

Need to demonstrate IBIS model optimization such as driver strength, ODT etc.