

Modeling DDR Clock Forwarding With a New AMI_GetWave API

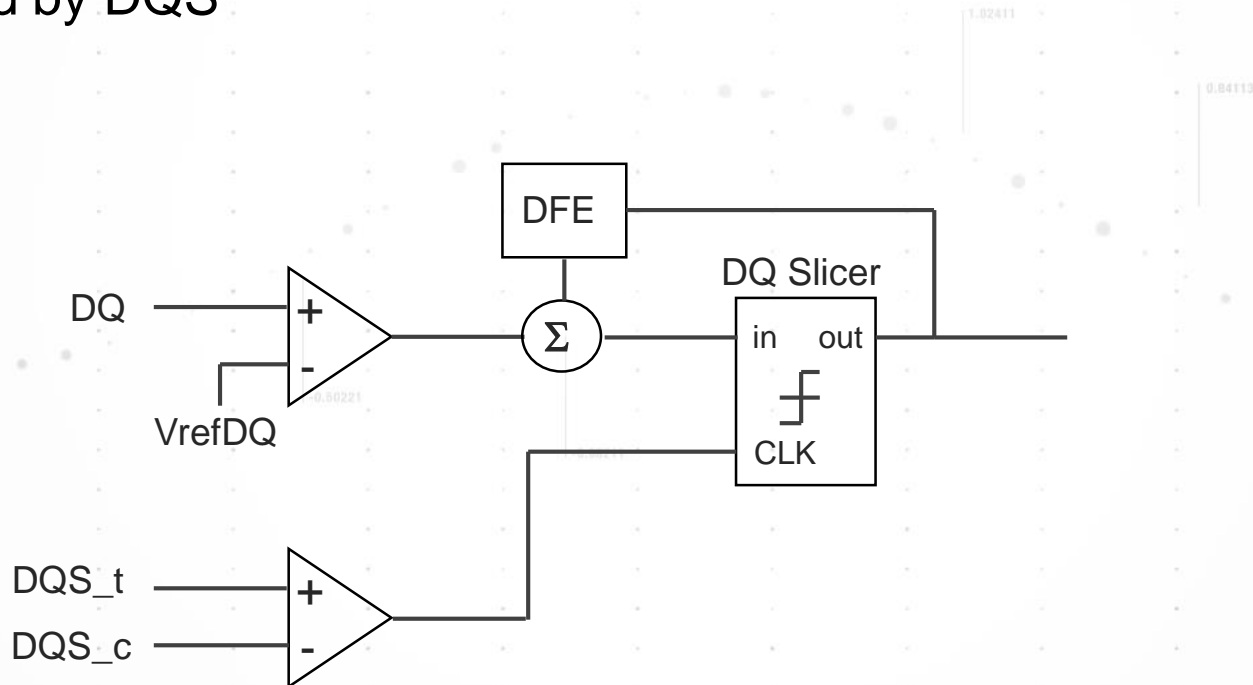
Fangyi Rao, Keysight Technologies

Wendem Beyene, Intel



DFE and Clock Forwarding Architectures in DDR5 Data Buffer

- DDR5 DQ Rx utilizes DFE to mitigate ISI
- DQ Rx does not have embedded CDR as in SerDes.
- DQ DFE is clocked by DQS



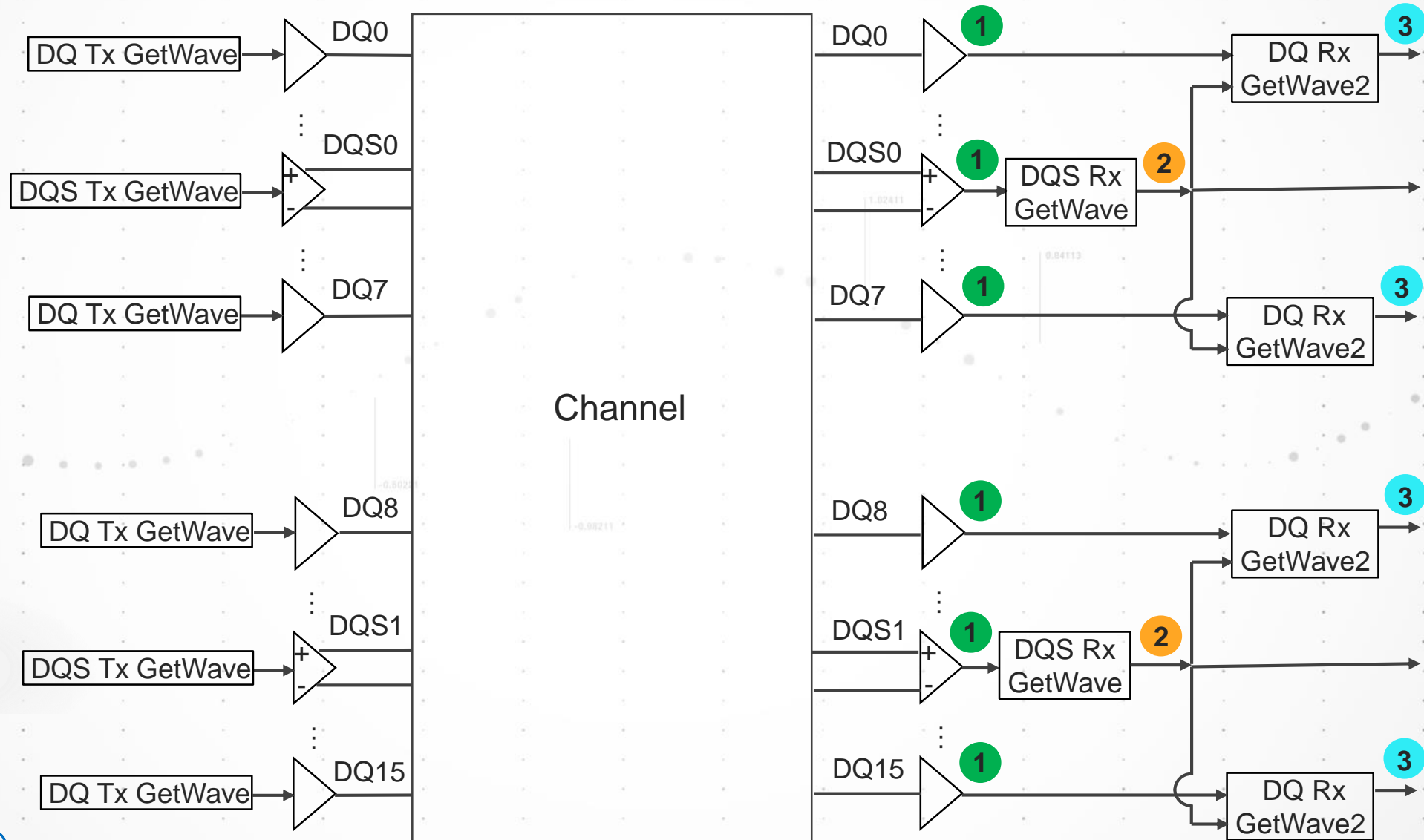
Proposal of A New GetWave API to Model Clock Forwarding

```
long AMI_GetWave2(double *wave,  
                 double *wave_ref,  
                 long wave_size,  
                 double *clock_times,  
                 char **AMI_parameters_out,  
                 void *AMI_memory);
```

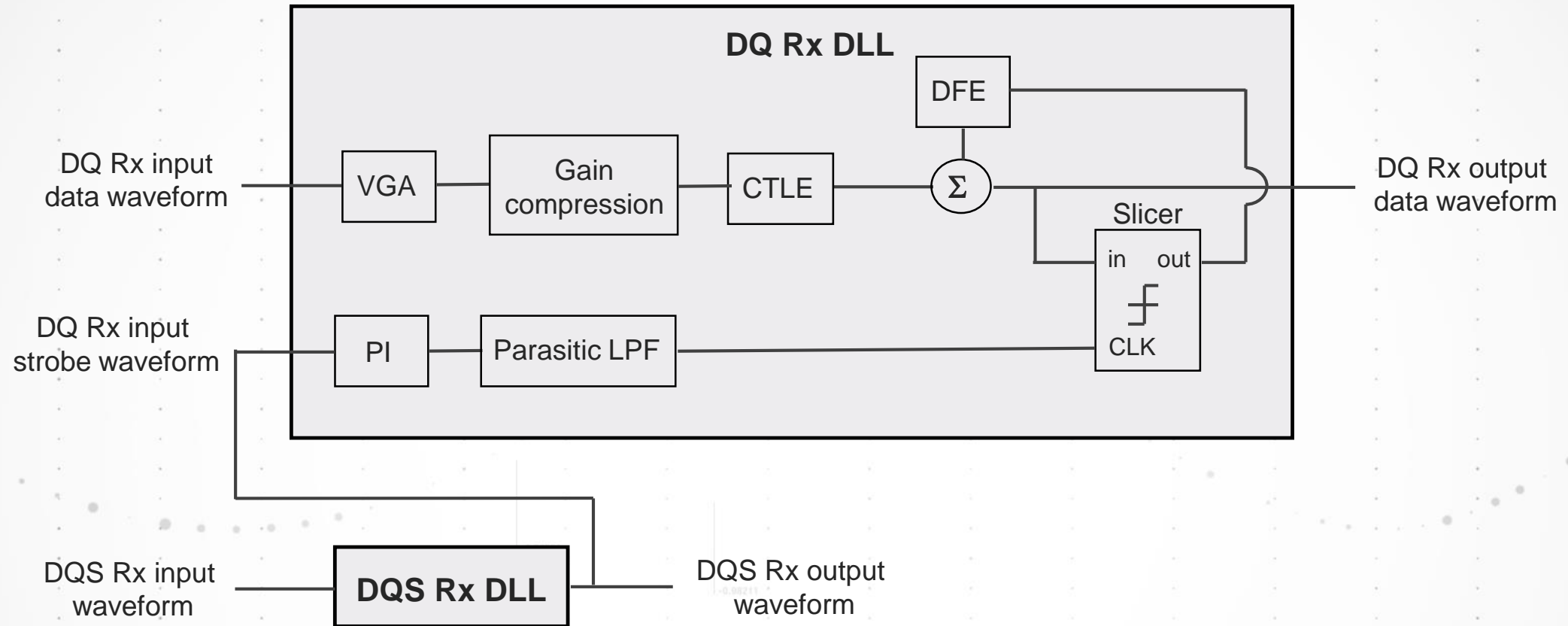
- The new GetWave2 function is implemented by DQ Rx model
- Two input waveforms and one output waveform
- wave: input and output data waveforms
- wave_ref: input strobe waveform, which is the output waveform of DQS Rx GetWave
- Sizes of wave and wave_ref are the same
- Rx model can choose to process wave_ref
- clock_times: output clock times generated by Rx model based on wave_ref

Time Domain Simulation Flow

- Step 1:** compute analog channel output according to current flow (crosstalk taken into account)
- Step 2:** compute output of all **DQS** Rx DLLs
- Step 3:** compute output of all **DQ** Rx DLLs



Example of Controller DQ Rx Model

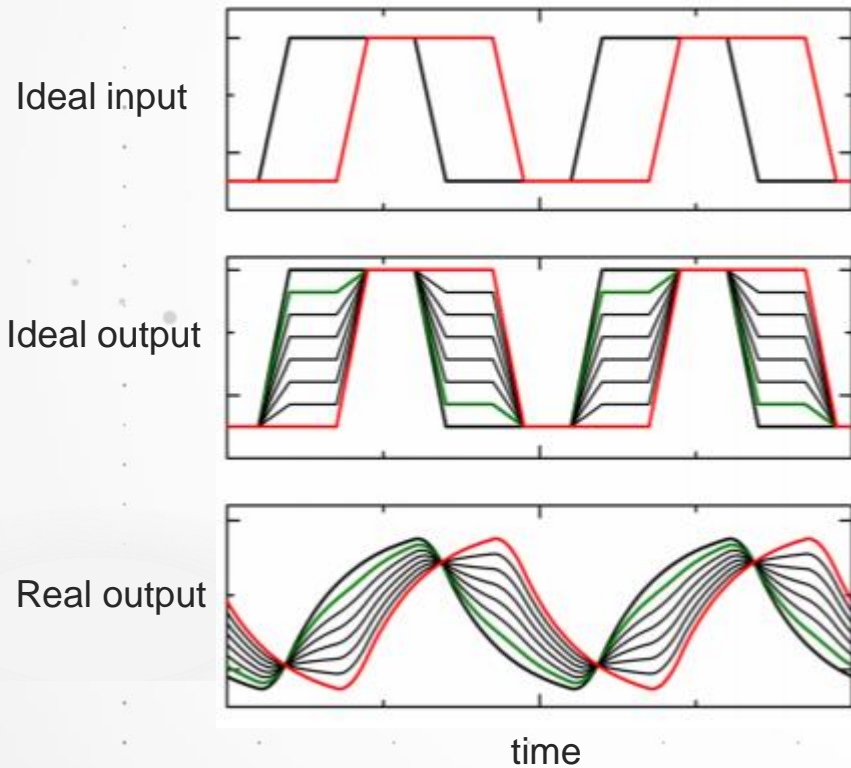


- Input strobe waveform is processed by a phase interpolator (PI)
- PI is dynamically trained by Rx DLL to adjust data-strobe skew for optimal DFE clocking
- Adaptive DFE
- Both PI training and DFE adaptation stop after Ignore_Bits to emulate system start-up training
- Captures DQ slicer sensitivity to strobe slew rate

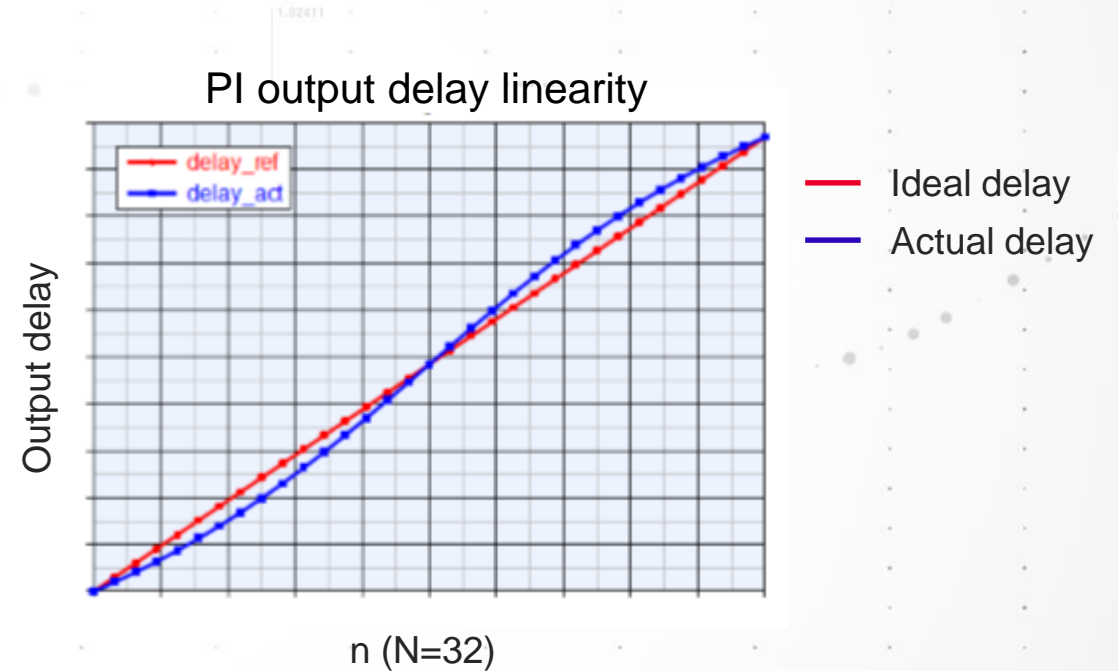
Phase Interpolator Review

$$v_{out}(t) = \frac{n}{N} v_{in}(t - \tau_1) + \frac{N - n}{N} v_{in}(t - \tau_2), \quad n = 0, 1, \dots, N$$

PI input/output waveform

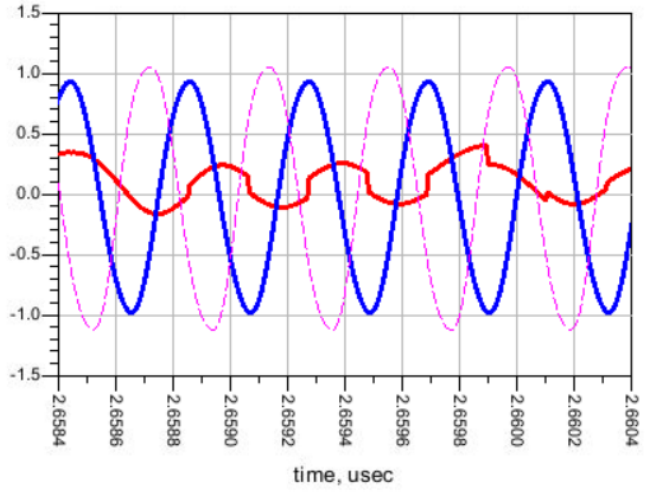
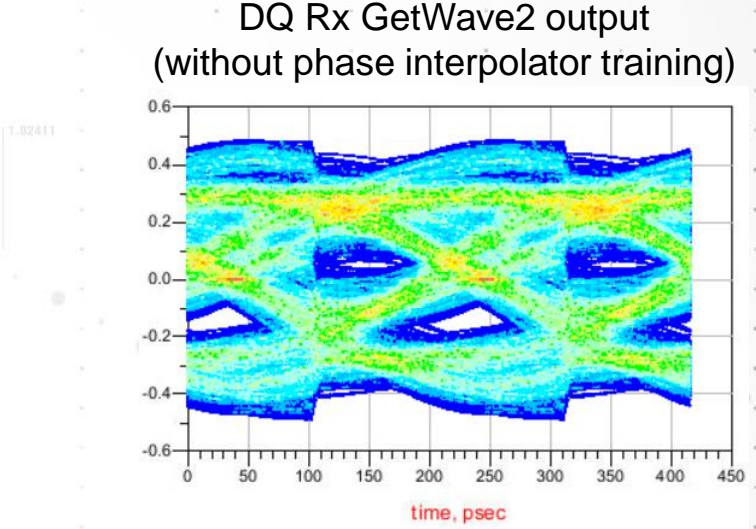
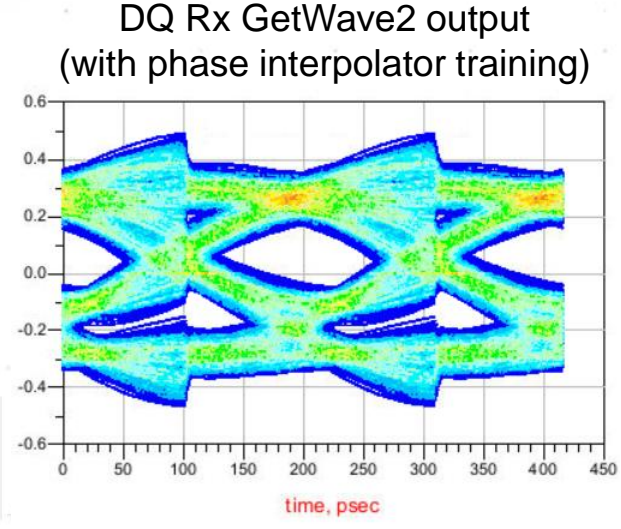
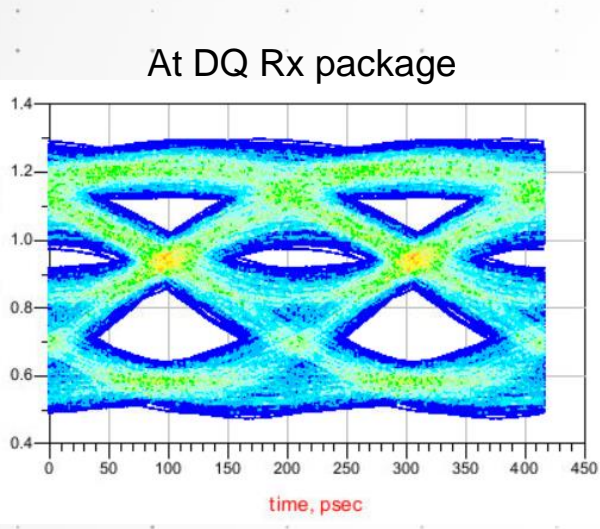


PI output delay linearity



Phase Interpolator Training in Controller DQ Rx Model

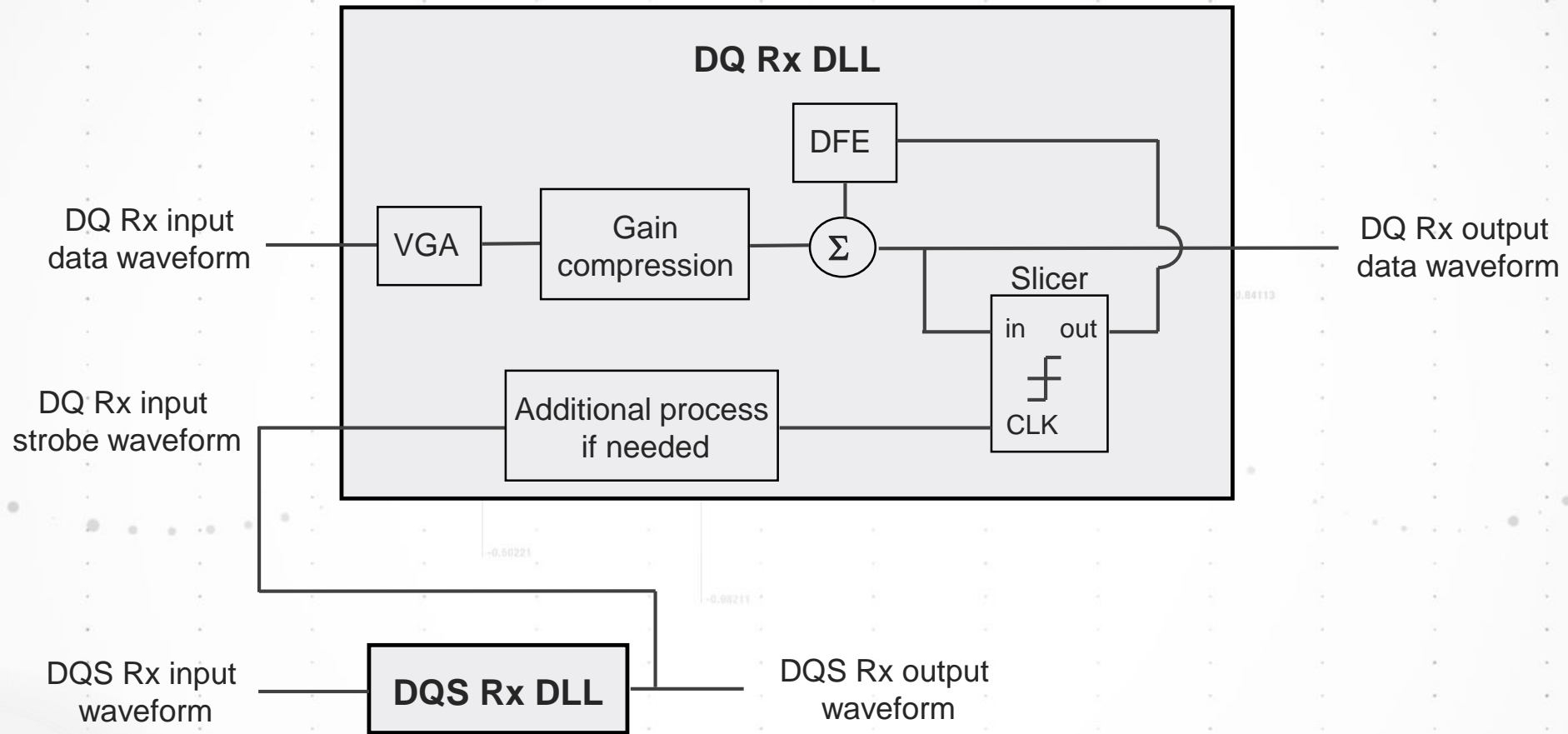
Controller DQ Rx model can internally train the phase interpolator to adjust data-strobe skew for optimal DFE clocking



- DQ Rx GetWave2 output
- - - DQ Rx GetWave2 strobe input
- Strobe after phase interpolator

Data and post-phase interpolator strobe are center-aligned

Example of DRAM DQ Rx Model



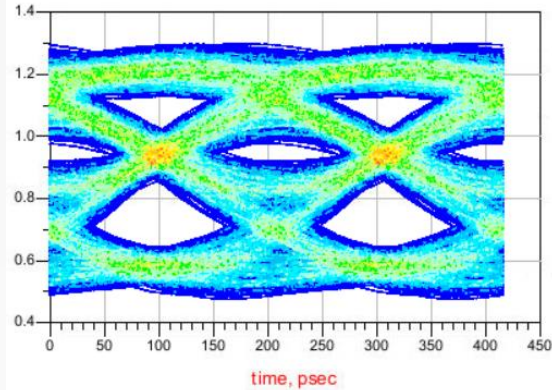
- DQ Rx DLL can process the input strobe waveform if needed. For example, DQ Rx can add internal delay to strobe.
- Captures DQ slicer sensitivity to strobe slew rate
- DQ-DQS skew is expected to be optimized on the controller Tx side by write leveling training

Jitter Tracking and Unmatched IO Rx

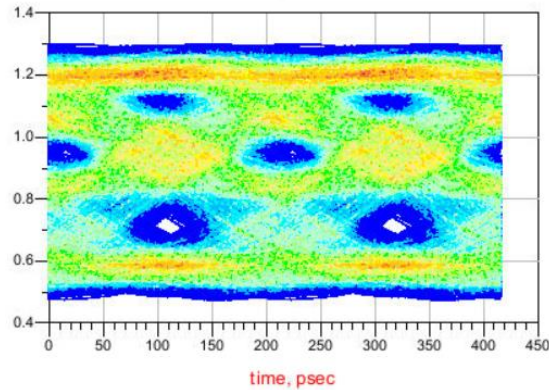
- Correlated jitters in DQ and DQS can be tracked in DQ Rx by clock forwarding
- DDR4 (and previous) DRAMs pad the DQ path to match DQ Rx to DQS
- DDR5 supports unmatched DQ and DQS Rx on both DRAM and controller sides
- Unmatched Rx reduces DQ-DQS jitter correlation and adversely impacts DQ Rx jitter tracking and DFE performance

Jitter Tracking and Unmatched IO Rx (cont'd)

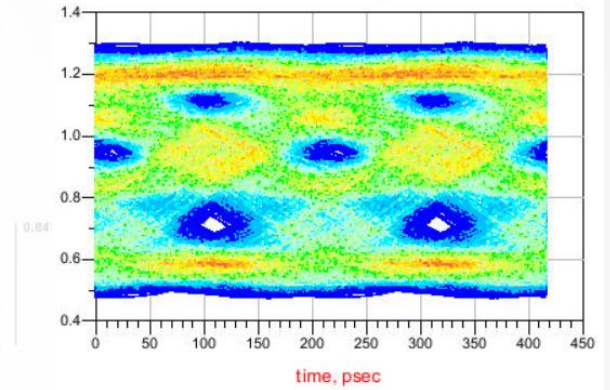
DQ Rx package without DQ & DQS Tx SJ



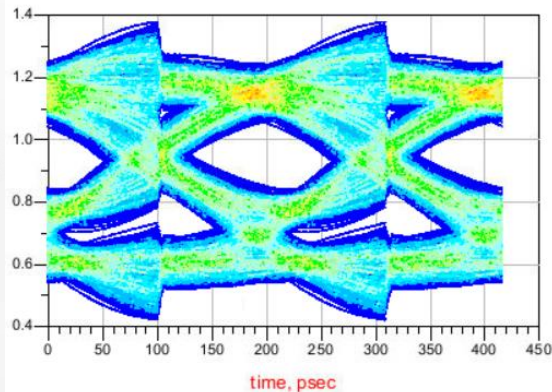
DQ Rx package with DQ & DQS Tx SJ
(0 DQS-to-DQ delay)



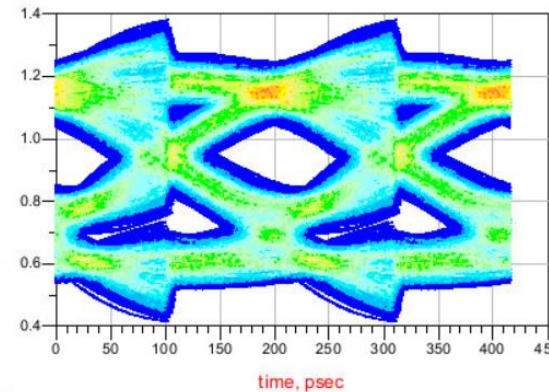
DQ Rx package with DQ & DQS Tx SJ
(5UI DQS-to-DQ delay)



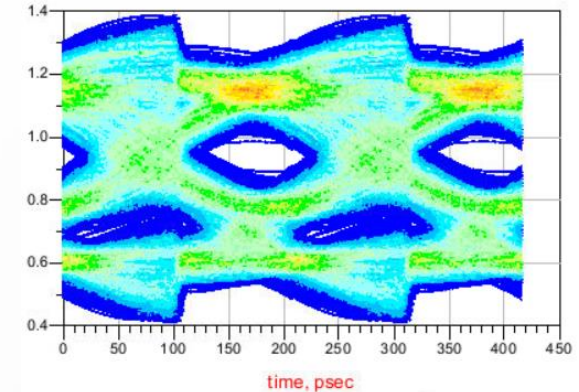
DQ Rx output without DQ & DQS Tx SJ



DQ Rx output with DQ & DQS Tx SJ
(0 DQS-to-DQ delay)



DQ Rx output with DQ & DQS Tx SJ
(5UI DQS-to-DQ delay)



Summary

- Data buffer DFE is clocked by strobe
- We propose a new GetWave2 function with two input waveforms for data and strobe to model clock forwarding in DQ Rx
- DQ Rx model can use the strobe waveform to model
 - Forwarded clock of DQ slicer and DFE
 - DQ slicer sensitivity to strobe slew rate
 - DQ phase interpolator
 - DQ phase interpolator parasitic LPF
 - DQ internal strobe delay
 - ...
- Controller DQ Rx model can internally adjust data-strobe skew to optimize DFE clocking
- Jitter tracking in DFE and unmatched IO Rx are captured by the new GetWave2 function