EyeKnowHow
Signal Integrity Consulting

Signal Integrity optimization of a complex (memory) bus system

Agilent ADS User Group Meeting
15th May 2009

Agenda

1) Introduction: What is the problem?

2) Channel Characterization: An Eye is Born

3) Channel Optimization: Conventional Method

4) Optimize pulse area

5) Optimize pulse shape

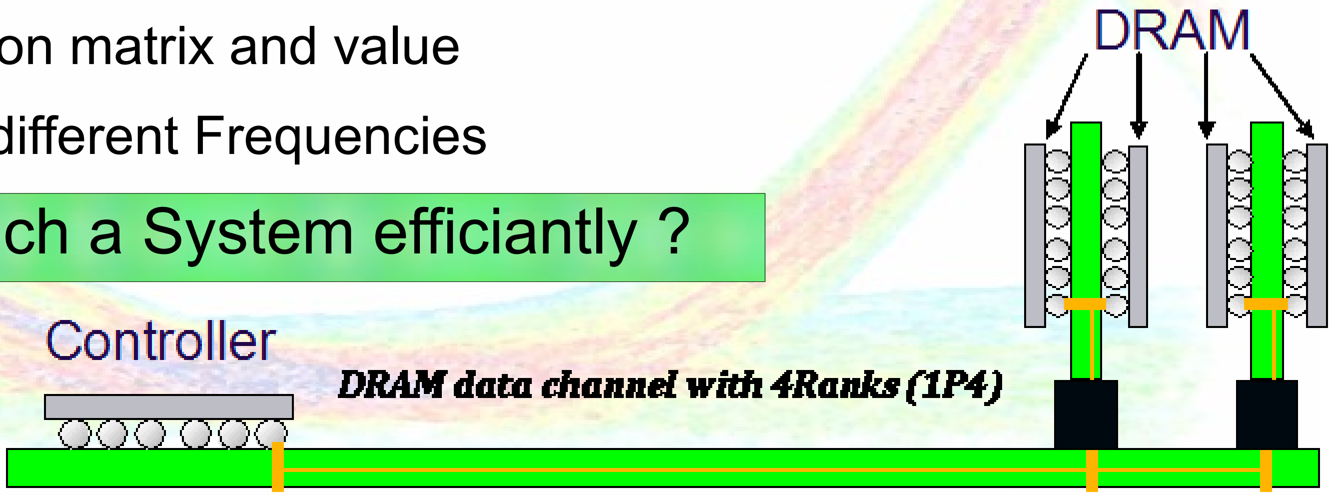
6) Outlook

7) Summary

1) Introduction
What is the Problem ?

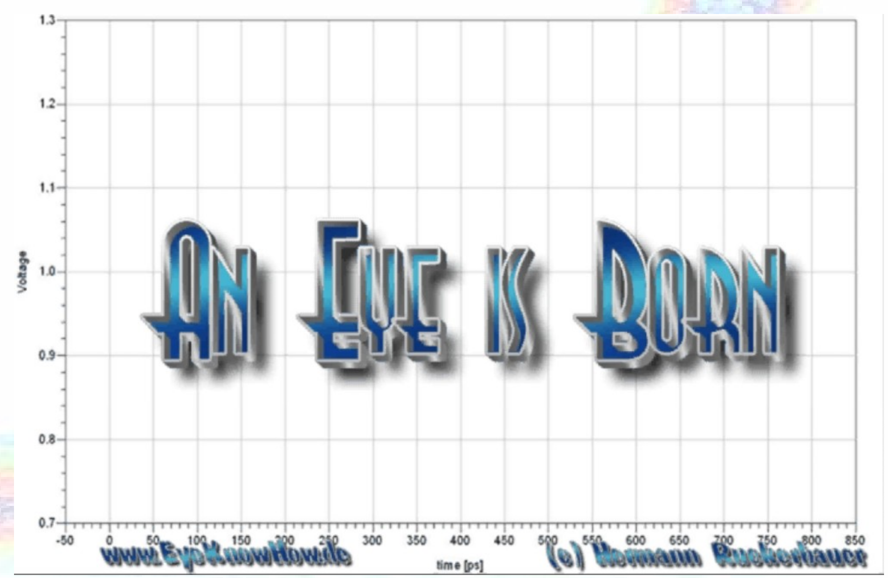
- Target: Optimization of a Signaling Channel
- Example: Multi Drop Memory Channel
- Problem: A huge matrix of variables might be varied:
  - Driver strength and linearity
  - Driver / Receiver Package parameter
    - Package Technology, Routing length/width/layer
  - Motherboard and DIMM routing
    - Topology (Flyby vs. T), Routing length/width/layer
  - Receiver Termination matrix and value
  - Write vs. Read @ different Frequencies
- How to optimize such a System efficiantly ?

System Target:
25.6GB/s (200Mb/s) on 64 lanes
~ 20cm Signaling on FR4
2 Slots with 2 loads each (1P4P)



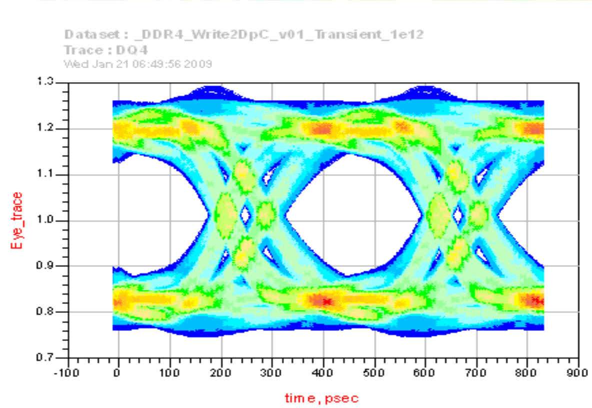
2) Channel Characterization
PRBS simulation with X-talk

- Simulating a long PRBS Pattern
  - Pattern length for worst case ISI is dependent on Channel memory
  - To get the worst case Eye requires combination with worst case X-talk
    - Do a VERY long Simulation with independent aggressors
    - Using even and odd Aggressor pattern only requires 2x PRBS length and results in nearly worst case Eye opening
- Calculate the Eye and evaluate:
  - Rise and Falltime
  - Jitter / Histogram
  - Eye Width and Height
  - Setup and Hold



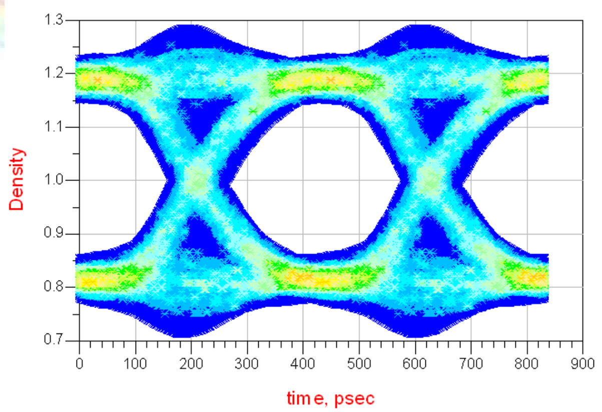
2) Channel Characterization Faster Alternatives

- Long Transient simulation

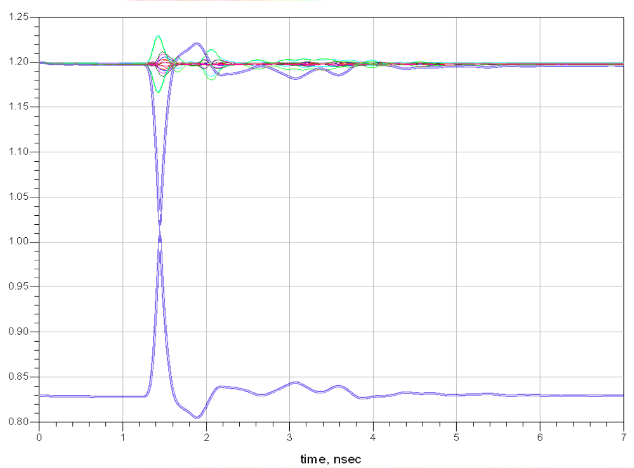


2^12 PRBS sim with even/odd X-talk

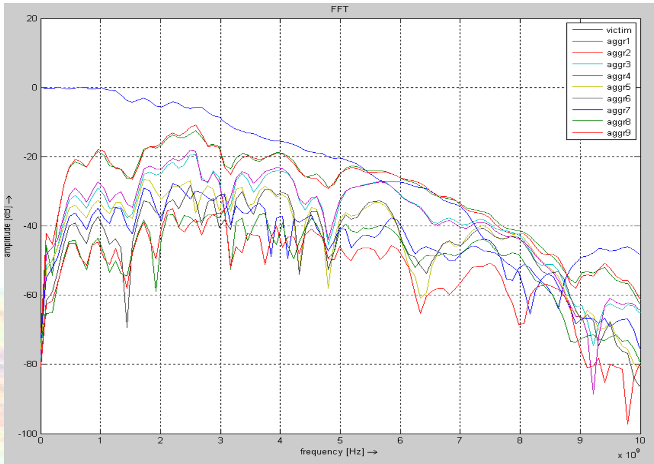
- ADS Fast channel Simulation



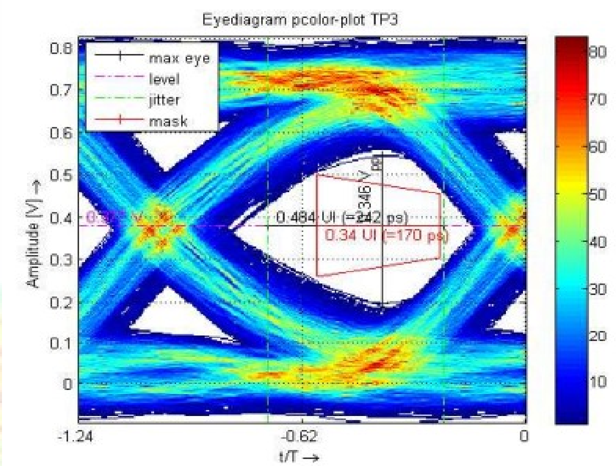
- Step Response



Rising and Falling Step response
Victim and Aggressors



Frequency domain
FFT on victim and aggressor



Eye Calculated from Step response

3) Channel Optimization Conventional Method

- Knowing the characteristics of the Channel is one thing.
  But how to optimize it ?
- The number of variables generates a huge matrix of possible combinations.
- One possible solution is to optimize each variable on it's own
  - Define a best guess combination of variables
  - Optimize Ron and set the new value as reference.
  - Optimize RTT and set the new value as reference.
  - … and so on and on and on …

This solution does takes a long time and will not find the global optimum as the variables are not independent!

3) Channel Optimization
Single Pulse as Eye Quality indicator

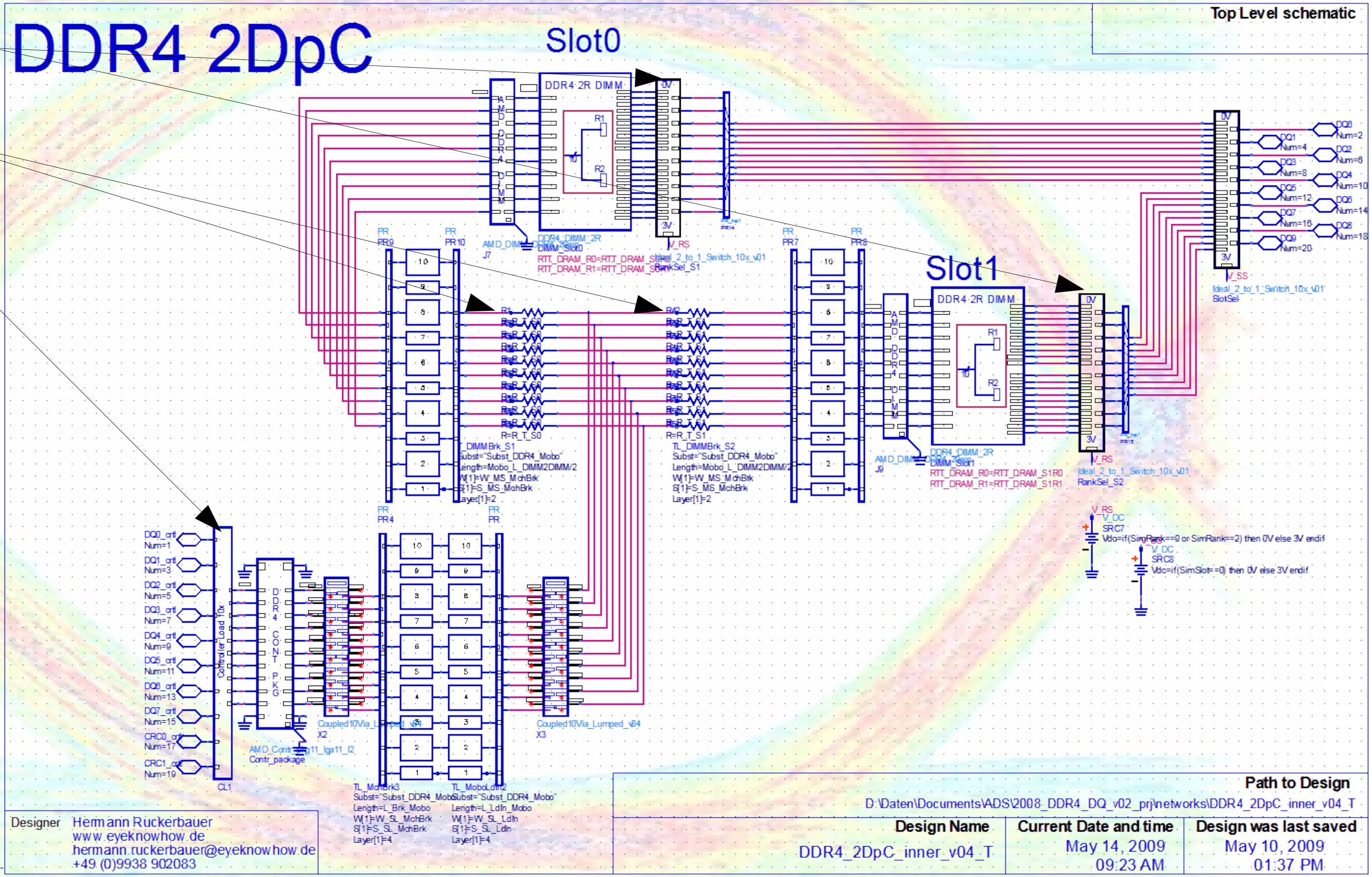
- How to utilize the ADS Optimizer for this task ?
- The Optimizer does needs direct feedback from the simulation, but usually Eye's are generated in the DataDisplay, or even in an external tool!
- BUT: Each function from the Data Display should work in the schematic too!
- Basically this would allow to use the Eye() function in the schematic. Disadvantage of this method:
  - Difficult to debug because of “blind” usage
  - Very long simulation times

Need a different parameter for the optimizer !

- **Use Pulse Response as indicator for eye quality!**

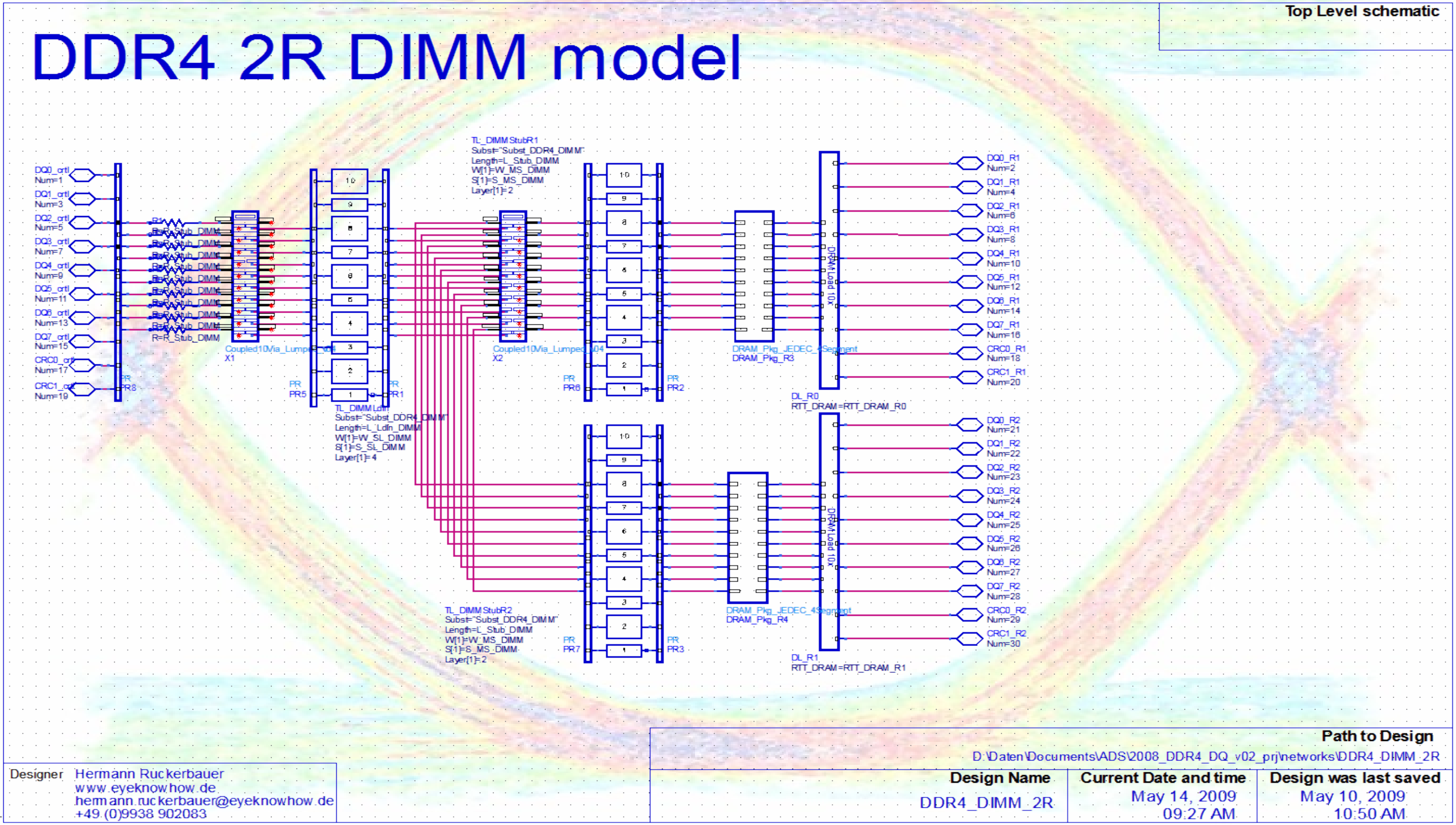
3) Channel Optimization
Example Simulation Schematic

- Variables for Optimizer:
  - RTT_eff
  - R_T
  - Ron



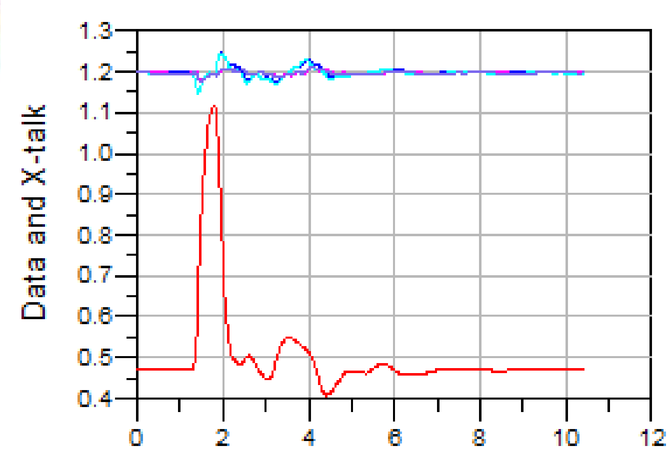
3) Channel Optimization
Example Simulation Schematic

- DIMM Schematic:

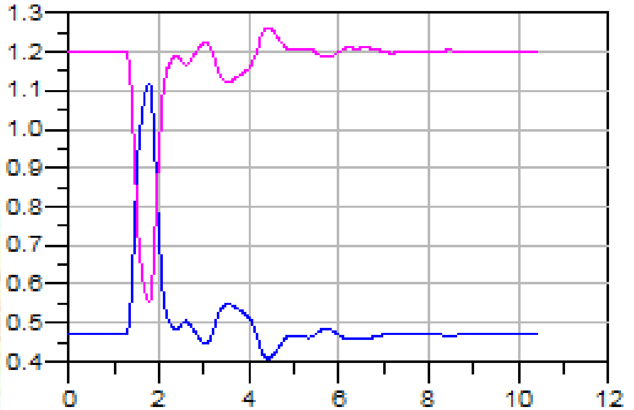


4) Optimize Pulse Area
Simulation setup with a single Pulse

- Characterizing the channel by UI wide pulse Response
  - Fast and simple setup
  - Includes rising and falling edge
  - Can include X-talk with single simulation
    - In first order passive X-talk is “symmetric”
    - Stimulate victim and look at aggressors

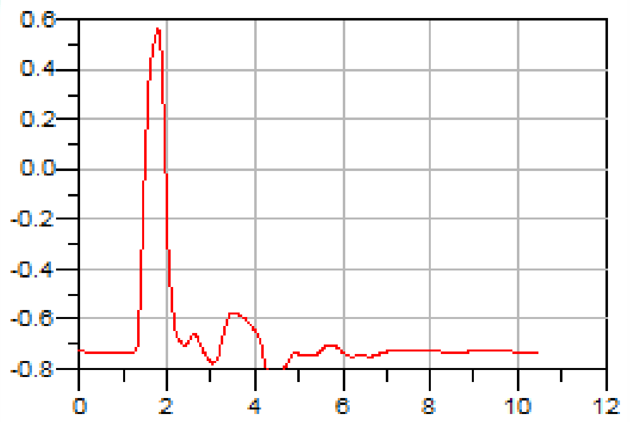


- Ideally two separate simulations are used for rising and falling pulse
  - Only a calculated “0” pulse was used:
    Sig_Fall=VTT-SigRise+LowLevel
  - This is not correctly including rising and falling characteristics of the driver, what would be possible with 2 simulations



4) Optimize Pulse Area
Separate "use" Signal Information

- Generation of a single Pulse by "Sig_Rise-Sig_Fall"
  - Area under this pulse is used for area inside eye
  - Integrate Result:



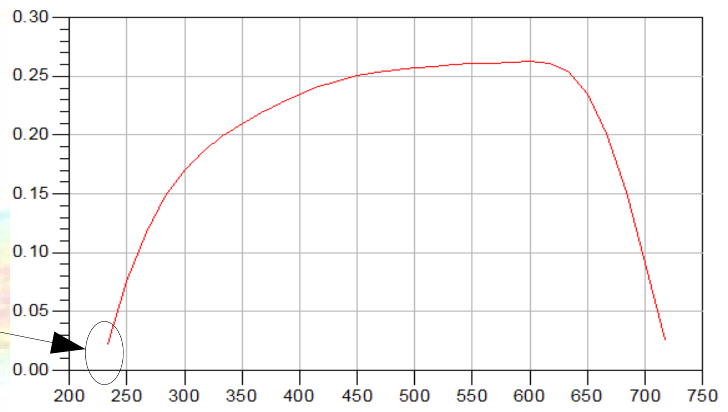
DQ4_Pulse_Area

  - Alternative is to use max Voltage level or Voltage @ sampling point

- Use 0-Xing to separate the interesting data UI
  - Extract index of first and second X-ing

```
UI_start_index=min(find(DQ4_PulseEye > 0))
UI_end_index=max(find(DQ4_PulseEye > 0))
```

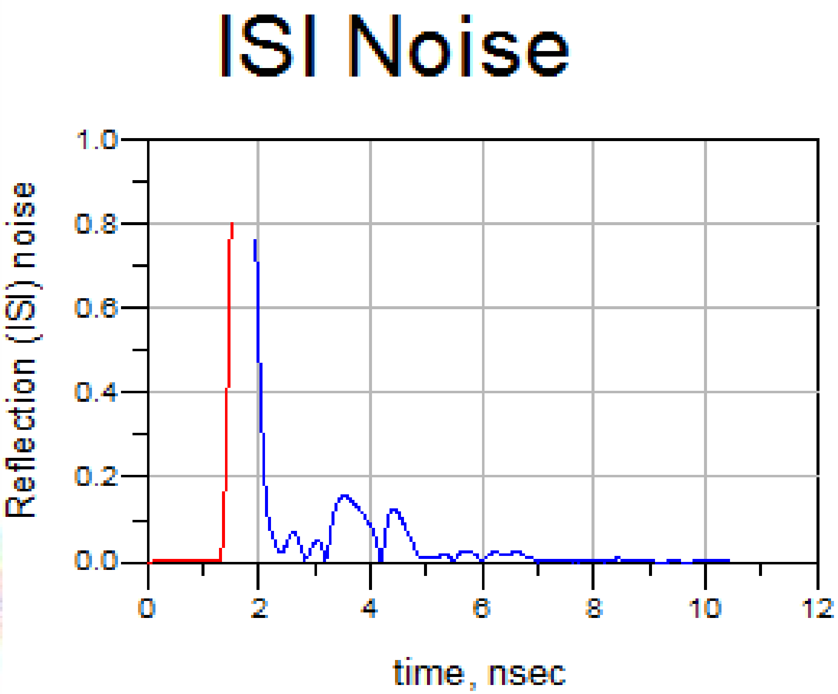
  - Implement a interpolation to get an exact 0 X-ing is a good idea for a stable function



4) Optimize Pulse Area
Calculate ISI noise

- Calculate ISI area
  - Take Signal from start to first X-ing
  - Take Signal trace from second X-ing to end
  - Integrate results and sum up to get the area (energy) of the ISI noise:

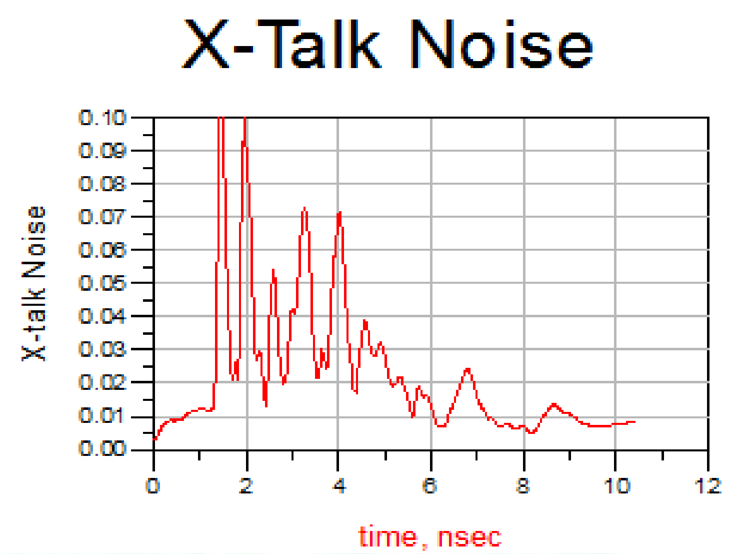
DQ4_ISI_Area



4) Optimize Pulse Area
Calculate X-talk noise

- Calculate X-talk area
  - In this example X-talk and ISI have been optimizted in parallel, but it might be a good idea to separate them:
    - The dominent variables for the effects are independent
    - X-talk Area might happen at crossing and therefore does not subtract from the “use Data” engergy
  - Sum of all: abs(VTT-Aggressor)
  - Can be done over the whole trace:
    - No need to separate data UI
  - In this example X-talk happens at X-ing ==> the area can be scaled down
- Integrate the result:

DQ4_Xtalk_Area

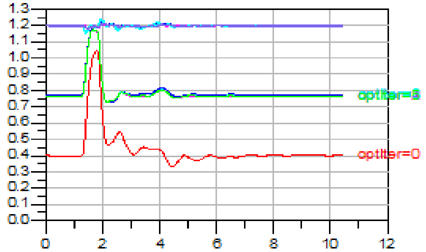


4) Optimize Pulse Area Data in the DDS

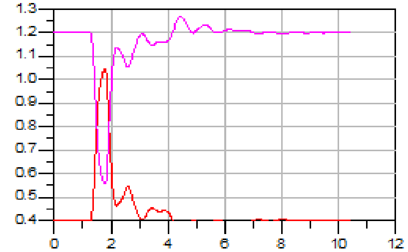
Project Directory	Design Name	Default Dataset Name
D:\Daten\Documents\ADS\2008_DDR4_DQ_v02_prj	_DDR4_Write2DpC_Pulse_TOP_v08_O... .dsn	_DDR4_Write2DpC_Pulse_TOP_v08_Opti... .ds

"use Data" processing

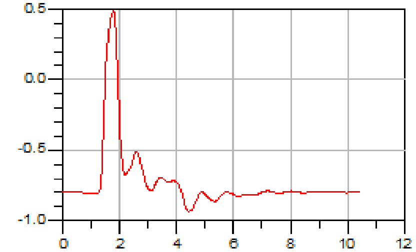
Simulated Pulse



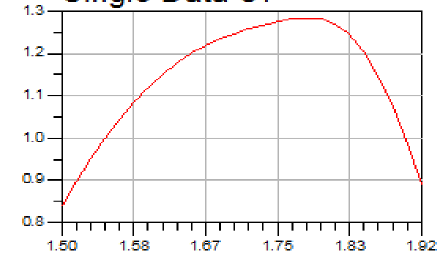
Rise and Fall Pulse



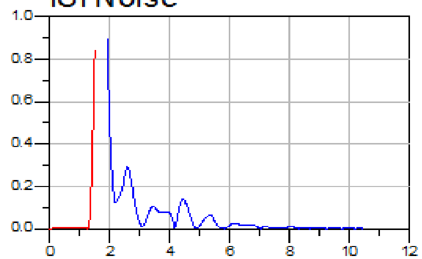
"Differential" Pulse



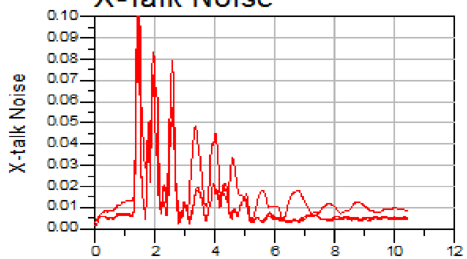
Single Data UI



ISI Noise



X-Talk Noise



Optimization Steps

optIter	..._DIMM2DIMM	OPTIM.Ron_Ctrl	...IM.RTT_SysEff	Goal_DQ_Area
0	30.000	10.000	35.000	-0.033
1	30.700	10.000	8.949	0.170
2	36.403	10.000	10.160	0.193
3	35.868	10.000	10.016	0.207
4	36.042	10.000	10.062	0.208
5	36.042	10.000	10.062	0.208

DQ4_Pulse_Area - DQ4_ISI_Area - DQ4_Xtalk_Area = Goal_DQ_Area

Designer
Hermann Ruckerbauer
hermann.ruckerbauer@EyeKnowHow.de
+49 (0)9938 902083
www.EyeKnowHow.de

Current time: Mon May 11, 2009 22:18:22

Simulated: Tue May 12, 2009 01:47:56

4) Optimize Pulse Area
Signal area – Noise Area

- Calculate Final Signal Area and run the optimizer on this

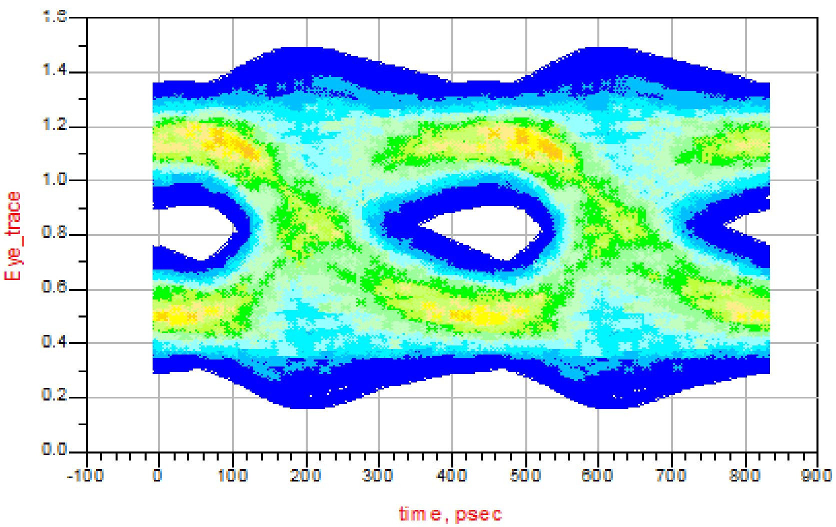
optIter	Goal_DQ_Area	OPTIM.R_DIMM2DIMM	OPTIM.Ron_Ctrl	OPTIM.RTT_SysEff
0	-33.06 m	30.00	10.00	35.00
1	169.9 m	30.70	10.00	8.949
2	193.4 m	36.40	10.00	10.16
3	207.1 m	35.87	10.00	10.02
4	207.8 m	36.04	10.00	10.06
5	207.8 m	36.04	10.00	10.06

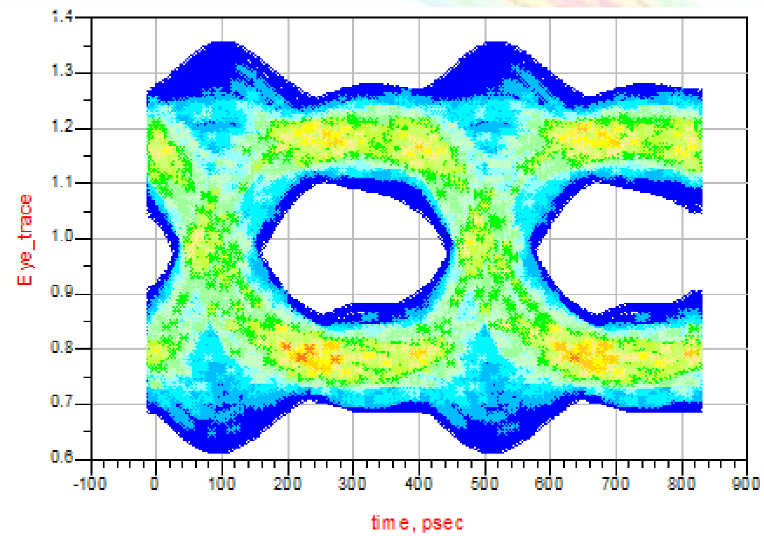
- Complete equation set for pulse area optimization

```
MeasEqn
DQ4_AreaCalculation
DQ4_RiseCalculated=if (sweep_dim(Optim_Sim.TRAN.DQ4) > 1) then DQ4[0,::] else Optim_Sim.TRAN.DQ4 endif
DQ4_RiseCalculated1=DQ4
DQ4_RiseDimAdjust=if (sweep_dim(DQ4_RiseCalculated) ==2) then (expand(DQ4_RiseCalculated[0])) else DQ4_RiseCalculated[0] endif
DQ4_FallCalculated=VDDQ-DQ4_RiseCalculated+DQ4_RiseCalculated[0]
DQ4_PulseEye=DQ4_RiseCalculated-DQ4_FallCalculated
UI_index=find(DQ4_PulseEye < 0)
UI_start_index=min(find(DQ4_PulseEye > 0))
UI_end_index=max(find(DQ4_PulseEye > 0))
DQ4_UI=DQ4_PulseEye[UI_start_index::UI_end_index]
UI_start_time=min(indep(DQ4_UI))
UI_end_time=max(indep(DQ4_UI))
DQ4_PulseWidth_index=UI_end_index-UI_start_index
DQ4_PulseWidth_time=UI_end_time-UI_start_time
DQ4_PulseHeight=max(DQ4_PulseEye)
DQ4_PulseArea=(integrate(DQ4_UI))
DQ4_Reflection_End=abs(DQ4_PulseEye[UI_end_index::max(find(DQ4_PulseEye))]-DQ4_PulseEye[0])
DQ4_Reflection_Start=abs(DQ4_PulseEye[0::UI_start_index]-DQ4_PulseEye[0])
DQ4_ReflectArea=integrate(DQ4_Reflection_Start)+integrate(DQ4_Reflection_End)
DQ4_Area=DQ4_PulseArea-DQ4_ReflectArea
DQ0_XtalkArea=integrate(abs(DQ0-DQ0[0]))
DQ1_XtalkArea=integrate(abs(DQ1-DQ1[0]))
DQ2_XtalkArea=integrate(abs(DQ2-DQ2[0]))
DQ3_XtalkArea=integrate(abs(DQ3-DQ3[0]))
DQ5_XtalkArea=integrate(abs(DQ5-DQ5[0]))
DQ6_XtalkArea=integrate(abs(DQ6-DQ6[0]))
DQ7_XtalkArea=integrate(abs(DQ7-DQ7[0]))
DQ8_XtalkArea=integrate(abs(DQ8-DQ8[0]))
DQ9_XtalkArea=integrate(abs(DQ9-DQ9[0]))
DQall_XtalkArea=DQ0_XtalkArea+DQ1_XtalkArea+DQ2_XtalkArea+DQ3_XtalkArea+DQ5_XtalkArea+DQ
DQ4_FinalArea=(DQ4_PulseArea-DQ4_ReflectArea-DQall_XtalkArea)*1e9
```

4) Optimize Pulse Area
Optimization Results and Data eyes

- **Data Eye before Optimization**
- **Data Eye after Optimization**





Measurement Results

Eye Level Zero	0.77866992863014
Eye Level One	1.1816964383729
Eye Level Mean	0.98018318350153
Eye Amplitude	0.40302650974278
Eye Height	0.21600396411977
Eye Height (db)	-6.655382785734
Eye Width	2.9194383e-010
Eye Opening Fa...	0.83984541245133
Eye Signal_to_N...	6.243967252553
Eye Duty Cycle ...	2.33075e-012
Eye Duty Cycle ...	0.55937954325496
Eye Rise Time	1.2126543e-010
Eye Fall Time	1.2194645e-010
Eye Jitter (PP)	4.9334812e-010
Eye Jitter (RMS)	2.0050531e-010

4) Optimize Pulse Area Advantages / Drawbacks

- Advantages of the Area calculation method
  - “Easy” to implement and use
  - No AEL programming is necessary
- Drawbacks of the Area calculation method
  - Area is not all: Sometimes it is helpful to trade timing vs. voltage margin
  - X-talk Area might be not reducing the “use” data energy dependent on the phase relation of the X-talk (e. g. Source Sync signals)
  - Eye is reduced to a single number
  - Debugging is difficult:
    - Variable Dimensions are different during optimization, in the the DDS and in a transient simulation
    - Need to create separate equations or an automatic dimension check and dimension adjustment (what is simpler in AEL)

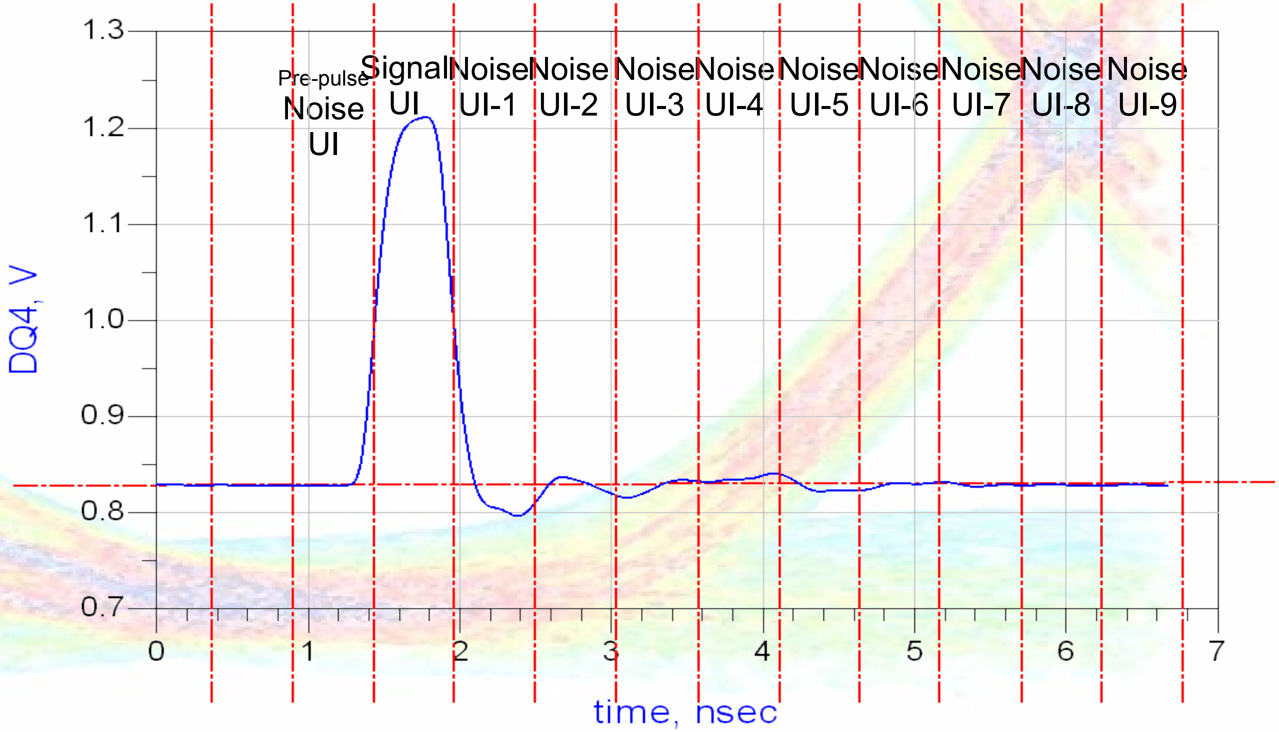
5) Optimize Pulse Shape Alternative Solution

- Alternative solution:
  - Completely slice all Traces into UIs
  - Subtract all the ISI-UI's from the “use Data” UI
  - Subtract all the X-talk UI's from the “use Data” UI
- Advantages:
  - Result is a pulse shape which is close to real eye shape
  - Optimizing can be done on Hight and/or Width and/or Area or ....
  - X-talk can be taken into account with the correct phase relation
    - Should be considered in the simulation setup
- Disavantage:
  - More complex to implement directly in schematic, better to use AEL

Original AEL code was implemented by Suhas Jawale!

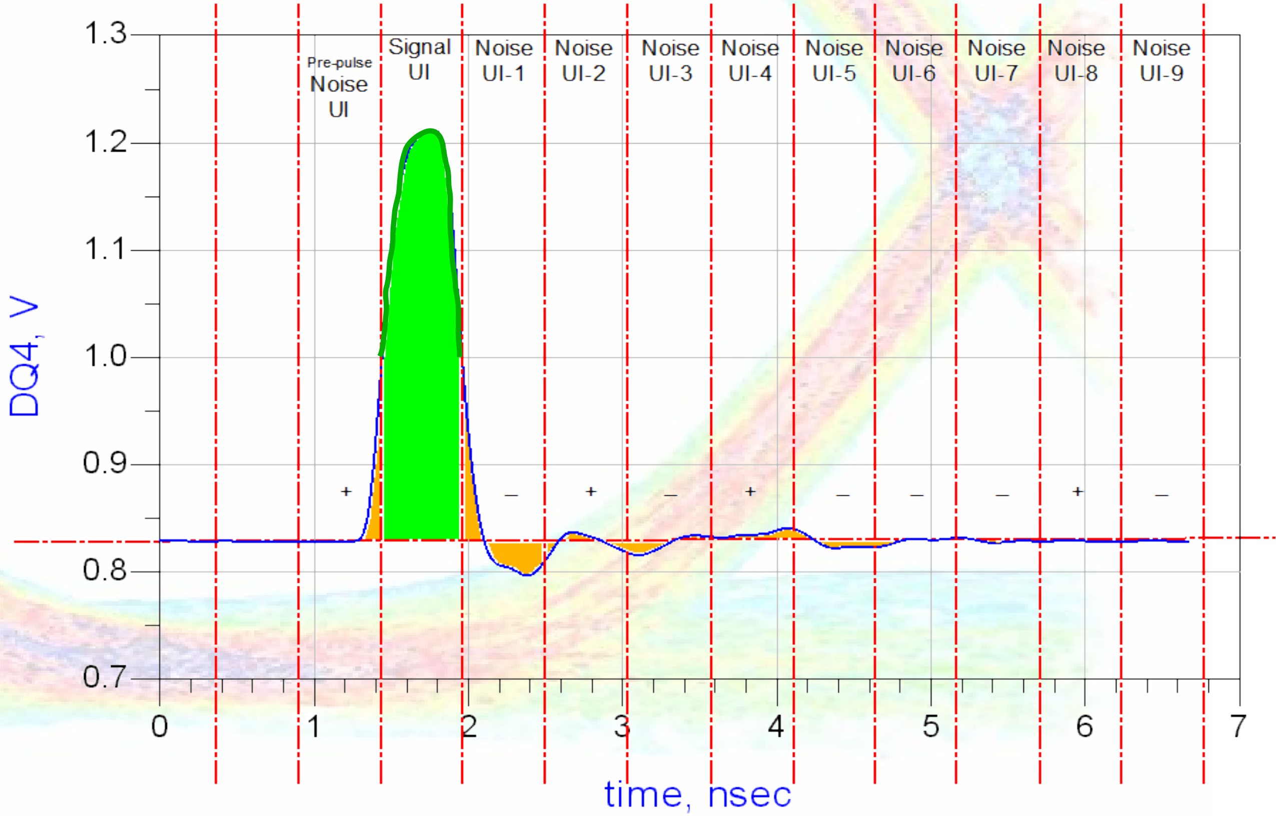
5) Optimize Pulse Shape
UI Slicing

- Cut the transient Pulses into slices
  - Take "differential" pulse and use "0" X-ing to find begin and end.
  - Center start to end inside UI



5) Optimize Pulse Shape
Subtracting noise UI's from use data

- Start with the green "use data" UI shape
  - Subtract (abs) Traces from all ISI UI's
  - Subtract (abs) Traces from all X-Talk UI's

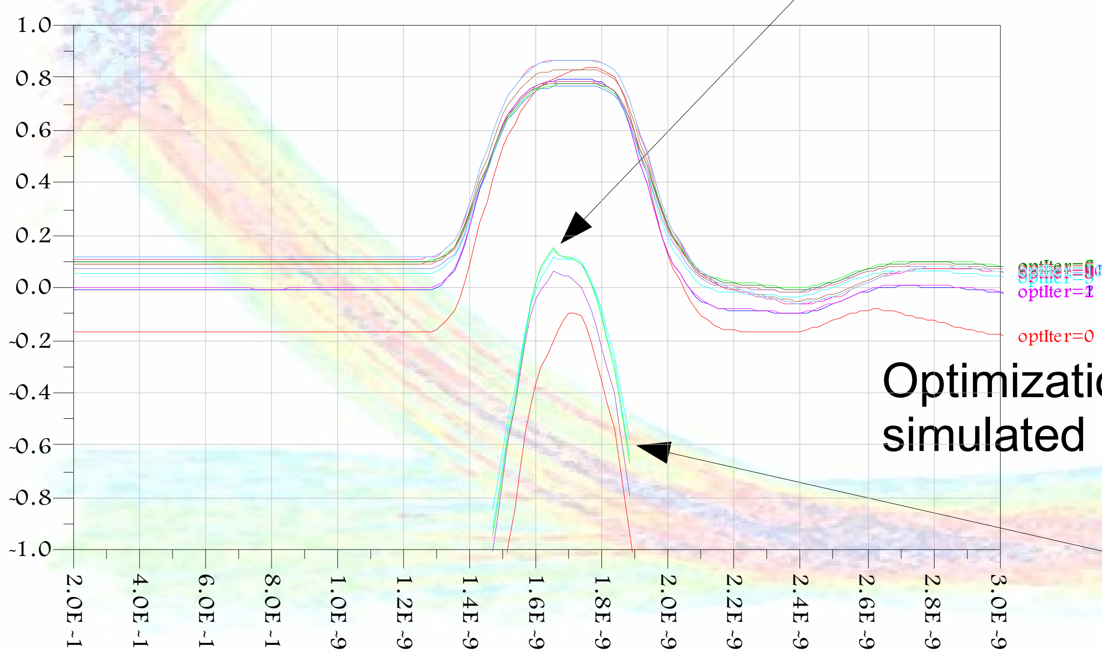


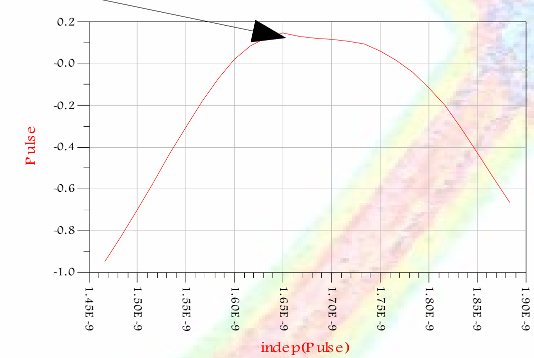
5) Optimize Pulse Shape
Optimization goal and steps

- AEL Function return:

indep(Pulse_Parameter)	Pulse_Parameter
max eye height[0]	0.149
eye_width[1]	1.747E-10
eye_start_time[2]	1.596E-9
eye_end_time[3]	1.771E-9
eye area[4]	1.690E-11
max pulse height[5]	0.660
pulse_width[6]	4.167E-10
pulse_start_time[7]	1.467E-9
pulse_end_time[8]	1.883E-9
pulse area[9]	2.538E-10
number of UIs[10]	22.000

- Optimizer Goal: maxeye_height





Optimization steps of simulated pulse

Optimization steps of Calculated UI

5) Optimize Pulse Shape Optimization Results

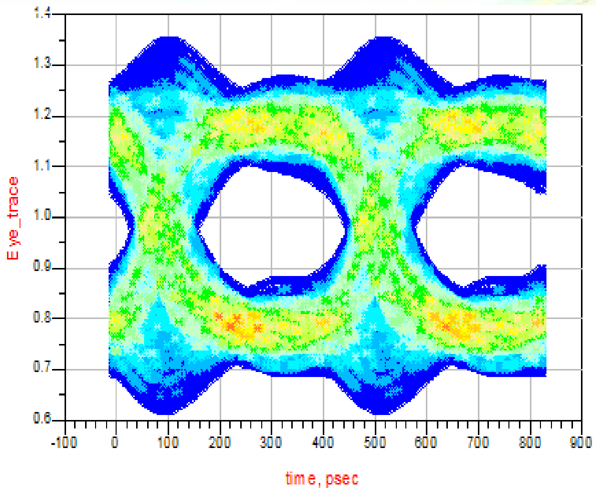
- Optimization sim results for EyeHeight sims:

optIter	...TIM.R_DIMM2DIMM	OPTIM.Ron_Ctrl	OPTIM.RTT_SysEff	Goal_DQ_UI_CAD
0	20.00	20.00	20.00	161.9 m
1	17.57	21.89	11.32	268.5 m
2	17.67	22.06	11.13	268.7 m
3	17.78	30.11	11.53	300.8 m
4	15.80	33.92	11.57	314.6 m
5	11.28	39.48	11.59	320.6 m
6	11.09	38.05	11.46	327.3 m
7	11.03	36.30	11.40	328.2 m
8	4.513	35.46	11.24	344.8 m
9	146.7 m	35.77	11.07	356.6 m
10	106.1 m	36.56	11.07	357.7 m
11	107.8 m	36.64	11.07	357.8 m
12	107.8 m	36.64	11.07	357.8 m

- Values used for characterization simulation:
  - RTT_SysEff: 11 Ohm
  - Ron = 36 Ohm
  - R_T = 0 Ohm

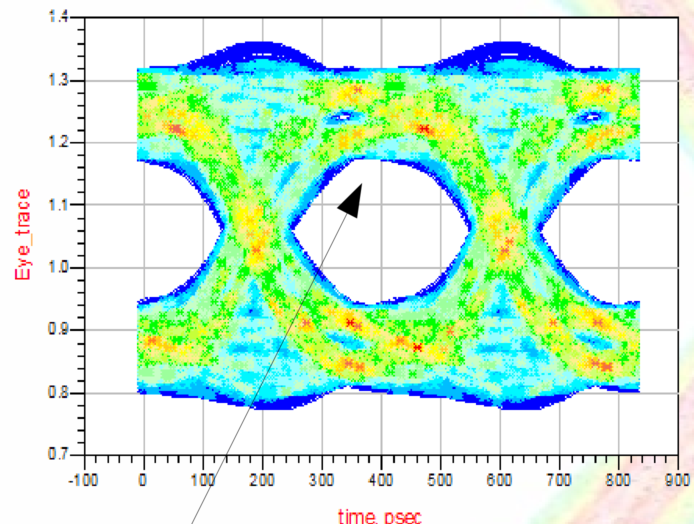
5) Optimize Pulse Shape Resulting Data Eye

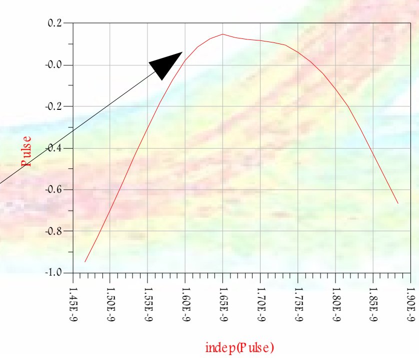
- Eye with parameters found by area optimization



Measurement Results	
Eye Level Zero	0.77866992863014
Eye Level One	1.1816964383729
Eye Level Mean	0.98018318350153
Eye Amplitude	0.40302650974278
Eye Height	0.21600396411977
Eye Height (db)	-6.655382785734
Eye Width	2.9194383e-010
Eye Opening Fa...	0.83984541245133
Eye Signal_to_N...	6.243967252553
Eye Duty Cycle ...	2.33075e-012
Eye Duty Cycle ...	0.55937954325496
Eye Rise Time	1.2126543e-010
Eye Fall Time	1.2194645e-010
Eye Jitter (PP)	4.9334812e-010
Eye Jitter (RMS)	2.0050531e-010

- Eye with parameters found by height optimization





Measurement Results	
Eye Level Zero	0.87643189221399
Eye Level One	1.2472481418715
Eye Level Mean	1.0618400170428
Eye Amplitude	0.37081624965755
Eye Height	0.22696551822416
Eye Height (db)	-6.4404011806007
Eye Width	3.1596452e-010
Eye Opening Fa...	0.81736505496026
Eye Signal_to_N...	5.4754034053144
Eye Duty Cycle ...	7.7344e-013
Eye Duty Cycle ...	0.18562663168974
Eye Rise Time	1.5108563e-010
Eye Fall Time	1.5104699e-010
Eye Jitter (PP)	1.1456024e-010
Eye Jitter (RMS)	2.511066e-011

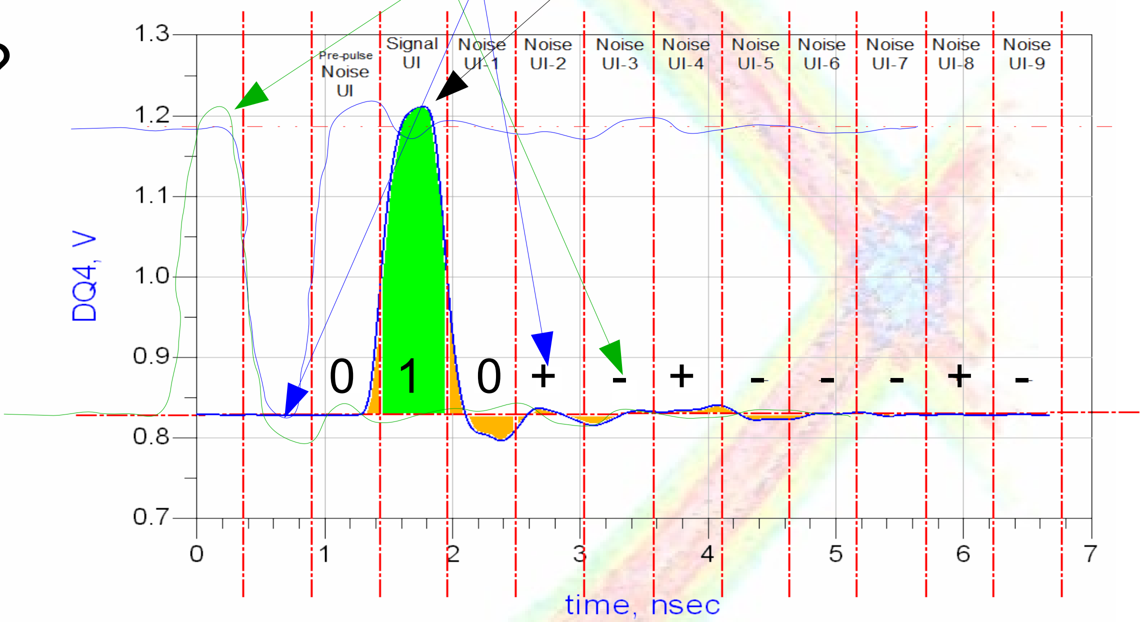
Good Matching of eye Shape

5) Optimize Pulse Shape Comparison with Area optimization

- Improvement of the shape over the area optimization
  - 5% better Eye Height
  - 6% (25ps) better Eye Width
  - Less Overshoot
  - Higher Rise/Fall Time
  - Better Eye Quality at the end of the eye (hold time improves)
- The result fo the pulse height optimization was around 12% better than the manual optimization of each parameter separatly (with more parameters, not shown in this presentation)

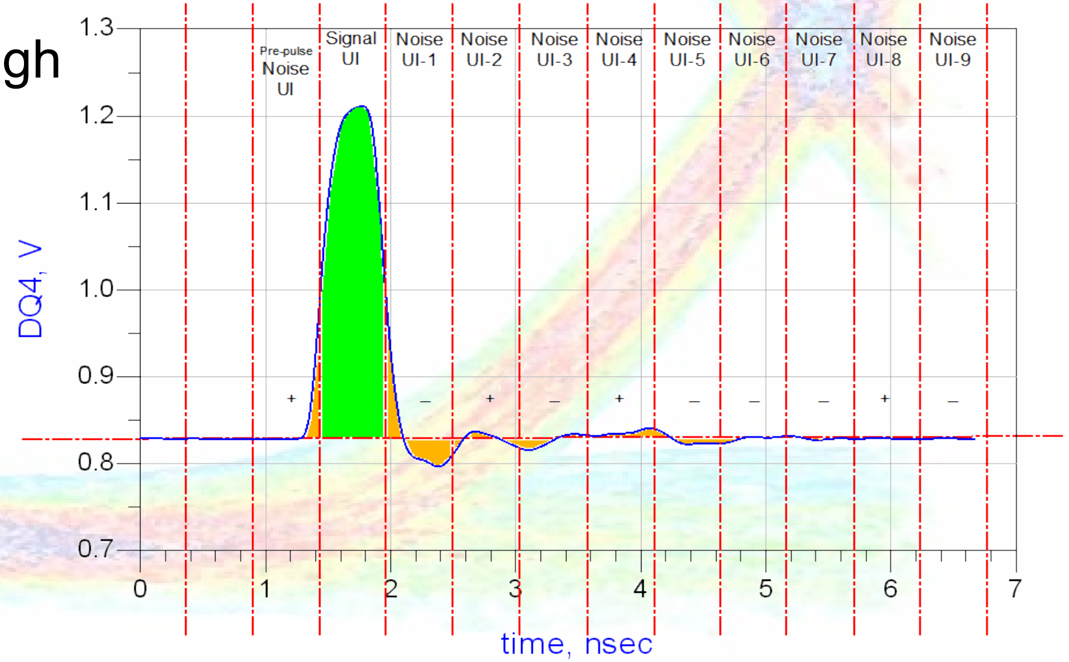
6) Outlook: Additional options
Calculate worst case pattern

- The “use” pulse needs to be a single pulse, so “010”
- Reverse ordering and polarity of + and – as bitsequence.
- In this case the pattern would be “10111010 010”.
- What to take as “+” or “-” ?
  - Area for this UI ?
  - Level at capture ?



6) Outlook: Additional options Length of Channel Memory

- After optimization the channel still needs to be characterized by Eye simulation and Timing Budget calculation.
- If a PRBS simulation is used (instead of fast channel sim) the simulation time can be limited by using the shortest necessary PRBS pattern
- Calculate what is the minimum PRBS for a characterization simulation:
  - Sum up the noise area for a 20bit pulse
  - Calculate after which bit you have 98% of the noise area
- In this case a 2^10 bit PRBS is enough



7) Summary

- Using pulse Repsonse as basis for optimization allows to use the ADS optimizer efficently for SI optimization.
- Both methods shown here can be used
  - There are options to improve the Area based optimization e. g. by scaling the effect of X-talk
  - Instead of using the Area the level at the sampling point can be used. In this case the x-talk effect is taken into account including the phase information!
- Things to consider:
  - Even it is possible to optimize ISI and X-talk at the same time it can make sense to optimize the the two effects separatly
  - Setting two Goals (e. g. height and width) is possible, but it is difficult to adjuste the weighting!
  - Normalization of the Signal might be a problem
  - Using different “OptimType” can lead to different results

EyeKnowHow
Signal Integrity Consulting

Thank you for your attention

Services and Know How