UBM

Optimal DDR4 System with Data Bus Inversion

Hing Yan (Thomas) To, (Xilinx Inc.)

Changyi Su (Xilinx Inc.), Juan Wang (Xilinx Inc.)
Dmitry Klokotov (Xilinx Inc.), Lizhi Zhu (Xilinx Inc.), John Schmitz (Xilinx Inc.)
Penglin Niu (Xilinx Inc.), Yong Wang (Xilinx Inc.)

SPEAKER

Hing Yan (Thomas) To

Technical Director, Xilinx Inc.

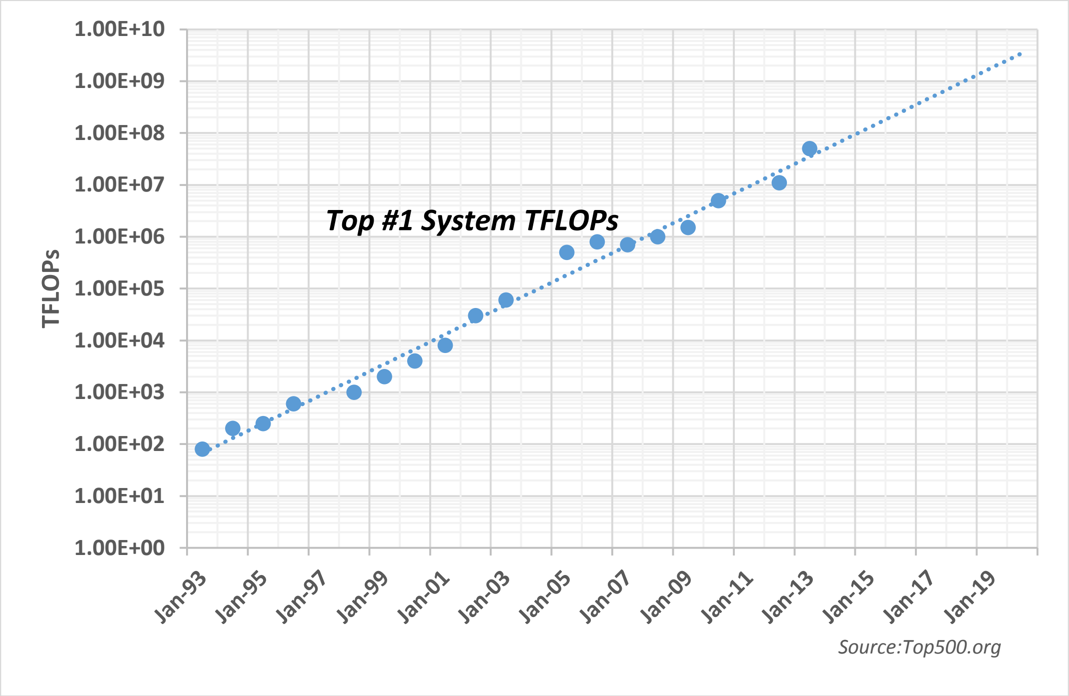
tto@xilinx.com

Thomas is a Technical Director in System Memory Signal Integrity & Device Power Group at Xilinx, Inc. Prior to joining Xilinx, Thomas was with NVIDIA Advanced Technology Group focused on high speed (32GTs) circuits & system channel designs and supported different test chips for different advanced process nodes such as 20nm SOC & 16nm FINFET process. Before NVIDIA, Thomas worked for Intel for more than 16 years covered and led many different types of system memory IO development such as Sandy Bridge Server DDR IO and covered many different system memory technology ranging from DDR1 to DDR4. Thomas received his PhD degree in Electrical Engineering from the Ohio State University in 1995 & he has over 37 patents in the fields of mixed signal IO circuits and system memory configurations as well as high speed clocking for high speed memory designs.

Outline

- High Performance Computing Performance Requirement Trend
- Typical Power Distribution in Computing System Example
- System Memory Power Improvement Approach
  - Technology Process Node Scaling Trend
  - IO Voltage Scaling Trend
  - DDR4 IO signaling
- Data Bus Inversion (DBI) in DDR4 Interface
  - DQ bus data Functional View with DBI enabled
  - DDR4 System Power Improvement Example
  - DDR4 IO Interface Training & Calibration with DBI
- Power Noise Improvement with DBI
- Experimental Data Margin Validation and Results
- Summary & Conclusions

Computation Requirement Trend

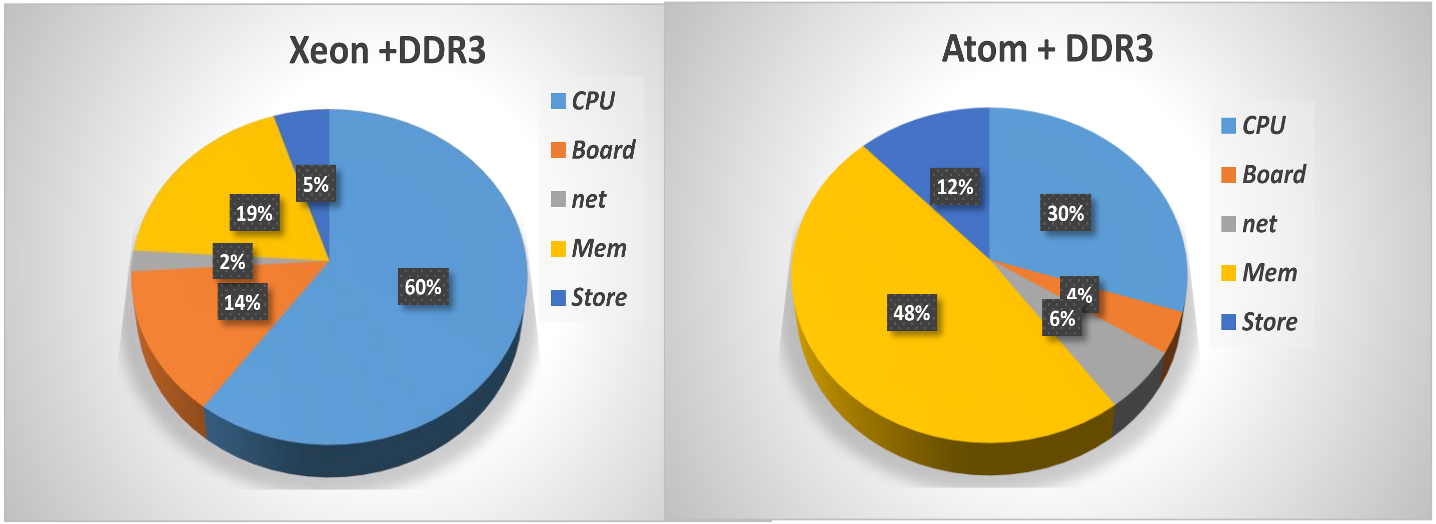


Source:Top500.org

→ *Computing Performance Requirement increases exponentially.*

→ *Expected to maintain similar or lower the Power Envelope.*

Typical Power Distribution Comparison



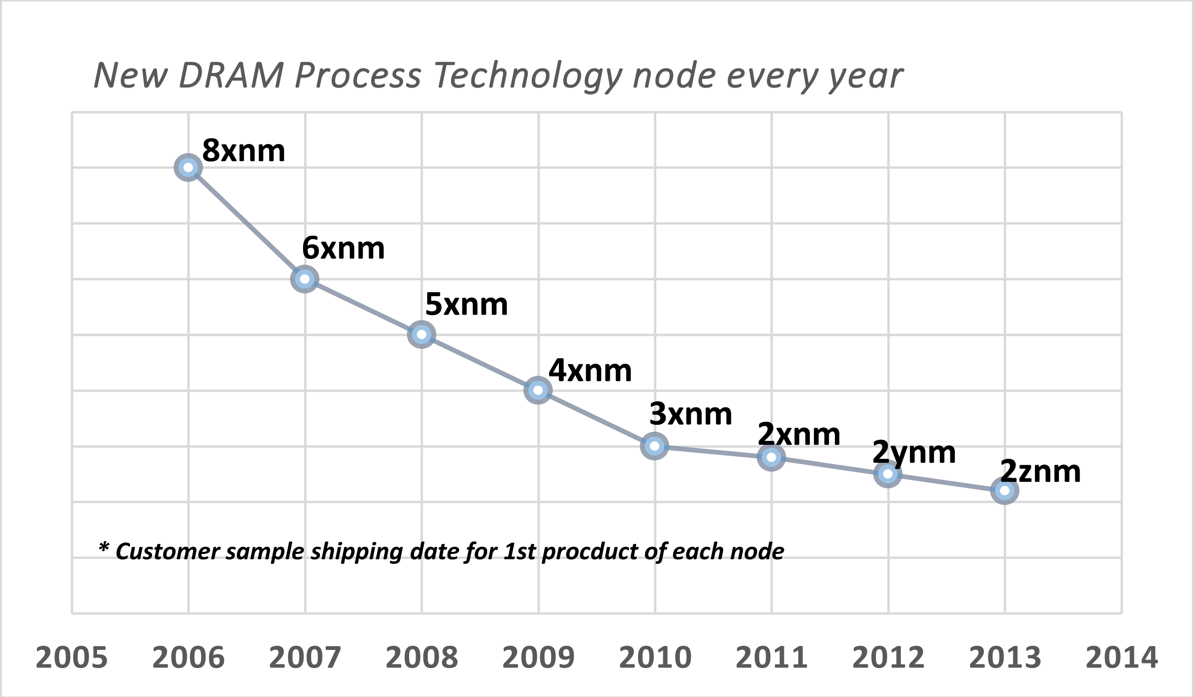
→ *Traditionally CPU has been the dominated component.*

→ *System Memory becomes a factor as CPU power improves relatively.*

System Memory Power Improvement Approach

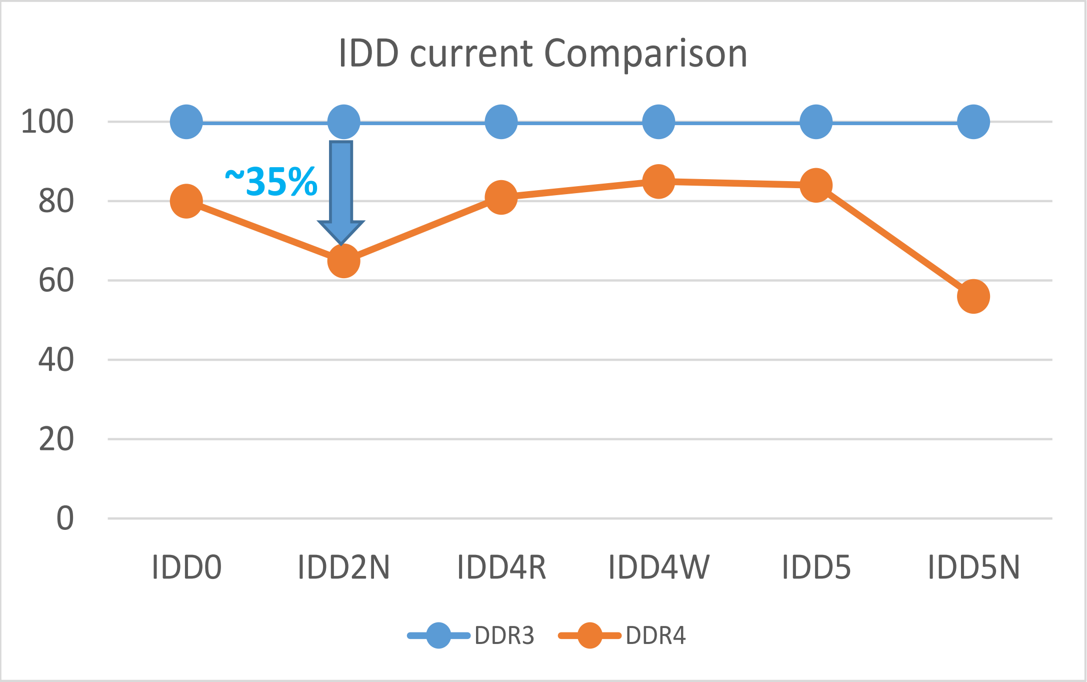
- Technology Process Node Scaling Trends
  - Improving Process Technology improves speed, power and memory density.
- IO Voltage Scaling Trends
  - Scaling down the IO voltage improves IO power.
- IO signaling Improvements
  - IO Signaling can improve IO power

DRAM Process Technology Trend



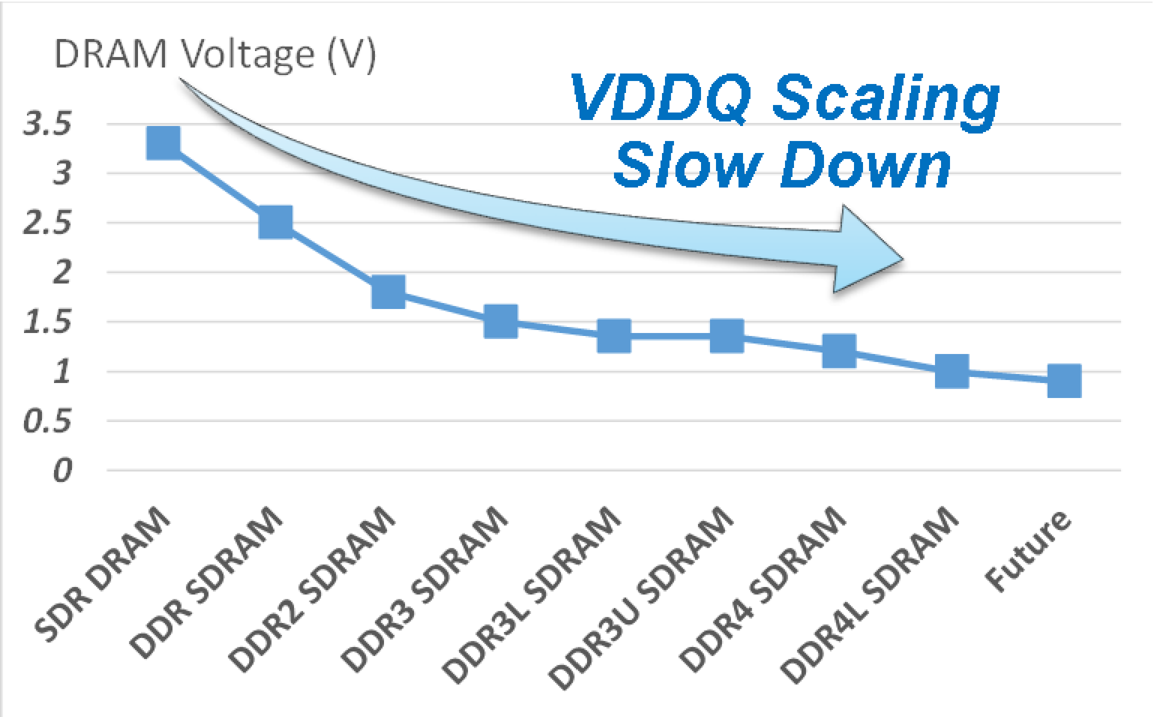
→ *DRAM introduced with new Process Technology Node every year .*

DRAM Power Improvement between DDR3 and DDR4



→ *DDR4 device improves power from DDR3 device*

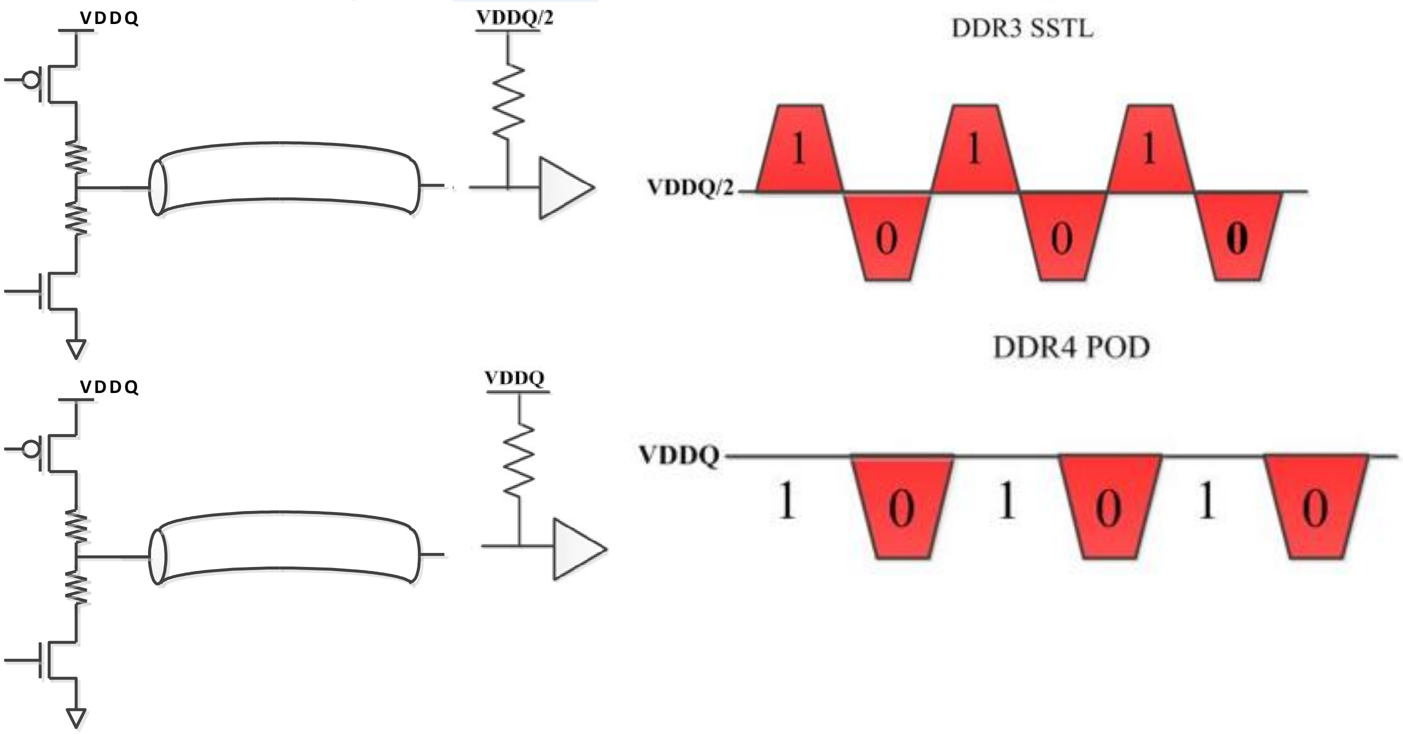
DRAM IO Voltage Scaling Trend



→ *DDR IO Voltage has been scaling down from generation to generation.*

→ *Scaling rate is slowing down.*

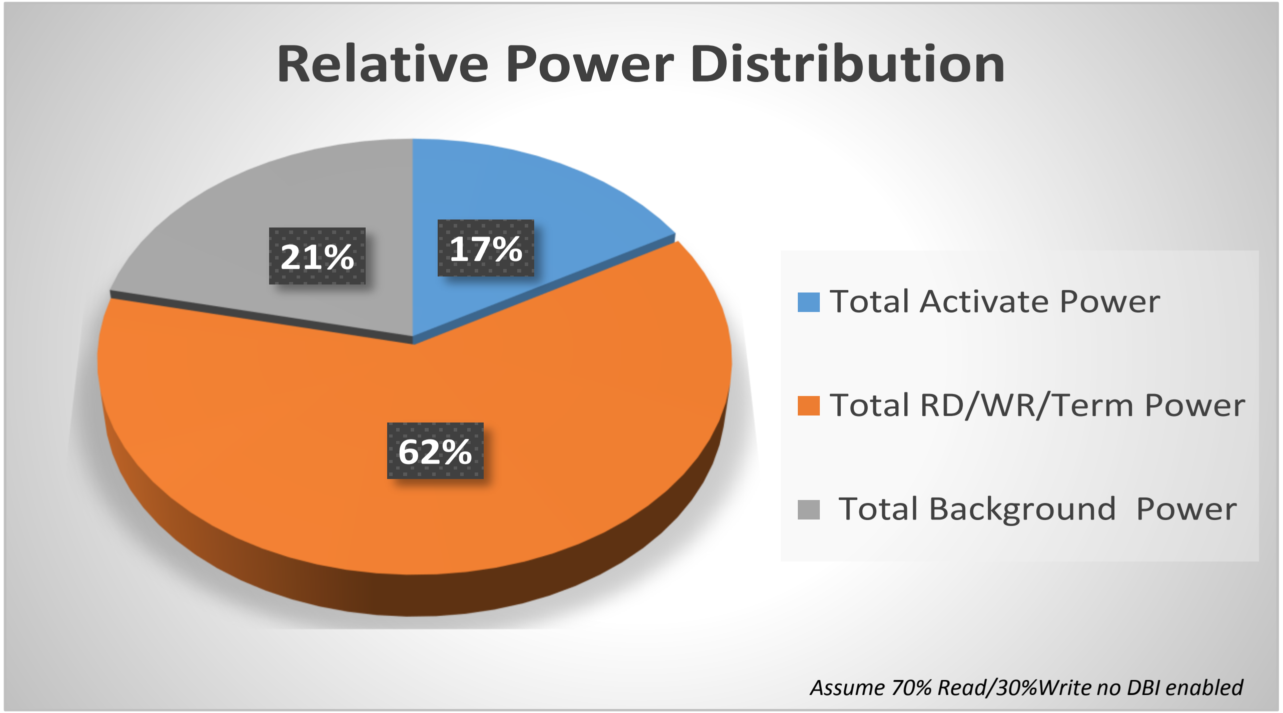
Change of IO Standard



🟥 DC Power Consumption

→ *Only Logic Low in DDR4 dissipates DC power.*

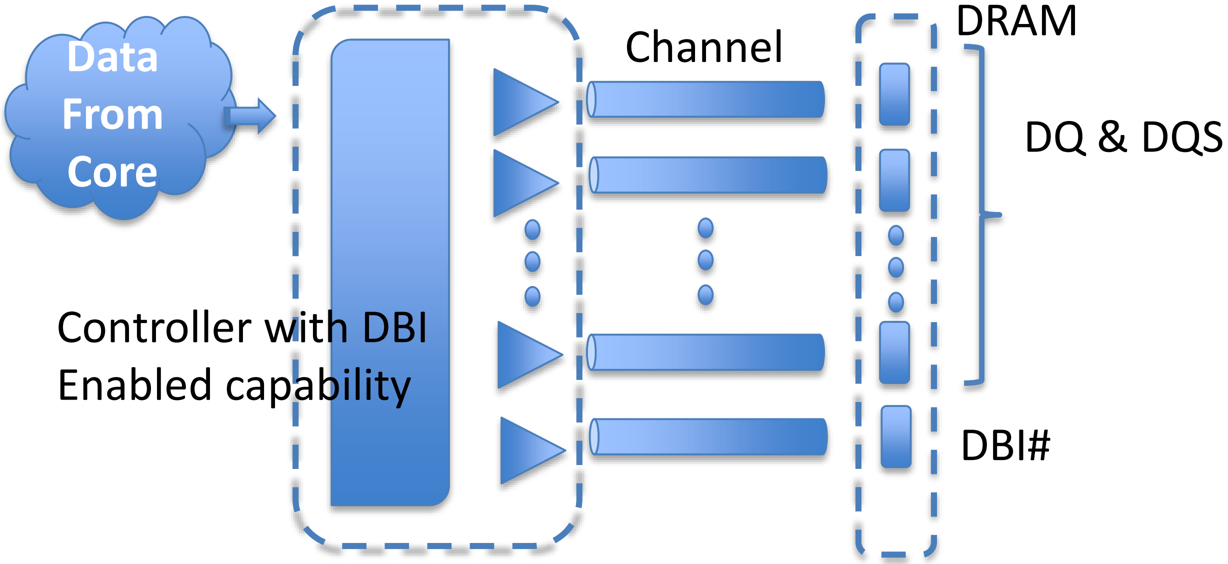
DDR4 Per Unit Power Distribution Comparison



→ *Even with Power Reduction w.r.t. DDR3, RD/WD/Term Power still a large portion.*

→ *DDR4 can enable DBI to further improve IO power opportunistically.*

DBI Functional View


Data From Core

Controller with DBI Enabled capability

Channel

DRAM

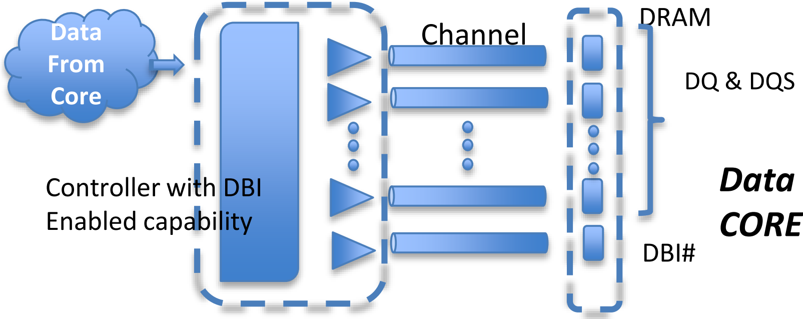
DQ & DQS

DBI#

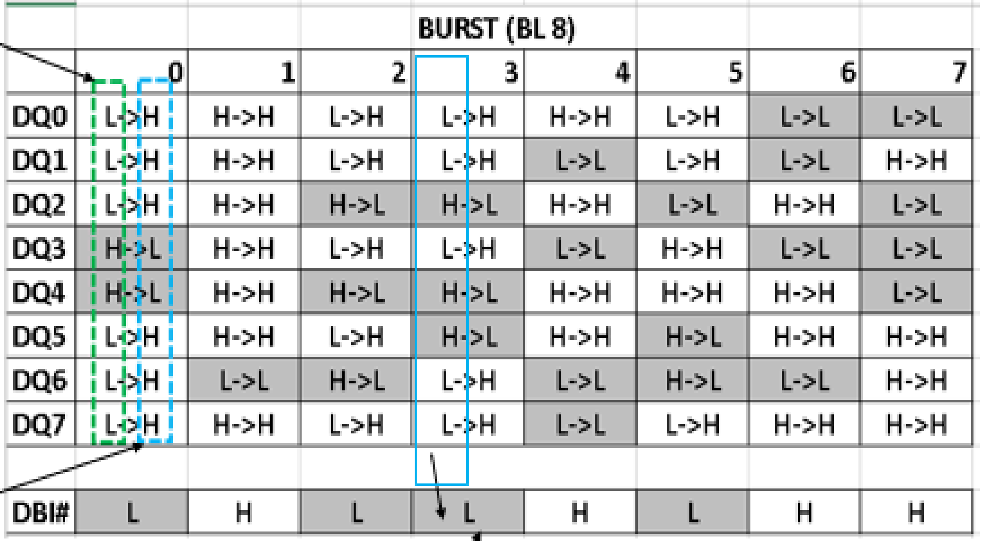
$$\boldsymbol{DBI\#[k] = func(DQ(7:0)[k])} \qquad \forall \quad \boldsymbol{k} \in \mathbb{N}_0$$

$$func(\eta) = sum_{logic_{low}}(\eta) > 4$$

DBI Functional Burst Length View



Data From CORE

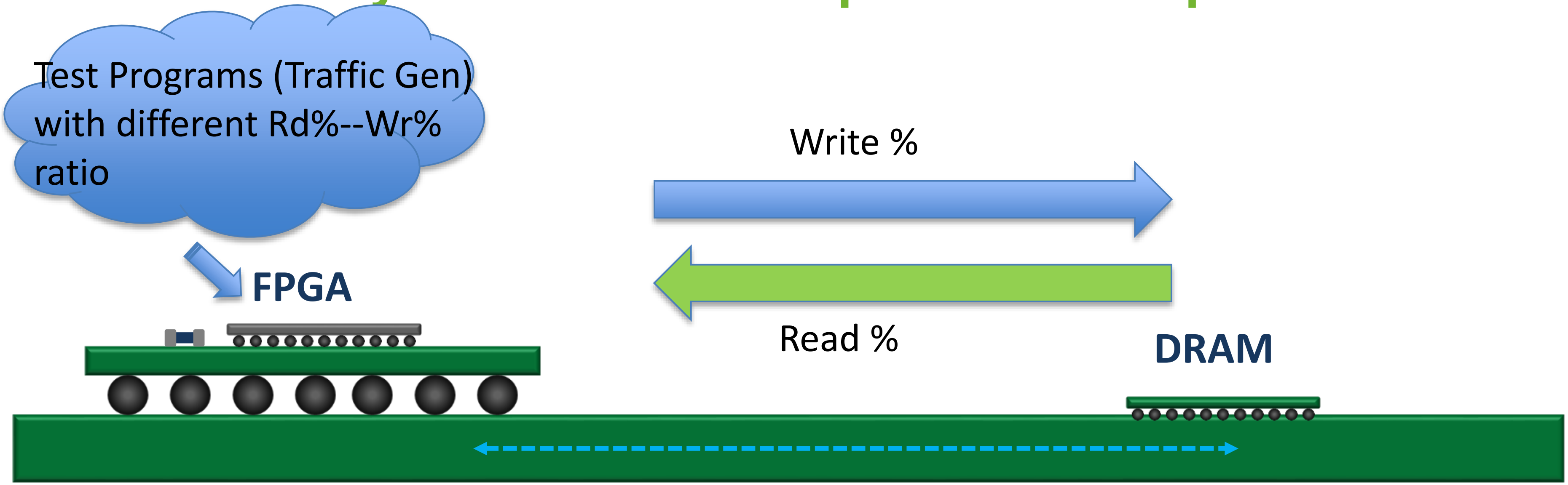


	BURST (BL 8)							
	0	1	2	3	4	5	6	7
DQ0	L->H	H->H	L->H	L->H	H->H	L->H	L->L	L->L
DQ1	L->H	H->H	L->H	L->H	L->L	L->H	L->L	H->H
DQ2	L->H	H->H	H->L	H->L	H->H	L->L	H->H	L->L
DQ3	H->L	H->H	L->H	L->H	L->L	H->H	L->L	L->L
DQ4	H->L	H->H	H->L	H->L	H->H	H->H	H->H	L->L
DQ5	L->H	H->H	L->H	H->L	H->H	H->L	H->H	H->H
DQ6	L->H	L->L	H->L	L->H	L->L	H->L	L->L	H->H
DQ7	L->H	H->H	L->H	L->H	L->L	L->H	H->H	H->H
DBI#	L	H	L	L	H	L	H	H

Bus Data

DBI#[3]=func(DQ_controller(7:0)[3])

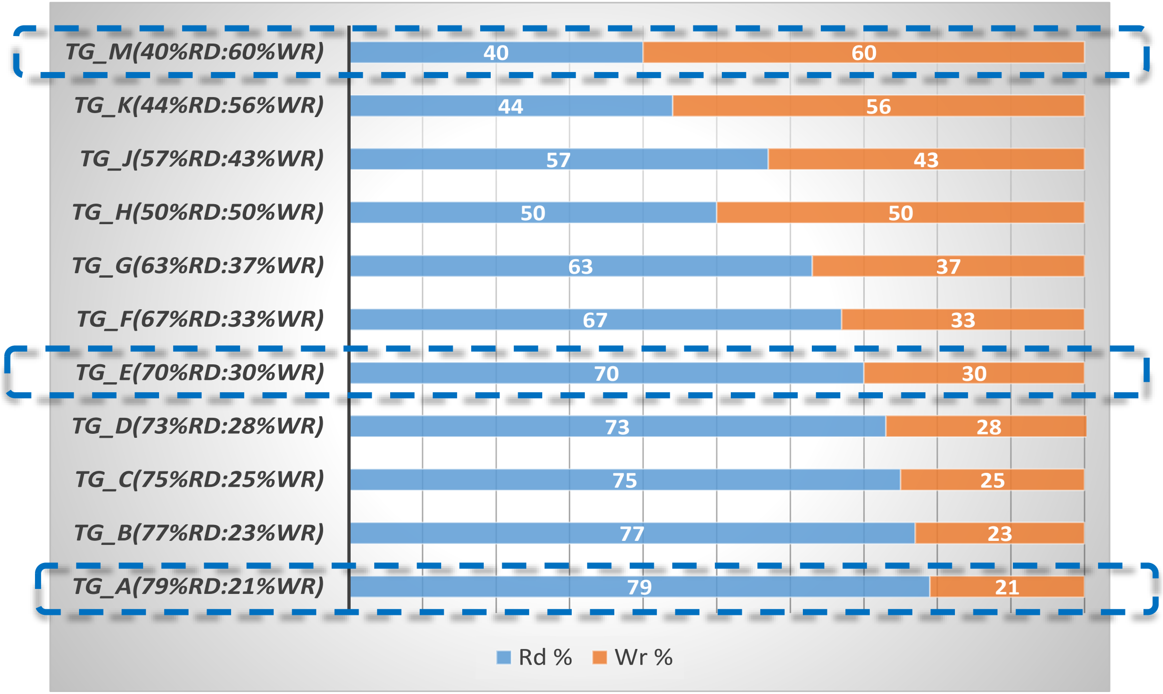
System Power Comparison Set Up


Test Programs (Traffic Gen) with different Rd%--Wr% ratio

Test Programs	with No DBI	with DBI
TG_a	✓	✓
⋮		
TG_m	✓	✓

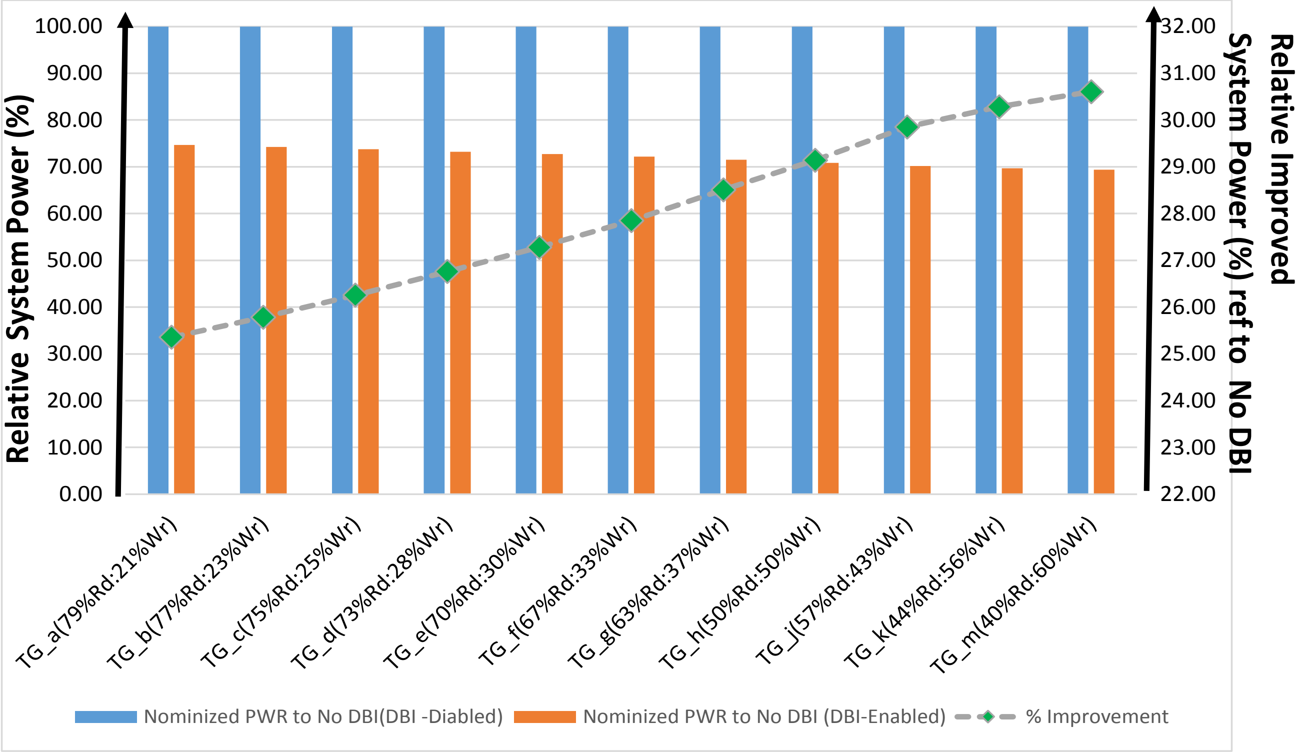
VS

Read & Write Percentage Ratio for Relative Power Comparison



→ *Analyze the relative power improvement with different work loads.*

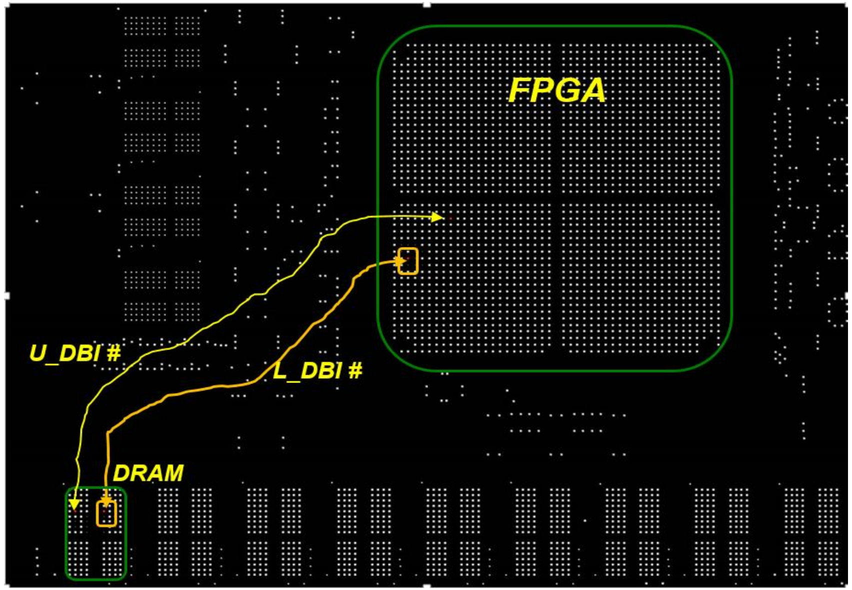
Relative Power Improvement with DBI

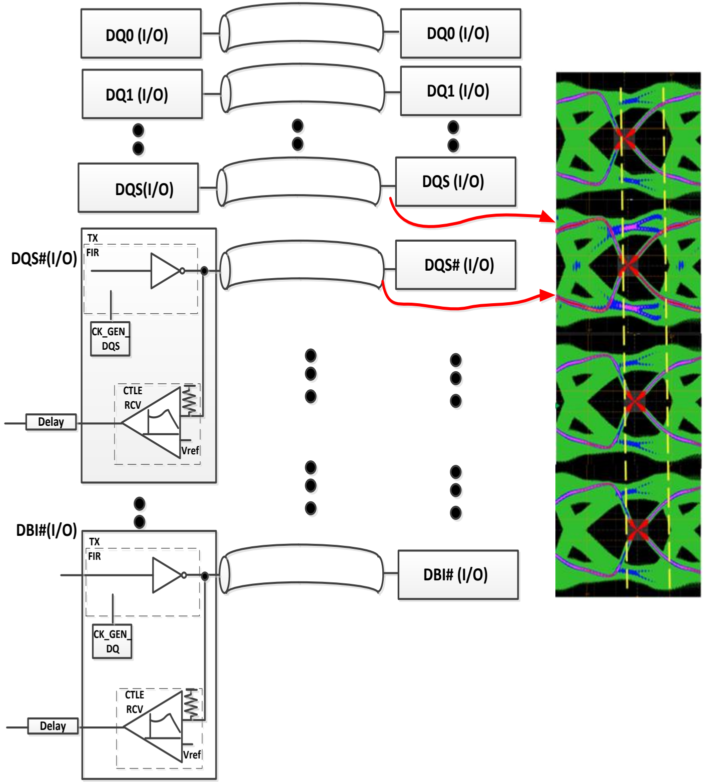


→ *System with DBI enabled shows relative power improvement.*

→ *Improved amount varies with Read and Write % ratio*

DBI need Calibration





→ *DBI bit need to be calibrated together with other DQ bits*

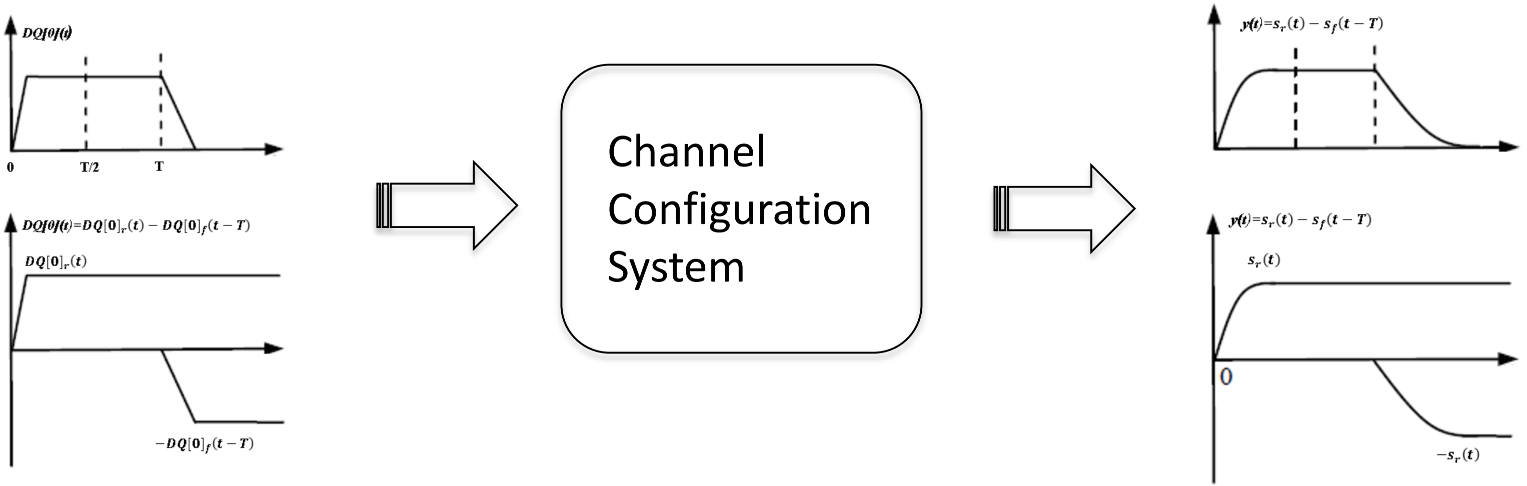
Step Function Representation of with DQ Pattern

$$DQ[0](t) = DQ[0]_r(t - r_1T) - DQ[0]_f(t - f_1T) + \cdots DQ[0]_r(t - r_iT) - DQ[0]_f(t - f_iT) + \cdots$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$DQ[7](t) = DQ[7]_r(t - r_1T) - DQ[7]_f(t - f_1T) + \cdots DQ[7]_r(t - r_iT) - DQ[7]_f(t - f_iT) + \cdots$$

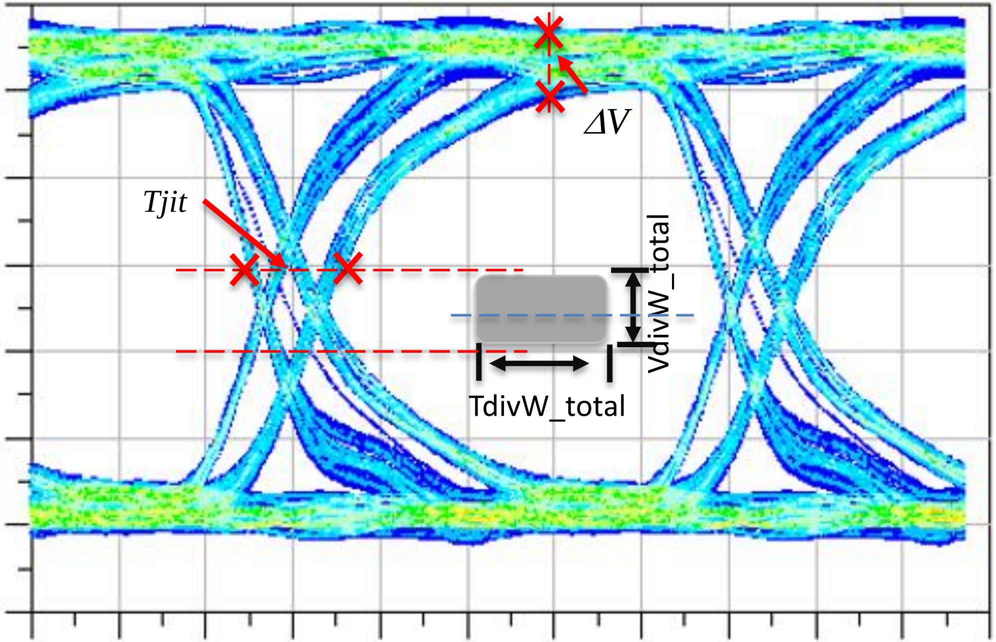
$$DQS(t) = DQS_r\left(t - r_1(T - \frac{T}{2})\right) - DQS_f\left(t - f_1\left(T - \frac{T}{2}\right)\right) + \cdots + DQS_r\left(t - r_i(T - \frac{T}{2})\right) - DQS_f\left(t - f_i(T - \frac{T}{2})\right) + \cdots$$



Channel Configuration System

DQ Eye Reference to DQS

$$DQ\_{}_{DQS}\,Eye(t) = \{\{y(t + k_i T) | 0 \le t \le T,\ \forall\ k_i \in \mathbb{N}_0\}\},\ i = [r, f]$$



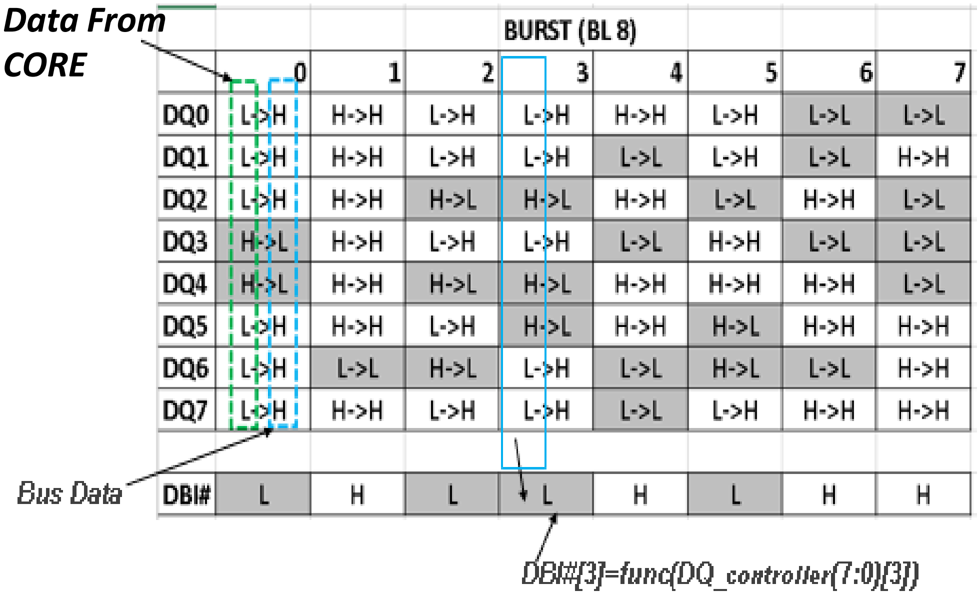
→ *Based on the rise and fall unit step response & their combinations:-*

→ *Construct calibration pattern & to search for worst case jitter and eye height.*

DBI bit Calibration with DQ

$$DBI\#[k] = func(DQ(7{:}0)[k]) \qquad \forall \quad k \in \mathbb{N}_0$$

$$func(\eta) = sum_{logic_{low}}(\eta) > 4$$



Data From CORE

	BURST (BL 8)							
	0	1	2	3	4	5	6	7
DQ0	L->H	H->H	L->H	L->H	H->H	L->H	L->L	L->L
DQ1	L->H	H->H	L->H	L->H	L->L	L->H	L->L	H->H
DQ2	L->H	H->H	H->L	H->L	H->H	L->L	H->H	L->L
DQ3	H->L	H->H	L->H	L->H	L->L	H->H	L->L	L->L
DQ4	H->L	H->H	H->L	H->L	H->H	H->H	H->H	L->L
DQ5	L->H	H->H	L->H	H->L	H->H	H->L	H->H	H->H
DQ6	L->H	L->L	H->L	L->H	L->L	H->L	L->L	H->H
DQ7	L->H	H->H	L->H	L->H	L->L	L->H	H->H	H->H
DBI#	L	H	L	L	H	L	H	H

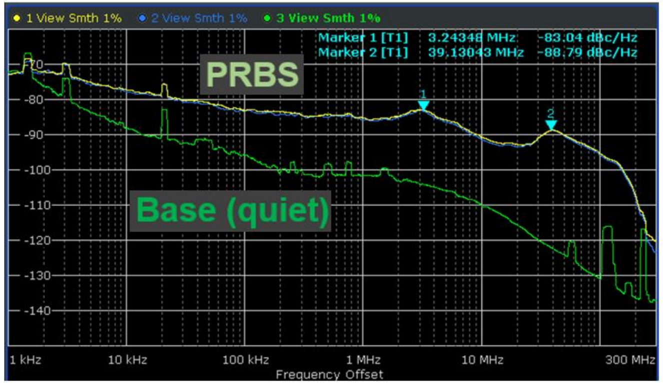
Bus Data

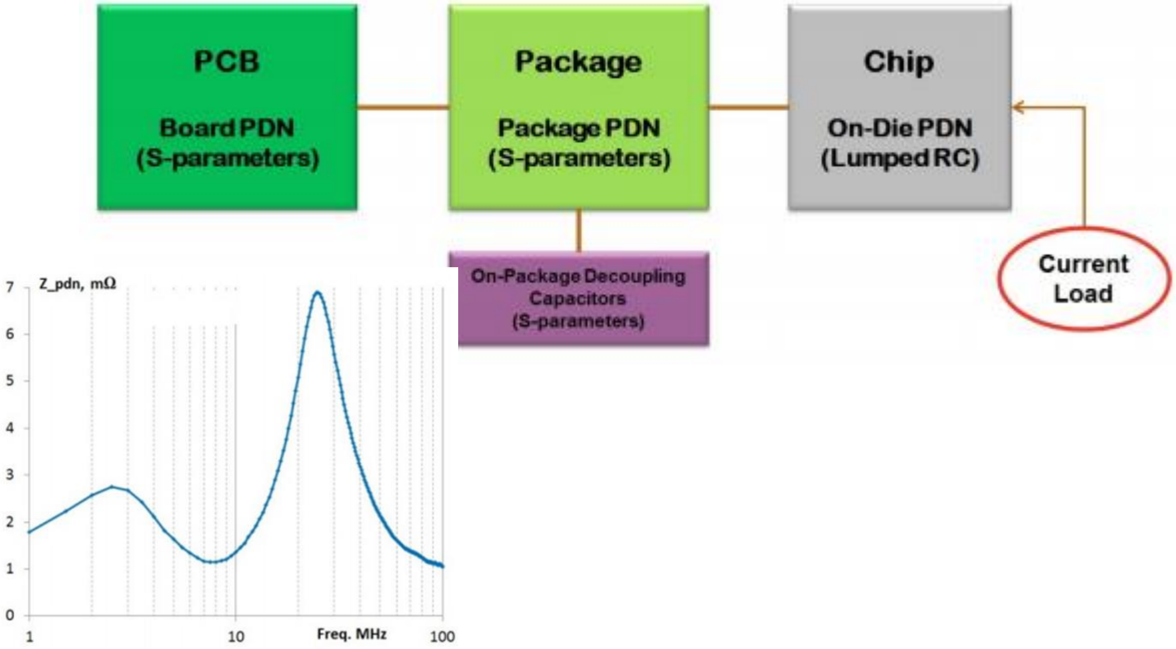
DBI#[3]=func(DQ_controller(7:0)[3])

$$DBI\#\,[k] = func(rand\_cal(DQ(7{:}0))[k]) \qquad \forall \quad k \in \mathbb{N}_0$$

→ *Make sure all DQ bits will have toggling coverage.*

Power Noise Improvement with DBI enabled

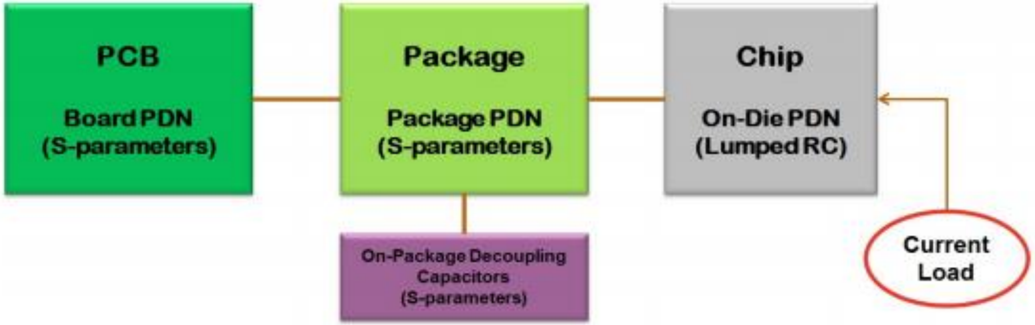


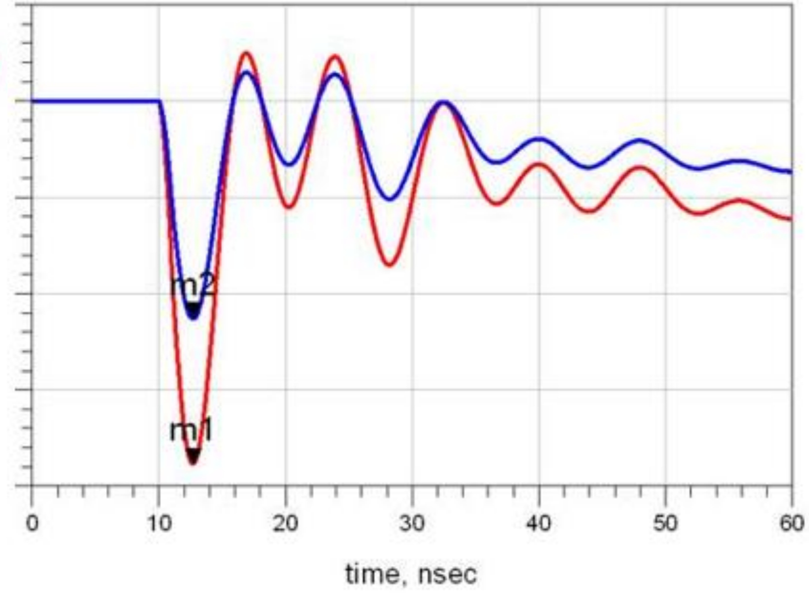


→ *PDN Impedance (Z_pdn) is a function of frequency*

→ *Jitter is a function of Z_pdn and step current load characteristic.*

Voltage Droop Improvement with DBI Enabled





→ *Average step current reduced by enabling DBI.*

→ *Voltage Droop performance improves.*

System Eye Margin Improvement Validation Set Up

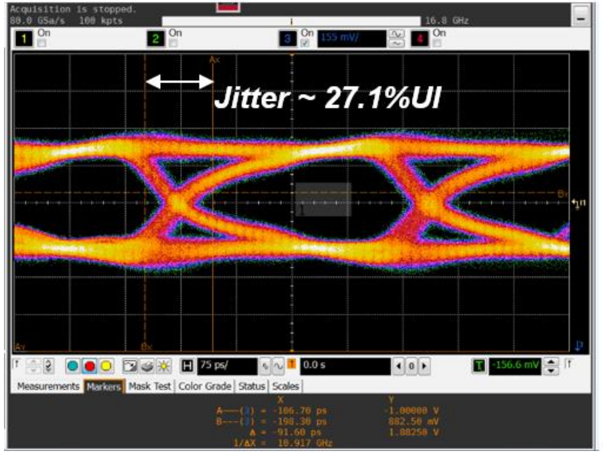
→ *Validation Methods:-*

→ *Direct measurement of DQ Eye at DRAM inputs.*

→ *Write and Read Eye Shmoo.*

→ *Compare with and without DBI enabled.*

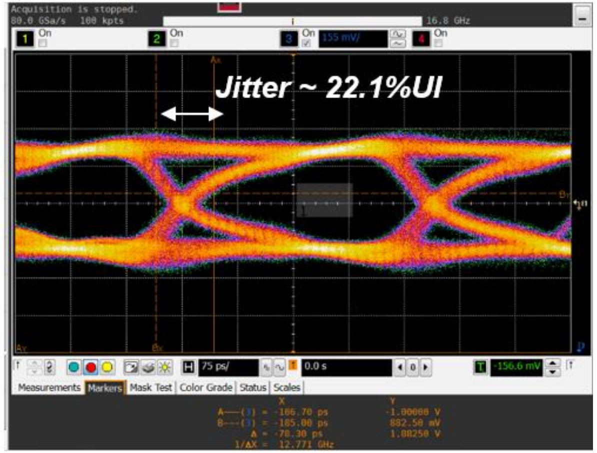
Direct Write Eye Measurement at DRAM



Jitter ~ 27.1%UI

```
Acquisition is stopped.
80.0 GSa/s   100 kpts                16.8 GHz

                 X                    Y
A----(3) = -106.70 ps          -1.00000 V
B----(3) = -198.30 ps           882.50 mV
       Δ = -91.60 ps            1.88250 V
    1/ΔX =  10.917 GHz
```



Jitter ~ 22.1%UI

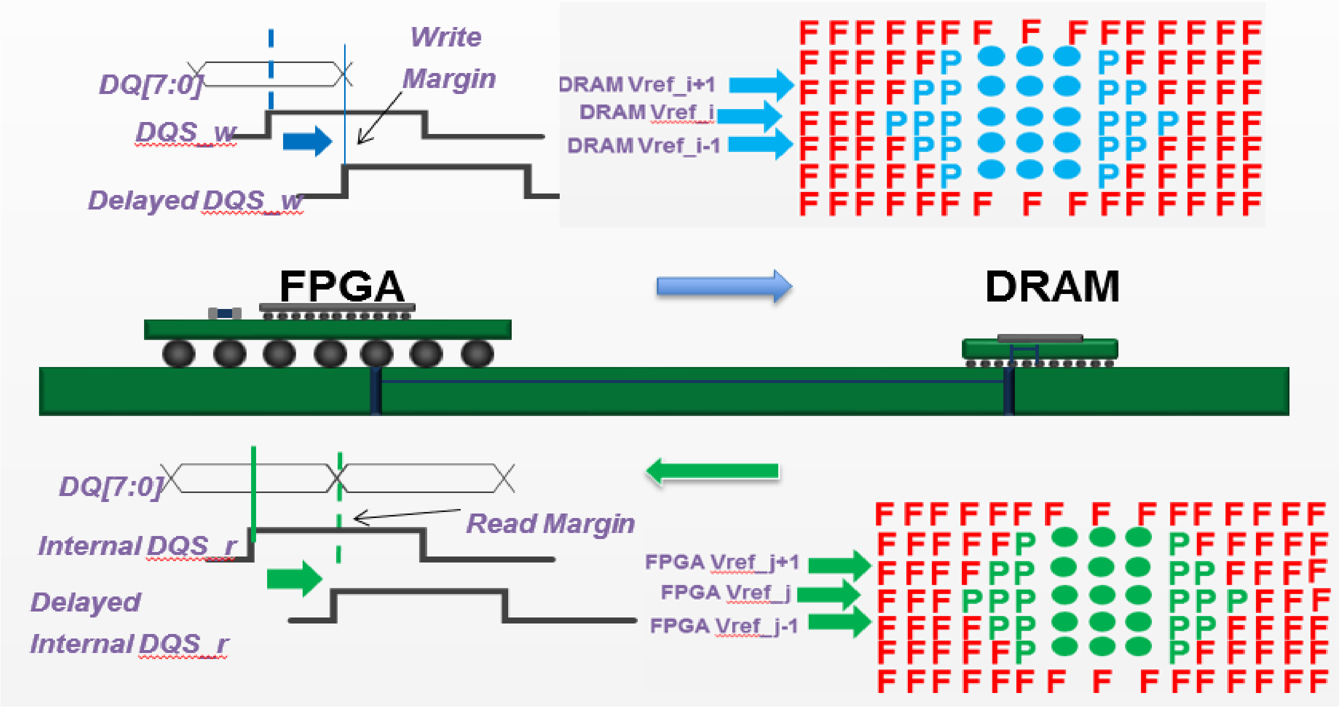
```
Acquisition is stopped.
80.0 GSa/s   100 kpts                16.8 GHz

                 X                    Y
A----(3) = -106.70 ps          -1.00000 V
B----(3) = -185.00 ps           882.50 mV
       Δ = -78.30 ps            1.88250 V
    1/ΔX =  12.771 GHz
```

→ *Write Eye measurement shows a 5% UI jitter improvement.*

→ *Validation extended to create functional Read and Write Eye shmoo next.*

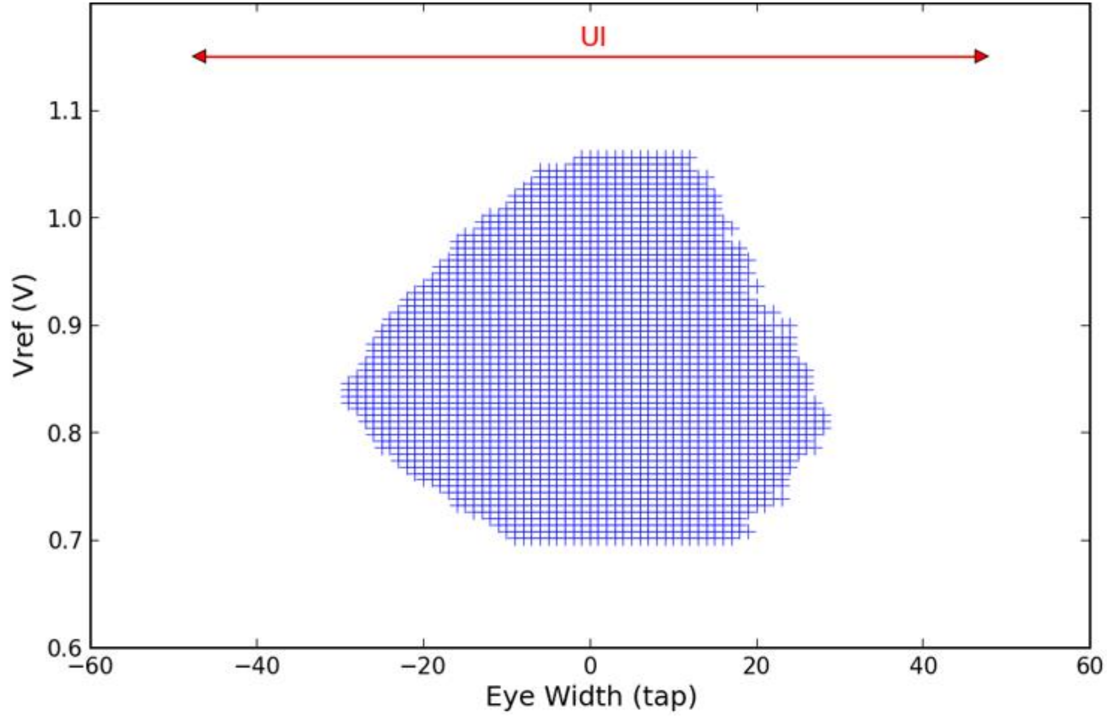
Read and Write Shmoo Set Up



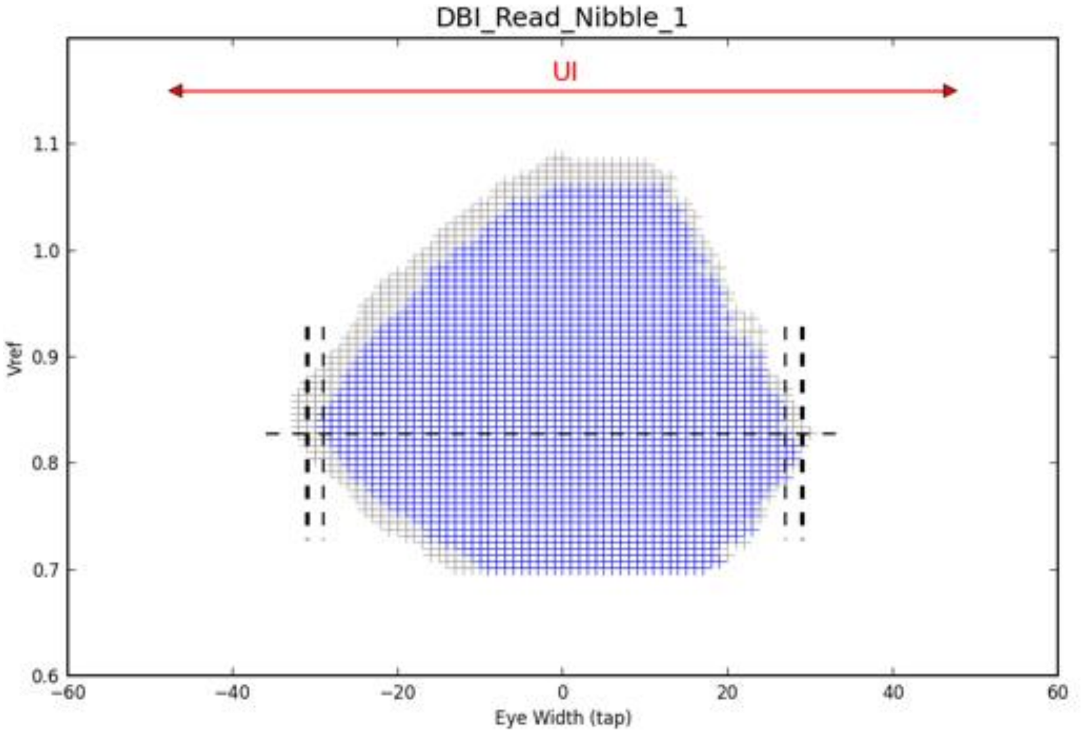
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
F	F	F	F	F	P	●	●	●	P	F	F	F	F	F
F	F	F	F	P	P	●	●	●	P	P	F	F	F	F
F	F	F	P	P	P	●	●	●	P	P	P	F	F	F
F	F	F	F	P	P	●	●	●	P	P	F	F	F	F
F	F	F	F	F	P	●	●	●	P	F	F	F	F	F
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
F	F	F	F	F	P	●	●	●	P	F	F	F	F	F
F	F	F	F	P	P	●	●	●	P	P	F	F	F	F
F	F	F	P	P	P	●	●	●	P	P	P	F	F	F
F	F	F	F	P	P	●	●	●	P	P	F	F	F	F
F	F	F	F	F	P	●	●	●	P	F	F	F	F	F
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

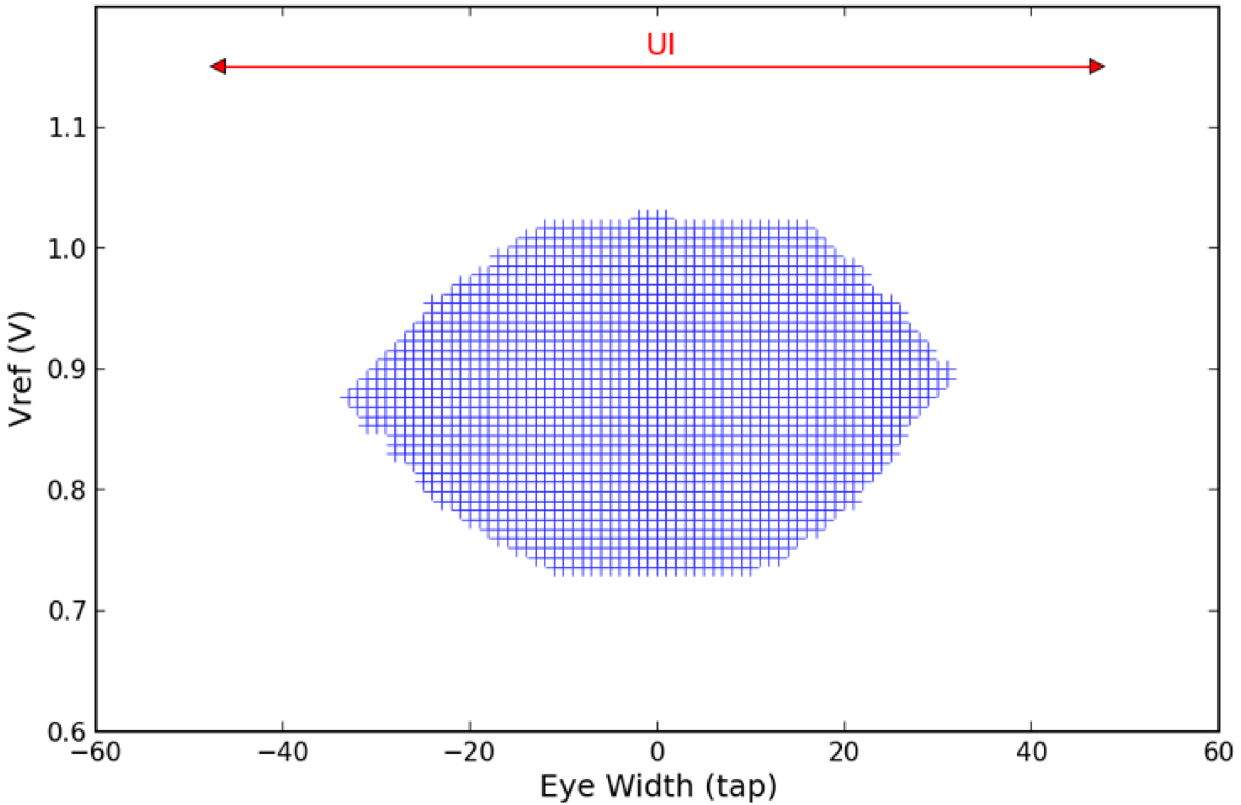
Read Eye Shmoo without DBI Enabled



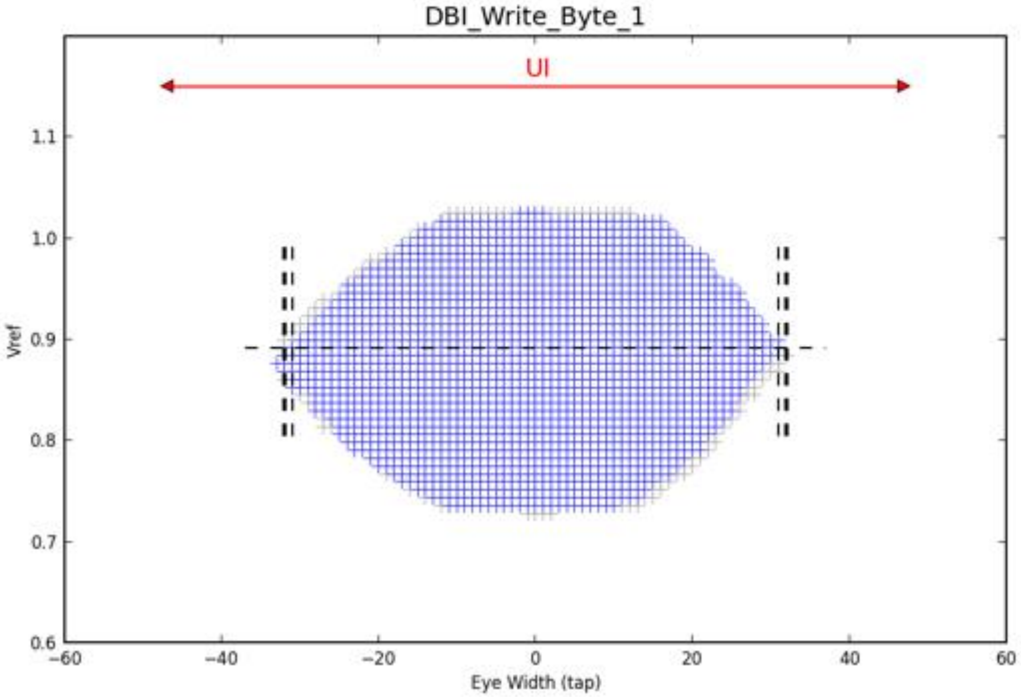
Read Eye Shmoo with and without DBI Enabled



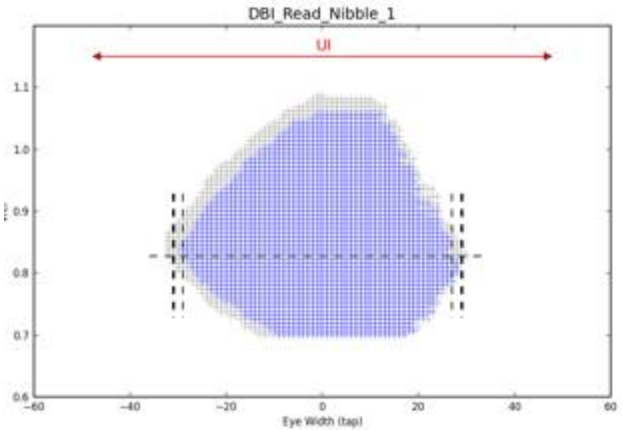
Write Eye Shmoo without DBI Enabled

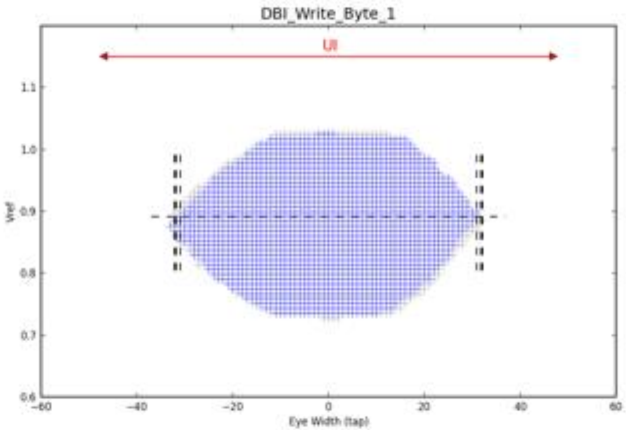


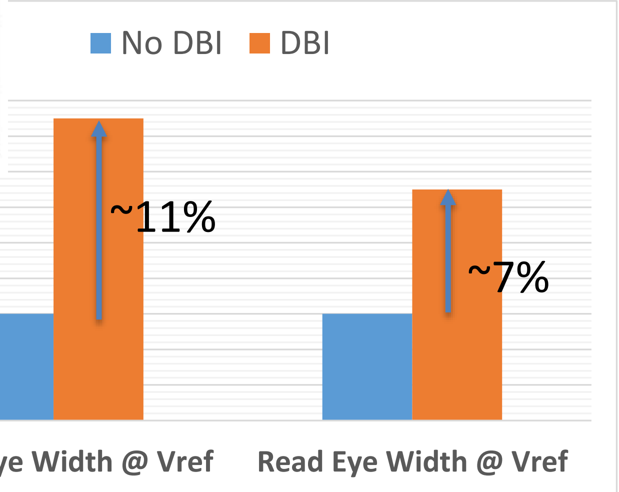
Write Eye Shmoo with and without DBI Enabled



Eye Shmoo Comparison







- → *Eye width improvement observed*
- → *Improvement amount are different.*
- → *Write improved by 11%*
- → *Read improved by 7%*
- → *Different improvement implies different step current impact*
- → *Different PDN between DRAM unit and controller PHY.*

Summary and Conclusions

- Computing Performance requirements drive the need to reduce system power.
- System memory Power became one of the major factor to the total system power.
- Traditional improvement methods, such as scaling process node and IO voltage, slow down.
- DDR4 IO introduced DBI function to opportunistically reduce the IO power.
- Power improvement amount varies with Write and Read Ratio.
- DBI reduced the average step current in memory system, hence improved channel margin.
- Experimental data showed the Channel Jitter improvement differs between Write and Read direction.

Thank you!

QUESTIONS?