Channel Simulation Over DDR4/5 and Above

Kumar Keshavan, Ambrish Varma, Ken Willis, Skipper Liang
Asian IBIS Summit
Shanghai, China
November 1, 2019

cadence®

Two Concerns:

- As the transmission rate of memory bus goes beyond 5Gbps, besides the well-known timing and overshoot/undershoot analysis, it requires BER prediction analysis and channel analysis

- Two additional concerns we need to face while using channel engine to deal with memory bus:
  - Asymmetric rising/falling edges

    Different from differential serial buses, single-ended memory buses will have non-symmetrical rising and falling edges due to the inherent difference between these two kinds of circuits

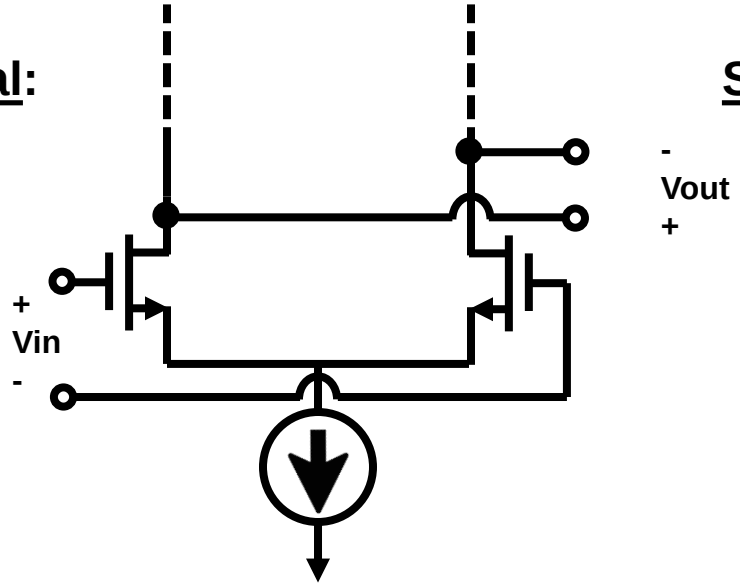
  - Strobes as timing reference

    While the sampling clock ticks in serial bus are recovered from the signal itself by CDR, the sampling clocks or the timing references in memory bus will be the strobes rather than any recoveries

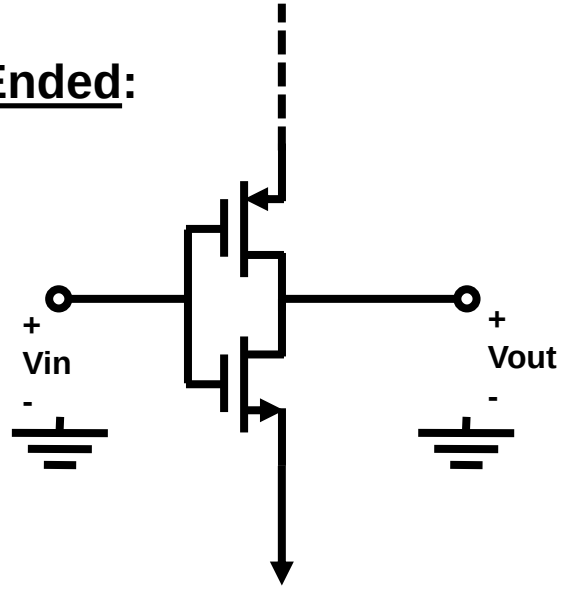
Asymmetric Rising and Falling Edges

Circuits

Differential:

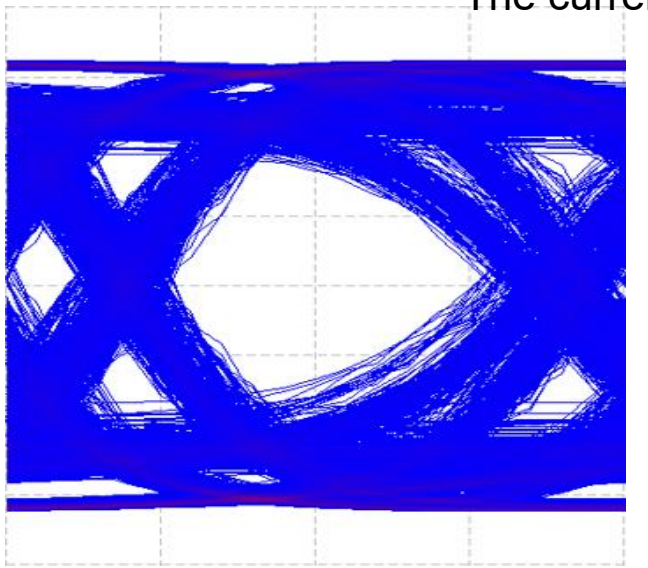


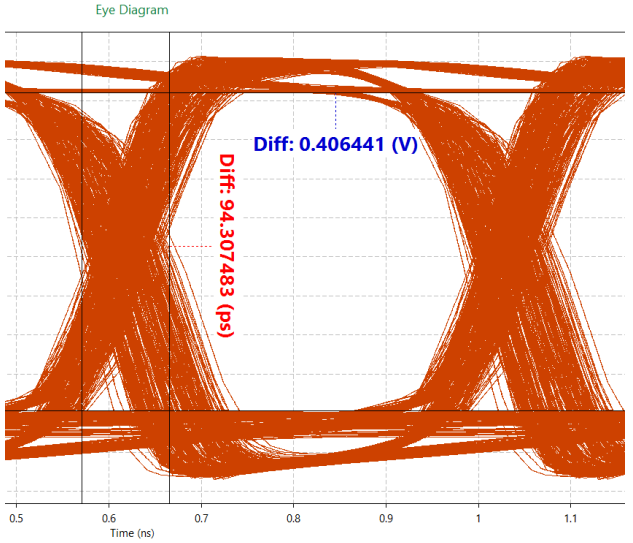
Single-Ended:



The current path during a binary “1” bit

The current path during a binary “0” bit



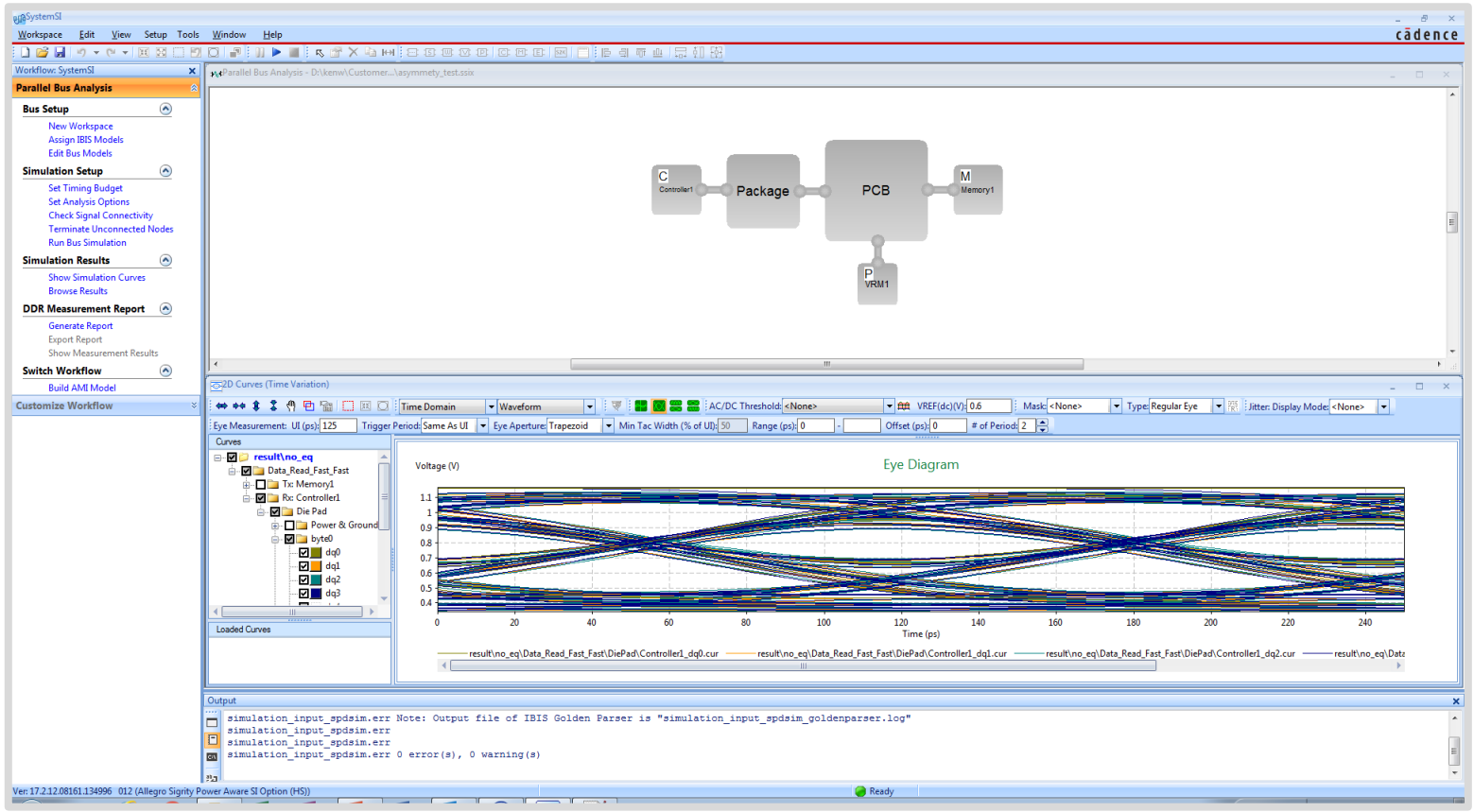


Overview

- As channel simulation and IBIS-AMI modeling methods are adapted from serial link to DDR interface analysis, we encounter IBIS I/O models with asymmetric rising and falling edges
- This is different from the highly symmetric drivers we typically encounter with serial link analysis
- Traditional single-step response methods for impulse response generation may not reproduce circuit simulation results accurately enough
- These slides show how an EDA tool can handle this (without changes to the IBIS specification)
- All cases use **Micron**'s **y11a.ibs** file for 8Gbps DDR5

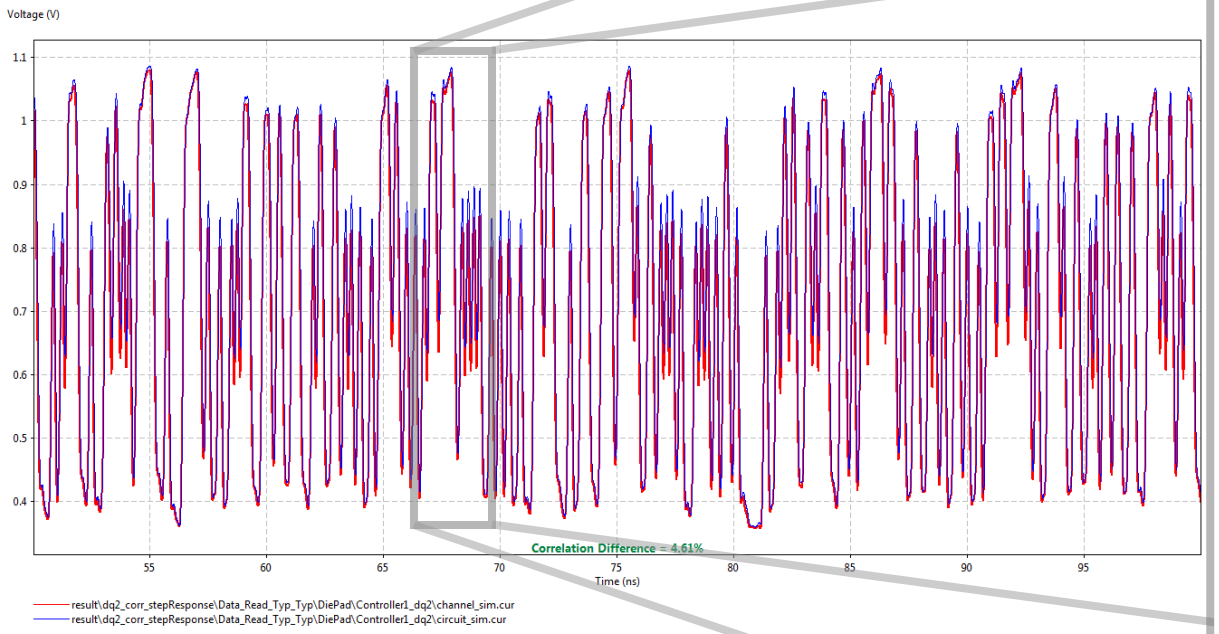
Testbench

- Package block uses an extracted RLCK SPICE model
- PCB block uses W-elements with 0.3 meter lengths



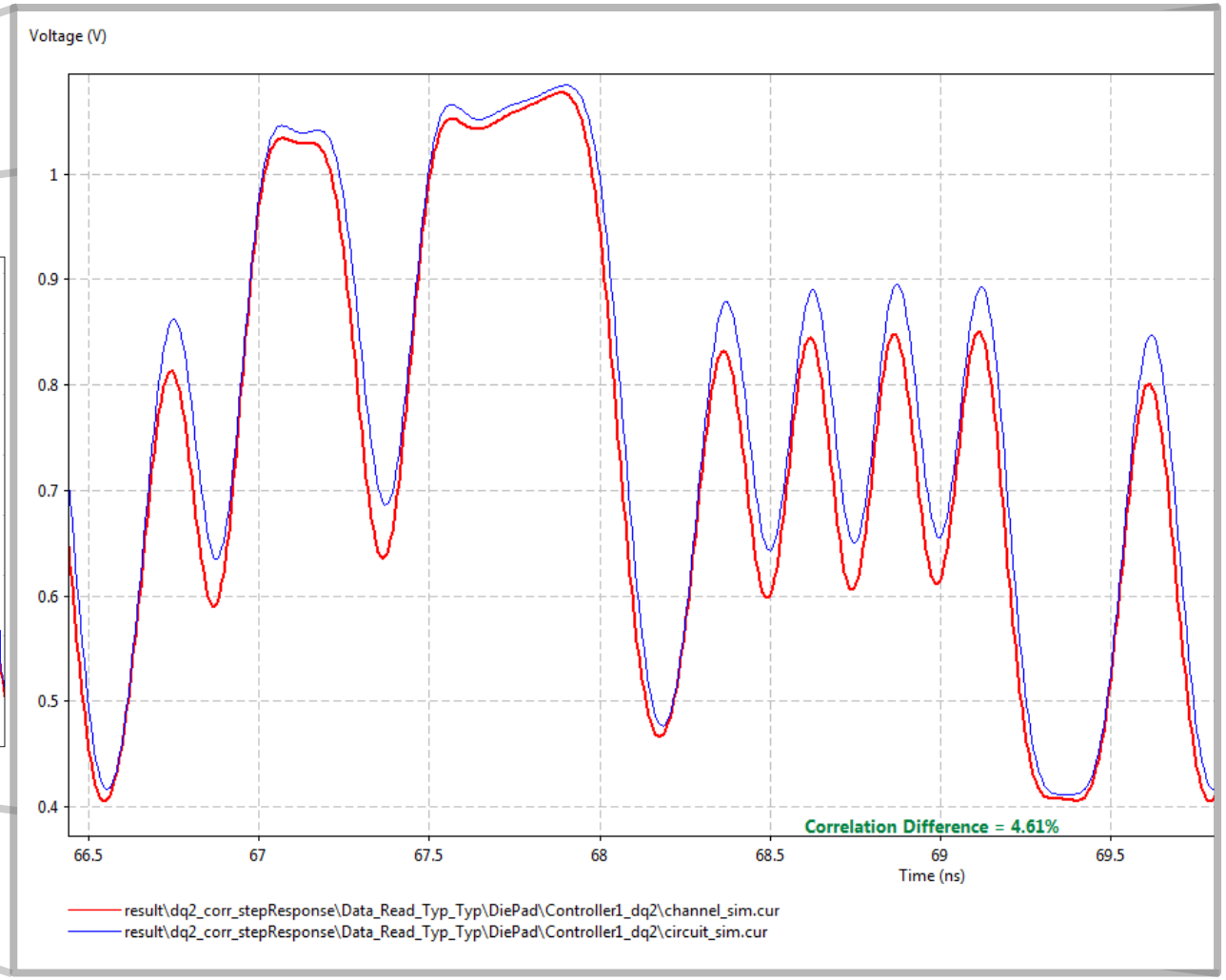
Characterizing with Step Function – One Rising Transition

- Correlation difference 4.61%



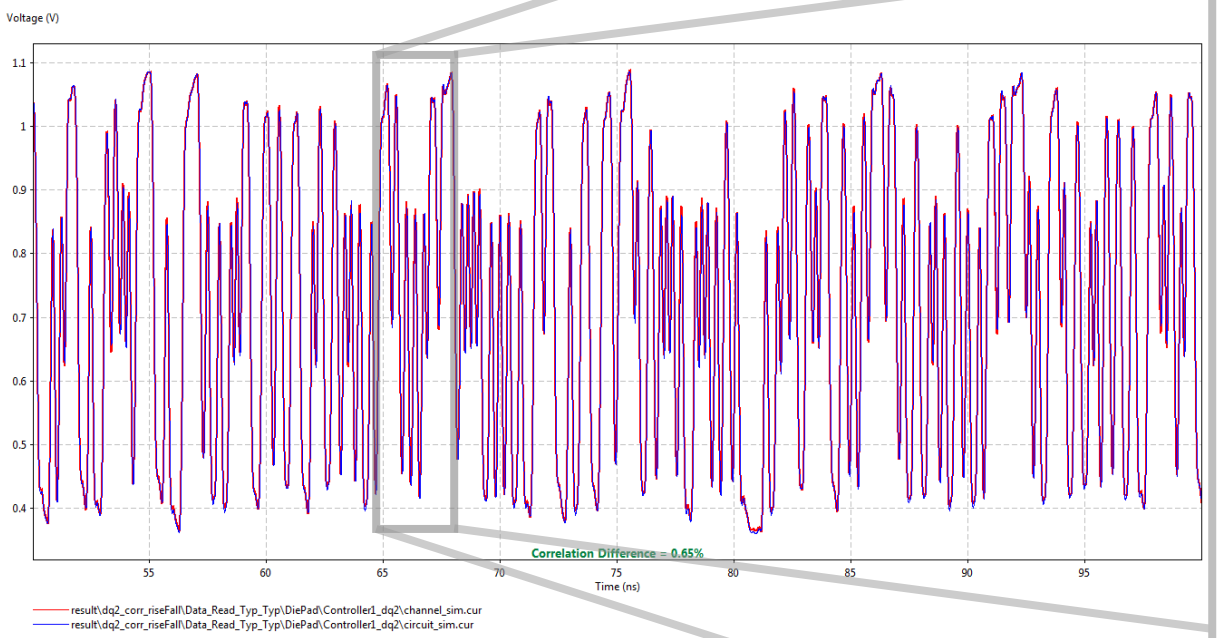
Circuit Sim

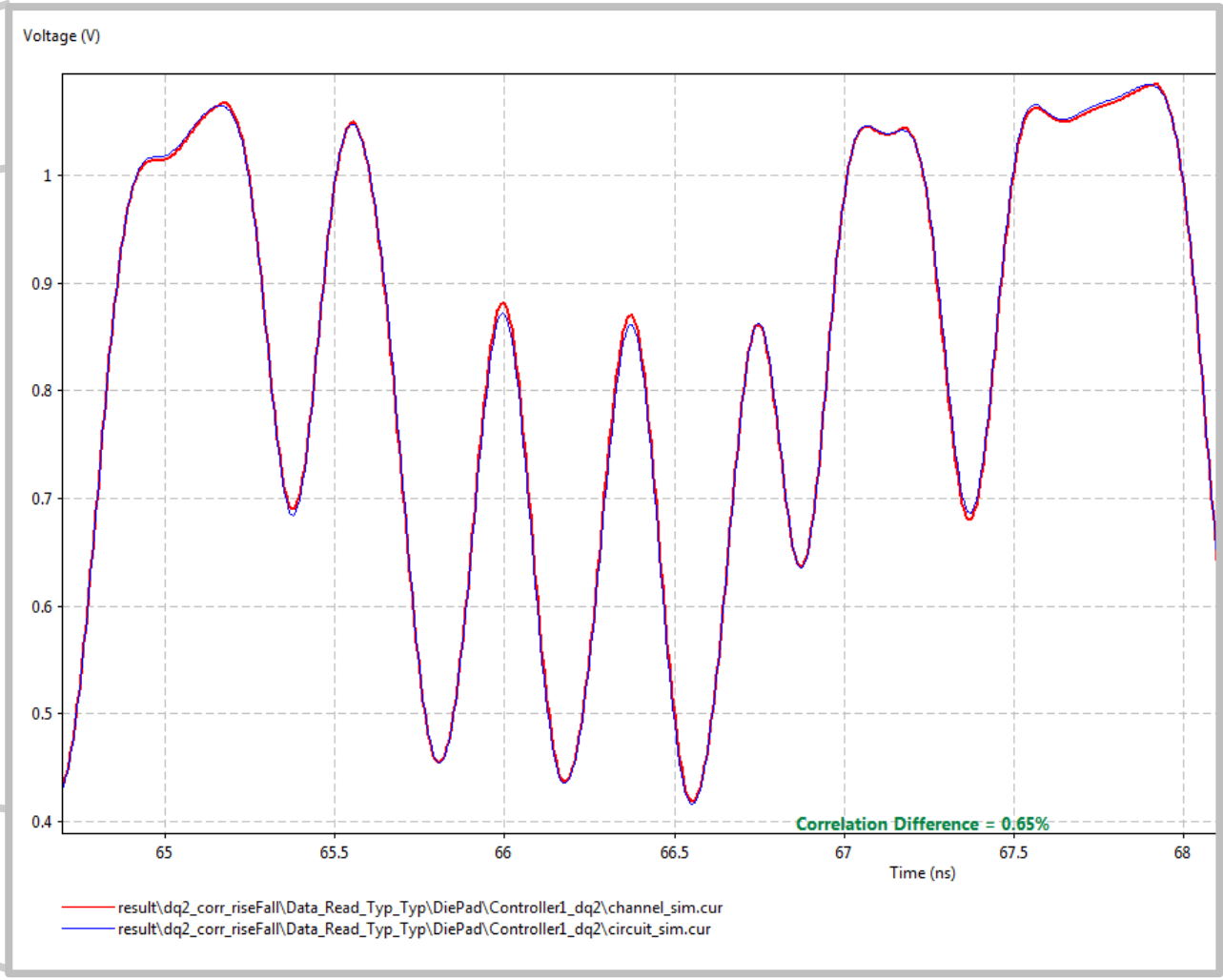
Channel Sim



Characterizing with both Rising and Falling Transition

- Correlation difference 0.65%



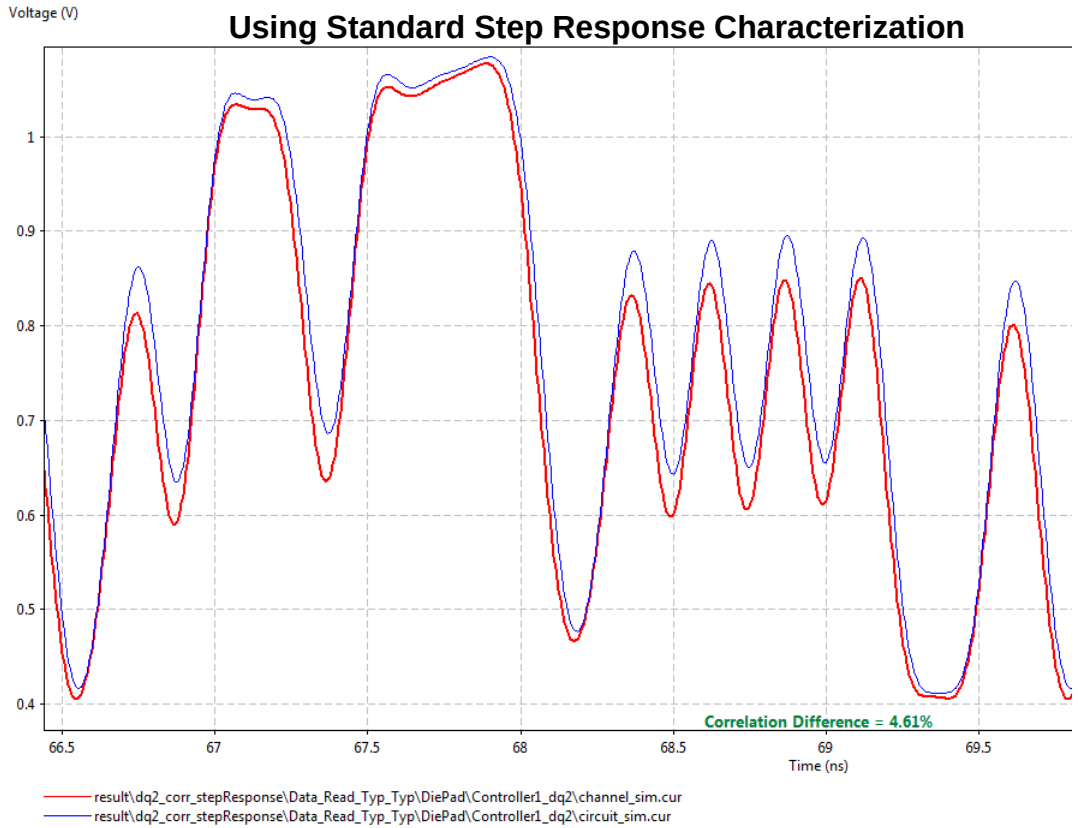


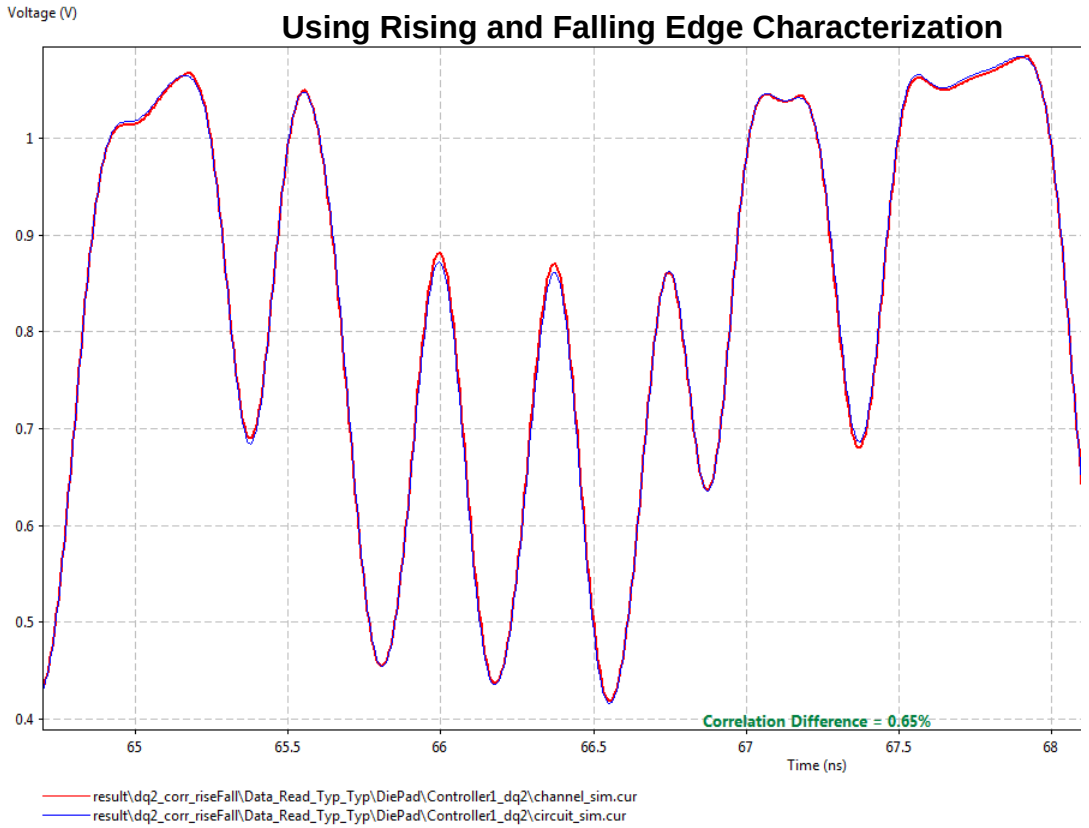
Circuit Sim

Channel Sim

Comparison between Two Different Methods

- Correlation error vs. circuit simulation reduced by about 4%





Summary

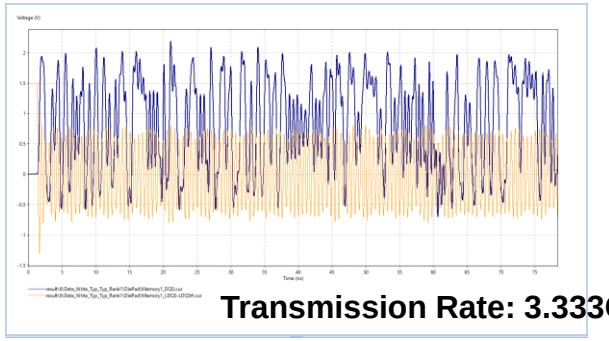
- The **DQ_34_3600** I/O model has some asymmetry in its rising and falling edges
- Standard SerDes step response characterization did not do a great job in capturing this behavior, as seen in the circuit / channel sim correlation
- Characterization methods using rising and falling edges captured this behavior very well for channel simulation

```
|**************************************************************************
|
[Ramp]
R_load = 50
|
|          typ                          min                          max
|
dV/dt_r 3.7990E-01/6.4024E-11    3.4860E-01/8.7846E-11    4.0938E-01/4.9359E-11
dV/dt_f 4.4605E-01/5.4840E-11    4.2048E-01/7.6564E-11    4.6793E-01/4.2557E-11
|**************************************************************************
```

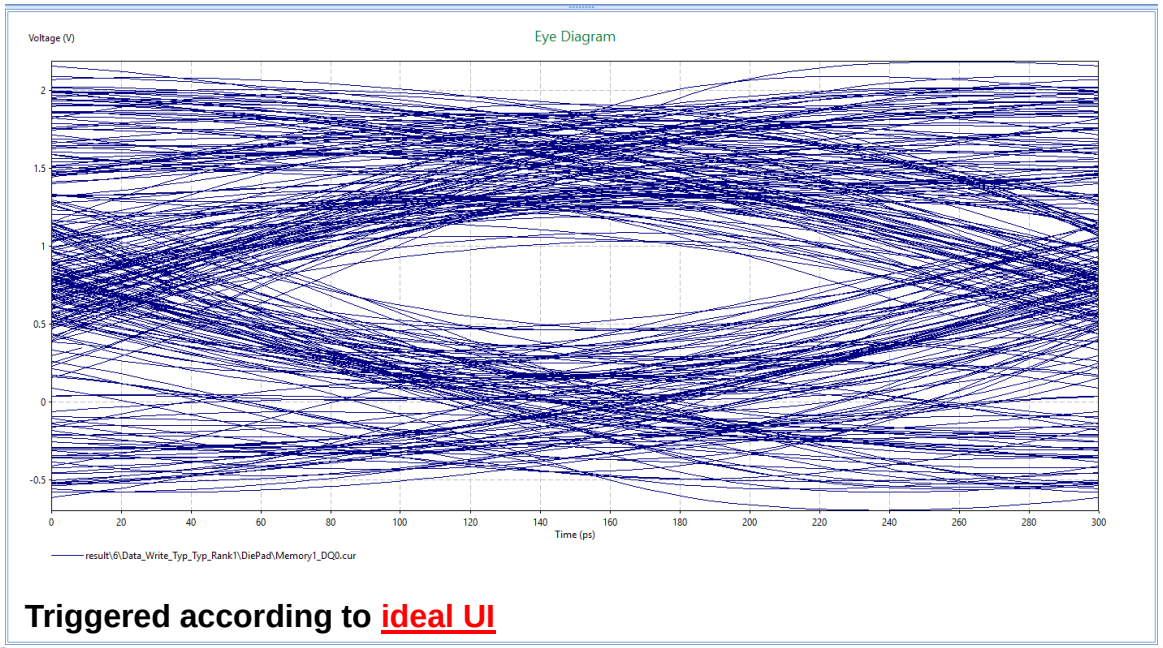
Strobe as Timing Reference

Overview(1)

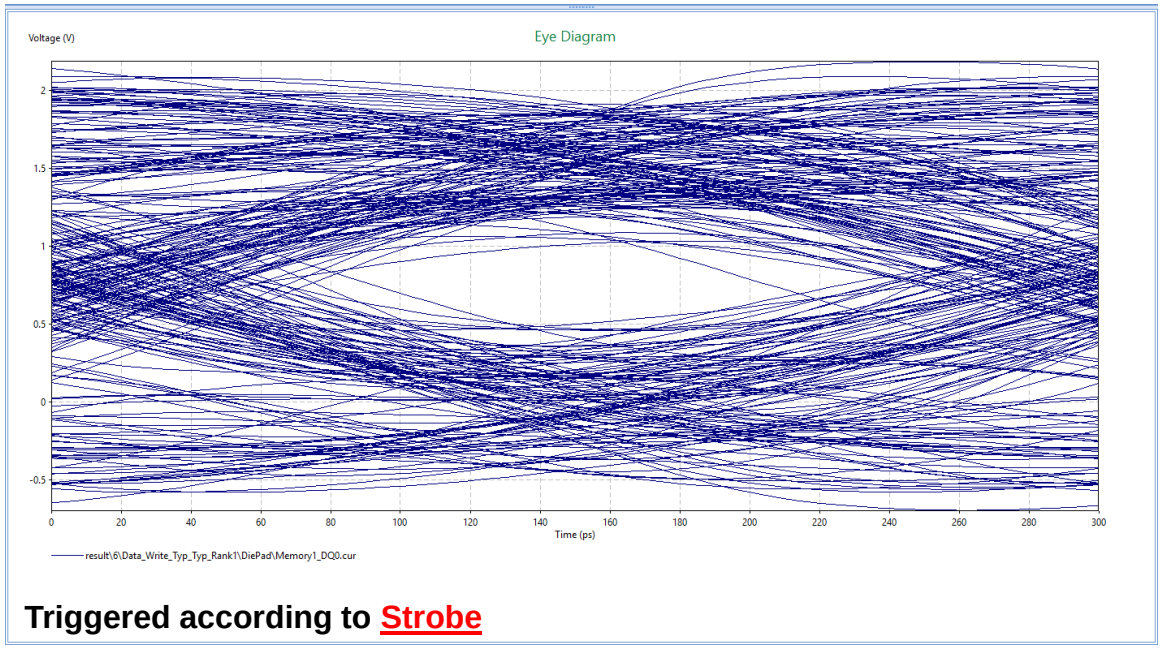
- Different timing reference – Different selection of “trigger” can result in different eye opening



Transmission Rate: 3.333Gbps



Triggered according to ideal UI

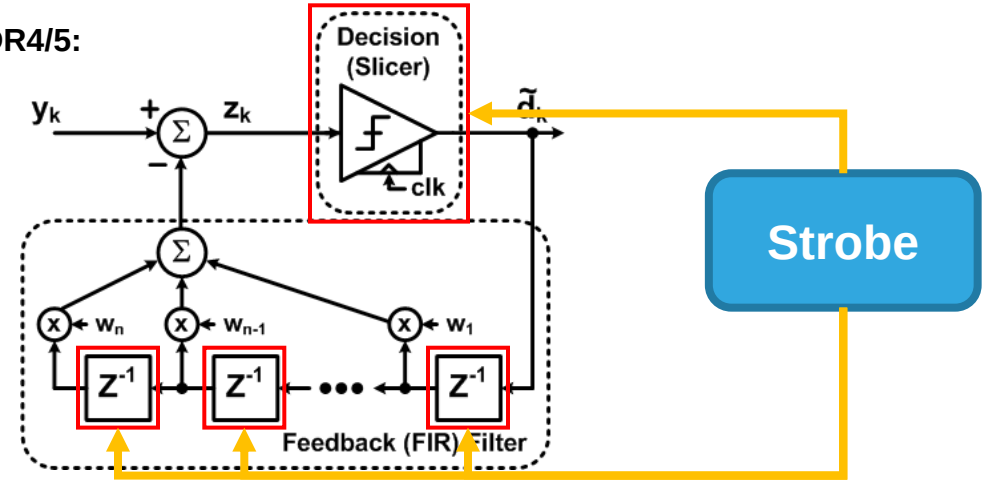


Triggered according to Strobe

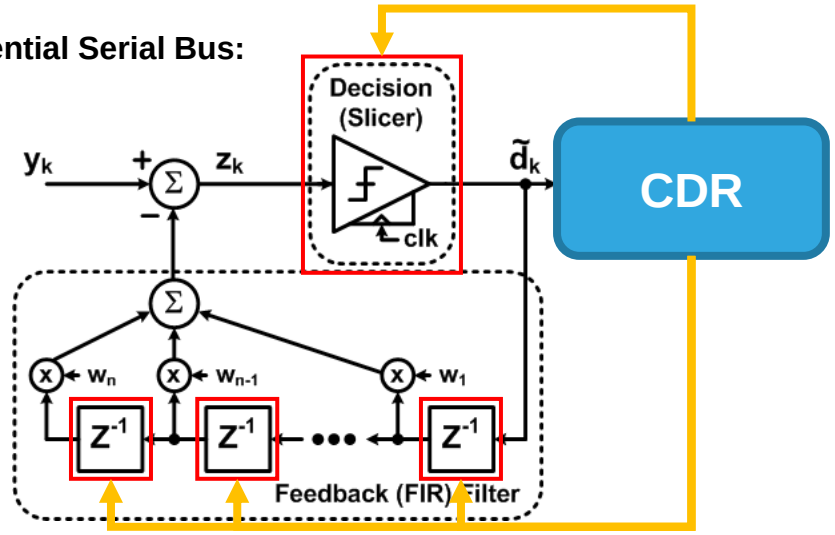
Overview(2)

- DFE's clock in memory bus will be supplied by Strobe rather than CDR, which can be seen in most differential serial bus

DDR4/5:



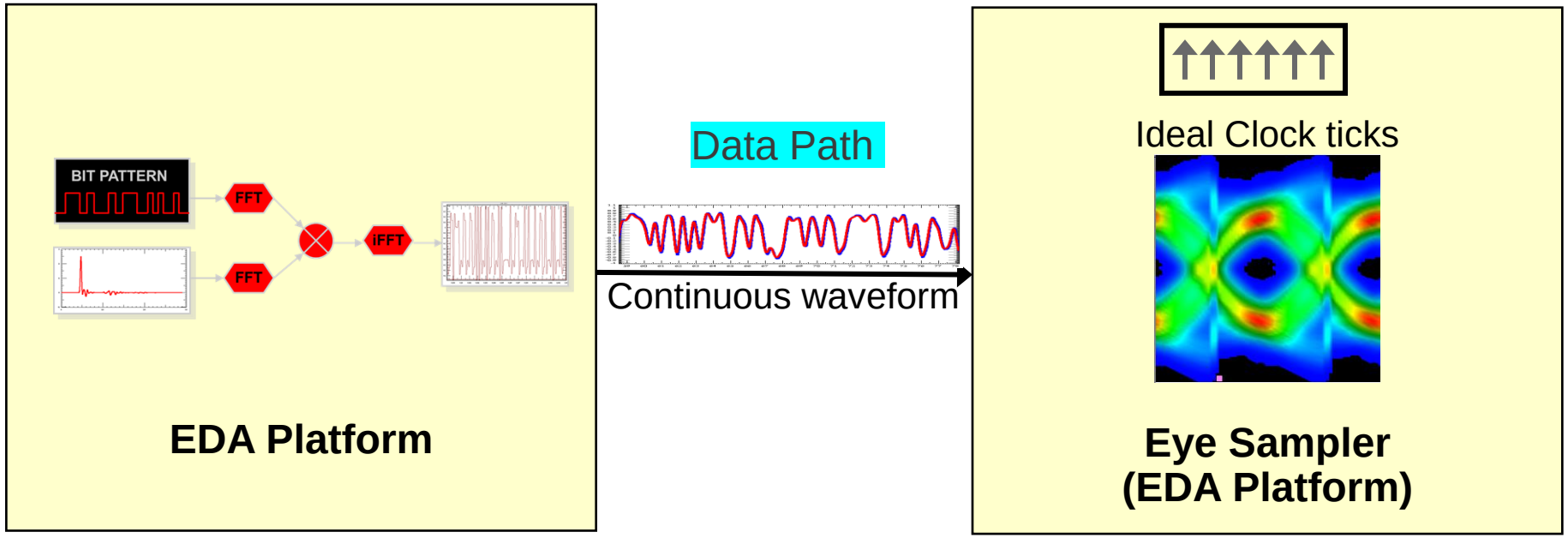
Differential Serial Bus:



- As channel simulation and IBIS-AMI modeling methods are adapted from serial link to DDR interface analysis, serial link CDR algorithms are often used for analysis
- What is the impact?

Current CDR-Based Method

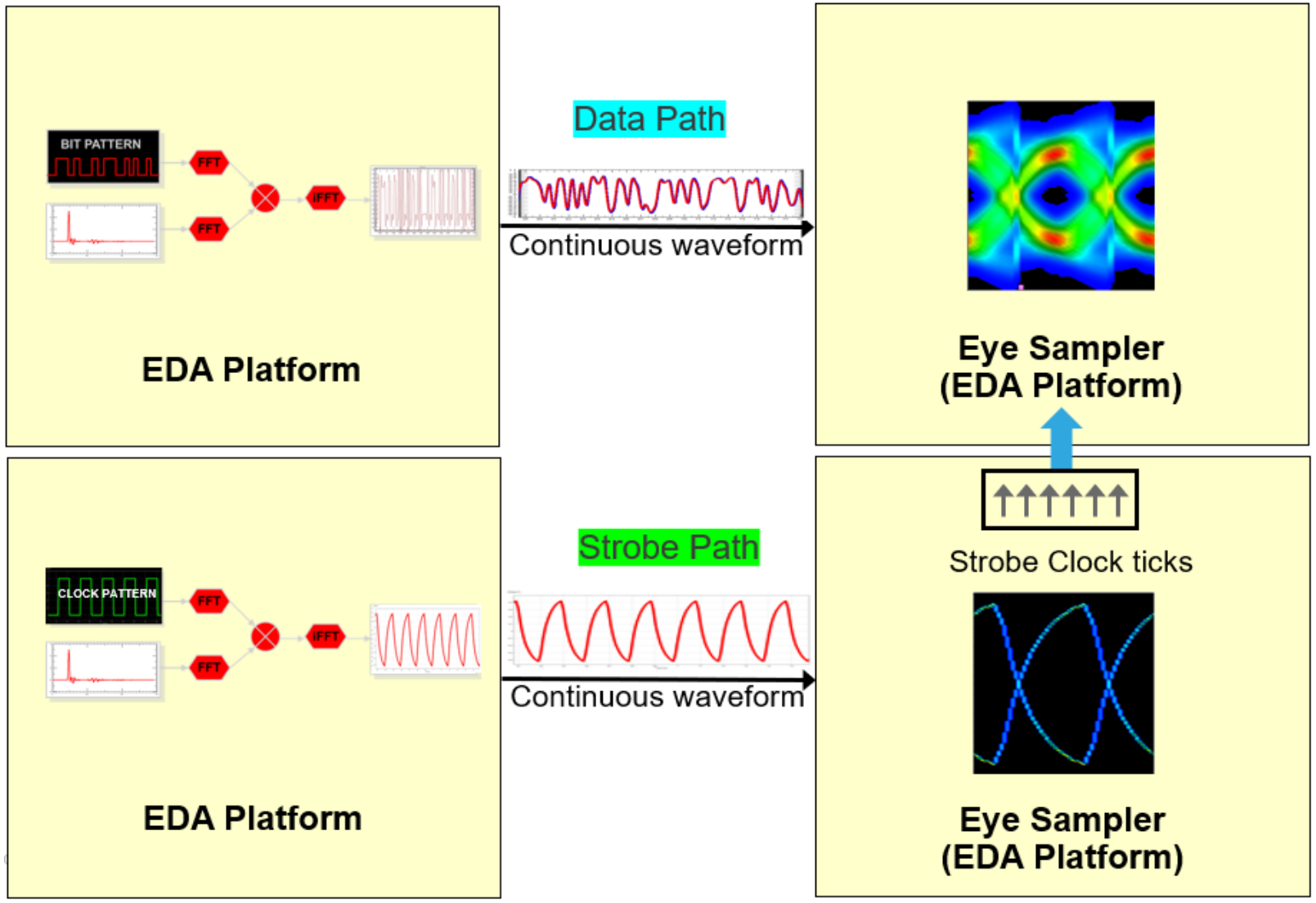
- Centers the eye for each individual signal
  - Ideal clock ticks are generated internally by the eye sampler
  - Clock ticks can also be generated by AMI models and sent to the eye sampler



- Some controllers have some individual bit de-skewing

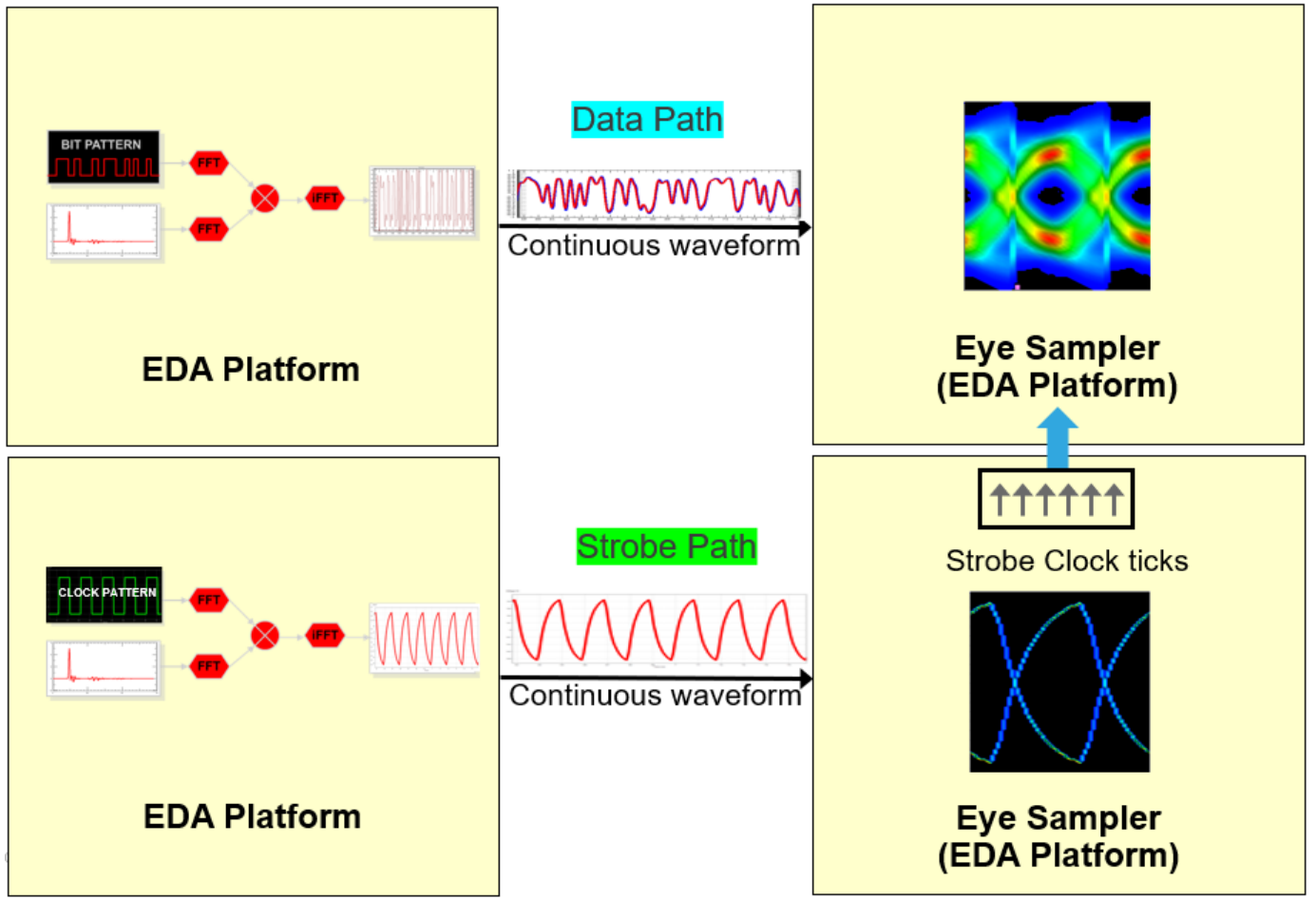
True Strobe Timing (TST)

- Clock ticks are collected from the strobe channel instead of the data channel
  - With real strobe, this is done for entire byte lane
- Strobe channel is only fed with 0101 data



Comparison of Results

- CDR vs. TST
- CDR vs. TST with jitter impairments



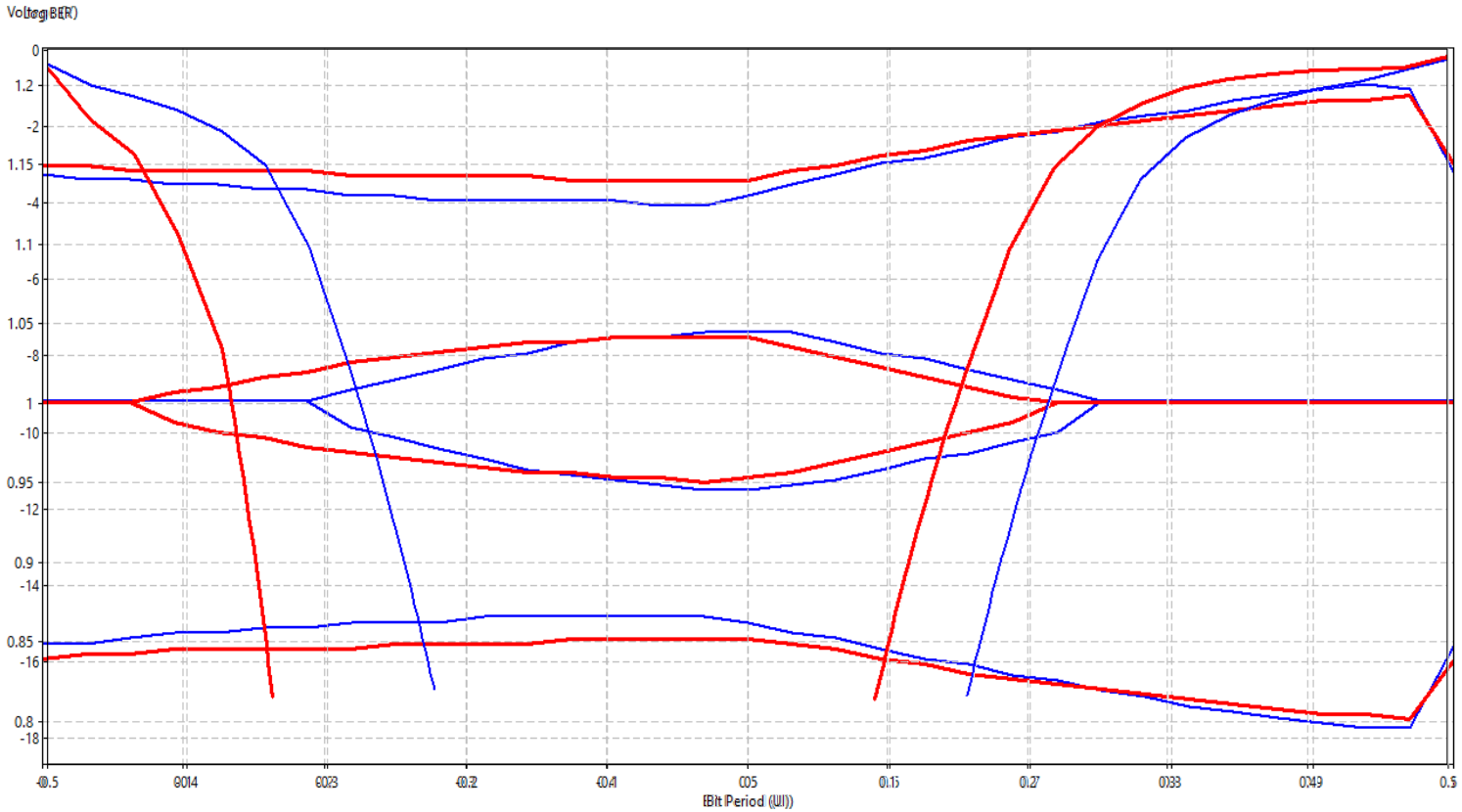
- Test configuration
  - 1 data line is used for simulations
  - 6 Gbps
  - Rx CTLE
  - Rx 4 tap DFE
  - In phase between strobe and data

cadence

CDR vs. TST

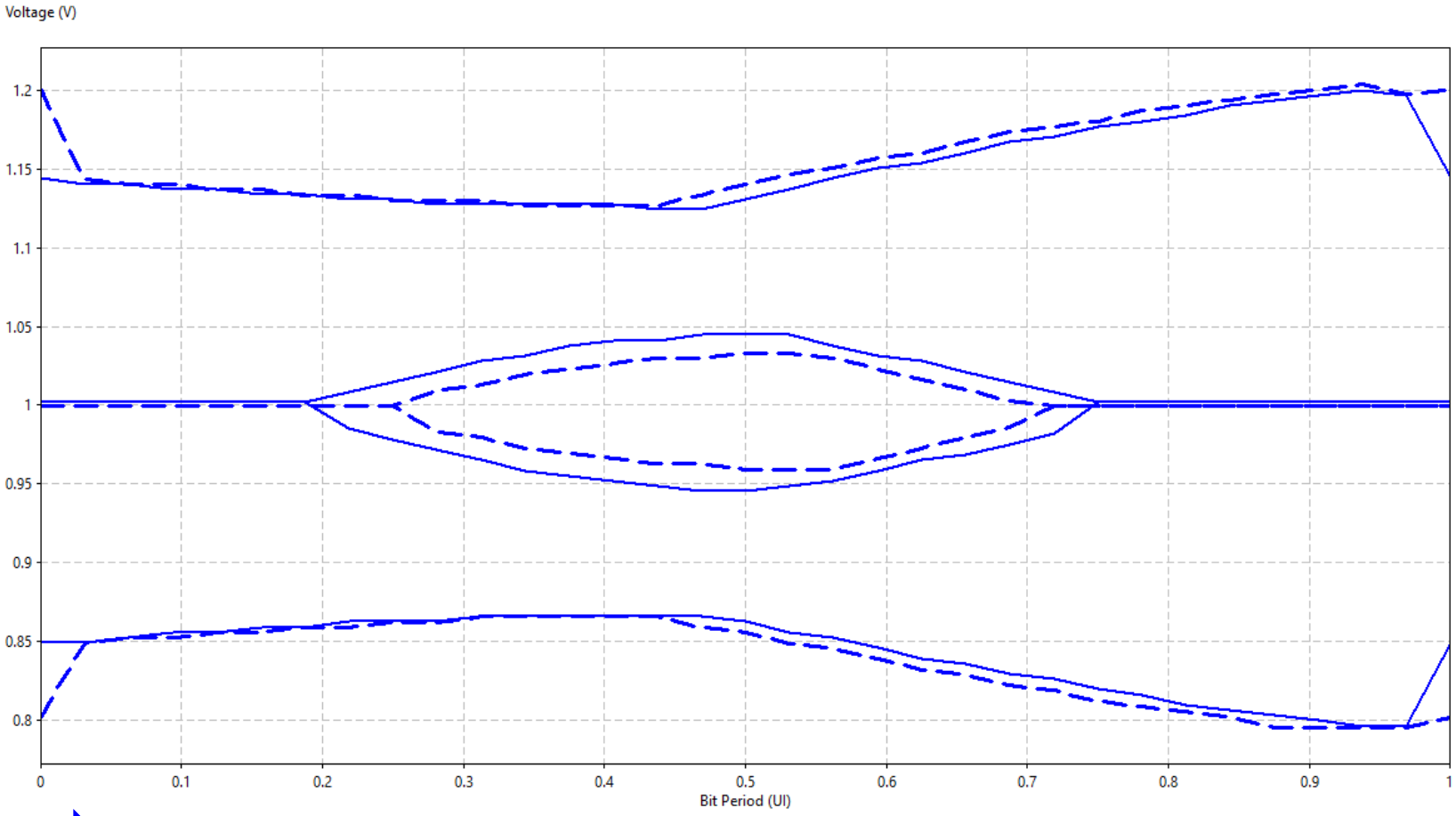
CDR

Strobe



cadence

CDR Results with Dj Applied at Tx

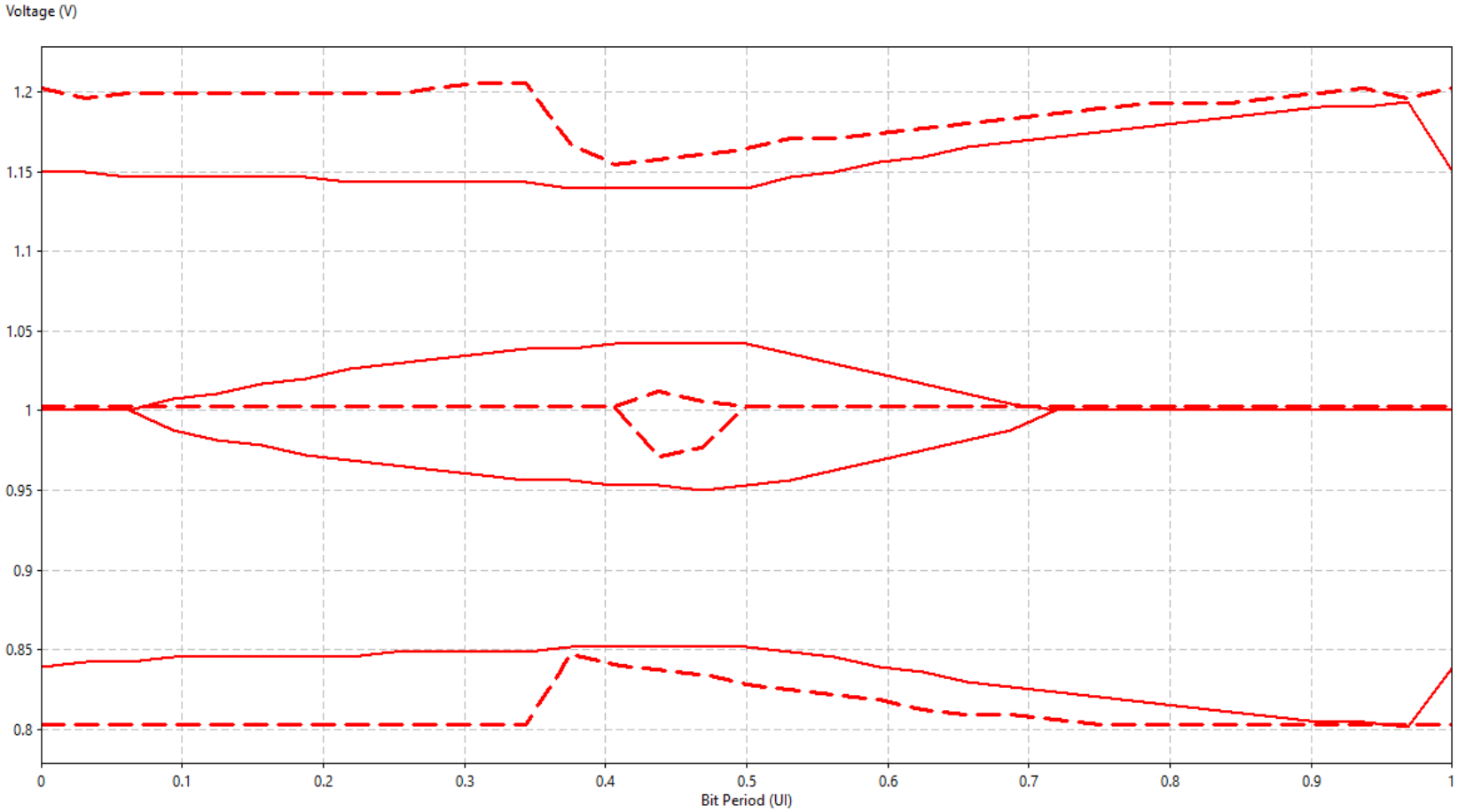


5%UI

15%UI

cadence

Strobe Results with Dj Applied at Tx



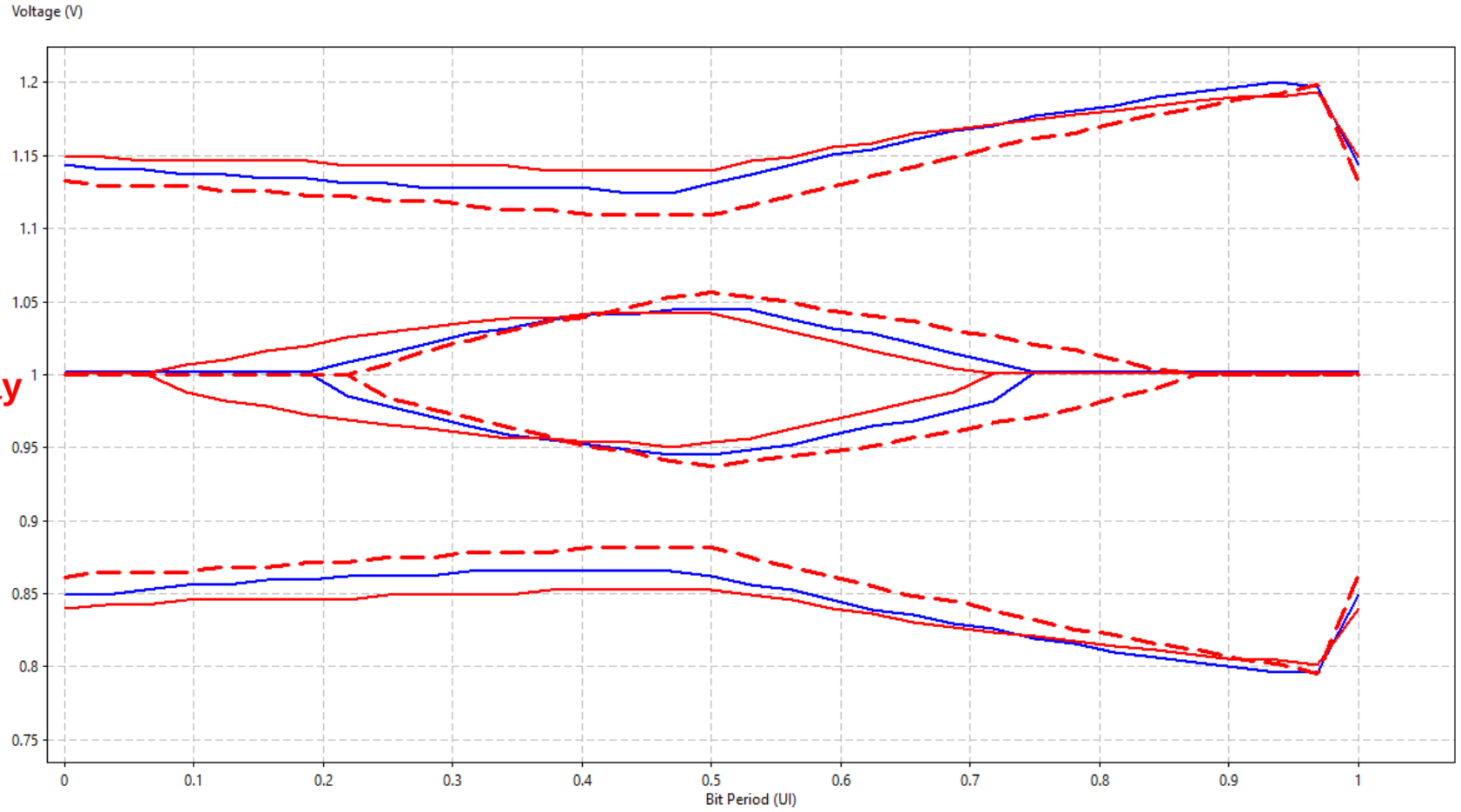
5%UI Dj on both strobe and data

15%UI Dj on both strobe and data

CDR vs. TST

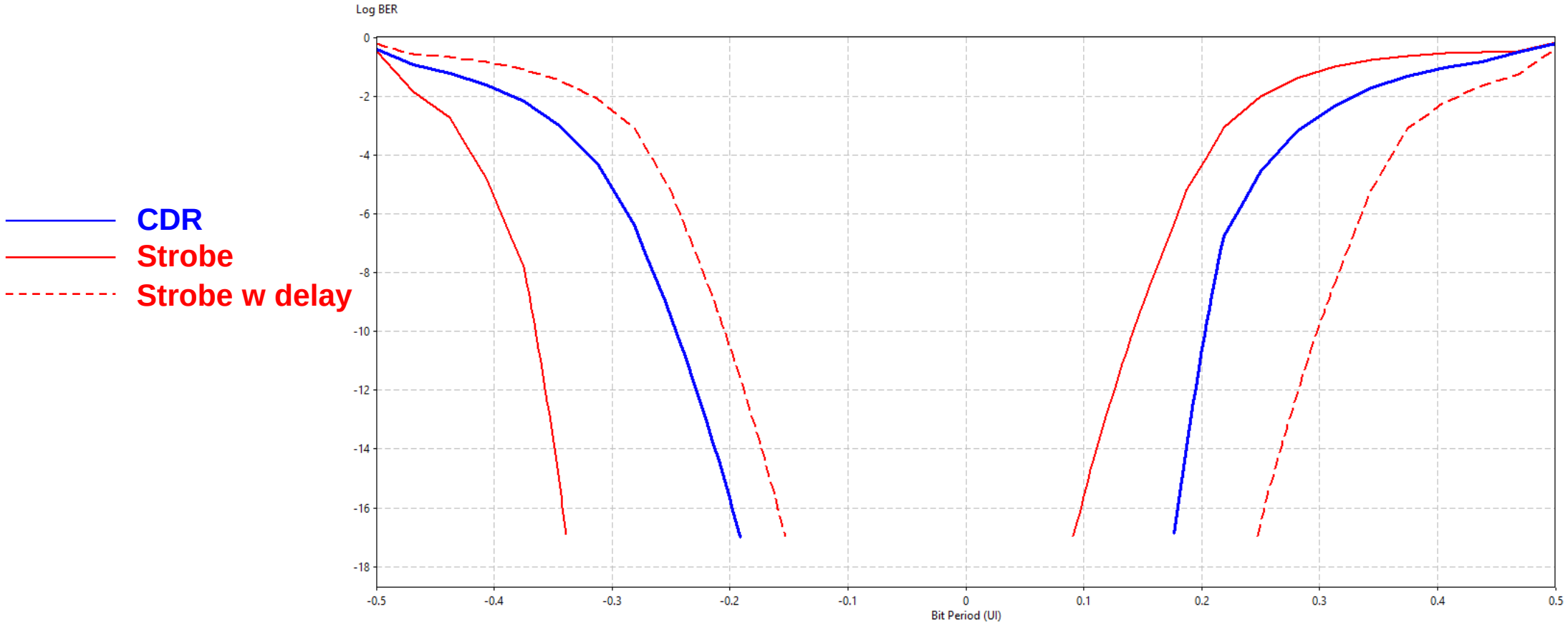
- After delaying by 0.2 UI

CDR
Strobe
Strobe w delay



CDR vs. TST

- After delaying by 0.2 UI



Summary

- Using default CDR instead of actual strobe to get clock risks will miss important impairments/jitter for parallel bus topology
- Analysis results show false optimism using CDR approach as compared to true strobe timing methodology
- Need to model delay accurately