POLITECNICO
DI TORINO

European IBIS Summit
Baveno, Italy
May 10, 2017

ACCURATE MACROMODELS OF OUTPUT BUFFERS WITH PRE-/DE-EMPHASIS

Gianni Signorini(1), Claudio Siviero(2), Igor Simone Stievano(2), Stefano Grivet-Talocia(2)

(1) – Intel Corporation, Munich – Germany
(2) – *Politecnico di Torino, Torino - Italy*

DISCLAIMER

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information. Contact your local Intel sales office or your distributor to obtain the latest specification before placing your product order.
INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications. Intel may make changes to specifications, product descriptions, and plans at any time, without notice. All products, dates, and figures are preliminary for planning purposes and are subject to change without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.
The code names Romley, Grantley, Brickland, Purley, Sandy Bridge, Ivy Bridge, Haswell, Skylake and Jordan Creek presented in this document are only for use by Intel to identify products, technologies, or services in development, that have not been made commercially available to the public, i.e., announced, launched or shipped. They are not "commercial" names for products or services and are not intended to function as trademarks.
Copies of documents which have an order number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.
*Other names and brands may be claimed as the property of others.

ADDITIONAL DISCLAIMER

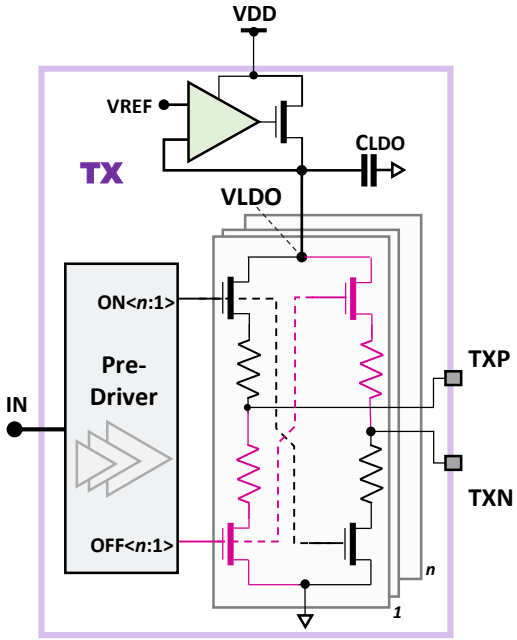
The following is solely the opinion of the authors and does not necessarily represent the policies or opinions of Intel Corporation or its subsidiaries.

Agenda

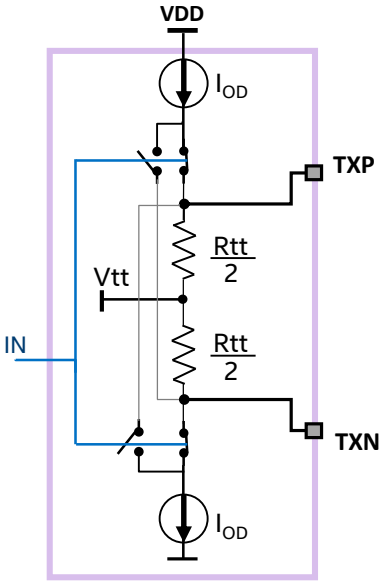
- **High Speed Differential Transmitters (HS-TX) & Pre-/De-Emphasis**
  - Circuit topologies & Basics
  - Model types (IBIS, IBIS-AMI)
- **MPILOG & Recent enhancements for HS-TX**
  - Model structure & Parametric Equations
  - SPICE simulations for Parameters Identification
  - Enhancements to support TX EQ
- **Results & Validations**
- **Learnings & Conclusions**

HS-TX w/EQ & IBIS Solutions

HS-TX Topologies



Low-Swing Voltage-Mode Differential Driver



Pseudo Current-Mode Differential Driver

High-Speed Differential Drivers:

- ***voltage-mode***
- ***current-mode***

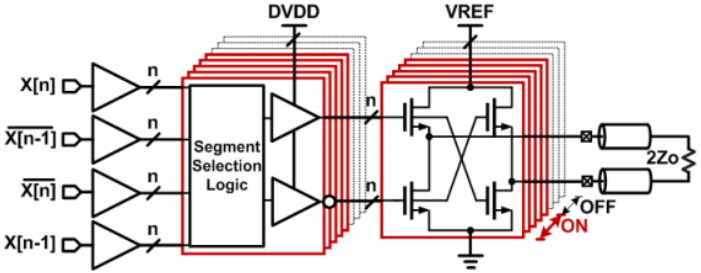
In the context of I/O-modeling, topologies are classified in:

- **Pseudo-Differential:**
  no explicit TXP/TXN current path
- **True-Differential:**
  explicit TXP/TXN current path exists

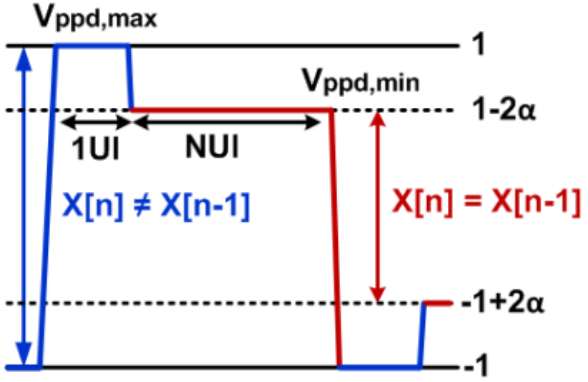
Pre-/De-Emphasis

Segmented Implementation [2]

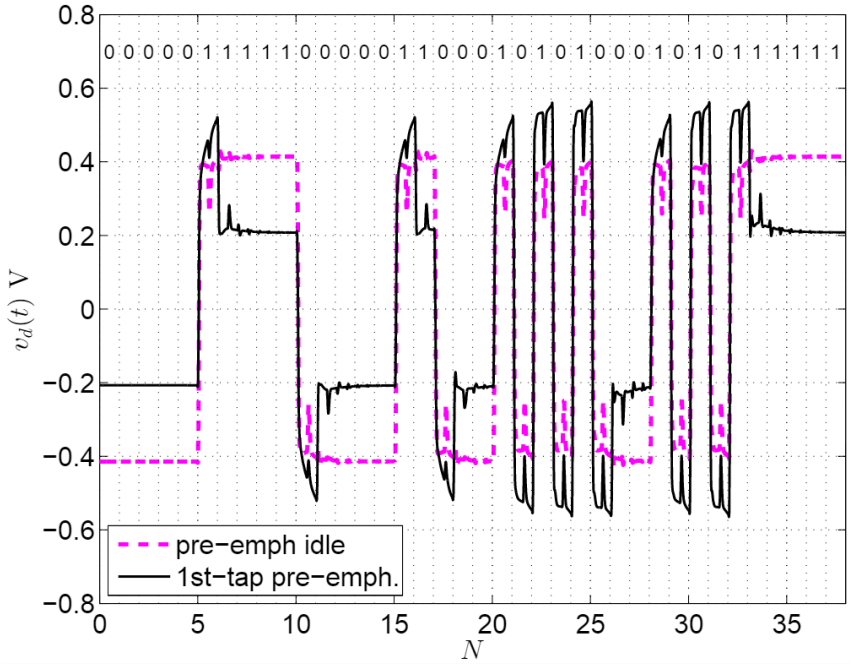
[2] R. Sredojevic, et al., JSSC 2011



Example: 1 post-tap



*Real Example: **no TX-EQ** vs **TX-EQ***

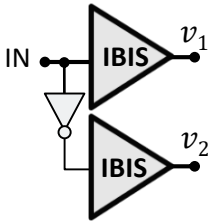


HS-TX & Model Types

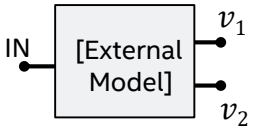
IBIS Standard

Baseline IBIS

Only pseudo-differential
No Pre-/De-Emphasis



Pseudo-Differential
(2x Single-Ended)



True-Differential

HS-TX & Model Types

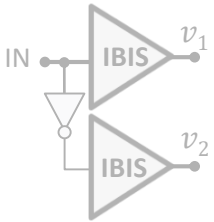
IBIS Standard

Baseline IBIS

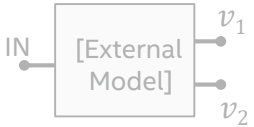
Only pseudo-differential
No Pre-/De-Emphasis

IBIS w/ [Driver Schedule]

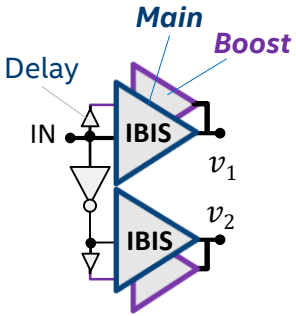
Pseudo-differential
Pre-/De-Emphasis
Fixed Coefficients



Pseudo-Differential
(2x Single-Ended)



True-Differential



Pseudo-Differential
w/ [Driver Schedule]

HS-TX & Model Types

** **AMI** = **A**lgorithmic **M**odeling **I**nterfaces*

IBIS Standard

Baseline IBIS

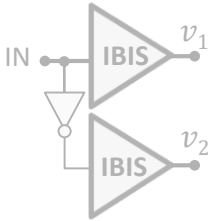
Only pseudo-differential
No Pre-/De-Emphasis

IBIS w/ [Driver Schedule]

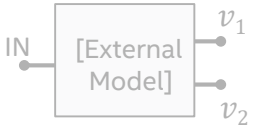
Pseudo-differential
Pre-/De-Emphasis
Fixed Coefficients

IBIS-AMI

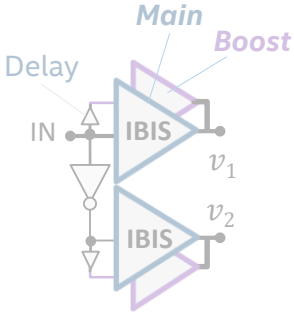
LTI: R-C approximation
Pre-/De-Emphasis / EQ
.ami/.dll
Only VDIFF, no VCM
Bit-by-bit and Statistical



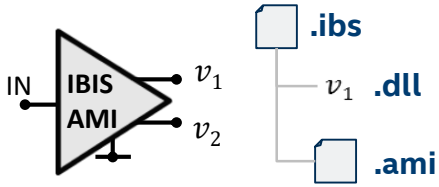
Pseudo-Differential
(2x Single-Ended)



True-Differential



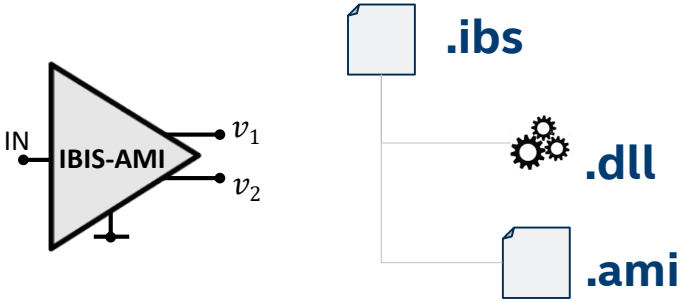
Pseudo-Differential
w/ [Driver Schedule]



Differential signal
LTI & DSP theory
Supports EQ

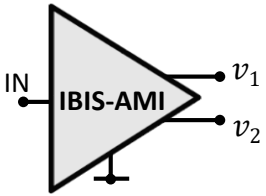
IBIS-AMI

** **AMI** = **A**lgorithmic **M**odeling **I**nterfaces*



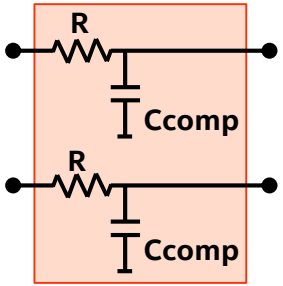
IBIS-AMI

** **AMI** = **A**lgorithmic **M**odeling **I**nterfaces*



- "*Analog part*"
- Linearized I-V (Resistive) + Ccomp + Ramp
- Link to algorithmic part (.dll/.ami)

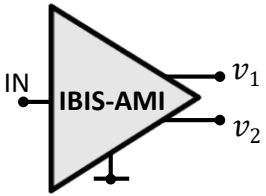
Analog TX

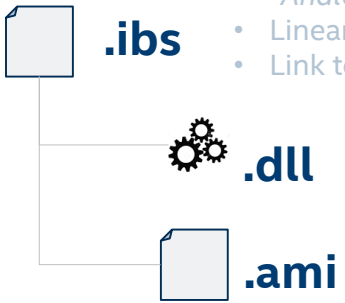


- Ccomp & R (i.e., I/V) from .ibs
- LTI assumption

IBIS-AMI

** **AMI** = **A**lgorithmic **M**odeling **I**nterfaces*



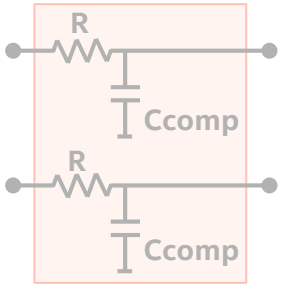


- *"Analog part"*
- Linearized I-V (Resistive) + Ccomp + Ramp
- Link to algorithmic part (.dll/.ami)

- Represent TX/RX Equalization
- Compiled "parametric" algorithms
- Uses DSP techniques (e.g., FIR, S/H, …)

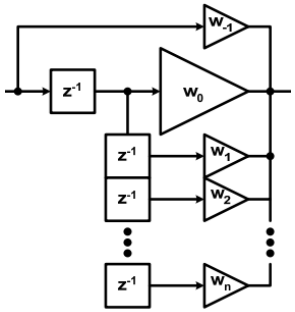
- Specifies configuration for parameters is .dll

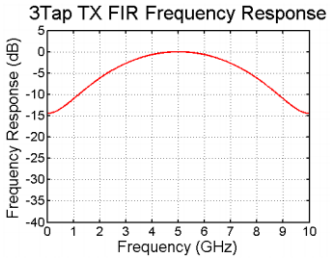
Analog TX



- Ccomp & R (i.e., I/V) from .ibs
- LTI assumption

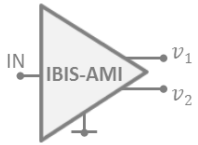
TX EQ





- TX EQ = Ideal FIR filter
- "Automatic .ami/.dll from TX-arch.

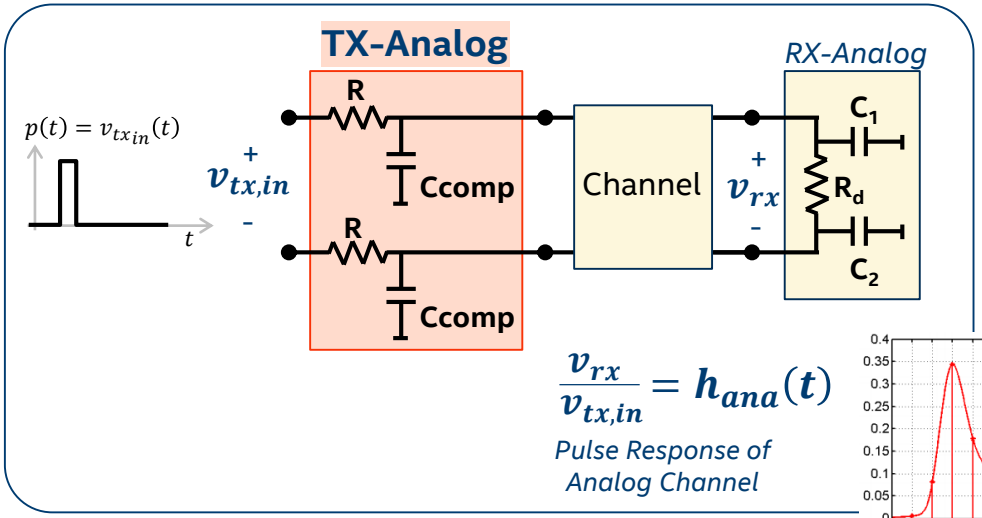
IBIS-AMI



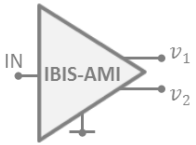
.ibs
- *"Analog part"*
- Linearized I-V (Resistive) + Ccomp + Ramp
- Link to algorithmic part (.dll/.ami)

.dll
- Represent TX/RX Equalization
- Compiled "parametric" algorithms
- Uses DSP techniques (e.g., FIR, S/H, ...)

.ami
- Specifies configuration for parameters is .dll



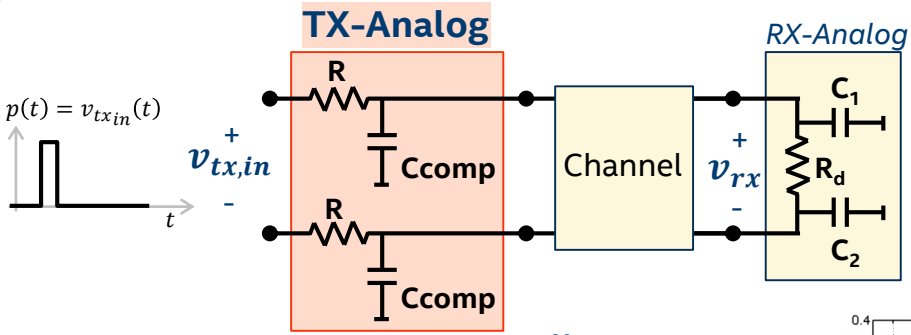
IBIS-AMI



.ibs
- *"Analog part"*
- Linearized I-V (Resistive) + Ccomp + Ramp
- Link to algorithmic part (.dll/.ami)

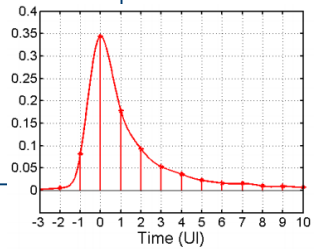
.dll
- Represent TX/RX Equalization
- Compiled "parametric" algorithms
- Uses DSP techniques (e.g., FIR, S/H, ...)

.ami
- Specifies configuration for parameters is .dll

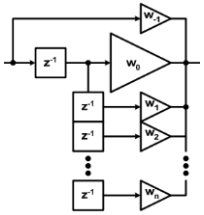


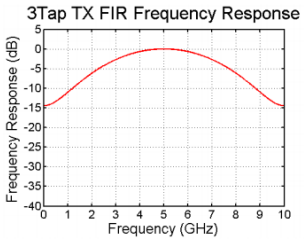
$$\frac{v_{rx}}{v_{tx,in}} = h_{ana}(t)$$

Pulse Response of Analog Channel



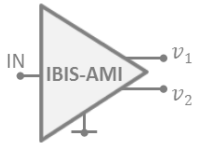
TX Equalizer





$h_{FIR}(t)$

IBIS-AMI

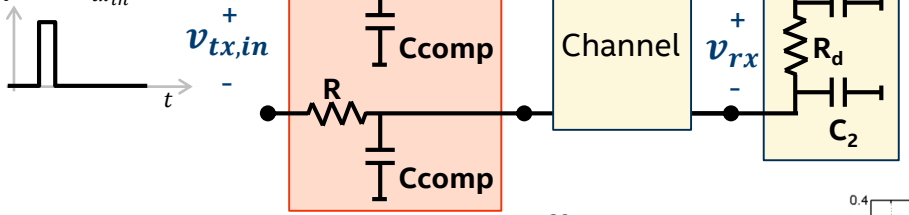


.ibs
- *"Analog part"*
- Linearized I-V (Resistive) + Ccomp + Ramp
- Link to algorithmic part (.dll/.ami)

.dll
- Represent TX/RX Equalization
- Compiled "parametric" algorithms
- Uses DSP techniques (e.g., FIR, S/H, ...)

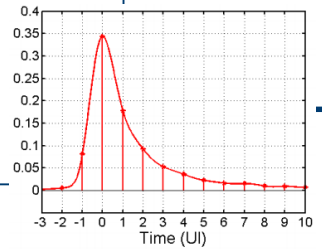
.ami
- Specifies configuration for parameters is .dll

TX-Analog

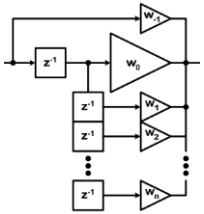


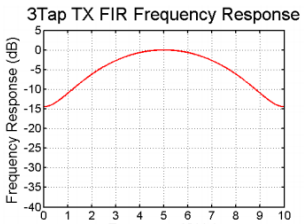
$$\frac{v_{rx}}{v_{tx,in}} = h_{ana}(t)$$

Pulse Response of Analog Channel



TX Equalizer

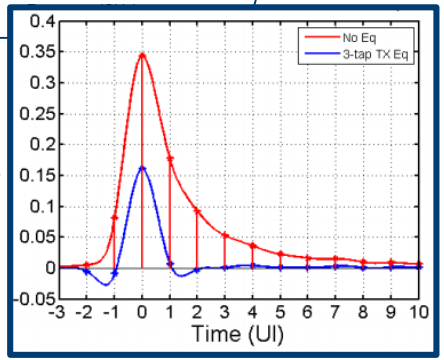




$h_{FIR}(t)$

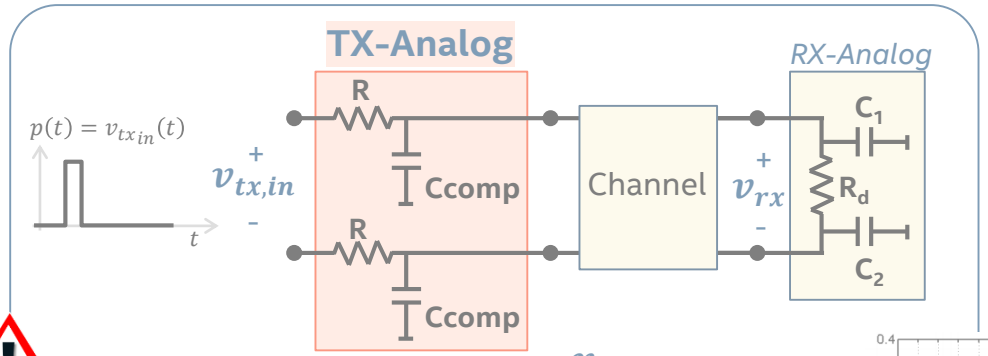
$h_{TX_{EQ}}$

Equalized Pulse Response



http://www.ece.tamu.edu/~spalermo/ecen689/lecture7_ee720_eq_intro_txeq.pdf

IBIS-AMI

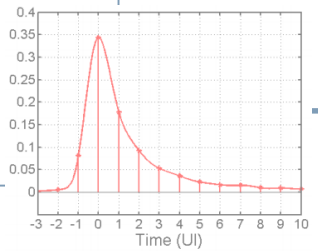


R+C+Lin-Slope approx!

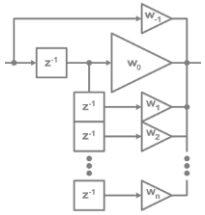
Only Differential signal!

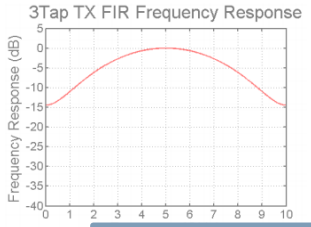
$$\frac{v_{rx}}{v_{tx,in}} = h_{ana}(t)$$

Pulse Response of Analog Channel



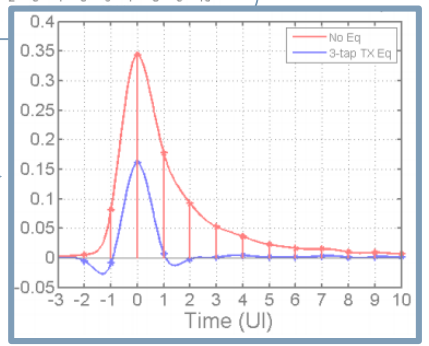
TX EQ = Ideal FIR filter





$h_{FIR}(t)$

$h_{TX_{EQ}}$



Approximations → possible accuracy limitations for some devices! …

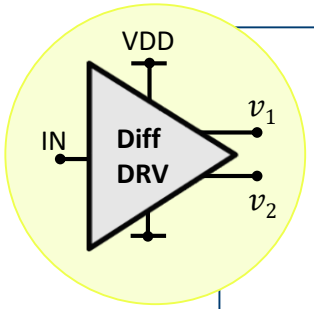
… improvements?

MPILOG & TX-EQ

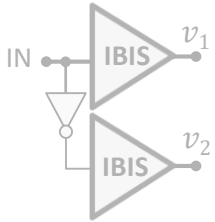
MPILOG Models

* MPILOG = **M**acromodeling via **P**arametric **I**dentification of **LO**gic **G**ates

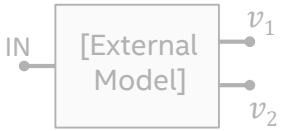
"MPILOG" *Framework*



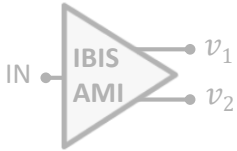
IBIS



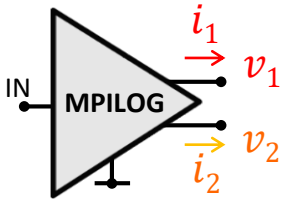
Pseudo-Differential
(2x Single-Ended)

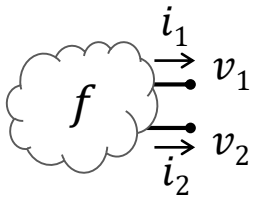


True-Differential



Support TX EQ



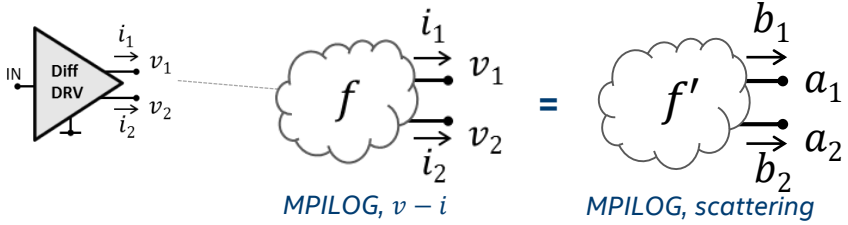


$$(i_1, i_2) = \boldsymbol{f}(v_1, v_2, IN, {}^{\partial}/_{\partial t})$$

- "Grey-box" macromodels
- Output currents i_1, i_2 reproduced with a "**mathematical function**" having output voltages v_1, v_2 as inputs *(true-differential)*

- Function $\boldsymbol{f}$ is a **parametric equation**, **fixed structure**
- Parameter Identification = **post-processing of ad-hoc SPICE simulations**
- Function f is cast as SPICE netlist (R, C, CS) or Verilog-A code

Enhanced MPILOG models for Pre/De-Emph



MPILOG, $v - i$ *MPILOG, scattering*

$$b_{1,2} = (v_{1,2} - Z_0 \cdot i_{1,2})/(2 \cdot \sqrt{Z_0})$$
$$a_{1,2} = (v_{1,2} + Z_0 \cdot i_{1,2})/(2 \cdot \sqrt{Z_0})$$

Scattering waves

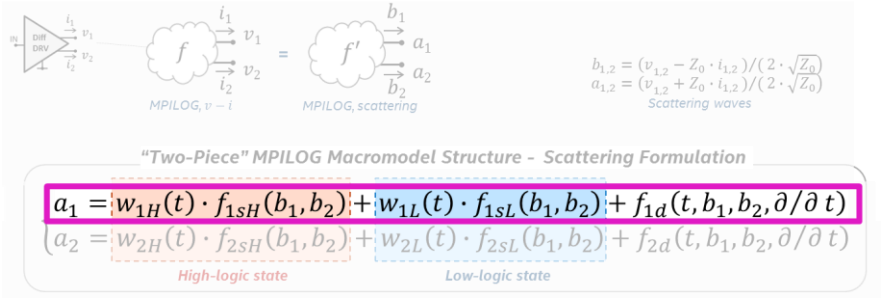
MPILOG Macromodel Structure - Scattering Formulation

High-logic state ***Low-logic state***

$$\begin{cases} a_1 = w_{1H}(t) \cdot f_{1sH}(b_1, b_2) + w_{1L}(t) \cdot f_{1sL}(b_1, b_2) + f_{1d}(t, b_1, b_2, \partial/\partial\, t) \\ a_2 = w_{2H}(t) \cdot f_{2sH}(b_1, b_2) + w_{2L}(t) \cdot f_{2sL}(b_1, b_2) + f_{2d}(t, b_1, b_2, \partial/\partial\, t) \end{cases}$$

≈ "rising/falling/**EQ**" ≈ "Pull-up" ≈ "falling/rising/**EQ**" ≈ "Pull-down" ≈ "Ccomp", dynamics

Enhanced MPILOG models for Pre/De-Emph



① *I-V → I-[v1,v2]*
3D Static Surfaces

③
MISO Dynamic Model

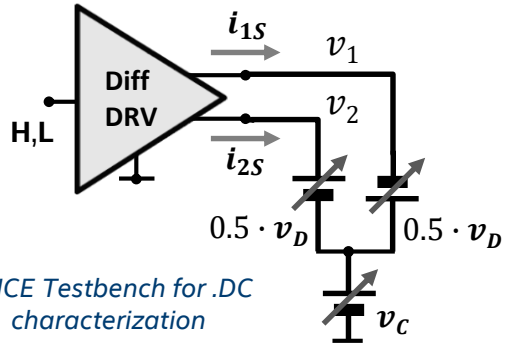
e.g., $a_1 = w_{1H}(t) \cdot f_{1sH}(b_1, b_2) + w_{1L}(t) \cdot f_{1sL}(b_1, b_2) + f_{1d}(t, b_1, b_2, \partial/\partial t)$

Weighting Functions
② *V-t curves & EQ*

Ccomp → Poles&Zeros

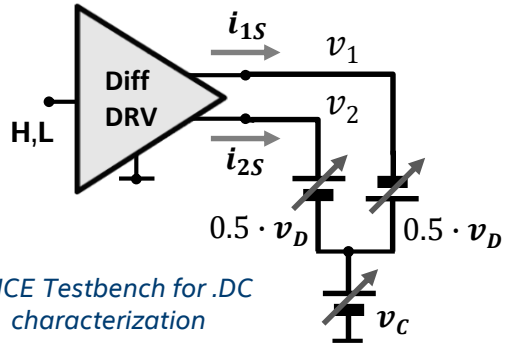
3 model "sub-components" to be identified from transistor-level SPICE responses!

Static Characteristics



SPICE Testbench for .DC characterization

Static Characteristics

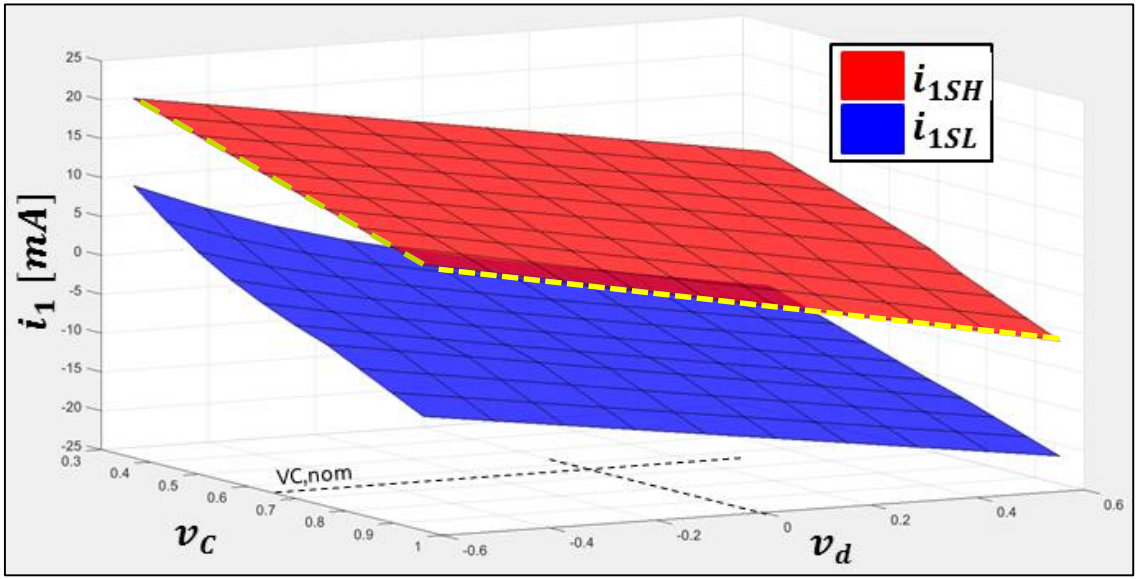


SPICE Testbench for .DC characterization

Two Slopes in Surface

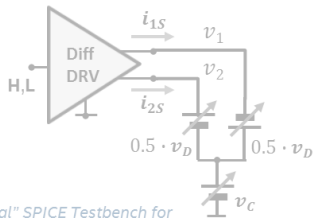
↕

Currents = function of <u>both</u> common-mode and differential voltage

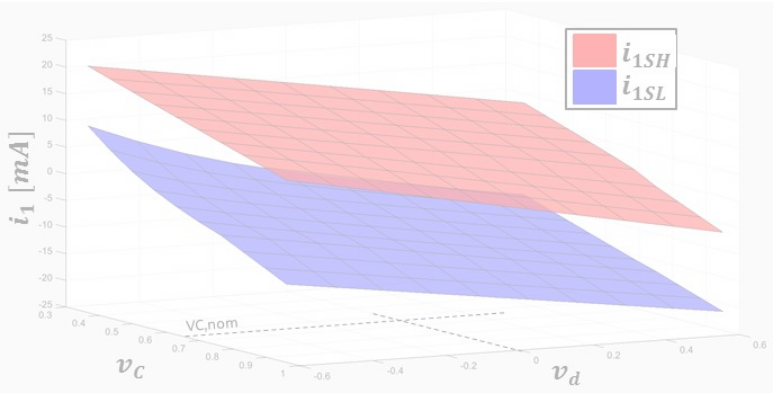


3D Surfaces representing Output Static Characteristics (true-differential behaviour)

Static Characteristics



"Ideal" SPICE Testbench for .DC characterization



MPILOG Solution

(True-Differential) **Static Surfaces → SVD Approx. → SPICE/Verilog-A**

Challenges/Open Topics:

- Nested .DC sweep characterization is *often* problematic → different method?
  (e.g., internal regulator may require proper start-up, clocked buffers, etc.)
- Surfaces look quite "regular" → is it possible to *simplify* the characterization?

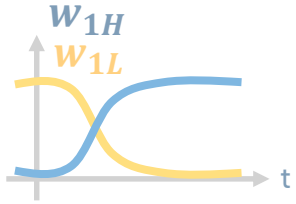
Weighting Functions

Differential drivers:
"complementary" behavior

$$w_{1H} = 1 - w_{1L}$$

$$a_1 = w_{1H} \cdot f_{1SH}(b_{C,NOM}, b_{C,NOM}) + w_{1L} \cdot f_{1SL}(b_{C,NOM}, b_{C,NOM}) + f_{1d}(b_{C,NOM}, b_{C,NOM})$$

Weighting Functions



Differential drivers: "complementary" behavior

$$w_{1H} = 1 - w_{1L}$$

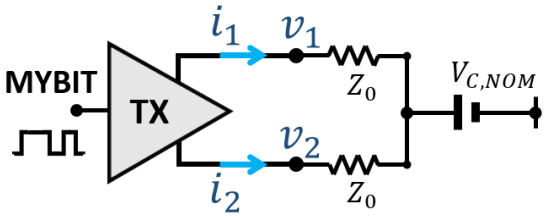
Unknown

$$a_1 = w_{1H} \cdot f_{1SH}(b_{C,NOM}, b_{C,NOM}) + w_{1L} \cdot f_{1SL}(b_{C,NOM}, b_{C,NOM}) + \cancel{f_{1d}(b_{C,NOM}, b_{C,NOM})}$$

From HSPICE

Known from static surface characterization

Imposing no dynamic current by testbench construction



Electrical SPICE Testbench for Weighting-function Characterization (maintain $b_1 = b_2 = b_{c,NOM}$)

Weighting Functions

Differential drivers: "complementary" behavior

$$w_{1H} = 1 - w_{1L}$$

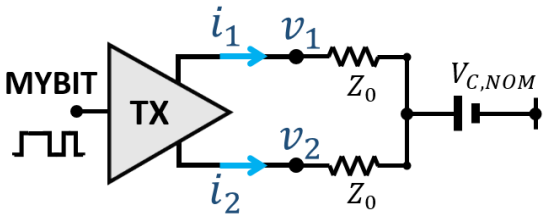
Unknown

$$a_1 = w_{1H} \cdot f_{1SH}(b_{C,NOM}, b_{C,NOM}) + w_{1L} \cdot f_{1SL}(b_{C,NOM}, b_{C,NOM}) + \cancel{f_{1d}(b_{C,NOM}, b_{C,NOM})}$$

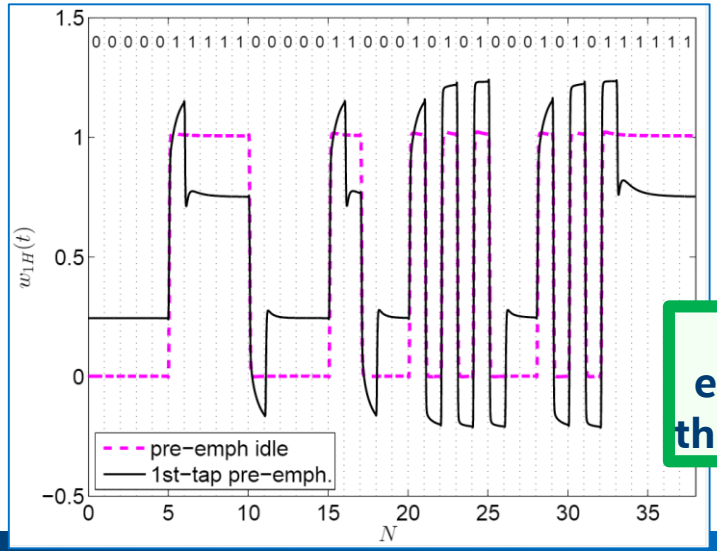
From HSPICE

Known from static surface characterization

Imposing no dynamic current by testbench construction

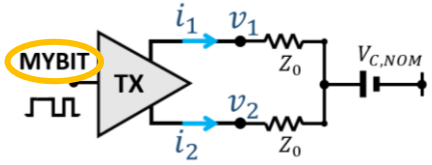


Electrical SPICE Testbench for Weighting-function Characterization (maintain $b_1 = b_2 = b_{c,NOM}$)



Pre-/De-emphasis effect is embedded in the weighting functions

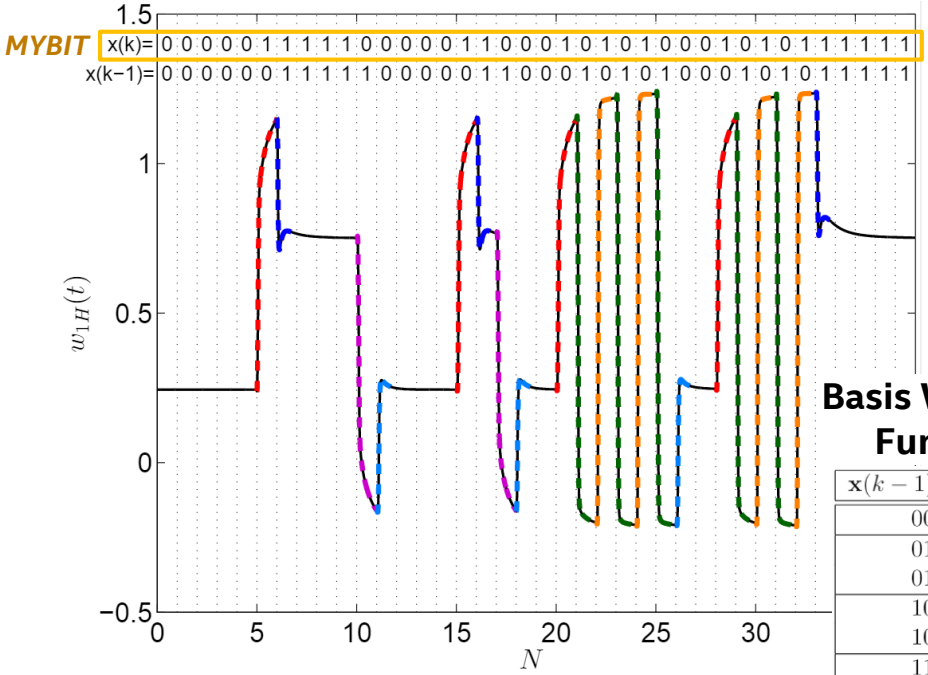
TX-EQ in Weighting Functions



Electrical SPICE Testbench for Weighting-function Characterization

MYBIT is synthesized in order to **stress all possible TX state-transitions**.

MYBIT length depends on TX EQ #PRE and #POST taps (*i.e., #states*)

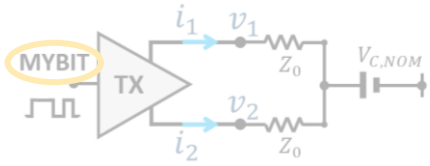


Basis Weighting Functions

$\mathbf{x}(k-1)$		$\mathbf{x}(k)$
00	→	10
01	→	10
01	→	00
10	→	01
10	→	11
11	→	01

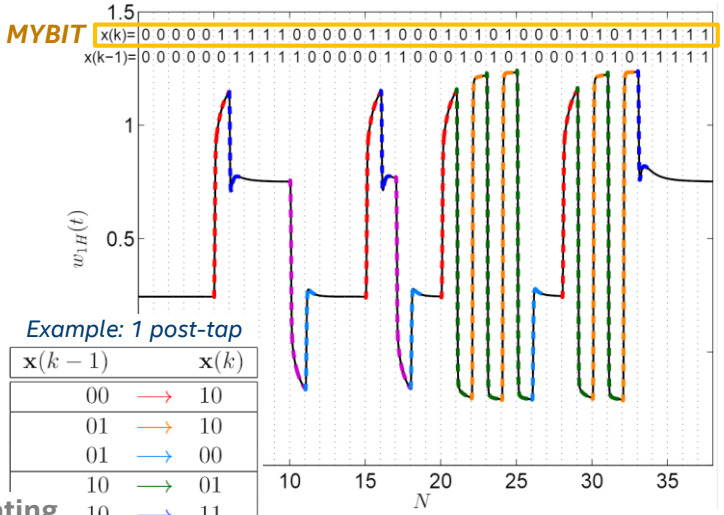
Example: 1 post-tap

TX-EQ in Weighting Functions



MYBIT is synthesized in order to stress all possible TX state-transitions.

MYBIT length depends on TX EQ #PRE and #POST taps (*i.e., #states*)



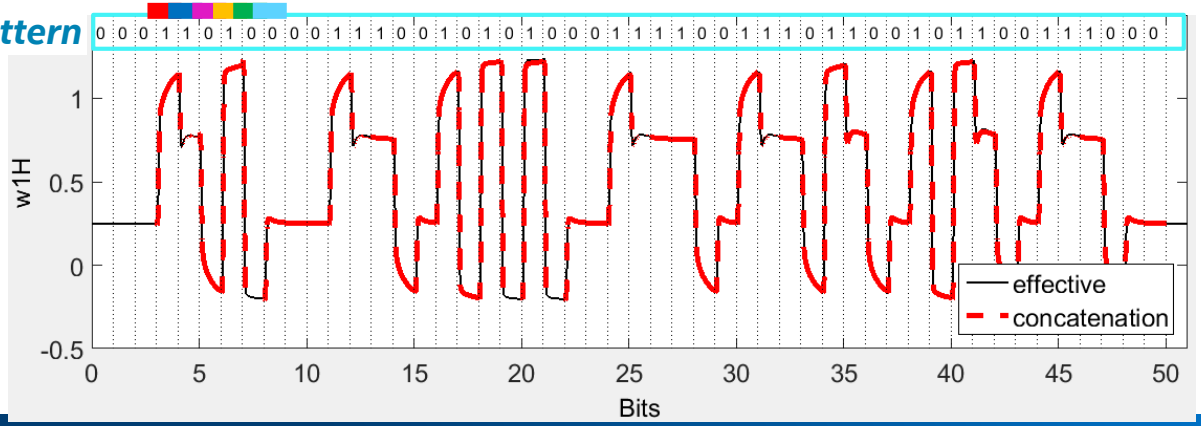
Example: 1 post-tap

$\mathbf{x}(k-1)$		$\mathbf{x}(k)$
00	→	10
01	→	10
01	→	00
10	→	01
10	→	11
11	→	01

For any **given bit-pattern**, the **global weighting functions** are calculated by **concatenation** of the **basis functions.**

Basis Weighting Functions

Generic Bit Pattern



Dynamic Models

$$a_1 = w_{1H} \cdot f_{1SH}(b_1, b_2) + w_{1L} \cdot f_{1SL}(b_1, b_2) + \boldsymbol{f_{1d}(b_1, b_2)}$$

Dynamic Models

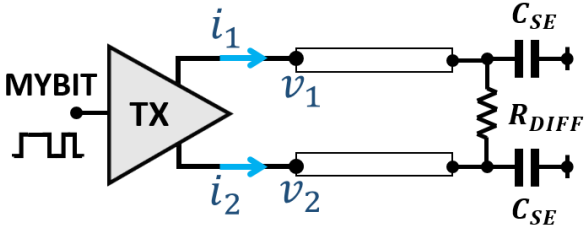
Known for MYBIT – previously calculated

Unknown

$$a_1 = w_{1H} \cdot f_{1SH}(b_1, b_2) + w_{1L} \cdot f_{1SL}(b_1, b_2) + \boldsymbol{f_{1d}(b_1, b_2)}$$

Known, from HSPICE

Known from static surface characterization



Electrical SPICE Testbench for Dynamic Function Characterization

Dynamic Models

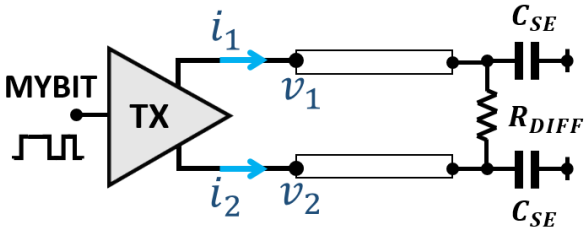
Known for MYBIT – previously calculated

Unknown

$$a_1 = w_{1H} \cdot f_{1SH}(b_1, b_2) + w_{1L} \cdot f_{1SL}(b_1, b_2) + f_{1d}(b_1, b_2)$$

Known, from HSPICE

Known from static surface characterization



Electrical SPICE Testbench for Dynamic Function Characterization

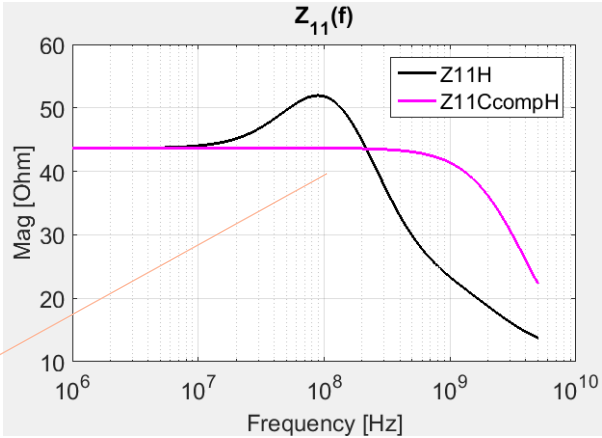
Transient "dynamic" signal

$$f_{1d}(b_1, b_2)$$

TD-VF [*]

Set of Poles & Zeros

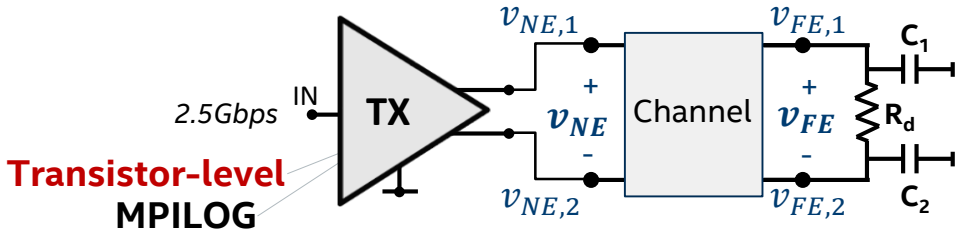
(≠ C_{COMP} approx.)



[*] TD-VF: Time-domain Vector Fitting
Prof. Grivet-Talocia

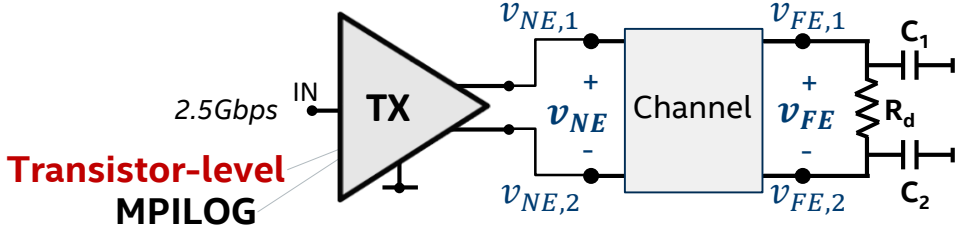
Validations & Results

Validation Results



Setup for the validation of MPILOG models in a "standard" Signal Integrity SPICE simulation

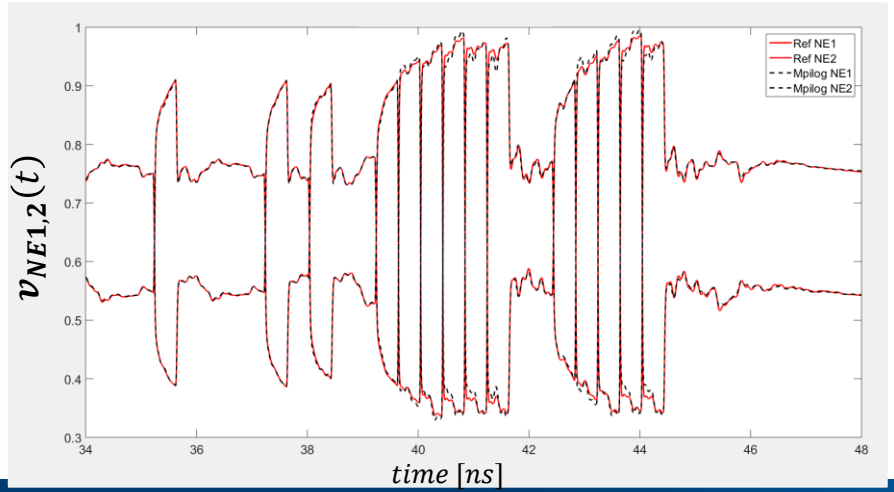
Validation Results



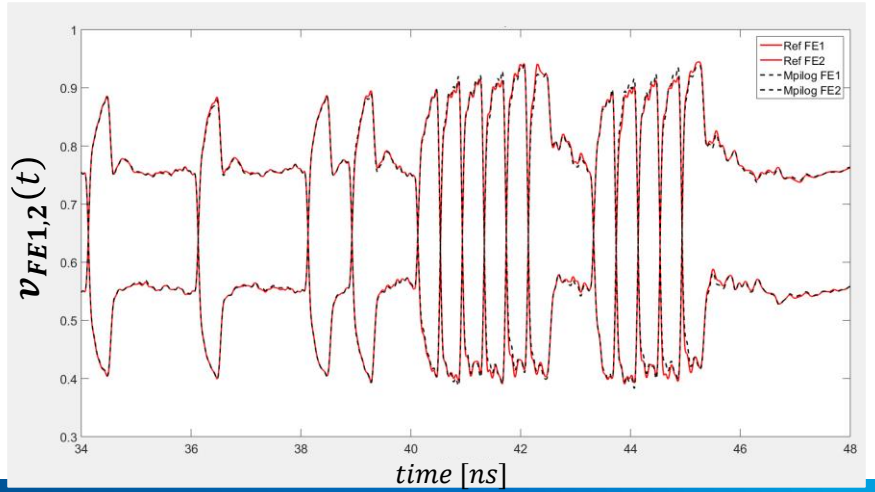
Setup for the validation of MPILOG models in a "standard" Signal Integrity SPICE simulation

Outstanding Accuracy of MPILOG w/ transistor-level simulations
(Near- and Far-end)

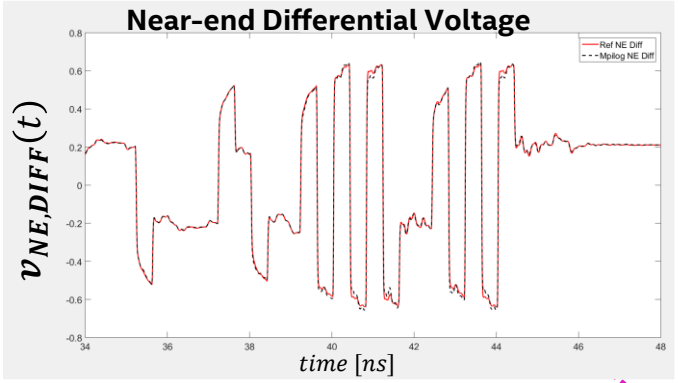
Near-end (single-ended) Signals

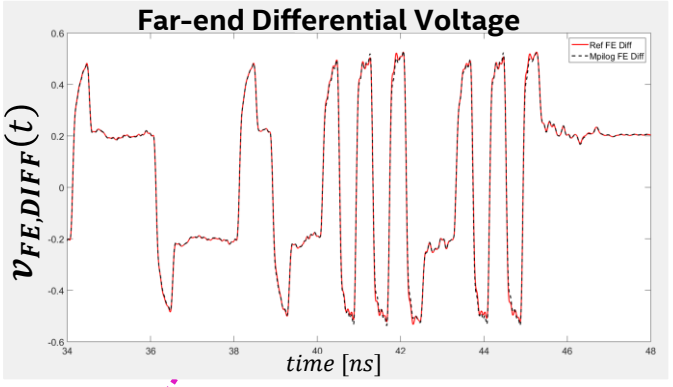


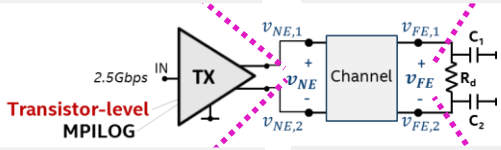
Far-end (single-ended) Signals



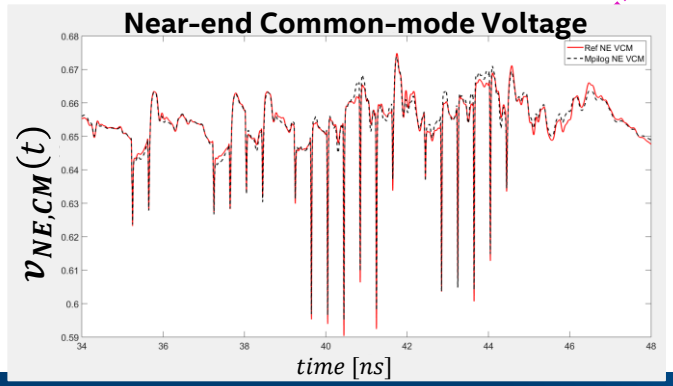
Validation Results

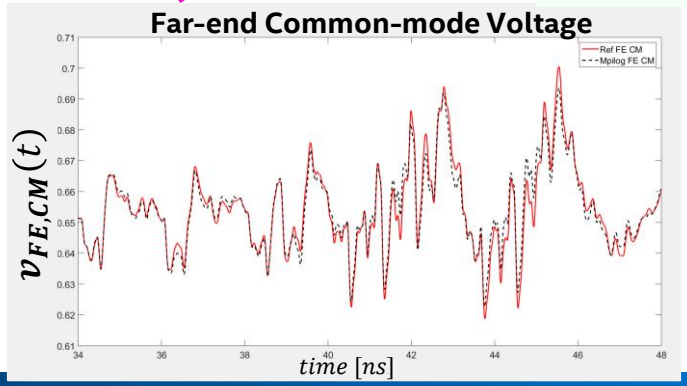




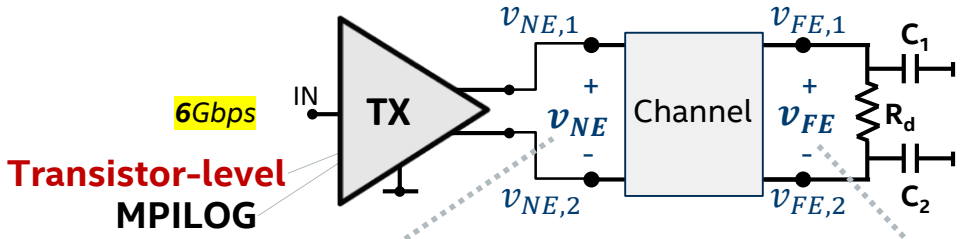


Outstanding Accuracy for both **Differential** and **Common-mode** Signals

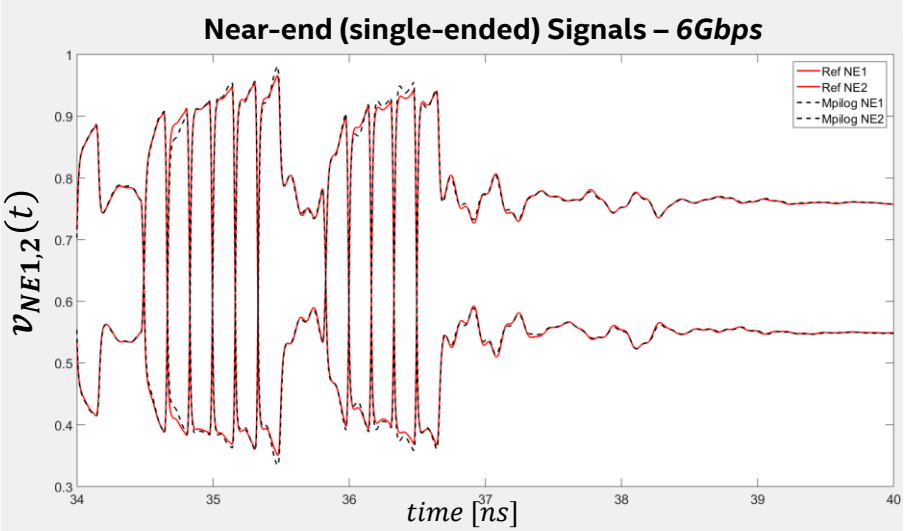


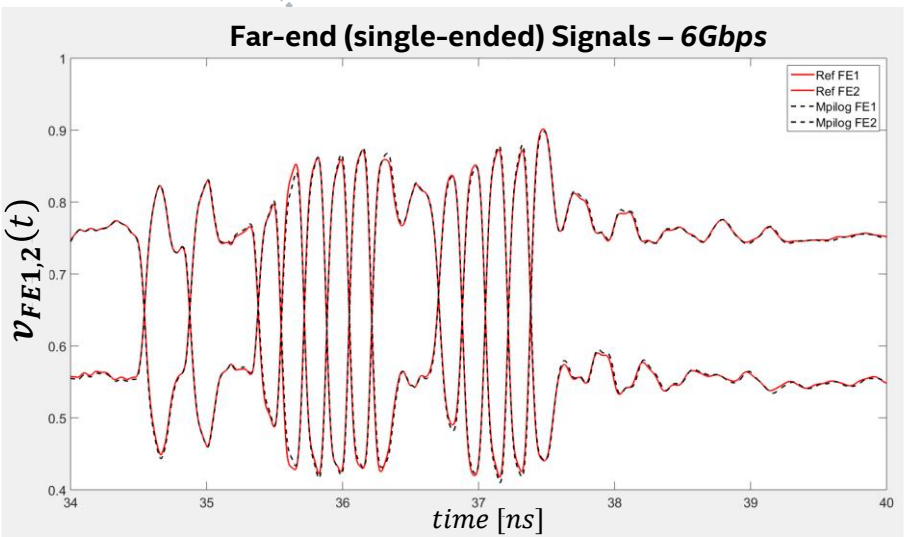


Validation Results

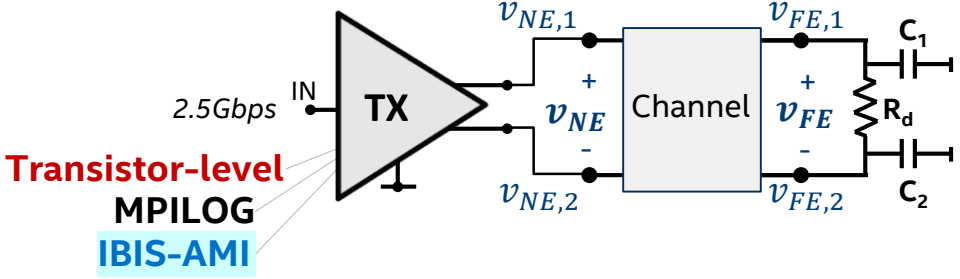


MPILOG Accuracy is confirmed also for higher data-rates





Validation Results

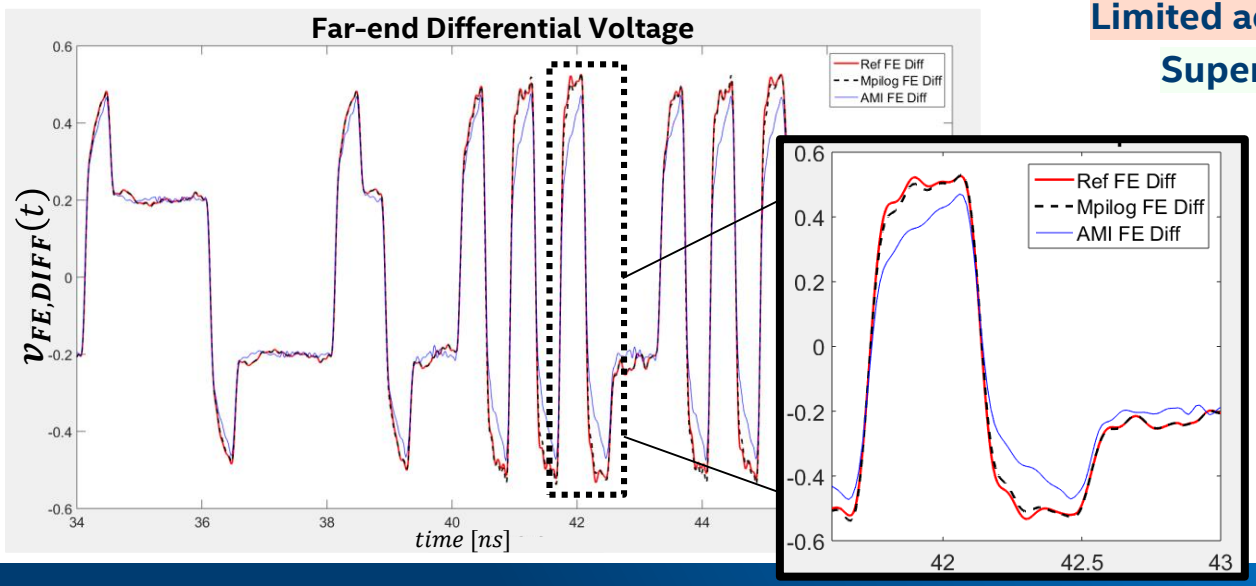


IBIS-AMI: only Differential Signal

MPILOG: both Differential & CM

Limited accuracy of IBIS-AMI *(bit-by-bit)*

Superior Accuracy of MPILOG



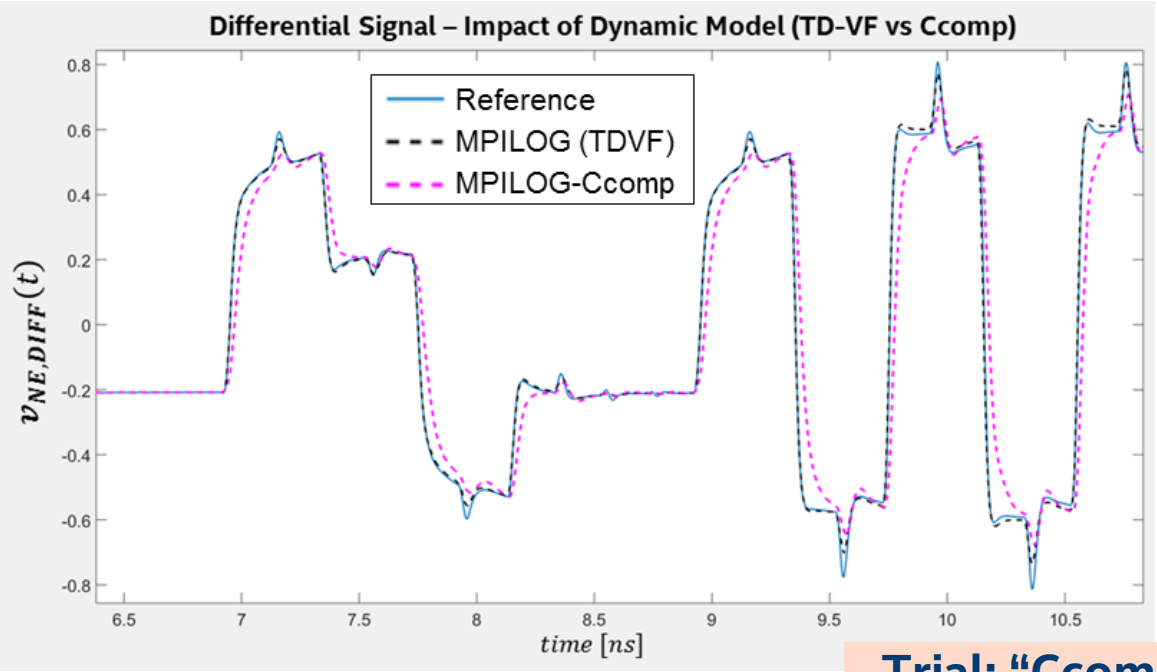
IBIS-AMI & MPILOG: Learnings

	IBIS-AMI	MPILOG
Static Model	Differential R-only	3D Surface (VDIFF & VCM)
Dynamic Model	Ccomp (1-pole)	TD-VF MIMO (n-poles & m-zeros)
V-t	Ramp *(VOH, VOL, Tr, Tf)*	"Database" of Weighting Functions
TX-EQ	Ideal FIR	

Possible sources of inaccuracy in IBIS-AMI:

- *Extend accuracy to VDIFF & CM*
- *C_{COMP} is not sufficient*
- *Better representation of TX EQ impact on rising/falling events is required*

Impact of C_{COMP}-only approximation



Trial: "Ccomp"-only approx. in MPILOG, accuracy worsens!

Conclusions

- In this presentation we've reviewed and discussed **state-of-the-art macromodeling** techniques for **High-speed Differential Transmitters** equipped with **Pre-/De-emphasis** block

- **IBIS/IBIS-AMI** are the most diffused models used and delivered for Signal Integrity simulations
  - **IBIS-AMI** models can run fast and support TX EQ
    Analog LTI (R-C) structure + "TX EQ = **ideal FIR**" may lead to **inaccuracies**
    Only differential signal can be observed.
  - However, accuracy may be ***limited*** in some specific cases.

- **MPILOG** modeling framework has been extended to support HSIO w/ Pre-/De-emphasis.
  - Models = parametric mathematical representations
    (3D surfaces, TDVF MIMO dynamic models, weighting functions to mimic TX EQ, etc.)
  - **Parameters are identified via suitable SPICE simulations** on transistor-level netlist
  - **Outstanding accuracy for both differential and common-mode signals**

- Learnings to be discussed w/ "IBIS Open Forum" to contribute to further IBIS/IBIS-AMI development.

Thank you
for your attention!

gianni.signorini@intel.com