Secrets of IBIS Sampling

Virtual IBIS Summit with DesignCon 2021

August 19, 2019

Presenter: Hansel Desmond Dsilva (Achronix Semiconductor)

Co-authors: Todd Bermensolo (Keysight), Adam Gregory (Samtec) and Michael Mirmak (IBIS Enthusiast)

Acknowledgement

- The authors would like to express their deep appreciation for contributions by colleagues at several EDA vendors, without whom this presentation would not have been possible.
- Unfortunately, their names and those of their employers cannot be shared without disclosing product-specific performance information, which would imply conclusions about product quality that authors do not intend.

Overview. IBIS Summit at DesignCon 2020

- **Title**
  - Gap in IBIS for sampling with statistical mode AMI models
  - https://ibis.org/summits/jan20/bermensolo.pdf

- **Problem statement**
  - In AMI_Init flow, no sampling information from model to EDA tool.

- **Outcome**
  - BIRD205. New AMI Reserved Parameter for Sampling Position in AMI_Init Flow.

- **Presentation content**
  - Comparing six EDA tools for eye contour using the AMI_Init Flow.

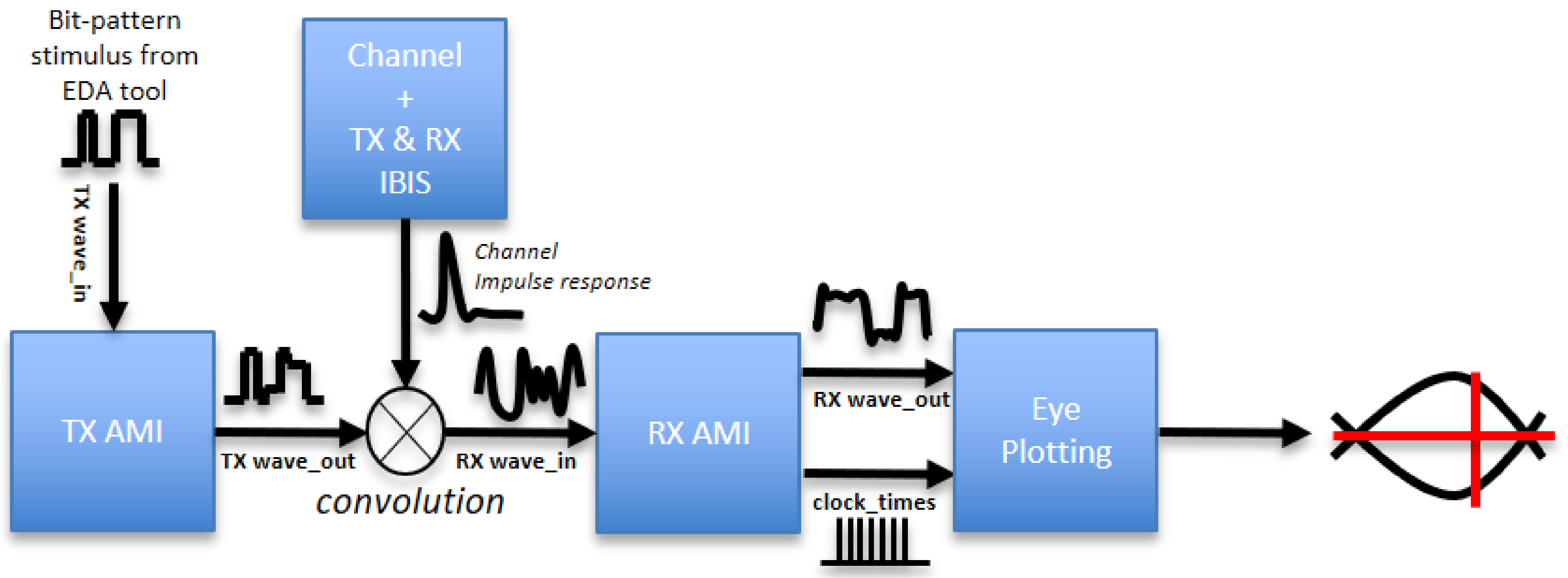


Figure. IBIS-AMI Bit-by-bit flow.

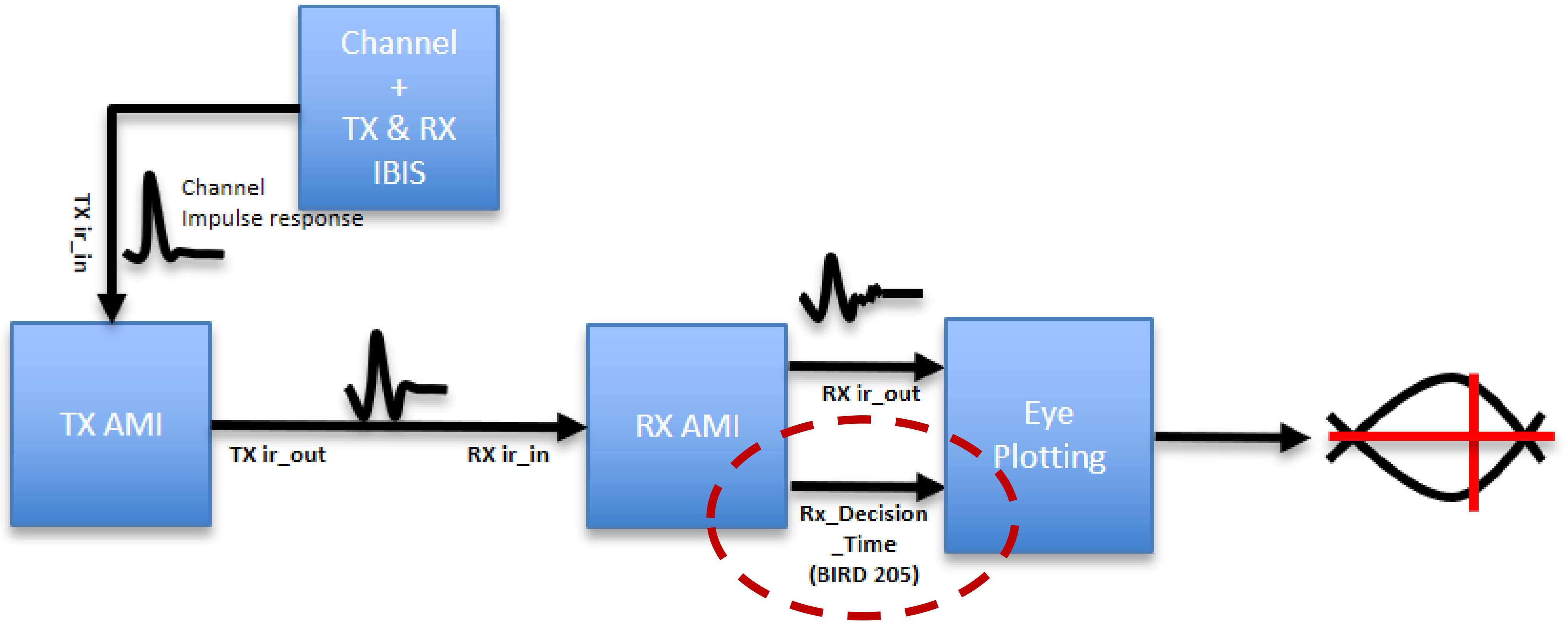


Figure. IBIS-AMI statistical flow.

Outline

1. Variation in eye margin across seven EDA tools
   - ❑Setup
   - ❑Results of bit-by-bit flow
   - ❑Results of statistical flow

2. Importance of sampling information in IBIS
   - ❑Sampling control in the IBIS Specification
   - ❑Experimenting with different sampling

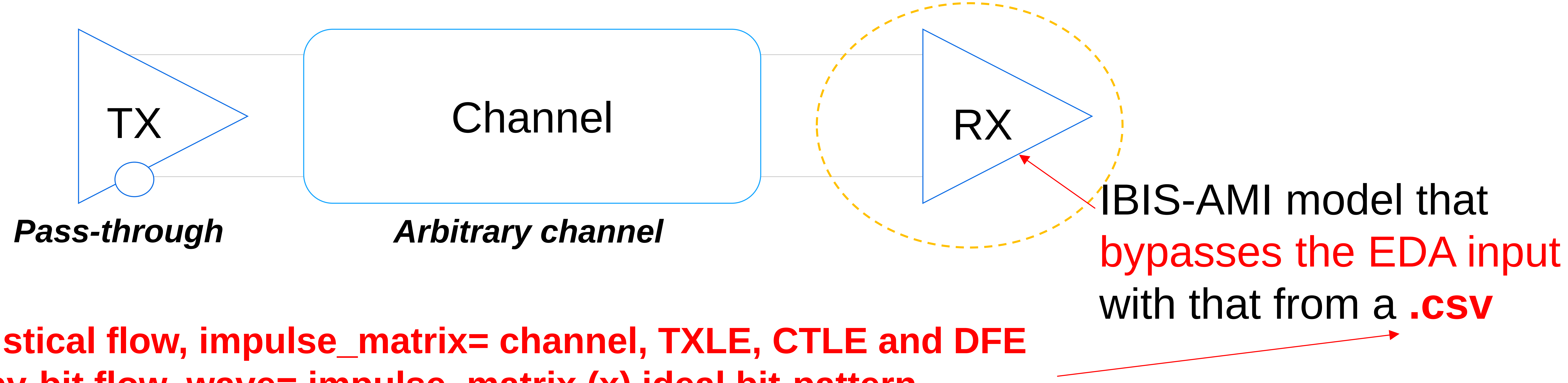
3. Key takeaway

1. Variation in eye margin across different EDA tools

Simulation setup

32 Gbps NRZ; 32 samples per UI
1e6 bits using PRBS-23

1e-5 BER (no extrapolation)
No jitter and noise

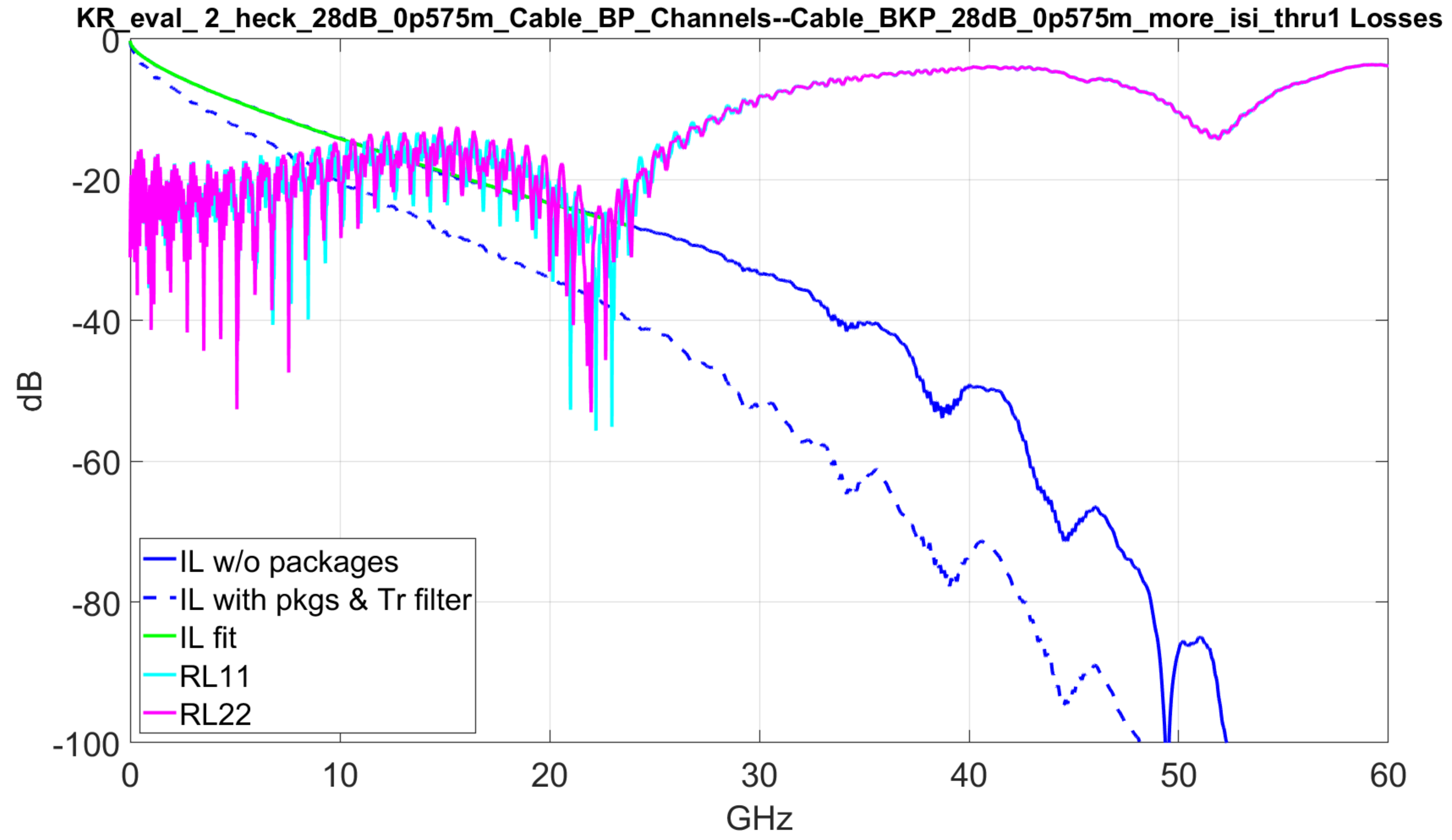


If statistical flow, impulse_matrix= channel, TXLE, CTLE and DFE
If bit-by-bit flow, wave= impulse_matrix (x) ideal bit-pattern

Figure. Channel simulation setup.

- Rx IBIS-AMI model reads the equalized impulse_matrix and wave from a .csv.
- The above setup ensures the same waveform along with sampling information across different EDA tools.
- The results of the different EDA tools is compared to a reference, which is generated using code shared in backup.

Channel and Equalization



	A	B	C	D	F	G	H	J	K	L
1	Table 93A-1 parameters				I/O control			Table 93A–3 parameters		
2	**Parameter**	**Setting**	**Units**	**Information**	DIAGNOSTICS	1	logical	**Parameter**	**Setting**	**Units**
3	f_b	32	GBd		DISPLAY_WINDOW	1	logical	package_tl_gamma0_a1_a2	[0 0.002 0.0003]	
4	f_min	0.05	GHz		CSV_REPORT	1	logical	package_tl_tau	6.141E-03	ns/mm
5	Delta_f	0.01	GHz		RESULT_DIR	.\results\100GEL_1_PK_KR_{date}		package_Z_c	[87.5 87.5 ; 92.5 92.5]	Ohm
6	C_d	[1.2e-4 1.2e-4]	nF	[TX RX]	SAVE_FIGURES	0	logical	benartsi_3ck_01_0119 & mellitz_3ck_01_0119		
7	L_s	[0 0]	nH	[TX RX]	Port Order	[1 3 2 4]		Table 92–12 parameters		
8	C_b	[0 0]	nF	[TX RX]	RUNTAG	KR_eval_		**Parameter**	**Setting**	
9	z_p select	[2]		[test cases to run]	COM_CONTRIBUTION	0	logical	board_tl_gamma0_a1_a2	*[0 3.8206e-04 9.5909e-05]*	
10	z_p (TX)	[12 31; 1.8 1.8]	mm	[test cases]	Operational			board_tl_tau	*5.790E-03*	ns/mm
11	z_p (NEXT)	[12 29; 1.8 1.8]	mm	[test cases]	COM Pass threshold	3	dB	board_Z_c	100	Ohm
12	z_p (FEXT)	[12 31; 1.8 1.8]	mm	[test cases]	ERL Pass threshold	10.5	dB	z_bp (TX)	*110.3*	mm
13	z_p (RX)	[12 29; 1.8 1.8]	mm	[test cases]	DER_0	1.00E-05		z_bp (NEXT)	*110.3*	mm
14	C_p	[0.87e-4 0.87e-4]	nF	[TX RX]	T_r	6.16E-03	ns	z_bp (FEXT)	*110.3*	mm
15	R_0	50	Ohm		FORCE_TR	1	logical	z_bp (RX)	*110.3*	mm
16	R_d	[50 50]	Ohm	[TX RX]				C_0	[0.29e-4]	nF
17	A_v	0.415	V		TDR and ERL options			C_1	[0.19e-4]	nF
18	A_fe	0.415	V		TDR	1	logical	**Include PCB**	**0**	**logical**
19	A_ne	0.608	V		ERL	1	logical			
20	L	2			ERL_ONLY	0	logical	Floating Tap Control		
21	M	32			TR_TDR	0.01	ns	N_bg	0	0 1 2 or 3 groups
22	filter and Eq				N	3000		N_bf	0	taps per group
23	f_r	0.75	*fb		beta_x	2.3407E+09		N_f	0	UI span for floating taps
24	c(0)	0.54		min	rho_x	0.19		bmaxg	0.2	max DFE value for floating taps
25	c(-1)	[-0.1667]		[min:step:max]	fixture delay time	[0 0]	[port1 port2]	cable assemblies require this for each HCB		
26	c(-2)	[0]		[min:step:max]	TDR_W_TXPKG	0		ICN parameters (v2.73)		
27	c(-3)	[0]		[min:step:max]	N_bx	3	UI	f_f	21.448	
28	c(1)	[-0.0417]		[min:step:max]	Receiver testing			f_n	21.448	
29	N_b	3	UI		RX_CALIBRATION	0	logical	f_2	24.000	
30	b_max(1)	0.8			Sigma BBN step	5.00E-03	V	A_ft	0.600	
31	b_max(2..N_b)	0.3			Noise, jitter			**A_nt**	**0.600**	
32	g_DC	[-15]	dB	[min:step:max]	sigma_RJ	0	UI	heck_3ck_03b_0319	**Adopted Mar 2019**	
33	f_z	12.8	GHz		A_DD	0	UI	**walker_3ck_01d_0719**	**Adopted July 2019**	
34	f_p1	12.8	GHz		eta_0	8.2E-09	V^2/GHz	**result of R_d=50**		
35	f_p2	32	GHz		SNR_TX	100	dB	**benartsi_3ck_01a_0719**	**no used for KR**	
36	g_DC_HP	[0]		[min:step:max]	R_LM	1		**mellitz_3ck_03_0919**		
37	f_HP_PZ	0.4	GHz		CDR	Mod-MM	M or Mod-MM	under consideration		

- .s4p taken from IEEE 802.3 public area.
- -28.53 dB @ 16 GHz.

- COM ver2.75.
- Fixed TXLE, Fixed CTLE and 3-tap DFE.

- Realistic channel and specification defined equalization used.

Top level overview

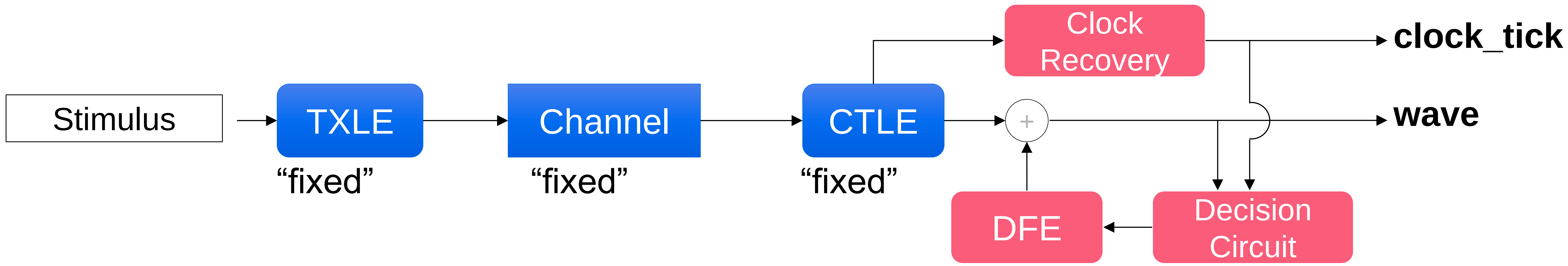


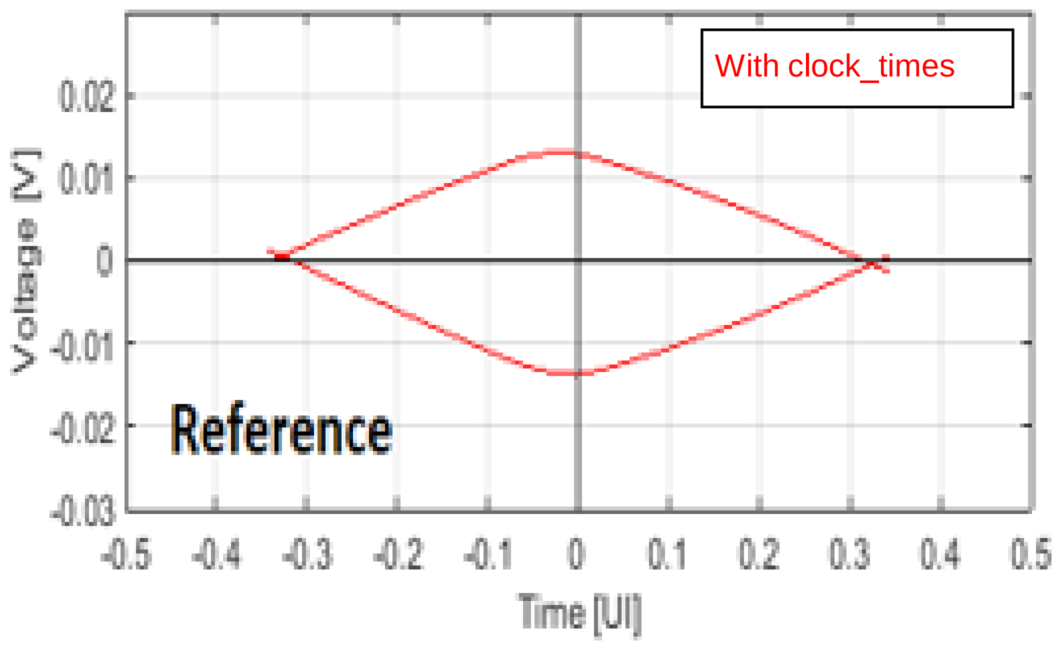
Figure. Block diagram overview.

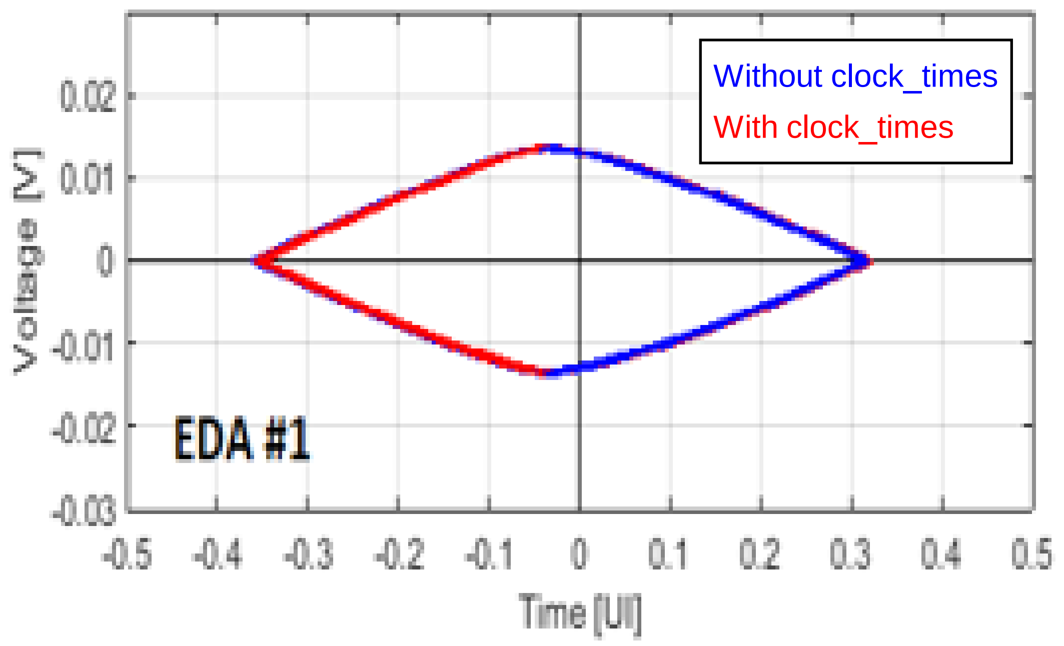
- "Fixed" TXLE, channel and CTLE.
- Zero-forcing DFE.
- Two different phase detectors (part of clock recovery).
  1. Mueller-Muller (MM) Phase detector
     - Align the sampling clock so that precursor ISI equals to post-cursor ISI.
  2. Modified Mueller-Muller (Mod-MM) Phase detector
     - Modified version of the Mueller-Muller PD where the impact of first pre-cursor (pre-1) is removed.

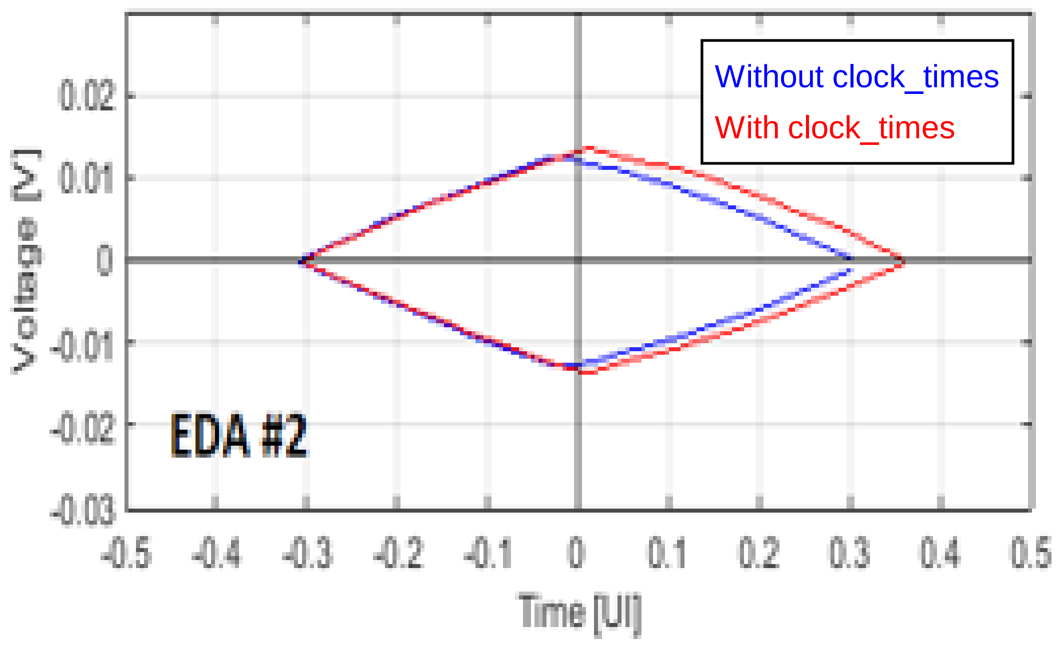
- The performance of MM PD and Mod-MM PD highly depends on the choice of transmitter equalization (TXLE).
- This work does compare one phase detector with another!

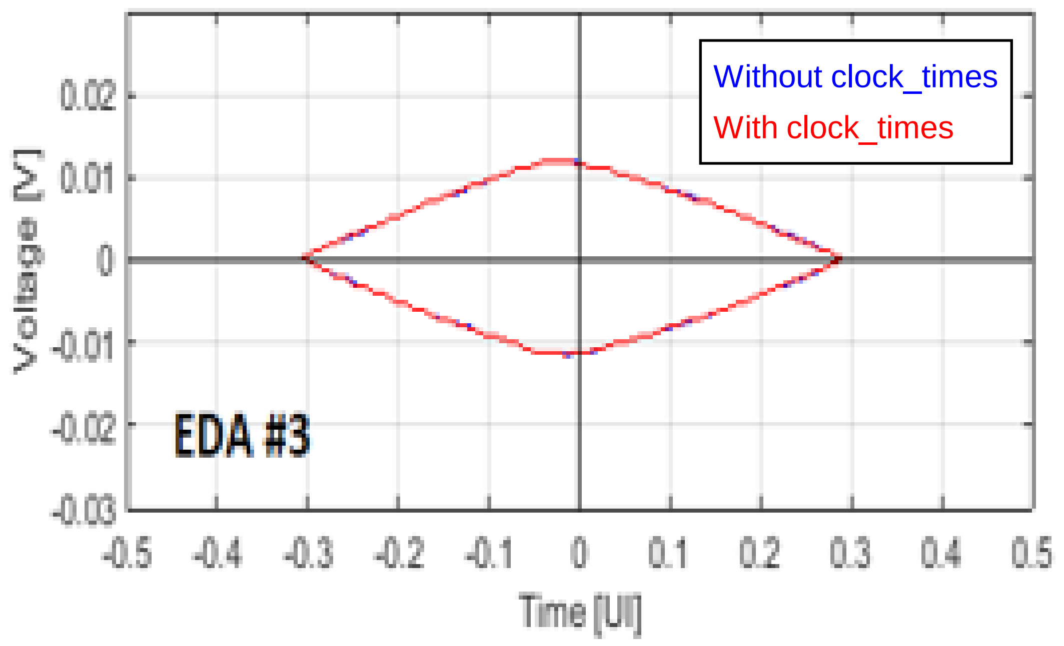
Results of Bit-by-bit flow

Results. Bit-by-bit flow. MM PD.

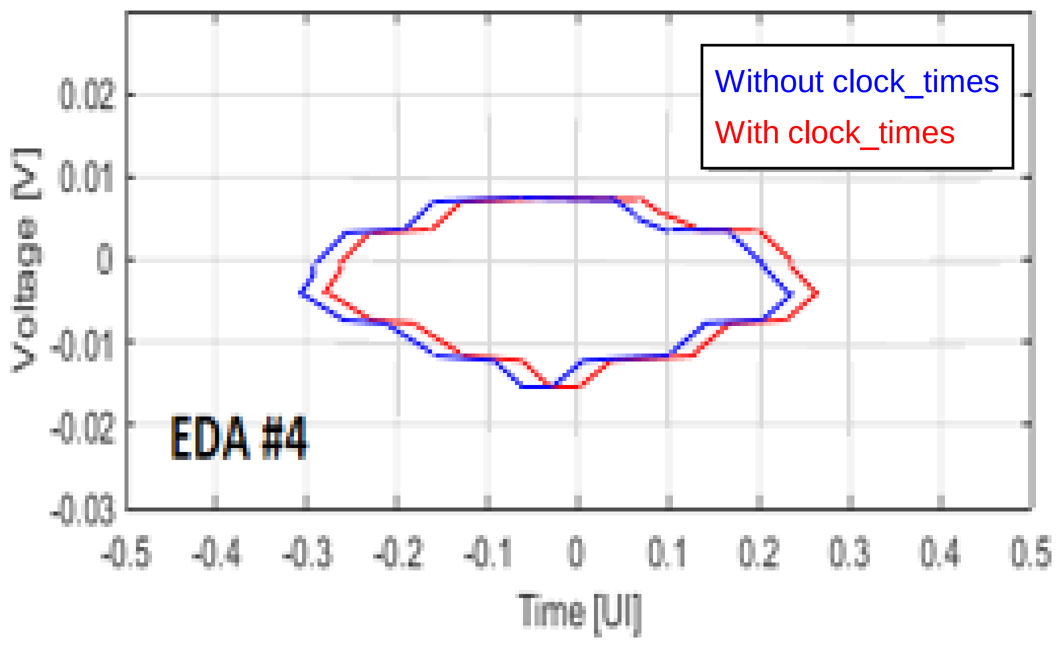


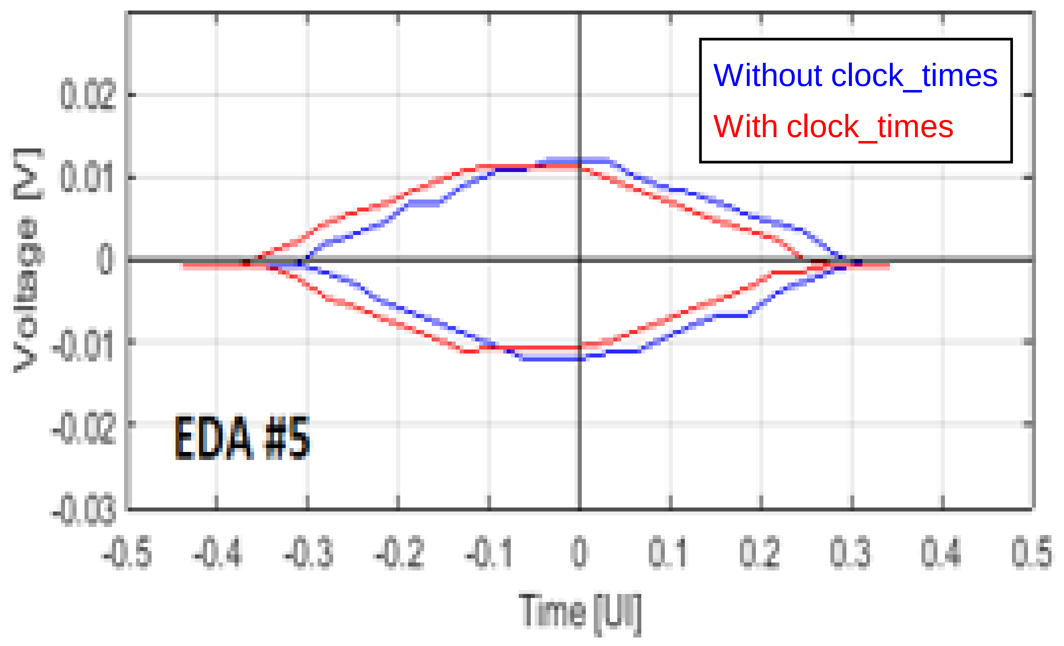


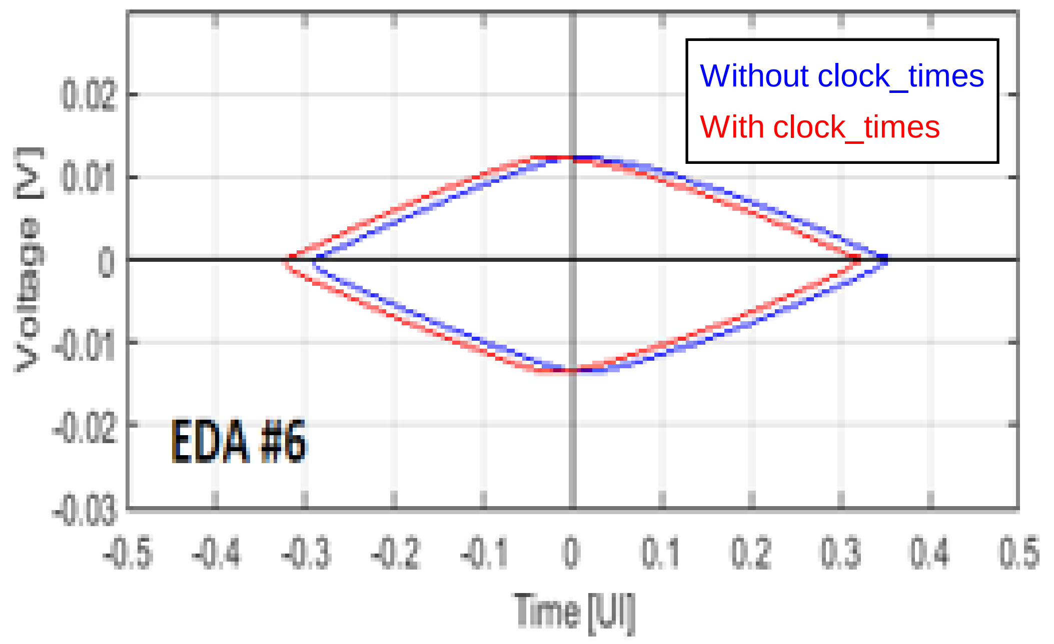


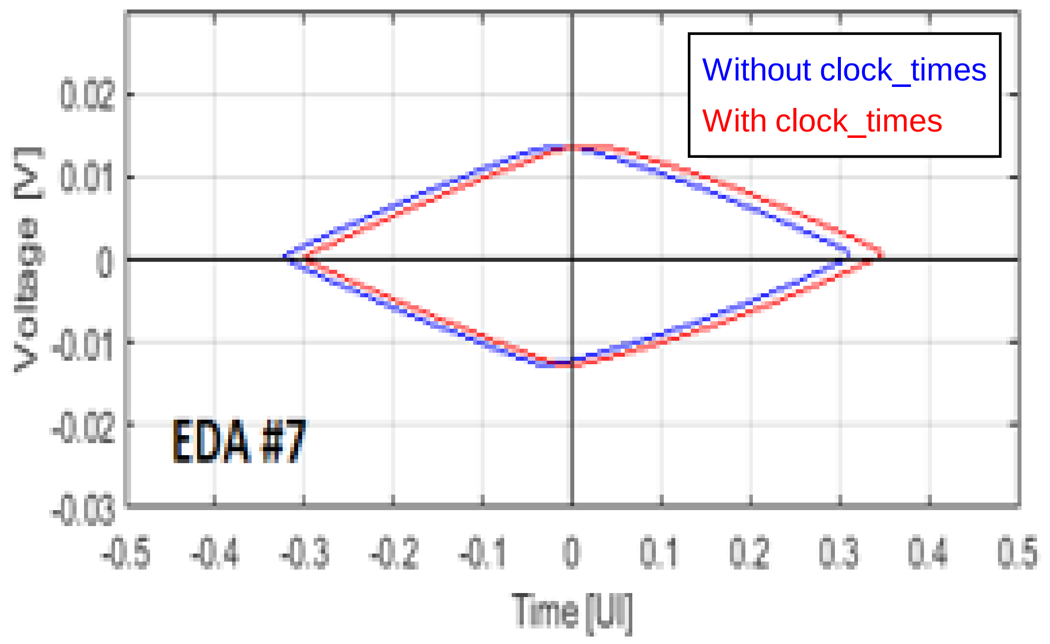


- Reasonable matching of the EDA tools to get a similar positioned eye to the reference.

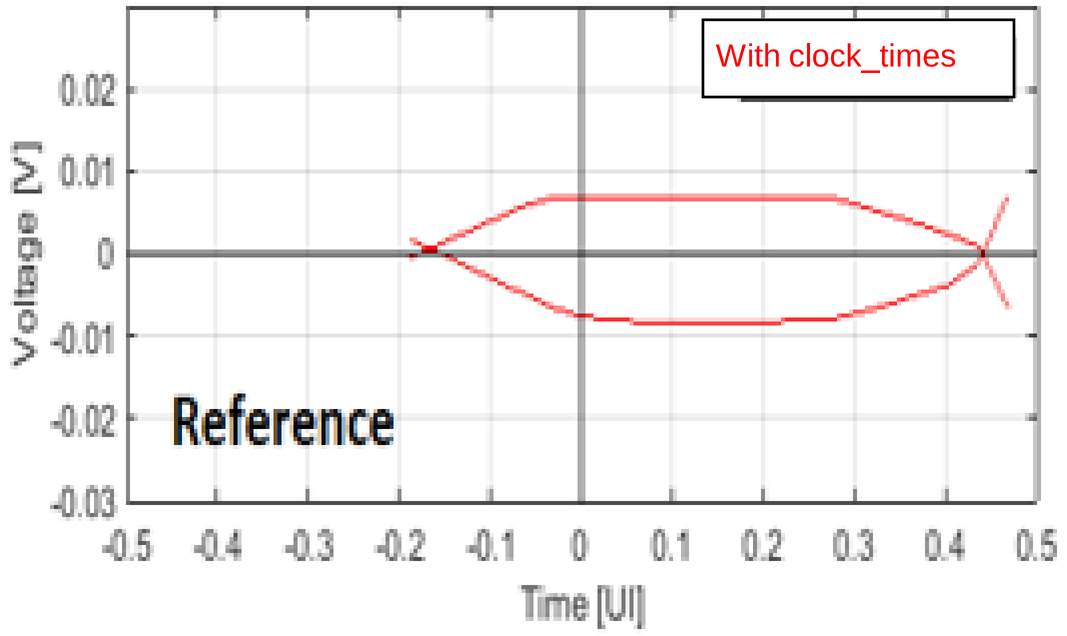


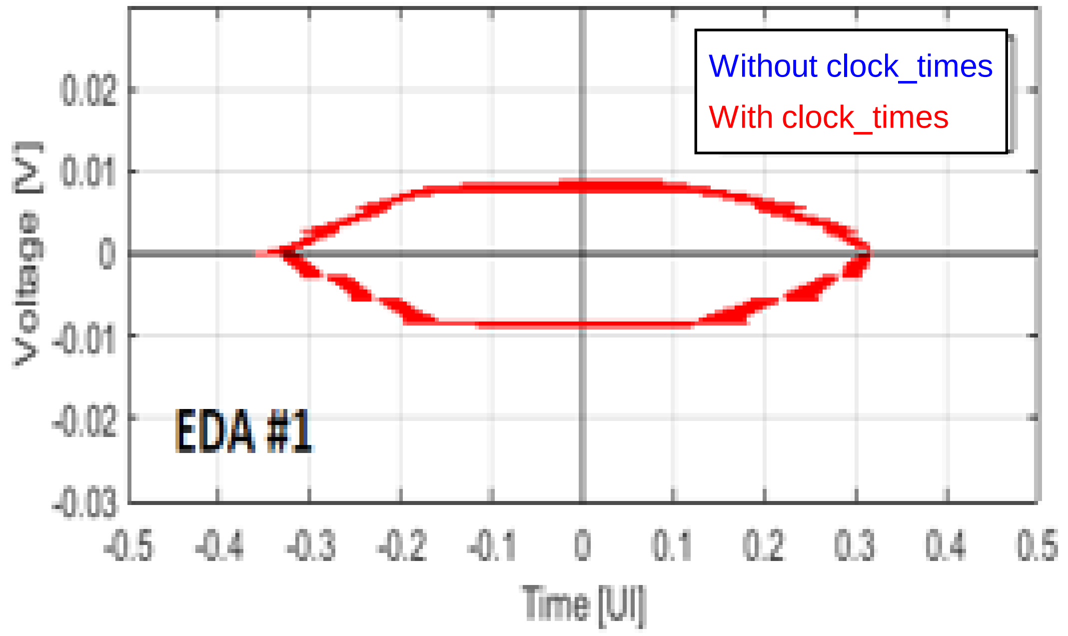


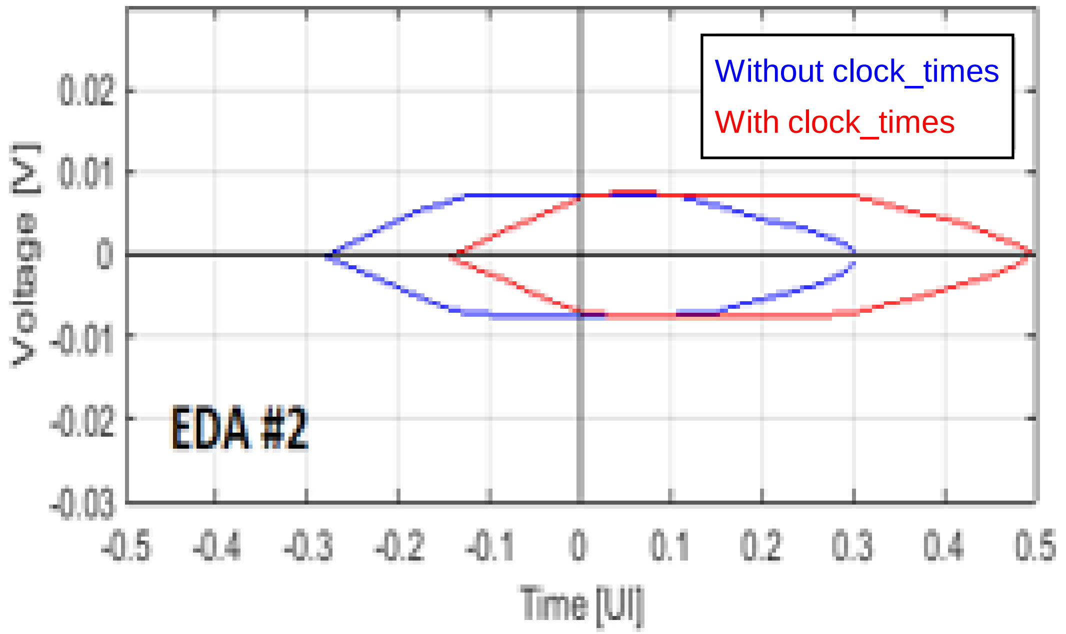


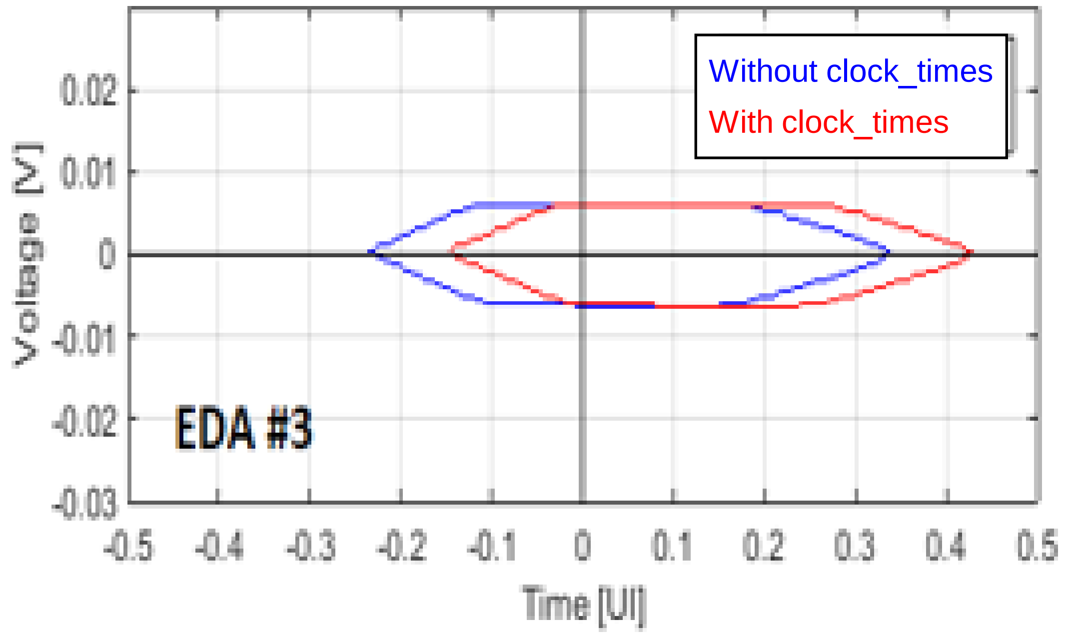


Results. Bit-by-bit flow. Mod-MM PD.

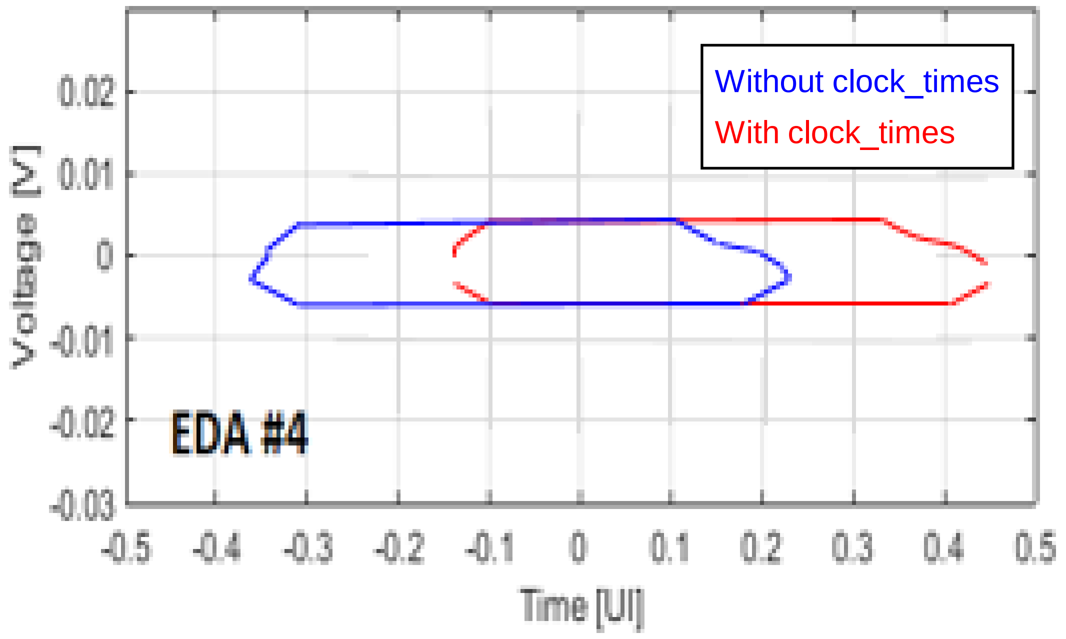


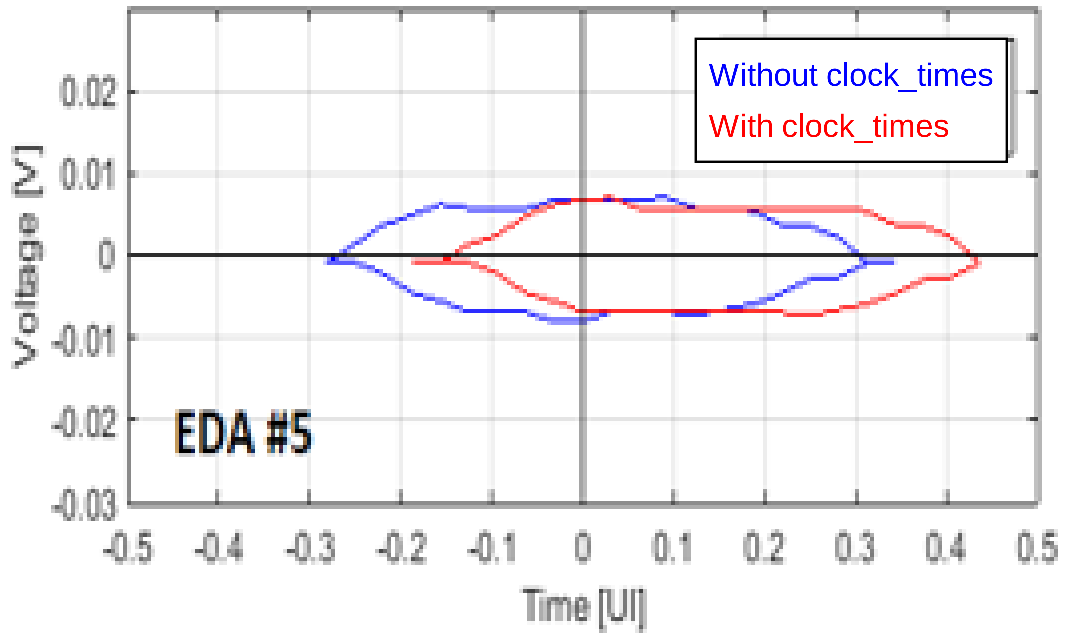


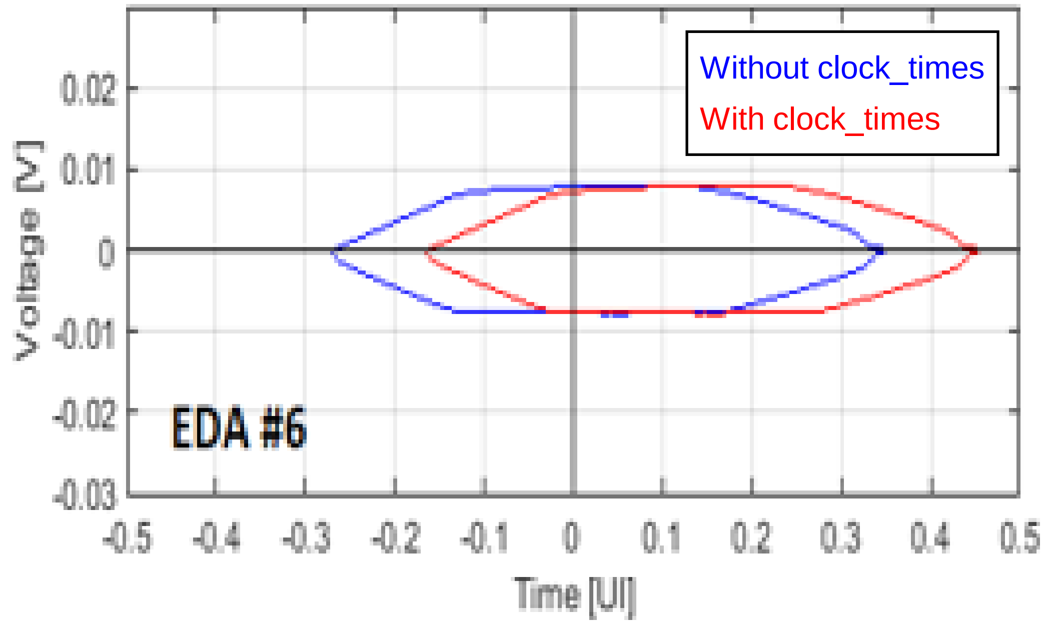


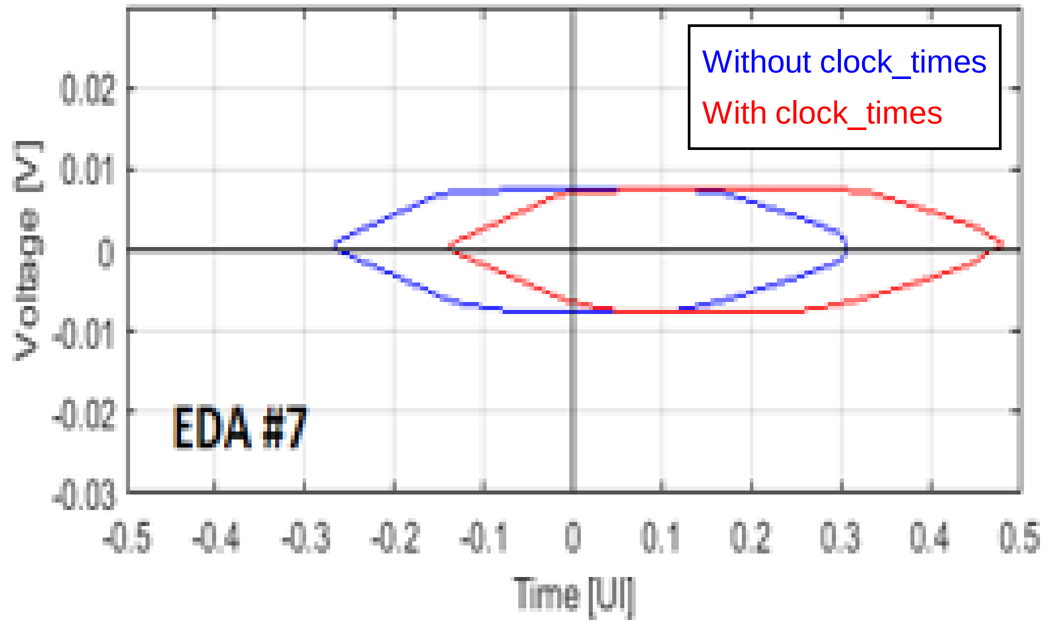


- Observed is that <u>none of the EDA tools are able to get a similar positioned eye matching the reference</u> in the absence of sampling information from the receiver IBIS-AMI (w/o clock_times).









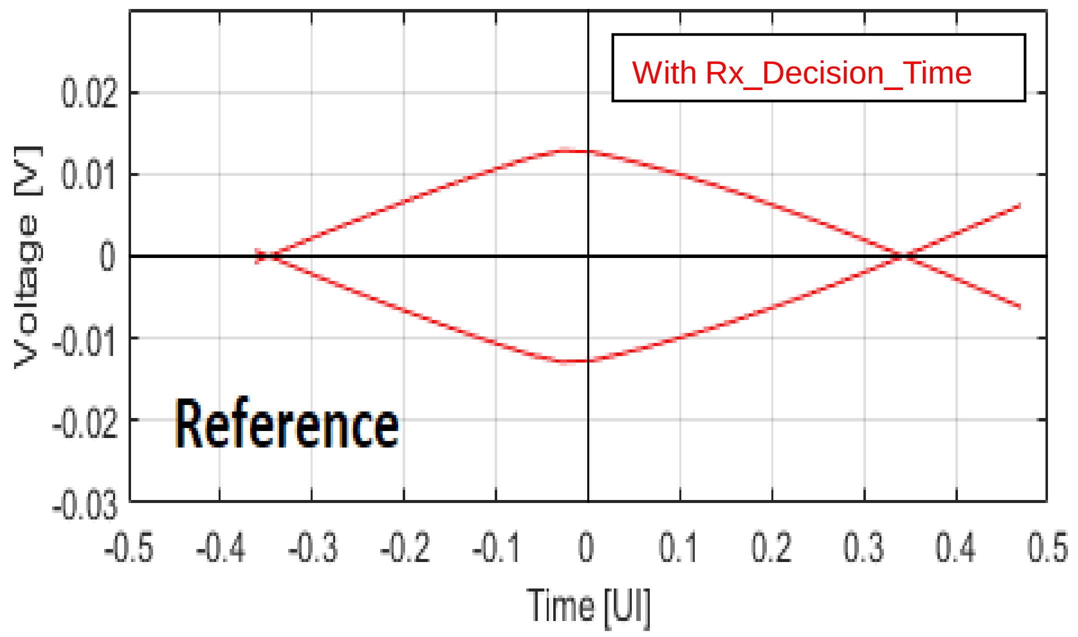
Results. Bit-by-bit BER contour margin.

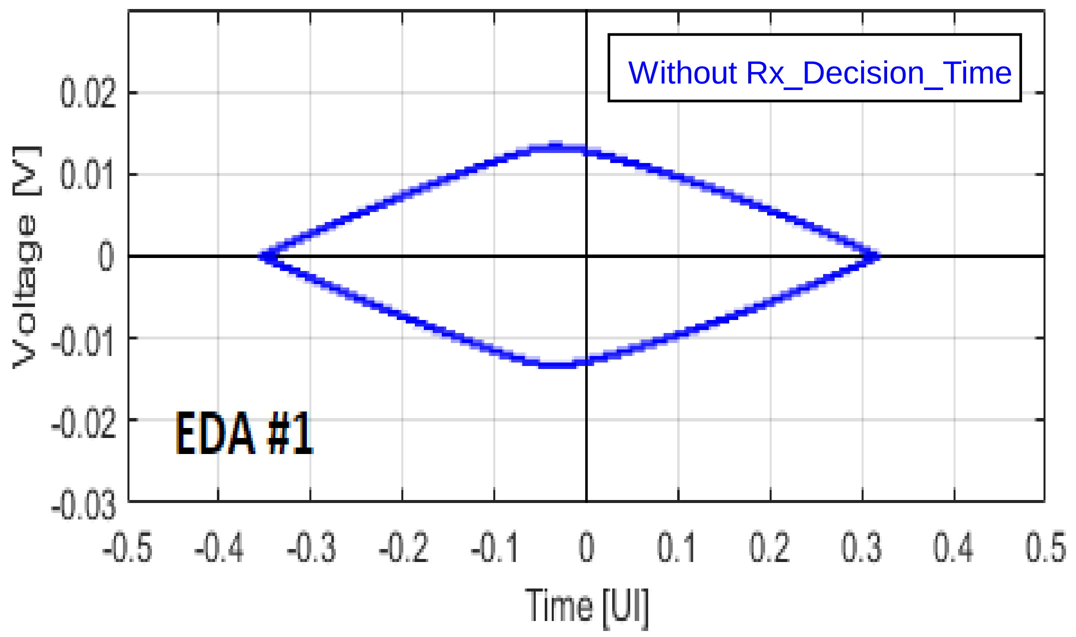
Sampling Method	EDA tool	EH [mV]	EW [UI]
MM-PD	Reference	26.6	0.66
	EDA #1	25.7	0.64
	EDA #2	26.4	0.67
	EDA #3	22.3	0.60
	EDA #4	24.7	0.54
	EDA #5	21.7	0.59
	EDA #6	25.6	0.65
	EDA #7	26.8	0.64
Mod-MM PD	Reference	14.4	0.64
	EDA #1	15.4	0.65
	EDA #2	13.8	0.64
	EDA #3	12.1	0.58
	EDA #4	12.0	0.50
	EDA #5	14.7	0.65
	EDA #6	15.5	0.62
	EDA #7	14.3	0.61

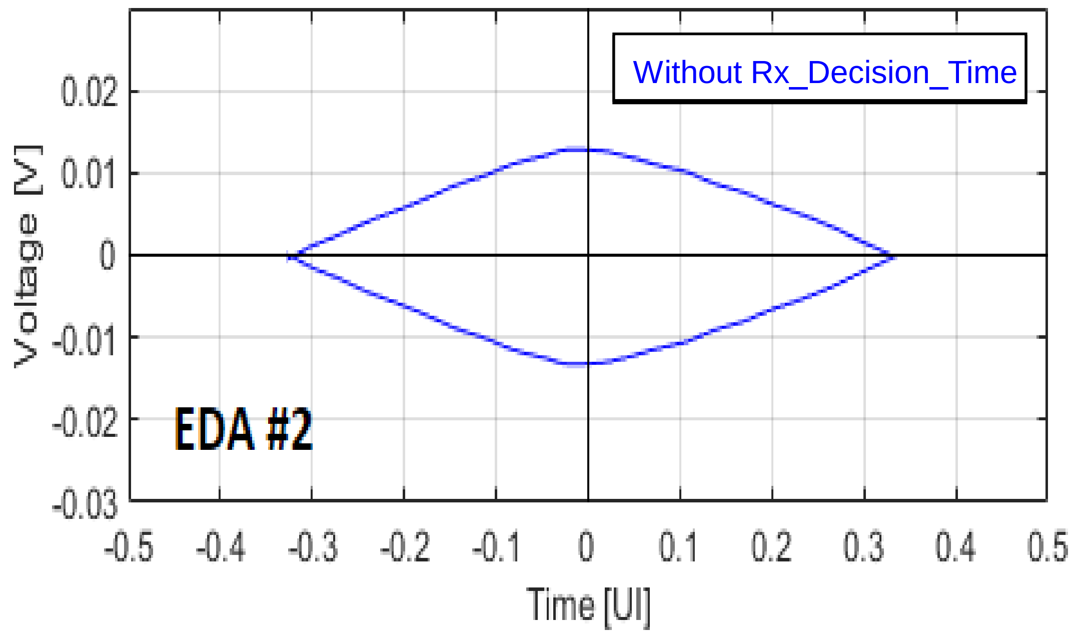
- The results among the different EDA tools are in the ballpark of the reference.
- Observed is a standard deviation of 2mV/0.05UI for the set of seven EDA tools.

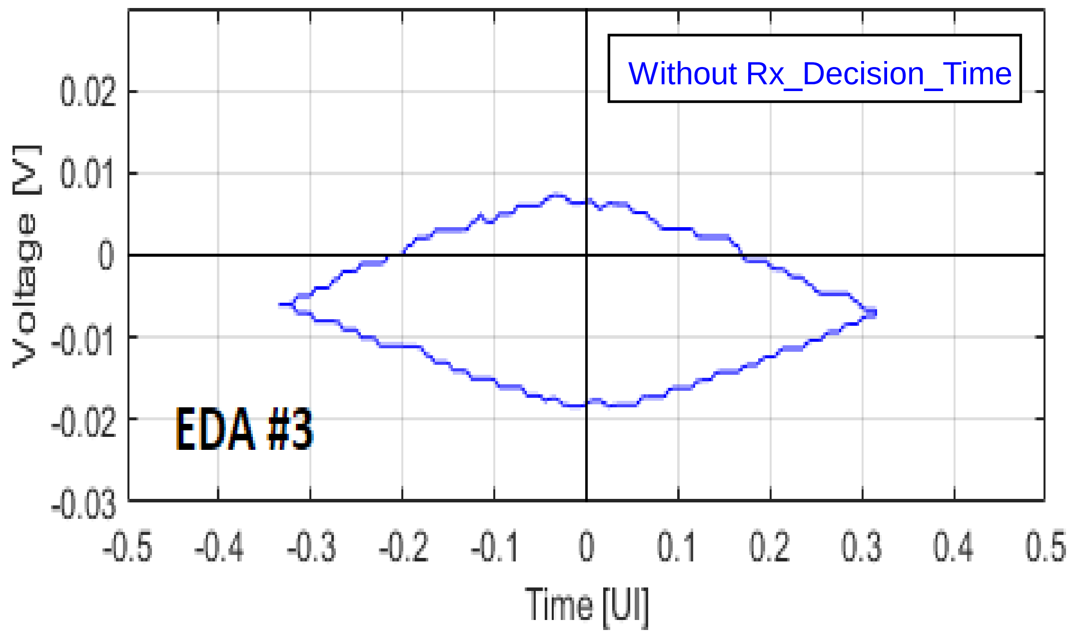
Results of statistical flow

Results. Statistical flow. MM PD.

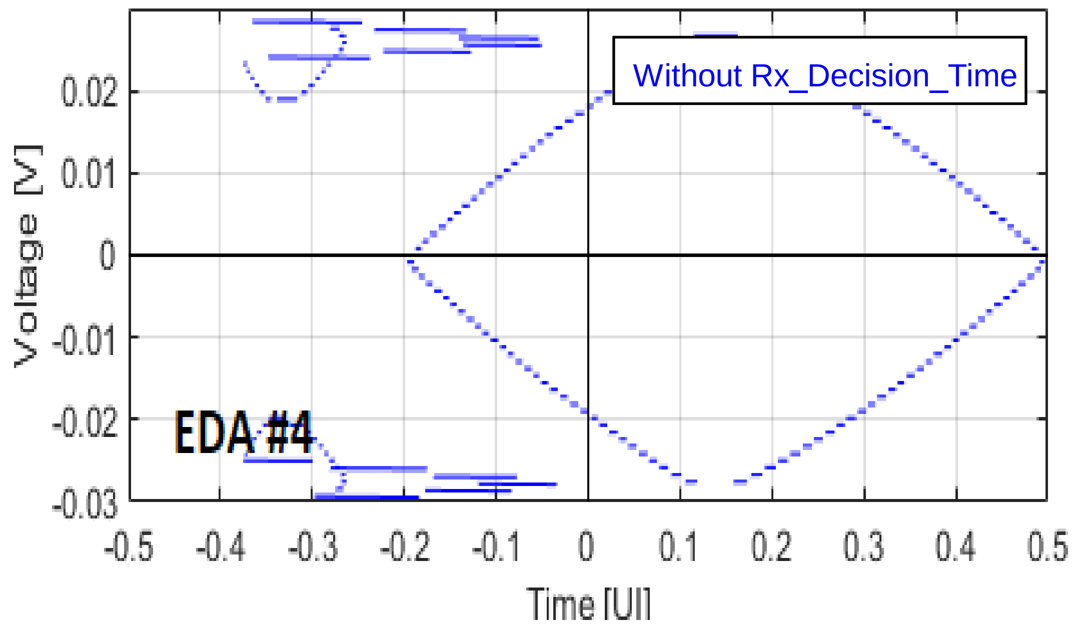


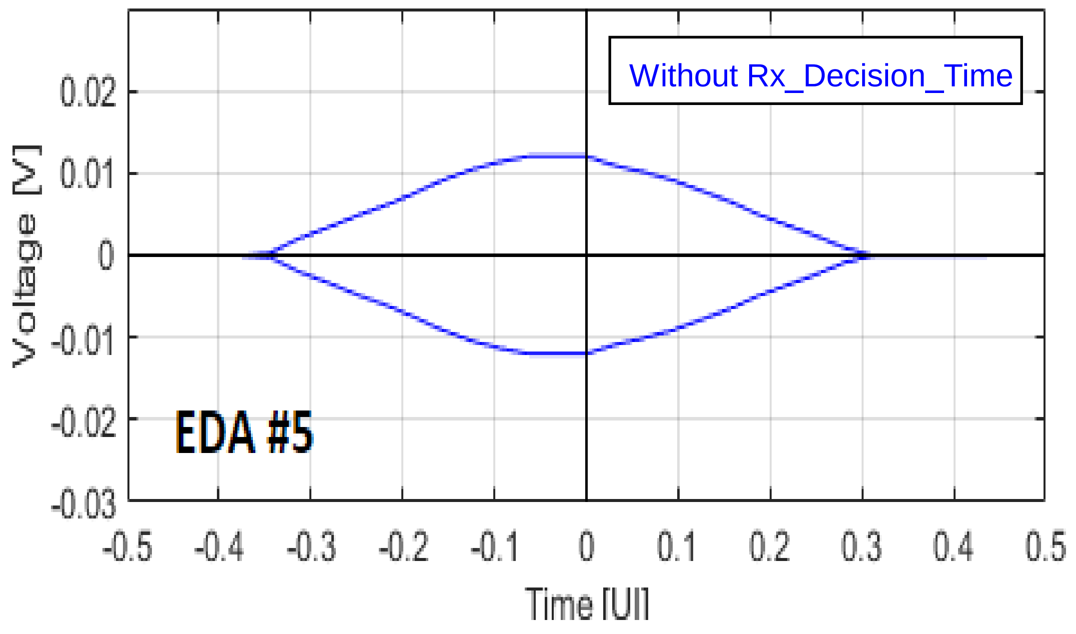


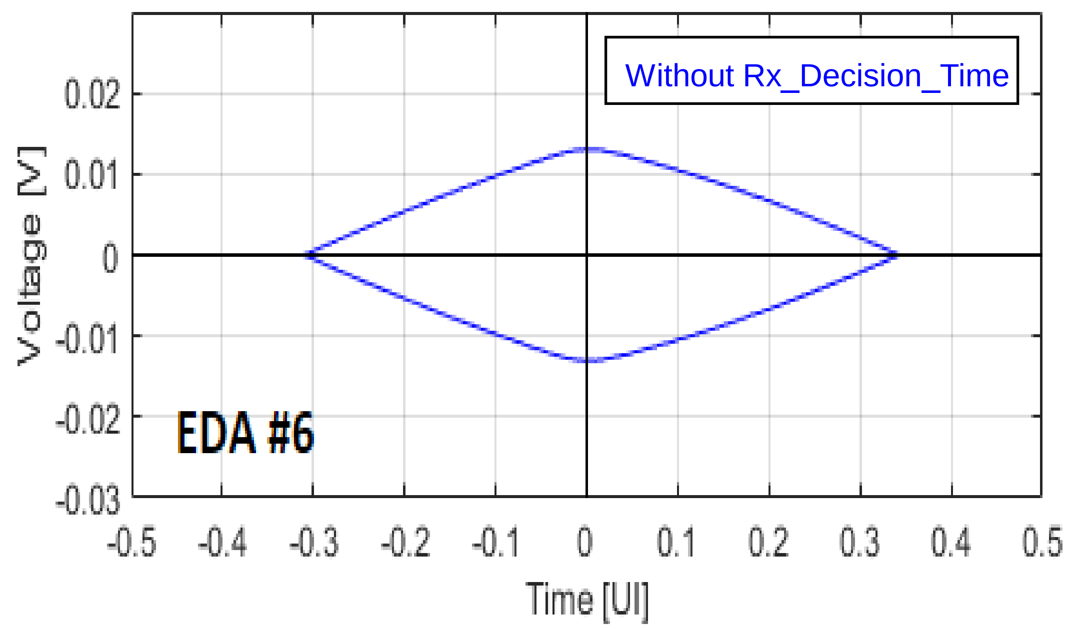


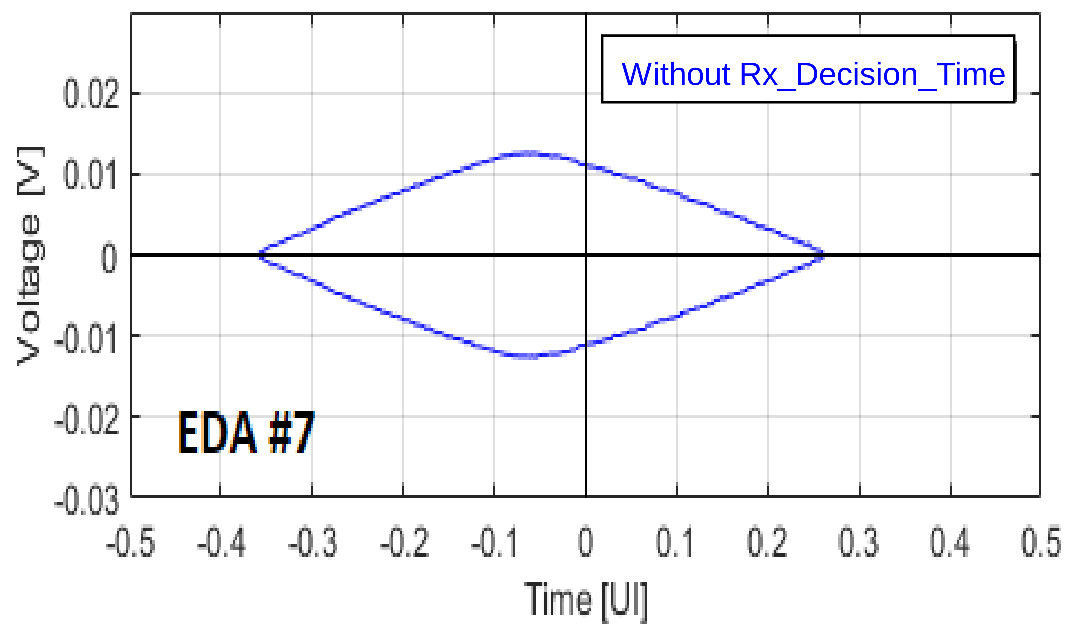


- EDA tools have to guess on sampling point information. About half the EDA tools give a good representation compared to the reference.
- Rx_Decision_Time to be incorporated into IBIS spec 7.1
- EDA #4 appears to be scaling the differential waveform by two and may be dropped when noting the margin difference across the different EDA tools.

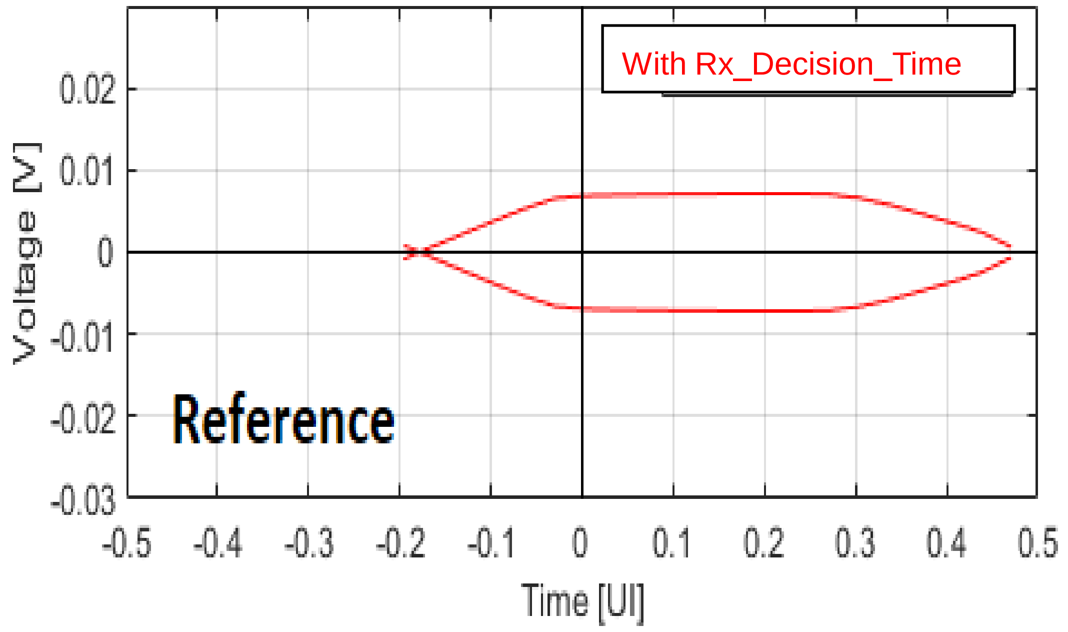


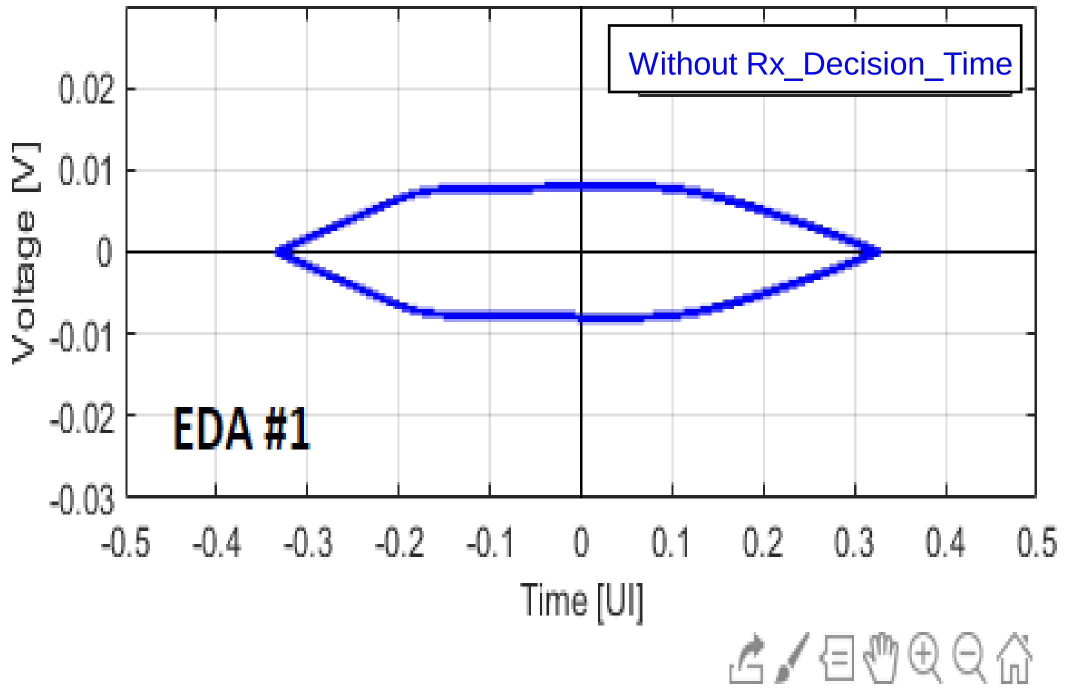


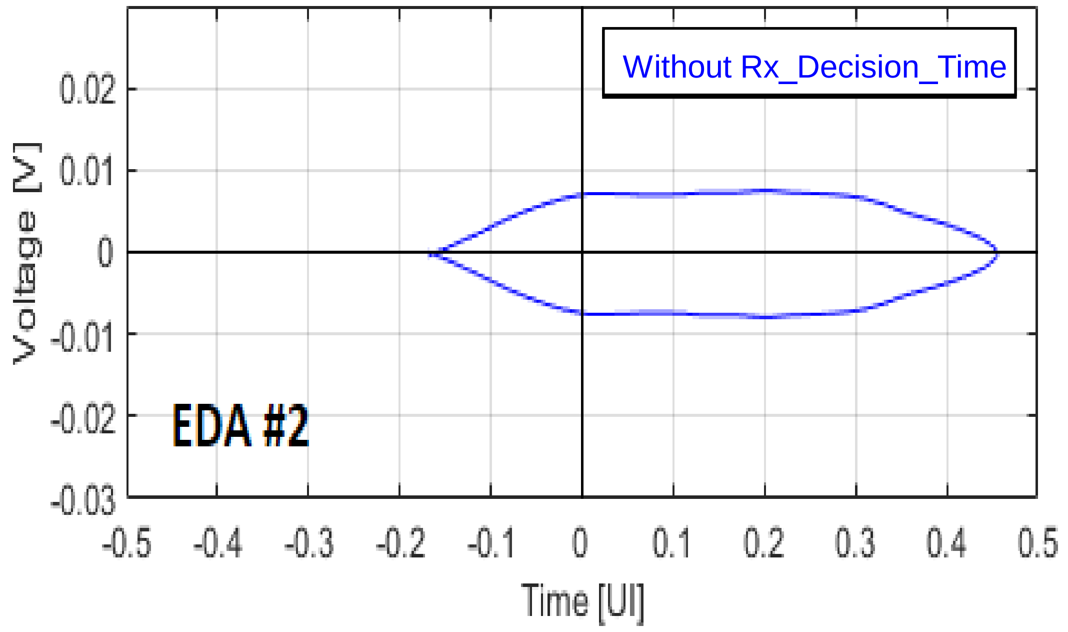


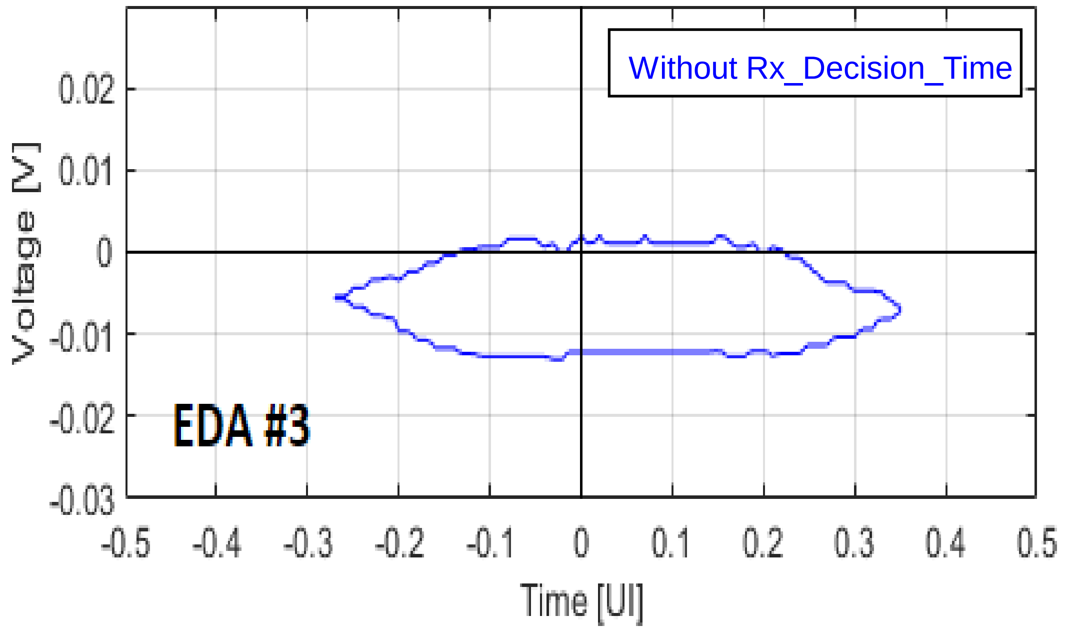


Results. Statistical flow. Mod-MM PD.



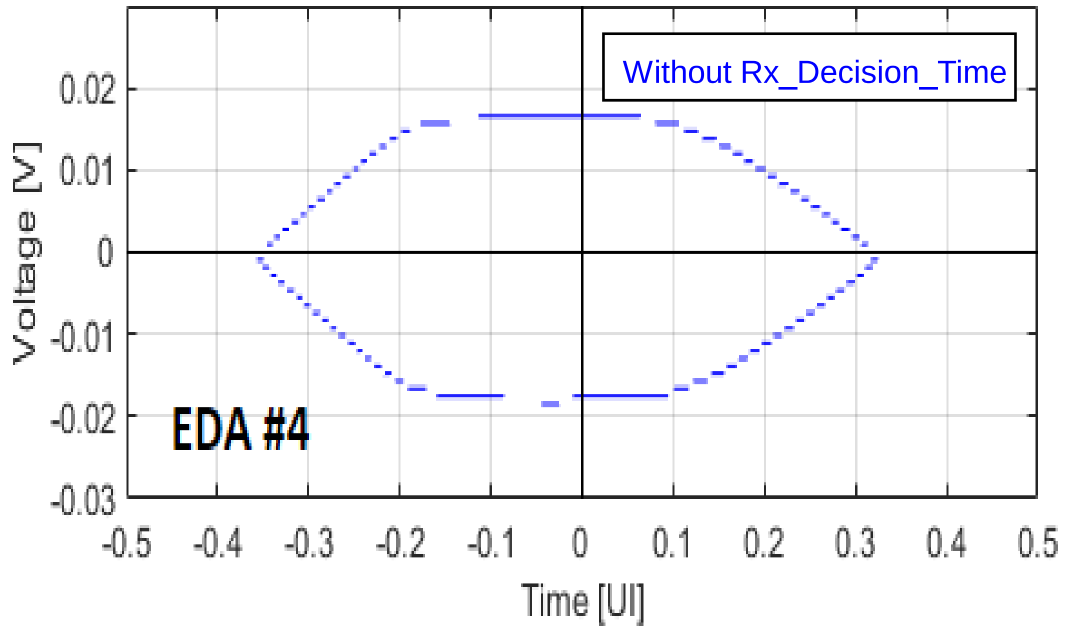


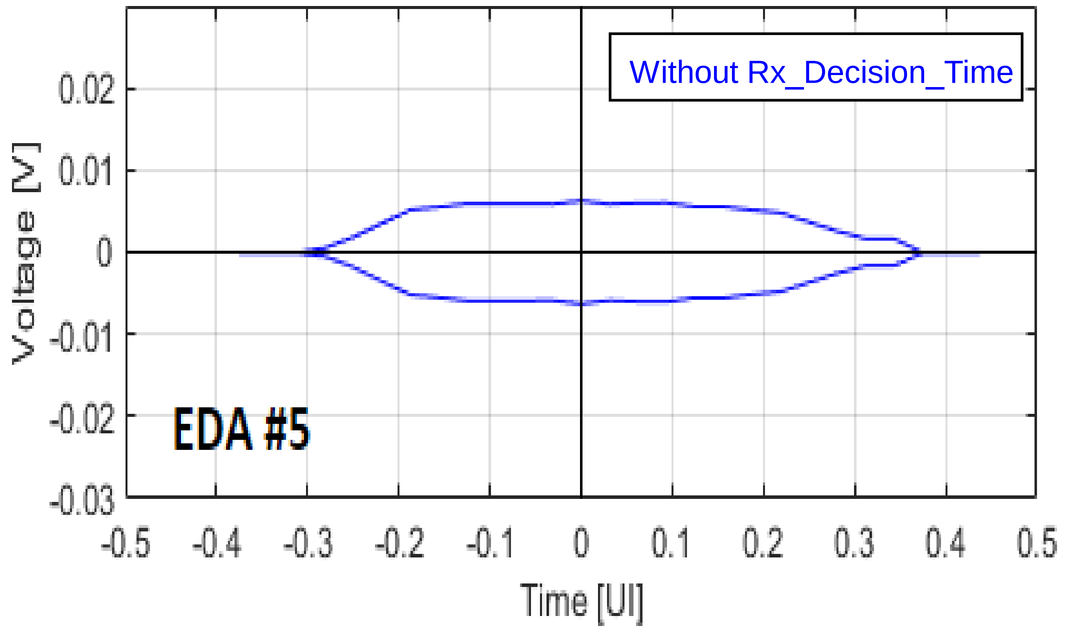


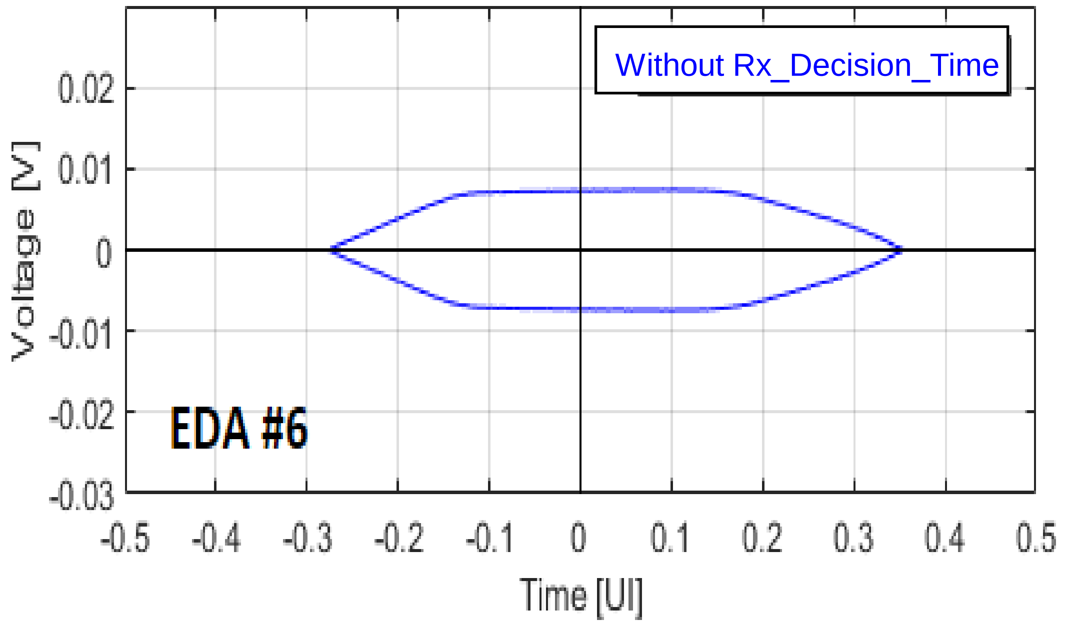


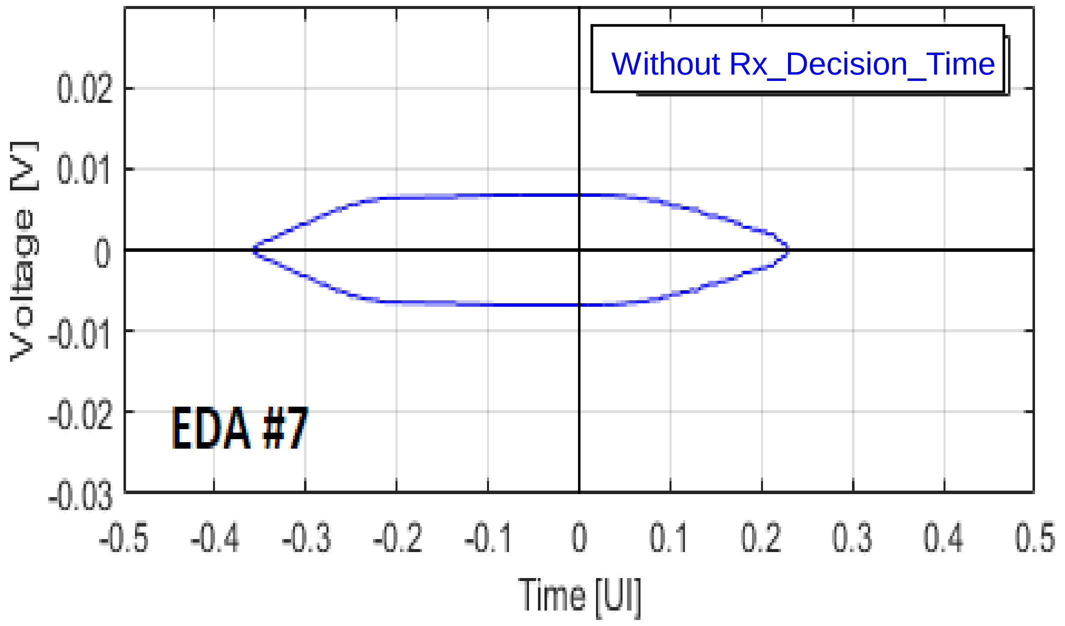
- Observed is that most of the EDA tools are not able to get a similar positioned eye matching the reference in the absence of sampling information from the receiver IBIS-AMI (w/o Rx_Decision_Time).

- EDA #4 appears to be scaling the differential waveform by two and may be dropped when noting the margin difference across the different EDA tools.









Results. Statistical BER contour margin.

Sampling Method	EDA tool	Max. EH [mV]	EW [UI]
MM-PD	Reference	25.6	0.67
	EDA #1	25.9	0.64
	EDA #2	25.9	0.66
	EDA #3	25.4	0.65
	EDA #4	53.3	0.66
	EDA #5	24.0	0.69
	EDA #6	26.0	0.65
	EDA #7	25.1	0.62
Mod-MM PD	Reference	14.2	0.67
	EDA #1	15.4	0.63
	EDA #2	14.9	0.61
	EDA #3	13.2	0.62
	EDA #4	33.2	0.67
	EDA #5	11.8	0.69
	EDA #6	14.7	0.63
	EDA #7	14.4	0.58

EDA incorrectly scaling by two

- The results among the different EDA tools are in the ballpark of the reference.
- Observed is a standard deviation of 1.3mV/0.04UI for the set of seven EDA tools.

2. Importance of sampling information in IBIS

Sampling controls in the IBIS specification

When model does not return sampling information

Statistical flow	Bit-by-bit flow	Comment
Rx_Decision_Time - BIRD 205; status: Accepted - New!	clock_times	Tells the EDA tool about the sampling position
Rx_Clock_PDF	Rx_Clock_PDF	Tells the EDA tool about the probability density function of the recovered clock (PDF)
Rx_Clock_Recovery_Mean - BIRD 206; status: Accepted - Editorial change!	Rx_Clock_Recovery_Mean - BIRD 206; status: Accepted - Editorial change!	Tells the EDA tool about the static offset between the recovered clock and the point half way between the PDF medians of consecutive edge threshold crossing times
Rx_Clock_Recovery_Rj	Rx_Clock_Recovery_Rj	Tells the EDA tool about the standard deviation of a Gaussian phase noise exhibited by the recovered clock
Rx_Clock_Recovery_Dj	Rx_Clock_Recovery_Dj	Tells the EDA tool about the worst-case half the peak to peak variation of the recovered clock
Rx_Clock_Recovery_Sj	Rx_Clock_Recovery_Sj	Tells the EDA tool about the half of the peak to peak variation of a sinusoidal phase noise exhibited by the recovered clock
Rx_Clock_Recovery_DCD	Rx_Clock_Recovery_DCD	Tells the EDA tool about the half of the peak to peak variation of a clock duty cycle distortion exhibited by the recovered clock

Table. Sampling controls in the IBIS specification.

Experimenting with different sampling

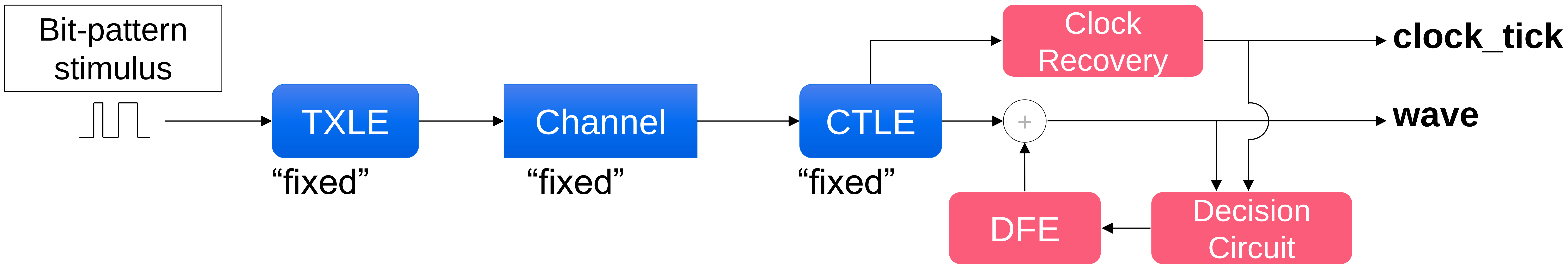
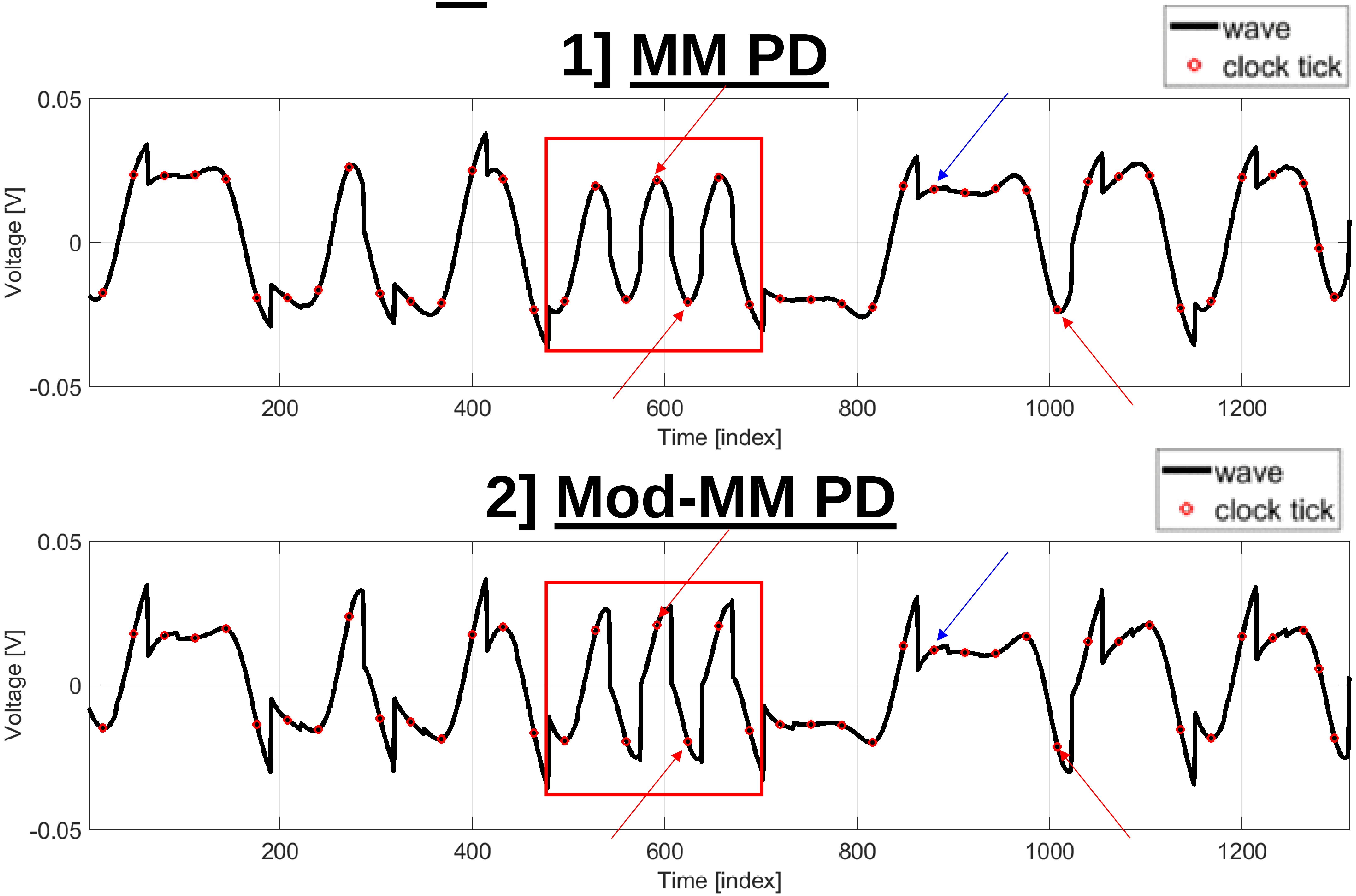


Figure. Block diagram overview.

- "Fixed" TXLE, channel and CTLE.
- Zero-forcing DFE.
- Two different phase detectors (part of clock recovery).
  1. Mueller-Muller (MM) Phase detector
     - Align the sampling clock so that precursor ISI equals to post-cursor ISI.
  2. Modified Mueller-Muller (Mod-MM) Phase detector
     - Modified version of the Mueller-Muller PD where the impact of first pre-cursor (pre-1) is removed.

- The performance of MM PD and Mod-MM PD highly depends on the choice of transmitter equalization (TXLE).
- This work does compare one phase detector with another!

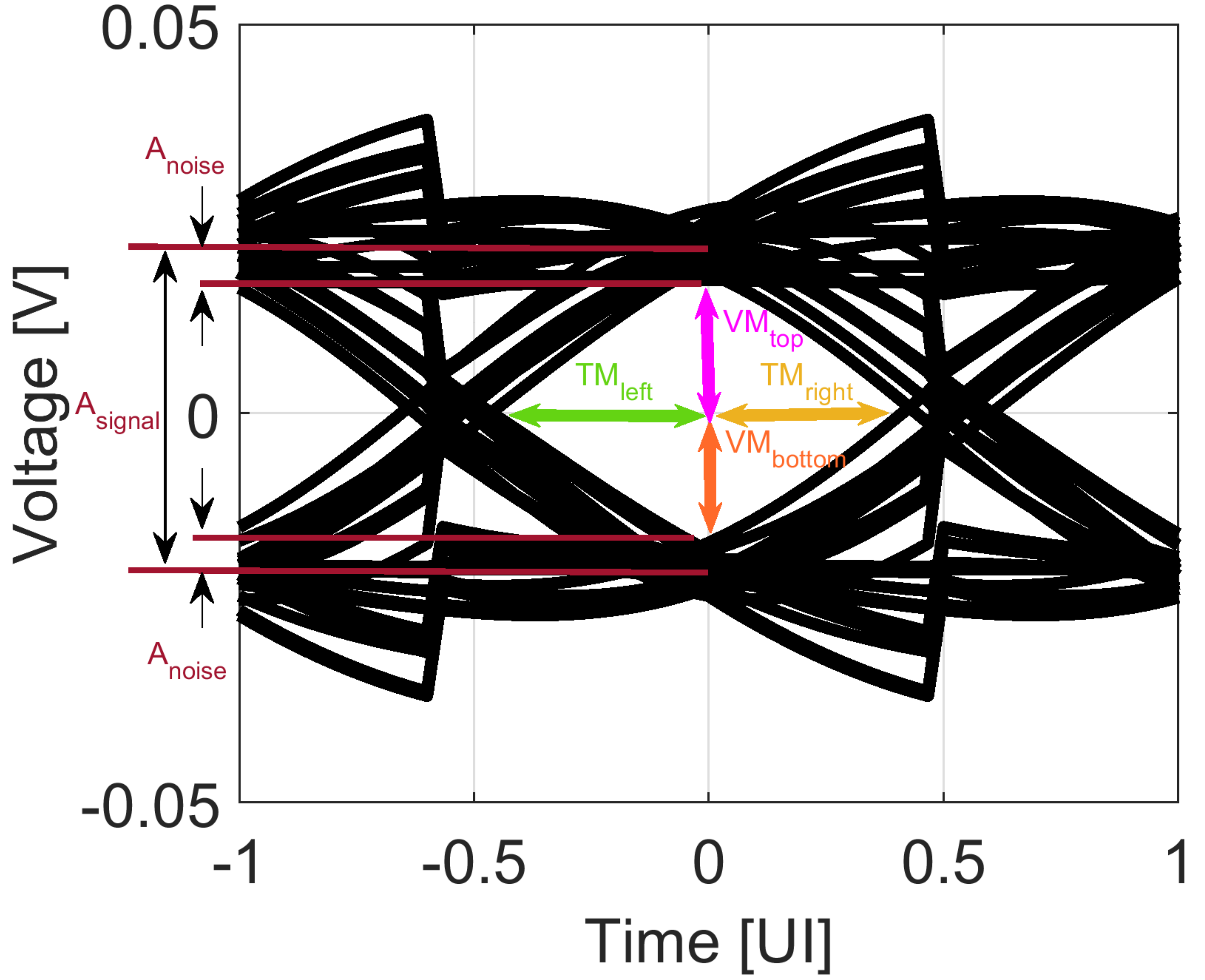
Wave and clock_ticks



- The **transition** and **non-transition bits** may serve to assess the clock tick placement.
- The MM PD places the clock tick close to the optimal sampling.

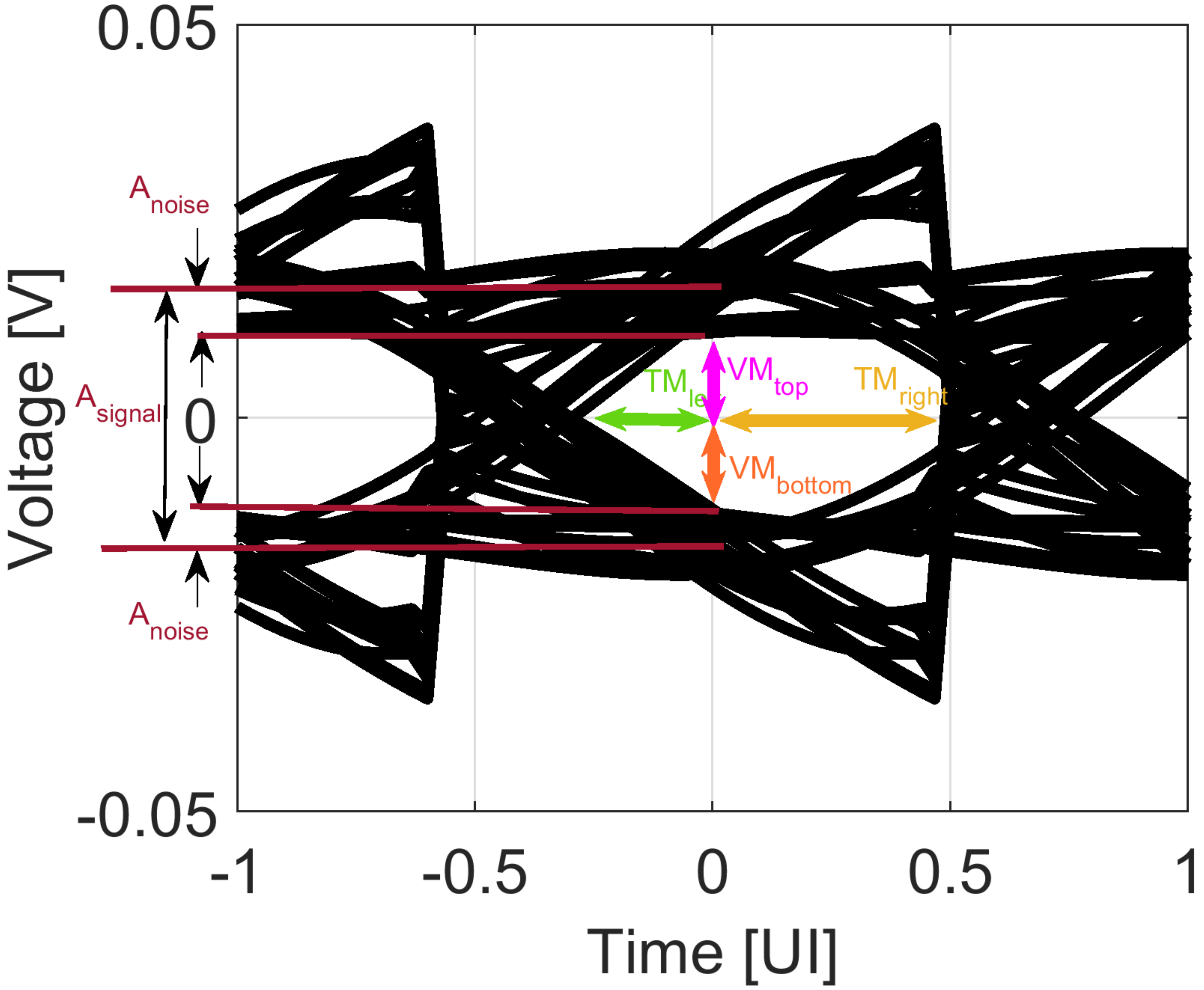
Eye-diagram

1] MM PD



More symmetric eye

2] Mod-MM PD



Less symmetric eye

- The MM PD eye diagram appears qualitatively better than the Mod-MM PD eye because it yields a more <u>symmetrical eye</u> opening as well as <u>more closely grouped trajectories</u>.

Eye-margin

1] MM PD

EH= 33.76mV

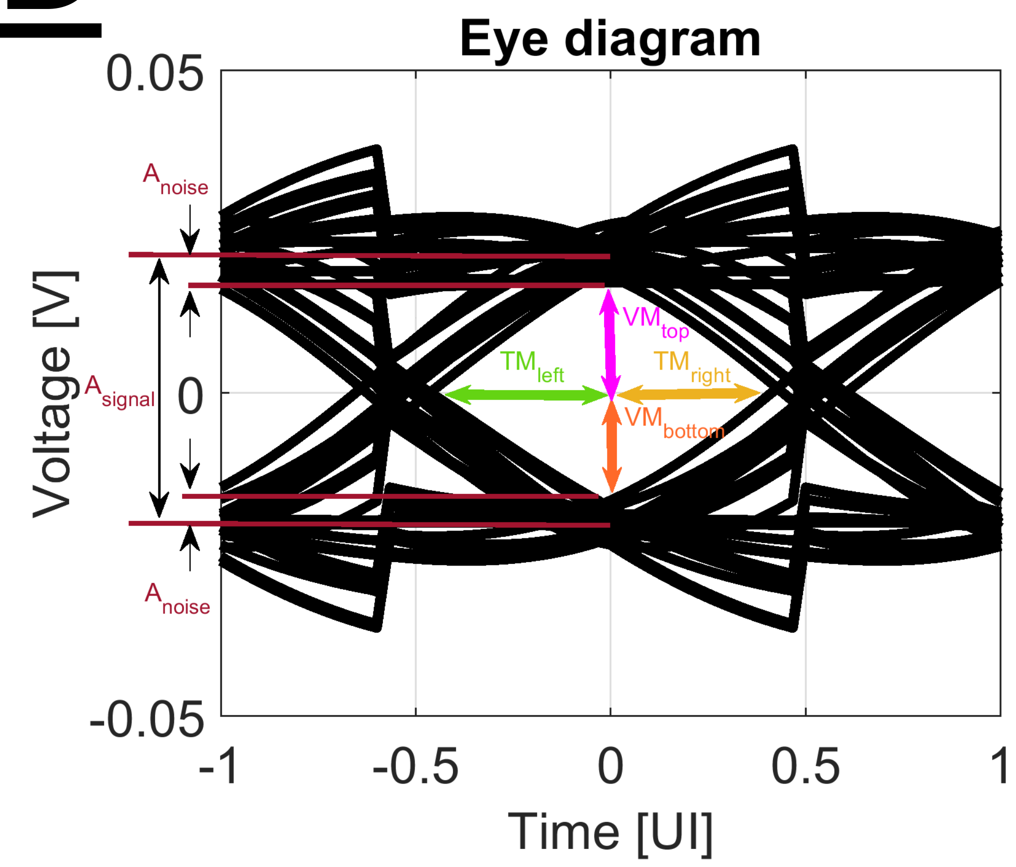
EW= 0.900 UI

VM_{top}: Voltage Margin- top= 17.16 mV

VM_{bottom}: Voltage Margin- bottom= 16.6 mV

TM_{left}: Timing Margin- left= 0.46667 UI

TM_{right}: Timing Margin- right= 0.43333 UI



2] Mod-MM PD

EH= 22.56mV

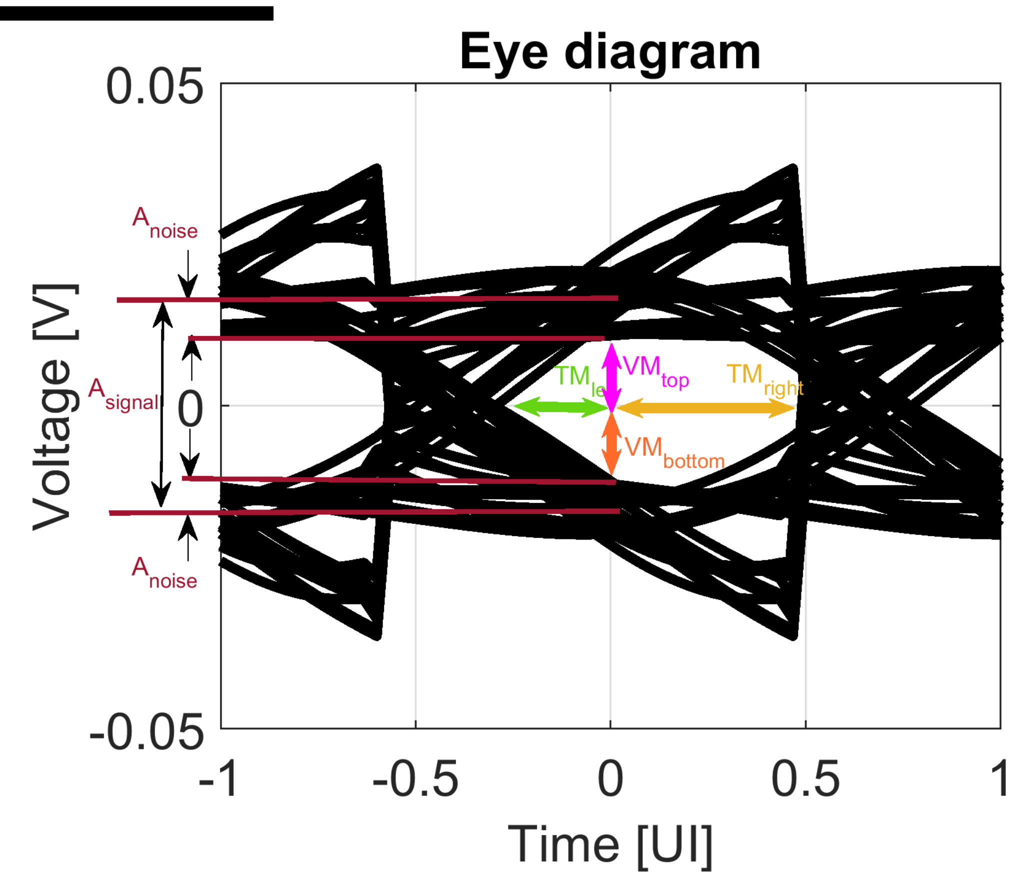
EW= 0.757 UI

VM_{top}: Voltage Margin- top= 10.91 mV

VM_{bottom}: Voltage Margin- bottom= 11.65 mV

TM_{left}: Timing Margin- left= 0.26667 UI

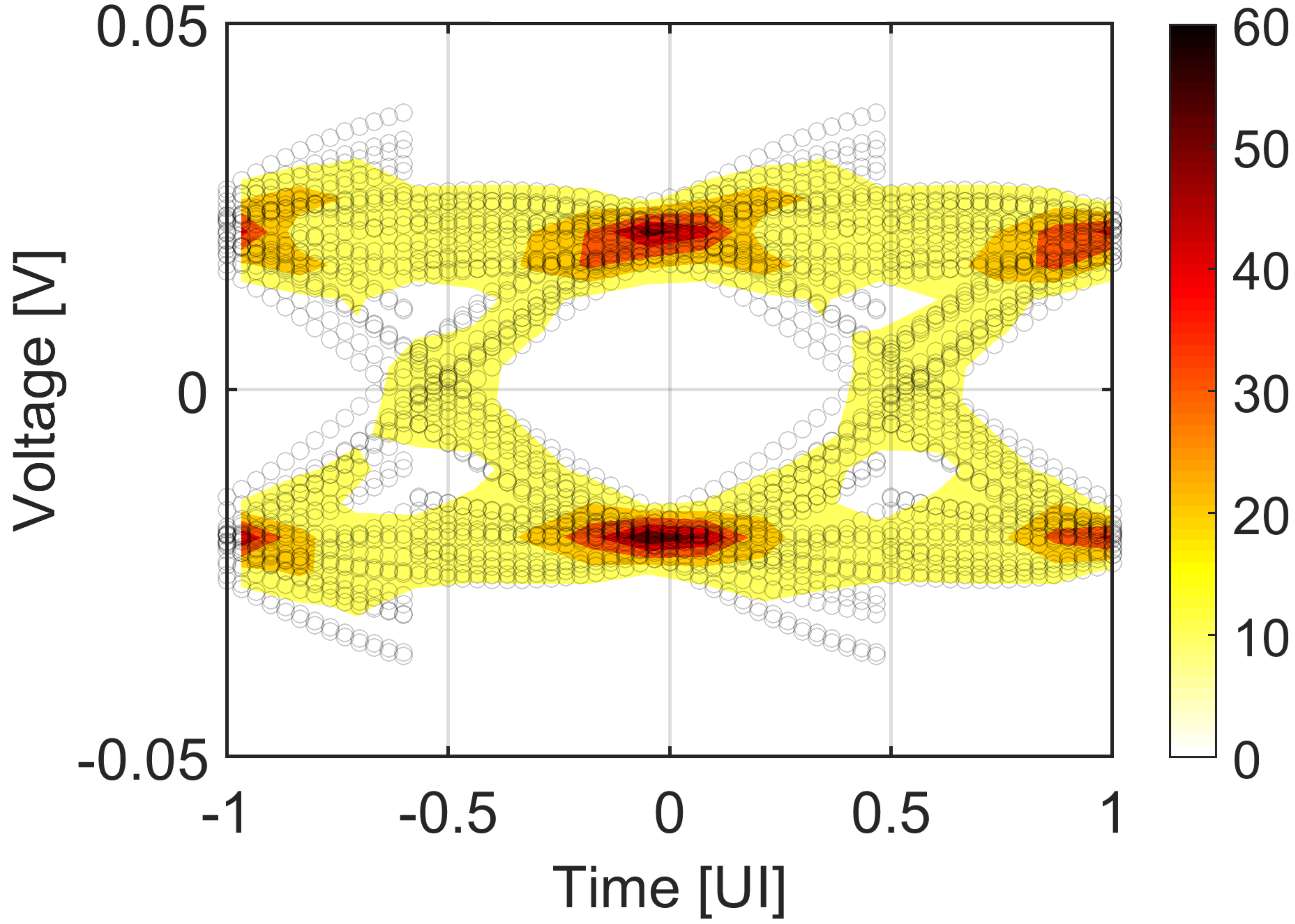
TM_{right}: Timing Margin- right= 0.49000 UI



- Eye height and eye width alone may be misleading, recommend to pay attention to the voltage and timing margin with regards to the sampling point.

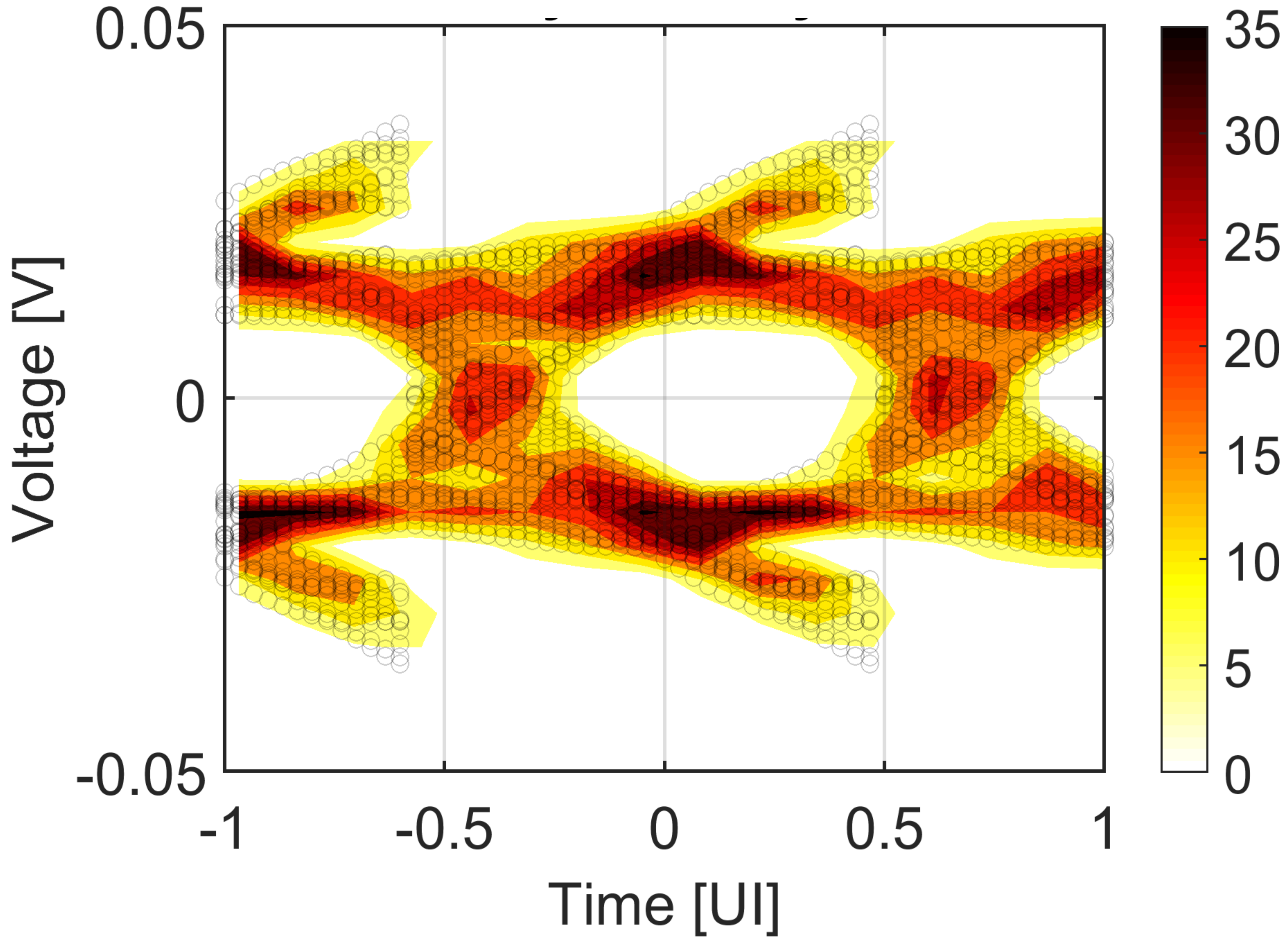
Eye-density

1] MM PD



SNR= 5.05 dB

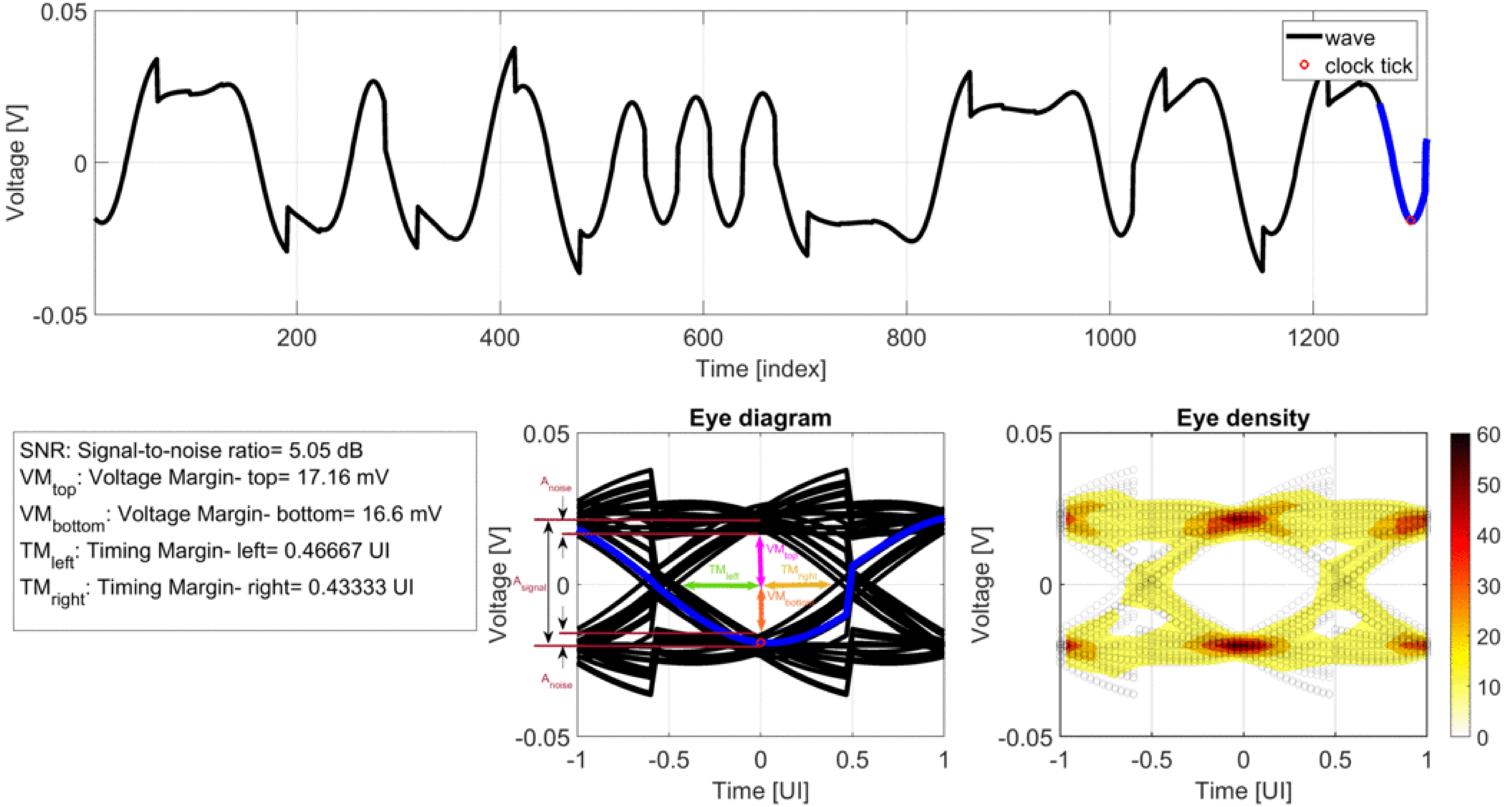
2] Mod-MM PD



SNR= 2.69 dB

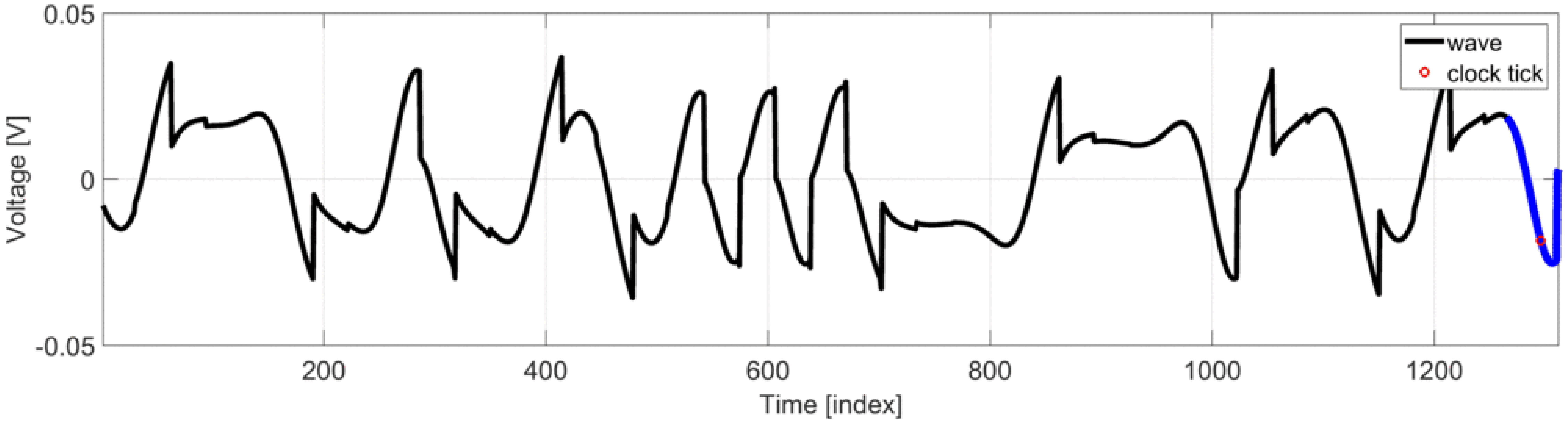
- Eye-density along with SNR are good tools in noting the noise coming due to residual ISI and jitter & may serve as diagnostic tools.

Animation showing eye generation (MM PD)

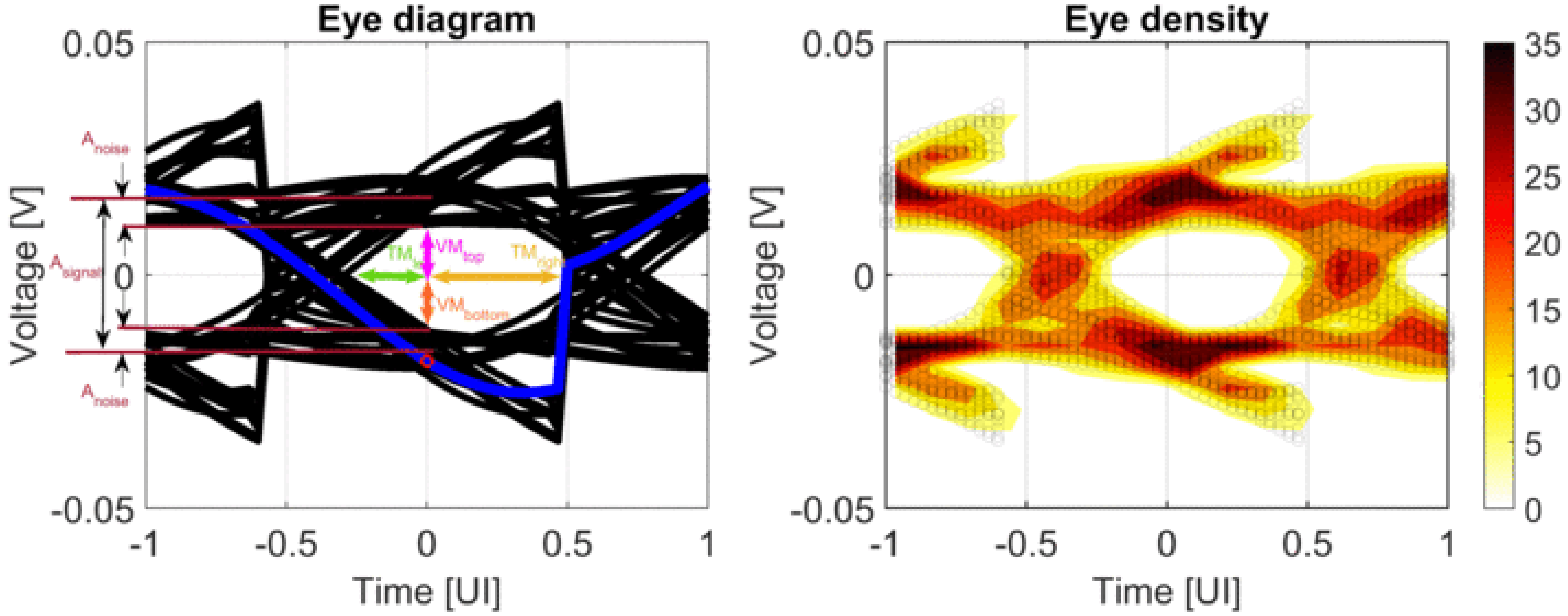


- The eye diagram is built by <u>overlaying</u> successive segments of the wave.
- It provides a simple and useful tool to <u>visualize ISI between data bits</u>.

Animation showing eye generation (Mod-MM PD)



SNR: Signal-to-noise ratio= 2.69 dB
VM_{top}: Voltage Margin- top= 10.91 mV
VM_{bottom}: Voltage Margin- bottom= 11.65 mV
TM_{left}: Timing Margin- left= 0.26667 UI
TM_{right}: Timing Margin- right= 0.49000 UI



- The eye diagram is built by <u>overlaying</u> successive segments of the wave.
- It provides a simple and useful tool to <u>visualize ISI between data bits</u>.

Importance of sampling

- It is important for the EDA tool to use the sampling information given by the receiver IBIS-AMI model.
- Optimal sampling may lead to optimistic margin and not be representative of the real hardware.

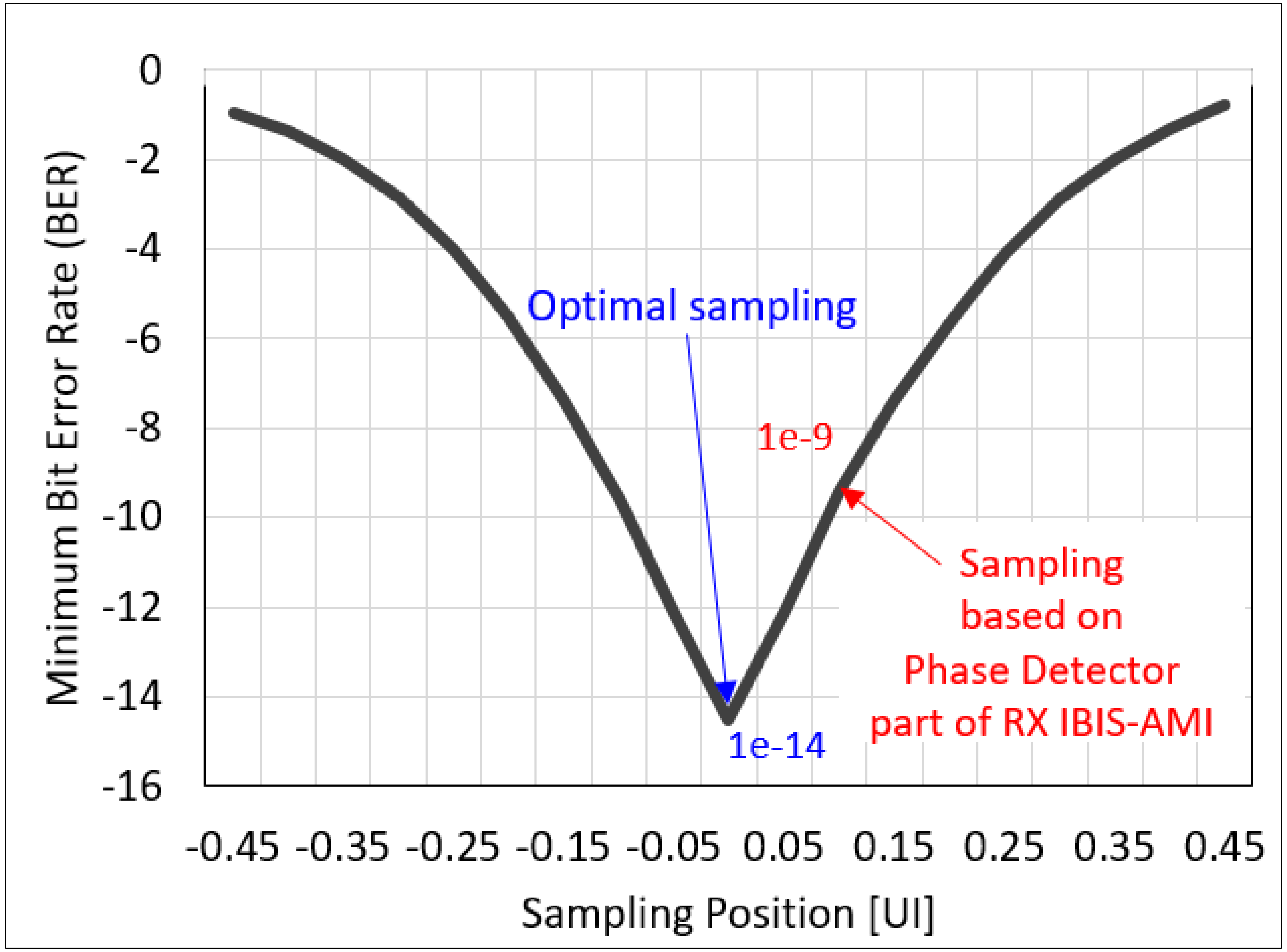


Figure. Impact of sampling on BER.

3. Key takeaway

Key takeaway

- It is important for the EDA tool to use the sampling information given by the receiver IBIS-AMI model.
  - With sampling information returned by the RX IBIS-AMI model to the EDA tool, observed similar eye shape and sampling point across the different EDA tools.
- The IBIS specification calls out 'RECEIVER RECOVERED CLOCK RESERVED PARAMETERS', which may be used to specify characteristics of the receiver's recovered clock.
- Results of bit-by-bit flow show variation in the absolute value of EH and EW at 1e-5 BER for 1e6 bits despite the same wave and clock_times along with bit-pattern.
  - Further work is needed in working with the different EDA tool vendors in looking at the results.

Thanks!

Any questions?

You can find me at: hanseldsilva@achronix.com

BER contour script

```
clc;
clear all;
close all;

%Author: Adam Gregory
%artificial impulse response
%triangle wave with risetime twice as fast as fall time
%followed by random noise with sigma= 0.5 mV combined with decaying
%exponential so noise falls to zero as delay increases

ir= [zeros(1,100) (0.1:0.04:1)*2.5e-3 (1-0.02:-0.02:0)*2.5e-3
randn(1,1000)*5e-4.*exp(-5e-3*[1:1000])];
UI= 32;
num_bits= 1e3;
pulse= filter(ones(1,UI),1,ir);
%set [-0.5 0.5] bit pattern (no DC)
bit_pattern= round(rand(1,num_bits))-0.5;
%set bit pattern to UI increments to conv with pulse
bit_pattern_UI(1:UI:UI*num_bits)= bit_pattern;
wave= conv(pulse,bit_pattern_UI);
%easy sample at peak of pulse
[tmp,ts]= max(pulse);
%clock times are UI increments of ts
clock_times= (1:UI:length(bit_pattern)*UI)+ts-1;
```

```
%eye contour
half_UI= ceil(UI/2);
%eye contour must know whether is a 1 or 0
%if the pattern was unknown, this can also be discovered by checking if
the
%sample voltage [wave(clock_times)] is positive/negative.  However that
%only works for open eye.
sv= wave(clock_times);
ones_idx= find(bit_pattern>0);
zeros_idx= find(bit_pattern<0);
for j=1:UI
    sample_vector= wave(clock_times-half_UI+j);
    %1st column=1 contour
    %2nd column= 0 contour
    eye_contour(j,1)= min(sample_vector(ones_idx));
    eye_contour(j,2)= max(sample_vector(zeros_idx));
end
%full eye density
for j=1:UI
    sample_vector= wave(clock_times-half_UI+j);
    eye_density(1:num_bits,j)= sample_vector;
end

figure;
plot(eye_contour);
figure;
plot(eye_density');
```