# Characterizing and Modeling of a Linear CTE

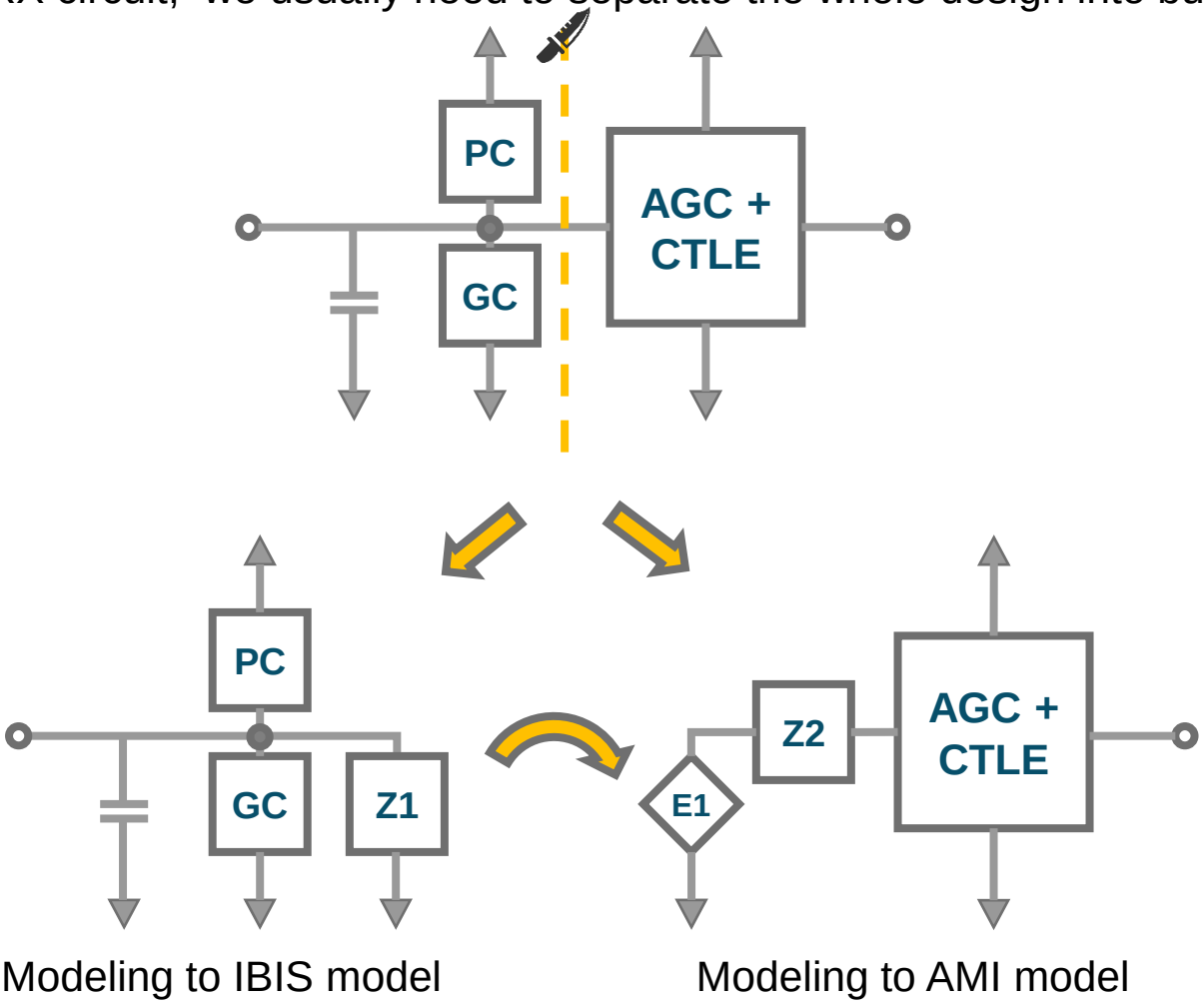
***Skipper Liang***

Asian IBIS Summit
Taipei, ROC
November 15, 2017

cadence®

# To Divide a RX Circuit:

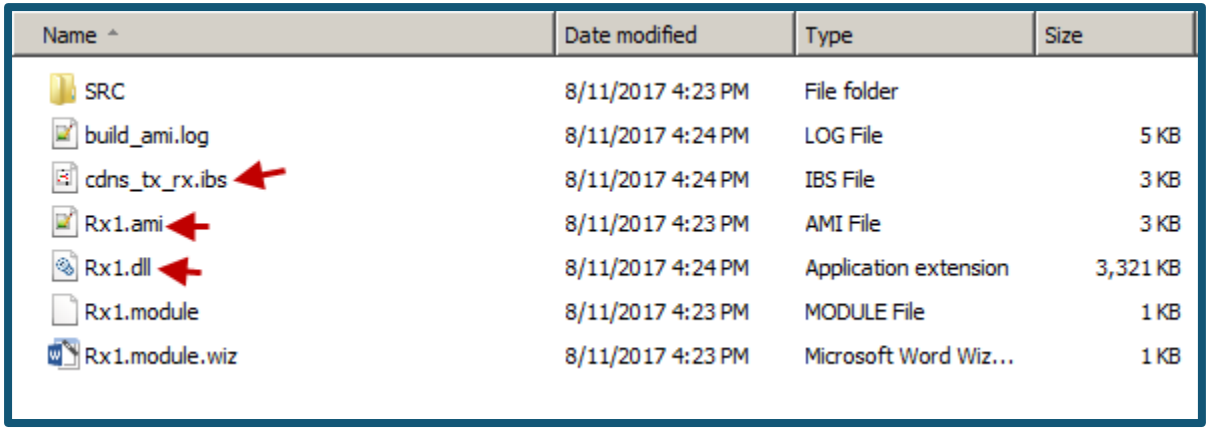
For modeling a RX circuit, we usually need to separate the whole design into buffer part and algorithm part:




1. Question 1: What's the value of **Z1** and **Z2**?
2. Question 2: What if the whole design is described in **an encrypted netlist**?

# IBIS model

- generated from AMI generation tools



| Name | Date modified | Type | Size |
|---|---|---|---|
| SRC | 8/11/2017 4:23 PM | File folder | |
| build_ami.log | 8/11/2017 4:24 PM | LOG File | 5 KB |
| cdns_tx_rx.ibs | 8/11/2017 4:24 PM | IBS File | 3 KB |
| Rx1.ami | 8/11/2017 4:23 PM | AMI File | 3 KB |
| Rx1.dll | 8/11/2017 4:24 PM | Application extension | 3,321 KB |
| Rx1.module | 8/11/2017 4:23 PM | MODULE File | 1 KB |
| Rx1.module.wiz | 8/11/2017 4:23 PM | Microsoft Word Wiz... | 1 KB |

Many AMI generation tools will generate an IBIS model along with the AMI models generation:

1. Question 1: Can we use this IBIS model? **YES**
2. Question 2: If yes, is there any requirement of the circuit while modeling this circuit in this way?

   **The circuit should be a RX one composed of linear components.**

3. Question 3: If my circuit could meet the requirement list above, how to do the modeling?

   **Detailed in the following pages**

# Thevenin's Theorem

## Thévenin's theorem

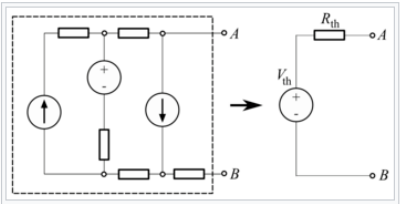
From Wikipedia, the free encyclopedia

As originally stated in terms of DC resistive circuits only, **Thévenin's theorem** holds that:

- Any linear electrical network with voltage and current sources and only resistances can be replaced at terminals A-B by an equivalent voltage source $V_{th}$ in series connection with an equivalent resistance $R_{th}$.
- The equivalent voltage $V_{th}$ is the voltage obtained at terminals A-B of the network with terminals A-B open circuited.
- The equivalent resistance $R_{th}$ is the resistance that the circuit between terminals A and B would have if all ideal voltage sources in the circuit were replaced by a short circuit and all ideal current sources were replaced by an open circuit.
- If terminals A and B are connected to one another, the current flowing from A to B will be $V_{th}/R_{th}$. This means that $R_{th}$ could alternatively be calculated as $V_{th}$ divided by the short-circuit current between A and B when they are connected together.

In circuit theory terms, the theorem allows any one-port network to be reduced to a single voltage source and a single impedance.

The theorem also applies to frequency domain AC circuits consisting of reactive and resistive impedances. It means the theorem applies for AC in an exactly same way to DC except that resistances are generalized to impedances.




Any black box containing resistances only and voltage and current sources can be replaced by a Thévenin equivalent circuit consisting of an equivalent voltage source in series connection with an equivalent resistance.

In short:

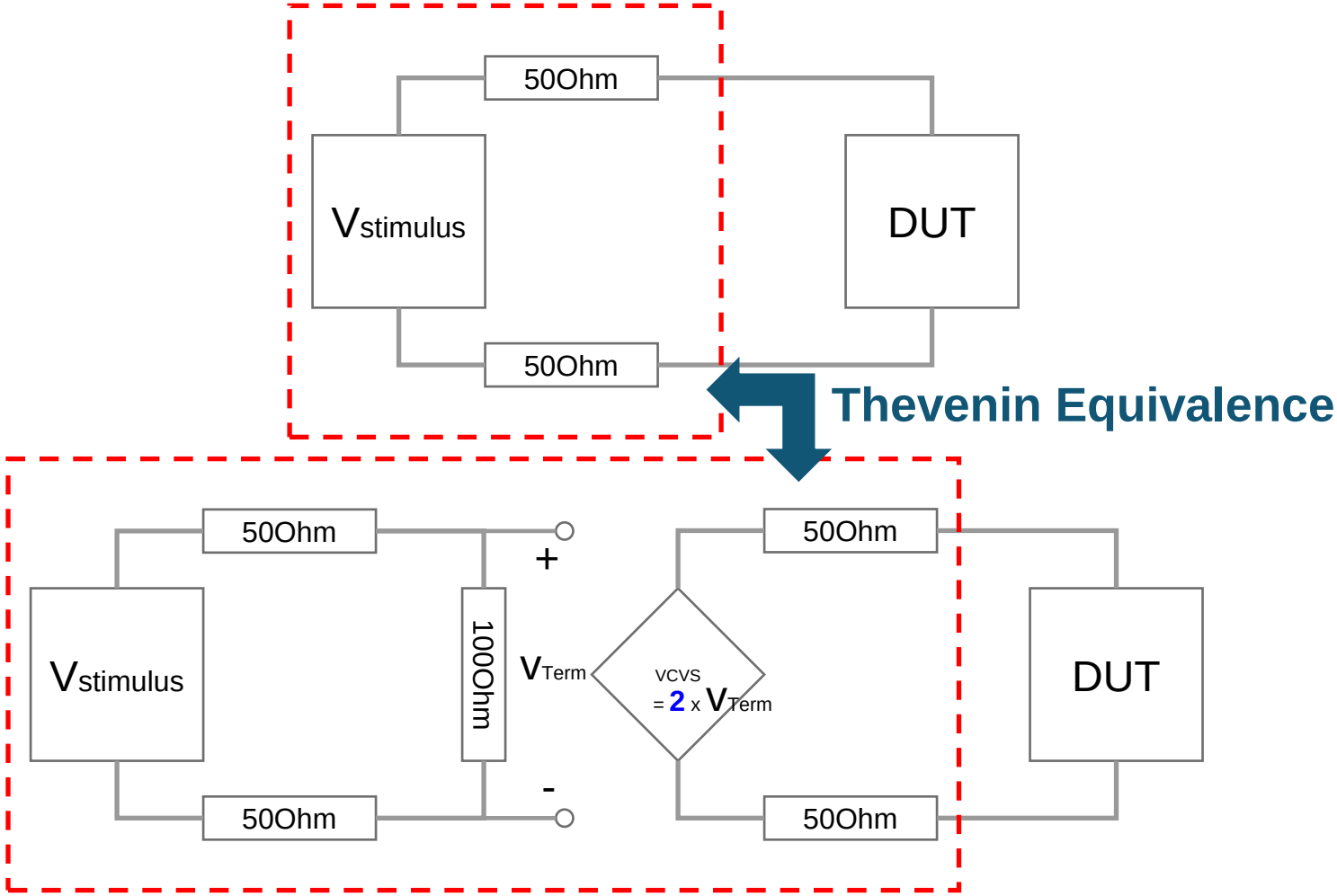
$V_{TH}$ = The voltage across the Port – "node A and B" while treating the Port – "node A and B" as **OPEN**

$I_{SC}$ = The current goes into node A and leaves node B while treating the Port – "node A and B" as **SHORT**

$R_{TH} = V_{TH}/I_{SC}$

# Thevenin Equivalence

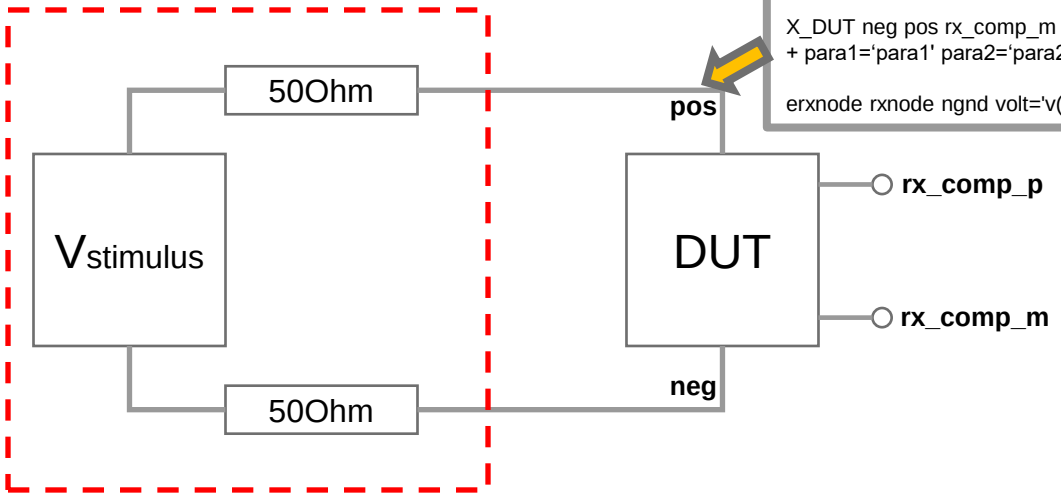
Typically, an RX test topology



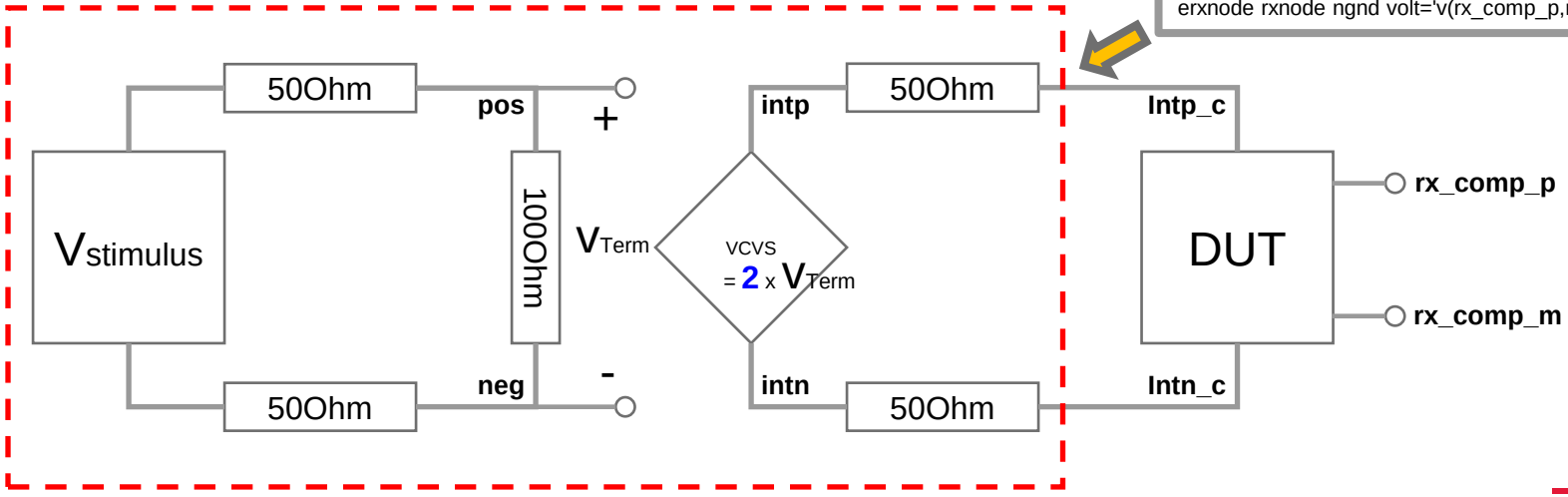
# Thevenin Equivalence(Cont'd)

You can do a small experiment – the following 2 netlist will give the same waveform result at the EQ's output

```
.INCLUDE './rx_model.sp'
.PARAM para1=0 para2=0

X_DUT neg pos rx_comp_m rx_comp_p ref_eq
+ para1='para1' para2='para2'

erxnode rxnode ngnd volt='v(rx_comp_p,rx_comp_m)'
```

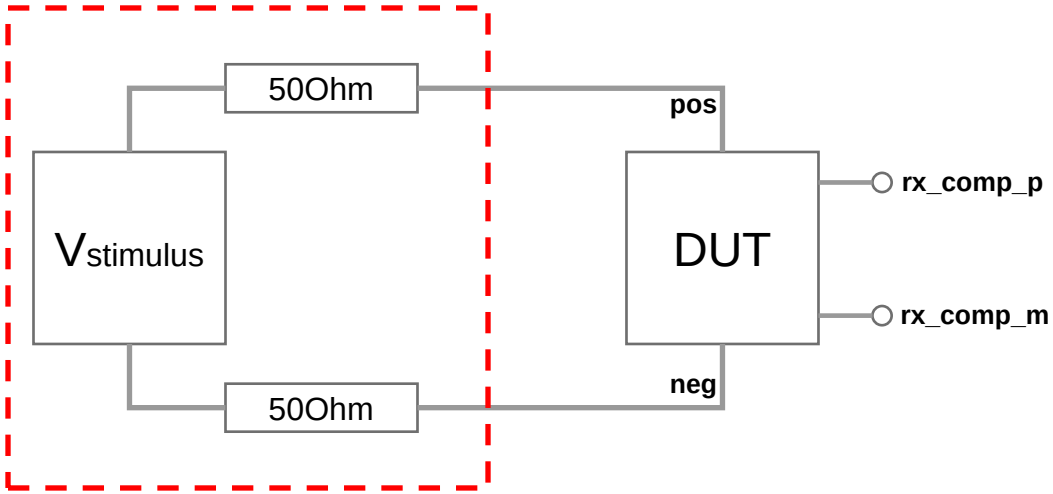


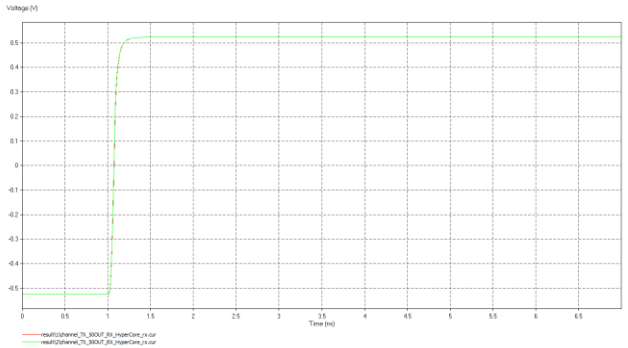

```
R100 pos neg 100

E1 intp intn volt='2*v(pos,neg)'

R50p intp intp_c 50
R50n intn intn_c 50

.INCLUDE './rx_model.sp'
.PARAM para1=0 para2=0

X_DUT intn_c intp_c rx_comp_m rx_comp_p ref_eq
+ para1='para1' para2='para2'

erxnode rxnode ngnd volt='v(rx_comp_p,rx_comp_m)'
```

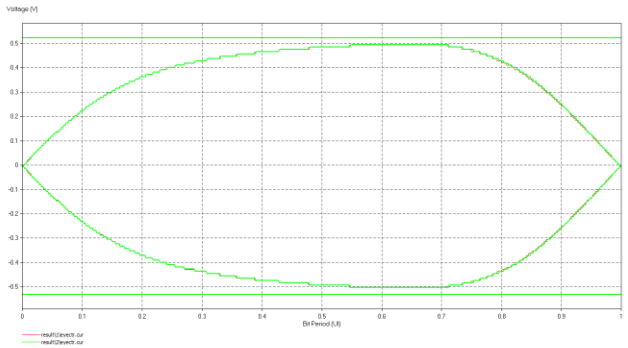


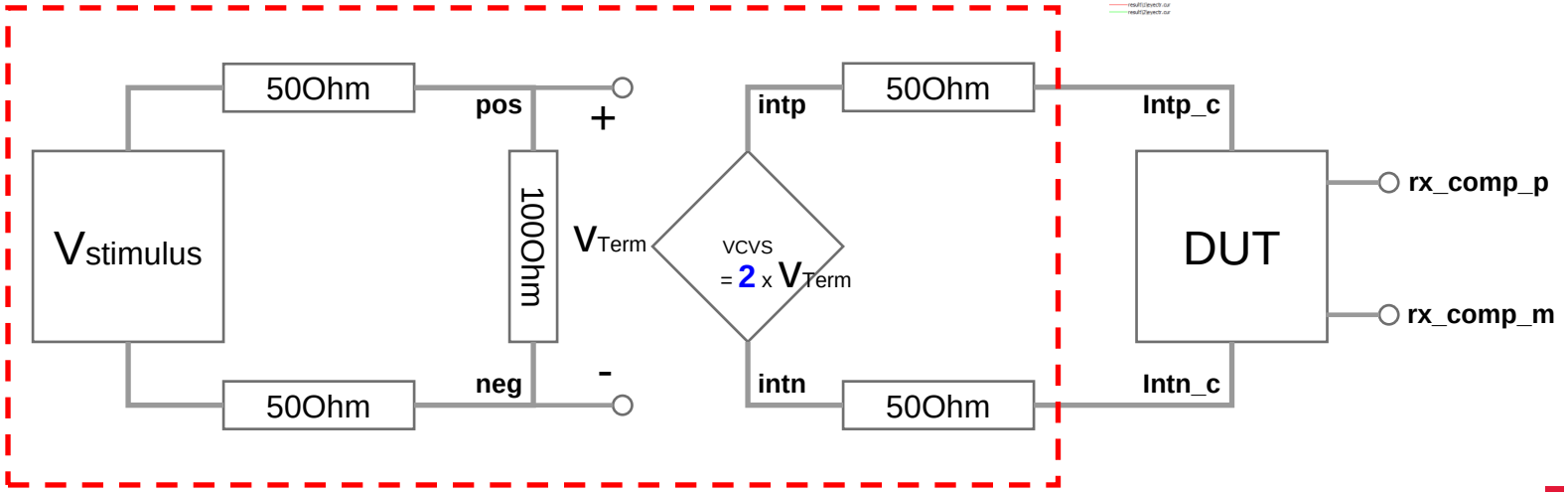
# Thevenin Equivalence(Cont'd)

Hard to tell the difference

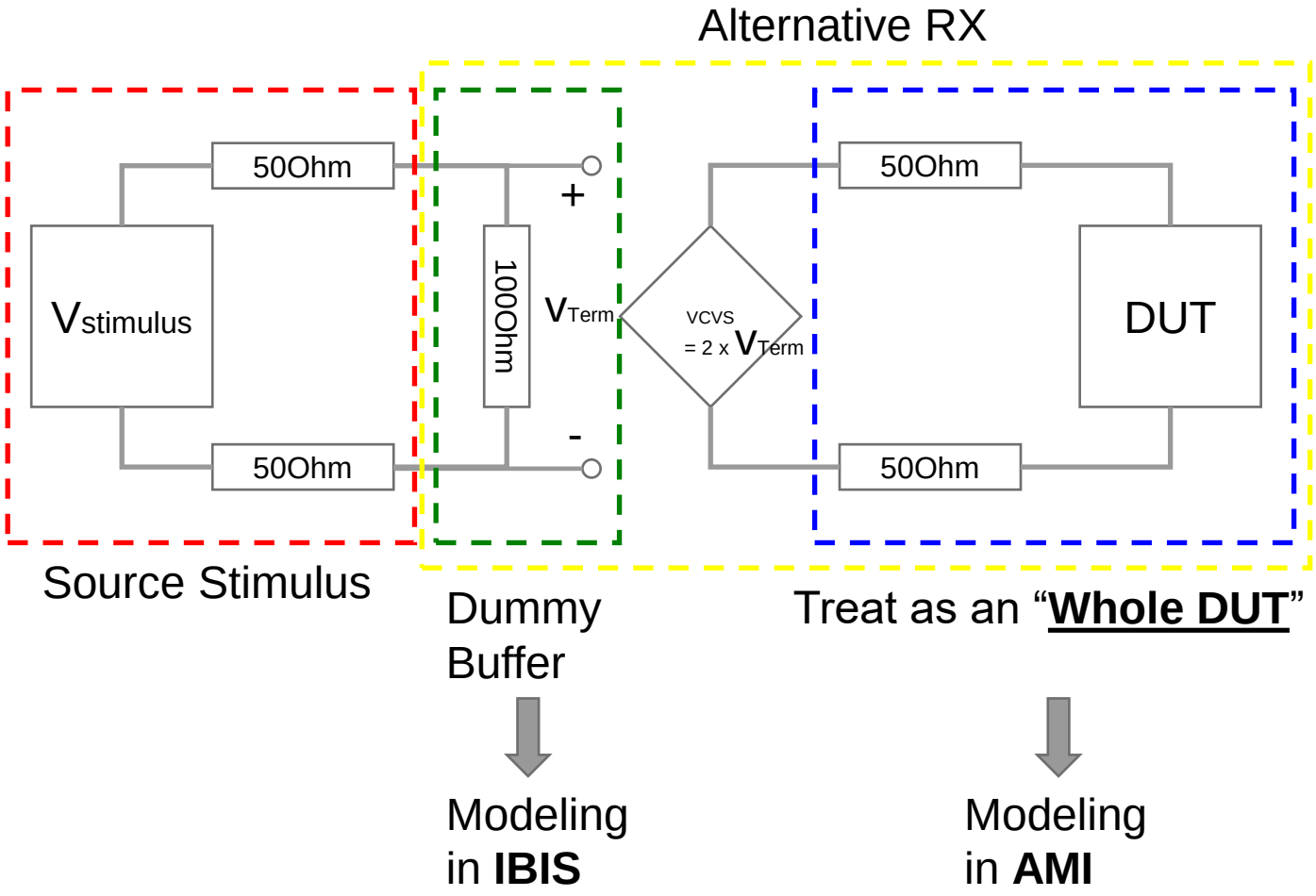






# Alternative RX

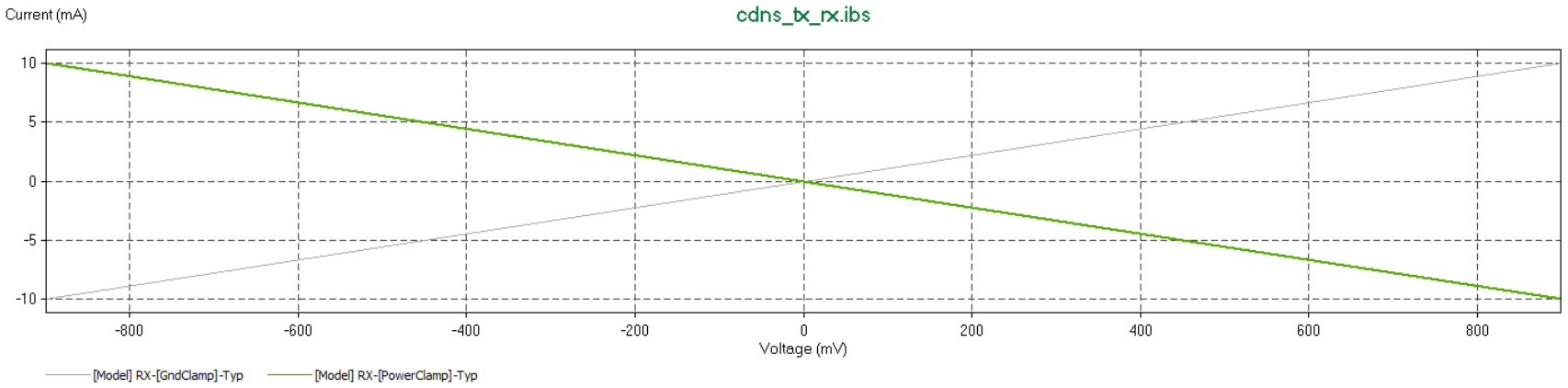


# IBIS of the Dummy Buffer

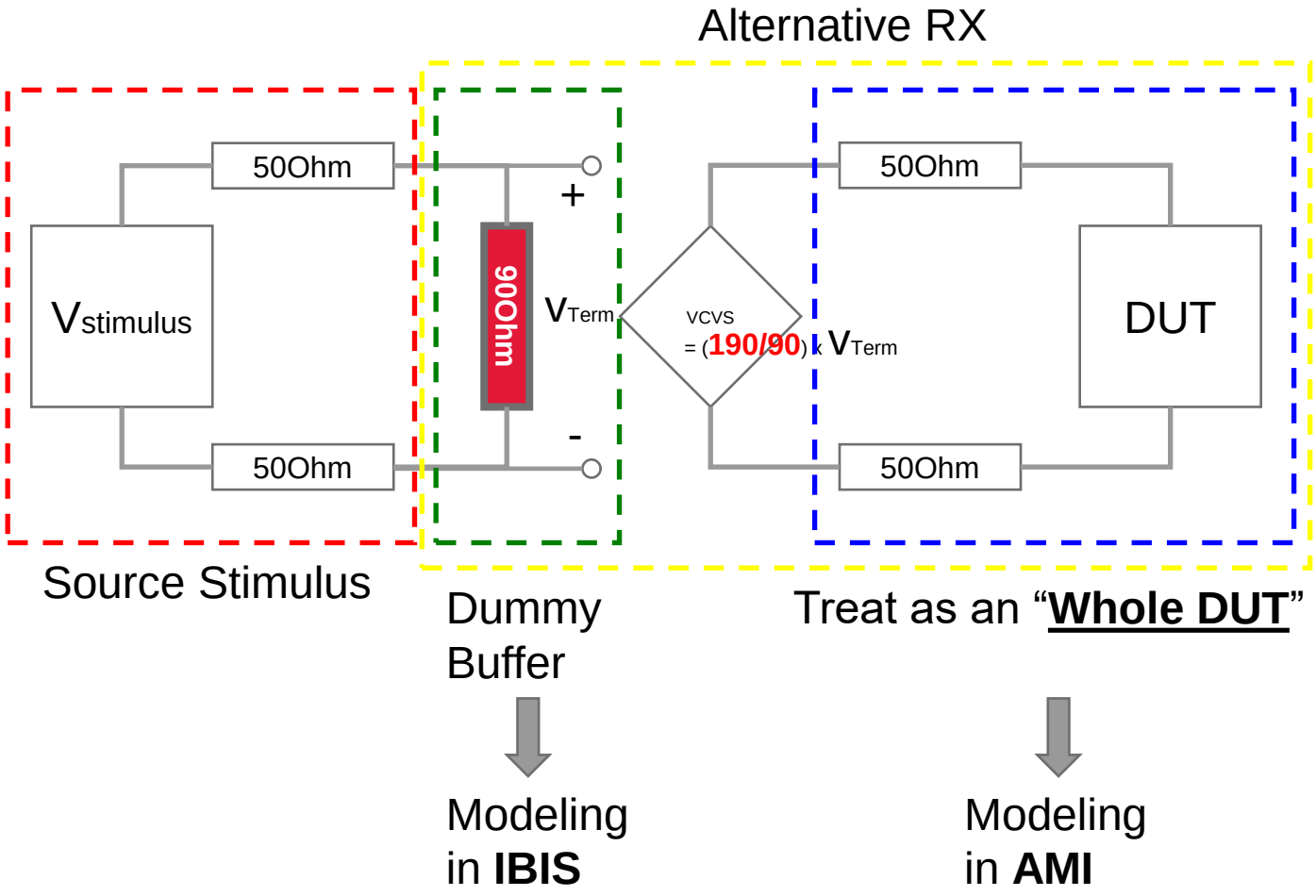
Check the RX IBIS model, which is generated along with the AMI model, to see if it's a 100Ohm terminator between positive node and negative node:

| Name | Date modified | Type | Size |
|---|---|---|---|
| SRC | 3/17/2017 3:30 PM | File folder | |
| build_ami | 3/17/2017 3:30 PM | LOG File | 5 KB |
| cdns_tx_rx | 3/17/2017 3:30 PM | IBS File | 3 KB |

For example, some AMI generation tools generate RX IBIS model with 90Ohm terminator between positive node and negative node – tell from the [Power Clamp] and [Ground Clamp]:

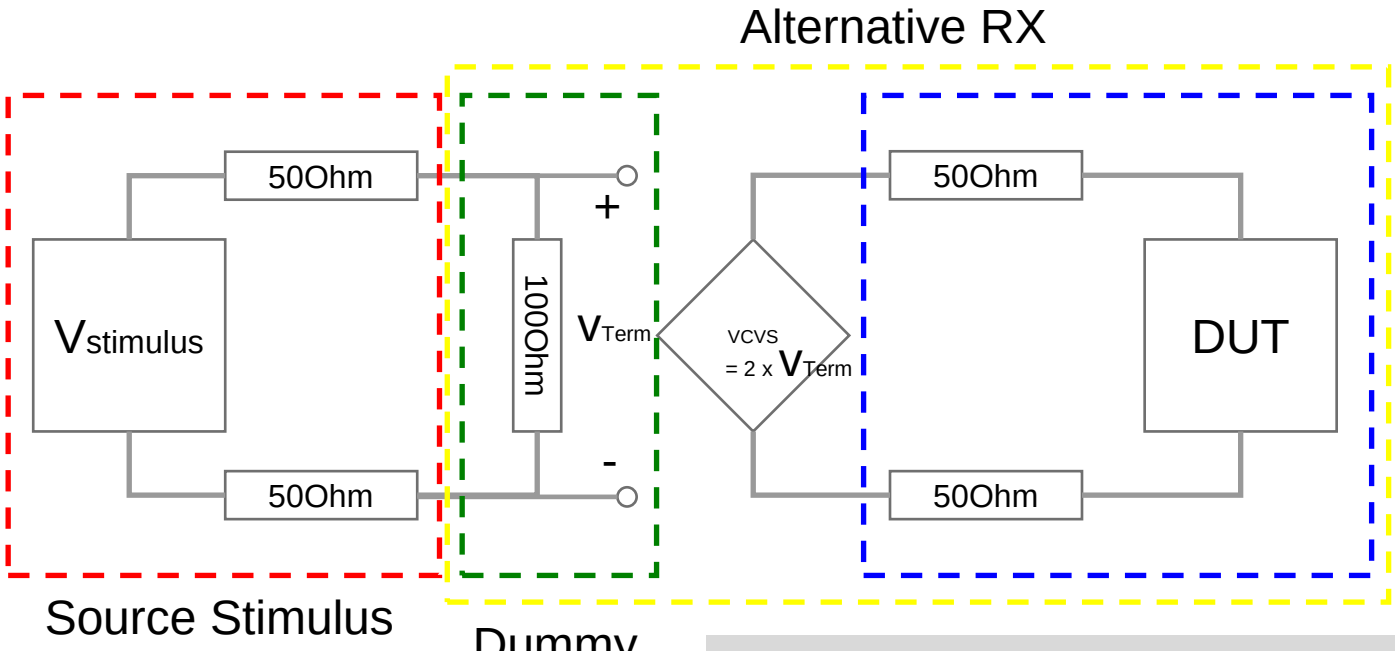


# Alternative RX for a 90Ohm terminator




## Or……

# IBIS model of 100Ohm terminator




And now, you don't need to model the IBIS part or use the non-100Ohm terminator IBIS model. You just use the files here attached:


# IBIS model of 100Ohm terminator (Cont'd)

The content of these files:

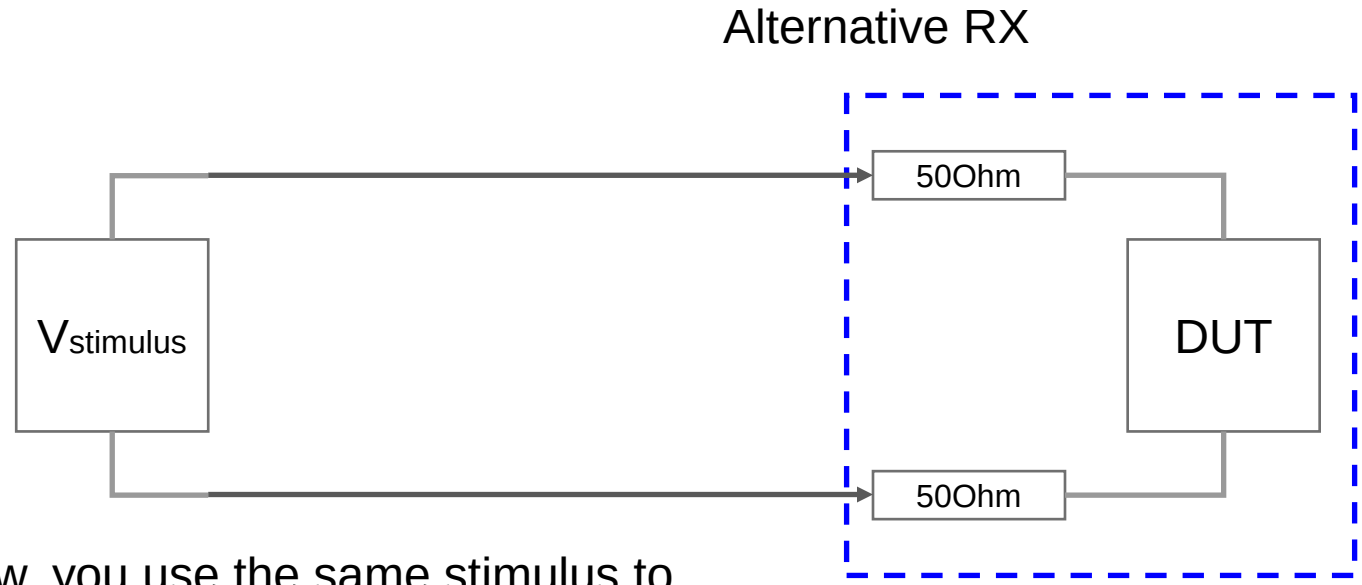
rx100term.ibs

```
[Model] RX
Model_type Input_diff
[Voltage Range]          1.000V            1.0V              1.0V
[Ramp]
| variable       typ                  min                  max
dV/dt_r 0.120/0.001n        0.120/0.001n        0.120/0.001n
dV/dt_f 0.120/0.001n        0.120/0.001n        0.120/0.001n
R_load = 50.000
|
[External Model]
Language SPICE
|
| Corner  corner_name  file_name  circuit_name (.subckt name)
Corner    Typ          R_100.ckt   R_100
Corner    Min          R_100.ckt   R_100
Corner    Max          R_100.ckt   R_100
|
| Ports List of port names (in same order as in SPICE)
Ports A_puref A_pdref A_signal_pos A_signal_neg
|
| A_to_D d_port port1 port2 vlow vhigh corner_name
A_to_D D_receive A_signal_pos A_signal_neg -0.0 0.1 Typ
A_to_D D_receive A_signal_pos A_signal_neg -0.0 0.1 Min
A_to_D D_receive A_signal_pos A_signal_neg -0.0 0.1 Max
|
[End External Model]
[Algorithmic Model]
Executable Windows_VisualStudio_64 RX_HyperCore.dll RX_HyperCore.ami
[End Algorithmic Model]
|
[End]
```

R_100.ckt

```
1  .subckt R_100 DVDD DVSS pos neg
2
3  Rp pos gnd 1e+15
4  Rn neg gnd 1e+15
5  Rdiff pos neg 100
6
7  .ends
```

# Characterizing:




Now, you use the same stimulus to characterize the circuit inside the Blue Dashed Box, that is:

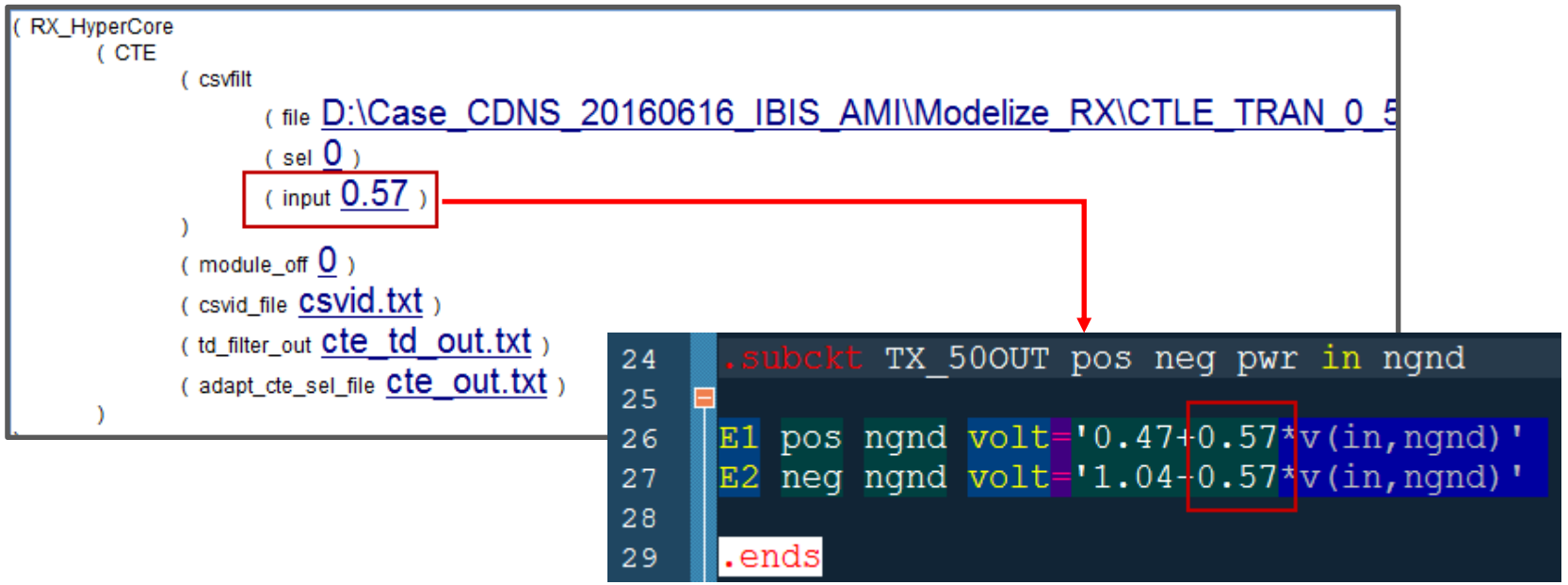
1. The same voltage swing
2. Fast rising time – 1e-21sec (1e-9ps)
3. Small time steps, no more than 1ps

Which we use to characterize the channel.

Treat as an "**Whole Equalizer**"

Modeling in **AMI**

# Characterizing - Normalizing:




Normalize to the voltage swing you use to characterize the channel and the equalizer.

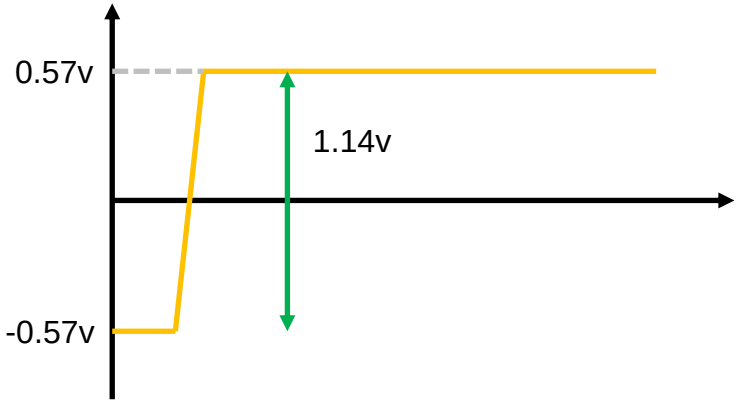
# Characterizing – Normalizing (Cont'd):

Beware!!:

```
( RX_HyperCore
        ( CTE
                ( csvfilt
                        ( file D:\Case_CDNS_20160616_IBIS_AMI\Modelize_RX\CTLE_TRAN_0_5
                        ( sel 0 )
                        ( input 0.57 )
                )
                ( module_off 0 )
                ( csvid_file csvid.txt )
                ( td_filter_out cte_td_out.txt )
                ( adapt_cte_sel_file cte_out.txt )
        )
```

```
24  .subckt TX_50OUT pos neg pwr in ngnd
25
26  E1 pos ngnd volt='0.47+0.57*v(in,ngnd)'
27  E2 neg ngnd volt='1.04-0.57*v(in,ngnd)'
28
29  .ends
```

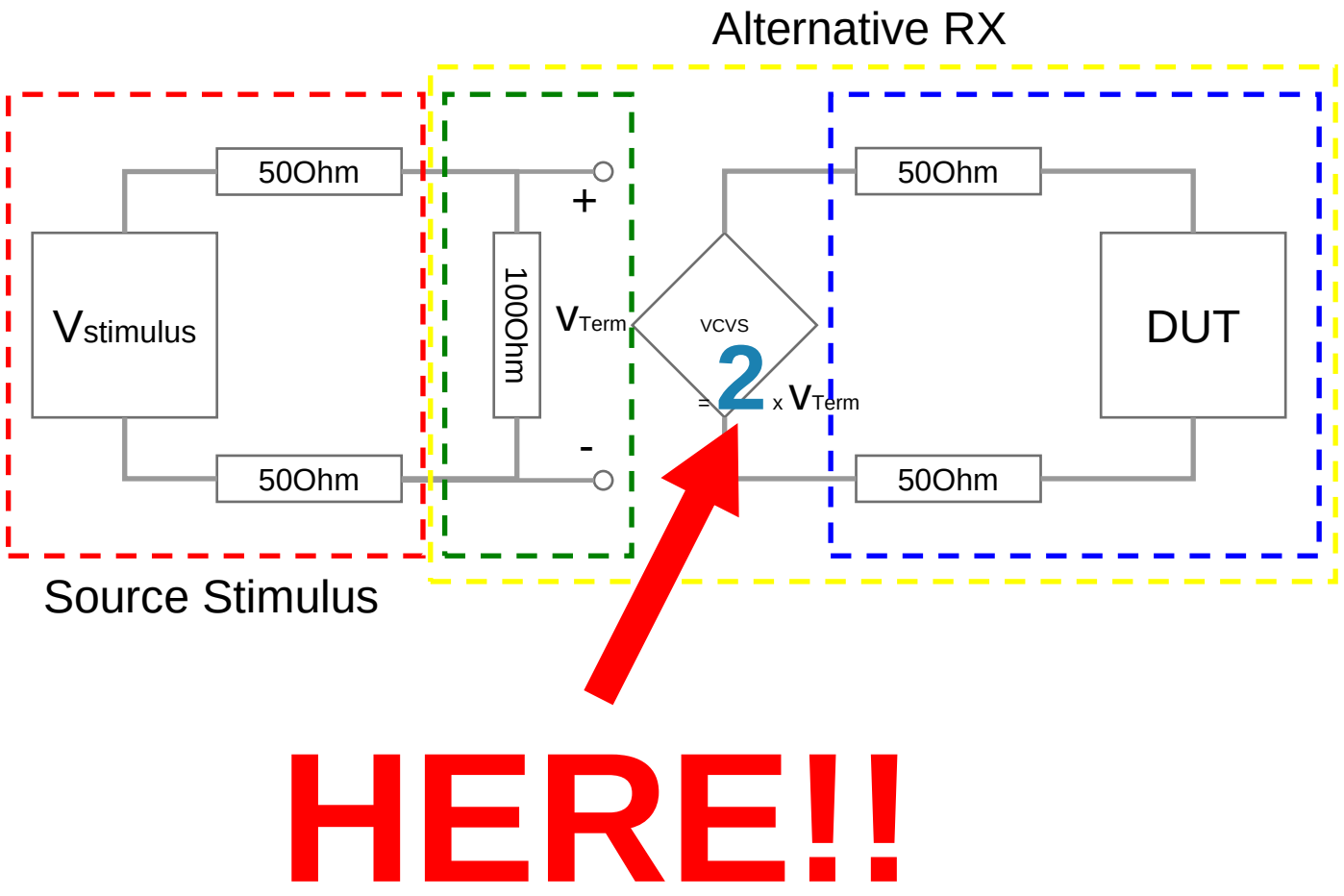
You may wonder the input stimulus is 2x0.57=1.14V,




Why do we normalize it to **0.57v**?

# Characterizing – Normalizing (Cont'd):

The answer is:

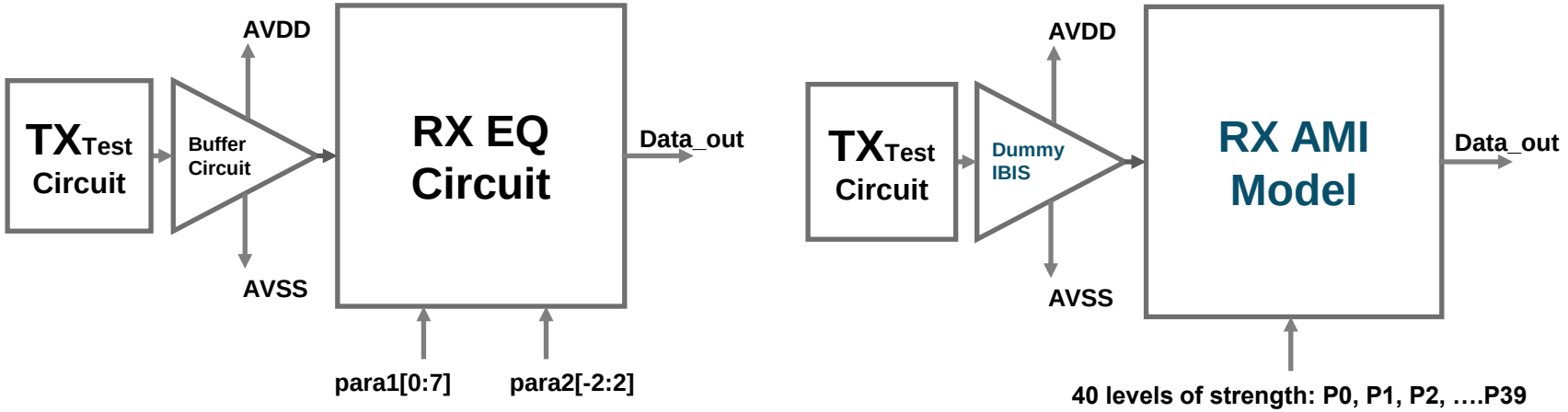


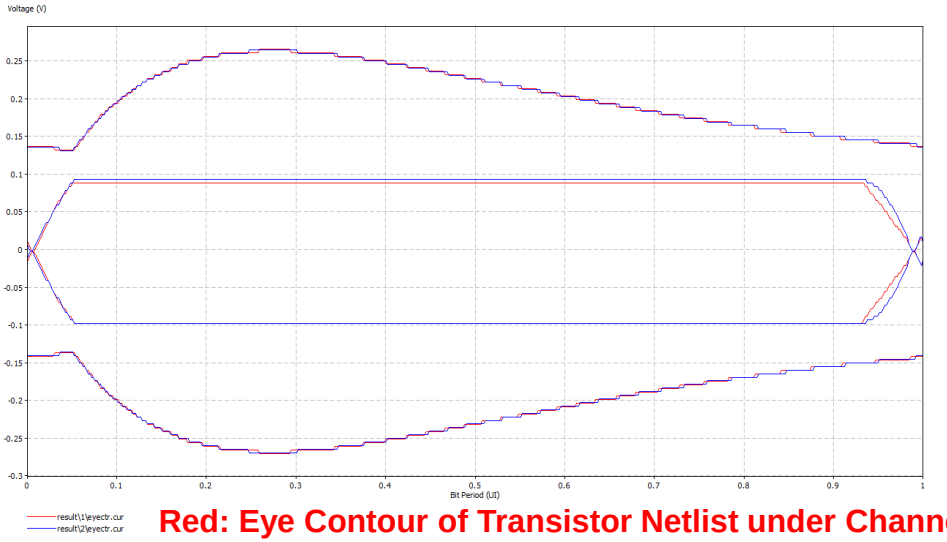

**Question**: What if you are using an IBIS model which is equivalent to a 90Ohm terminator instead of a 100Ohm?

# Correlation – Channel Analysis

between the IBIS-AMI and the Transistor Netlist

**Under a simple test environment:**



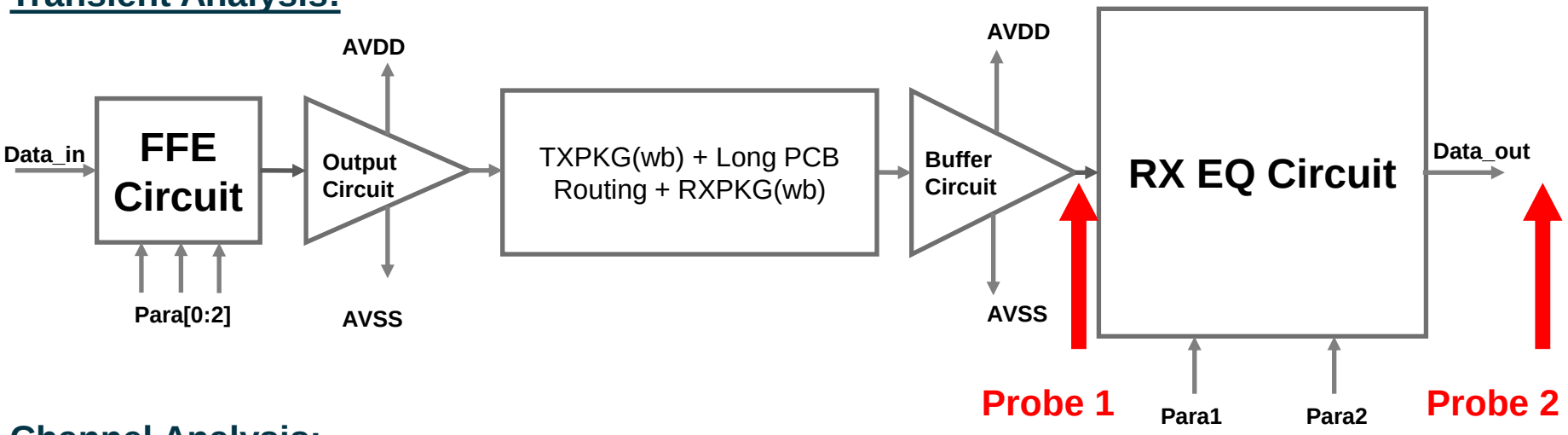




**Red: Eye Contour of Transistor Netlist under Channel Analysis**

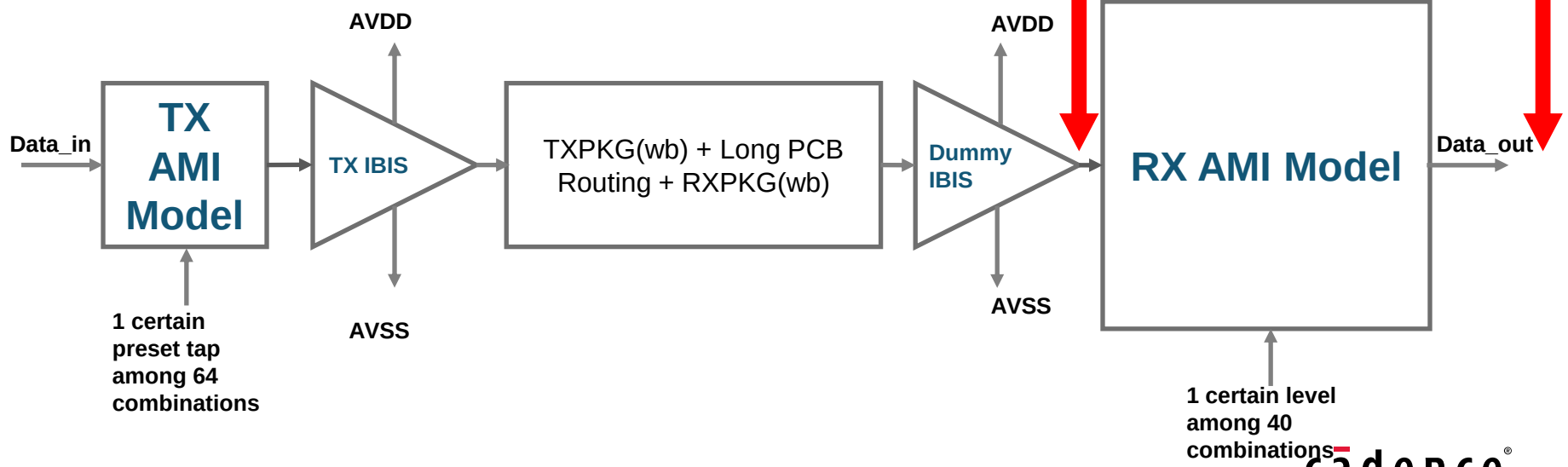
**Blue: Eye Contour of IBIS-AMI model under Channel Analysis**

# Correlation – Mid Length Channel

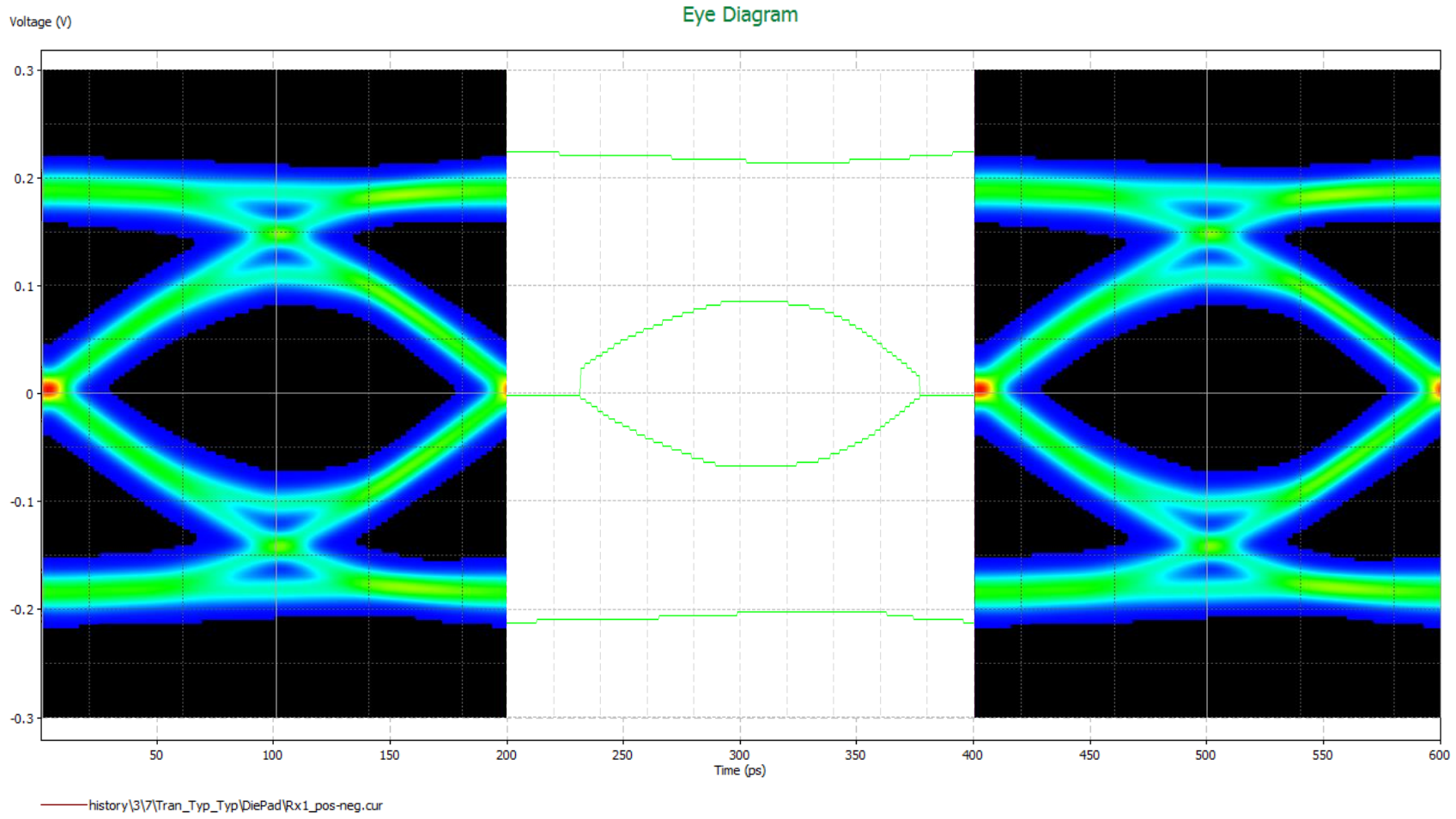
## Transient Analysis:



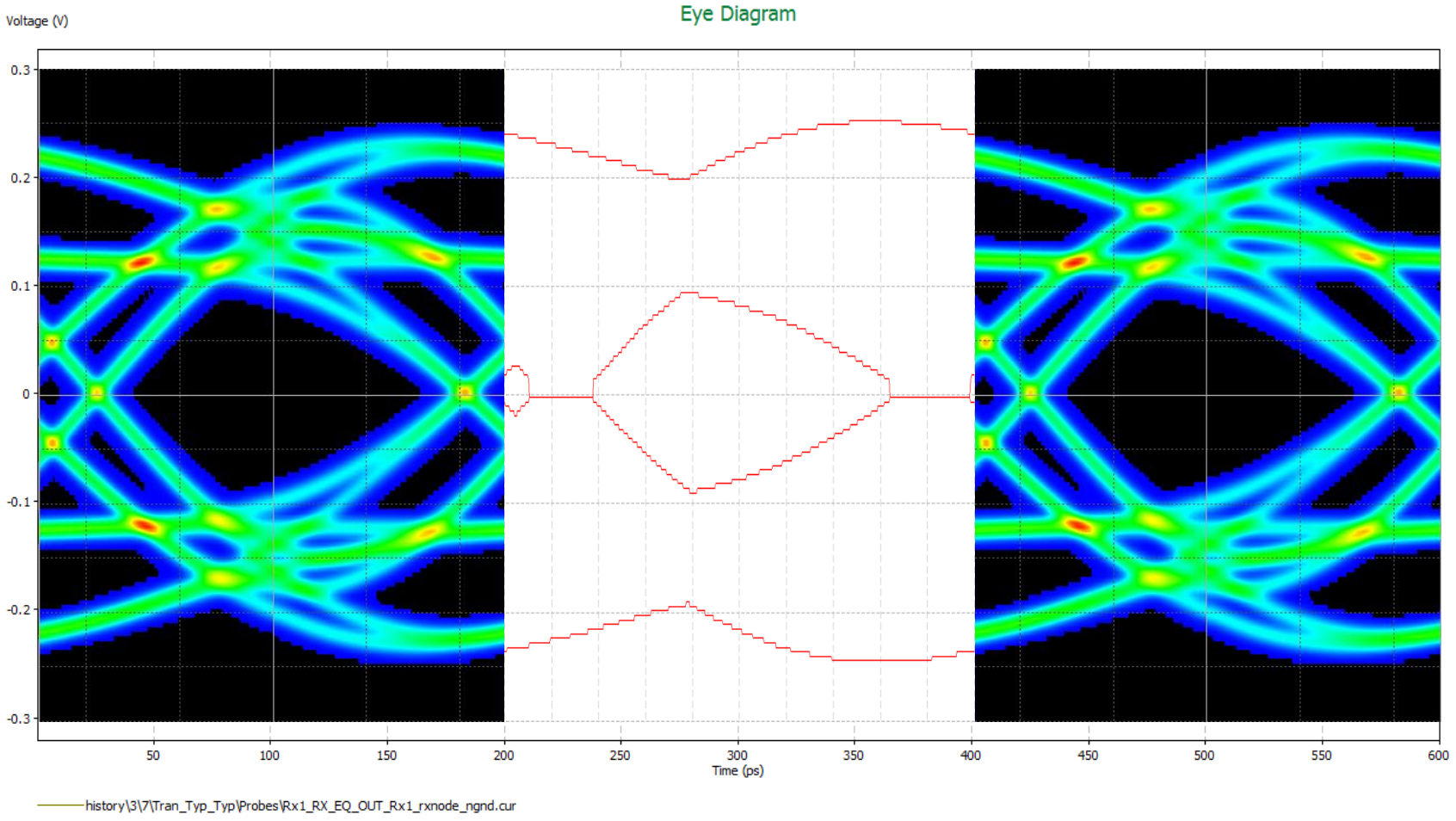

## Channel Analysis:



# Correlation – Mid Length Channel (Probe 1)

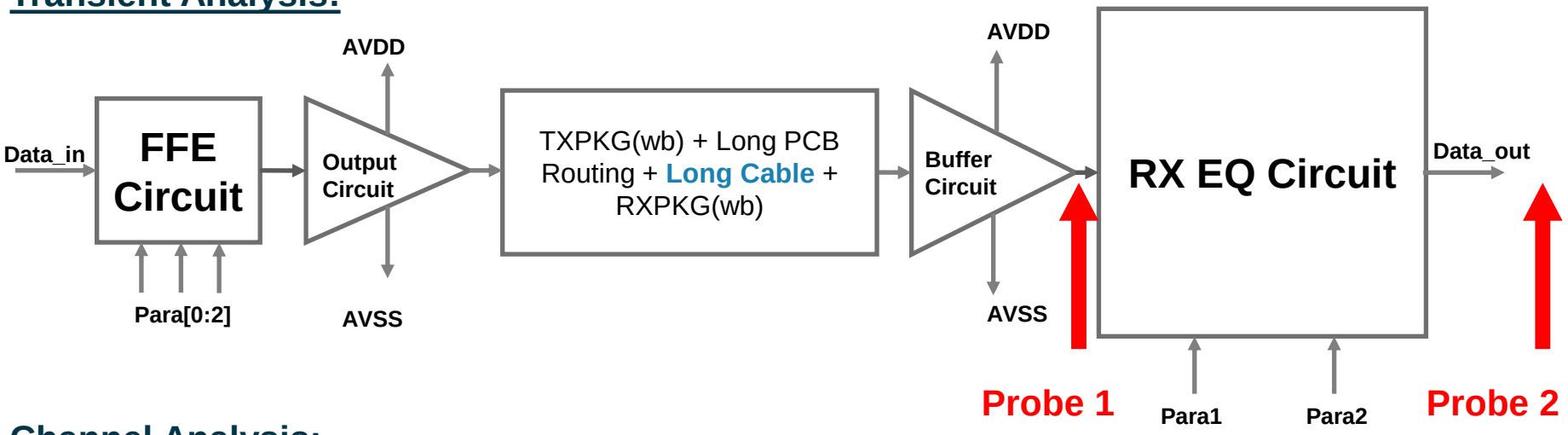


# Correlation – Mid Length Channel (Probe 2)

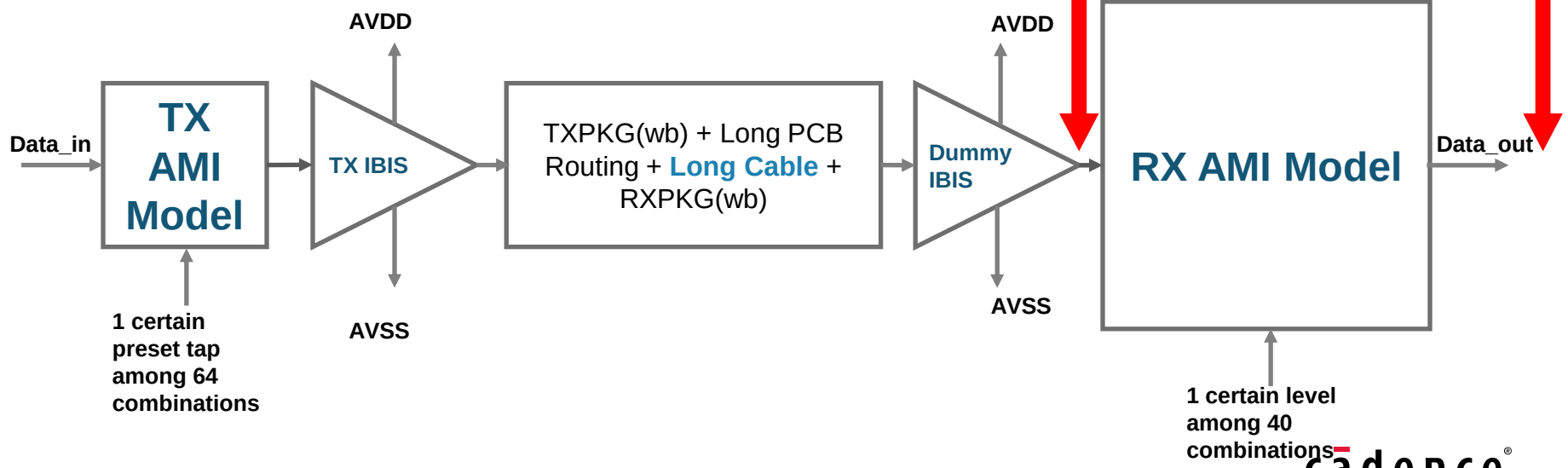


# Correlation – Long Length Channel

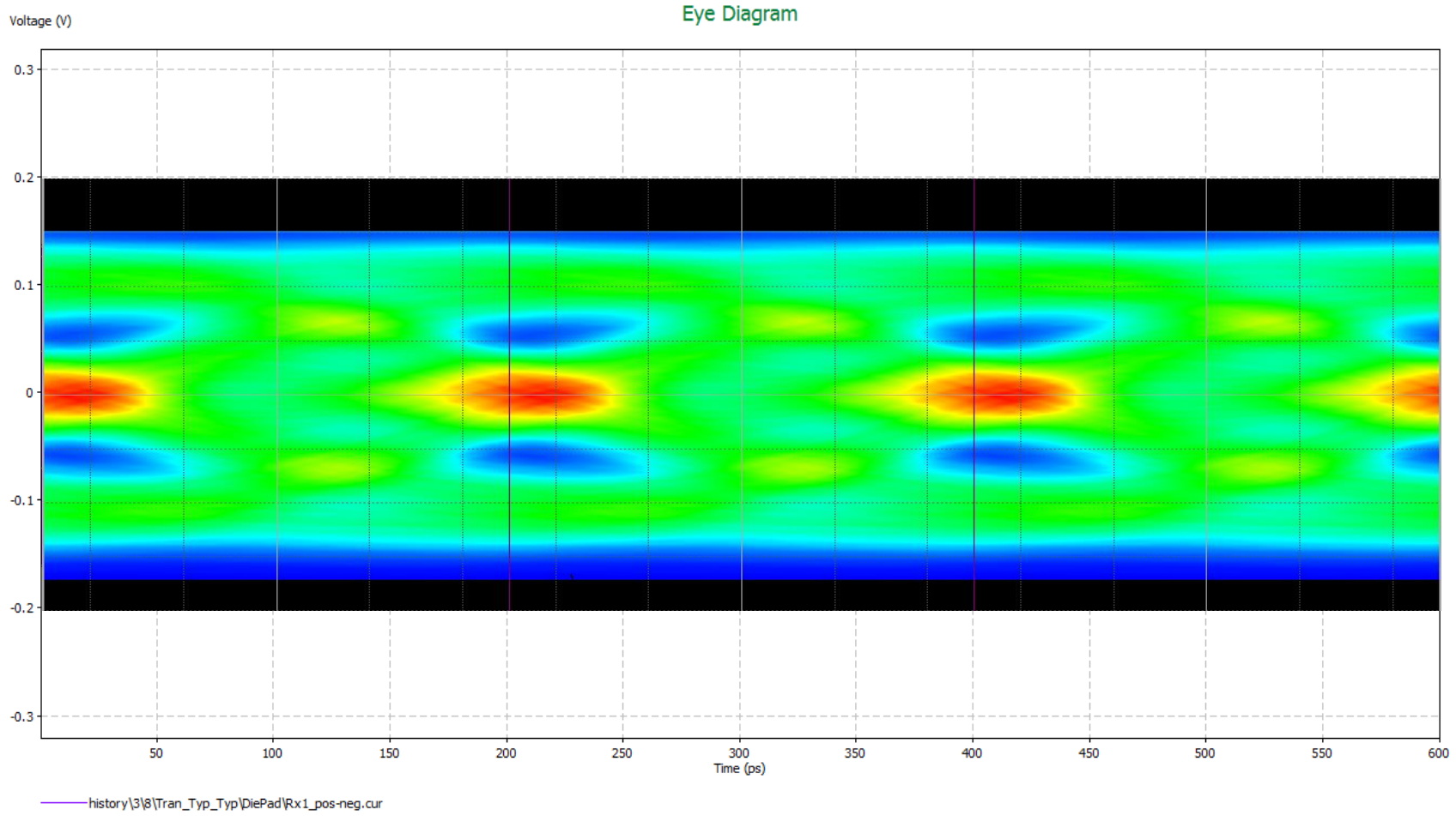
## Transient Analysis:



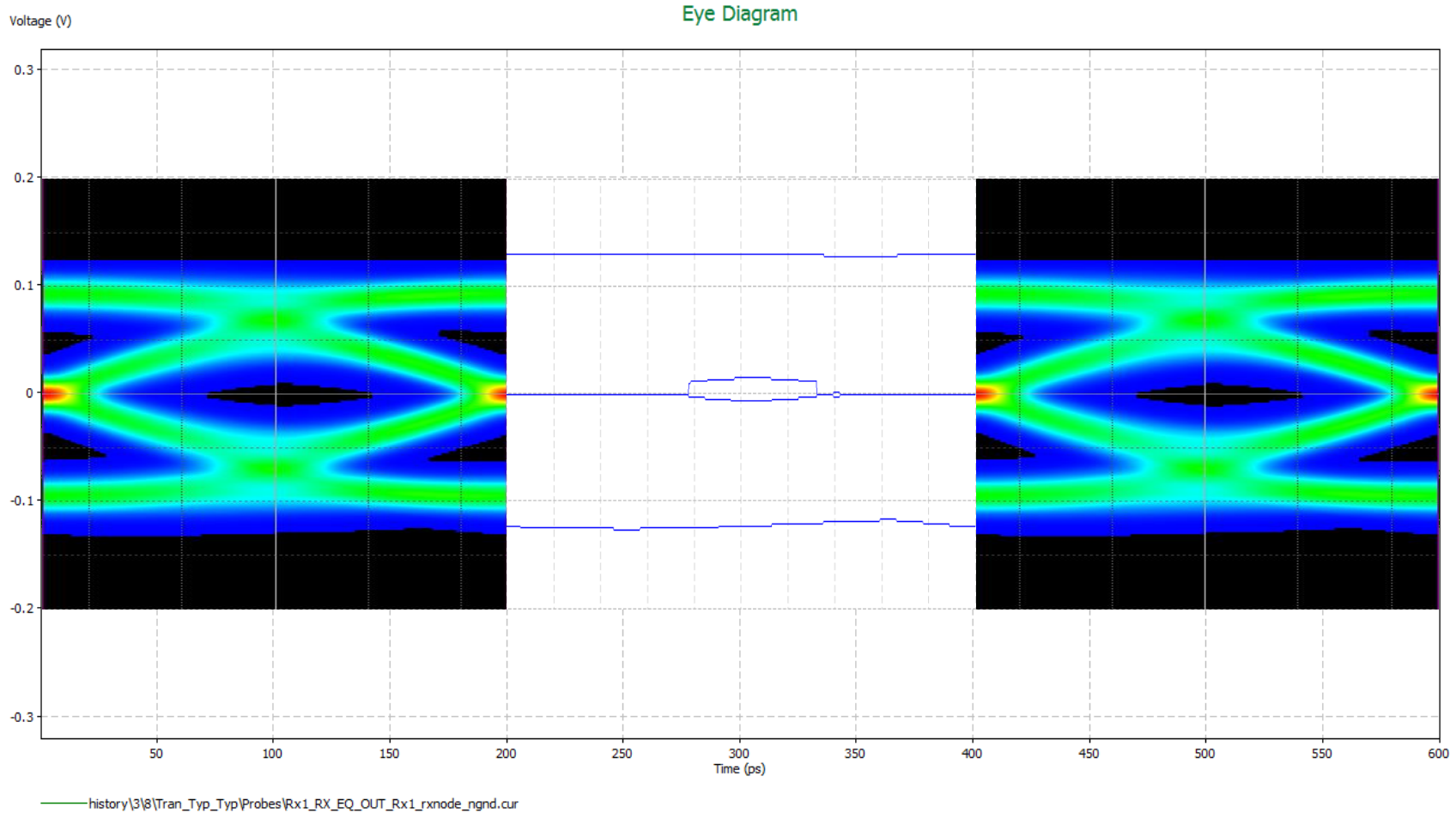

## Channel Analysis:



# Correlation – Long Length Channel (Probe 1)



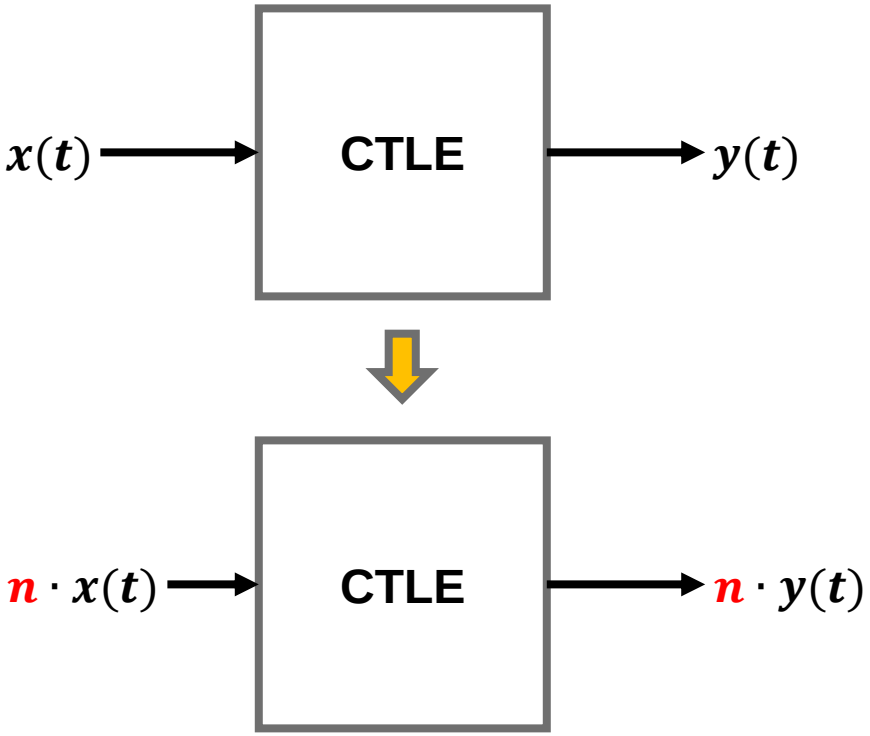
# Correlation – Long Length Channel (Probe 2)



# Limitation

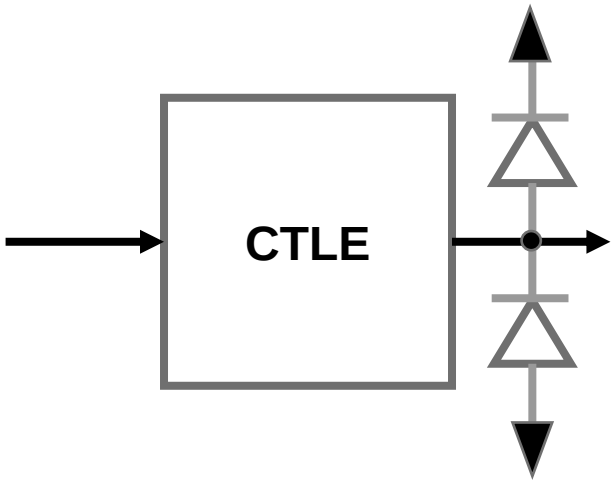
This method is only valid while being applied to a pure CTLE which is composed of linear components, such as R, L, C, Linear E(VCVS), Linear F(CCCS)…etc.

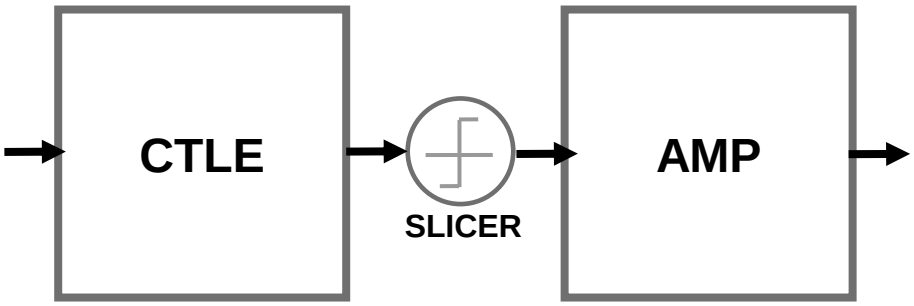
Or in short, a CTLE which satisfies:



# Limitation (Cont'd)

There're still lots of circuits not suitable for this method. For example:



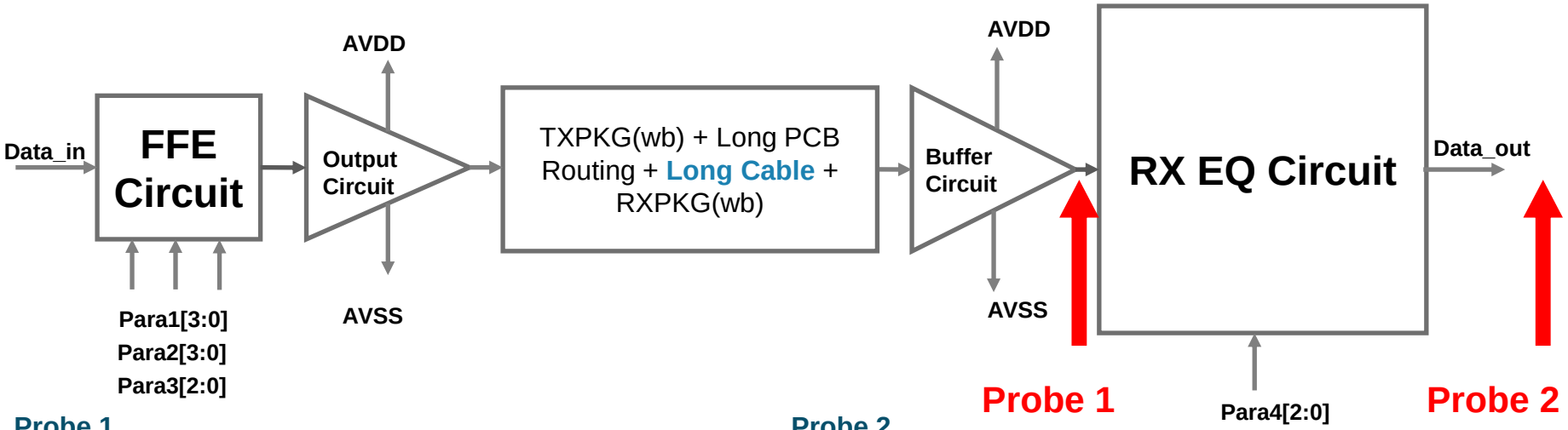



# Conclusion

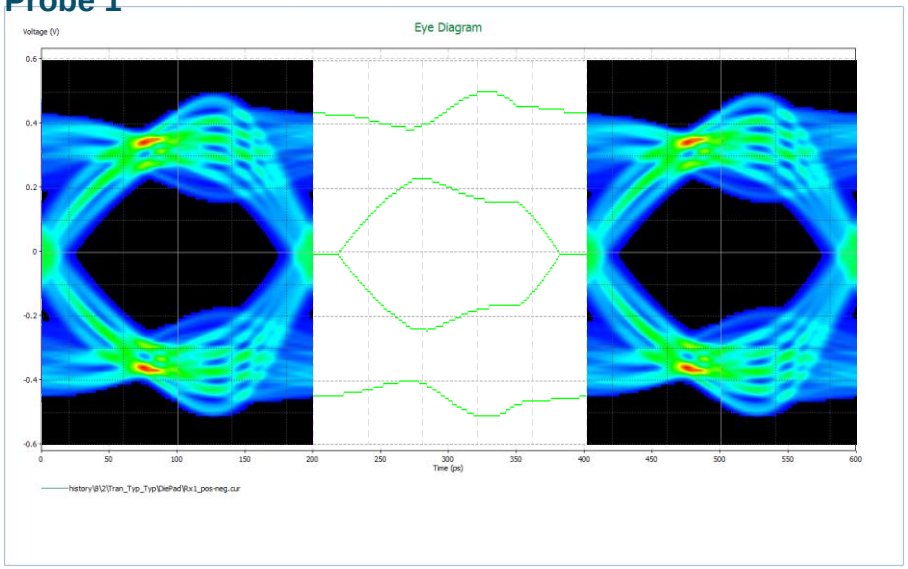
- Cutting/Dividing the whole design is a necessary process during RX IBIS-AMI modeling. This slides provides a method which guarantees the combination of the sub-designs so-divided is exactly equivalent to the original whole design.
- Also, the method in this slides benefits modelers that they will no longer need to model a RX IBIS model. A dummy IBIS will be used for all cases while the buffer characteristics has been modeled into the AMI model. No cutting/dividing is needed any more.
- This slides provides a method to generate IBIS-AMI simply by characterizing the V/T of the netlist – away more accurate than generating IBIS-AMI by inputting parameters values.
- However, this method is only valid while being applied to a purely linear equalizer, that is, there exists a purely linear relationship between the input and output of the equalizer.
- What else? Can a TX FFE be modeled simply by characterizing? Can an non-linear RX CTE be modeled simply by characterizing?

# Correlation – Mid Length Channel

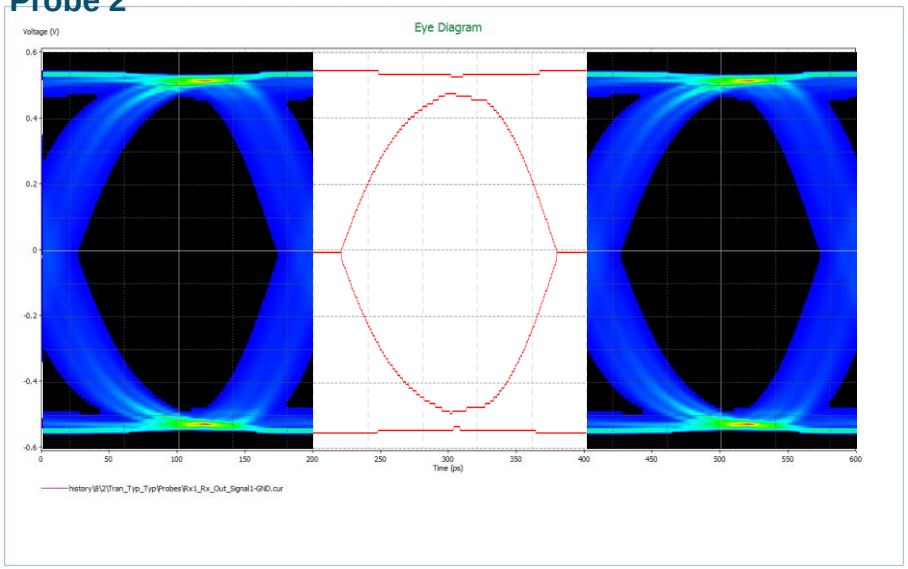
In fact, now we've even developed a flow which can successfully model a **non-linear DFE-free** RX EQ with very good accuracy simply by characterizing: **Example 1**




Probe 1

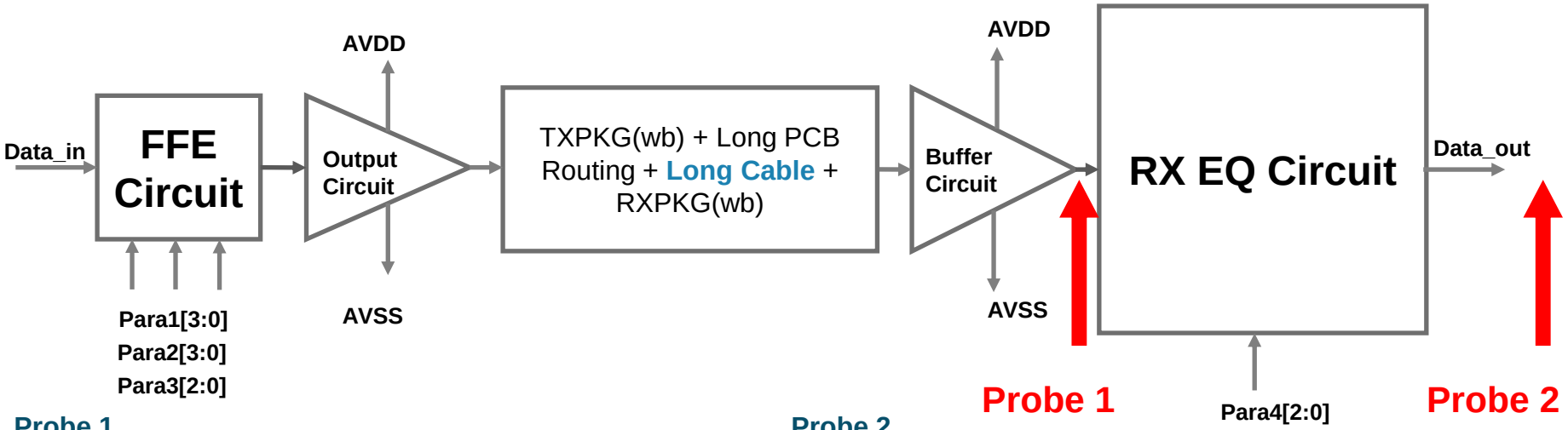



Probe 2

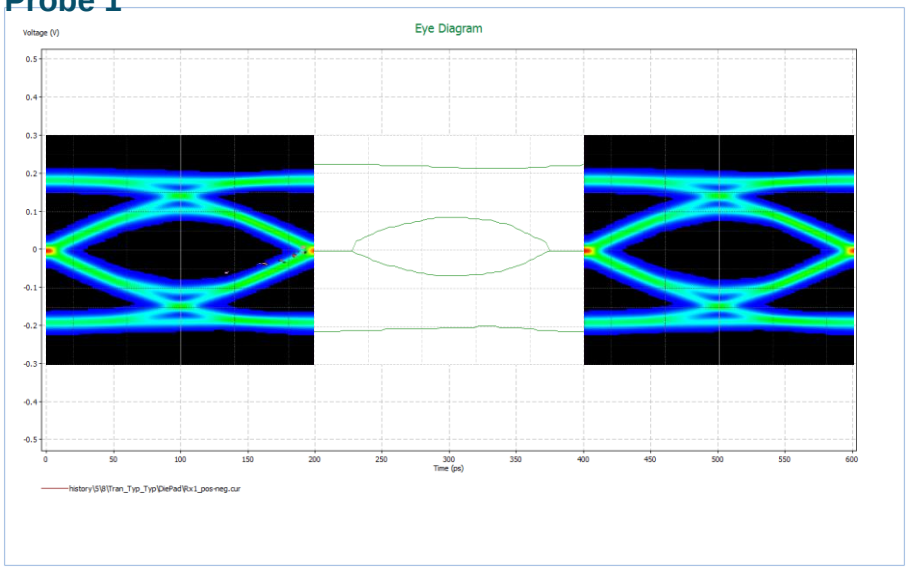


# Correlation – Long Length Channel

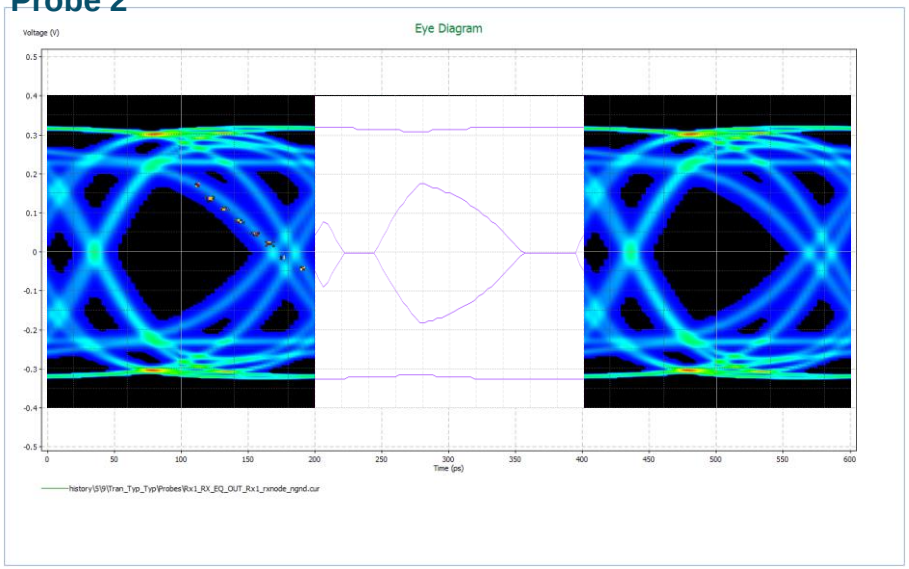
In fact, now we've even developed a flow which can successfully model a **non-linear DFE-free** RX EQ with very good accuracy simply by characterizing: **Example 2**




**Probe 1**




**Probe 2**



*See you on IBIS Summit 2018*

cadence®