WELCOME FROM MIKE LABONTE, IBIS OPEN FORUM

Ladies and Gentlemen,

As chair of the IBIS Open Forum it is my pleasure to welcome you to the 2016 Asian IBIS Summit in Taipei and to thank you for your presentations and participation. We are grateful to our sponsors Cadence Design Systems, IO Methodology, Peace Giant Corporation, Synopsys, and Xpeedic Technology for making this event possible.

Since 1993 IBIS has provided the digital electronics industry with specifications to make signal, timing, and power integrity analyses much easier and faster. The challenges keep changing and the IBIS community keeps responding with new enhancements to the IBIS family of specifications, which the IC vendors, EDA tool companies, and system designers adopt readily.

Support for IBIS in Asia has been strong, and the IBIS Open Forum looks forward to continued innovation and contributions from technology companies in this part of the world. Thank you!

Michael R LaBonte

Mike LaBonte
Signal Integrity Software (SiSoft)
Chair, IBIS Open Forum

WELCOME FROM MIKE LABONTE, IBIS OPEN FORUM

女士们，先生们，

作为 IBIS 开放论坛的主席我很高兴地欢迎大家来到 2016 年亚洲 IBIS 台北峰会，并感谢您的演讲和参与。我们特别要感谢我们的赞助商 Cadence Design Systems , IO Methodology Inc， Peace Giant Corporation， Synopsys **和** Xpeedic Technology 是他们使本次活动成为可能。

从 1993 年至今，IBIS 为高速数字电路设计的信号，时序和功率完整性分析方面提供了更加容易和快捷电子模型行业规范。为适应不断变化的挑战，在 IC 供应商， EDA 工具公司和系统设计师的共同努力下， IBIS 正在继续不断加入新的功能和完善已有的 IBIS 系列规范。

在亚洲， IBIS 一直受到广泛的支持。 IBIS 开放论坛期待着亚洲工程师有更多的技术创新和贡献。

谢谢！

Michael R LaBonte

Mike LaBonte (迈克 拉邦地)
SiSoft 公司
主席， IBIS 开放论坛

AGENDA AND ORDER OF THE PRESENTATIONS

(The actual agenda might be modified)

IBIS SUMMIT MEETING AGENDA

8:15 **SIGN IN**
- Vendor Tables Open at 8:30

9:00 **WELCOME**
- Mike LaBonte (Chair, IBIS Open Forum)
  (Signal Integrity Software (SiSoft), USA)

9:10 **IBIS Chair's Report 5**
Mike LaBonte (Signal Integrity Software (SiSoft), USA)

9:40 **Case Study: Modeling IBIS for Open_drain True Differetial Pair Buffer 14**
Lance Wang* and Liang Yan**
(*IO Methodology* and **Maxim Integrated, USA)

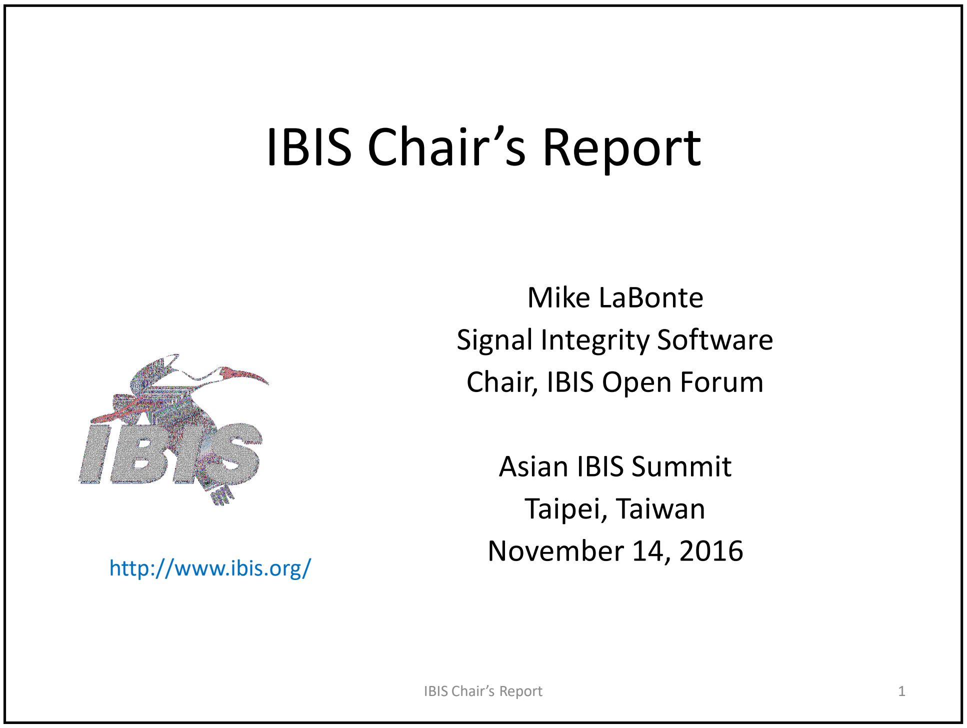
10:10 **Differential Modeling Flow with series Model in Verilog-A 22**
Wei-hsing Huang* and Sanjeev Gupta**
(*SPISim, USA and **Sigintegrity Solutions, India)

10:40 **BREAK** (Refreshments and Vendor Tables)

11:00 **IBIS AMI Model Generation with Quality 32**
Skipper Liang (Cadence Design Systems, ROC)

12:00 **FREE BUFFET LUNCH** (Hosted by Sponsors)
- Vendor Tables

AGENDA AND ORDER OF THE PRESENTATIONS (Continued)

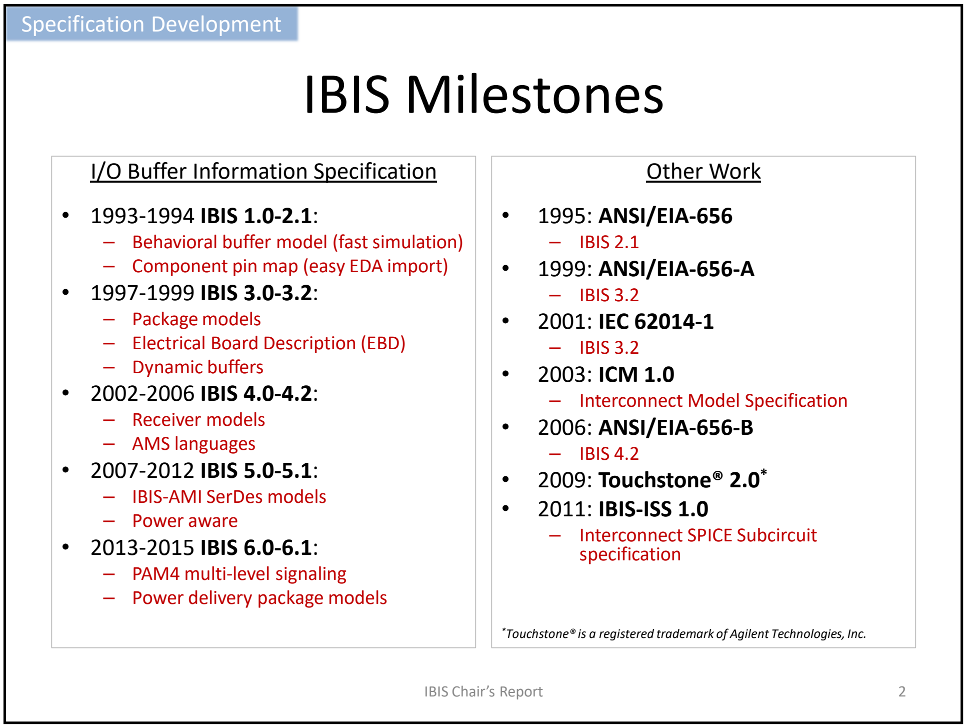


IBIS Chair's Report

Mike LaBonte
Signal Integrity Software
Chair, IBIS Open Forum

Asian IBIS Summit
Taipei, Taiwan
November 14, 2016

http://www.ibis.org/



Specification Development

IBIS Milestones

<u>I/O Buffer Information Specification</u>

- 1993-1994 **IBIS 1.0-2.1**:
  - Behavioral buffer model (fast simulation)
  - Component pin map (easy EDA import)
- 1997-1999 **IBIS 3.0-3.2**:
  - Package models
  - Electrical Board Description (EBD)
  - Dynamic buffers
- 2002-2006 **IBIS 4.0-4.2**:
  - Receiver models
  - AMS languages
- 2007-2012 **IBIS 5.0-5.1**:
  - IBIS-AMI SerDes models
  - Power aware
- 2013-2015 **IBIS 6.0-6.1**:
  - PAM4 multi-level signaling
  - Power delivery package models

<u>Other Work</u>

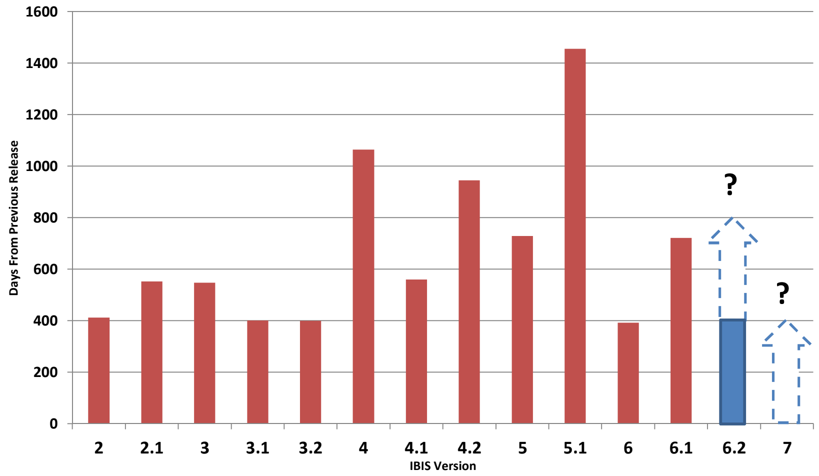
- 1995: **ANSI/EIA-656**
  - IBIS 2.1
- 1999: **ANSI/EIA-656-A**
  - IBIS 3.2
- 2001: **IEC 62014-1**
  - IBIS 3.2
- 2003: **ICM 1.0**
  - Interconnect Model Specification
- 2006: **ANSI/EIA-656-B**
  - IBIS 4.2
- 2009: **Touchstone® 2.0***
- 2011: **IBIS-ISS 1.0**
  - Interconnect SPICE Subcircuit specification

**Touchstone® is a registered trademark of Agilent Technologies, Inc.*

Specification Development

IBIS Version Development

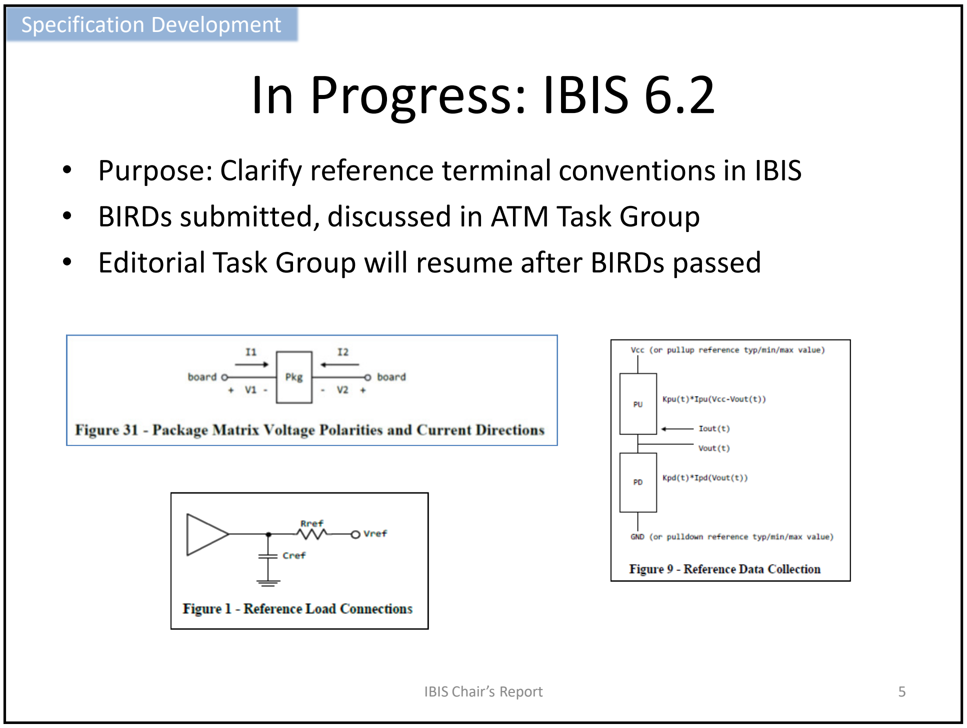
As of 23-Oct-2016



Specification Development

Work In Progress

- Advanced Technology Modeling Task Group
  - IBIS 6.2 dedicated to reference node clarifications
  - Back-channel support (BIRD147.3)
  - C_comp model enhancements
  - Redriver flow enhancements
- Interconnect Task Group
  - External Package/on-die models using IBIS-ISS and Touchstone®
- IBIS Quality Task Group
  - IBISCHK enhancements and documentation



Specification Development

In Progress: IBIS 6.2

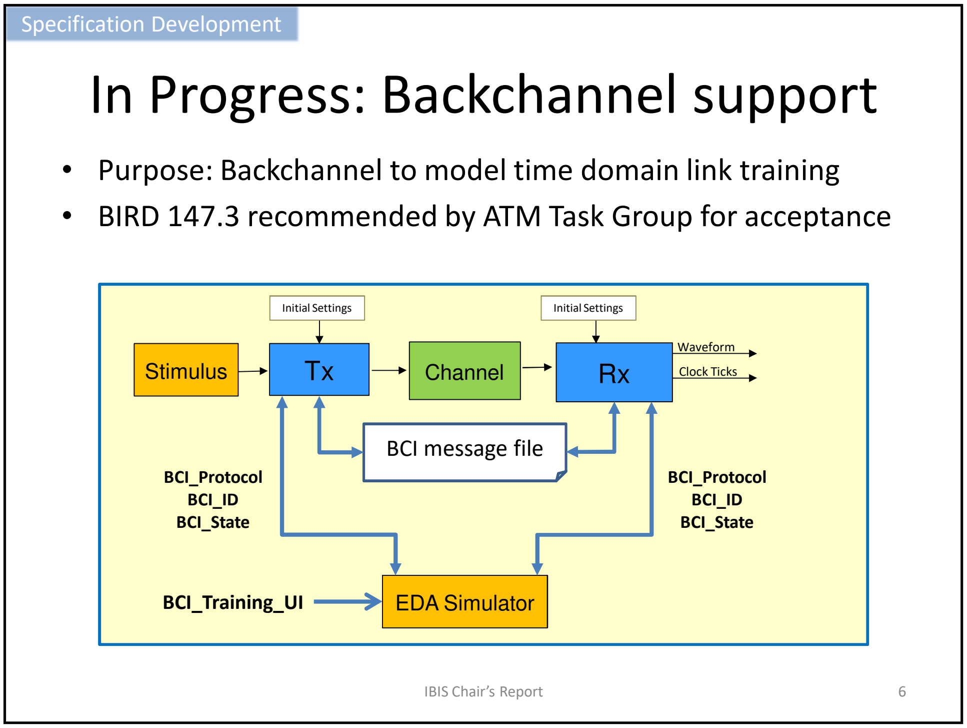
- Purpose: Clarify reference terminal conventions in IBIS
- BIRDs submitted, discussed in ATM Task Group
- Editorial Task Group will resume after BIRDs passed

Figure 31 - Package Matrix Voltage Polarities and Current Directions

Figure 1 - Reference Load Connections

Vcc (or pullup reference typ/min/max value)
PU
Kpu(t)*Ipu(Vcc-Vout(t))
Iout(t)
Vout(t)
PD
Kpd(t)*Ipd(Vout(t))
GND (or pulldown reference typ/min/max value)

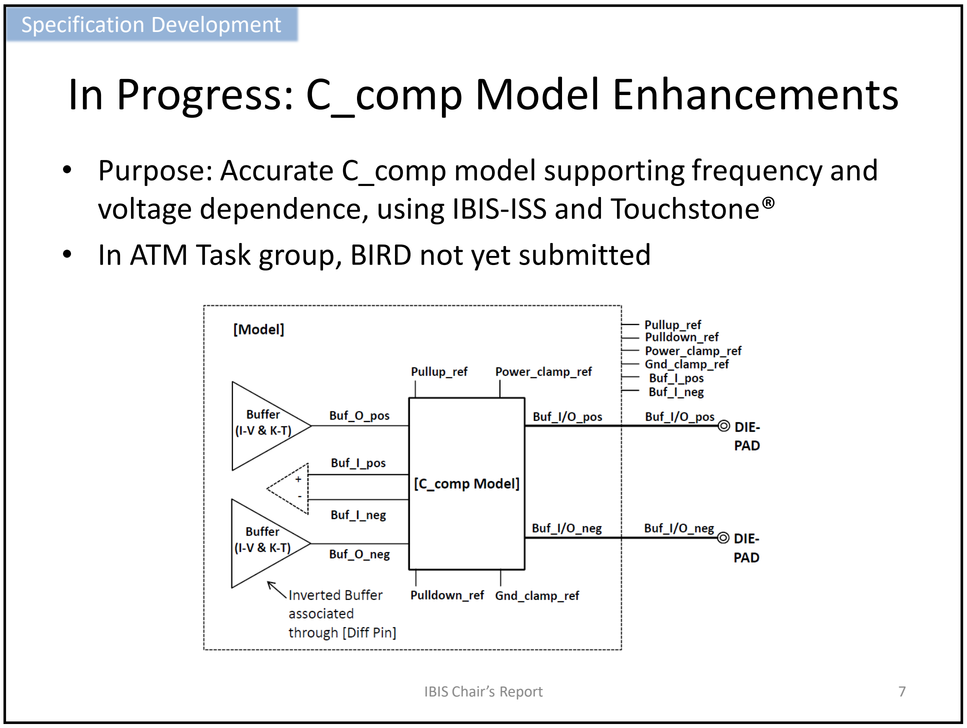
Figure 9 - Reference Data Collection



Specification Development

In Progress: Backchannel support

- Purpose: Backchannel to model time domain link training
- BIRD 147.3 recommended by ATM Task Group for acceptance

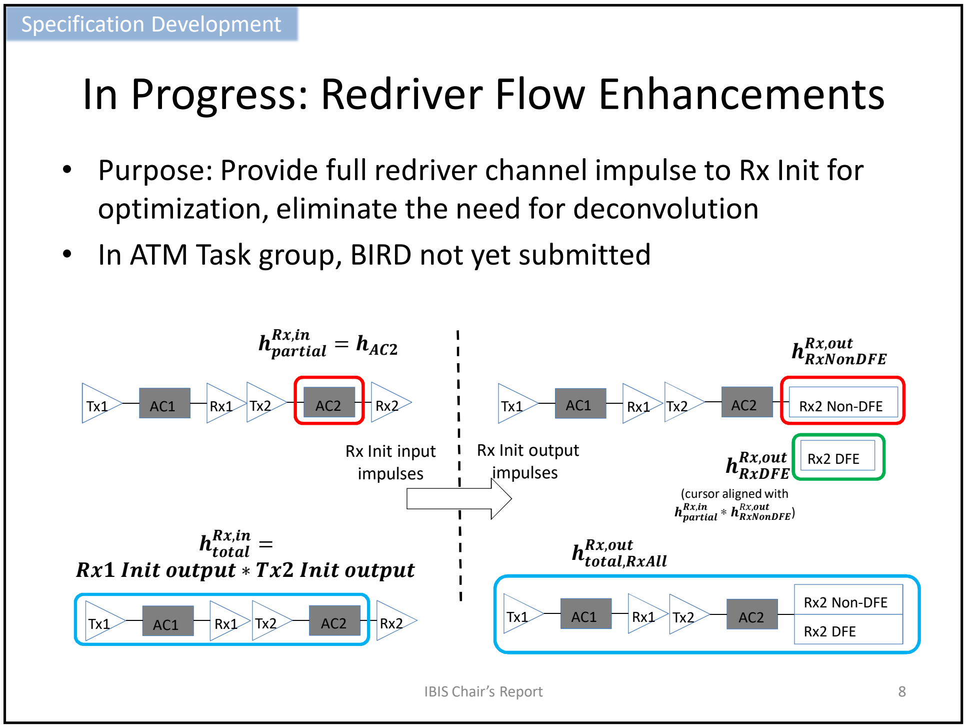


Specification Development

In Progress: C_comp Model Enhancements

- Purpose: Accurate C_comp model supporting frequency and voltage dependence, using IBIS-ISS and Touchstone®
- In ATM Task group, BIRD not yet submitted

IBIS Chair's Report 7

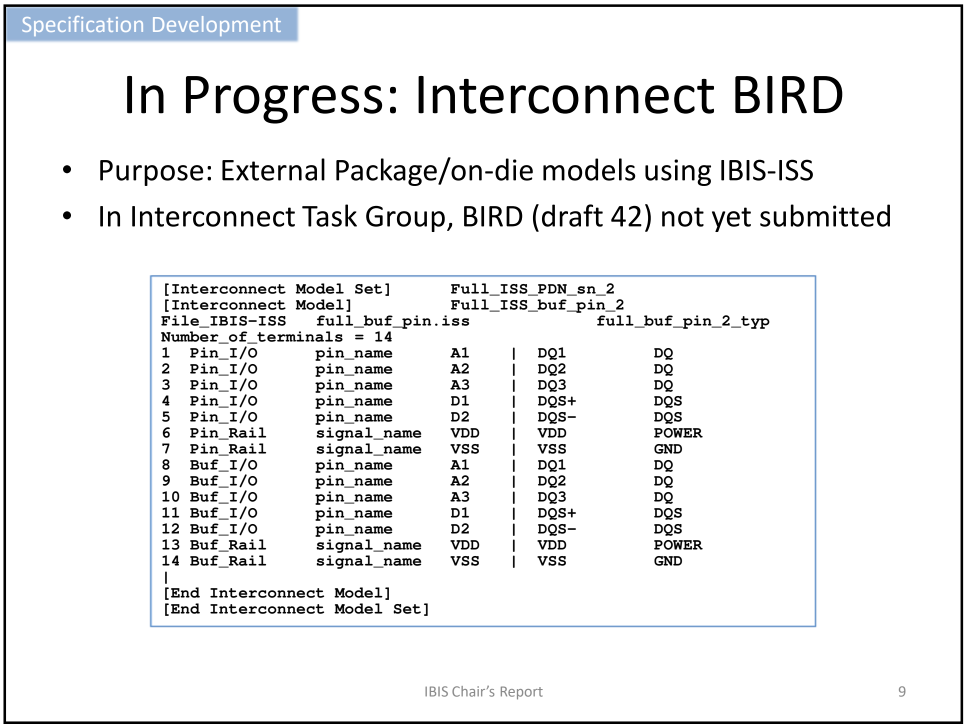


Specification Development

In Progress: Redriver Flow Enhancements

- Purpose: Provide full redriver channel impulse to Rx Init for optimization, eliminate the need for deconvolution
- In ATM Task group, BIRD not yet submitted

IBIS Chair's Report 8



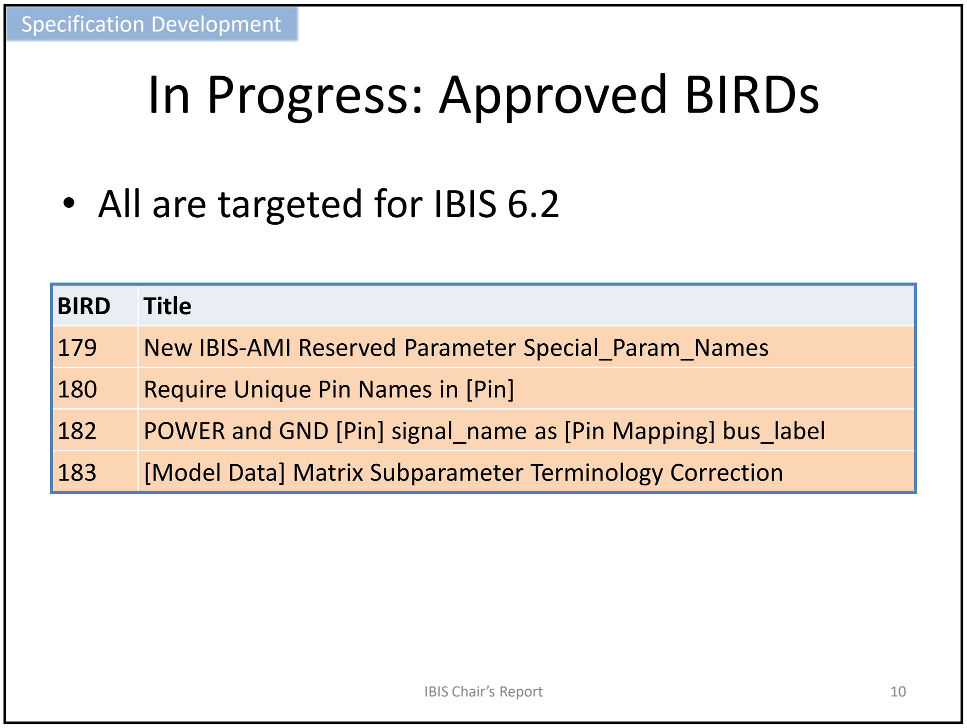
Specification Development

In Progress: Interconnect BIRD

- Purpose: External Package/on-die models using IBIS-ISS
- In Interconnect Task Group, BIRD (draft 42) not yet submitted

```
[Interconnect Model Set]      Full_ISS_PDN_sn_2
[Interconnect Model]          Full_ISS_buf_pin_2
File_IBIS-ISS   full_buf_pin.iss             full_buf_pin_2_typ
Number_of_terminals = 14
1  Pin_I/O      pin_name      A1    |  DQ1         DQ
2  Pin_I/O      pin_name      A2    |  DQ2         DQ
3  Pin_I/O      pin_name      A3    |  DQ3         DQ
4  Pin_I/O      pin_name      D1    |  DQS+        DQS
5  Pin_I/O      pin_name      D2    |  DQS-        DQS
6  Pin_Rail     signal_name   VDD   |  VDD         POWER
7  Pin_Rail     signal_name   VSS   |  VSS         GND
8  Buf_I/O      pin_name      A1    |  DQ1         DQ
9  Buf_I/O      pin_name      A2    |  DQ2         DQ
10 Buf_I/O      pin_name      A3    |  DQ3         DQ
11 Buf_I/O      pin_name      D1    |  DQS+        DQS
12 Buf_I/O      pin_name      D2    |  DQS-        DQS
13 Buf_Rail     signal_name   VDD   |  VDD         POWER
14 Buf_Rail     signal_name   VSS   |  VSS         GND
|
[End Interconnect Model]
[End Interconnect Model Set]
```

IBIS Chair's Report 9



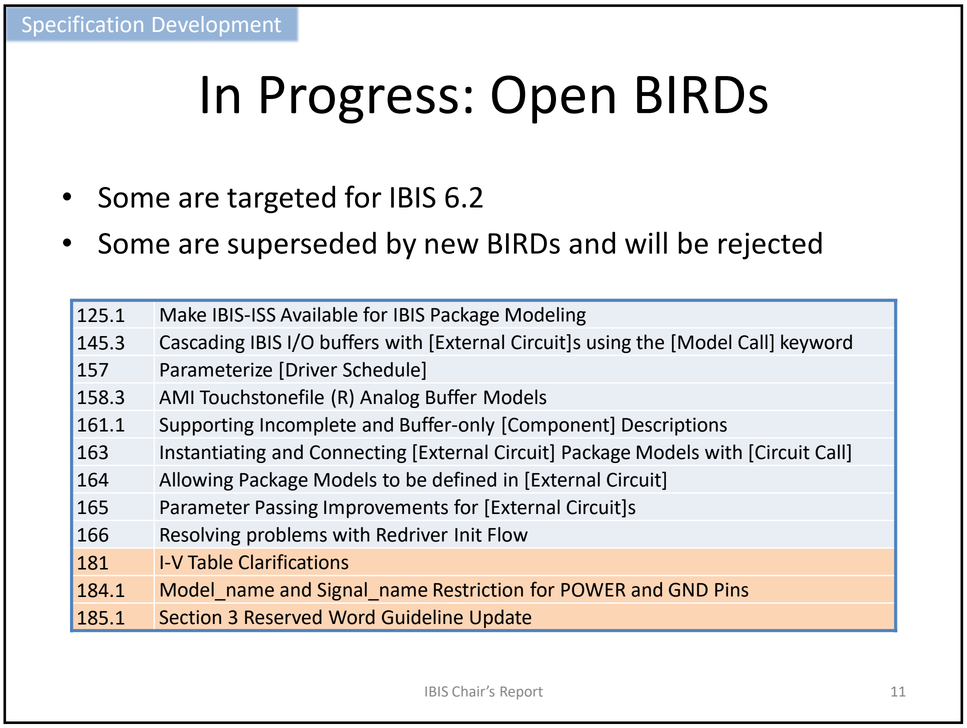
Specification Development

In Progress: Approved BIRDs

- All are targeted for IBIS 6.2

BIRD	Title
179	New IBIS-AMI Reserved Parameter Special_Param_Names
180	Require Unique Pin Names in [Pin]
182	POWER and GND [Pin] signal_name as [Pin Mapping] bus_label
183	[Model Data] Matrix Subparameter Terminology Correction

IBIS Chair's Report 10

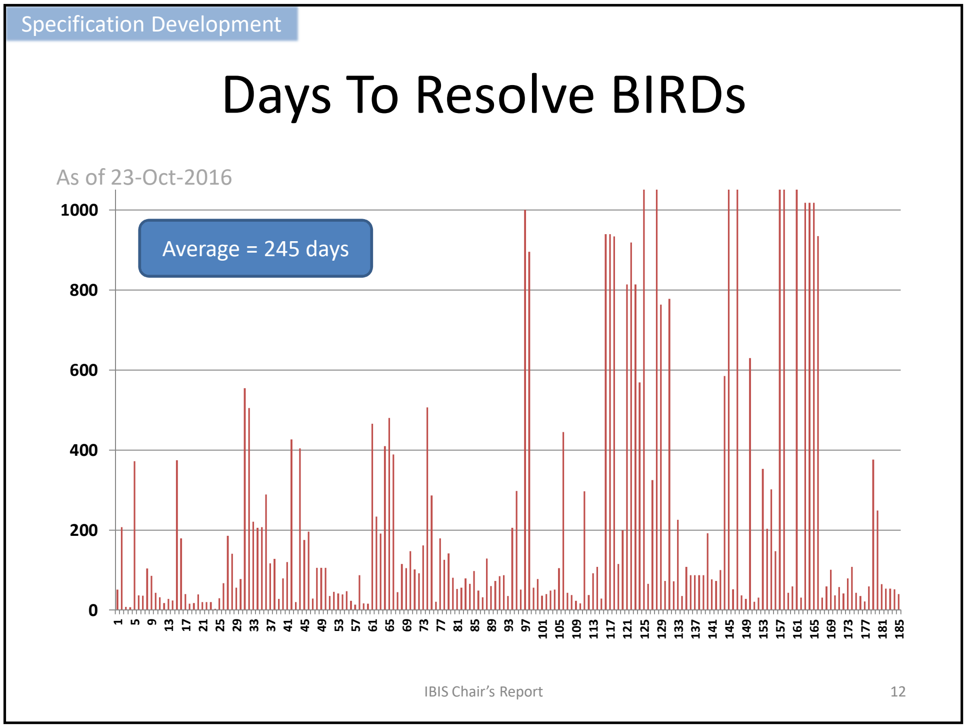


Specification Development

In Progress: Open BIRDs

- Some are targeted for IBIS 6.2
- Some are superseded by new BIRDs and will be rejected

125.1	Make IBIS-ISS Available for IBIS Package Modeling
145.3	Cascading IBIS I/O buffers with [External Circuit]s using the [Model Call] keyword
157	Parameterize [Driver Schedule]
158.3	AMI Touchstonefile (R) Analog Buffer Models
161.1	Supporting Incomplete and Buffer-only [Component] Descriptions
163	Instantiating and Connecting [External Circuit] Package Models with [Circuit Call]
164	Allowing Package Models to be defined in [External Circuit]
165	Parameter Passing Improvements for [External Circuit]s
166	Resolving problems with Redriver Init Flow
181	I-V Table Clarifications
184.1	Model_name and Signal_name Restriction for POWER and GND Pins
185.1	Section 3 Reserved Word Guideline Update



Specification Development

Days To Resolve BIRDs

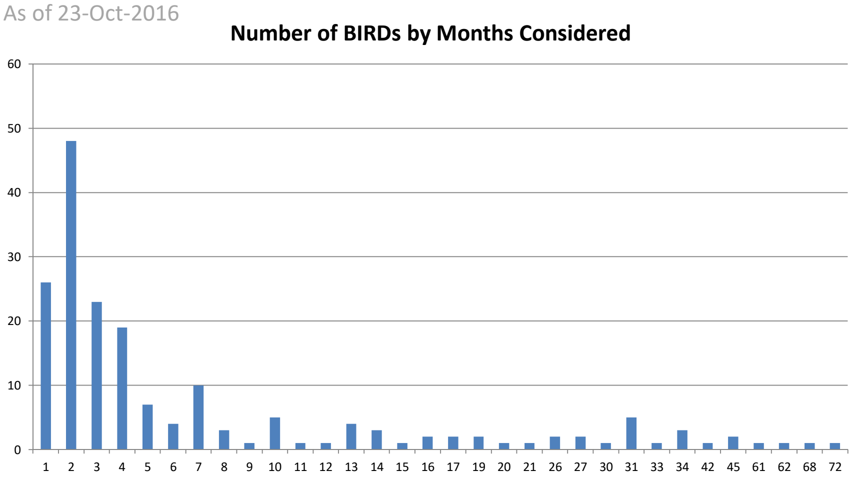
As of 23-Oct-2016

Average = 245 days

Specification Development

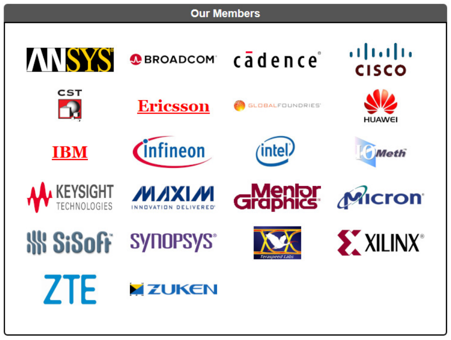
Most BIRDs Resolved < 9 Months

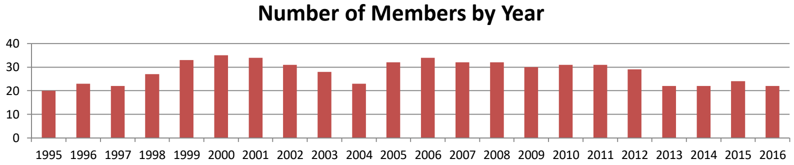
As of 23-Oct-2016

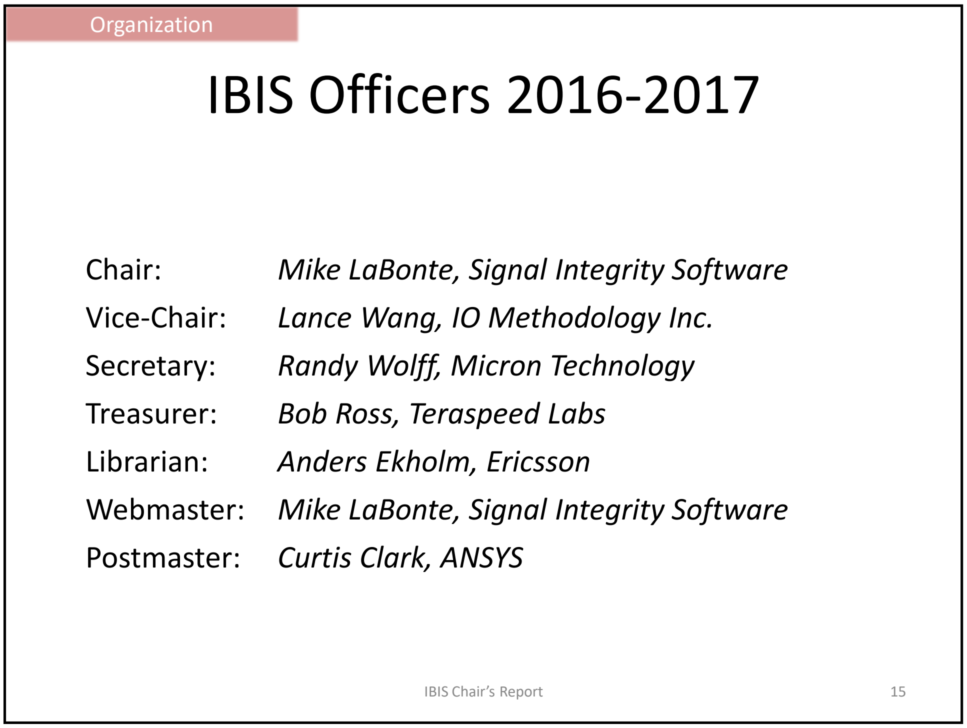


Organization

22 IBIS Members



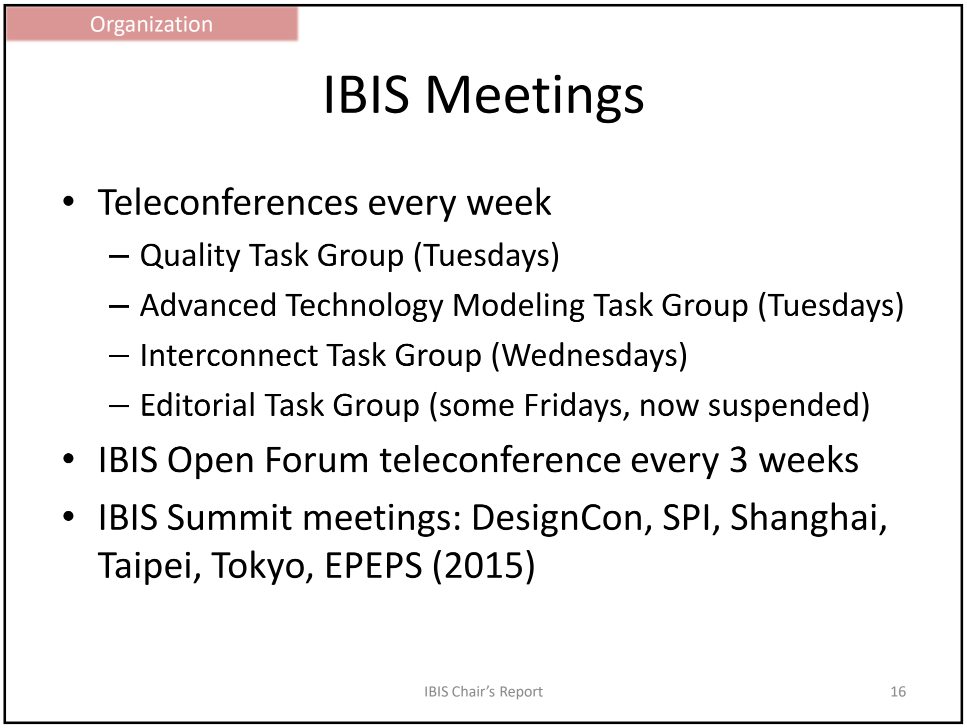




Organization

IBIS Officers 2016-2017

Chair:	*Mike LaBonte, Signal Integrity Software*
Vice-Chair:	*Lance Wang, IO Methodology Inc.*
Secretary:	*Randy Wolff, Micron Technology*
Treasurer:	*Bob Ross, Teraspeed Labs*
Librarian:	*Anders Ekholm, Ericsson*
Webmaster:	*Mike LaBonte, Signal Integrity Software*
Postmaster:	*Curtis Clark, ANSYS*



Organization

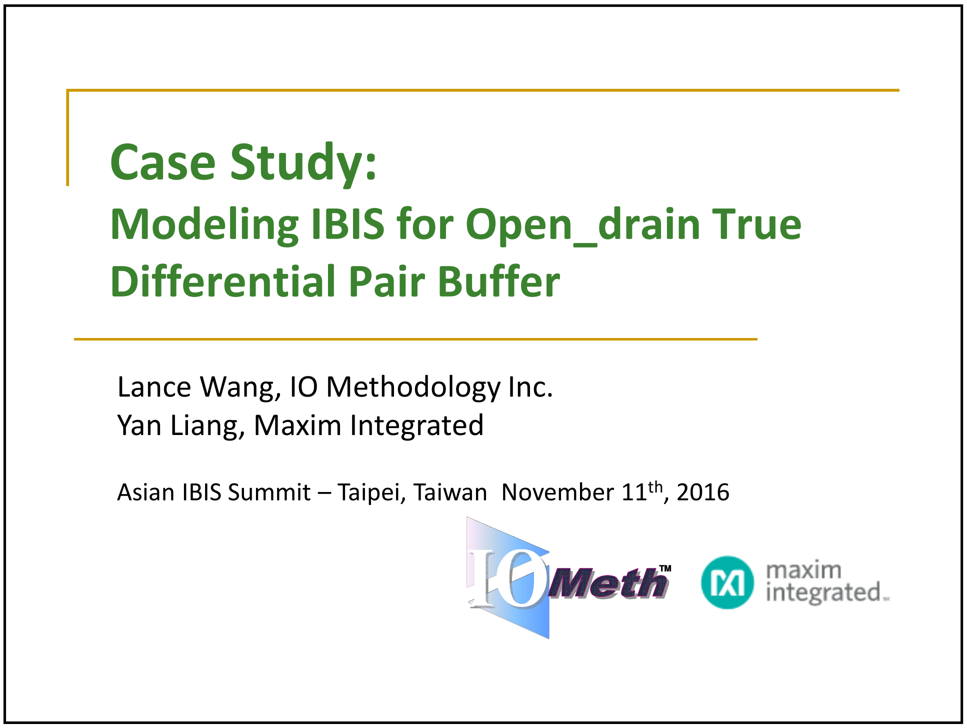
IBIS Meetings

- Teleconferences every week
  - Quality Task Group (Tuesdays)
  - Advanced Technology Modeling Task Group (Tuesdays)
  - Interconnect Task Group (Wednesdays)
  - Editorial Task Group (some Fridays, now suspended)
- IBIS Open Forum teleconference every 3 weeks
- IBIS Summit meetings: DesignCon, SPI, Shanghai, Taipei, Tokyo, EPEPS (2015)

[Thank You]

IBIS Open Forum:
Web: http://www.ibis.org
Email: ibis-info@freelists.org

We welcome participation by all IBIS model makers, EDA tool vendors, IBIS model users, and interested parties.

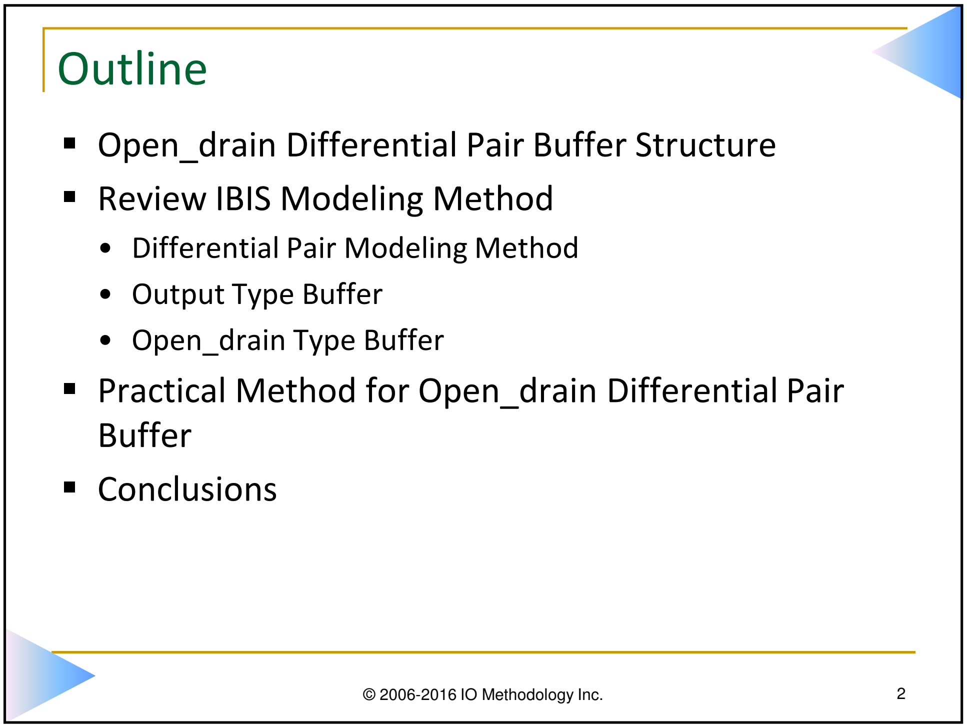


Case Study:

Modeling IBIS for Open_drain True Differential Pair Buffer

Lance Wang, IO Methodology Inc.
Yan Liang, Maxim Integrated

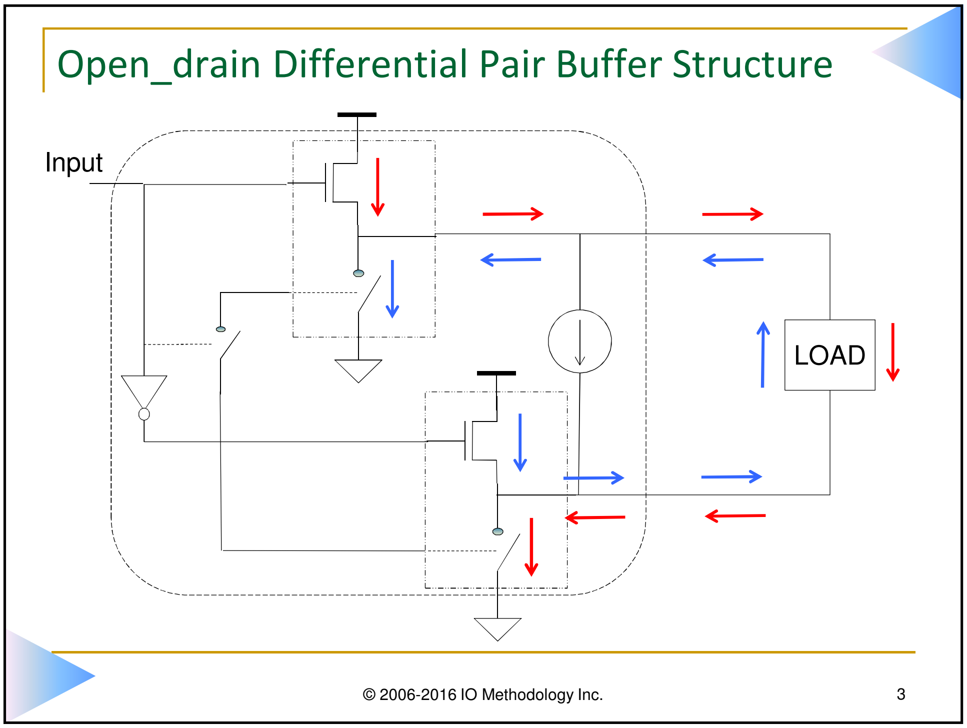
Asian IBIS Summit – Taipei, Taiwan November 11th, 2016



Outline

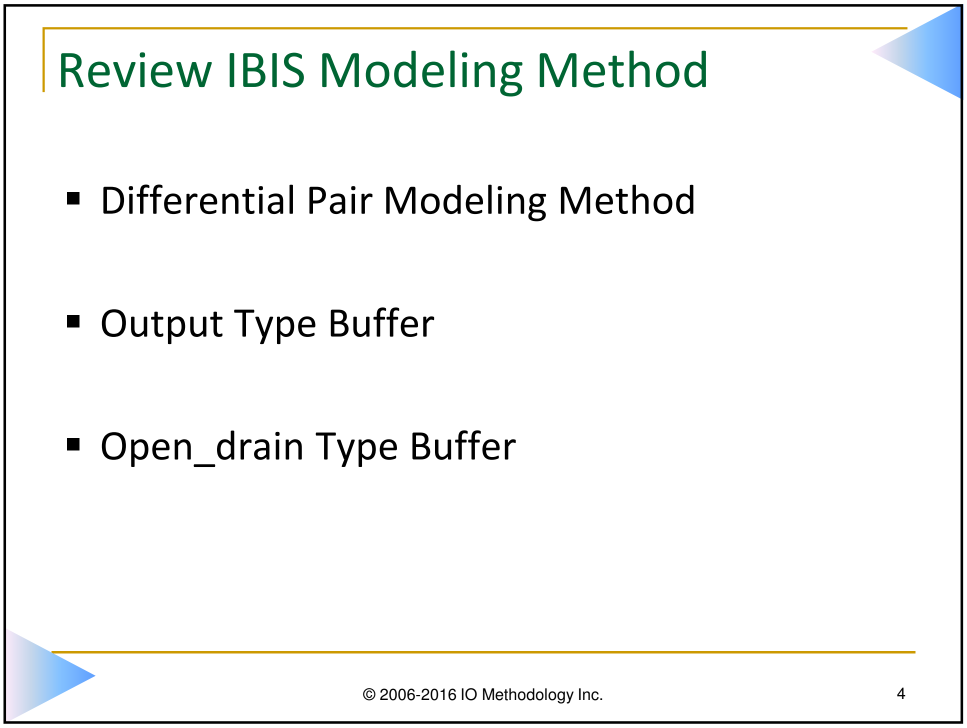
- Open_drain Differential Pair Buffer Structure
- Review IBIS Modeling Method
  - Differential Pair Modeling Method
  - Output Type Buffer
  - Open_drain Type Buffer
- Practical Method for Open_drain Differential Pair Buffer
- Conclusions

© 2006-2016 IO Methodology Inc.



Open_drain Differential Pair Buffer Structure

© 2006-2016 IO Methodology Inc.



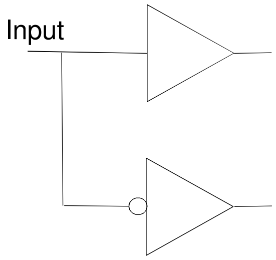
Review IBIS Modeling Method

- Differential Pair Modeling Method
- Output Type Buffer
- Open_drain Type Buffer

© 2006-2016 IO Methodology Inc.

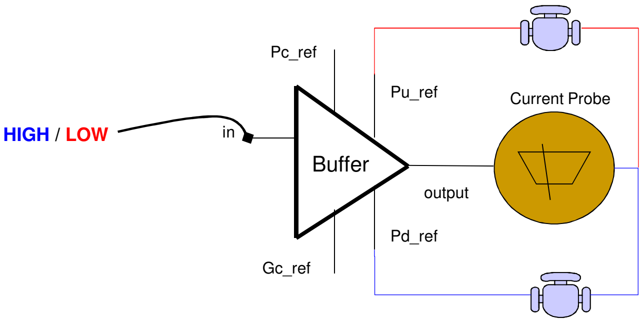
Differential Pair Modeling Method

- IBIS uses two single-end models to be a differential pair
- IBIS uses [Diff Pin] to define two pins to be a differential pair pins
- Uses two opposite inputs as required



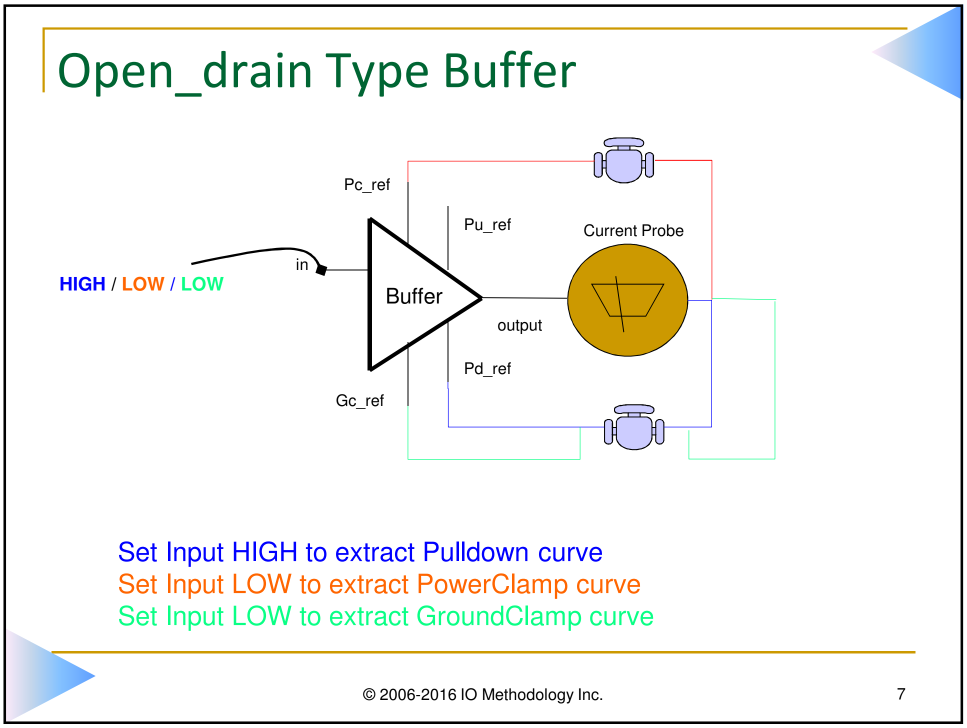
© 2006-2016 IO Methodology Inc.

Output Type Buffer (non-inverting)



Set Input LOW to extract Pullup curve
Set Input HIGH to extract Pulldown curve

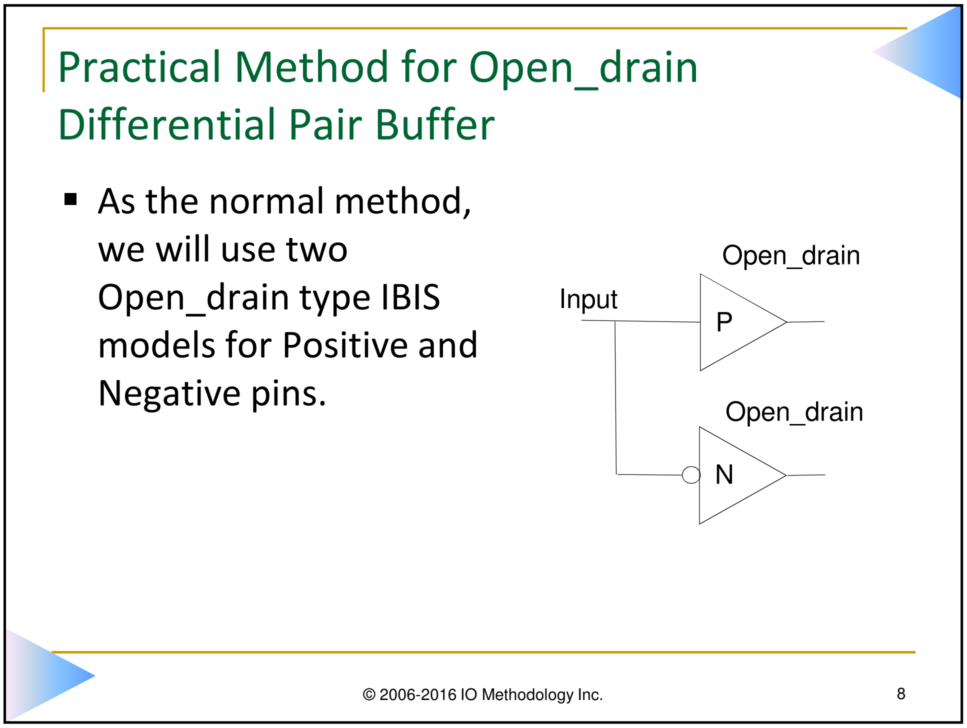
© 2006-2016 IO Methodology Inc.



Open_drain Type Buffer

Set Input HIGH to extract Pulldown curve
Set Input LOW to extract PowerClamp curve
Set Input LOW to extract GroundClamp curve

© 2006-2016 IO Methodology Inc. 7



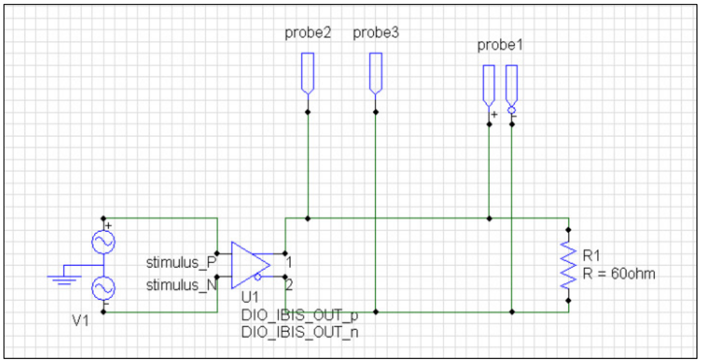
Practical Method for Open_drain Differential Pair Buffer

- As the normal method, we will use two Open_drain type IBIS models for Positive and Negative pins.

© 2006-2016 IO Methodology Inc. 8

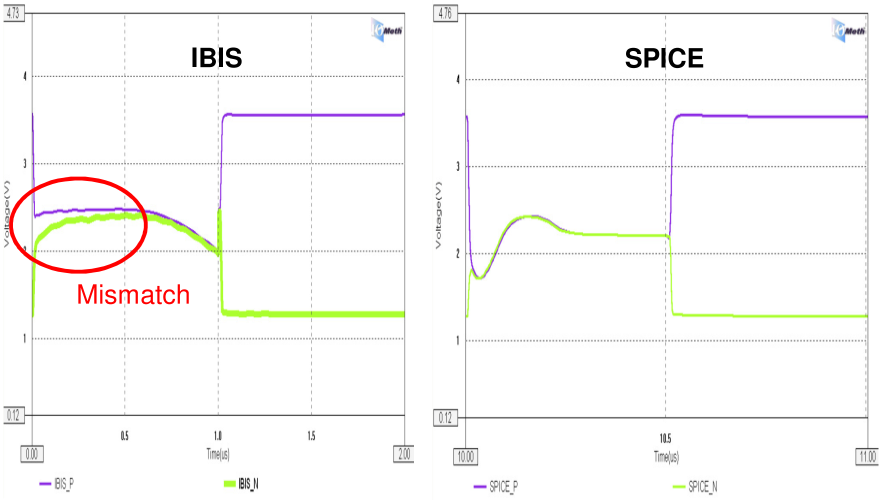
Let’s validate

The Topology for Validation

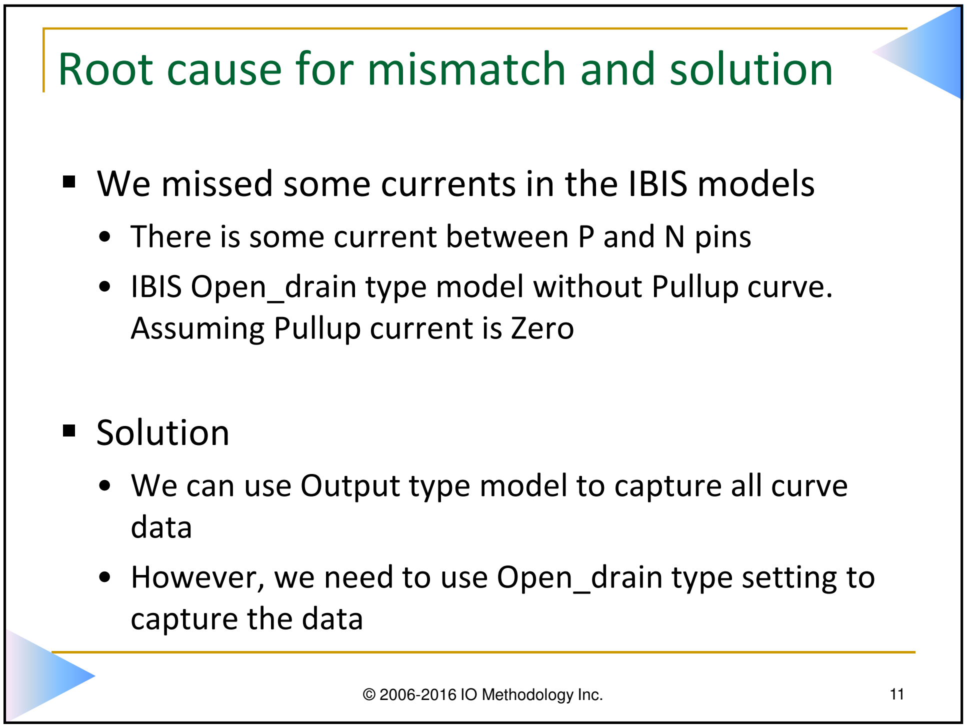


© 2006-2016 IO Methodology Inc.

Validation Results



© 2006-2016 IO Methodology Inc.

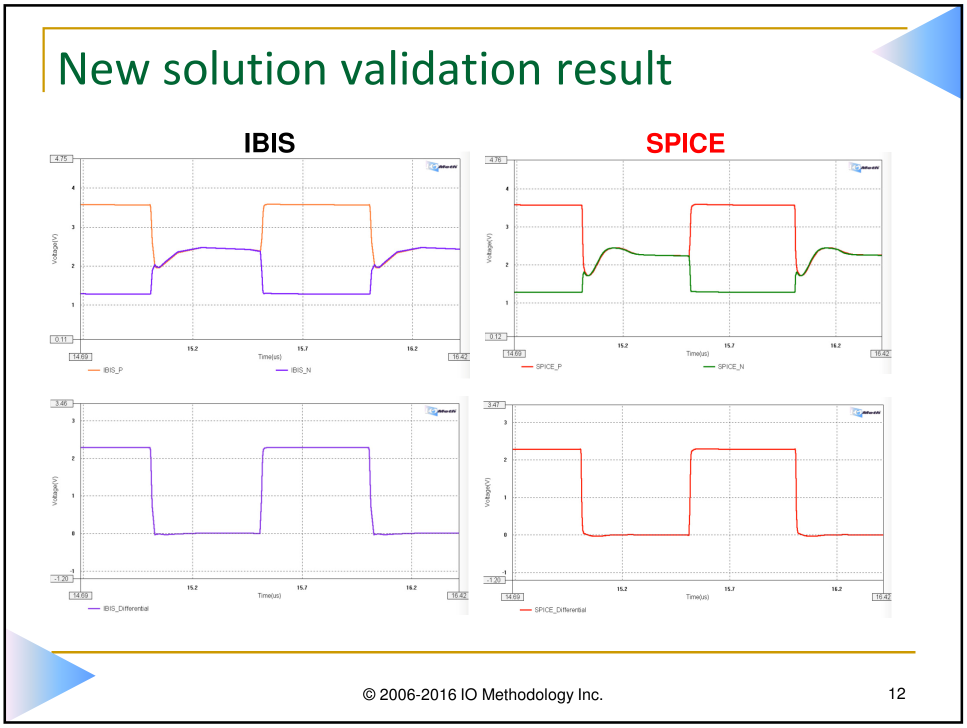


Root cause for mismatch and solution

- We missed some currents in the IBIS models
  - There is some current between P and N pins
  - IBIS Open_drain type model without Pullup curve. Assuming Pullup current is Zero

- Solution
  - We can use Output type model to capture all curve data
  - However, we need to use Open_drain type setting to capture the data

© 2006-2016 IO Methodology Inc.



New solution validation result

© 2006-2016 IO Methodology Inc.

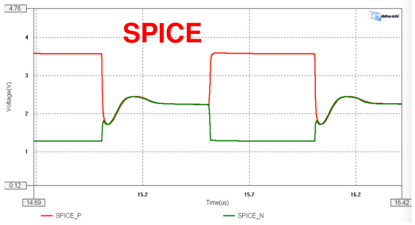
Conclusion

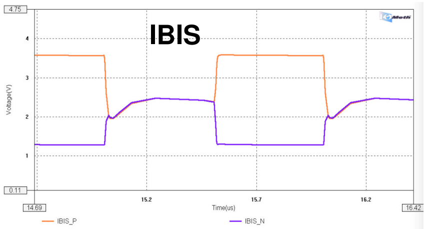
- Open_drain differential pair is a special case for IBIS modeling
  - IBIS Open_drain model is without the Pullup data
  - We need to use Output/IO type IBIS model to capture the Pullup data for this kind of differential pair buffer
    - However we need to IBIS Open_drain modeling setting for extractions
- IBIS C_comp needs to improve to be matched better

© 2006-2016 IO Methodology Inc.

C_comp

- Currently, IBIS Spec only allows 4 values at the most
- We might need to have more C_comp values according DC levels and frequency changes
- Study is in process …



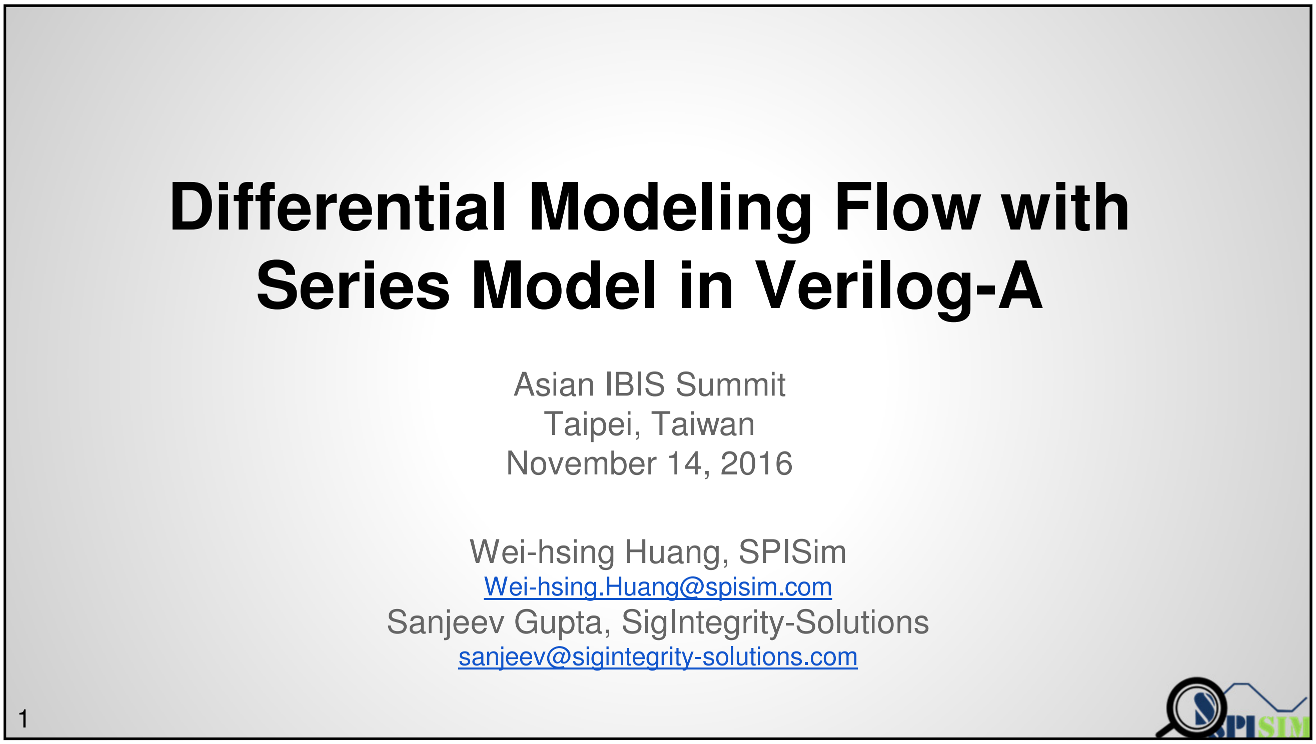


© 2006-2016 IO Methodology Inc.

The Modeling Specialist

IOMeth™

http://www.iometh.com

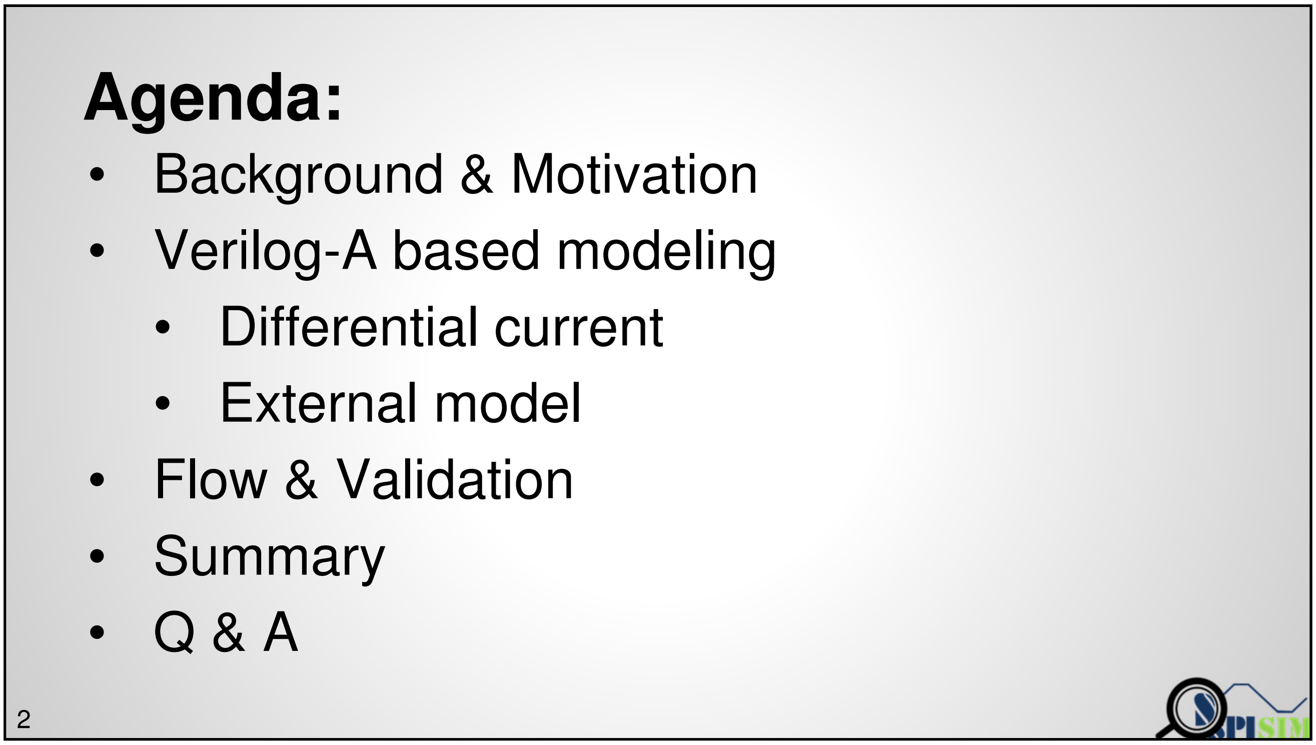


Differential Modeling Flow with Series Model in Verilog-A

Asian IBIS Summit
Taipei, Taiwan
November 14, 2016

Wei-hsing Huang, SPISim
Wei-hsing.Huang@spisim.com
Sanjeev Gupta, SigIntegrity-Solutions
sanjeev@sigintegrity-solutions.com

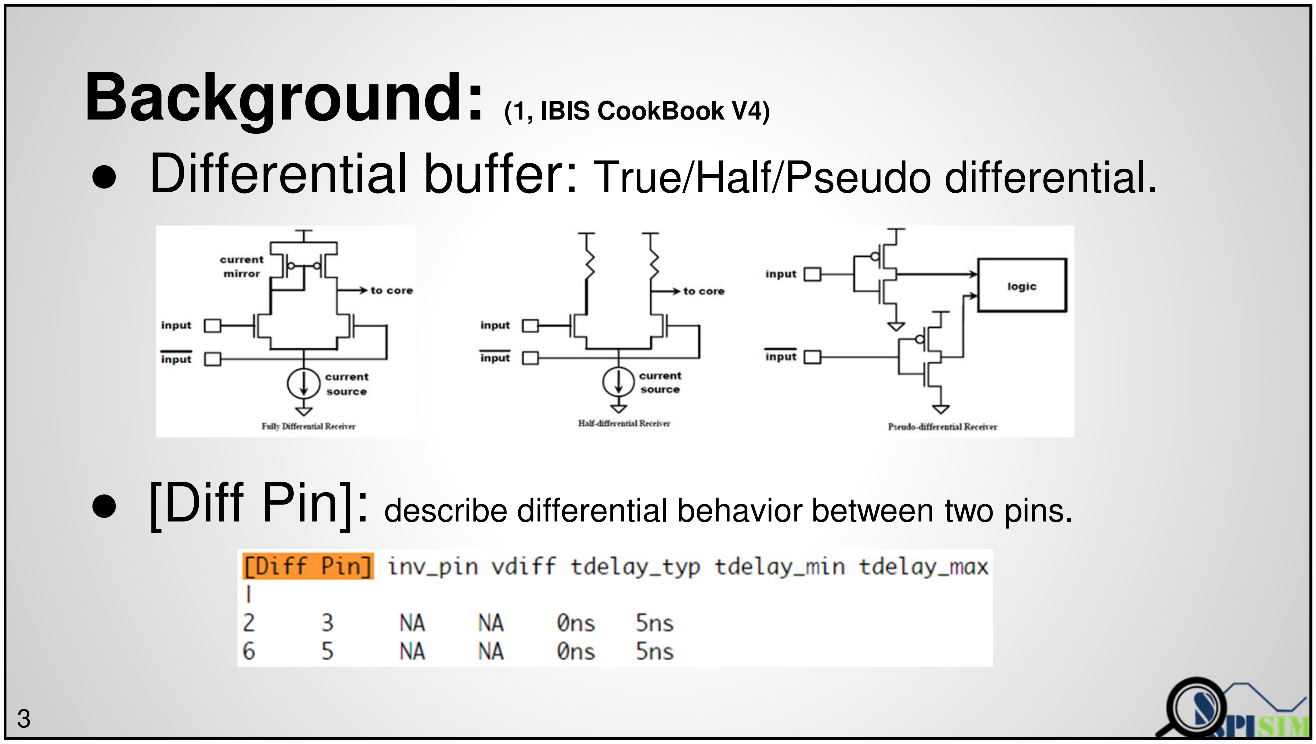
1



Agenda:

- Background & Motivation
- Verilog-A based modeling
  - Differential current
  - External model
- Flow & Validation
- Summary
- Q & A

2



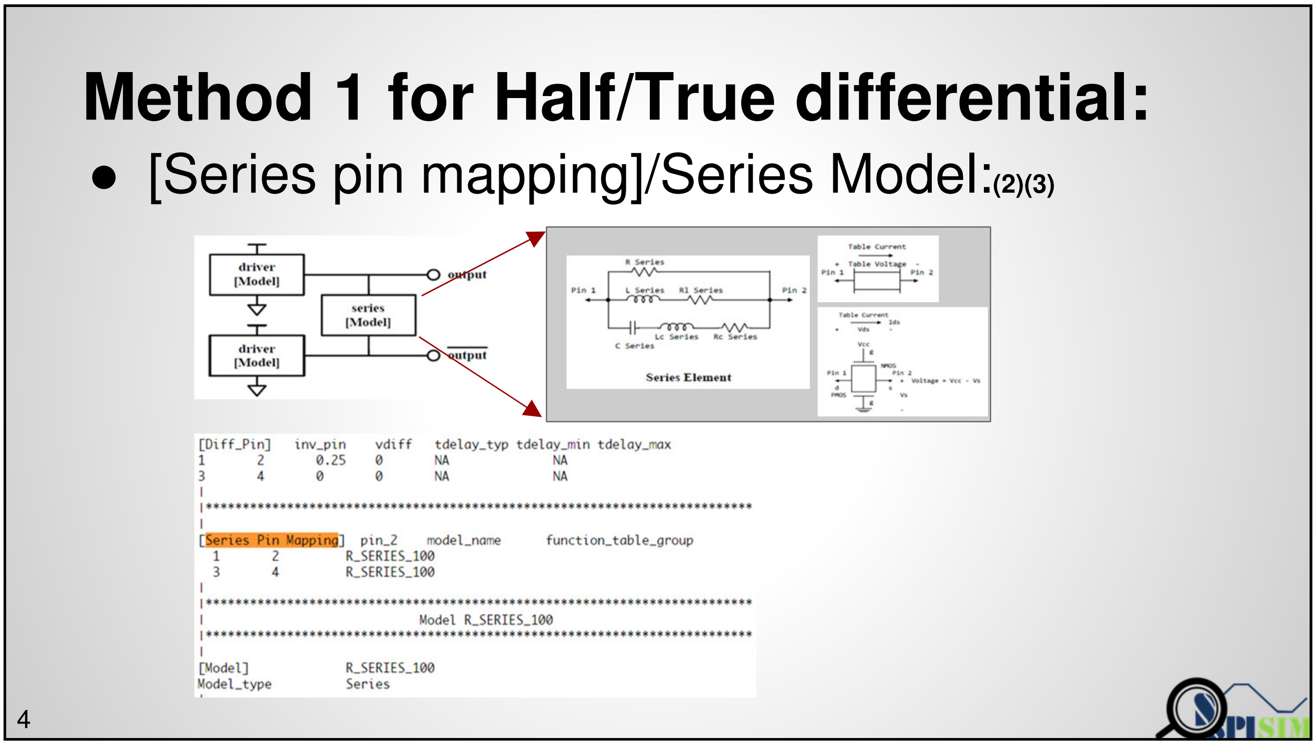
Background: (1, IBIS CookBook V4)

- Differential buffer: True/Half/Pseudo differential.

- [Diff Pin]: describe differential behavior between two pins.

```
[Diff Pin] inv_pin vdiff tdelay_typ tdelay_min tdelay_max
|
2      3      NA     NA     0ns    5ns
6      5      NA     NA     0ns    5ns
```

3



Method 1 for Half/True differential:

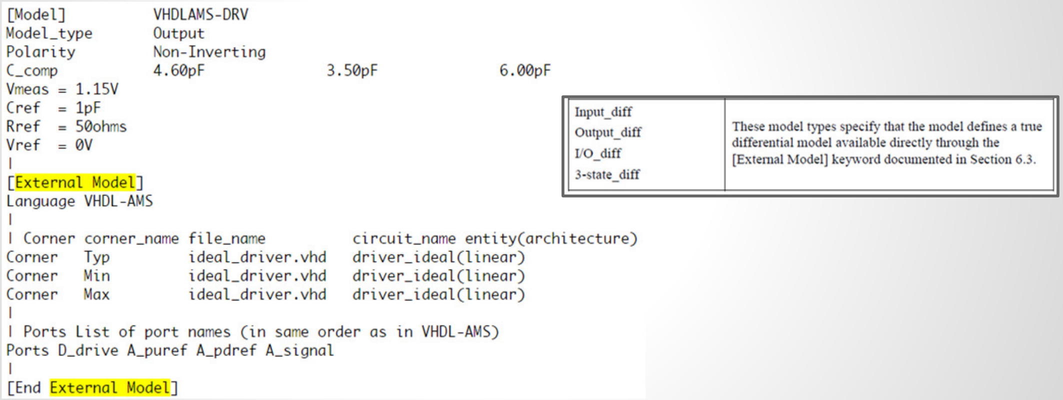
- [Series pin mapping]/Series Model:(2)(3)

```
[Diff_Pin]    inv_pin    vdiff    tdelay_typ tdelay_min tdelay_max
1        2      0.25     0        NA              NA
3        4      0        0        NA              NA
|
|**************************************************************************
|
[Series Pin Mapping]  pin_2     model_name       function_table_group
  1        2        R_SERIES_100
  3        4        R_SERIES_100
|
|**************************************************************************
|                            Model R_SERIES_100
|**************************************************************************
|
[Model]                R_SERIES_100
Model_type             Series
```

4

Method 2 for Half/True differential:

- [External Model]: Spice/VHDL-AMS/Verilog-AMS/IBIS-ISS**(4)**



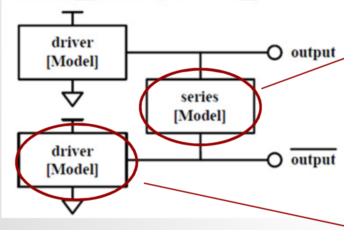
```
[Model]          VHDLAMS-DRV
Model_type       Output
Polarity         Non-Inverting
C_comp           4.60pF           3.50pF           6.00pF
Vmeas = 1.15V
Cref  = 1pF
Rref  = 50ohms
Vref  = 0V
|
[External Model]
Language VHDL-AMS
|
| Corner corner_name file_name         circuit_name entity(architecture)
Corner   Typ         ideal_driver.vhd   driver_ideal(linear)
Corner   Min         ideal_driver.vhd   driver_ideal(linear)
Corner   Max         ideal_driver.vhd   driver_ideal(linear)
|
| Ports List of port names (in same order as in VHDL-AMS)
Ports D_drive A_puref A_pdref A_signal
|
[End External Model]
```

Input_diff Output_diff I/O_diff 3-state_diff	These model types specify that the model defines a true differential model available directly through the [External Model] keyword documented in Section 6.3.

Method 1 & 2 can be used together!

Background:

- Data extraction:
  - ○ Common-mode current
  - ○ Differential current
  - ○ C_comp & C_diff



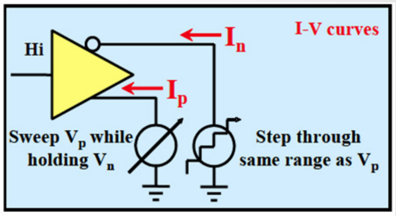

I-V curves
Hi
I_n
I_p
Sweep V_p while holding V_n
Step through same range as V_p

Figure 4.13 – I-V Table Extraction Fixture for a Differential Buffer

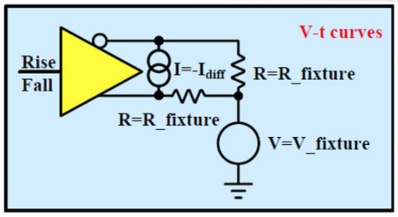
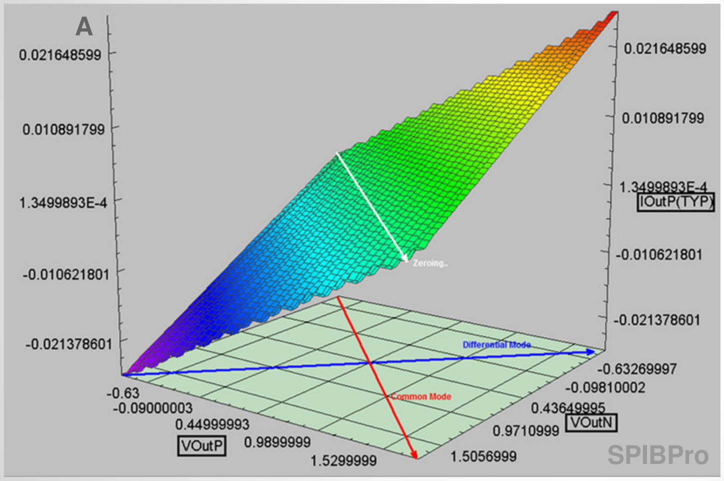
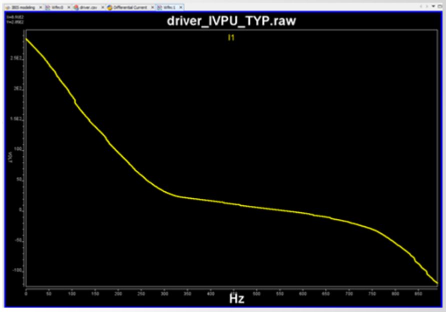


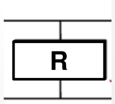
Figure 4.18 – V-T Table Extraction Fixture for a Differential Buffer

Design 1:





I Comm. Mode as PU/PD, PC/GC

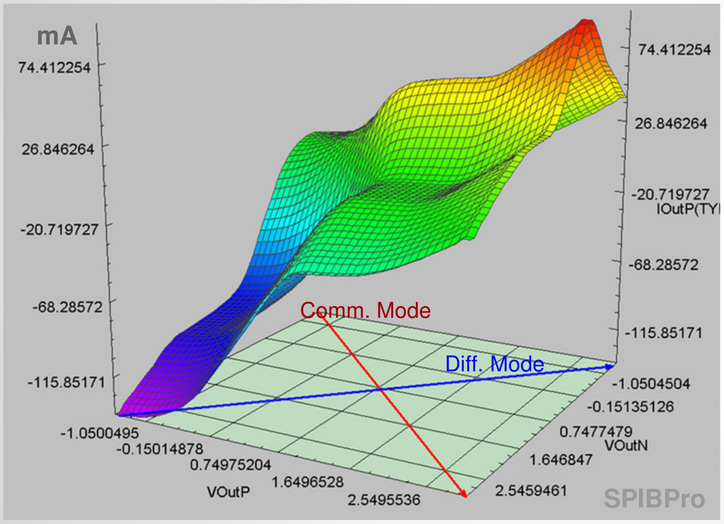


A simple “Series R” can describe this particular design

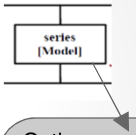
Shifted surface as differential series elem.

7

Design 2:



Need to describe this surface data in design 2 for V-T extraction
Affect DC steady states (e.g. mismatch)

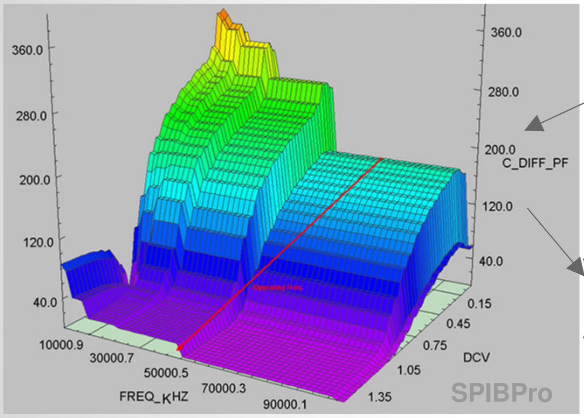


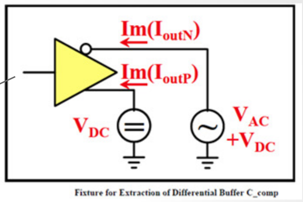
Options:

- Surface fit in MSE sense:
  - Need to check residue
  - Translate to EFGH elements
- Series MOSFET
  - Or Series current
- **Behavioral model?**

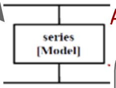
8

Design 2 C_diff:





Fixture for Extraction of Differential Buffer C_comp



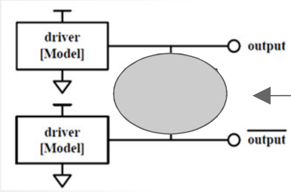
Need to describe this surface data for V-T extraction as well
Affects final transient accuracy

Options:

- Summarize and add single Series C
- **Behavioral model?**

Motivations:

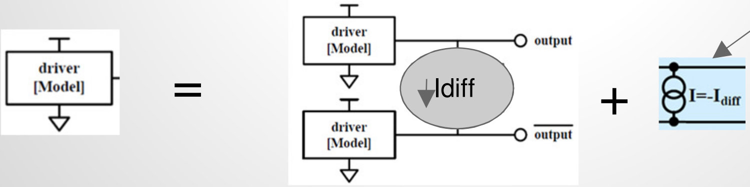
- Limitation of "Generic" series model:



Series Elements:

1. Rigid syntax
2. Condition is fixed (e.g. no polarity)
3. Modeling flow interruption
   a. Surface fit ?
   b. Generate tentative series-elem?

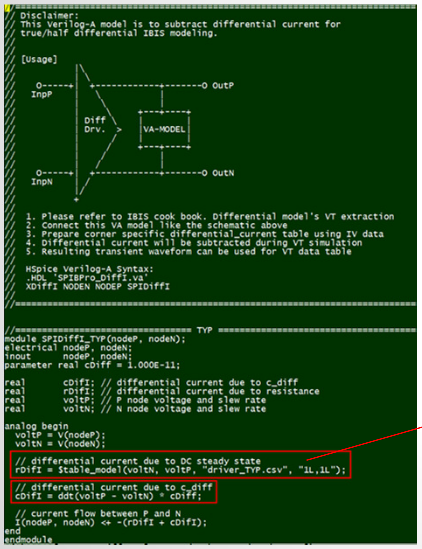
- Accuracy of transient data for V-T extraction:



?

Inaccurate TR data will pollute waveform...

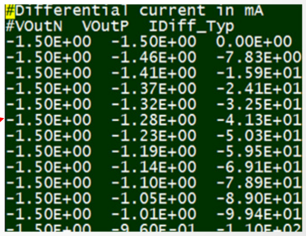
Verilog-A based Diff. current model:



```
//==========================================================================
// Disclaimer:
// This Verilog-A model is to subtract differential current for
// true/half differential IBIS modeling.
//
//
// [Usage]
//          |\
//          | \
//   O-----+|  +-------------+--------O OutP
//   InpP   |   \             |
//          |    \            |
//          |     \       +---+----+
//          | Diff \      |        |
//          | Drv.  >     |VA-MODEL|
//          |      /      |        |
//          |     /       +---+----+
//          |    /            |
//          |   /             |
//   O-----+|  +-------------+--------O OutN
//   InpN   | /
//          |/
//          +
//
//  1. Please refer to IBIS cook book. Differential model's VT extraction
//  2. Connect this VA model like the schematic above
//  3. Prepare corner specific differential_current table using IV data
//  4. Differential current will be subtracted during VT simulation
//  5. Resulting transient waveform can be used for VT data table
//
//  HSpice Verilog-A Syntax:
//  .HDL 'SPIBPro_DiffI.va'
//  XDiffI NODEN NODEP SPIDiffI
//
//==========================================================================


//=============================== TYP ======================================
module SPIDiffI_TYP(nodeP, nodeN);
electrical nodeP, nodeN;
inout      nodeP, nodeN;
parameter real cDiff = 1.000E-11;

real       cDifI; // differential current due to c_diff
real       rDifI; // differential current due to resistance
real       voltP; // P node voltage and slew rate
real       voltN; // N node voltage and slew rate

analog begin
  voltP = V(nodeP);
  voltN = V(nodeN);

  // differential current due to DC steady state
  rDifI = $table_model(voltN, voltP, "driver_TYP.csv", "1L,1L");

  // differential current due to c_diff
  cDifI = ddt(voltP - voltN) * cDiff;

  // current flow between P and N
  I(nodeP, nodeN) <+ -(rDifI + cDifI);
end
endmodule
```

A Verilog-A device can be used in differential V-T extraction.

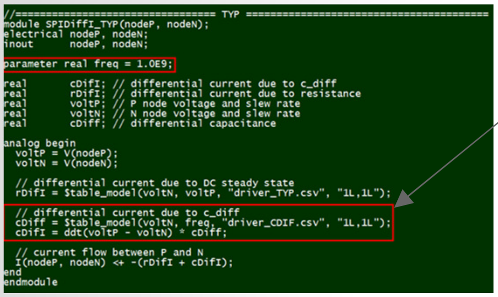
- Behavioral device is very versatile
- Support operator like ddt, if .. else
- Supports 1/2D look-up table **(5), (6)**



```
#Differential current in mA
#VOutN    VOutP    IDiff_Typ
-1.50E+00 -1.50E+00  0.00E+00
-1.50E+00 -1.46E+00 -7.83E+00
-1.50E+00 -1.41E+00 -1.59E+01
-1.50E+00 -1.37E+00 -2.41E+01
-1.50E+00 -1.32E+00 -3.25E+01
-1.50E+00 -1.28E+00 -4.13E+01
-1.50E+00 -1.23E+00 -5.03E+01
-1.50E+00 -1.19E+00 -5.95E+01
-1.50E+00 -1.14E+00 -6.91E+01
-1.50E+00 -1.10E+00 -7.89E+01
-1.50E+00 -1.05E+00 -8.90E+01
-1.50E+00 -1.01E+00 -9.94E+01
-1.50E+00 -9.60E-01 -1.10E+02
```

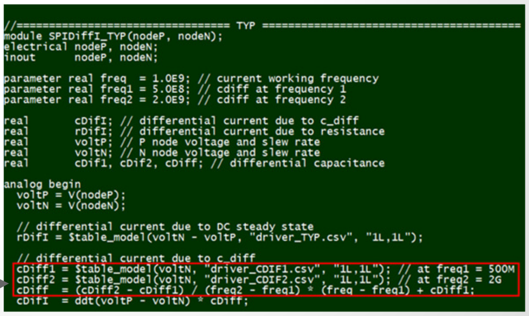
11

Verilog-A for V-T extraction:



```
//=============================== TYP ======================================
module SPIDiffI_TYP(nodeP, nodeN);
electrical nodeP, nodeN;
inout      nodeP, nodeN;

parameter real freq = 1.0E9;

real       cDifI; // differential current due to c_diff
real       rDifI; // differential current due to resistance
real       voltP; // P node voltage and slew rate
real       voltN; // N node voltage and slew rate
real       cDiff; // differential capacitance

analog begin
  voltP = V(nodeP);
  voltN = V(nodeN);

  // differential current due to DC steady state
  rDifI = $table_model(voltN, voltP, "driver_TYP.csv", "1L,1L");

  // differential current due to c_diff
  cDiff = $table_model(voltN, freq, "driver_CDIF.csv", "1L,1L");
  cDifI = ddt(voltP - voltN) * cDiff;

  // current flow between P and N
  I(nodeP, nodeN) <+ -(rDifI + cDifI);
end
endmodule
```

Voltage & freq. Dependent C_Diff
(or use cross() to find freq. dynamically)

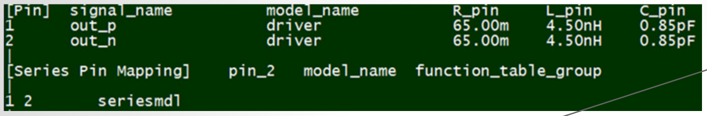


```
//=============================== TYP ======================================
module SPIDiffI_TYP(nodeP, nodeN);
electrical nodeP, nodeN;
inout      nodeP, nodeN;

parameter real freq  = 1.0E9; // current working frequency
parameter real freq1 = 5.0E8; // cdiff at frequency 1
parameter real freq2 = 2.0E9; // cdiff at frequency 2

real       cDifI; // differential current due to c_diff
real       rDifI; // differential current due to resistance
real       voltP; // P node voltage and slew rate
real       voltN; // N node voltage and slew rate
real       cDif1, cDif2, cDiff; // differential capacitance

analog begin
  voltP = V(nodeP);
  voltN = V(nodeN);

  // differential current due to DC steady state
  rDifI = $table_model(voltN - voltP, "driver_TYP.csv", "1L,1L");

  // differential current due to c_diff
  cDiff1 = $table_model(voltN, "driver_CDIF1.csv", "1L,1L"); // at freq1 = 500M
  cDiff2 = $table_model(voltN, "driver_CDIF2.csv", "1L,1L"); // at freq2 = 2G
  cDiff  = (cDiff2 - cDiff1) / (freq2 - freq1) * (freq - freq1) + cDiff1;
  cDifI  = ddt(voltP - voltN) * cDiff;
```

Simulator only supports 1D table?
2D bi-linear look-up can still be done

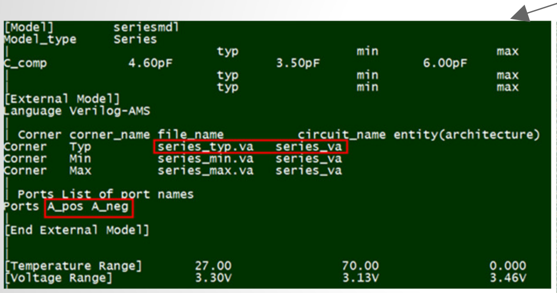
12

Completed Series model:

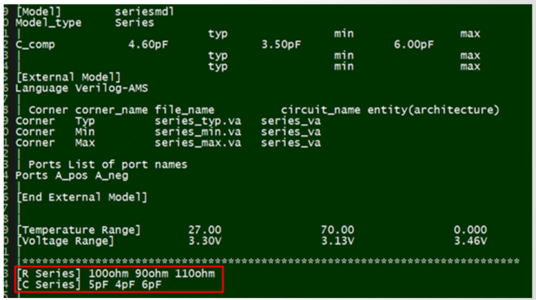


```
[Pin]  signal_name           model_name          R_pin     L_pin     C_pin
1      out_p                 driver              65.00m    4.50nH    0.85pF
2      out_n                 driver              65.00m    4.50nH    0.85pF
|
[Series Pin Mapping]    pin_2   model_name  function_table_group
|
1 2        seriesmdl
```

- Verilog-A as external model for model type “Series”



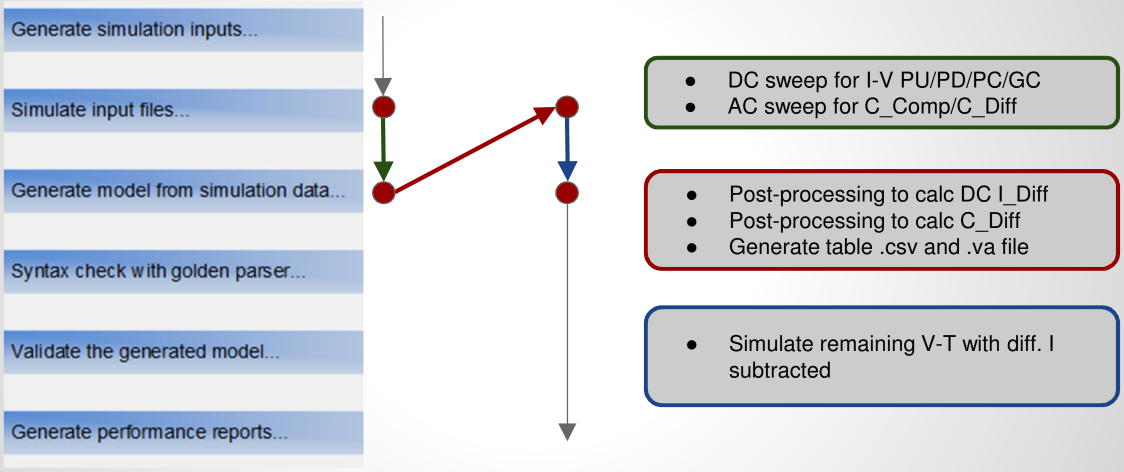
```
[Model]        seriesmdl
Model_type     Series
|                          typ                      min                      max
C_comp          4.60pF              3.50pF                   6.00pF
|                          typ                      min                      max
|                          typ                      min                      max
[External Model]
Language Verilog-AMS
|
| Corner corner_name file_name           circuit_name entity(architecture)
Corner   Typ         series_typ.va   series_va
Corner   Min         series_min.va   series_va
Corner   Max         series_max.va   series_va
|
| Ports List of port names
Ports A_pos A_neg
|
[End External Model]
|
|
[Temperature Range]      27.00                    70.00                    0.000
[Voltage Range]          3.30V                    3.13V                    3.46V
```



```
[Model]        seriesmdl
Model_type     Series
|                          typ                      min                      max
C_comp          4.60pF              3.50pF                   6.00pF
|                          typ                      min                      max
|                          typ                      min                      max
[External Model]
Language Verilog-AMS
|
| Corner corner_name file_name           circuit_name entity(architecture)
Corner   Typ         series_typ.va   series_va
Corner   Min         series_min.va   series_va
Corner   Max         series_max.va   series_va
|
| Ports List of port names
Ports A_pos A_neg
|
[End External Model]
|
|
[Temperature Range]      27.00                    70.00                    0.000
[Voltage Range]          3.30V                    3.13V                    3.46V
|
|*********************************************************************************
[R Series] 100ohm 90ohm 110ohm
[C Series] 5pF 4pF 6pF
```

- Verilog-A can work with existing (generic) series model to provide extra accuracy if needed.

Differential modeling flow:



Generate simulation inputs...

Simulate input files...

Generate model from simulation data...

Syntax check with golden parser...

Validate the generated model...

Generate performance reports...

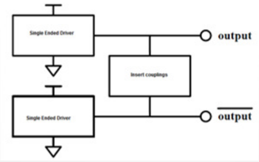
- DC sweep for I-V PU/PD/PC/GC
- AC sweep for C_Comp/C_Diff

- Post-processing to calc DC I_Diff
- Post-processing to calc C_Diff
- Generate table .csv and .va file

- Simulate remaining V-T with diff. I subtracted

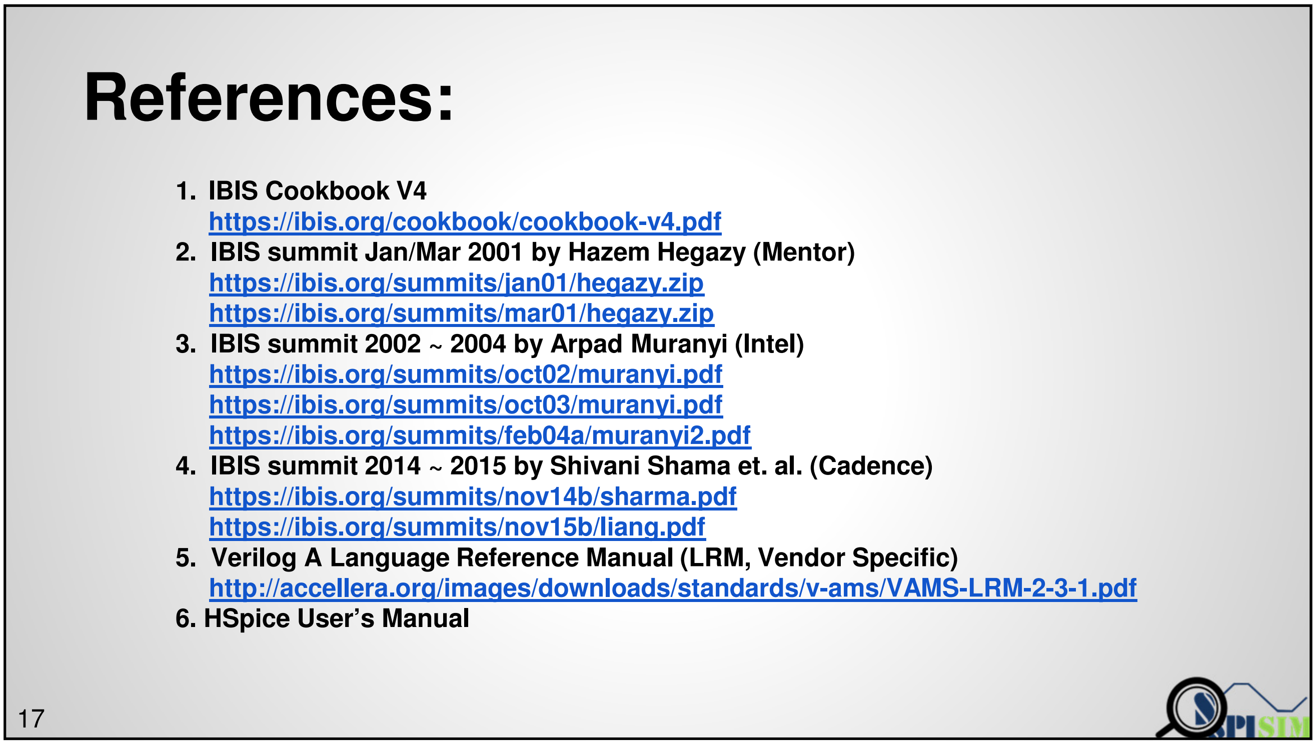
Modeling flow validation:

- Use an existing single-ended driver for P and N
- Insert approximate non-linear behavioral R/L/C elements between P and N outputs
- Flow should recreate same PU/PD/PC/GC as driver
- I-V surface plot should reveal inserted resistance
- C_Diff/C_Comp surface should reveal inserted cap
- Correlations of V-T table depends on I_Diff accuracies.



Summary:

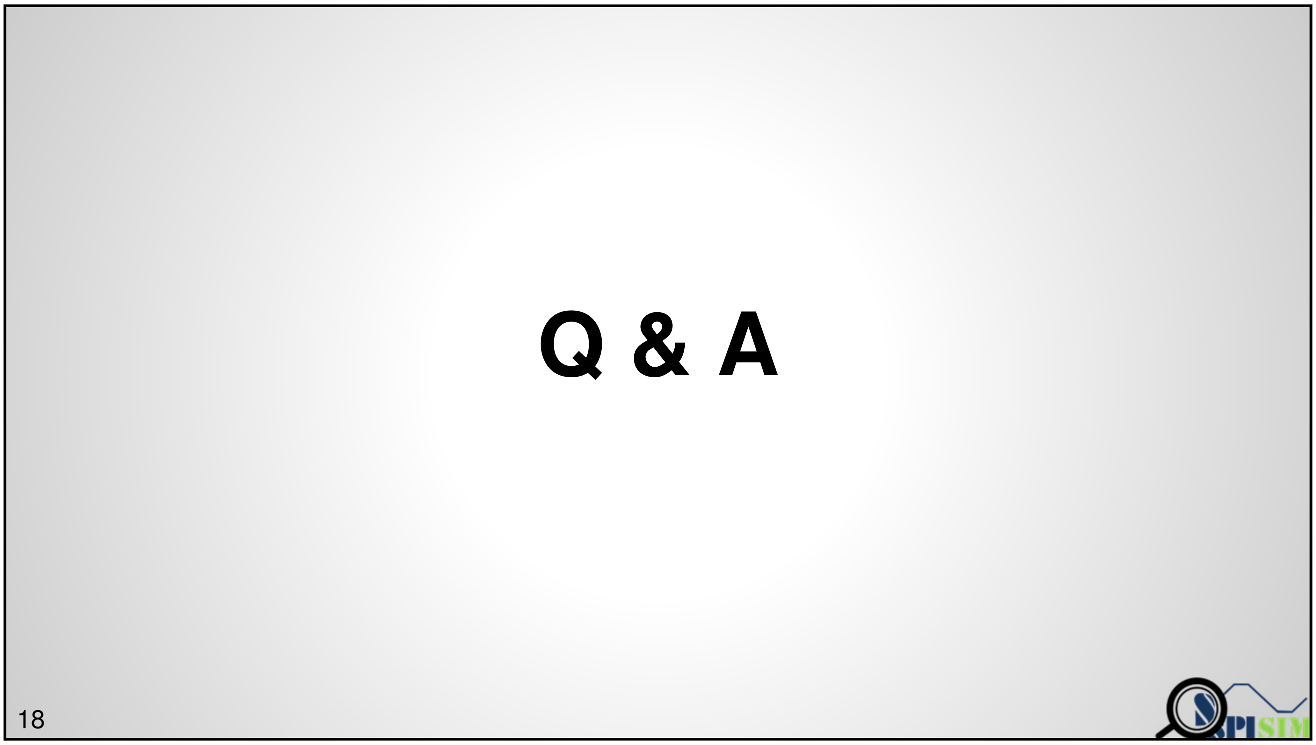
- Verilog-A for differential V-T extraction
  - Versatile, supports many operators
    - E.g. ddt(Vx), $table_model for 1D/2D lookup
    - Streamlines modeling flow
  - Extract transient differential current
    - Improve V-T extraction accuracies
    - Use Verilog-A to remove rigid series syntax
- External model for “Series”:
  - [External model] supports “Series” type model
  - Can work with generic series model



References:

1. IBIS Cookbook V4
   https://ibis.org/cookbook/cookbook-v4.pdf
2. IBIS summit Jan/Mar 2001 by Hazem Hegazy (Mentor)
   https://ibis.org/summits/jan01/hegazy.zip
   https://ibis.org/summits/mar01/hegazy.zip
3. IBIS summit 2002 ~ 2004 by Arpad Muranyi (Intel)
   https://ibis.org/summits/oct02/muranyi.pdf
   https://ibis.org/summits/oct03/muranyi.pdf
   https://ibis.org/summits/feb04a/muranyi2.pdf
4. IBIS summit 2014 ~ 2015 by Shivani Shama et. al. (Cadence)
   https://ibis.org/summits/nov14b/sharma.pdf
   https://ibis.org/summits/nov15b/liang.pdf
5. Verilog A Language Reference Manual (LRM, Vendor Specific)
   http://accellera.org/images/downloads/standards/v-ams/VAMS-LRM-2-3-1.pdf
6. HSpice User's Manual

17



Q & A

18

EDA Expertise in Signal, Power Integrity & Simulation

SPISim is an InSync member.

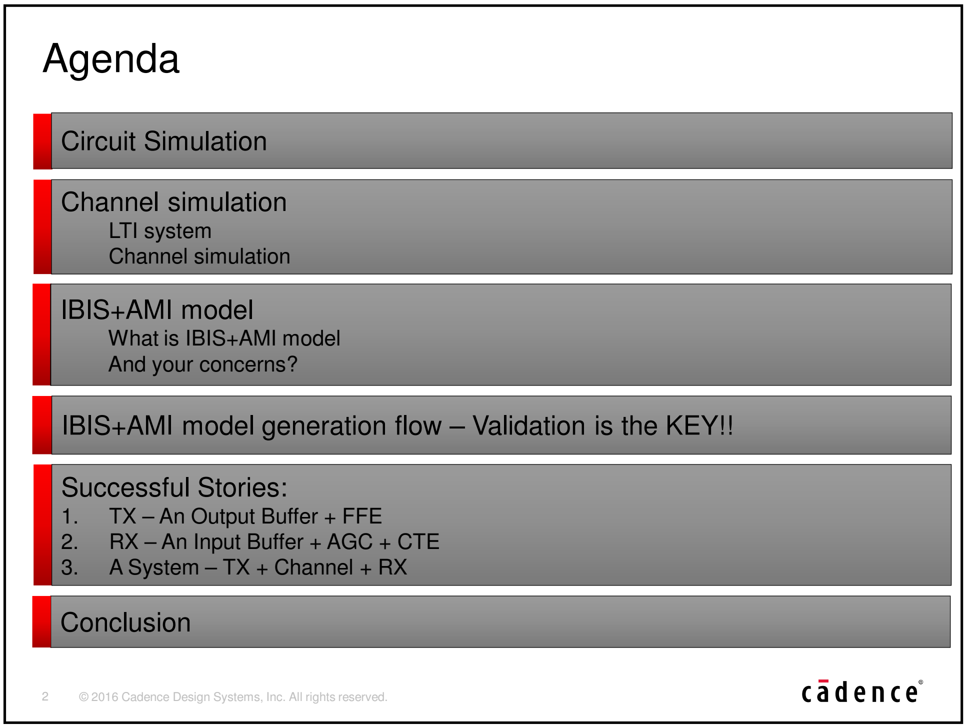
19

IBIS-AMI Model Generation
-**With Quality**

Skipper Liang

Asian IBIS Summit, Taipei, Taiwan

November 14, 2016



Agenda

Circuit Simulation

Channel simulation
- LTI system
- Channel simulation

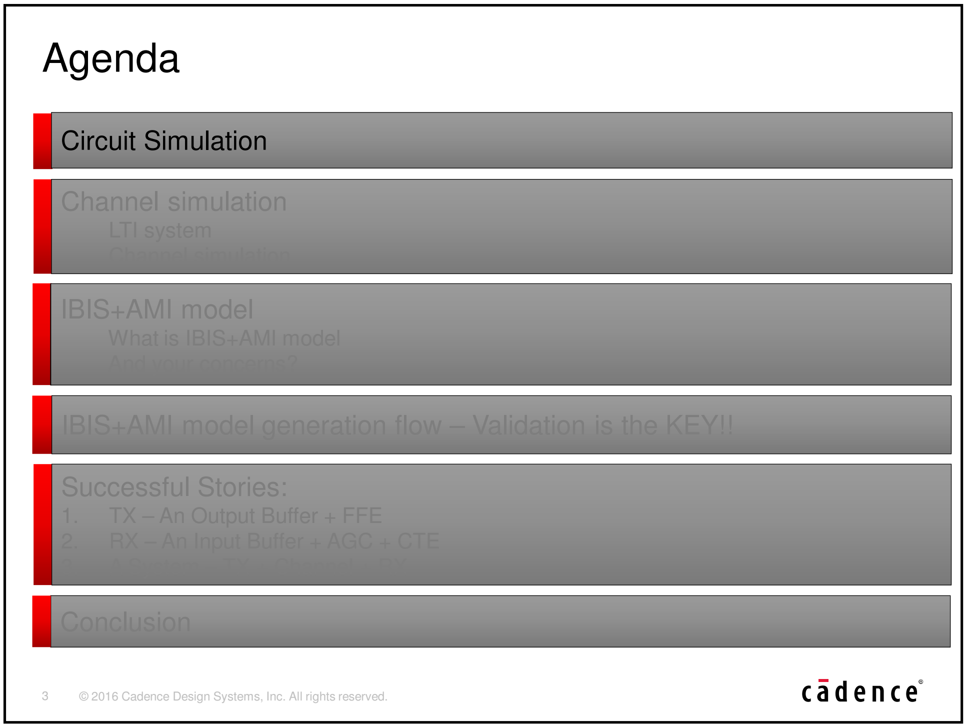
IBIS+AMI model
- What is IBIS+AMI model
- And your concerns?

IBIS+AMI model generation flow – Validation is the KEY!!

Successful Stories:
1. TX – An Output Buffer + FFE
2. RX – An Input Buffer + AGC + CTE
3. A System – TX + Channel + RX

Conclusion

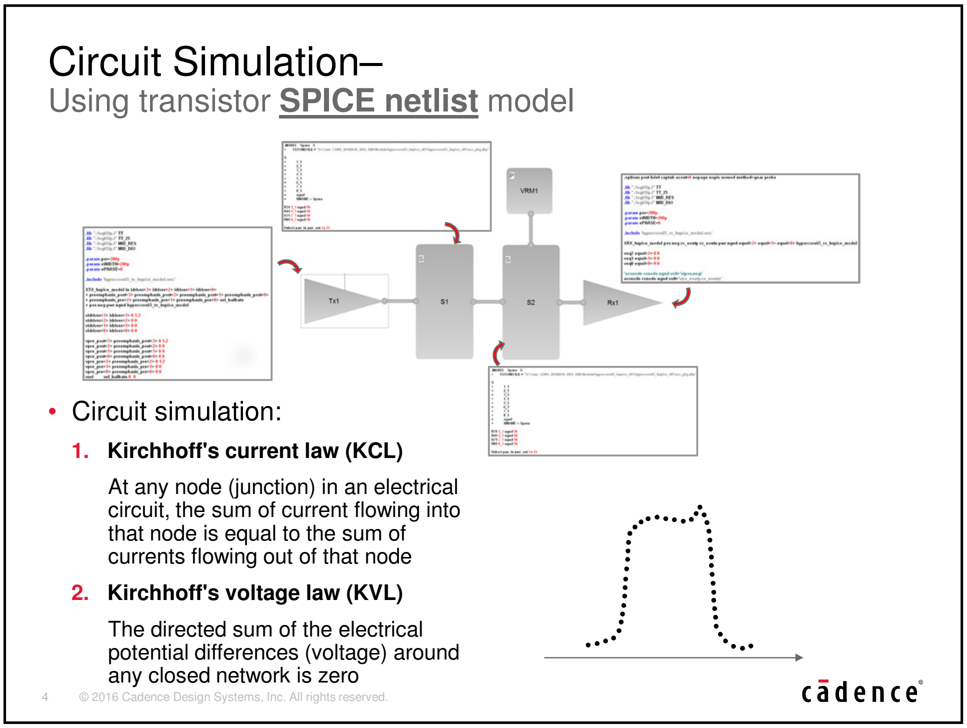
2 © 2016 Cadence Design Systems, Inc. All rights reserved.



Agenda

- Circuit Simulation
- Channel simulation
  - LTI system
  - Channel simulation
- IBIS+AMI model
  - What is IBIS+AMI model
  - And your concerns?
- IBIS+AMI model generation flow – Validation is the KEY!!
- Successful Stories:
  1. TX – An Output Buffer + FFE
  2. RX – An Input Buffer + AGC + CTE
  3. A System – TX + Channel + RX
- Conclusion

3 © 2016 Cadence Design Systems, Inc. All rights reserved.



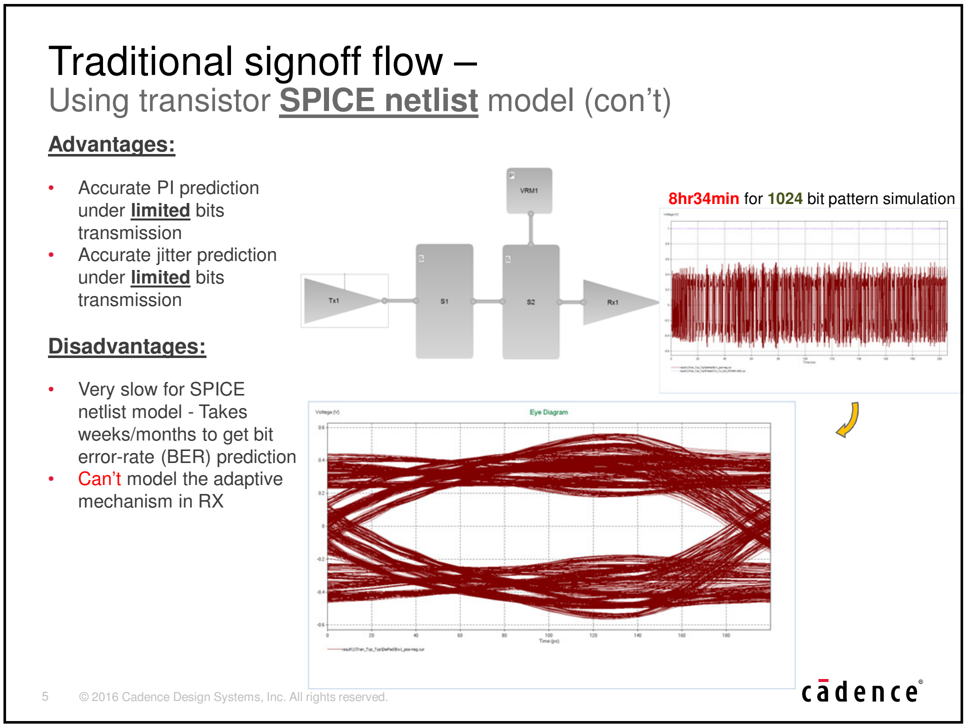
Circuit Simulation–
Using transistor **SPICE netlist** model

- Circuit simulation:
  1. **Kirchhoff's current law (KCL)**

     At any node (junction) in an electrical circuit, the sum of current flowing into that node is equal to the sum of currents flowing out of that node
  2. **Kirchhoff's voltage law (KVL)**

     The directed sum of the electrical potential differences (voltage) around any closed network is zero

4 © 2016 Cadence Design Systems, Inc. All rights reserved.



Traditional signoff flow –

Using transistor **SPICE netlist** model (con't)

Advantages:

- Accurate PI prediction under **limited** bits transmission
- Accurate jitter prediction under **limited** bits transmission

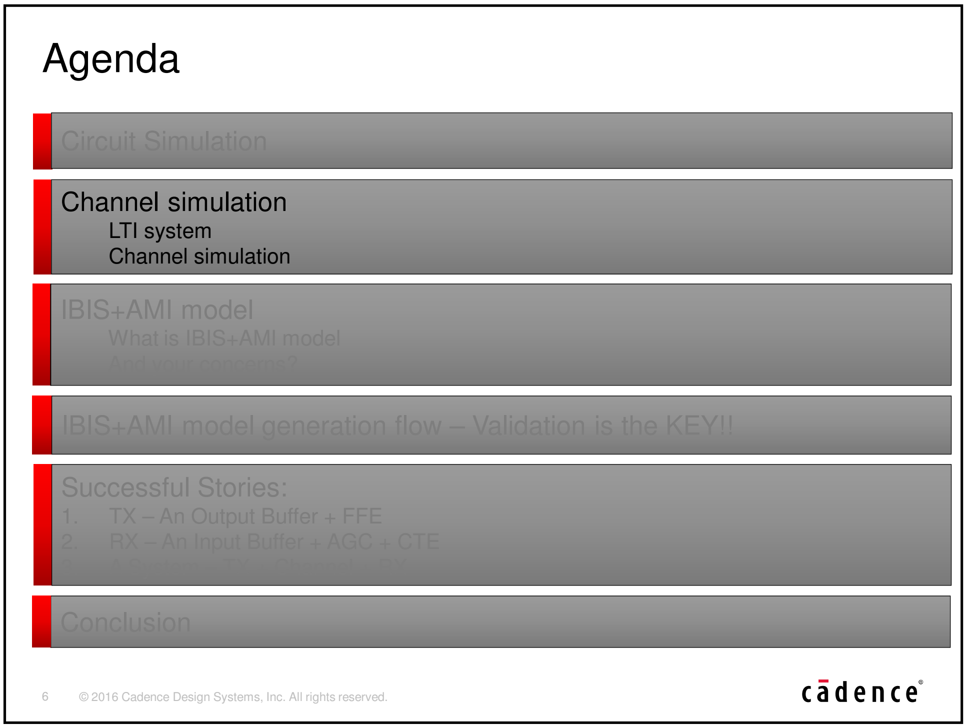
Disadvantages:

- Very slow for SPICE netlist model - Takes weeks/months to get bit error-rate (BER) prediction
- Can't model the adaptive mechanism in RX

8hr34min for 1024 bit pattern simulation

5 © 2016 Cadence Design Systems, Inc. All rights reserved.

cadence



Agenda

Circuit Simulation

Channel simulation
    LTI system
    Channel simulation

IBIS+AMI model
    What is IBIS+AMI model
    And your concerns?

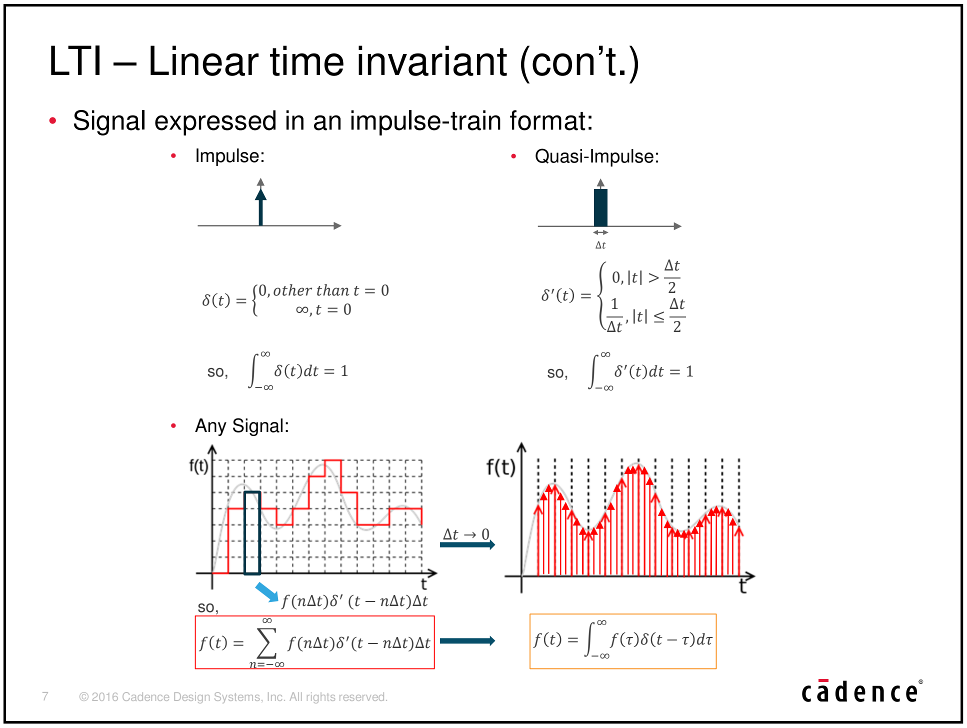
IBIS+AMI model generation flow – Validation is the KEY!!

Successful Stories:
1. TX – An Output Buffer + FFE
2. RX – An Input Buffer + AGC + CTE
3. A System – TX + Channel + RX

Conclusion

6 © 2016 Cadence Design Systems, Inc. All rights reserved.

cadence



LTI – Linear time invariant (con't.)

- Signal expressed in an impulse-train format:
  - Impulse:
  - Quasi-Impulse:

$$\delta(t) = \begin{cases} 0, other\ than\ t = 0 \\ \infty, t = 0 \end{cases}$$

so, $\int_{-\infty}^{\infty} \delta(t)dt = 1$

$$\delta'(t) = \begin{cases} 0, |t| > \frac{\Delta t}{2} \\ \frac{1}{\Delta t}, |t| \le \frac{\Delta t}{2} \end{cases}$$

so, $\int_{-\infty}^{\infty} \delta'(t)dt = 1$

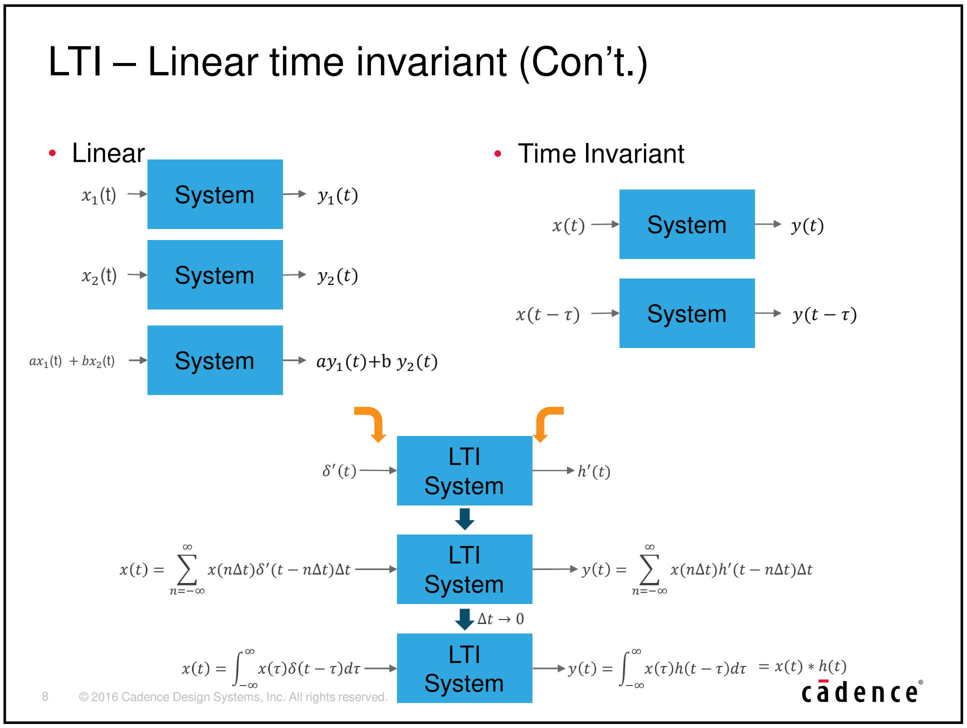
- Any Signal:

$f(n\Delta t)\delta'\ (t - n\Delta t)\Delta t$

so,

$$\boxed{f(t) = \sum_{n=-\infty}^{\infty} f(n\Delta t)\delta'(t - n\Delta t)\Delta t} \longrightarrow \boxed{f(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau}$$

7 © 2016 Cadence Design Systems, Inc. All rights reserved.



LTI – Linear time invariant (Con't.)

- Linear

x_1(t) → System → $y_1(t)$

x_2(t) → System → $y_2(t)$

ax_1(t) $+ bx_2$(t) → System → $ay_1(t)$+b $y_2(t)$

- Time Invariant

$x(t)$ → System → $y(t)$

$x(t-\tau)$ → System → $y(t-\tau)$

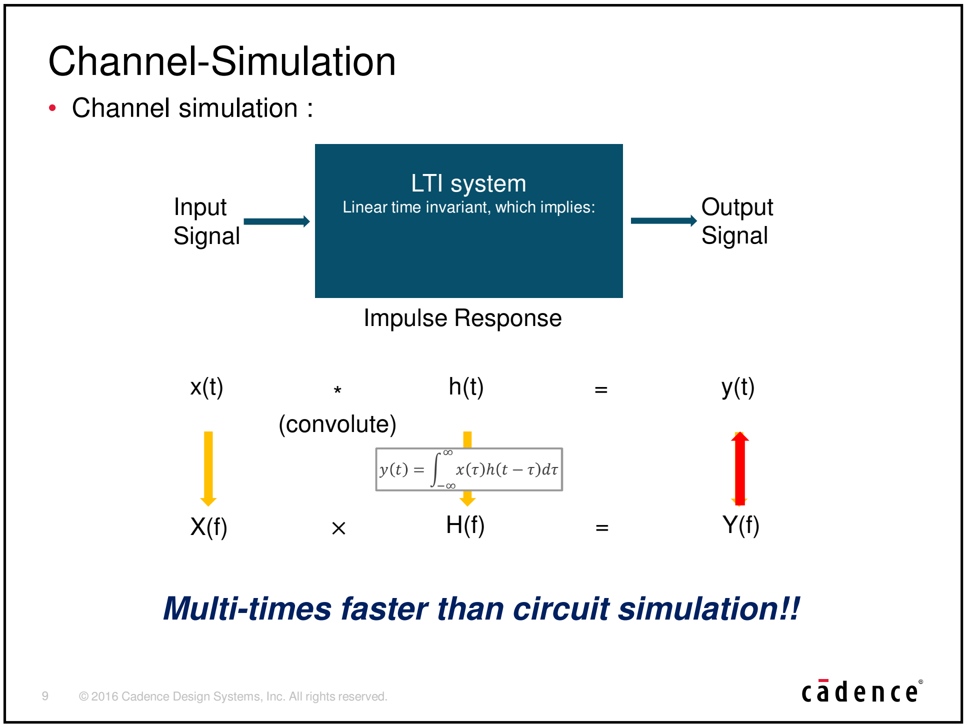
$\delta'(t)$ → LTI System → $h'(t)$

$x(t) = \sum_{n=-\infty}^{\infty} x(n\Delta t)\delta'(t - n\Delta t)\Delta t$ → LTI System → $y(t) = \sum_{n=-\infty}^{\infty} x(n\Delta t)h'(t - n\Delta t)\Delta t$

$\Delta t \to 0$

$x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)d\tau$ → LTI System → $y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \ = x(t) * h(t)$

8 © 2016 Cadence Design Systems, Inc. All rights reserved.



Channel-Simulation

- Channel simulation :

Input Signal → LTI system
Linear time invariant, which implies: → Output Signal

Impulse Response

x(t) * h(t) = y(t)

(convolute)

$$y(t)=\int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

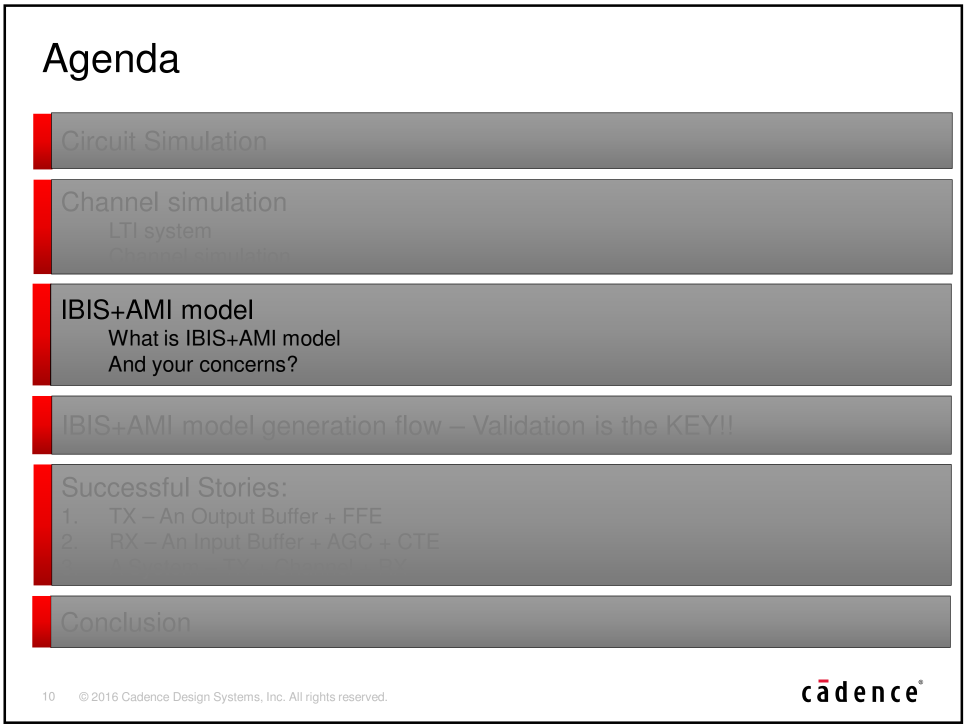
X(f) × H(f) = Y(f)

Multi-times faster than circuit simulation!!

9

© 2016 Cadence Design Systems, Inc. All rights reserved.

cādence®



Agenda

Circuit Simulation

Channel simulation
LTI system
Channel simulation

IBIS+AMI model
What is IBIS+AMI model
And your concerns?

IBIS+AMI model generation flow – Validation is the KEY!!

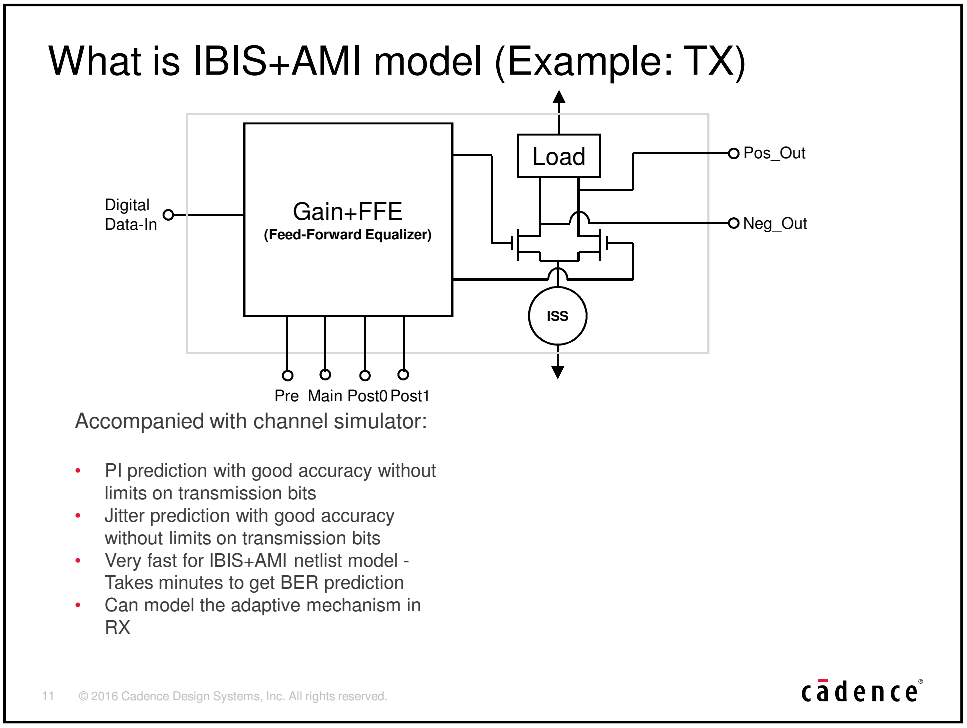
Successful Stories:
1. TX – An Output Buffer + FFE
2. RX – An Input Buffer + AGC + CTE
3. A System – TX + Channel + RX

Conclusion

10

© 2016 Cadence Design Systems, Inc. All rights reserved.

cādence®

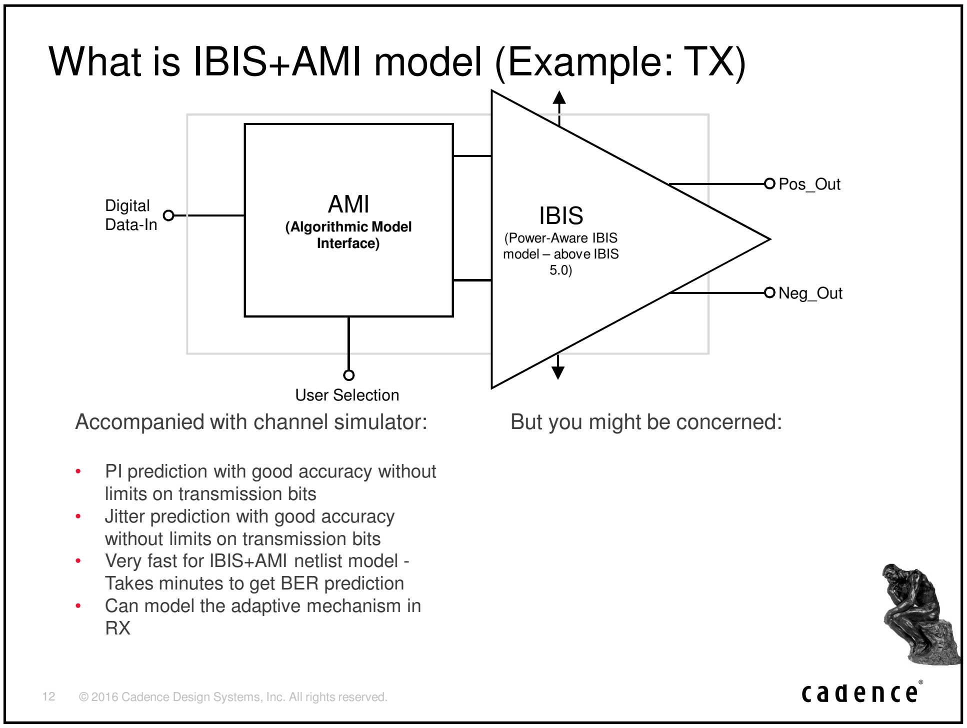


What is IBIS+AMI model (Example: TX)

Accompanied with channel simulator:

- PI prediction with good accuracy without limits on transmission bits
- Jitter prediction with good accuracy without limits on transmission bits
- Very fast for IBIS+AMI netlist model - Takes minutes to get BER prediction
- Can model the adaptive mechanism in RX

11 © 2016 Cadence Design Systems, Inc. All rights reserved.



What is IBIS+AMI model (Example: TX)

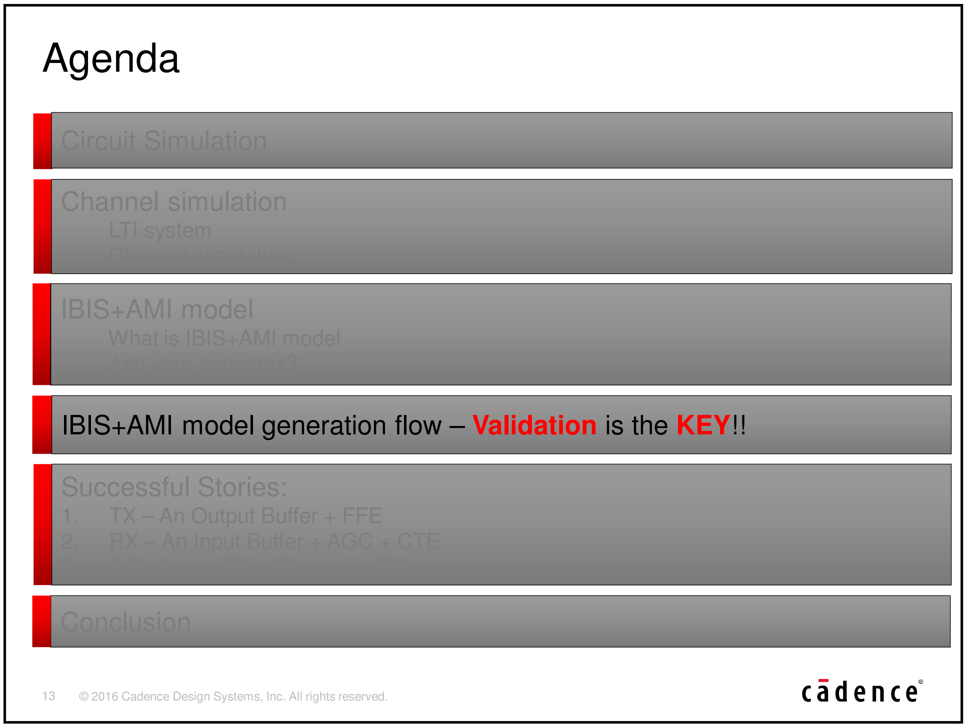
Digital
Data-In
AMI
(Algorithmic Model Interface)
IBIS
(Power-Aware IBIS model – above IBIS 5.0)
Pos_Out
Neg_Out
User Selection

Accompanied with channel simulator:

- PI prediction with good accuracy without limits on transmission bits
- Jitter prediction with good accuracy without limits on transmission bits
- Very fast for IBIS+AMI netlist model - Takes minutes to get BER prediction
- Can model the adaptive mechanism in RX

But you might be concerned:

12 © 2016 Cadence Design Systems, Inc. All rights reserved.



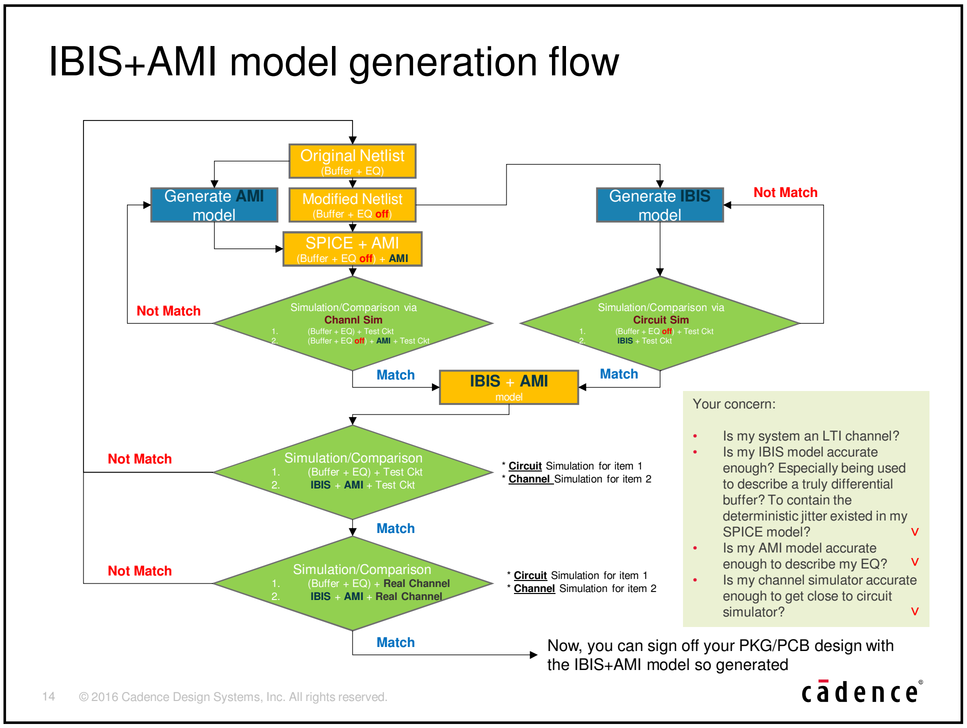
Agenda

- Circuit Simulation
- Channel simulation
  - LTI system
  - Channel simulation
- IBIS+AMI model
  - What is IBIS+AMI model
  - And your concerns?
- IBIS+AMI model generation flow – **Validation** is the **KEY**!!
- Successful Stories:
  1. TX – An Output Buffer + FFE
  2. RX – An Input Buffer + AGC + CTE
  3. A System – TX + Channel + RX
- Conclusion

13

© 2016 Cadence Design Systems, Inc. All rights reserved.

cādence®



IBIS+AMI model generation flow

Original Netlist (Buffer + EQ)

Generate **AMI** model

Modified Netlist (Buffer + EQ **off**)

SPICE + AMI (Buffer + EQ **off**) + **AMI**

Generate **IBIS** model

Not Match

Simulation/Comparison via **Channl Sim**
1. (Buffer + EQ) + Test Ckt
2. (Buffer + EQ **off**) + **AMI** + Test Ckt

Not Match

Simulation/Comparison via **Circuit Sim**
1. (Buffer + EQ **off**) + Test Ckt
2. **IBIS** + Test Ckt

Match

IBIS + AMI model

Match

Not Match

Simulation/Comparison
1. (Buffer + EQ) + Test Ckt
2. **IBIS** + **AMI** + Test Ckt

* **Circuit** Simulation for item 1
* **Channel** Simulation for item 2

Match

Not Match

Simulation/Comparison
1. (Buffer + EQ) + **Real Channel**
2. **IBIS** + **AMI** + **Real Channel**

* **Circuit** Simulation for item 1
* **Channel** Simulation for item 2

Match

Now, you can sign off your PKG/PCB design with the IBIS+AMI model so generated

Your concern:

- Is my system an LTI channel?
- Is my IBIS model accurate enough? Especially being used to describe a truly differential buffer? To contain the deterministic jitter existed in my SPICE model? V
- Is my AMI model accurate enough to describe my EQ? V
- Is my channel simulator accurate enough to get close to circuit simulator? V

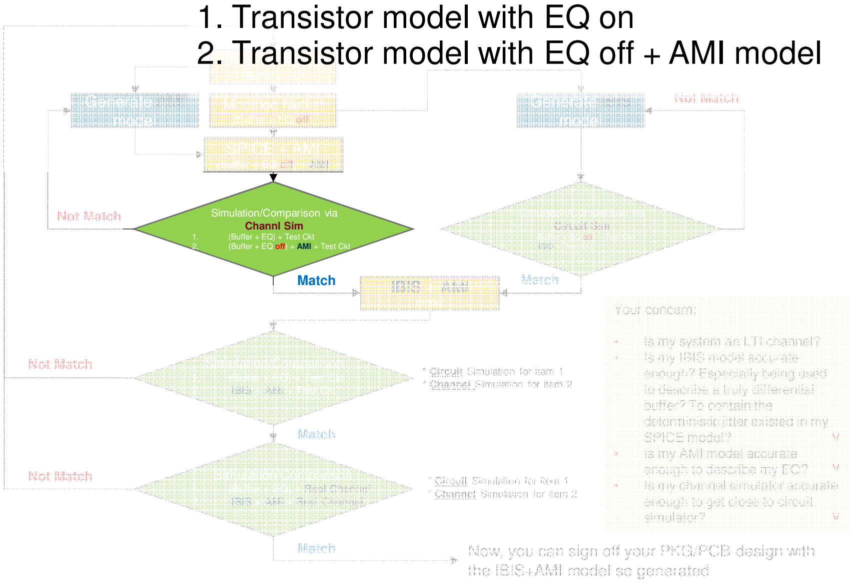
14

© 2016 Cadence Design Systems, Inc. All rights reserved.

cādence®

Validation 1: Channel Simulation for

1. Transistor model with EQ on
2. Transistor model with EQ off + AMI model

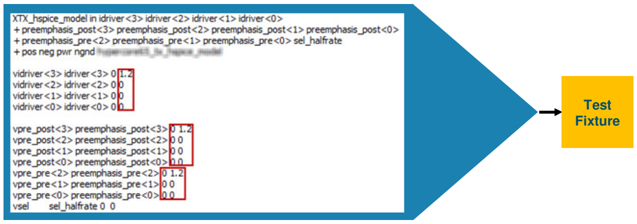

Generate model
Not Match
SPICE + AMI
off
AMI
Not Match
Simulation/Comparison via
Channl Sim
1. (Buffer + EQ) + Test Ckt
2. (Buffer + EQ off) + AMI + Test Ckt
Circuit Sim
Match
IBIS + AMI
Match
Not Match
IBIS AMI
* Circuit Simulation for item 1
* Channel Simulation for item 2
Match
Not Match
Simulation/Comparison
Real Channel
IBIS AMI Real Channel
* Circuit Simulation for item 1
* Channel Simulation for item 2
Match
Now, you can sign off your PKG/PCB design with the IBIS+AMI model so generated
Your concern:
Is my system an LTI channel?
Is my IBIS model accurate enough? Especially being used to describe a truly differential buffer? To contain the deterministic jitter existed in my SPICE model?
V
Is my AMI model accurate enough to describe my EQ?
V
Is my channel simulator accurate enough to get close to circuit simulator?
V

15 © 2016 Cadence Design Systems, Inc. All rights reserved.

Validation 1: Channel Simulation for

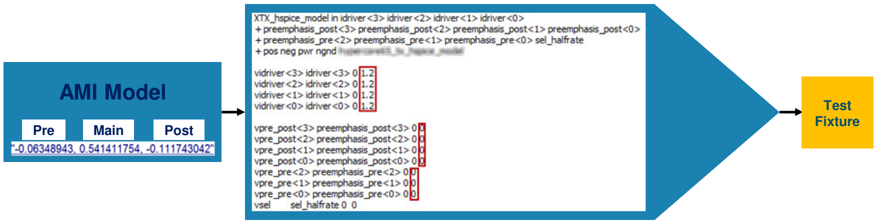
To qualify the AMI model so generated.

1. Transistor model with EQ on



```
XTX_hspice_model in idriver<3> idriver<2> idriver<1> idriver<0>
+ preemphasis_post<3> preemphasis_post<2> preemphasis_post<1> preemphasis_post<0>
+ preemphasis_pre<2> preemphasis_pre<1> preemphasis_pre<0> sel_halfrate
+ pos neg pwr ngnd [redacted]

vidriver<3> idriver<3> 0 1.2
vidriver<2> idriver<2> 0 0
vidriver<1> idriver<1> 0 0
vidriver<0> idriver<0> 0 0

vpre_post<3> preemphasis_post<3> 0 1.2
vpre_post<2> preemphasis_post<2> 0 0
vpre_post<1> preemphasis_post<1> 0 0
vpre_post<0> preemphasis_post<0> 0 0
vpre_pre<2> preemphasis_pre<2> 0 1.2
vpre_pre<1> preemphasis_pre<1> 0 0
vpre_pre<0> preemphasis_pre<0> 0 0
vsel      sel_halfrate 0  0
```

2. Transistor model with EQ off + AMI model

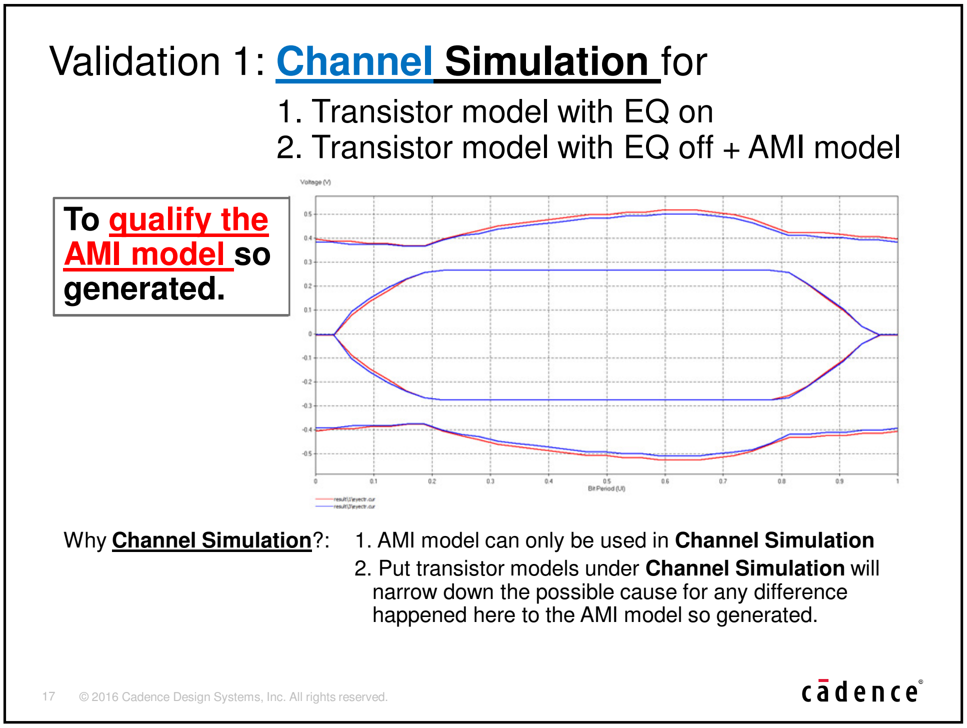


AMI Model

Pre	Main	Post
"-0.06348943,	0.541411754,	-0.111743042"

```
XTX_hspice_model in idriver<3> idriver<2> idriver<1> idriver<0>
+ preemphasis_post<3> preemphasis_post<2> preemphasis_post<1> preemphasis_post<0>
+ preemphasis_pre<2> preemphasis_pre<1> preemphasis_pre<0> sel_halfrate
+ pos neg pwr ngnd [redacted]

vidriver<3> idriver<3> 0 1.2
vidriver<2> idriver<2> 0 1.2
vidriver<1> idriver<1> 0 1.2
vidriver<0> idriver<0> 0 1.2

vpre_post<3> preemphasis_post<3> 0 0
vpre_post<2> preemphasis_post<2> 0 0
vpre_post<1> preemphasis_post<1> 0 0
vpre_post<0> preemphasis_post<0> 0 0
vpre_pre<2> preemphasis_pre<2> 0 0
vpre_pre<1> preemphasis_pre<1> 0 0
vpre_pre<0> preemphasis_pre<0> 0 0
vsel      sel_halfrate 0  0
```

16 © 2016 Cadence Design Systems, Inc. All rights reserved.



Validation 1: **Channel Simulation** for

1. Transistor model with EQ on
2. Transistor model with EQ off + AMI model

To qualify the AMI model so generated.

Why **Channel Simulation**?:

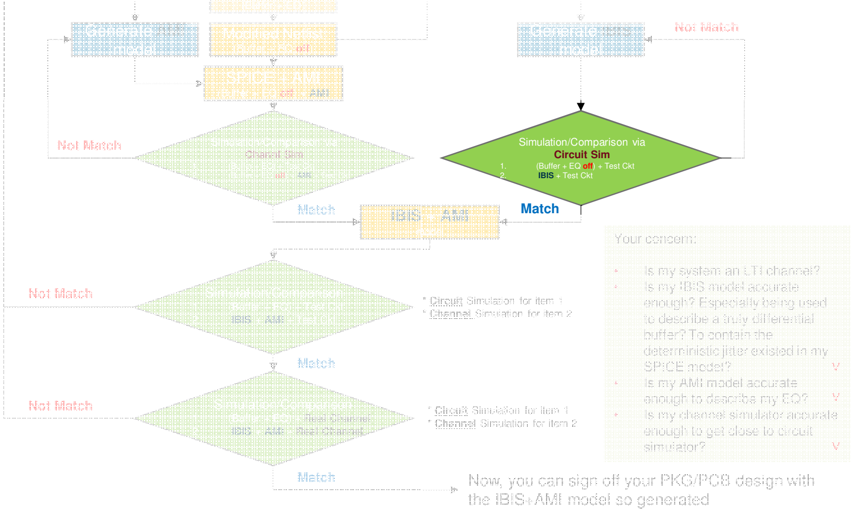
1. AMI model can only be used in **Channel Simulation**
2. Put transistor models under **Channel Simulation** will narrow down the possible cause for any difference happened here to the AMI model so generated.

17 © 2016 Cadence Design Systems, Inc. All rights reserved.

cādence®

Validation 2: **Circuit Simulation** for

1. Transistor model with EQ off
2. IBIS model so generated



* Circuit Simulation for item 1
* Channel Simulation for item 2

* Circuit Simulation for item 1
* Channel Simulation for item 2

Your concern:

- Is my system an LTI channel?
- Is my IBIS model accurate enough? Especially being used to describe a truly differential buffer? To contain the deterministic jitter existed in my SPICE model? V
- Is my AMI model accurate enough to describe my EQ? V
- Is my channel simulator accurate enough to get close to circuit simulator? V

Now, you can sign off your PKG/PCB design with the IBIS+AMI model so generated

18 © 2016 Cadence Design Systems, Inc. All rights reserved.

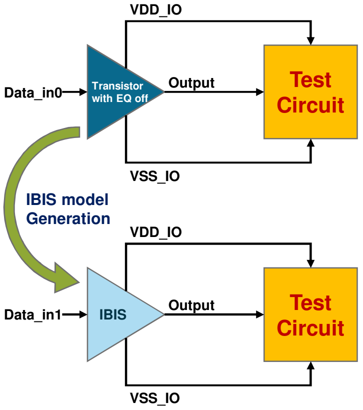
cādence®

9 .

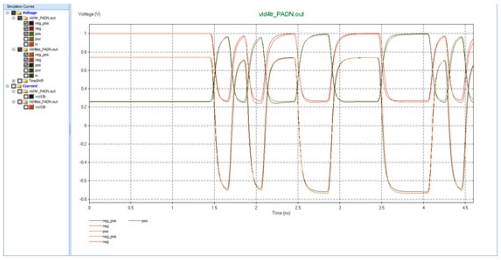
Validation 2: Circuit Simulation for

1. Transistor model with EQ off
2. IBIS model so generated

To qualify the IBIS model so generated.



Validated by Circuit Simulator



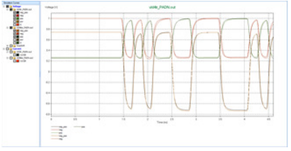
© 2016 Cadence Design Systems, Inc. All rights reserved.

cādence

Validation 2: Circuit Simulation for

1. Transistor model with EQ off
2. IBIS model so generated

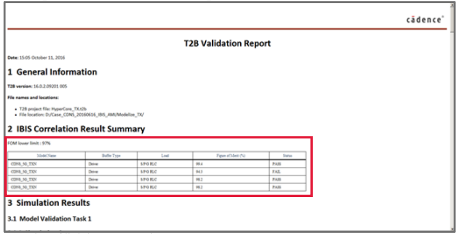
To qualify the IBIS model so generated.



Define a "mark" and a "target" to tell the quality of the IBIS model so generated

$$FOM = 100 \bullet \left[1 - \frac{\sum_{i=1}^{N} \left| Y_i(LAB) - Y_i(IBIS) \right|}{\Delta Y \bullet N} \right]$$

ΔY: (Max-Min) of Circuit Simulation Waveform



cādence

T2B Validation Report

Date: 15:05 October 11, 2016

1 General Information

T2B version: 16.0.2.09201 005

File names and locations:

- T2B project file: HyperCore_TX.t2b
- File location: D:/Case_CDNS_20160616_IBIS_AMI/Modelize_TX/

2 IBIS Correlation Result Summary

FOM lower limit : 97%

Model Name	Buffer Type	Load	Figure of Merit (%)	Status
CDNS_50_TXN	Driver	I/P G RLC	99.4	PASS
CDNS_50_TXN	Driver	I/P G RLC	94.3	FAIL
CDNS_50_TXN	Driver	I/P G RLC	98.2	PASS
CDNS_50_TXN	Driver	I/P G RLC	98.2	PASS

3 Simulation Results

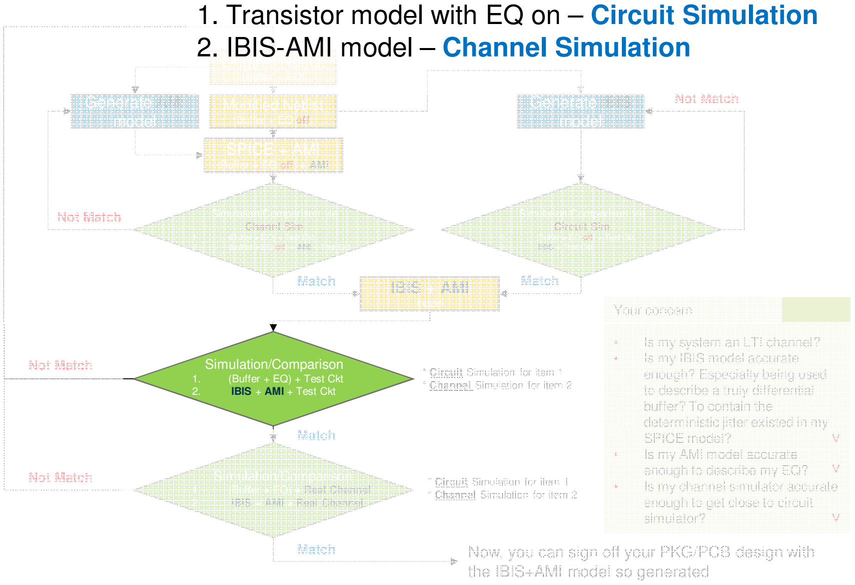
3.1 Model Validation Task 1

© 2016 Cadence Design Systems, Inc. All rights reserved.

cādence

Validation 3: Test Circuit follows

1. Transistor model with EQ on – **Circuit Simulation**
2. IBIS-AMI model – **Channel Simulation**



* Circuit Simulation for item 1
* Channel Simulation for item 2

* Circuit Simulation for item 1
* Channel Simulation for item 2

Your concern:

- Is my system an LTI channel?
- Is my IBIS model accurate enough? Especially being used to describe a truly differential buffer? To contain the deterministic jitter existed in my SPICE model? V
- Is my AMI model accurate enough to describe my EQ? V
- Is my channel simulator accurate enough to get close to circuit simulator? V

Now, you can sign off your PKG/PCB design with the IBIS+AMI model so generated

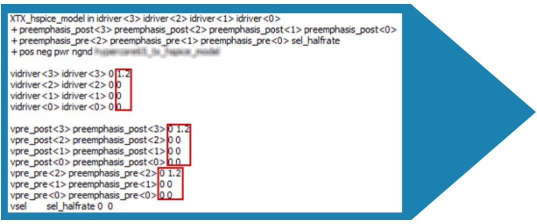
© 2016 Cadence Design Systems, Inc. All rights reserved.

cādence®

Validation 3: Test Circuit follows

1. Transistor model with EQ on – **Circuit Simulation**

To qualify the IBIS-AMI model so generated.



```
XTX_hspice_model in idriver<3> idriver<2> idriver<1> idriver<0>
 + preemphasis_post<3> preemphasis_post<2> preemphasis_post<1> preemphasis_post<0>
 + preemphasis_pre<2> preemphasis_pre<1> preemphasis_pre<0> sel_halfrate
 + pos neg pwr ngnd [redacted]

vidriver<3> idriver<3> 0 1.2
vidriver<2> idriver<2> 0 0
vidriver<1> idriver<1> 0 0
vidriver<0> idriver<0> 0 0

vpre_post<3> preemphasis_post<3> 0 1.2
vpre_post<2> preemphasis_post<2> 0 0
vpre_post<1> preemphasis_post<1> 0 0
vpre_post<0> preemphasis_post<0> 0 0
vpre_pre<2> preemphasis_pre<2> 0 1.2
vpre_pre<1> preemphasis_pre<1> 0 0
vpre_pre<0> preemphasis_pre<0> 0 0
vsel      sel_halfrate 0  0
```

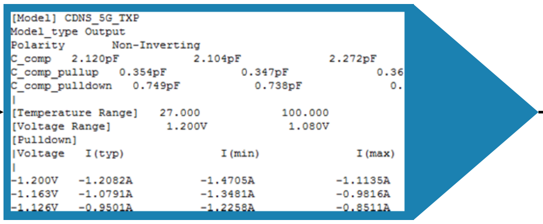
Test Fixture

2. IBIS-AMI model – **Channel Simulation**

AMI Model

Pre	Main	Post

"-0.06348943, 0.541411754, -0.111743042"

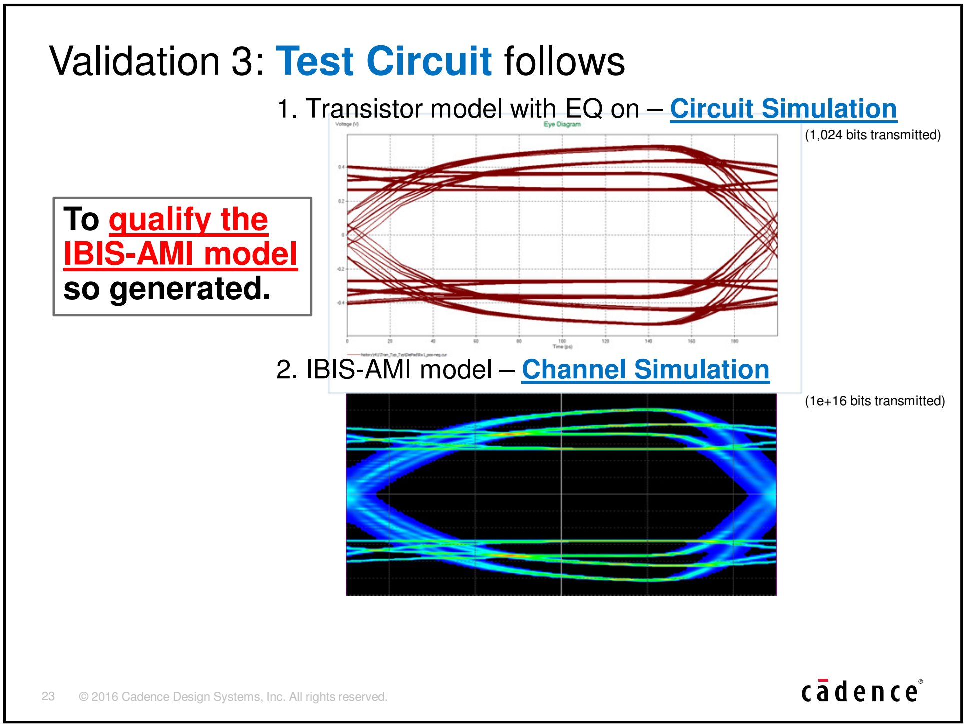


```
[Model] CDNS_5G_TXP
Model_type Output
Polarity        Non-Inverting
C_comp   2.120pF             2.104pF              2.272pF
C_comp_pullup   0.354pF             0.347pF              0.36
C_comp_pulldown   0.749pF             0.738pF              0.
|
[Temperature Range]   27.000           100.000
[Voltage Range]        1.200V            1.080V
[Pulldown]
|Voltage   I(typ)              I(min)               I(max)
|
-1.200V   -1.2082A         -1.4705A             -1.1135A
-1.163V   -1.0791A         -1.3481A             -0.9816A
-1.126V   -0.9501A         -1.2258A             -0.8511A
```

Test Fixture

© 2016 Cadence Design Systems, Inc. All rights reserved.

cādence®



Validation 3: Test Circuit follows

1. Transistor model with EQ on – Circuit Simulation

(1,024 bits transmitted)

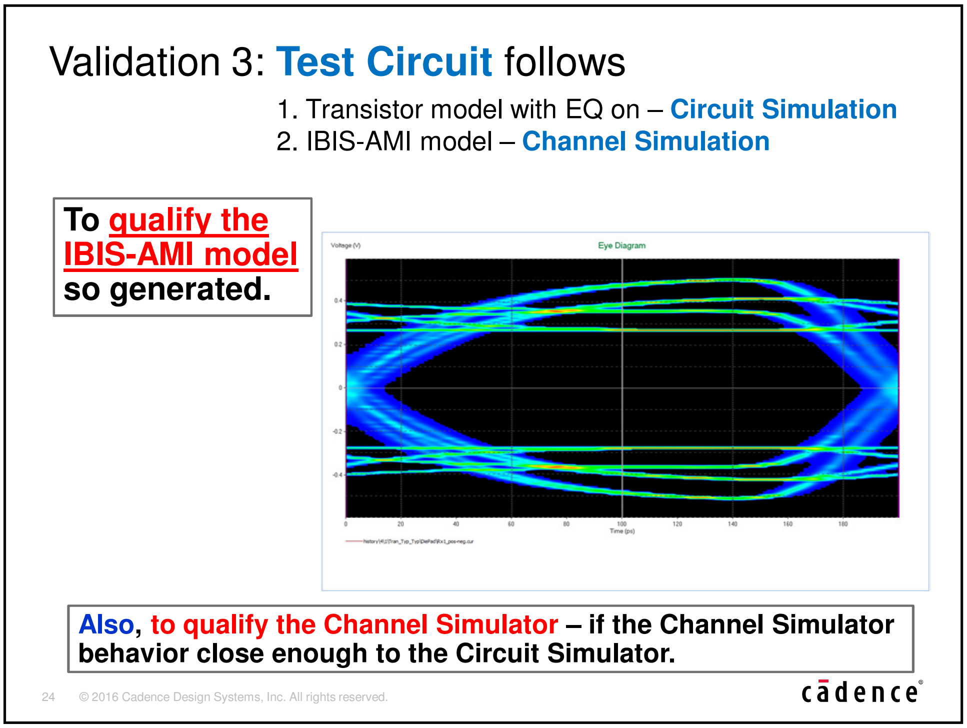
To qualify the IBIS-AMI model so generated.

2. IBIS-AMI model – Channel Simulation

(1e+16 bits transmitted)

23 © 2016 Cadence Design Systems, Inc. All rights reserved.

cadence



Validation 3: Test Circuit follows

1. Transistor model with EQ on – Circuit Simulation
2. IBIS-AMI model – Channel Simulation

To qualify the IBIS-AMI model so generated.

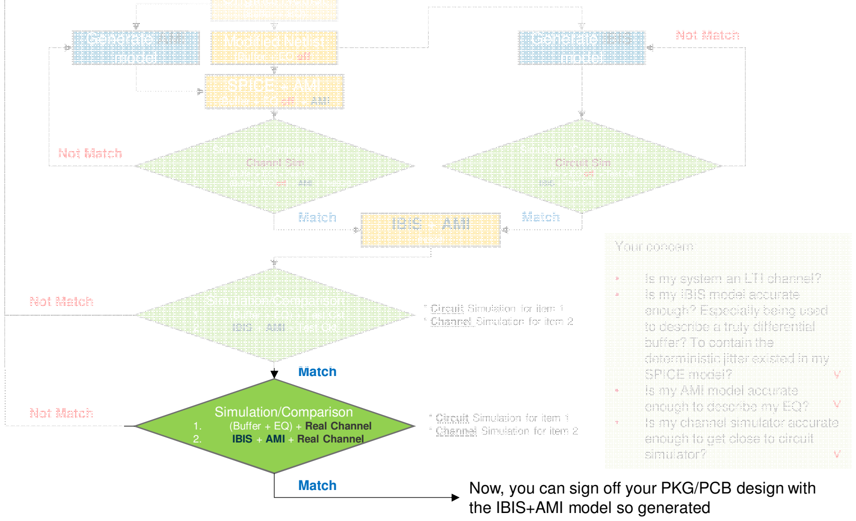
Also, to qualify the Channel Simulator – if the Channel Simulator behavior close enough to the Circuit Simulator.

24 © 2016 Cadence Design Systems, Inc. All rights reserved.

cadence

Validation 4: **Real Channel** follows

1. Transistor model with EQ on – **Circuit Simulation**
2. IBIS-AMI model – **Channel Simulation**


Generate model
Modified Netlist (Buffer + EQ off)
SPICE + AMI (Buffer + EQ off + AMI)
Generate model
Not Match
Not Match
Simulation/Comparison Channel Sim
Simulation/Comparison Circuit Sim
Match
IBIS + AMI
Match
Not Match
Simulation/Comparison
IBIS + AMI
* Circuit Simulation for item 1
* Channel Simulation for item 2
Match
Not Match
Simulation/Comparison
1. (Buffer + EQ) + Real Channel
2. IBIS + AMI + Real Channel
* Circuit Simulation for item 1
* Channel Simulation for item 2
Match

Your concern:

- Is my system an LTI channel?
- Is my IBIS model accurate enough? Especially being used to describe a truly differential buffer? To contain the deterministic jitter existed in my SPICE model? V
- Is my AMI model accurate enough to describe my EQ? V
- Is my channel simulator accurate enough to get close to circuit simulator? V

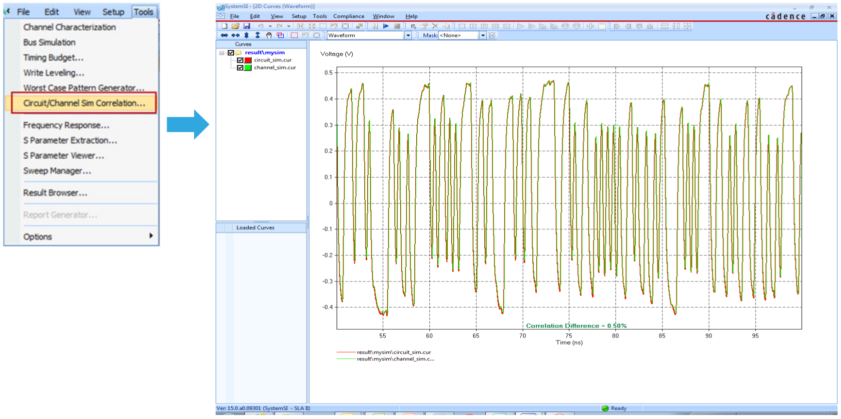
Now, you can sign off your PKG/PCB design with the IBIS+AMI model so generated

© 2016 Cadence Design Systems, Inc. All rights reserved.

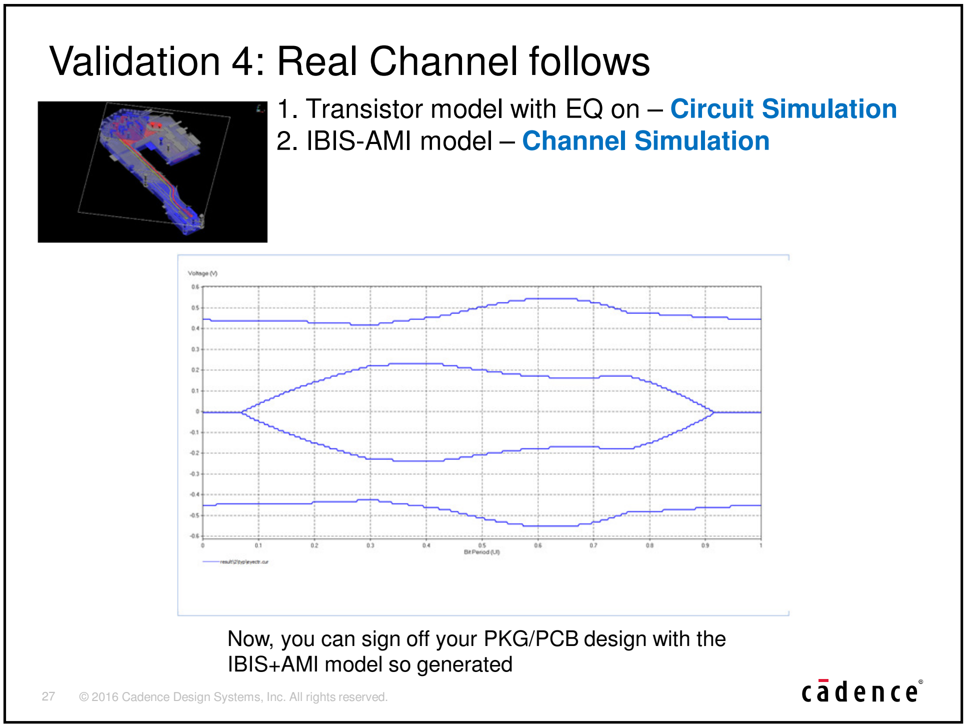
Validation 4: Real Channel follows

1. Transistor model with EQ on – **Circuit Simulation**
2. IBIS-AMI model – **Channel Simulation**

- First of all, check if your system/channel to be analyzed can be treated as LTI or not:



© 2016 Cadence Design Systems, Inc. All rights reserved.



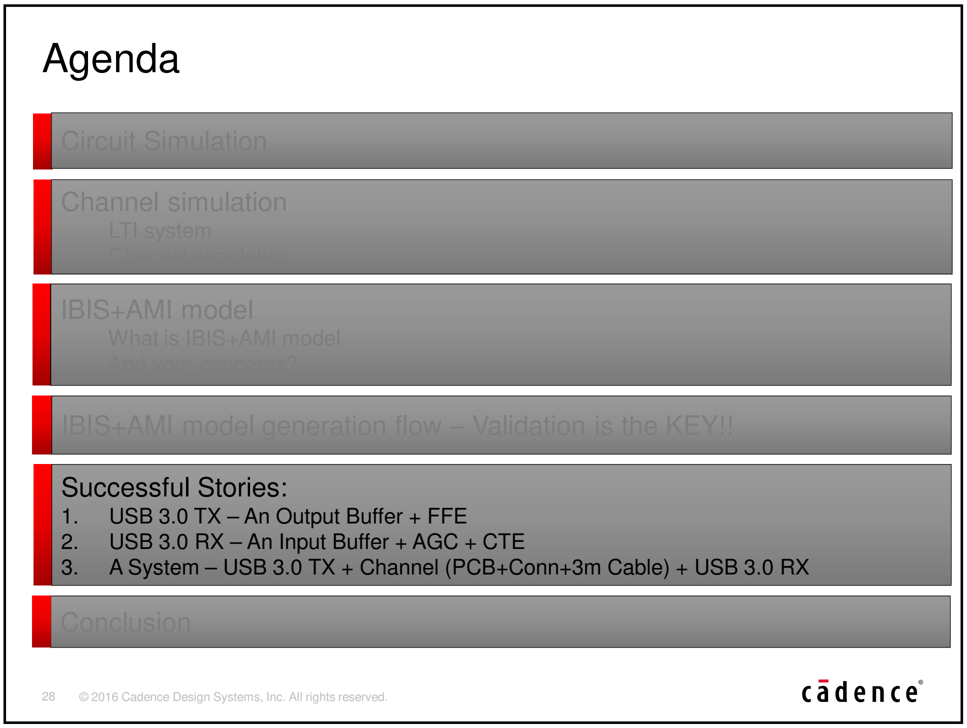
Validation 4: Real Channel follows

1. Transistor model with EQ on – **Circuit Simulation**
2. IBIS-AMI model – **Channel Simulation**

Now, you can sign off your PKG/PCB design with the IBIS+AMI model so generated

27 © 2016 Cadence Design Systems, Inc. All rights reserved.

cādence



Agenda

Circuit Simulation

Channel simulation
- LTI system
- Channel simulation

IBIS+AMI model
- What is IBIS+AMI model
- And your concerns?

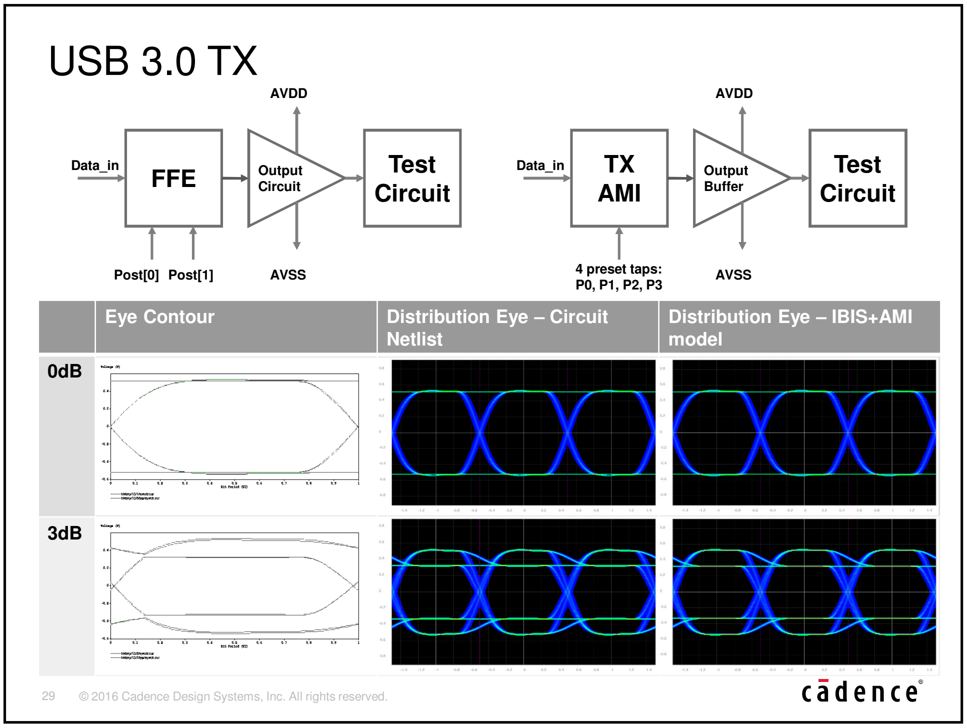
IBIS+AMI model generation flow – Validation is the KEY!!

Successful Stories:
1. USB 3.0 TX – An Output Buffer + FFE
2. USB 3.0 RX – An Input Buffer + AGC + CTE
3. A System – USB 3.0 TX + Channel (PCB+Conn+3m Cable) + USB 3.0 RX

Conclusion

28 © 2016 Cadence Design Systems, Inc. All rights reserved.

cādence



USB 3.0 TX

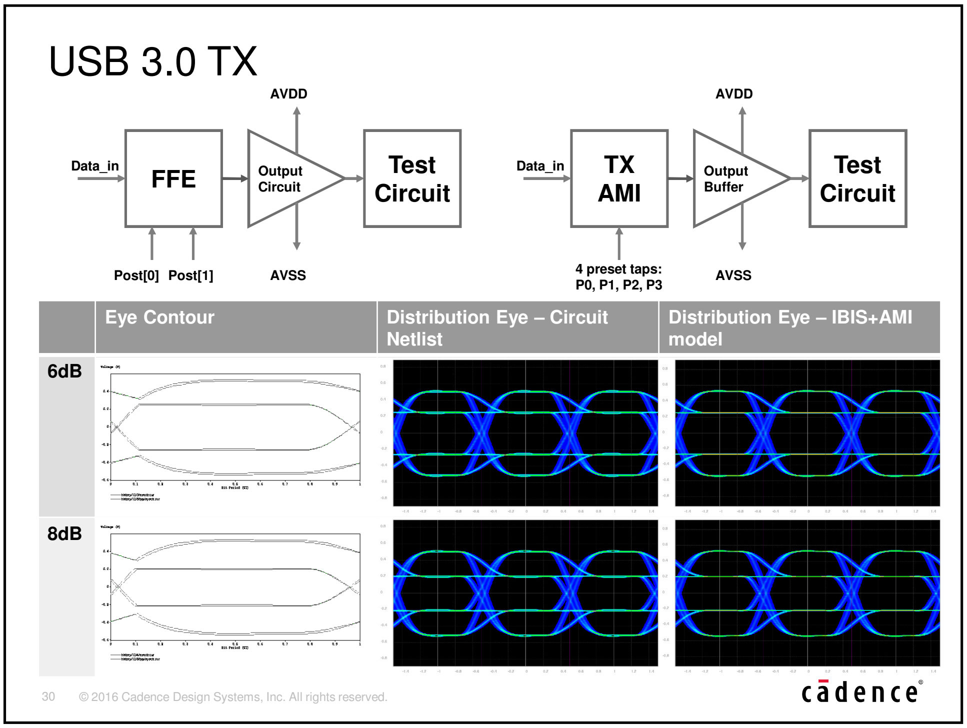
Data_in → FFE → Output Circuit → Test Circuit; Post[0], Post[1]; AVDD; AVSS

Data_in → TX AMI → Output Buffer → Test Circuit; 4 preset taps: P0, P1, P2, P3; AVDD; AVSS

	Eye Contour	Distribution Eye – Circuit Netlist	Distribution Eye – IBIS+AMI model
0dB			
3dB			

29 © 2016 Cadence Design Systems, Inc. All rights reserved.

cādence



USB 3.0 TX

Data_in → FFE → Output Circuit → Test Circuit; Post[0], Post[1]; AVDD; AVSS

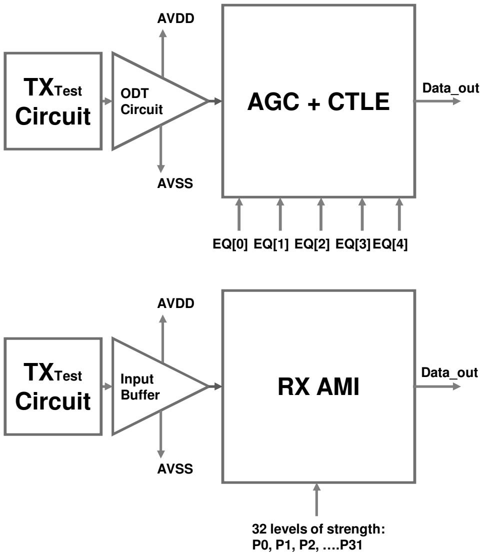
Data_in → TX AMI → Output Buffer → Test Circuit; 4 preset taps: P0, P1, P2, P3; AVDD; AVSS

	Eye Contour	Distribution Eye – Circuit Netlist	Distribution Eye – IBIS+AMI model
6dB			
8dB			

30 © 2016 Cadence Design Systems, Inc. All rights reserved.

cādence

USB 3.0 RX



TXTest Circuit → ODT Circuit → AGC + CTLE → Data_out

AVDD, AVSS

EQ[0] EQ[1] EQ[2] EQ[3] EQ[4]

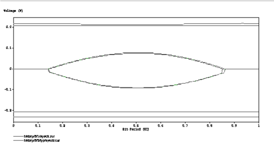
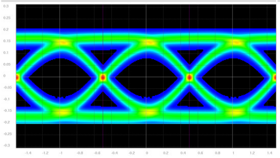
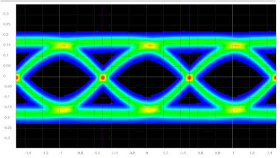
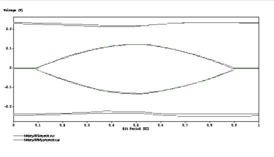
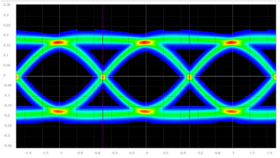
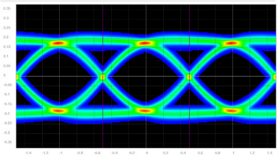
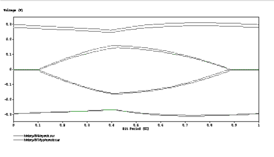
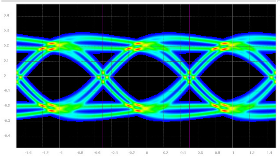
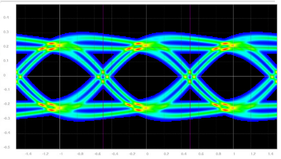
TXTest Circuit → Input Buffer → RX AMI → Data_out

AVDD, AVSS

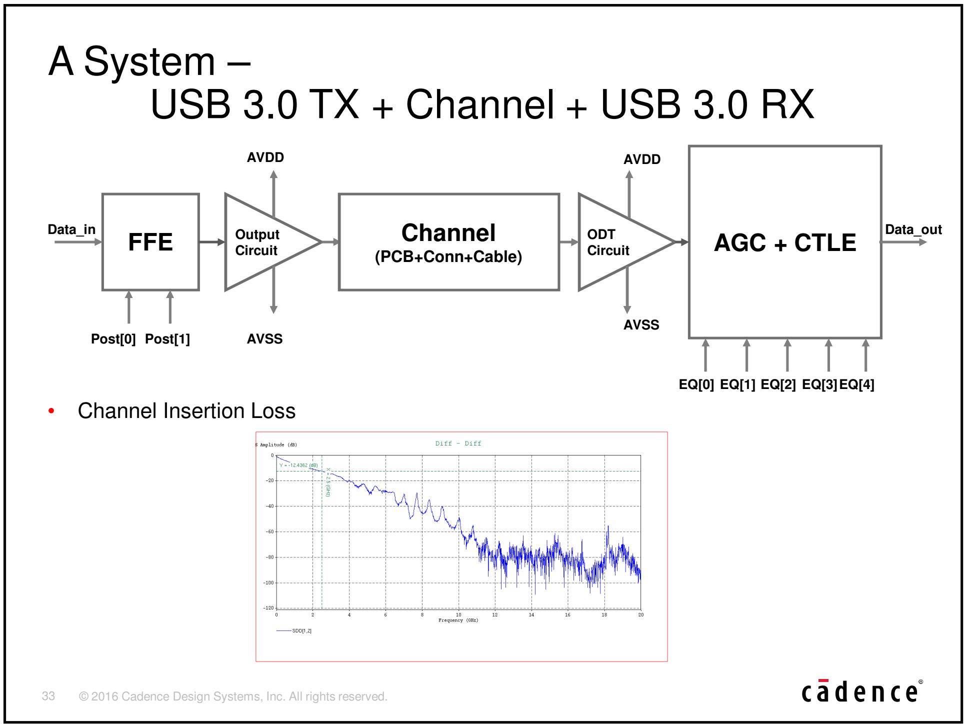
32 levels of strength:
P0, P1, P2, ….P31

31 © 2016 Cadence Design Systems, Inc. All rights reserved.

USB 3.0 RX

	Eye Contour	Distribution Eye – Circuit Netlist	Distribution Eye – IBIS+AMI model
00100			
00110			
01010			

32 © 2016 Cadence Design Systems, Inc. All rights reserved.

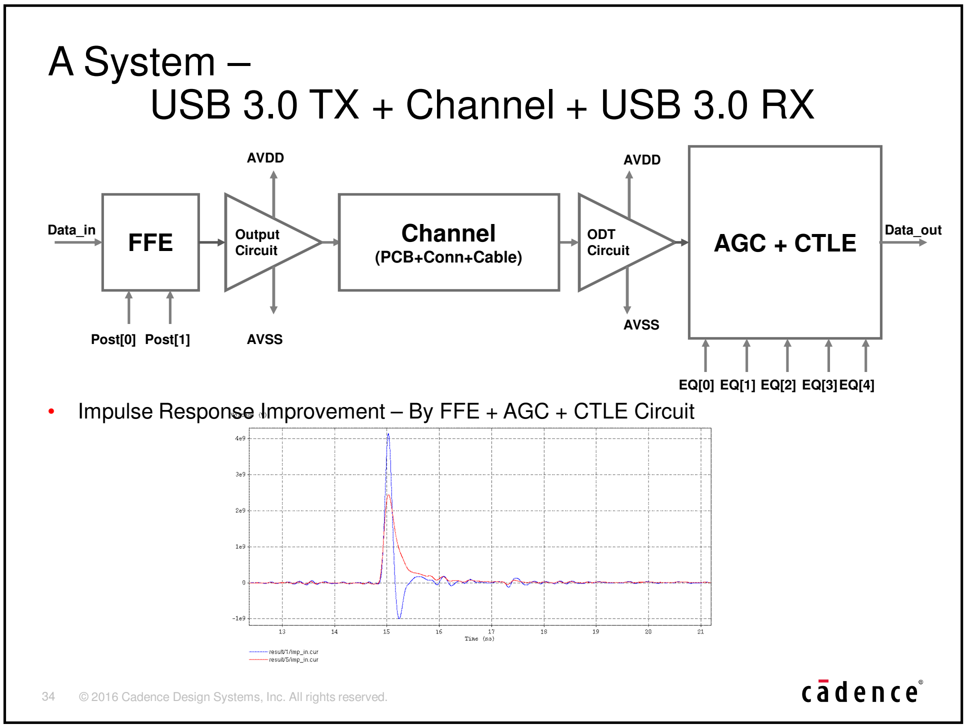


A System – USB 3.0 TX + Channel + USB 3.0 RX

- Channel Insertion Loss

© 2016 Cadence Design Systems, Inc. All rights reserved.

cādence®

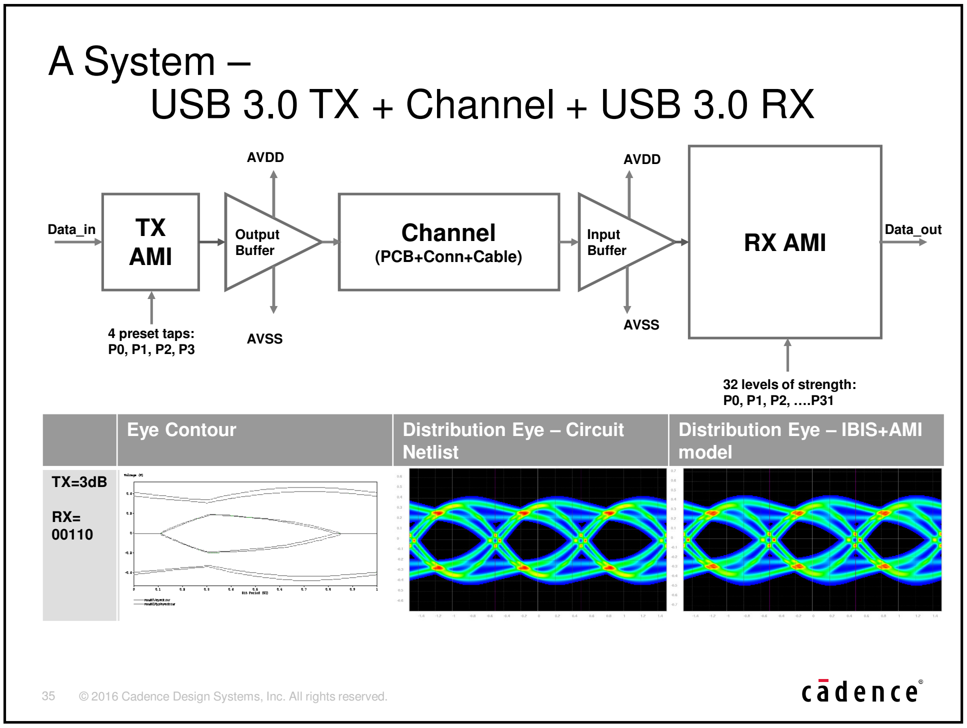


A System – USB 3.0 TX + Channel + USB 3.0 RX

- Impulse Response Improvement – By FFE + AGC + CTLE Circuit

© 2016 Cadence Design Systems, Inc. All rights reserved.

cādence®



A System – USB 3.0 TX + Channel + USB 3.0 RX

Data_in → TX AMI → Output Buffer → Channel (PCB+Conn+Cable) → Input Buffer → RX AMI → Data_out

AVDD

AVSS

4 preset taps: P0, P1, P2, P3

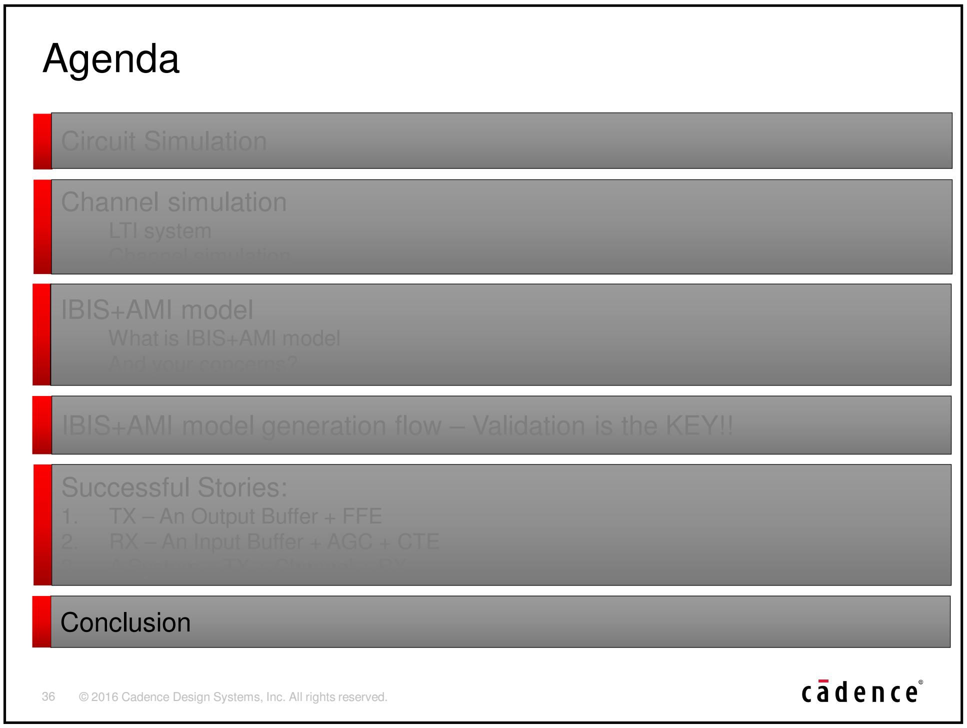
32 levels of strength: P0, P1, P2, ...P31

	Eye Contour	Distribution Eye – Circuit Netlist	Distribution Eye – IBIS+AMI model

TX=3dB
RX= 00110			

35 © 2016 Cadence Design Systems, Inc. All rights reserved.

cādence®

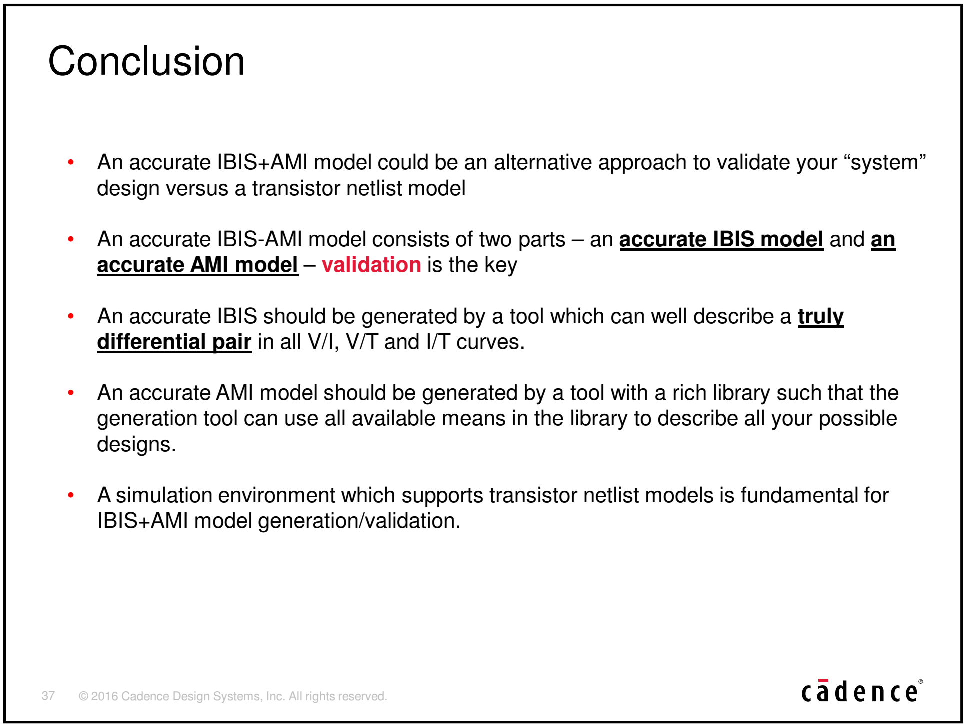


Agenda

- Circuit Simulation
- Channel simulation
  - LTI system
  - Channel simulation
- IBIS+AMI model
  - What is IBIS+AMI model
  - And your concerns?
- IBIS+AMI model generation flow – Validation is the KEY!!
- Successful Stories:
  1. TX – An Output Buffer + FFE
  2. RX – An Input Buffer + AGC + CTE
  3. A System – TX + Channel + RX
- Conclusion

36 © 2016 Cadence Design Systems, Inc. All rights reserved.

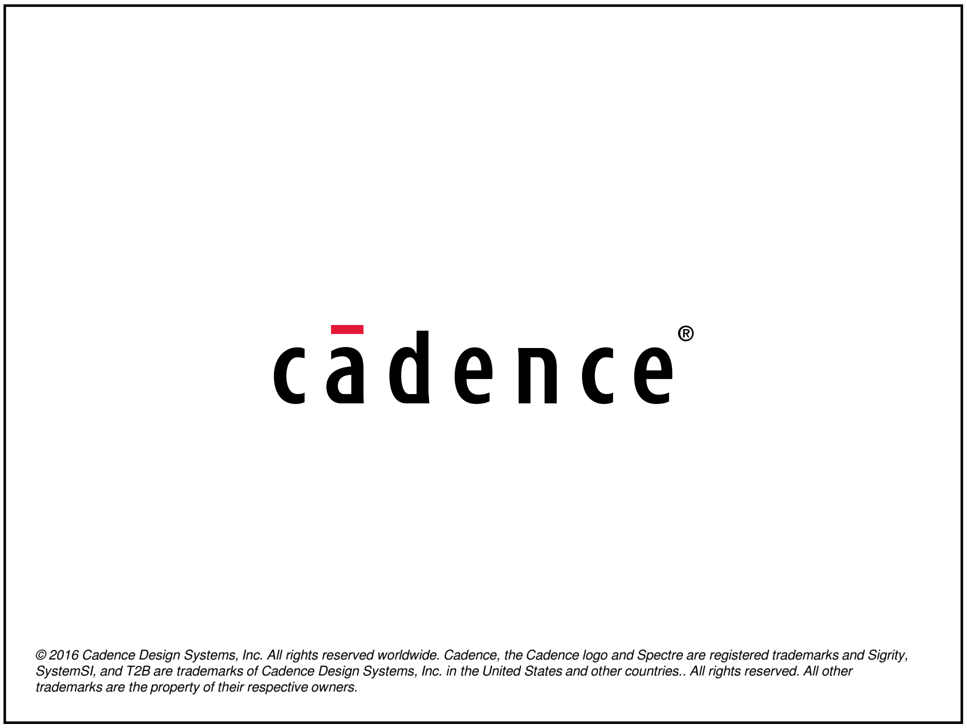
cādence®



Conclusion

- An accurate IBIS+AMI model could be an alternative approach to validate your “system” design versus a transistor netlist model
- An accurate IBIS-AMI model consists of two parts – an **accurate IBIS model** and **an accurate AMI model** – **validation** is the key
- An accurate IBIS should be generated by a tool which can well describe a **truly differential pair** in all V/I, V/T and I/T curves.
- An accurate AMI model should be generated by a tool with a rich library such that the generation tool can use all available means in the library to describe all your possible designs.
- A simulation environment which supports transistor netlist models is fundamental for IBIS+AMI model generation/validation.

© 2016 Cadence Design Systems, Inc. All rights reserved.



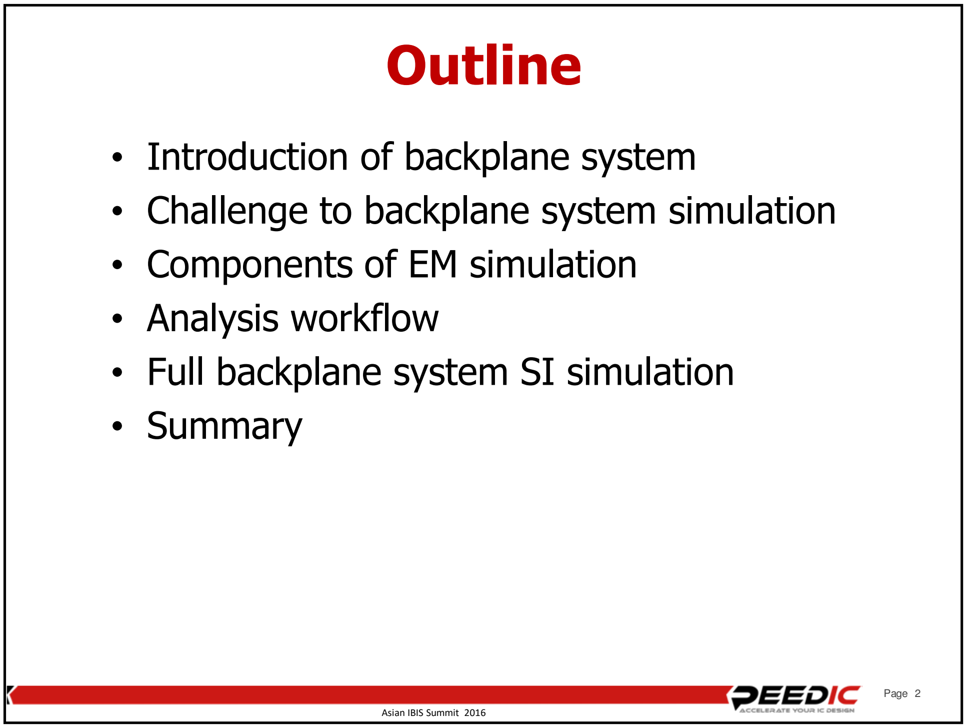
cadence®

© 2016 Cadence Design Systems, Inc. All rights reserved worldwide. Cadence, the Cadence logo and Spectre are registered trademarks and Sigrity, SystemSI, and T2B are trademarks of Cadence Design Systems, Inc. in the United States and other countries.. All rights reserved. All other trademarks are the property of their respective owners.

Achieving Full System Signal Integrity for High Speed Backplane System

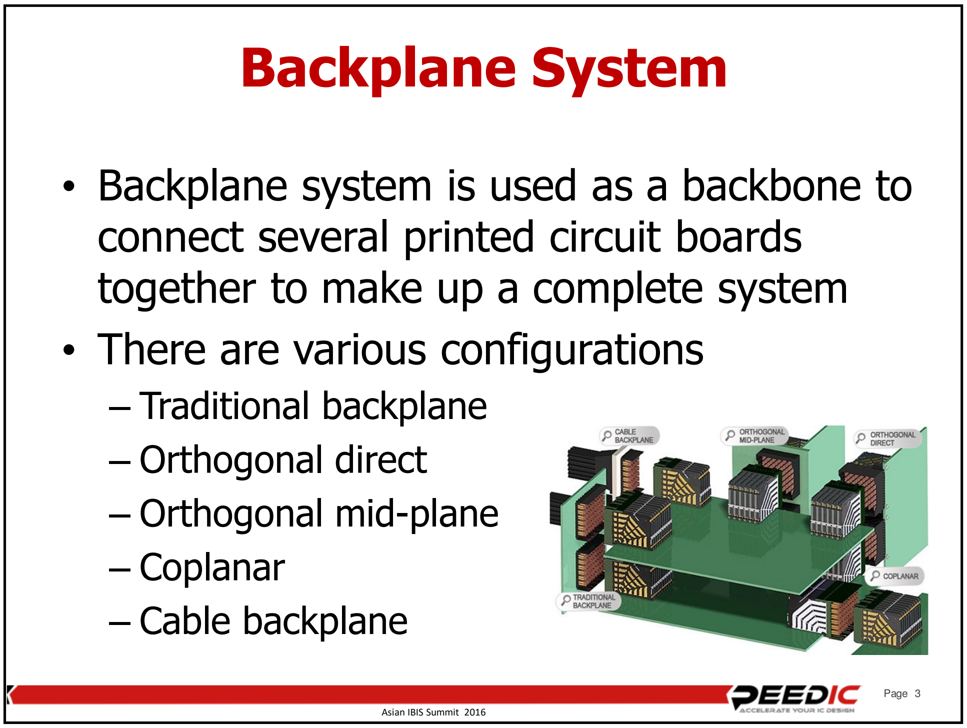
Dr. Wenliang Dai
Xpeedic Technology Co., Ltd.

Asian IBIS Summit
Taipei, Taiwan
November 14, 2016



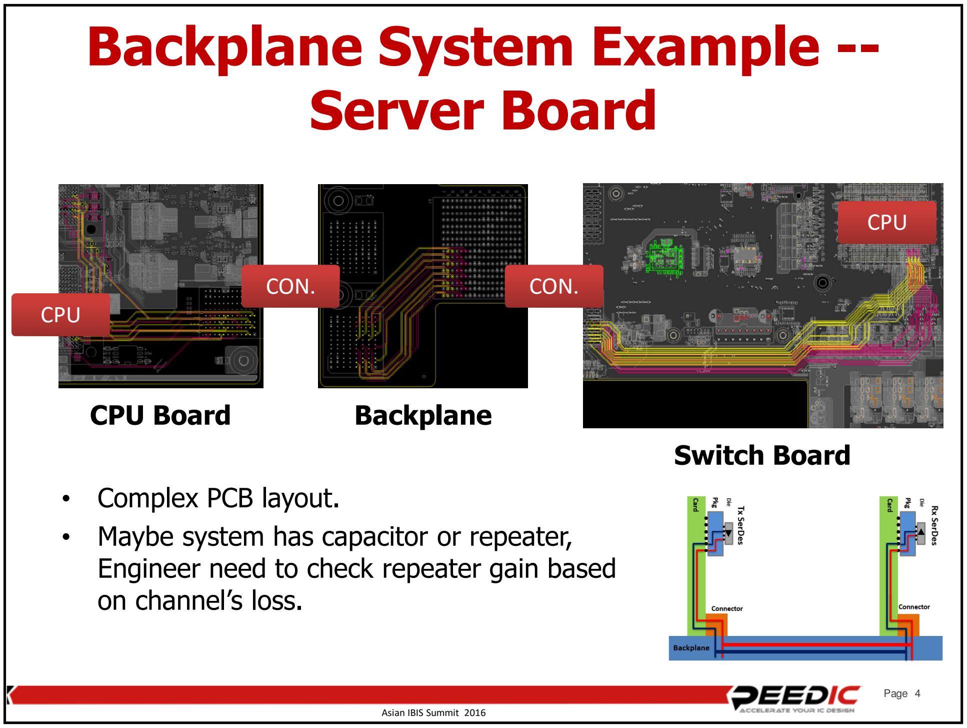
Outline

- Introduction of backplane system
- Challenge to backplane system simulation
- Components of EM simulation
- Analysis workflow
- Full backplane system SI simulation
- Summary



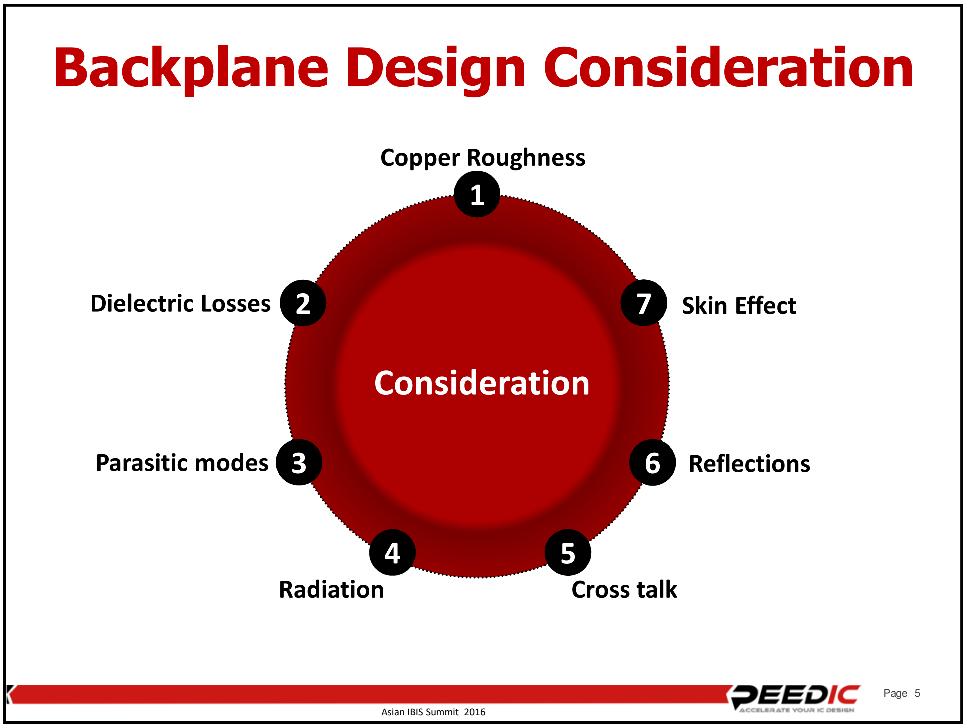
Backplane System

- Backplane system is used as a backbone to connect several printed circuit boards together to make up a complete system
- There are various configurations
  - Traditional backplane
  - Orthogonal direct
  - Orthogonal mid-plane
  - Coplanar
  - Cable backplane



Backplane System Example -- Server Board

- Complex PCB layout.
- Maybe system has capacitor or repeater, Engineer need to check repeater gain based on channel's loss.



Backplane Design Consideration

Consideration

1 Copper Roughness

2 Dielectric Losses

3 Parasitic modes

4 Radiation

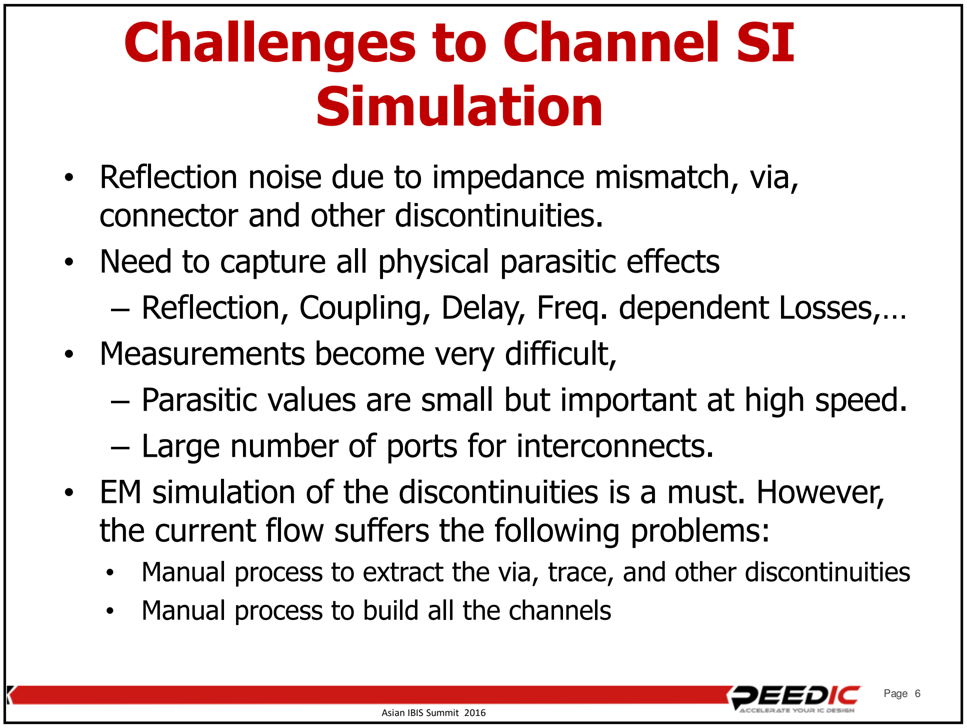
5 Cross talk

6 Reflections

7 Skin Effect

Asian IBIS Summit 2016

Page 5

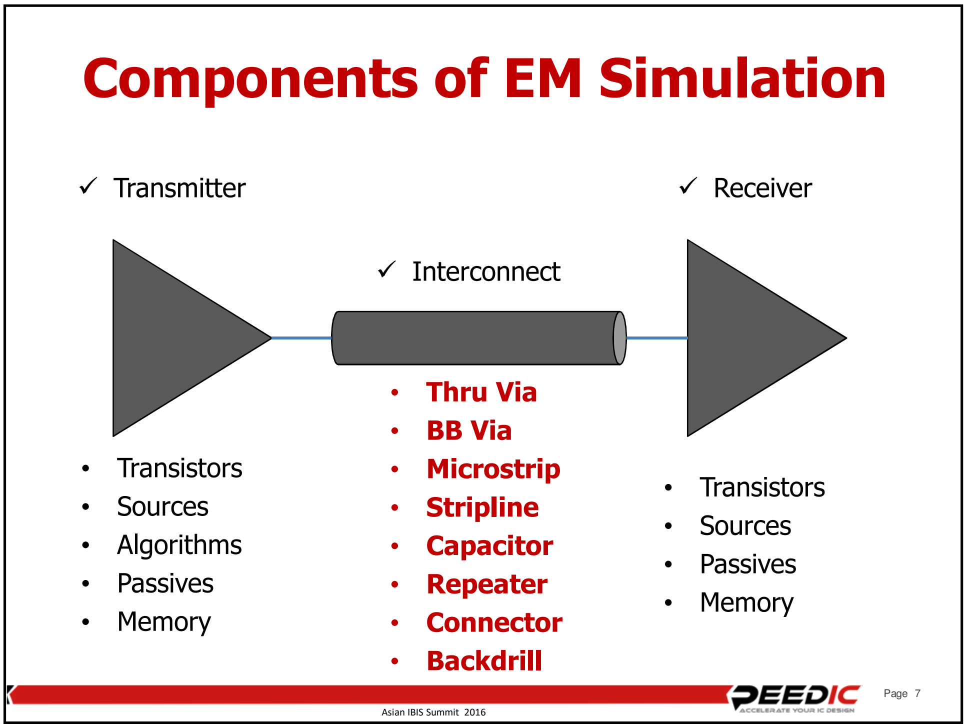


Challenges to Channel SI Simulation

- Reflection noise due to impedance mismatch, via, connector and other discontinuities.
- Need to capture all physical parasitic effects
  – Reflection, Coupling, Delay, Freq. dependent Losses,…
- Measurements become very difficult,
  – Parasitic values are small but important at high speed.
  – Large number of ports for interconnects.
- EM simulation of the discontinuities is a must. However, the current flow suffers the following problems:
  - Manual process to extract the via, trace, and other discontinuities
  - Manual process to build all the channels

Asian IBIS Summit 2016

Page 6



Components of EM Simulation

✓ Transmitter

✓ Interconnect

✓ Receiver

- Transistors
- Sources
- Algorithms
- Passives
- Memory

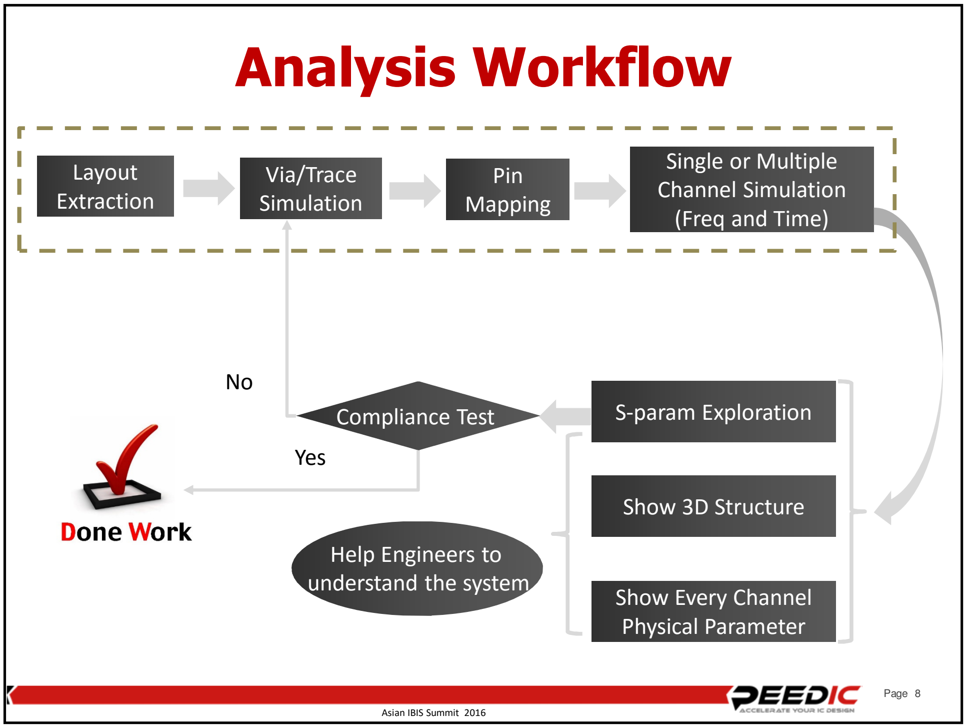
- **Thru Via**
- **BB Via**
- **Microstrip**
- **Stripline**
- **Capacitor**
- **Repeater**
- **Connector**
- **Backdrill**

- Transistors
- Sources
- Passives
- Memory

PEEDIC ACCELERATE YOUR IC DESIGN

Asian IBIS Summit 2016

Page 7



Analysis Workflow

Layout Extraction → Via/Trace Simulation → Pin Mapping → Single or Multiple Channel Simulation (Freq and Time)

S-param Exploration

Show 3D Structure

Show Every Channel Physical Parameter

Compliance Test

No

Yes

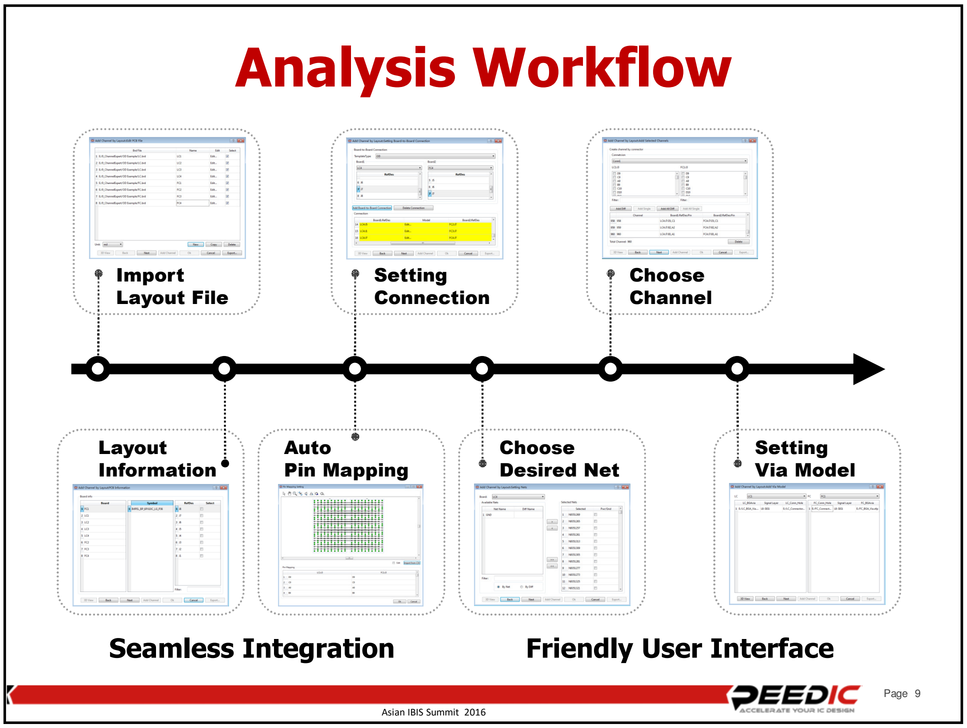
Done Work

Help Engineers to understand the system

PEEDIC ACCELERATE YOUR IC DESIGN

Asian IBIS Summit 2016

Page 8



Analysis Workflow

Import Layout File

Layout Information

Setting Connection

Auto Pin Mapping

Choose Desired Net

Choose Channel

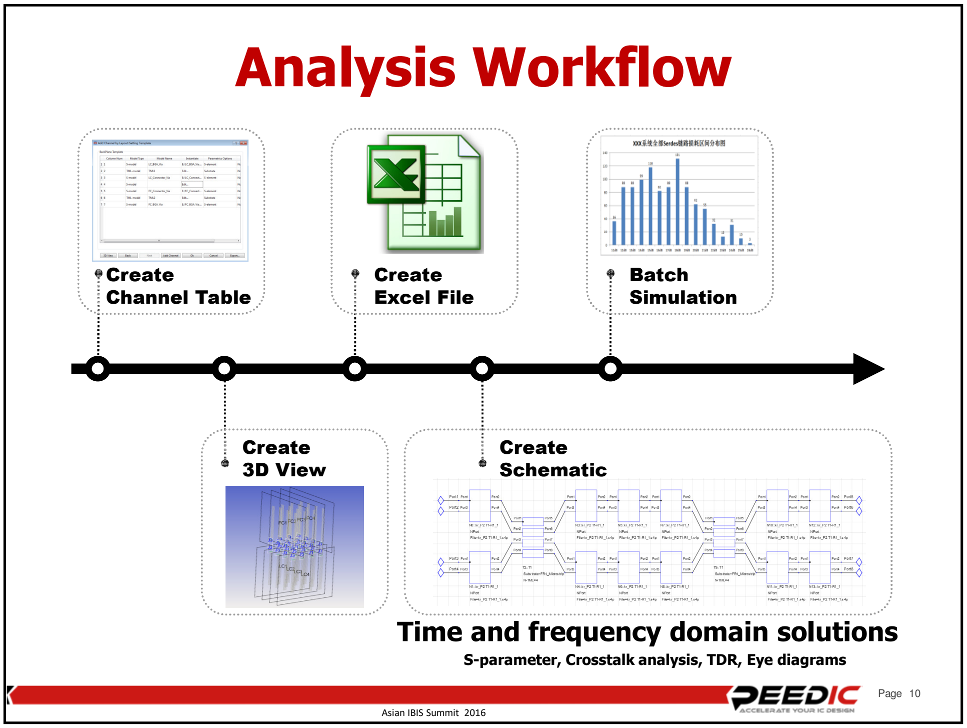
Setting Via Model

Seamless Integration

Friendly User Interface

Page 9

Asian IBIS Summit 2016



Analysis Workflow

Create Channel Table

Create 3D View

Create Excel File

Create Schematic

Batch Simulation

Time and frequency domain solutions

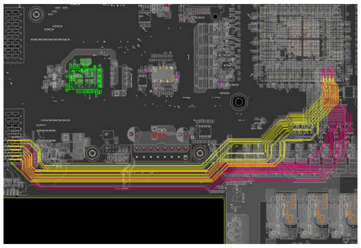
S-parameter, Crosstalk analysis, TDR, Eye diagrams

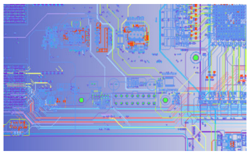
Page 10

Asian IBIS Summit 2016

Post-Layout Extraction

- Import all the boards for the backplane system
- Extract vias and traces from layout

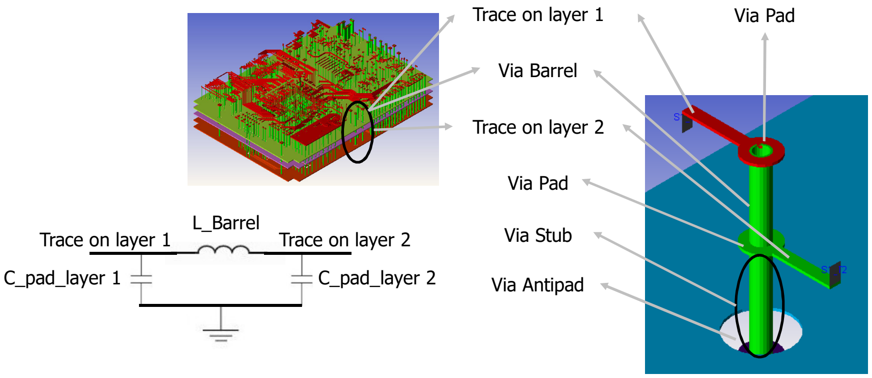




- Layout Files
- Model Template

- Self cleaning process
- Discrete components
- Area/Net selection
- Stackup definition
- Materials definition
- BB, Back drill definition
- Auto port definition
- Parameter/Optimization
- High flexibility

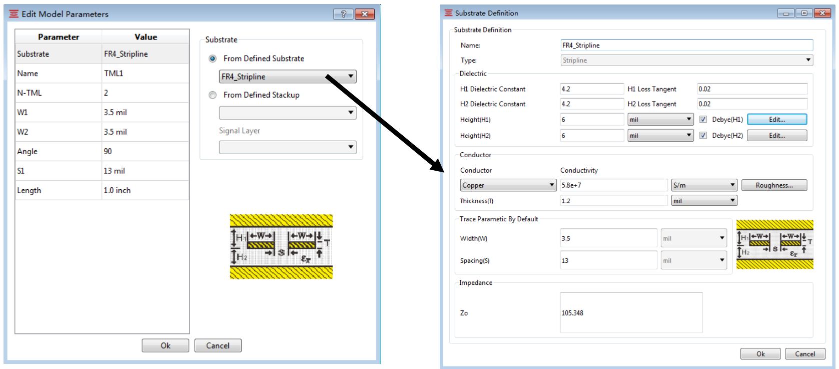
3D Via Modeling



- Via and ground plane will lead to parasitic capacitance and parasitic inductance.
- ***How to deal with a lot of via models?***

Trace Modeling

- For trace, the common approach is to use 2D models for main high speed interconnect.



Edit Model Parameters

Parameter	Value
Substrate	FR4_Stripline
Name	TML1
N-TML	2
W1	3.5 mil
W2	3.5 mil
Angle	90
S1	13 mil
Length	1.0 inch

Substrate

● From Defined Substrate: FR4_Stripline

○ From Defined Stackup

Signal Layer

Ok Cancel

Substrate Definition

Substrate Definition

Name: FR4_Stripline

Type: Stripline

Dielectric

H1 Dielectric Constant	4.2	H1 Loss Tangent	0.02	
H2 Dielectric Constant	4.2	H2 Loss Tangent	0.02	
Height(H1)	6	mil	☑ Debye(H1)	Edit...
Height(H2)	6	mil	☑ Debye(H2)	Edit...

Conductor

Conductor	Conductivity		
Copper	5.8e+7	S/m	Roughness...
Thickness(T)	1.2	mil	

Trace Parametic By Default

Width(W)	3.5	mil
Spacing(S)	13	mil

Impedance

Zo 105.348

Ok Cancel

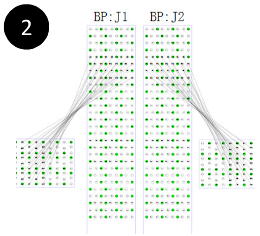
Connection between Boards through Pin Mapping

1

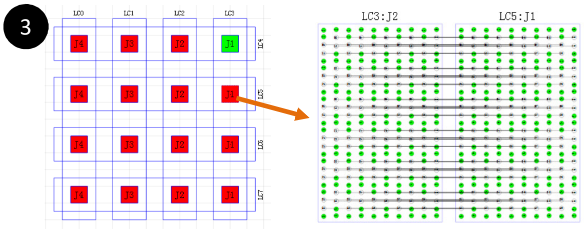
	DC1:J1	BP:J1	Net	BP:J2	DC2:J1
1	A5	A5	W30...	A5	A5
2	A7	A7	W30...	A7	A7
3	B5	B5	W30...	B5	B5
4	B6	B6	W30...	B6	B6
5	B7	B7	W30...	B7	B7
6	B8	B8	W30...	B8	B8
7	C6	C6	W30...	C6	C6
8	C8	C8	W30...	C8	C8
9	D5	D5	W30...	D5	D5
10	D7	D7	W30...	D7	D7

□ Edit Import from CSV

2

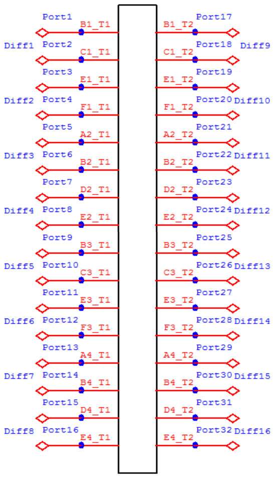


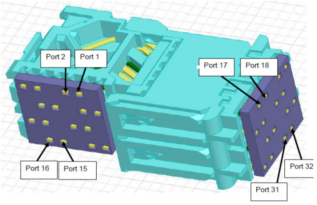
3

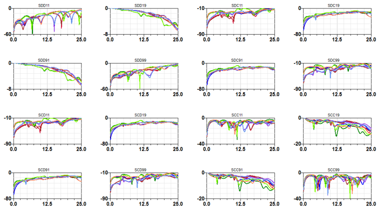


- Import .csv file.
- Create connection manually
- Create connection base on slot ID.

Connector S-parameter from Vendor

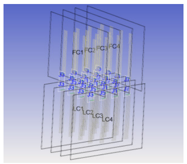




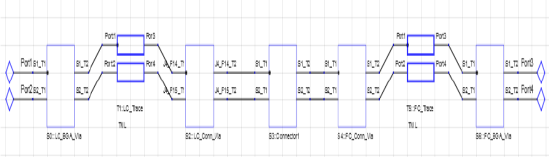


- Connector S-parameter file comes from vendor

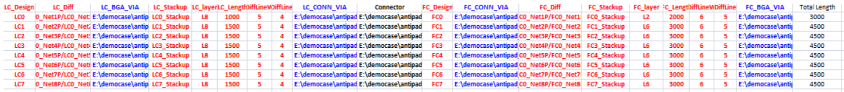
Auto Cascading to Create Channel



Channelview



Cascading



LC_Design	LC_Diff	LC_BGA_VIA	LC_Stackup	LC_layer	LC_Length	iffLineW	iffLine	LC_CONN_VIA	Connector	FC_Design	FC_CONN_VIA	FC_Diff	FC_Stackup	FC_layer	C_Lengt	iffLineW	DiffLine	FC_BGA_VIA	Total Length
LC0	0_Net1P/LC0_Net	E:\democase\antip	LC0_Stackup	L8	1000	5	4	E:\democase\antipad	E:\democase\antipad	FC0	E:\democase\antipad	C0_Net1P/FC0_Net1	FC0_Stackup	L2	2000	6	5	E:\democase\anti	3000
LC1	0_Net2P/LC0_Net	E:\democase\antip	LC1_Stackup	L8	1500	5	4	E:\democase\antipad	E:\democase\antipad	FC1	E:\democase\antipad	C0_Net2P/FC0_Net2	FC1_Stackup	L6	3000	6	5	E:\democase\anti	4500
LC2	0_Net3P/LC0_Net	E:\democase\antip	LC2_Stackup	L8	1500	5	4	E:\democase\antipad	E:\democase\antipad	FC2	E:\democase\antipad	C0_Net3P/FC0_Net3	FC2_Stackup	L6	3000	6	5	E:\democase\anti	4500
LC3	0_Net4P/LC0_Net	E:\democase\antip	LC3_Stackup	L8	1500	5	4	E:\democase\antipad	E:\democase\antipad	FC3	E:\democase\antipad	C0_Net4P/FC0_Net4	FC3_Stackup	L6	3000	6	5	E:\democase\anti	4500
LC4	0_Net5P/LC0_Net	E:\democase\antip	LC4_Stackup	L8	1500	5	4	E:\democase\antipad	E:\democase\antipad	FC4	E:\democase\antipad	C0_Net5P/FC0_Net5	FC4_Stackup	L6	3000	6	5	E:\democase\anti	4500
LC5	0_Net6P/LC0_Net	E:\democase\antip	LC5_Stackup	L8	1500	5	4	E:\democase\antipad	E:\democase\antipad	FC5	E:\democase\antipad	C0_Net6P/FC0_Net6	FC5_Stackup	L6	3000	6	5	E:\democase\anti	4500
LC6	0_Net7P/LC0_Net	E:\democase\antip	LC6_Stackup	L8	1500	5	4	E:\democase\antipad	E:\democase\antipad	FC6	E:\democase\antipad	C0_Net7P/FC0_Net7	FC6_Stackup	L6	3000	6	5	E:\democase\anti	4500
LC7	0_Net8P/LC0_Net	E:\democase\antip	LC7_Stackup	L8	1500	5	4	E:\democase\antipad	E:\democase\antipad	FC7	E:\democase\antipad	C0_Net8P/FC0_Net8	FC7_Stackup	L6	3000	6	5	E:\democase\anti	4500

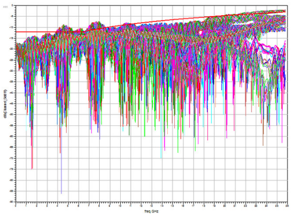
Excel File

- 3D structure of BP system
- Hide/Show board or net in 3D
- Show entire channel path
- Single vs. multiple simulation and cascading

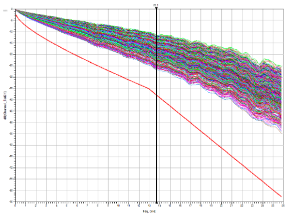
- Fast and memory efficient
- Show channel's structure size in Excel file
- Easily find worst channel
- Easily generate Excel report

Full Backplane SI Simulation - Frequency Domain

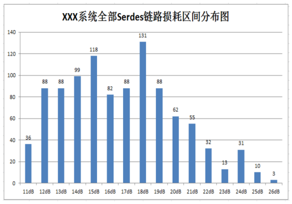
- Full backplane SI simulation is achieved by sweeping all the channels



Full Net Return Loss

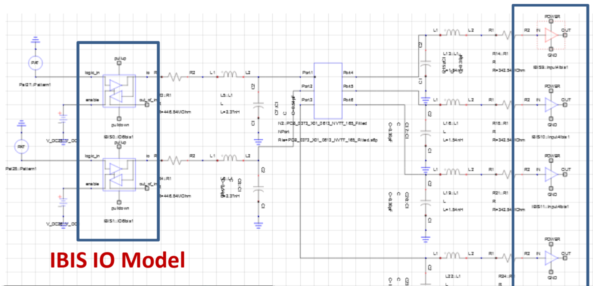


Full Net Insertion Loss

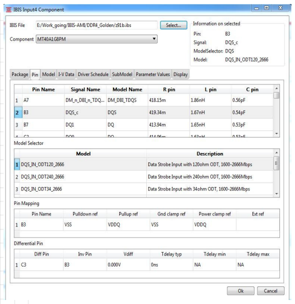


Net Distribution @ 13.67 GHz

Full Backplane SI Simulation - Time Domain



IBIS IO Model



IBIS Input4 Component

IBIS File: E:/Work_going/IBIS-AMI/DDR4_Golden/z91b.ibs

Component: MT40A1G8PM

Information on selected
Pin: B3
Signal: DQS_c
ModelSelector: DQS
Model: DQS_IN_ODT120_2666

	Pin Name	Signal Name	Model Name	R pin	L pin	C pin
1	A7	DM_n_DBI_n_TDQ...	DM_DBI_TDQS	418.15m	1.86nH	0.56pF
2	B3	DQS_c	DQS	419.34m	1.67nH	0.54pF
3	B7	DQ1	DQ	413.94m	1.65nH	0.53pF
4	C2	DQ0	DQ	414.06m	1.63nH	0.52pF

Model Selector

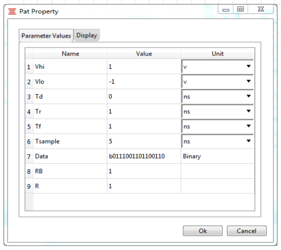
	Model	Description
1	DQS_IN_ODT120_2666	Data Strobe Input with 120ohm ODT, 1600-2666Mbps
2	DQS_IN_ODT240_2666	Data Strobe Input with 240ohm ODT, 1600-2666Mbps
3	DQS_IN_ODT34_2666	Data Strobe Input with 34ohm ODT, 1600-2666Mbps

Pin Mapping

	Pin Name	Pulldown ref	Pullup ref	Gnd clamp ref	Power clamp ref	Ext ref
1	B3	VSS	VDDQ	VSS	VDDQ	

Differential Pin

	Diff Pin	Inv Pin	Vdiff	Tdelay typ	Tdelay min	Tdelay max
1	C3	B3	0.000V	0ns	NA	NA



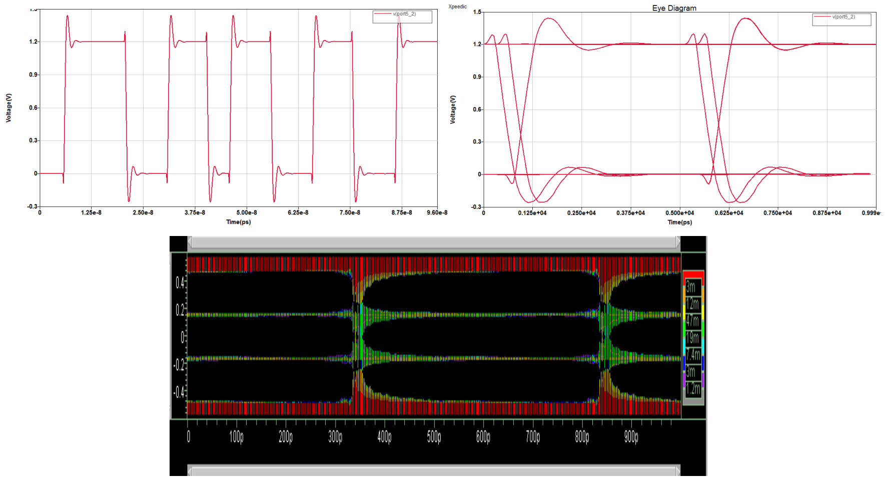
Pat Property

	Name	Value	Unit
1	Vhi	1	v
2	Vlo	-1	v
3	Td	0	ns
4	Tr	1	ns
5	Tf	1	ns
6	Tsample	5	ns
7	Data	b0111001101100110	Binary
8	RB	1	
9	R	1	

IBIS Input Model

- Support IBIS, IBIS-AMI, Spice model.
- Add proper pattern model.
- Fast modeling and high accuracy.

Full Backplane SI Simulation - Time Domain

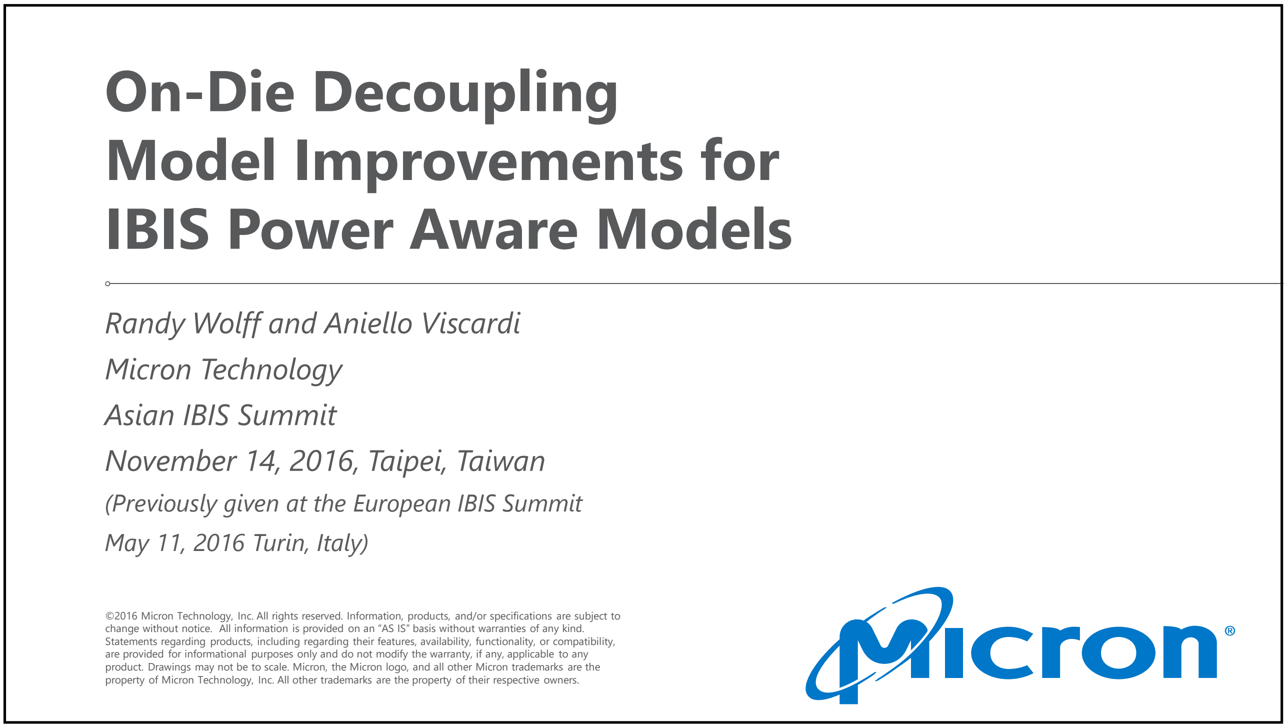


Summary

- Passive channel modeling and simulation is essential to high speed channel design.
- Optimal channel design requires user friendly EDA tool to do layout extraction, via optimization, trace simulation, S-parameter cascading, S-parameter exploration, etc.
- Full backplane system SI simulation is achieved by sweeping all the channels with correct models.

XPEEDIC

ACCELERATE YOUR IC DESIGN



On-Die Decoupling Model Improvements for IBIS Power Aware Models

Randy Wolff and Aniello Viscardi

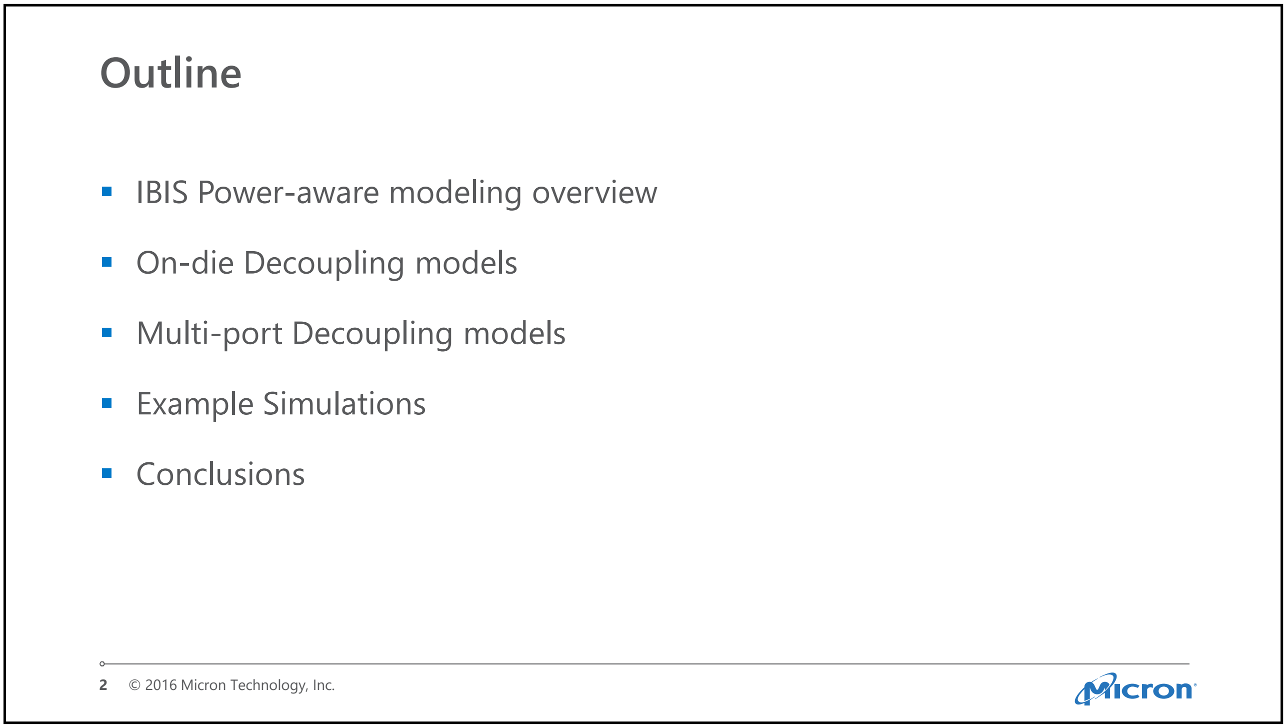
Micron Technology

Asian IBIS Summit

November 14, 2016, Taipei, Taiwan

(Previously given at the European IBIS Summit May 11, 2016 Turin, Italy)

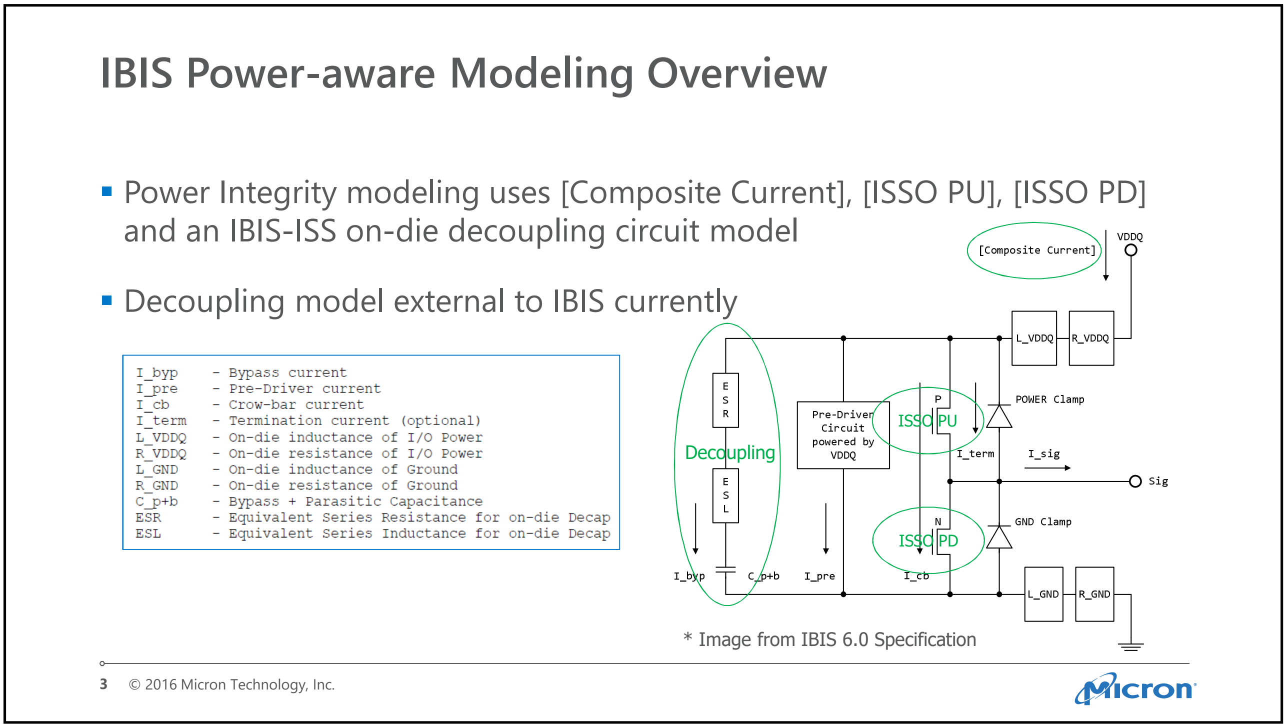
©2016 Micron Technology, Inc. All rights reserved. Information, products, and/or specifications are subject to change without notice. All information is provided on an "AS IS" basis without warranties of any kind. Statements regarding products, including regarding their features, availability, functionality, or compatibility, are provided for informational purposes only and do not modify the warranty, if any, applicable to any product. Drawings may not be to scale. Micron, the Micron logo, and all other Micron trademarks are the property of Micron Technology, Inc. All other trademarks are the property of their respective owners.



Outline

- IBIS Power-aware modeling overview
- On-die Decoupling models
- Multi-port Decoupling models
- Example Simulations
- Conclusions

2 © 2016 Micron Technology, Inc.



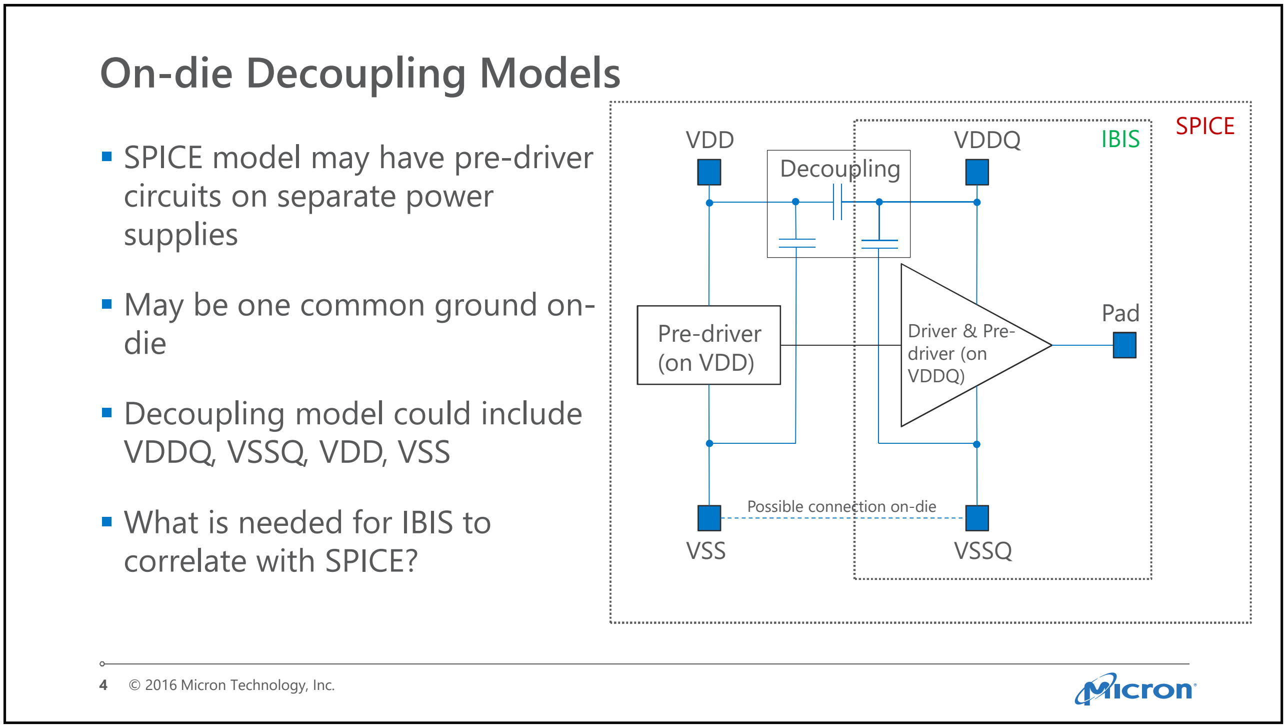
IBIS Power-aware Modeling Overview

- Power Integrity modeling uses [Composite Current], [ISSO PU], [ISSO PD] and an IBIS-ISS on-die decoupling circuit model
- Decoupling model external to IBIS currently

I_byp	- Bypass current
I_pre	- Pre-Driver current
I_cb	- Crow-bar current
I_term	- Termination current (optional)
L_VDDQ	- On-die inductance of I/O Power
R_VDDQ	- On-die resistance of I/O Power
L_GND	- On-die inductance of Ground
R_GND	- On-die resistance of Ground
C_p+b	- Bypass + Parasitic Capacitance
ESR	- Equivalent Series Resistance for on-die Decap
ESL	- Equivalent Series Inductance for on-die Decap

* Image from IBIS 6.0 Specification

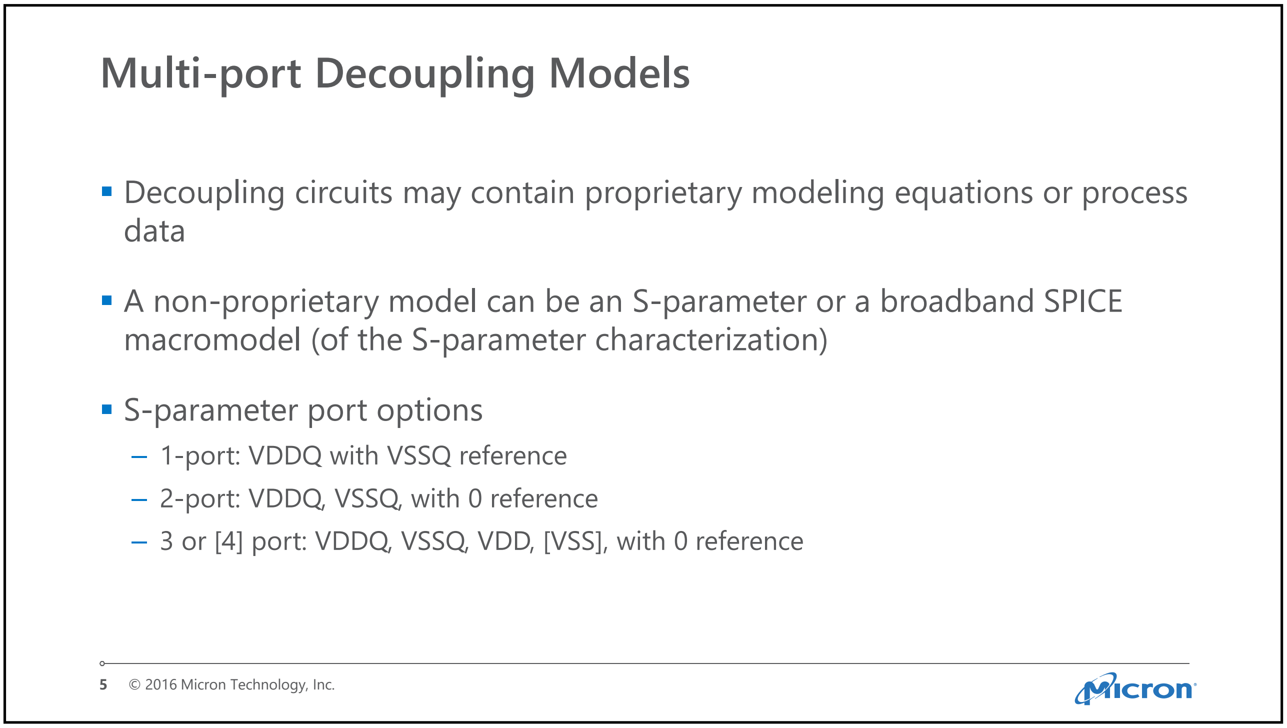
3 © 2016 Micron Technology, Inc.



On-die Decoupling Models

- SPICE model may have pre-driver circuits on separate power supplies
- May be one common ground on-die
- Decoupling model could include VDDQ, VSSQ, VDD, VSS
- What is needed for IBIS to correlate with SPICE?

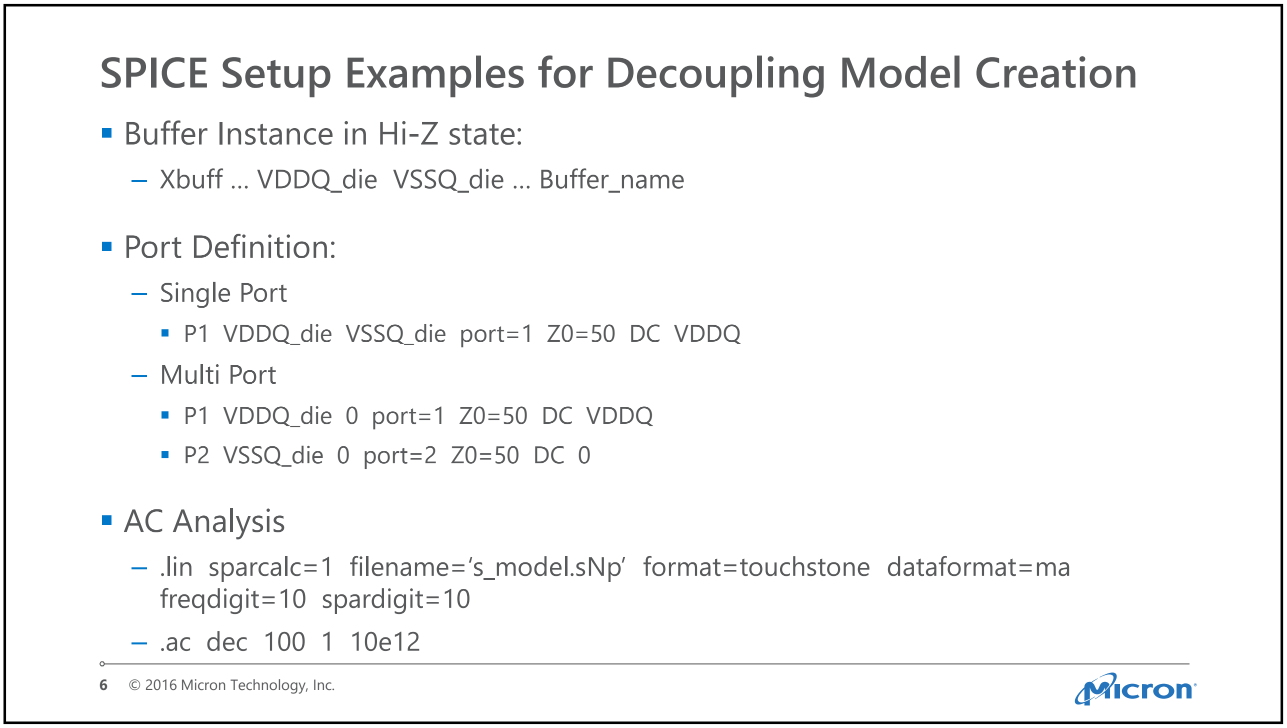
4 © 2016 Micron Technology, Inc.



Multi-port Decoupling Models

- Decoupling circuits may contain proprietary modeling equations or process data
- A non-proprietary model can be an S-parameter or a broadband SPICE macromodel (of the S-parameter characterization)
- S-parameter port options
  - 1-port: VDDQ with VSSQ reference
  - 2-port: VDDQ, VSSQ, with 0 reference
  - 3 or [4] port: VDDQ, VSSQ, VDD, [VSS], with 0 reference

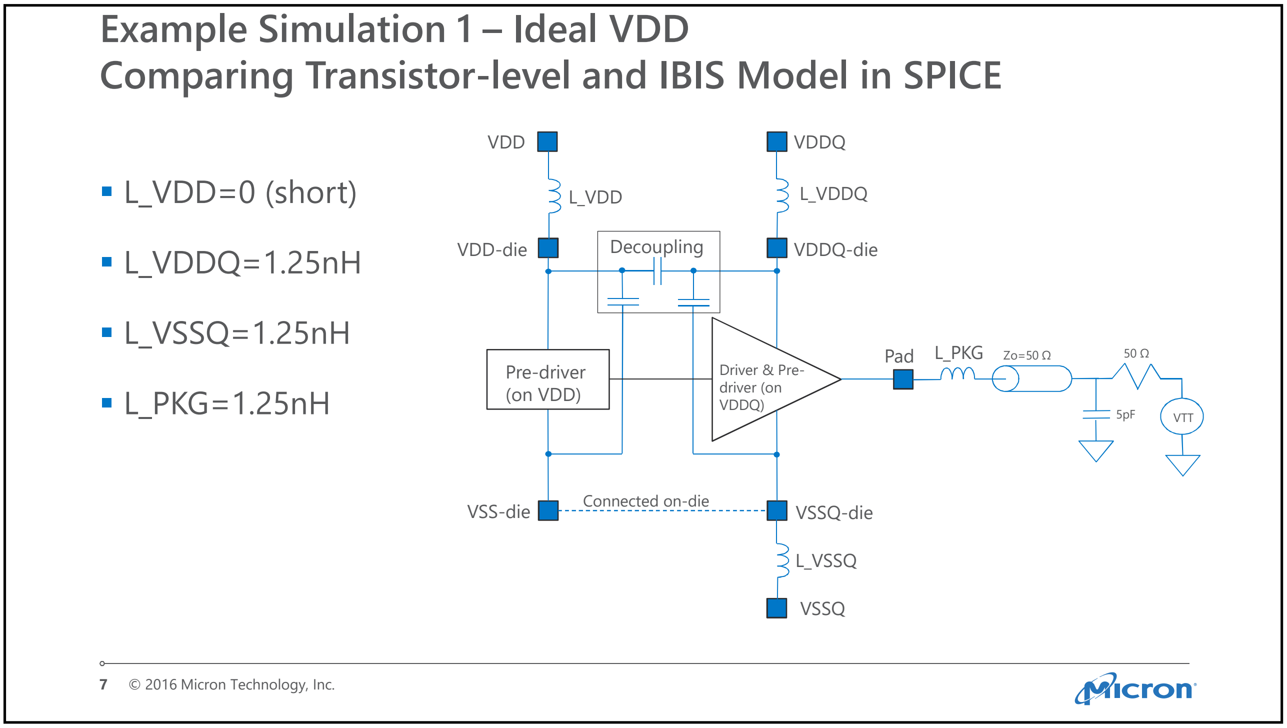
© 2016 Micron Technology, Inc.



SPICE Setup Examples for Decoupling Model Creation

- Buffer Instance in Hi-Z state:
  - Xbuff … VDDQ_die VSSQ_die … Buffer_name
- Port Definition:
  - Single Port
    - P1 VDDQ_die VSSQ_die port=1 Z0=50 DC VDDQ
  - Multi Port
    - P1 VDDQ_die 0 port=1 Z0=50 DC VDDQ
    - P2 VSSQ_die 0 port=2 Z0=50 DC 0
- AC Analysis
  - .lin sparcalc=1 filename=‘s_model.sNp’ format=touchstone dataformat=ma freqdigit=10 spardigit=10
  - .ac dec 100 1 10e12

© 2016 Micron Technology, Inc.



Example Simulation 1 – Ideal VDD
Comparing Transistor-level and IBIS Model in SPICE

- L_VDD=0 (short)
- L_VDDQ=1.25nH
- L_VSSQ=1.25nH
- L_PKG=1.25nH

VDD

L_VDD

VDD-die

Decoupling

VDDQ

L_VDDQ

VDDQ-die

Pre-driver (on VDD)

Driver & Pre-driver (on VDDQ)

Pad

L_PKG

Zo=50 Ω

50 Ω

5pF

VTT

VSS-die

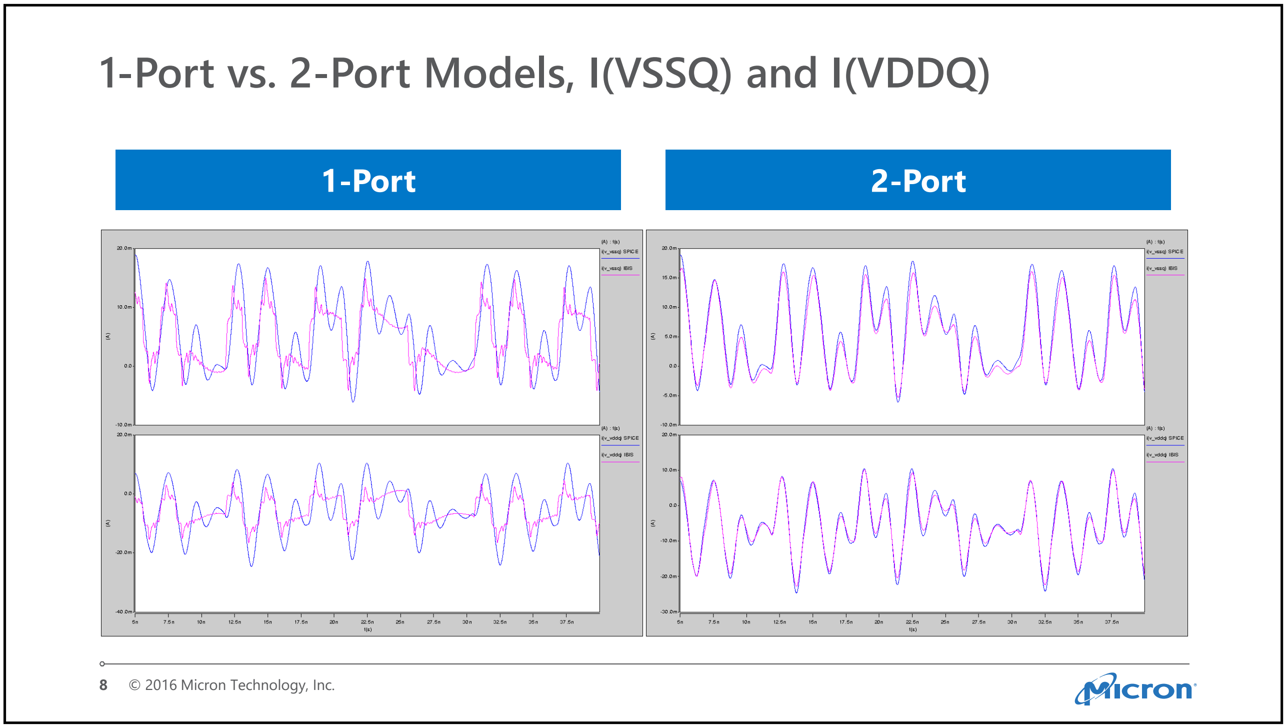
Connected on-die

VSSQ-die

L_VSSQ

VSSQ

7 © 2016 Micron Technology, Inc.



1-Port vs. 2-Port Models, I(VSSQ) and I(VDDQ)

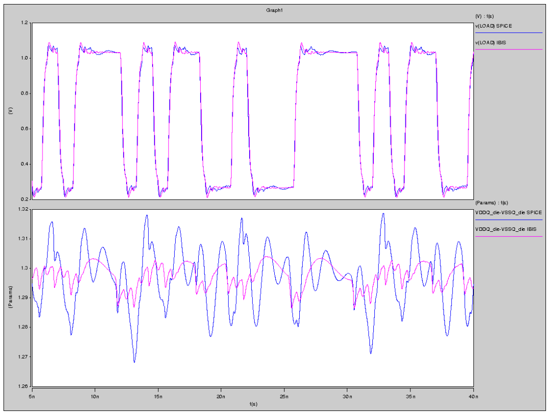
1-Port

2-Port

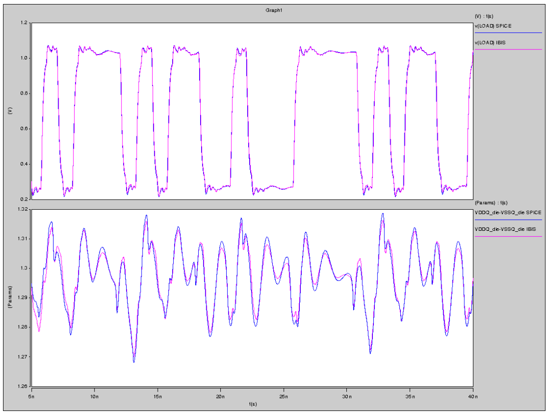
8 © 2016 Micron Technology, Inc.

V(LOAD) and V(VDDQ_die)-V(VSSQ_die)

1-Port

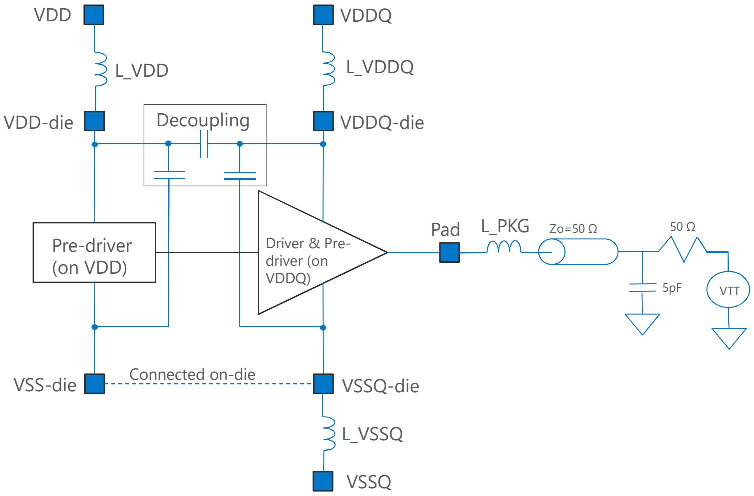


2-Port



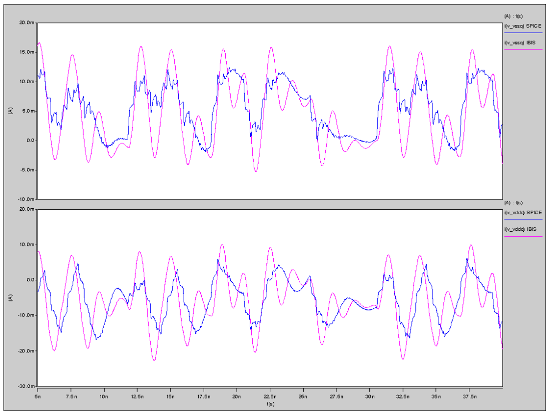
Example Simulation 2 – Non-ideal VDD
Comparing Transistor-level and IBIS Model in SPICE

- L_VDD=1.25nH
- L_VDDQ=1.25nH
- L_VSSQ=1.25nH
- L_PKG=1.25nH

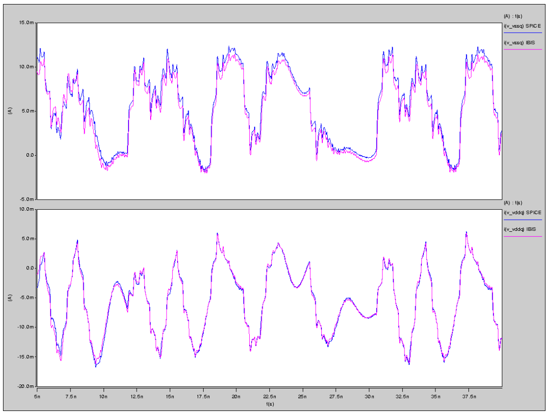


2-Port vs. 3-Port Models, I(VSSQ) and I(VDDQ)

2-Port



3-Port

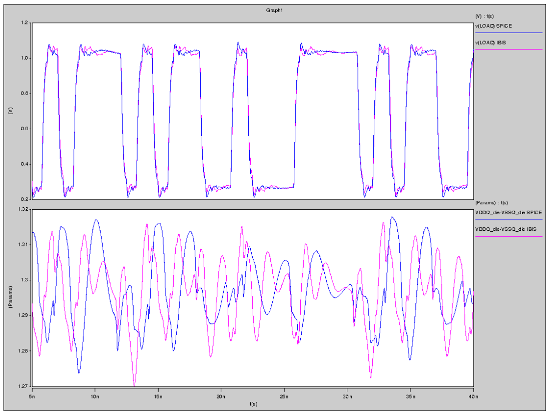


© 2016 Micron Technology, Inc.

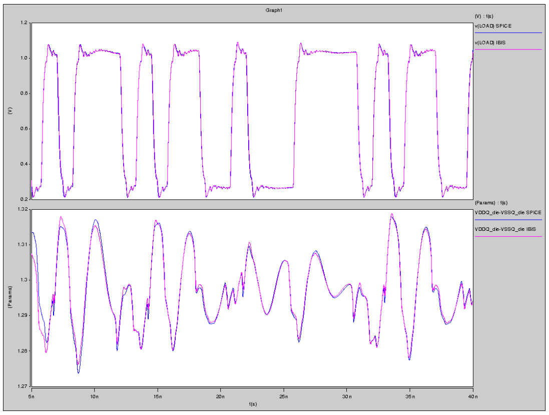
Micron

V(LOAD) and V(VDDQ_die)-V(VSSQ_die)

2-Port

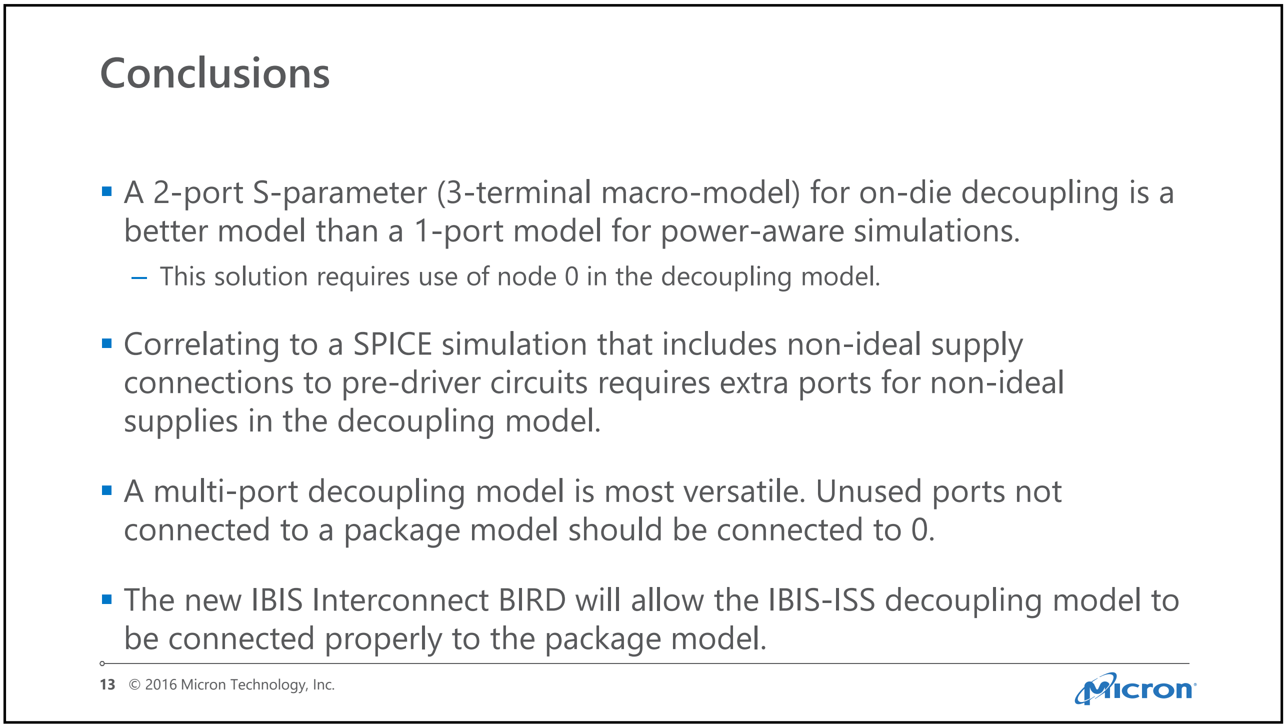


3-Port



© 2016 Micron Technology, Inc.

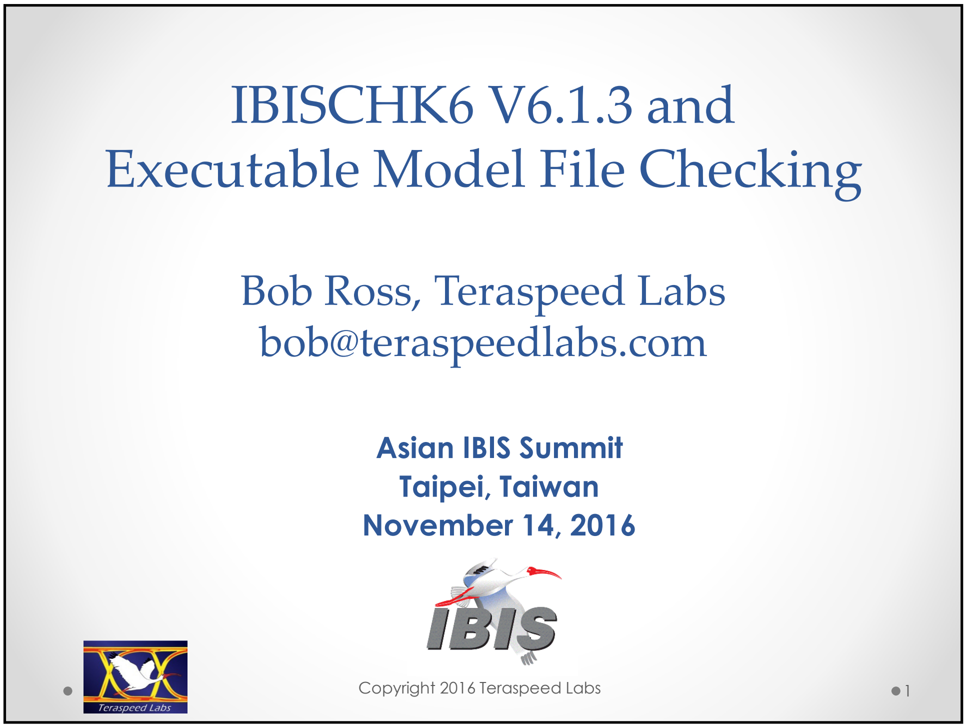
Micron



Conclusions

- A 2-port S-parameter (3-terminal macro-model) for on-die decoupling is a better model than a 1-port model for power-aware simulations.
  - This solution requires use of node 0 in the decoupling model.
- Correlating to a SPICE simulation that includes non-ideal supply connections to pre-driver circuits requires extra ports for non-ideal supplies in the decoupling model.
- A multi-port decoupling model is most versatile. Unused ports not connected to a package model should be connected to 0.
- The new IBIS Interconnect BIRD will allow the IBIS-ISS decoupling model to be connected properly to the package model.

13 © 2016 Micron Technology, Inc.

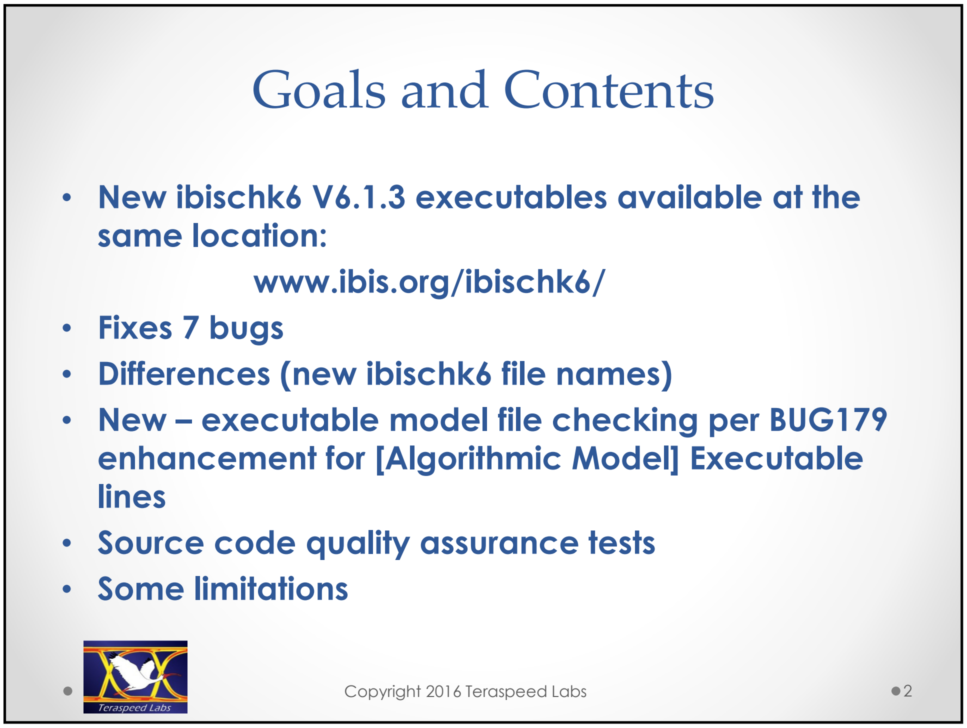


IBISCHK6 V6.1.3 and Executable Model File Checking

Bob Ross, Teraspeed Labs
bob@teraspeedlabs.com

Asian IBIS Summit
Taipei, Taiwan
November 14, 2016

Copyright 2016 Teraspeed Labs



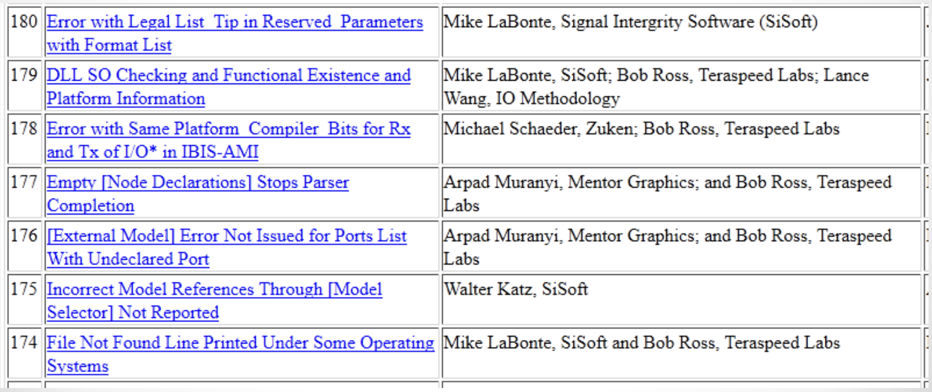
Goals and Contents

- **New ibischk6 V6.1.3 executables available at the same location:**

  www.ibis.org/ibischk6/
- **Fixes 7 bugs**
- **Differences (new ibischk6 file names)**
- **New – executable model file checking per BUG179 enhancement for [Algorithmic Model] Executable lines**
- **Source code quality assurance tests**
- **Some limitations**

Copyright 2016 Teraspeed Labs

New BUGs 174-180 Fixed www.ibis.com/bugs/ibischk/



180	Error with Legal List Tip in Reserved Parameters with Format List	Mike LaBonte, Signal Intergrity Software (SiSoft)
179	DLL SO Checking and Functional Existence and Platform Information	Mike LaBonte, SiSoft; Bob Ross, Teraspeed Labs; Lance Wang, IO Methodology
178	Error with Same Platform Compiler Bits for Rx and Tx of I/O* in IBIS-AMI	Michael Schaeder, Zuken; Bob Ross, Teraspeed Labs
177	Empty [Node Declarations] Stops Parser Completion	Arpad Muranyi, Mentor Graphics; and Bob Ross, Teraspeed Labs
176	[External Model] Error Not Issued for Ports List With Undeclared Port	Arpad Muranyi, Mentor Graphics; and Bob Ross, Teraspeed Labs
175	Incorrect Model References Through [Model Selector] Not Reported	Walter Katz, SiSoft
174	File Not Found Line Printed Under Some Operating Systems	Mike LaBonte, SiSoft and Bob Ross, Teraspeed Labs

Copyright 2016 Teraspeed Labs

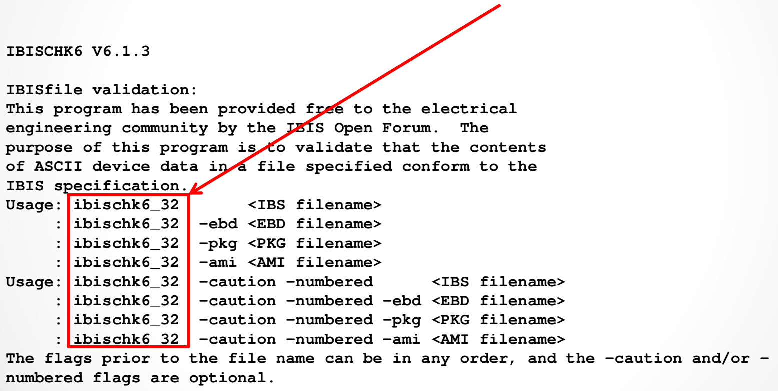
New Names for ibischk6

- Command line operation:
- Windows:
  - ibischk6_32.exe (versus ibischk6.exe)
  - ibischk6_64.exe (versus ibischk6.exe)
- Linux (Ubuntu):
  - ibischk6_32 (versus ibischk6)
  - ibischk6_64 (versus ibischk6)
- Used for source code quality assurance tests per BUG179
- Macintosh osx_32, osx_64 (also uploaded), but not part of source code release

Copyright 2016 Teraspeed Labs

New Command Line Based on ibischk6 Name

Example: ibischk6_32.exe ➔"ibischk6_32" command



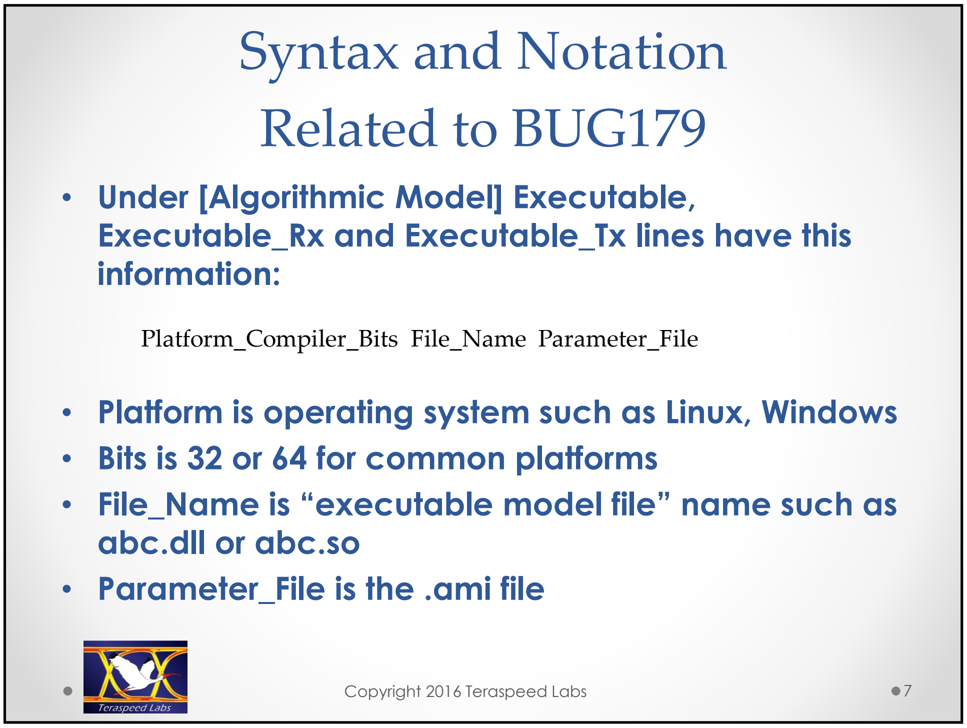
```
IBISCHK6 V6.1.3

IBISfile validation:
This program has been provided free to the electrical
engineering community by the IBIS Open Forum.  The
purpose of this program is to validate that the contents
of ASCII device data in a file specified conform to the
IBIS specification.
Usage: ibischk6_32       <IBS filename>
     : ibischk6_32  -ebd <EBD filename>
     : ibischk6_32  -pkg <PKG filename>
     : ibischk6_32  -ami <AMI filename>
Usage: ibischk6_32  -caution -numbered      <IBS filename>
     : ibischk6_32  -caution -numbered -ebd <EBD filename>
     : ibischk6_32  -caution -numbered -pkg <PKG filename>
     : ibischk6_32  -caution -numbered -ami <AMI filename>
The flags prior to the file name can be in any order, and the -caution and/or -
numbered flags are optional.
```

Copyright 2016 Teraspeed Labs

Can Rename ibischk6 File (Examples Below)

- **ibischk6.exe as in earlier versions for "ibischk6" command**
- **ibischk613_32.exe showing version and bits detail for "ibischk613_32" command**
- **Etc. - the new ibischk6 file name shows up in the Usage: lines, as shown on previous slide**

Copyright 2016 Teraspeed Labs



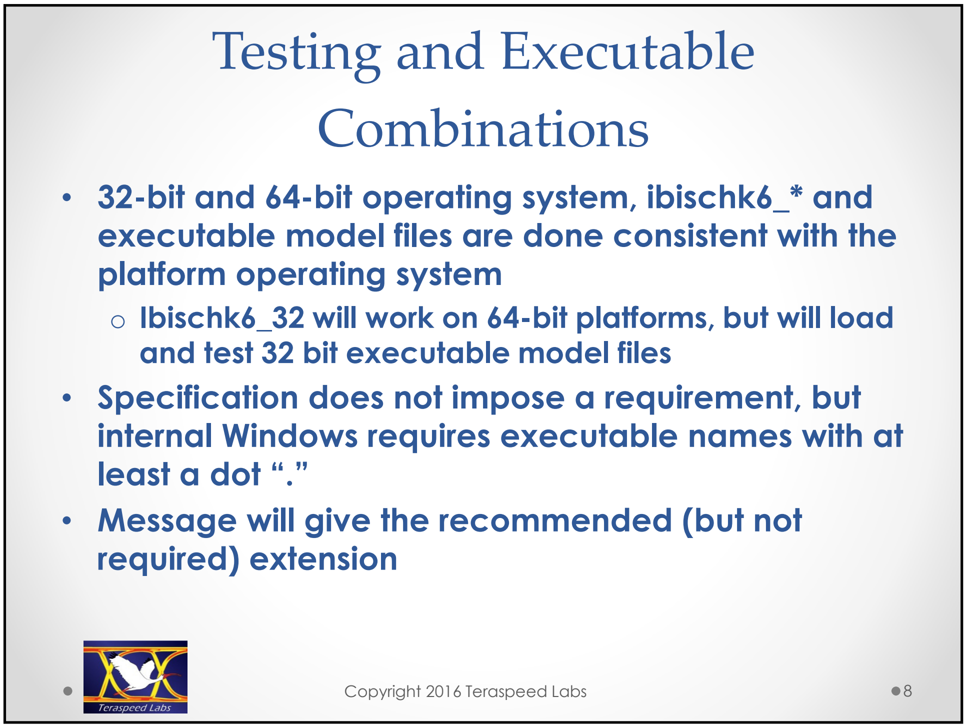
Syntax and Notation Related to BUG179

- Under [Algorithmic Model] Executable, Executable_Rx and Executable_Tx lines have this information:

```
Platform_Compiler_Bits  File_Name  Parameter_File
```

- Platform is operating system such as Linux, Windows
- Bits is 32 or 64 for common platforms
- File_Name is “executable model file” name such as abc.dll or abc.so
- Parameter_File is the .ami file

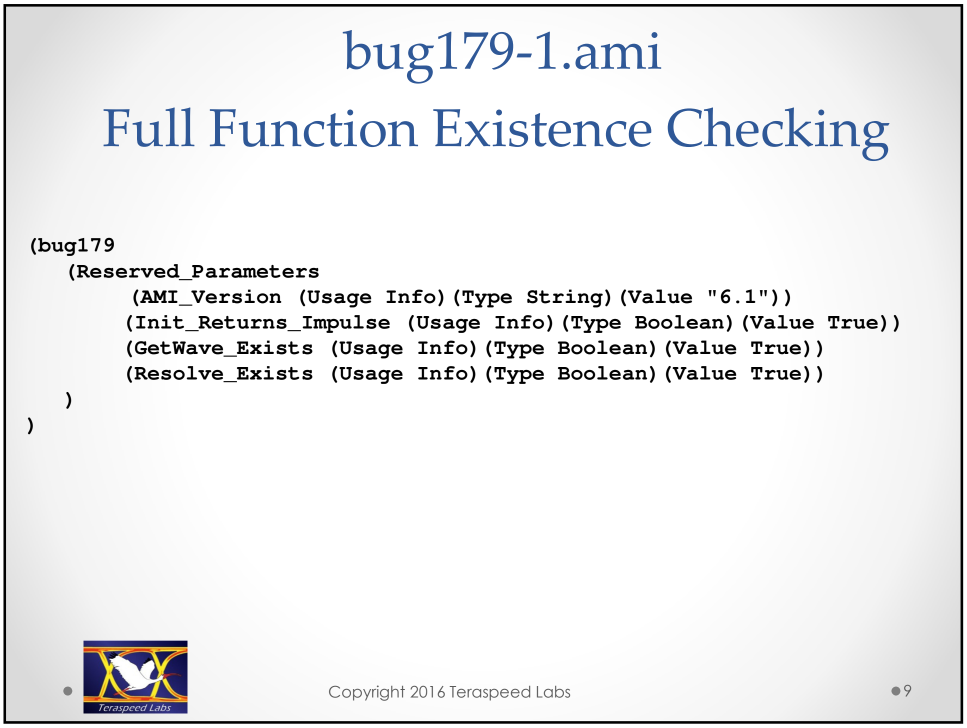
Copyright 2016 Teraspeed Labs



Testing and Executable Combinations

- 32-bit and 64-bit operating system, ibischk6_* and executable model files are done consistent with the platform operating system
  - Ibischk6_32 will work on 64-bit platforms, but will load and test 32 bit executable model files
- Specification does not impose a requirement, but internal Windows requires executable names with at least a dot “.”
- Message will give the recommended (but not required) extension

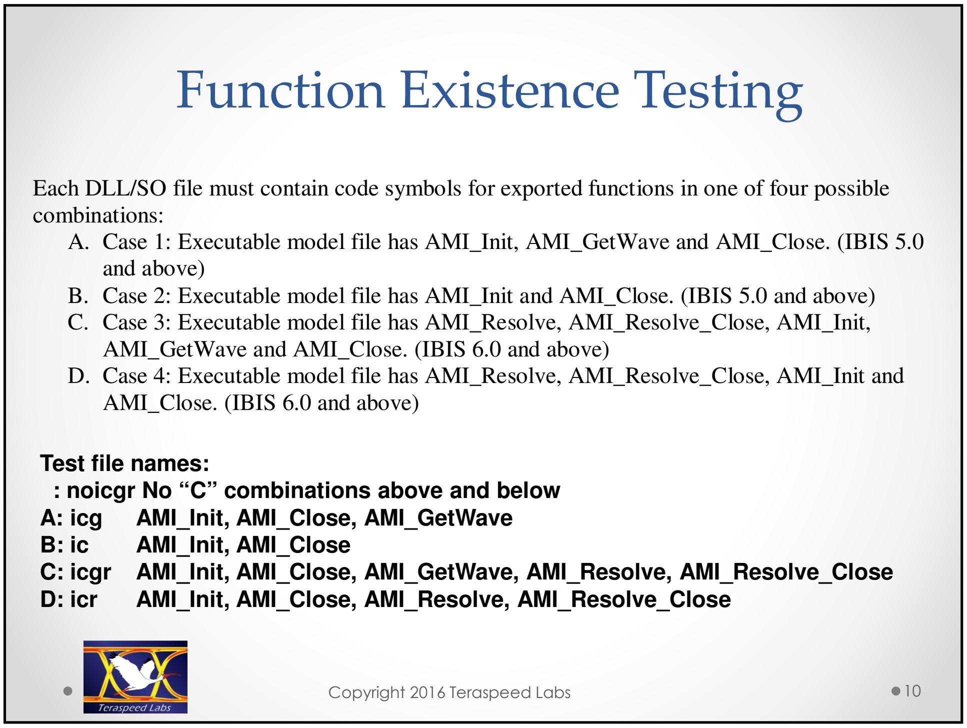
Copyright 2016 Teraspeed Labs



bug179-1.ami
Full Function Existence Checking

```
(bug179
   (Reserved_Parameters
        (AMI_Version (Usage Info)(Type String)(Value "6.1"))
       (Init_Returns_Impulse (Usage Info)(Type Boolean)(Value True))
       (GetWave_Exists (Usage Info)(Type Boolean)(Value True))
       (Resolve_Exists (Usage Info)(Type Boolean)(Value True))
   )
)
```

Copyright 2016 Teraspeed Labs



Function Existence Testing

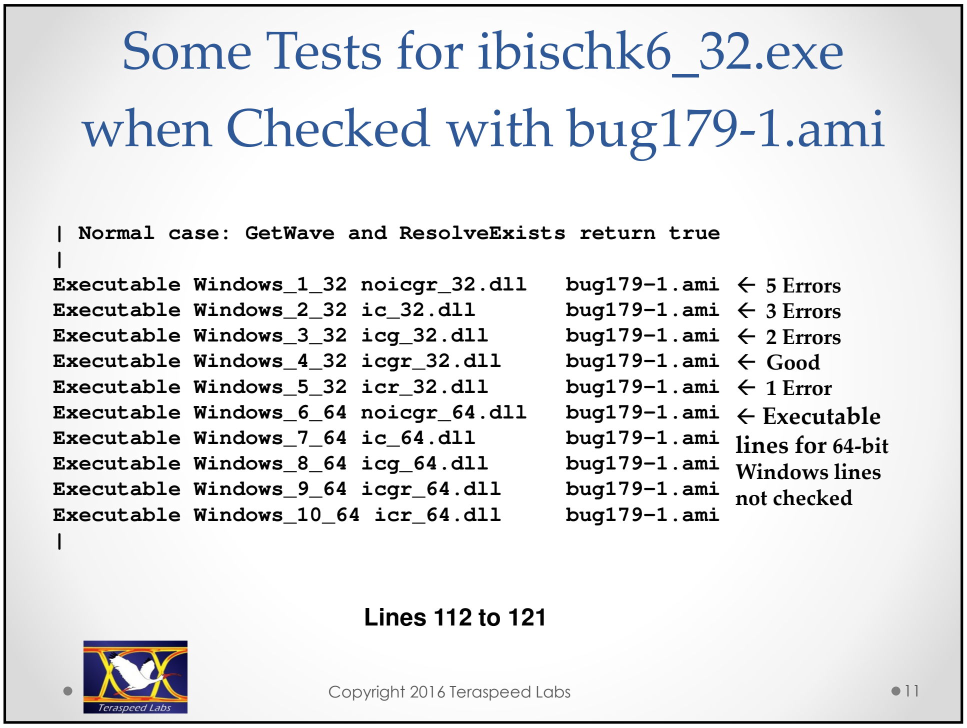
Each DLL/SO file must contain code symbols for exported functions in one of four possible combinations:

A. Case 1: Executable model file has AMI_Init, AMI_GetWave and AMI_Close. (IBIS 5.0 and above)
B. Case 2: Executable model file has AMI_Init and AMI_Close. (IBIS 5.0 and above)
C. Case 3: Executable model file has AMI_Resolve, AMI_Resolve_Close, AMI_Init, AMI_GetWave and AMI_Close. (IBIS 6.0 and above)
D. Case 4: Executable model file has AMI_Resolve, AMI_Resolve_Close, AMI_Init and AMI_Close. (IBIS 6.0 and above)

Test file names:

	: noicgr	**No "C" combinations above and below**
A:	**icg**	**AMI_Init, AMI_Close, AMI_GetWave**
B:	**ic**	**AMI_Init, AMI_Close**
C:	**icgr**	**AMI_Init, AMI_Close, AMI_GetWave, AMI_Resolve, AMI_Resolve_Close**
D:	**icr**	**AMI_Init, AMI_Close, AMI_Resolve, AMI_Resolve_Close**

Copyright 2016 Teraspeed Labs

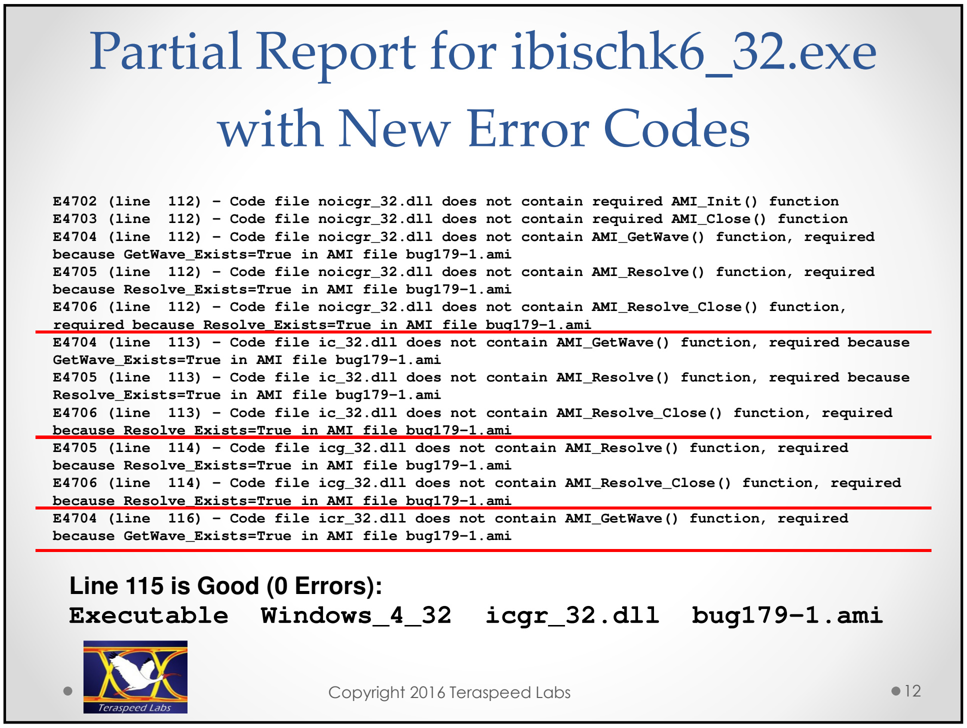


Some Tests for ibischk6_32.exe when Checked with bug179-1.ami

```
| Normal case: GetWave and ResolveExists return true
|
Executable Windows_1_32 noicgr_32.dll   bug179-1.ami  ← 5 Errors
Executable Windows_2_32 ic_32.dll       bug179-1.ami  ← 3 Errors
Executable Windows_3_32 icg_32.dll      bug179-1.ami  ← 2 Errors
Executable Windows_4_32 icgr_32.dll     bug179-1.ami  ← Good
Executable Windows_5_32 icr_32.dll      bug179-1.ami  ← 1 Error
Executable Windows_6_64 noicgr_64.dll   bug179-1.ami  ← Executable
Executable Windows_7_64 ic_64.dll       bug179-1.ami  lines for 64-bit
Executable Windows_8_64 icg_64.dll      bug179-1.ami  Windows lines
Executable Windows_9_64 icgr_64.dll     bug179-1.ami  not checked
Executable Windows_10_64 icr_64.dll     bug179-1.ami
|
```

Lines 112 to 121

Copyright 2016 Teraspeed Labs



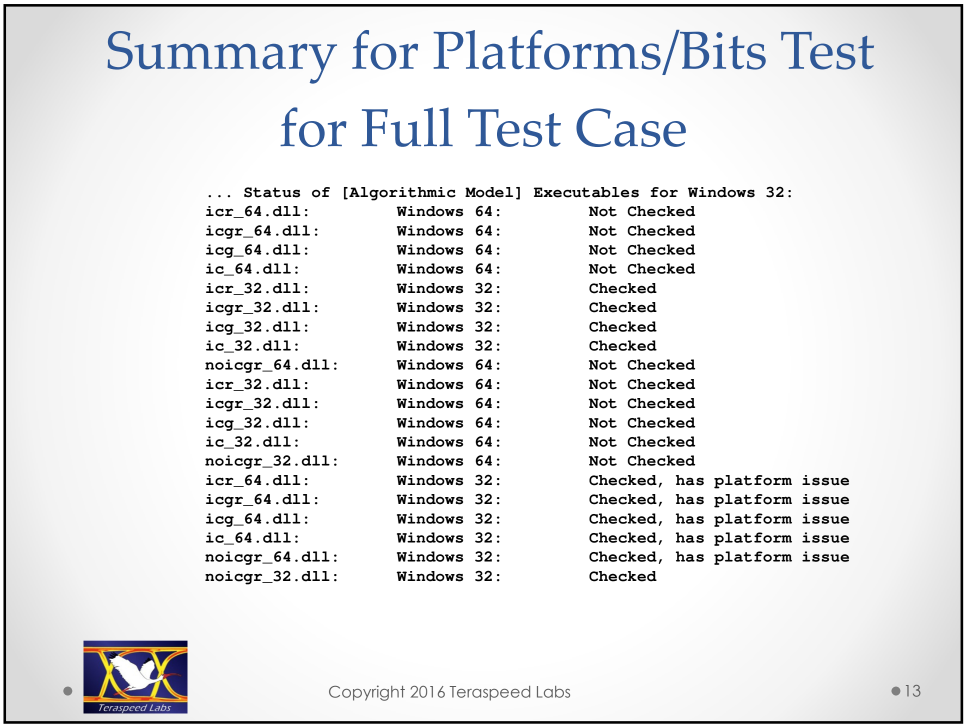
Partial Report for ibischk6_32.exe with New Error Codes

```
E4702 (line  112) - Code file noicgr_32.dll does not contain required AMI_Init() function
E4703 (line  112) - Code file noicgr_32.dll does not contain required AMI_Close() function
E4704 (line  112) - Code file noicgr_32.dll does not contain AMI_GetWave() function, required
because GetWave_Exists=True in AMI file bug179-1.ami
E4705 (line  112) - Code file noicgr_32.dll does not contain AMI_Resolve() function, required
because Resolve_Exists=True in AMI file bug179-1.ami
E4706 (line  112) - Code file noicgr_32.dll does not contain AMI_Resolve_Close() function,
required because Resolve_Exists=True in AMI file bug179-1.ami
E4704 (line  113) - Code file ic_32.dll does not contain AMI_GetWave() function, required because
GetWave_Exists=True in AMI file bug179-1.ami
E4705 (line  113) - Code file ic_32.dll does not contain AMI_Resolve() function, required because
Resolve_Exists=True in AMI file bug179-1.ami
E4706 (line  113) - Code file ic_32.dll does not contain AMI_Resolve_Close() function, required
because Resolve_Exists=True in AMI file bug179-1.ami
E4705 (line  114) - Code file icg_32.dll does not contain AMI_Resolve() function, required
because Resolve_Exists=True in AMI file bug179-1.ami
E4706 (line  114) - Code file icg_32.dll does not contain AMI_Resolve_Close() function, required
because Resolve_Exists=True in AMI file bug179-1.ami
E4704 (line  116) - Code file icr_32.dll does not contain AMI_GetWave() function, required
because GetWave_Exists=True in AMI file bug179-1.ami
```

Line 115 is Good (0 Errors):

```
Executable  Windows_4_32  icgr_32.dll  bug179-1.ami
```

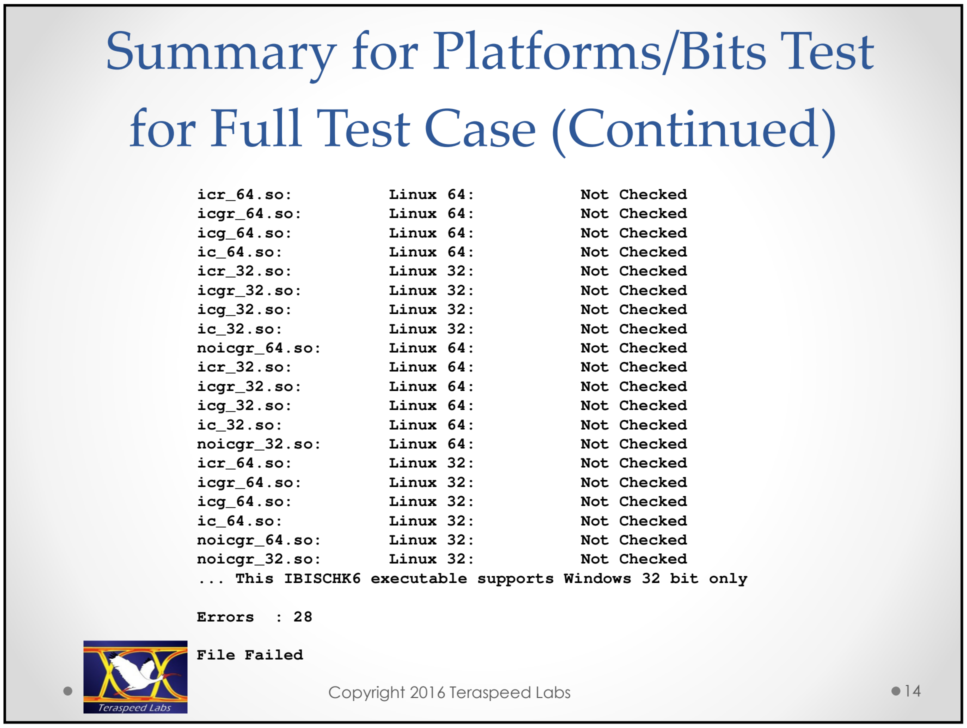
Copyright 2016 Teraspeed Labs



Summary for Platforms/Bits Test for Full Test Case

```
... Status of [Algorithmic Model] Executables for Windows 32:
icr_64.dll:         Windows 64:         Not Checked
icgr_64.dll:        Windows 64:         Not Checked
icg_64.dll:         Windows 64:         Not Checked
ic_64.dll:          Windows 64:         Not Checked
icr_32.dll:         Windows 32:         Checked
icgr_32.dll:        Windows 32:         Checked
icg_32.dll:         Windows 32:         Checked
ic_32.dll:          Windows 32:         Checked
noicgr_64.dll:      Windows 64:         Not Checked
icr_32.dll:         Windows 64:         Not Checked
icgr_32.dll:        Windows 64:         Not Checked
icg_32.dll:         Windows 64:         Not Checked
ic_32.dll:          Windows 64:         Not Checked
noicgr_32.dll:      Windows 64:         Not Checked
icr_64.dll:         Windows 32:         Checked, has platform issue
icgr_64.dll:        Windows 32:         Checked, has platform issue
icg_64.dll:         Windows 32:         Checked, has platform issue
ic_64.dll:          Windows 32:         Checked, has platform issue
noicgr_64.dll:      Windows 32:         Checked, has platform issue
noicgr_32.dll:      Windows 32:         Checked
```

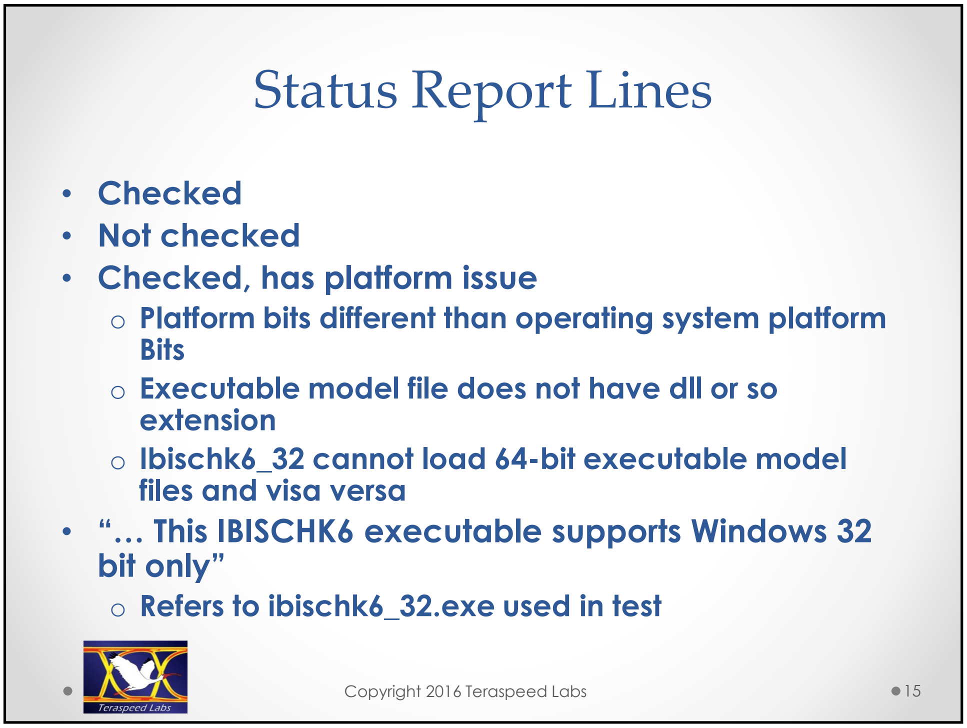
Copyright 2016 Teraspeed Labs



Summary for Platforms/Bits Test for Full Test Case (Continued)

```
icr_64.so:          Linux 64:           Not Checked
icgr_64.so:         Linux 64:           Not Checked
icg_64.so:          Linux 64:           Not Checked
ic_64.so:           Linux 64:           Not Checked
icr_32.so:          Linux 32:           Not Checked
icgr_32.so:         Linux 32:           Not Checked
icg_32.so:          Linux 32:           Not Checked
ic_32.so:           Linux 32:           Not Checked
noicgr_64.so:       Linux 64:           Not Checked
icr_32.so:          Linux 64:           Not Checked
icgr_32.so:         Linux 64:           Not Checked
icg_32.so:          Linux 64:           Not Checked
ic_32.so:           Linux 64:           Not Checked
noicgr_32.so:       Linux 64:           Not Checked
icr_64.so:          Linux 32:           Not Checked
icgr_64.so:         Linux 32:           Not Checked
icg_64.so:          Linux 32:           Not Checked
ic_64.so:           Linux 32:           Not Checked
noicgr_64.so:       Linux 32:           Not Checked
noicgr_32.so:       Linux 32:           Not Checked
... This IBISCHK6 executable supports Windows 32 bit only

Errors  : 28

File Failed
```

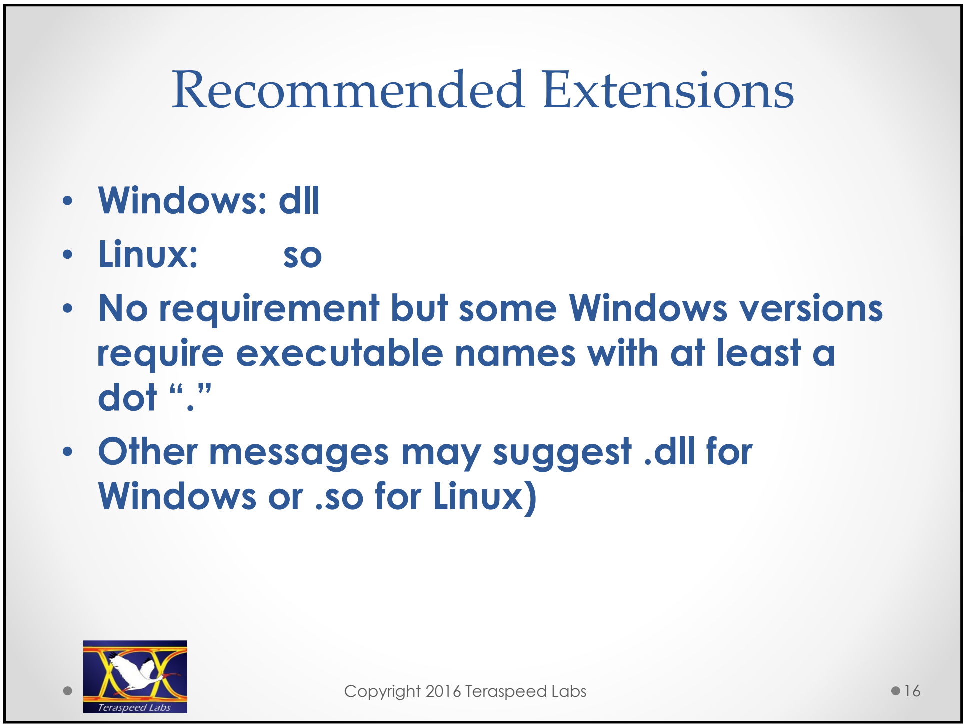
Copyright 2016 Teraspeed Labs



Status Report Lines

- Checked
- Not checked
- Checked, has platform issue
  - Platform bits different than operating system platform Bits
  - Executable model file does not have dll or so extension
  - Ibischk6_32 cannot load 64-bit executable model files and visa versa
- "… This IBISCHK6 executable supports Windows 32 bit only"
  - Refers to ibischk6_32.exe used in test

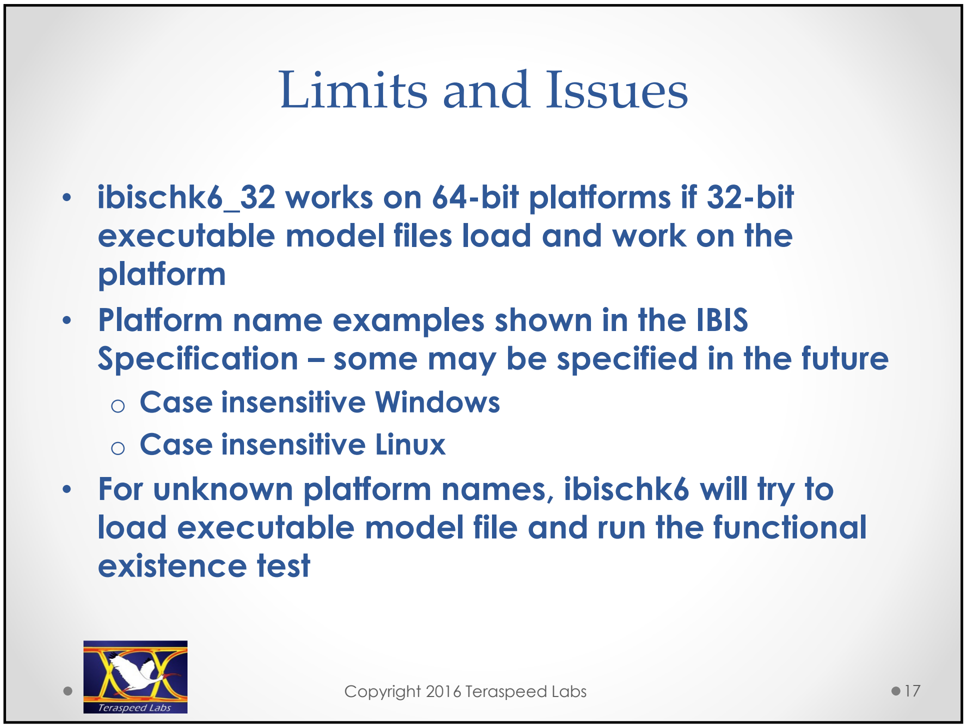
Copyright 2016 Teraspeed Labs



Recommended Extensions

- Windows: dll
- Linux: so
- No requirement but some Windows versions require executable names with at least a dot "."
- Other messages may suggest .dll for Windows or .so for Linux)

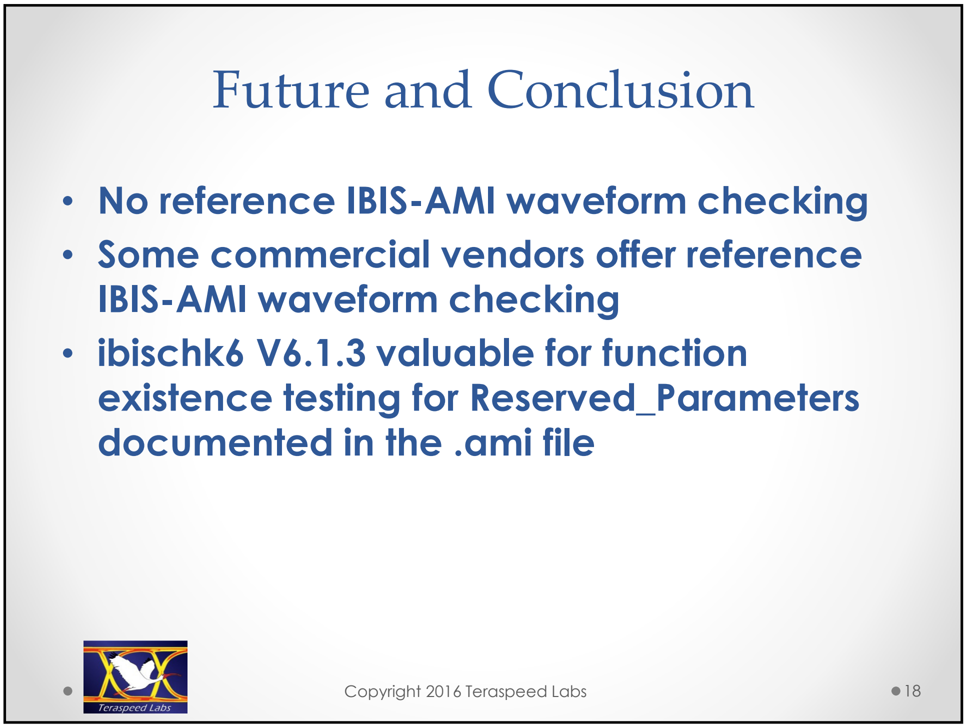
Copyright 2016 Teraspeed Labs



Limits and Issues

- ibischk6_32 works on 64-bit platforms if 32-bit executable model files load and work on the platform
- Platform name examples shown in the IBIS Specification – some may be specified in the future
  - Case insensitive Windows
  - Case insensitive Linux
- For unknown platform names, ibischk6 will try to load executable model file and run the functional existence test

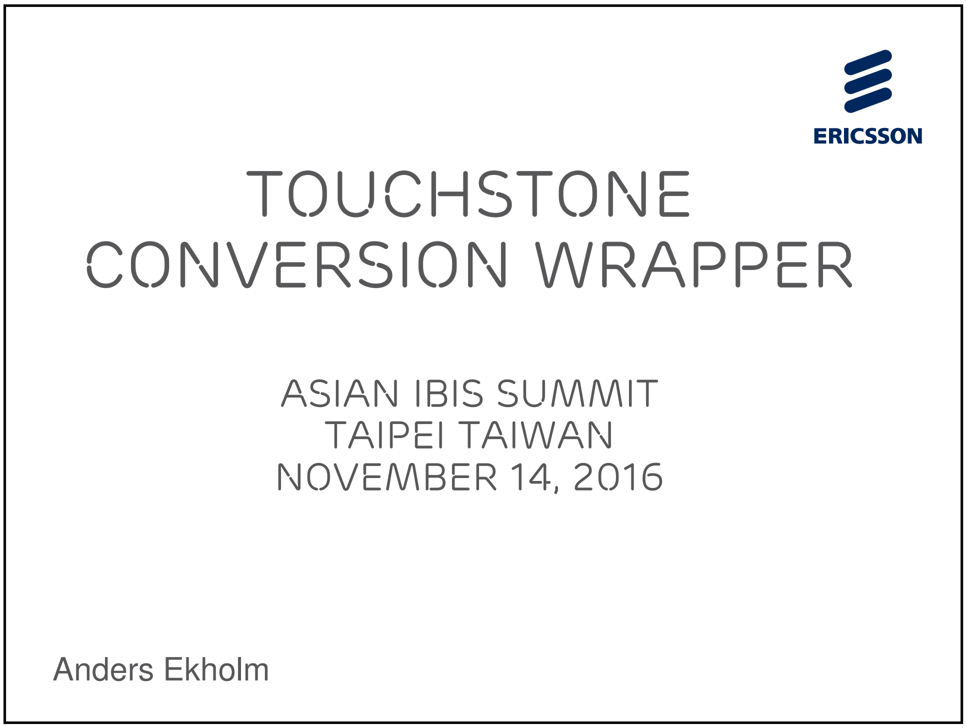
Copyright 2016 Teraspeed Labs



Future and Conclusion

- No reference IBIS-AMI waveform checking
- Some commercial vendors offer reference IBIS-AMI waveform checking
- ibischk6 V6.1.3 valuable for function existence testing for Reserved_Parameters documented in the .ami file

Copyright 2016 Teraspeed Labs

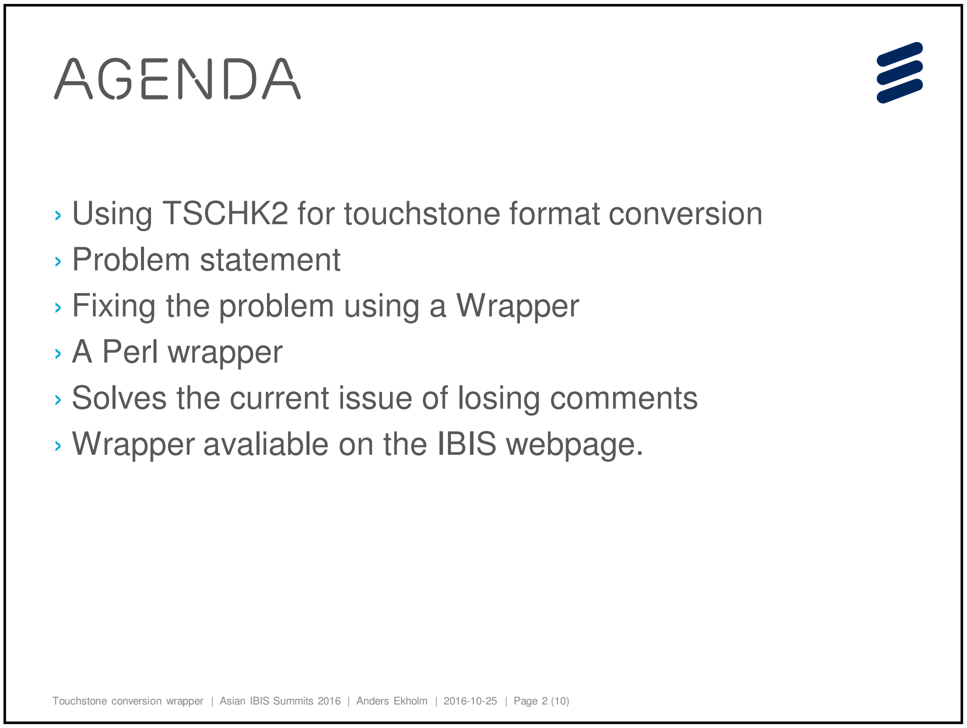


ERICSSON

TOUCHSTONE CONVERSION WRAPPER

ASIAN IBIS SUMMIT
TAIPEI TAIWAN
NOVEMBER 14, 2016

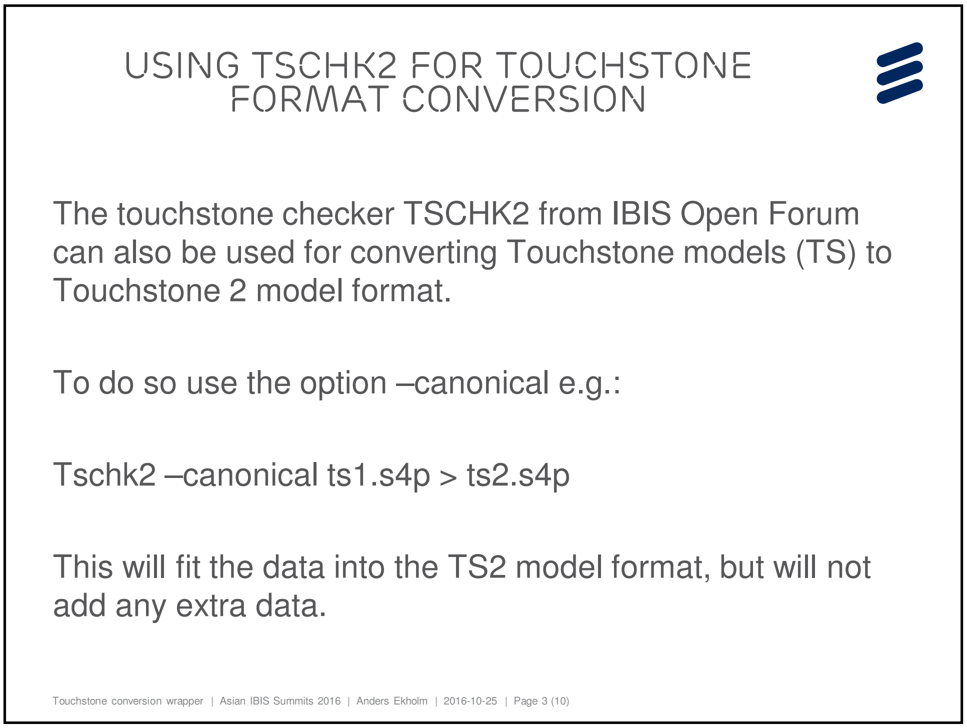
Anders Ekholm



AGENDA

- › Using TSCHK2 for touchstone format conversion
- › Problem statement
- › Fixing the problem using a Wrapper
- › A Perl wrapper
- › Solves the current issue of losing comments
- › Wrapper avaliable on the IBIS webpage.

Touchstone conversion wrapper | Asian IBIS Summits 2016 | Anders Ekholm | 2016-10-25 | Page 2 (10)



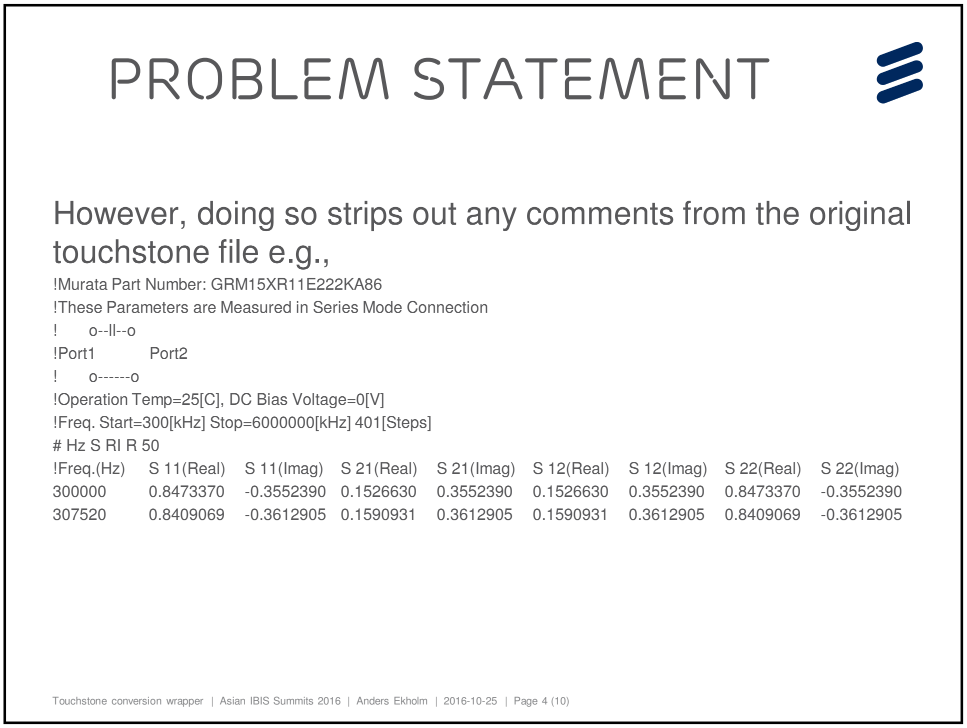
USING TSCHK2 FOR TOUCHSTONE FORMAT CONVERSION

The touchstone checker TSCHK2 from IBIS Open Forum can also be used for converting Touchstone models (TS) to Touchstone 2 model format.

To do so use the option –canonical e.g.:

Tschk2 –canonical ts1.s4p > ts2.s4p

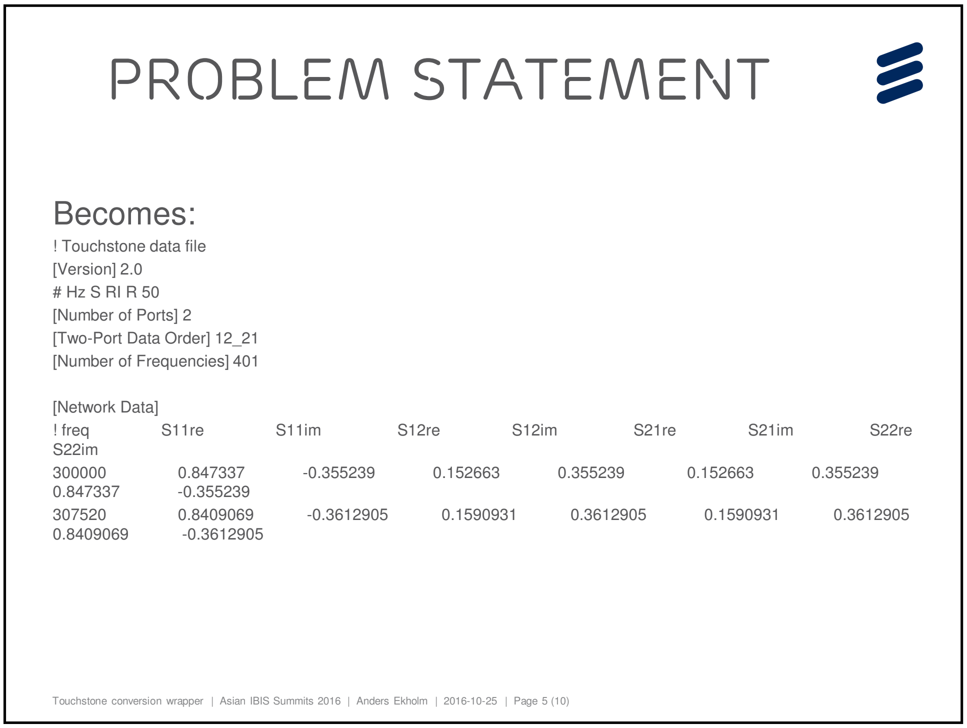
This will fit the data into the TS2 model format, but will not add any extra data.



PROBLEM STATEMENT

However, doing so strips out any comments from the original touchstone file e.g.,

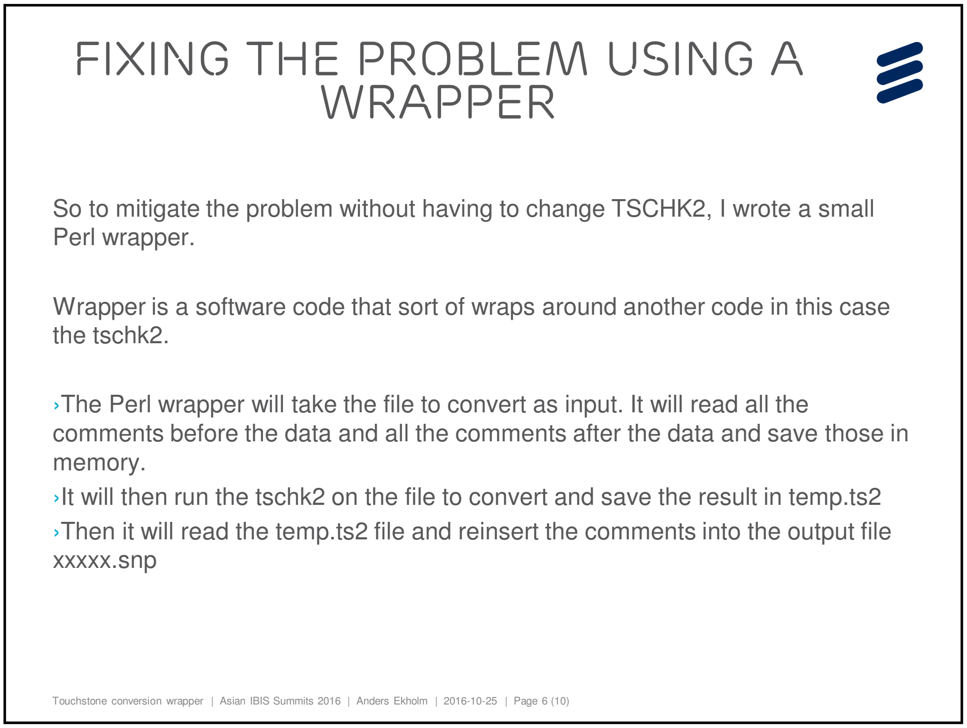
```
!Murata Part Number: GRM15XR11E222KA86
!These Parameters are Measured in Series Mode Connection
!       o--II--o
!Port1           Port2
!       o------o
!Operation Temp=25[C], DC Bias Voltage=0[V]
!Freq. Start=300[kHz] Stop=6000000[kHz] 401[Steps]
# Hz S RI R 50
!Freq.(Hz)   S 11(Real)   S 11(Imag)   S 21(Real)   S 21(Imag)   S 12(Real)   S 12(Imag)   S 22(Real)   S 22(Imag)
300000       0.8473370    -0.3552390   0.1526630    0.3552390    0.1526630    0.3552390    0.8473370    -0.3552390
307520       0.8409069    -0.3612905   0.1590931    0.3612905    0.1590931    0.3612905    0.8409069    -0.3612905
```



PROBLEM STATEMENT

Becomes:

```
! Touchstone data file
[Version] 2.0
# Hz S RI R 50
[Number of Ports] 2
[Two-Port Data Order] 12_21
[Number of Frequencies] 401

[Network Data]
! freq          S11re           S11im           S12re           S12im           S21re           S21im           S22re
S22im
300000          0.847337        -0.355239       0.152663        0.355239        0.152663        0.355239
0.847337        -0.355239
307520          0.8409069       -0.3612905      0.1590931       0.3612905       0.1590931       0.3612905
0.8409069       -0.3612905
```

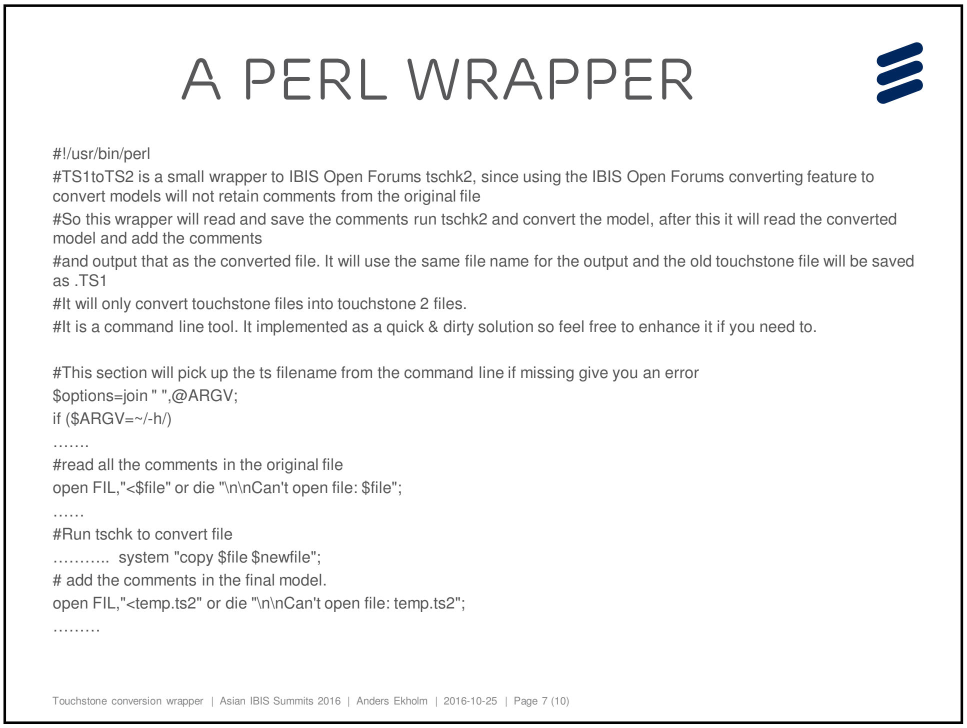


FIXING THE PROBLEM USING A WRAPPER

So to mitigate the problem without having to change TSCHK2, I wrote a small Perl wrapper.

Wrapper is a software code that sort of wraps around another code in this case the tschk2.

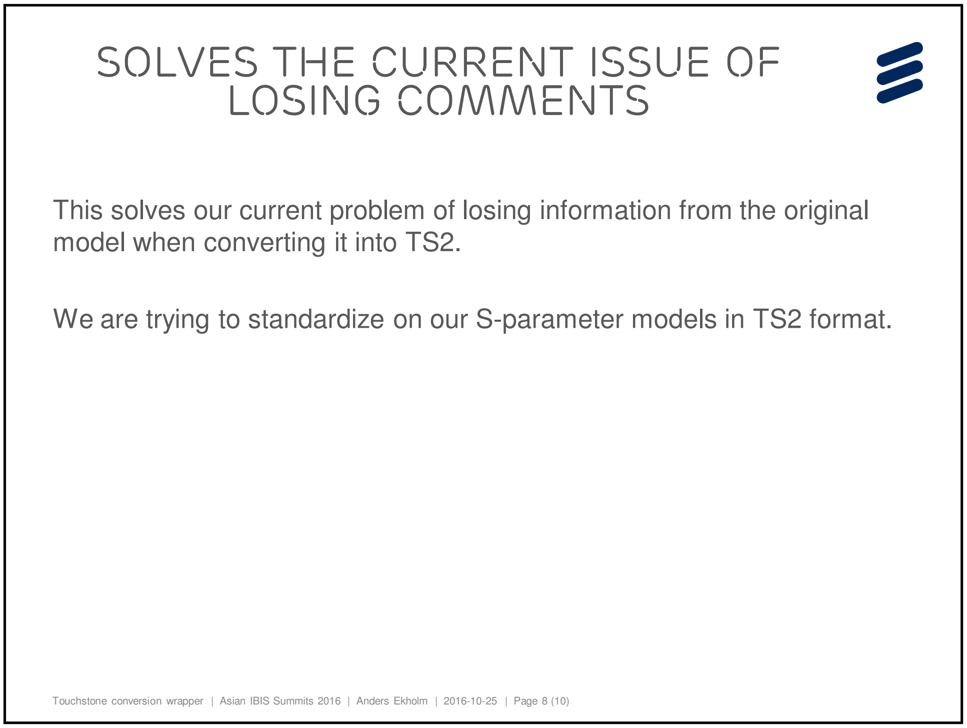
- The Perl wrapper will take the file to convert as input. It will read all the comments before the data and all the comments after the data and save those in memory.
- It will then run the tschk2 on the file to convert and save the result in temp.ts2
- Then it will read the temp.ts2 file and reinsert the comments into the output file xxxxx.snp



A PERL WRAPPER

```
#!/usr/bin/perl
#TS1toTS2 is a small wrapper to IBIS Open Forums tschk2, since using the IBIS Open Forums converting feature to convert models will not retain comments from the original file
#So this wrapper will read and save the comments run tschk2 and convert the model, after this it will read the converted model and add the comments
#and output that as the converted file. It will use the same file name for the output and the old touchstone file will be saved as .TS1
#It will only convert touchstone files into touchstone 2 files.
#It is a command line tool. It implemented as a quick & dirty solution so feel free to enhance it if you need to.

#This section will pick up the ts filename from the command line if missing give you an error
$options=join " ",@ARGV;
if ($ARGV=~/-h/)
…….
#read all the comments in the original file
open FIL,"<$file" or die "\n\nCan't open file: $file";
……
#Run tschk to convert file
………..  system "copy $file $newfile";
# add the comments in the final model.
open FIL,"<temp.ts2" or die "\n\nCan't open file: temp.ts2";
………
```

Touchstone conversion wrapper | Asian IBIS Summits 2016 | Anders Ekholm | 2016-10-25 | Page 7 (10)

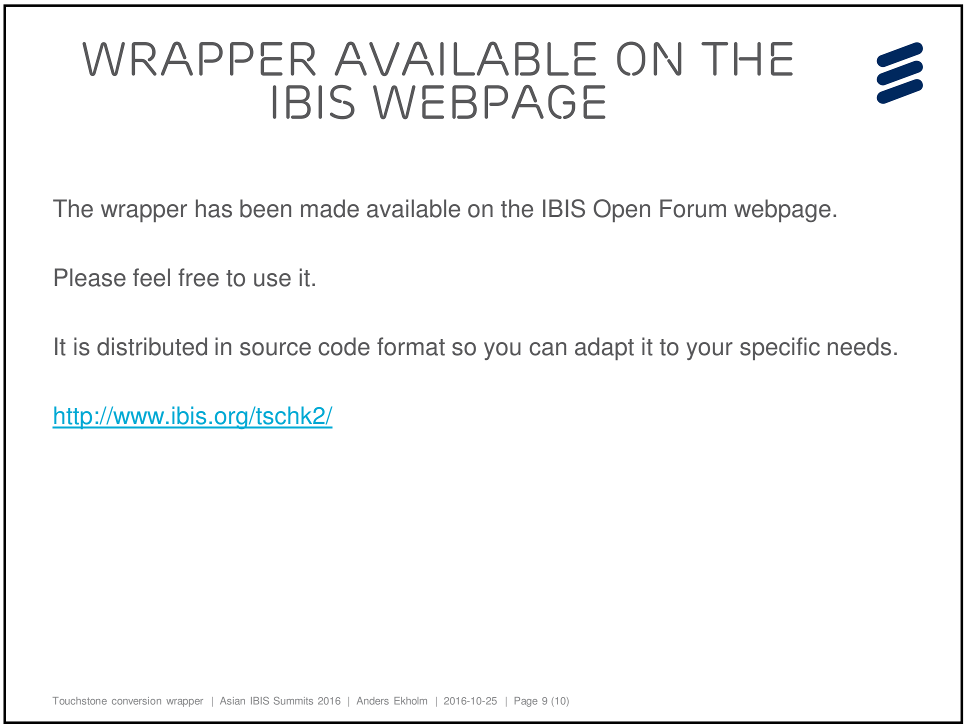


SOLVES THE CURRENT ISSUE OF LOSING COMMENTS

This solves our current problem of losing information from the original model when converting it into TS2.

We are trying to standardize on our S-parameter models in TS2 format.

Touchstone conversion wrapper | Asian IBIS Summits 2016 | Anders Ekholm | 2016-10-25 | Page 8 (10)



WRAPPER AVAILABLE ON THE IBIS WEBPAGE

The wrapper has been made available on the IBIS Open Forum webpage.

Please feel free to use it.

It is distributed in source code format so you can adapt it to your specific needs.

http://www.ibis.org/tschk2/

ERICSSON