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 2017 Asian IBIS Summit in Taipei and to thank you for your presentations and participation. We are grateful to our sponsors IO Methodology, Peace Giant (SPISim), and Synopsys 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. With the introduction of IBIS-AMI in 2008, the IBIS community generated new energy for high speed electronic design. IBIS is now known by engineers worldwide and is a required technology for many applications.

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

Mike LaBonte
SiSoft
Chair, IBIS Open Forum

WELCOME FROM MIKE LABONTE, IBIS OPEN FORUM

女士們先生們，

作為 IBIS 開放論壇的主席，我很高興地歡迎您參加 2017 年在台北舉辦的亞洲 IBIS 峰會，感謝您的介紹和參與。我們非常感謝我們的讚助商 IO Methodology，Peace Giant（SPISim）和 Synopsys，以使這一事件成為可能。

自 1993 年以來，IBIS 為數字電子行業提供了使信號，時序和電源完整性分析更容易和更快速的規範。隨著 IBIS-AMI 在 2008 年的推出，IBIS 社區為高速電子設計創造了新的能量。 IBIS 現在已被世界各地的工程師所了解，是許多應用所需的技術。

IBIS 在亞洲的支持一直很強，IBIS 開放論壇期待著亞洲技術公司的不斷創新和貢獻。

谢谢！

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)

       I B I S   S U M M I T   M E E T I N G   A G E N D A

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

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

9:10   **IBIS Update** **5**
          Mike LaBonte (SiSoft, USA)

9:45   **IBIS Interconnect Modeling Using IBIS-ISS and Touchstone** **14**
          Michael Mirmak (Intel Corporation, USA)
          [Presented by Mike LaBonte (SiSoft, USA)]

10:20   **Leveraging IBIS Capabilities for Multi-Gigabit Interfaces** **27**
          Ken Willis (Cadence Design Systems, USA)
          [Presented by Skipper Liang (Cadence Design Systems, ROC)]

10:50   **BREAK** (Refreshments and Vendor Tables)

11:10   **Comparison of Time Domain and Statistical IBIS-AMI Analyses** **39**
          Mike LaBonte (SiSoft, USA)

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

AGENDA AND ORDER OF THE PRESENTATIONS (Continued)

15:10 **BREAK** (Refreshments and Vendor Tables)

15:30 **DISCUSSION**

16:20 **CONCLUDING ITEMS**

16:30 **END OF IBIS SUMMIT MEETING**

IBIS Update

Mike LaBonte
SiSoft
Chair, IBIS Open Forum

2017 Asian IBIS Summit
Taipei, ROC
November 15, 2017

http://www.ibis.org/

IBIS Update   1
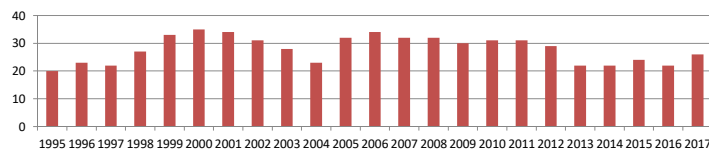

Organization

26 IBIS Members

Number of Members by Year

IBIS Update   2

Organization

IBIS Officers 2017-2018

Chair: *Mike LaBonte, SiSoft*

Vice-Chair: *Lance Wang, IO Methodology Inc.*

Secretary: *Randy Wolff, Micron Technology*

Treasurer: *Bob Ross, Teraspeed Labs*

Librarian: *Anders Ekholm, Ericsson*

Postmaster: *Curtis Clark, ANSYS*

Webmaster: *Mike LaBonte, SiSoft*

*2018 Officer Election nominations open May 17*

IBIS Update   3

Organization

IBIS Meetings

- Weekly teleconferences
  - Quality Task Group (Tuesdays)
  - Advanced Technology Modeling Task Group (Tuesdays)
  - Interconnect Task Group (Wednesdays)
  - Editorial Task Group (some Fridays)
- IBIS Open Forum teleconference every 3 weeks
  - 486 meetings so far
- IBIS Summit meetings: DesignCon, IEEE SPI, EDICON USA, EPEPS, Shanghai, Taipei, Tokyo
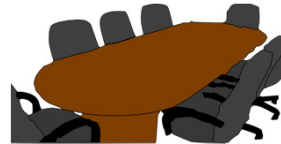
IBIS Update   4

Organization

SAE ITC

- SAE Industry Technologies Consortia is the parent organization of the IBIS Open Forum
- IBIS is assisted by SAE employees Thomas Munns, Phyllis Gross, Dorothy Lloyd
- SAE ITC provides financial, legal, and other services
- http://itc.sae.org/

IBIS Update

5

Organization

Task Groups

- Interconnect Task Group
  - Chair: Michael Mirmak
  - http://ibis.org/interconn_wip/
  - Develop on-die/package/module/connector interconnect modeling BIRDs
- Advanced Technology Modeling Task Group
  - Chair: Arpad Muranyi
  - http://ibis.org/atm_wip/
  - Develop most other technical BIRDs
- Quality Task Group
  - Chair: Mike LaBonte
  - http://ibis.org/quality_wip/
  - Oversee IBISCHK parser testing and development
- Editorial Task Group
  - Chair: Michael Mirmak
  - http://ibis.org/editorial_wip/
  - Produce IBIS Specification documents

IBIS Update
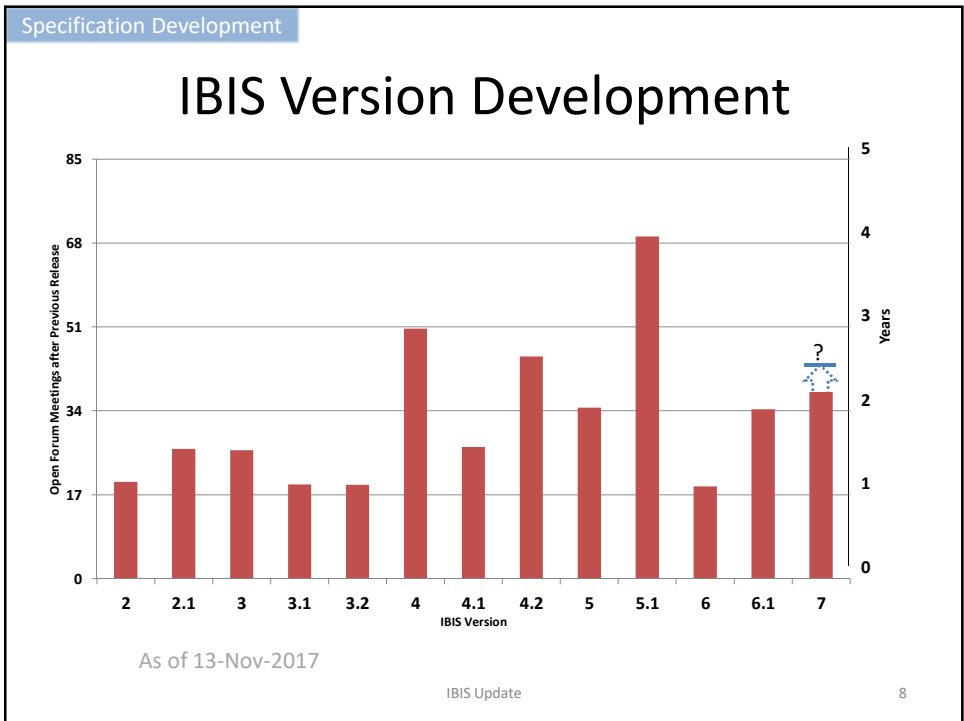
6

IBIS Milestones

I/O Buffer Information Specification

- 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
- **2018? IBIS 7.0** *Current development*

Other Work

- 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

IBIS Update 7

IBIS Version Development

As of 13-Nov-2017

IBIS Update 8

Possible IBIS 7.0 Timeline

Meeting Date	Milestone
4/21/2017	Vote to establish 7.0 as the next IBIS version passes
5/12/2017	*BIRD review and acceptance (10 meetings)*
...	...
2/16/2018	Vote to approve 7.0 BIRD set is scheduled for next meeting
3/9/2018	7.0 BIRD set accepted. Editorial work begins
3/30/2018	
4/20/2018	
5/11/2018	Editorial announces 7.0 ready. Review period begins
6/1/2018	
6/22/2018	Vote to ratify 7.0 scheduled for next meeting
7/13/2018	7.0 ratified

New!
IBIS 7.0

BIRD = Buffer Issue Resolution Document

IBIS Update   9

Specification Development

BIRDs Possibly Included in IBIS 7.0

BIRD	Title
147.6	Back-channel Support
158.7	AMI Ts4file Analog Buffer Models
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
184.2	Model_name and Signal_name Restriction for POWER and GND Pins
185.2	Section 3 Reserved Word Guideline Update
186.4	File Naming Rules
187.3	Format and Usage Out Clarifications
188.1	Expanded Rx Noise Support for AMI
189.4	Interconnect Modeling Using IBIS-ISS and Touchstone
191.2	Clarifying Locations for Si_location and Timing_location
192.1	Clarification of List Default Rules

Green = currently accepted BIRD

IBIS Update   10

BIRDs Possibly Excluded from IBIS 7.0

BIRD	Title
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
166.2	Resolving problems with Redriver Init Flow
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
181.1	I-V Table Clarifications
190	Clarification for Redriver Flow

White = currently not an accepted BIRD

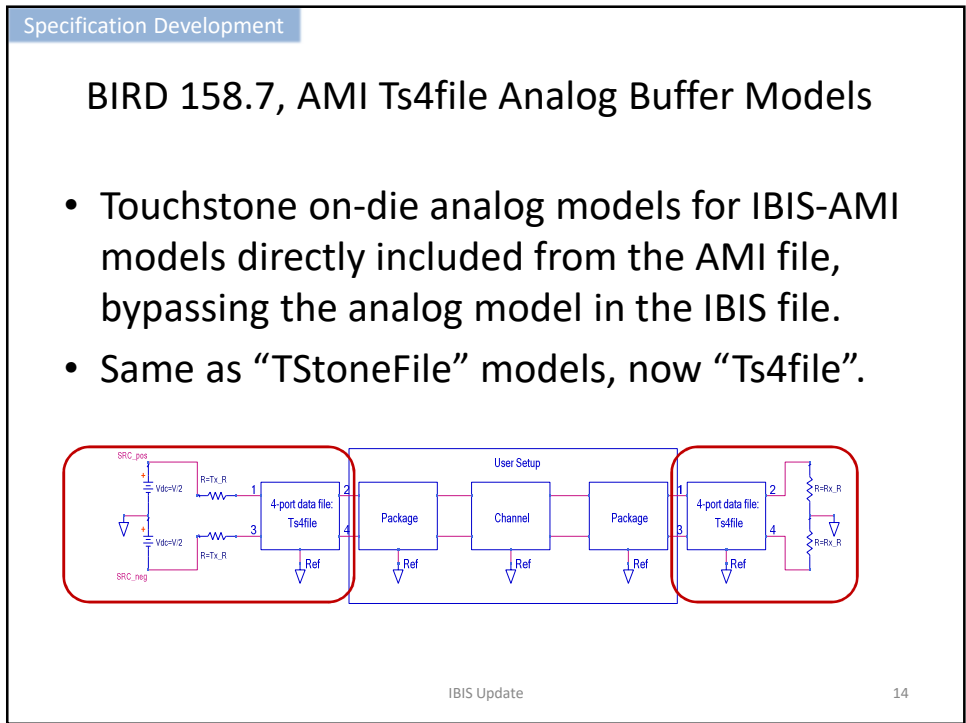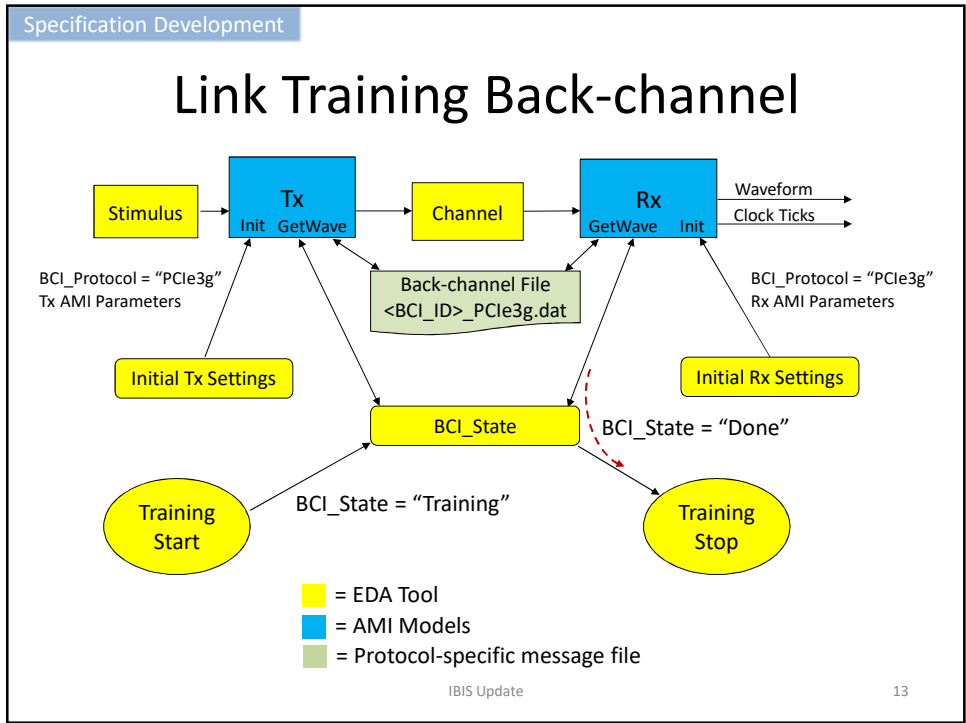IBIS Update   11

BIRD 147.6, Back-channel Support

- Enable back-channel link training messages between the Tx and Rx executable models to enable link training to optimize equalization settings during time domain (AMI_GetWave) simulations.

- New AMI Parameters:
  - BCI_Protocol, BCI_State, BCI_ID, BCI_Message_Interval_UI, BCI_Training_UI

IBIS Update   12

Link Training Back-channel

Specification Development

BIRD 158.7, AMI Ts4file Analog Buffer Models

- Touchstone on-die analog models for IBIS-AMI models directly included from the AMI file, bypassing the analog model in the IBIS file.
- Same as "TStoneFile" models, now "Ts4file".

Specification Development

BIRD 188.1, Expanded Rx Noise Support for AMI

- Bounded (uniform) Rx Noise must be supported by IBIS-AMI, separately from the existing Gaussian random Rx Noise parameter.

Parameter:	**Rx_Noise, Rx_GaussianNoise**
Required:	No, and Rx_Noise is illegal before AMI_Version 6.0;
	No, and Rx_GaussianNoise is illegal before AMI_Version 7.0

Parameter:	**Rx_UniformNoise**
Required:	No, and illegal before AMI_Version 7.0

IBIS Update   15

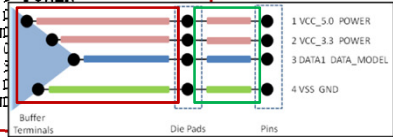Specification Development

BIRD 189.4, Interconnect Modeling Using IBIS-ISS and Touchstone

```
[Interconnect Model Set]        Full_ISS_buf_pad_pin_PDN_4

[Interconnect Model]            Full_ISS_pad_pin_IO
File_IBIS-ISS    full_pad_pin_io.iss          full_pad_pin_IO_typ
Number_of_terminals = 8
1  Pin_Rail     pin_name      1     |  VCC_5.0 POWER
2  Pin_Rail     pin_name      2     |  VCC_3.3 POWER
3  Pin_I/O      pin_name      3     |  DATA1 DATA_MODEL
4  Pin_Rail     pin_name      4     |  VSS GND
5  Pad_Rail     pad_name      VCC1  |  VCC_5.0 POWER
6  Pad_Rail     pad_name      VCC2  |  VCC_3.3 POWER
7  Pad_I/O      pin_name      3     |  DATA1 DATA_MODEL
8  Pad_Rail     pad_name      VSS1  |  VSS GND
[End Interconnect Model]

[Interconnect Model]            Full_ISS_buf_pad_IO
File_TS          full_buf_pad_io.s8p |       full_buf_pad_IO_typ
Number_of_terminals = 8
1  Pad_Rail     pad_name      VCC1  |  VCC_5.0 POWER
2  Pad_Rail     pad_name      VCC2  |  VCC_3.3 POWER
3  Pad_I/O      pin_name      3     |  DATA1
4  Pad_Rail     pad_name      VSS1  |  VSS GND
5  Buffer_Rail  pin_name      1     |  VCC_5.
6  Buffer_Rail  pin_name      2     |  VCC_3.
7  Buffer_I/O   pin_name      3     |  DATA1
8  Buffer_Rail  pin_name      4     |  VSS GN
[End Interconnect Model]

[End Interconnect Model Set]
```


IBIS Update   16

[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.

IBIS Update   17

Interconnect Modeling Update Using IBIS-ISS and Touchstone

Michael Mirmak
Intel Corporation
michael.mirmak@intel.com

Asian IBIS Summit
Taipei, ROC
November 15, 2017

Agenda

- ❑ History
- ❑ The need for improved interconnect support
- ❑ Principles of the Interconnect Proposal
  - – Structure
  - – Terminals, Models and Sets, and Groups (new)
  - – New Keywords
- ❑ An example explained
- ❑ New or changed features
- ❑ Summary

Asian IBIS Summit, November, 2017   2

History

- Interconnect Task Group resumed meeting in early 2014
  - Received draft BIRD from Walter Katz (SiSoft) to support IBIS-ISS packages within IBIS

- BIRD189.x uploaded (awaiting approval)
  - http://www.ibis.org/birds/
  - 41 pages with examples
  - Still resolving some issues
  - Comments welcome

- Intended for IBIS Version 7.0

- Brief overview with some key points is given here

Why Update Interconnect Modeling?

- Improve package models with IBIS-ISS (an HSPICE subset) and Touchstone support

- Package modeling in IBIS stable since 2000
  - [Pin], [Package], [Package Model]
  - [Alternate Package Models] selector added
  - Limited support of loss, crosstalk and/or partitioning

- EBD (Electrical Board Description) for boards; No coupling and limited package model application

- IBIS, IBIS-ISS, Touchstone 2.0 and ICM are separate specifications
  - Limited interaction between them for package modeling
  - ICM (Interconnect Model) never adopted by industry

Features of the Interconnect Proposal

❑ Supports…

– IBIS-ISS and Touchstone models (common in industry)

– Both I/O and supply (POWER and GND) connections

– (New) optional Die pad interface between Pins and Buffers

– I/O pin_names as terminal qualifiers

▪ *May have optional Aggressor_Only designation*

– POWER and GND terminal qualifiers by pin_name, pad_name, signal_name or [Pin Mapping] bus_label for rail connections with direct or combined terminals

– Many other features not covered here

> A few objectives for the
> Interconnect Modeling proposal

Asian IBIS Summit, November, 2017   5

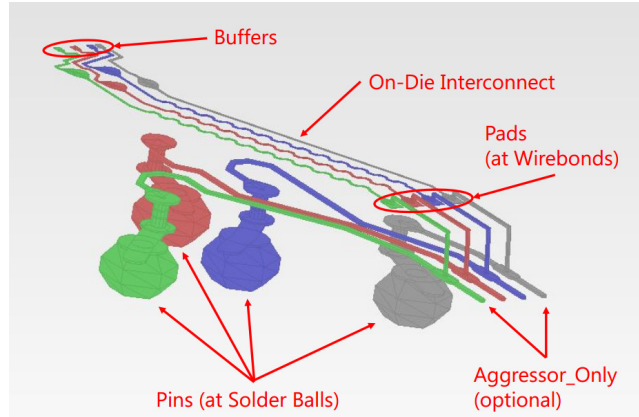# Structure of the Interconnect Proposal

> Introduces optional Die pad interface for terminals
> separate from Buffer and Pin interface terminals

Asian IBIS Summit, November, 2017   6
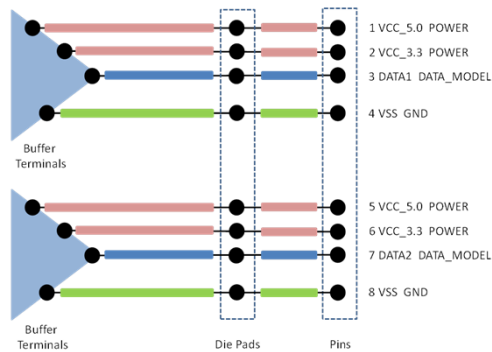
Relates to Physical Structures

One-to-one path connection; Die pad interface optional;
Aggressor_Only designation optional

Terminals at
Buffer, Die Pad and Pin Interfaces

Original IBIS (4.0 and earlier)
• Pins are explicit
• Buffer terminals implicit in [Model]
• Die pad terminals same as buffer terminals
• Packages defined connections between pins and buffers

Current Proposal
• Die pad terminals are now explicit
• Buffer terminals are now explicit
• [Pin]s are…. still pins
• Separate interconnect definitions can be created between …
  • Pin-to-Die pad terminals,
  • Die pad-to-Buffer terminals,
  • Pin-to-Buffer terminals (still) supported

Physical Rails (Can be Merged)

Die pads at Wirebonds circled in red

Pin (at Solder Ball)
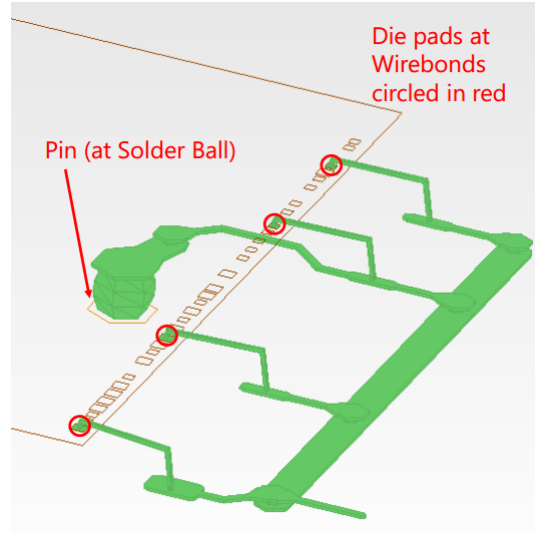
Asian IBIS Summit, November, 2017   9

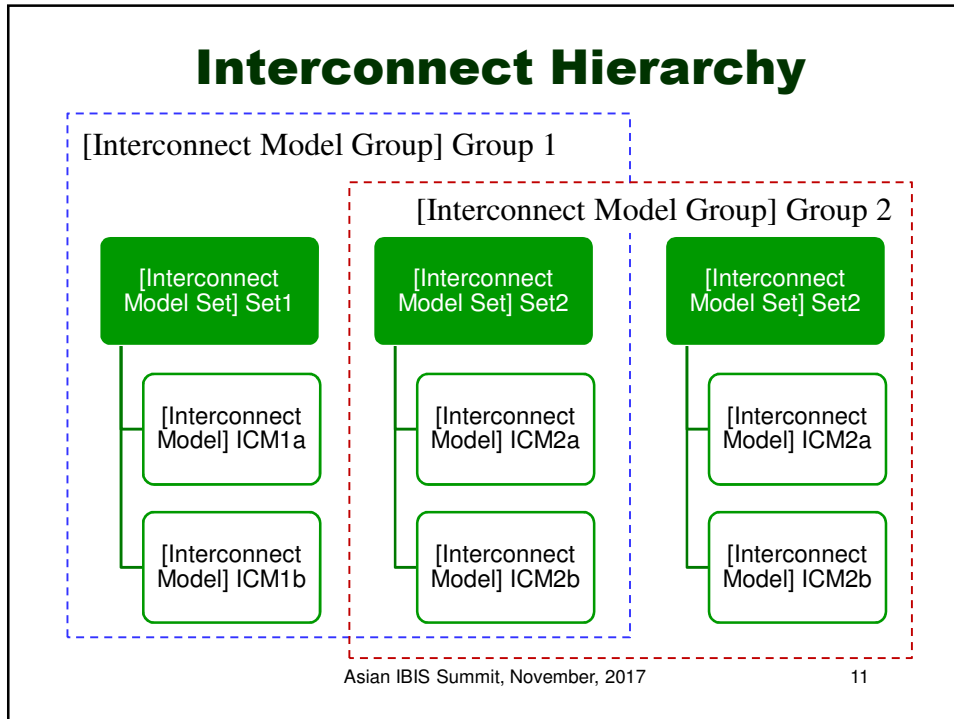
New Keywords and Subparameters (Limited Discussion Here)

- [Bus Labels] | bus_label
- [Die Supply Pads] | pad_name, optional bus_label
- [Interconnect Model]/[End Interconnect Model]
  - Unused_port_termination <Open | Ref.> | Unused port ref. Z
  - Param | parameter passing
  - File_IBIS-ISS | names IBIS-ISS file
  - File_TS, File_TS0 | names Tstone file
  - Number_of_terminals=<value> | number of terminals
  - <terminal lines> | described later
- [Interconnect Model Set]/[End Interconnect Model Set]
- [Interconnect Model Set Group]/[End Interconnect Model Set Group] (New and changed from "Selector")

Asian IBIS Summit, November, 2017   10

Interconnect Hierarchy

[Interconnect Model Group] Group 1

[Interconnect Model Group] Group 2

[Interconnect Model Set] Set1

[Interconnect Model Set] Set2

[Interconnect Model Set] Set2

[Interconnect Model] ICM1a

[Interconnect Model] ICM2a

[Interconnect Model] ICM2a

[Interconnect Model] ICM1b

[Interconnect Model] ICM2b

[Interconnect Model] ICM2b

Asian IBIS Summit, November, 2017   11

[Interconnect Model Set]s
[Interconnect Model Set Group]s (New)

❑ [Interconnect Model Set] <set_name>
  – Encapsulates one or more Interconnect Models

❑ [Interconnect Model Set Group] <group_name>
  – References one or more Interconnect Model Sets to be used together
  – Should be used to establish a complete path for selected buffers
  – <group_name> helps in identify the buffers that are selected for simulation

❑ Some Example Groupings and Applications
  – Separate groups: one per interface (e.g., memory, network)
  – Separate groups for coupled vs. single-line simulations
  – Different sets for different power delivery network complexities
    ▪ *POWER connected at single pin, single buffer terminal*
    ▪ *POWER connected through multiple pins, rails to individual buffer terminals*

Asian IBIS Summit, November, 2017   12

[Interconnect Model]

- [Interconnect Model] <interconnect_model_name>
    - Connections between terminals with IBIS-ISS or Touchstone files
    - Terminal connection points at <u>Buffer</u>, <u>Die pad</u>, or <u>Pin</u> interfaces
    - Identifies <u>rail</u> or <u>I/O</u> terminals
    - Allows pin_name, signal_name, pad_name, or bus_label terminal qualifiers for rails (and pin_name for I/O terminals)
    - Identifies whether a coupled signal is only an aggressor or also "experiences" coupling from other sources

> How package and on-die electrical information
> is generated and delivered today

Asian IBIS Summit, November, 2017   13

<Terminal lines> Syntax

- All column entries on one line:

  <Terminal_number> <Terminal_type>
   <Terminal_type_qualifier> <Qualifier_entry>
   [Aggressor_Only]

- <Terminal_number> is <u>IBIS-ISS</u> <u>node</u> <u>position</u> or <u>Touchstone</u> <u>port</u> <u>number</u>
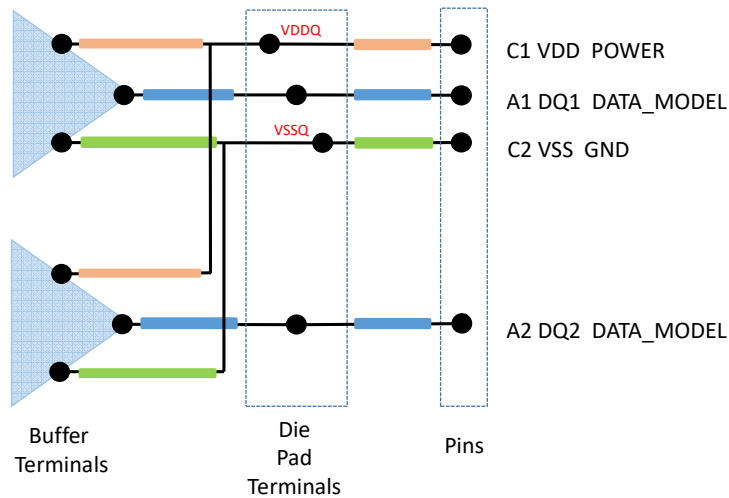
- Allowable <Terminal_type> names and associations next

Asian IBIS Summit, November, 2017   14

Allowable <Terminal_type> Associations

<Terminal_number> <Terminal_type> <Terminal_type_qualifier> <Qualifier_entry> [Aggressor_Only]

Terminal_type	Terminal_type_qualifier				Aggressor_Only
	pin_name	signal_name	bus_label	pad_name	
Pin_I/O	X				A
Pad_I/O	X				A
Buffer_I/O	X				A
Pin_Rail	Y	Y	Y		
Pad_Rail		Y	Y	Z	
Buffer_Rail		Y	Y		
Pullup_ref	X				
Pulldown_ref	X				
Power_clamp_ref	X				
Gnd_clamp_ref	X				
Ext_ref	X				

<Qualifier_entry>: "X" I/O pin_name; "Y," or "Z": POWER or GND name. Optional "A": "Aggressor_Only"

Asian IBIS Summit, November, 2017   15

Example Showing Connections

VDDQ

C1 VDD POWER

A1 DQ1 DATA_MODEL

VSSQ

C2 VSS GND

A2 DQ2 DATA_MODEL

Buffer
Terminals

Die
Pad
Terminals

Pins

Asian IBIS Summit, November, 2017   16

[Die Supply Pads] for pad_names Shown in Example

The [Die Supply Pads] keyword establishes pad_name
<Qualifier_entries> for rails, and associates them with
signal_name (and optionally with bus_label entries)

```
[Die Supply Pads] signal_name bus_label
| pad_name
VDDQ                VDD
VSSQ                VSS
```

[Interconnect Model] for Buffer-to-Die Pad Side


```
[Interconnect Model Set] Full_ISS_PDN
|
[Interconnect Model] Partial_ISS_buf_pad
|
File_IBIS-ISS        buf_pad.iss  buf_pad_2_typ
Number_of_terminals = 10
|
1  Pad_I/O       pin_name    A1 |  DQ1 (DQ signal)
2  Pad_I/O       pin_name    A2 |  DQ2 (DQ signal)
|
| POWER and GND terminals with pad_names and pin_names
3  Pullup_ref    pin_name    A1 |  VDD (POWER connection)
4  Pulldown_ref  pin_name    A1 |  VSS (GND connection)
5  Buffer_I/O    pin_name    A1 |  DQ1 (DQ signal)
6  Pullup_ref    pin_name    A2 |  VDD (POWER connection)
7  Pulldown_ref  pin_name    A2 |  VSS (GND connection)
8  Buffer_I/O    pin_name    A2 |  DQ2 (DQ signal)
|
| POWER and GND terminals with signal_names
9  Pad_Rail      pad_name    VDDQ | VDD  POWER
10 Pad_Rail      pad_name    VSSQ | VSS  GND
|
[End Interconnect Model]
```
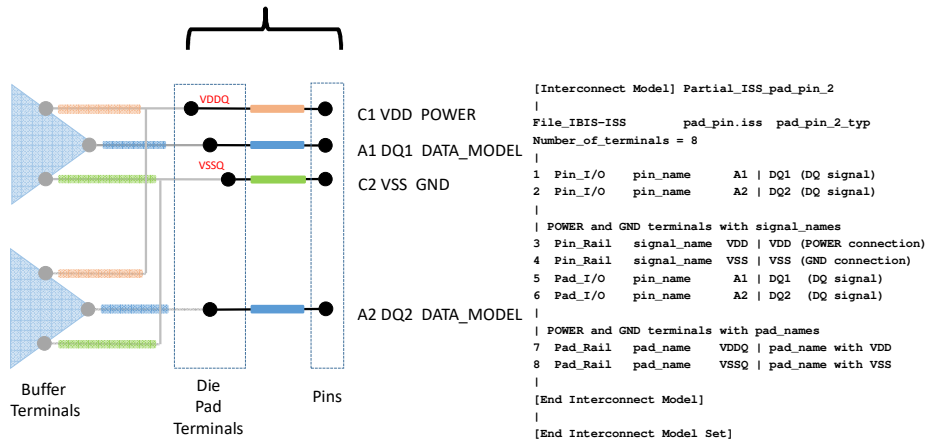
VDDQ

VSSQ

C1 VDD POWER

A1 DQ1 DATA_MODEL

C2 VSS GND

A2 DQ2 DATA_MODEL

Buffer
Terminals

Die
Pad
Terminals

Pins

[Interconnect Model] for
Buffer-to-Die Pad Side (Expanded)

```
[Interconnect Model Set] Full_ISS_PDN
|
[Interconnect Model] Partial_ISS_buf_pad
|
File_IBIS-ISS        buf_pad.iss  buf_pad_2_typ
Number_of_terminals = 10
|
1  Pad_I/O       pin_name    A1 |  DQ1 (DQ signal)
2  Pad_I/O       pin_name    A2 |  DQ2 (DQ signal)
|
| POWER and GND terminals with pad_names and pin_names
3  Pullup_ref    pin_name    A1 |  VDD (POWER connection)
4  Pulldown_ref  pin_name    A1 |  VSS (GND connection)
5  Buffer_I/O    pin_name    A1 |  DQ1 (DQ signal)
6  Pullup_ref    pin_name    A2 |  VDD (POWER connection)
7  Pulldown_ref  pin_name    A2 |  VSS (GND connection)
8  Buffer_I/O    pin_name    A2 |  DQ2 (DQ signal)
|
| POWER and GND terminals with signal_names
9  Pad_Rail      pad_name    VDDQ |  VDD  POWER
10 Pad_Rail      pad_name    VSSQ |  VSS  GND
|
[End Interconnect Model]
```

[Interconnect Model]
File and subcircuit

<Terminal lines> for connecting the subcircuit nodes (by position) to the interconnect structure

Asian IBIS Summit, November, 2017   19

[Interconnect Model] for
Die Pad-to-Pin Side

C1 VDD POWER
A1 DQ1 DATA_MODEL
C2 VSS GND
A2 DQ2 DATA_MODEL

Buffer Terminals
Die Pad Terminals
Pins

```
[Interconnect Model] Partial_ISS_pad_pin_2
|
File_IBIS-ISS        pad_pin.iss  pad_pin_2_typ
Number_of_terminals = 8
|
1  Pin_I/O    pin_name      A1 | DQ1 (DQ signal)
2  Pin_I/O    pin_name      A2 | DQ2 (DQ signal)
|
| POWER and GND terminals with signal_names
3  Pin_Rail   signal_name  VDD | VDD (POWER connection)
4  Pin_Rail   signal_name  VSS | VSS (GND connection)
5  Pad_I/O    pin_name      A1 | DQ1  (DQ signal)
6  Pad_I/O    pin_name      A2 | DQ2  (DQ signal)
|
| POWER and GND terminals with pad_names
7  Pad_Rail   pad_name     VDDQ | pad_name with VDD
8  Pad_Rail   pad_name     VSSQ | pad_name with VSS
|
[End Interconnect Model]
|
[End Interconnect Model Set]
```

Asian IBIS Summit, November, 2017   20

[Interconnect Model] for Die Pad-to-Pin Side (Expanded)

[Interconnect Model]

File and subcircuit

<Terminal lines> for connecting the subcircuit nodes (by position) to the interconnect structure

```
[Interconnect Model] Partial_ISS_pad_pin_2
|
File_IBIS-ISS        pad_pin.iss  pad_pin_2_typ
Number_of_terminals = 8
|
1  Pin_I/O     pin_name      A1 | DQ1 (DQ signal)
2  Pin_I/O     pin_name      A2 | DQ2 (DQ signal)
|
| POWER and GND terminals with signal_names
3  Pin_Rail    signal_name  VDD | VDD (POWER connection)
4  Pin_Rail    signal_name  VSS | VSS (GND connection)
5  Pad_I/O     pin_name      A1 | DQ1  (DQ signal)
6  Pad_I/O     pin_name      A2 | DQ2  (DQ signal)
|
| POWER and GND terminals with pad_names
7  Pad_Rail    pad_name     VDDQ | pad_name with VDD
8  Pad_Rail    pad_name     VSSQ | pad_name with VSS
|
[End Interconnect Model]
|
[End Interconnect Model Set]
```

Asian IBIS Summit, November, 2017   21

Complete [Interconnect Model Set] With Both [Interconnect Model]s

[Interconnect Model Set]   [Interconnect Model]s

```
[Interconnect Model Set] Full_ISS_PDN
|
[Interconnect Model] Partial_ISS_buf_pad
|
File_IBIS-ISS      buf_pad.iss  buf_pad_2_typ
Number_of_terminals = 10
|
1  Pad_I/O        pin_name      A1 | DQ1 (DQ signal)
2  Pad_I/O        pin_name      A2 | DQ2 (DQ signal)
|
| POWER and GND terminals with pad_names and pin_names
3  Pullup_ref     pin_name      A1 |  VDD (POWER connection)
4  Pulldown_ref   pin_name      A1 |  VSS (GND connection)
5  Buffer_I/O     pin_name      A1 |  DQ1 (DQ signal)
6  Pullup_ref     pin_name      A2 |  VDD (POWER connection)
7  Pulldown_ref   pin_name      A2 |  VSS (GND connection)
8  Buffer_I/O     pin_name      A2 |  DQ2 (DQ signal)
|
| POWER and GND terminals with signal_names
9  Pad_Rail       pad_name     VDDQ |  VDD  POWER
10 Pad_Rail       pad_name     VSSQ |  VSS  GND
|
[End Interconnect Model]
```

```
[Interconnect Model] Partial_ISS_pad_pin_2
|
File_IBIS-ISS        pad_pin.iss  pad_pin_2_typ
Number_of_terminals = 8
|
1  Pin_I/O     pin_name      A1 | DQ1 (DQ signal)
2  Pin_I/O     pin_name      A2 | DQ2 (DQ signal)
|
| POWER and GND terminals with signal_names
3  Pin_Rail    signal_name  VDD | VDD (POWER connection)
4  Pin_Rail    signal_name  VSS | VSS (GND connection)
5  Pad_I/O     pin_name      A1 | DQ1  (DQ signal)
6  Pad_I/O     pin_name      A2 | DQ2  (DQ signal)
|
| POWER and GND terminals with pad_names
7  Pad_Rail    pad_name     VDDQ | VDD is signal name
8  Pad_Rail    pad_name     VSSQ | VSS is signal name
|
[End Interconnect Model]
|
[End Interconnect Model Set]
```

[End Interconnect Model Set]

Asian IBIS Summit, November, 2017   22

[Interconnect Model Set Group] (New) for a Selected Group

```
[Interconnect Model Set Group]     A1_A2_PDN
|
| Interconnect Model Set Name      File_reference
Full_ISS_PDN                       NA
|
[End Interconnect Model Set Group]
```

[Interconnect Model Set Group] is at same level as [Package Model] for selected group of Buffer_IO pin(s)

Name should be descriptive for easy selection (e.g., A1-A2_PDN)

Can contain several references to [Interconnect Model Set]s

Sets can be in the .ibs file (NA) or in a separate directories

[Interconnect Model]s within a Group must be connected

Asian IBIS Summit, November, 2017   23

File_TS, File_TS0 (New) Issue

❑ Touchstone files can now be documented with a single reference (File_TS) or ground "Node 0" reference (File_TS0)
  – Offering both choices eliminates issue about which is better
  – File_TS0 would not be used for power-aware simulations
❑ File_IBIS-ISS with S model can be used for more references, if needed

Asian IBIS Summit, November, 2017   24

Touchstone Unused Port Termination

- Not an issue with IBIS-ISS – all terminal connections are required
- For Touchstone files
  - Unused_port_termination <Open | Reference>
  - Reference: reference impedance reduces the number of Touchstone ports through matrix reduction
  - Open: represents the physically disconnected port
  - EDA tools might still provide an interface to override the choices
- Other options still being discussed

Summary

- BIRD189.x improves IBIS package modeling

- Links IBIS, IBIS-ISS and Touchstone for package models
  - Adds flexible support for package loss, crosstalk and partitioning

- Formalizes and separates Die pads and Buffers

- Other extensions (not covered here) included

New advanced Interconnect format for IBIS Version 7.0!

Leveraging IBIS Capabilities for Multi-Gigabit Interfaces

Ken Willis - Cadence Design Systems
Asian IBIS Summit,
Taipei, ROC
November 15, 2017

cādence®

Overview

- In writing EDI CON paper "Signal Integrity Methodology for Double-Digit Multi-Gigabit Interfaces", different IBIS modeling techniques were applied
- Wanted to share some of the IBIS modeling methods we've been using

cādence®

1

Agenda

- [External Model]
- AMI equalization adaptation
- Backchannel training
- Applying IBIS-AMI to DDR applications

cādence®

Agenda

- ***[External Model]***
- AMI equalization adaptation
- Backchannel training
- Applying IBIS-AMI to DDR applications

cādence®

2

Preliminary IBIS Modeling

- Can get started with:
  - Voltage swing
  - Pad capacitance
  - Output impedance
  - Rise time
- Even easier to model using [External Model] syntax

```
|
[Model]    pcie4_out
Model_type Output
Polarity Non-Inverting
Vmeas =    0.5
|
| variable                       typ        min        max
C_comp                           0.50pF     0.50pF     0.50pF
[Temperature Range]              20         80         -50
[Voltage range]                  1.0        1.0        1.0
|
|********************************************************************
[Pulldown]
|
|               Voltage          I(typ)     I(min)     I(max)
|
|   |   |   |   -1.0000e+00      -2.0000e-02 -2.0000e-02 -2.0000e-02
|               0.0000e-00       0.0000e-00 0.0000e-00 0.0000e-00
|               1.0000e+00       2.0000e-02 2.0000e-02 2.0000e-02
|
|********************************************************************
[Pullup]
|
|               Voltage          I(typ)     I(min)     I(max)
|
|   |   |   |   -1.0000e+00      2.0000e-02 2.0000e-02 2.0000e-02
|               0.0000e-00       0.0000e-00 0.0000e-00 0.0000e-00
|               1.0000e+00       -2.0000e-02 -2.0000e-02 -2.0000e-02
|
|********************************************************************
[Ramp]
|
| variable       typ          min           max
dV/dt_r        0.60/0.020n  0.60/0.020n   0.60/0.020n
dV/dt_f        0.60/0.020n  0.60/0.020n   0.60/0.020n
|
R_load = 50
|
```

cādence®

External Model

- Originally introduced in IBIS 4.1 !
- Enabled VHDL, Verilog-A, and Spice syntax to be used for buffer models instead of VI/VT table syntax
- Spice syntax has proven very useful to us

cādence®

3

Spice [External Model]s Convenient for Early Feasibility

- Sometimes the IBIS model you want is not available for preliminary / pre-design analysis
- Easy to write (or use) simple parameterized Spice subcircuits for IO buffers when IBIS availability does not align with your project schedule
- Can sweep parameters and cover a big portion of the design space quickly and easily

cādence®

Analog Buffer Model Extensions

- Incorporated into IBIS 6.0

```
[External Model]
Language IBIS-ISS
|
| Corner corner_name file_name_circuit_name (.subckt name)
Corner    Typ        cdnstxrx.cir    tx_subckt
|
| List of parameters
Parameters  TSFile1      = cdns_tx.param(CDNS_Tx(Model_Specific(Tstonefile)))
Parameters  Tx_Rseries   = cdns_tx.param(CDNS_Tx(Model_Specific(Tx_R)))
|
| List of converter parameters
Converter_Parameters  Vtx_h = cdns_tx.param(CDNS_Tx(Model_Specific(Tx_V)))
|
| Ports List of port names (in same order as in SPICE)
Ports A_signal_pos A_signal_neg my_driveP  A_pdref  A_puref
|
| D_to_A d_port port1 port2 vlow vhigh trise tfall corner_name polarity
D_to_A  D_drive  my_driveP  A_pdref  0  Vtx_h  0.05n  0.03n  Typ
|
|
[End External Model]
```

Tx/Rx Model described in external file
- Replaces the VI/VT curves
- Can now be parameterized when Language 'IBIS-ISS' is used

Parameter Definition is in a separate file
- Similar to AMI Parameters
- Can be in .ami or any other file
- User Selects Parameter Values from GUI

Parameter Value is passed to Simulator
- Using .param, the parameter value is passed to the simulator
- Converter_Parameters are used appropriately in the stimulus/D_to_A

cādence®

4

Agenda

- [External Model]
- ***AMI equalization adaptation***
- Backchannel training
- Applying IBIS-AMI to DDR applications

cādence®

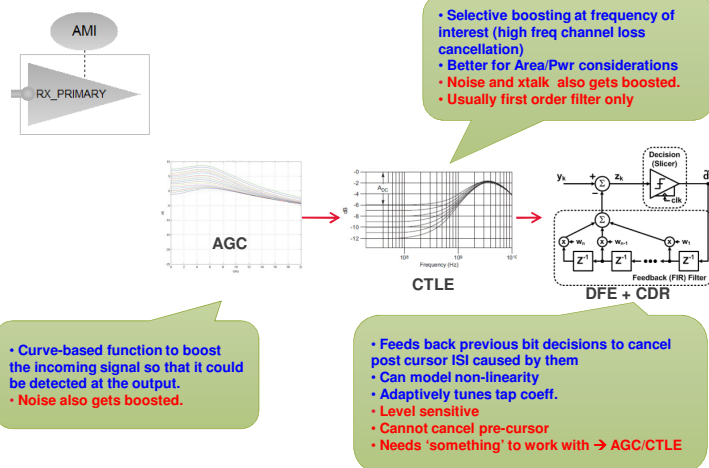
Anatomy of a Receiver AMI Model

- These different modules typically adapt at different rates
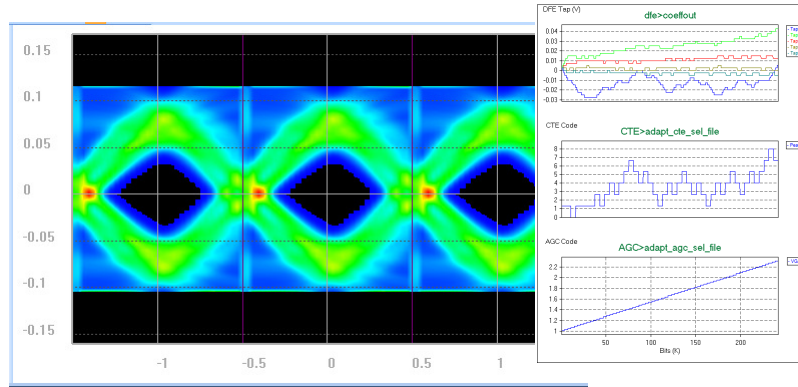- Initial modules usually adapt more slowly than later ones

- Selective boosting at frequency of interest (high freq channel loss cancellation)
- Better for Area/Pwr considerations
- Noise and xtalk also gets boosted.
- Usually first order filter only

AMI

RX_PRIMARY

AGC

CTLE

DFE + CDR

- Curve-based function to boost the incoming signal so that it could be detected at the output.
- Noise also gets boosted.

- Feeds back previous bit decisions to cancel post cursor ISI caused by them
- Can model non-linearity
- Adaptively tunes tap coeff.
- Level sensitive
- Cannot cancel pre-cursor
- Needs 'something' to work with → AGC/CTLE

cādence®

5

Rx With Default Adaptation

• Note that adaptation coefficients don't converge

cādence®

With Faster AGC Adaptation

• Coefficients converge, but after 150k bits of traffic are passed

cādence®

6
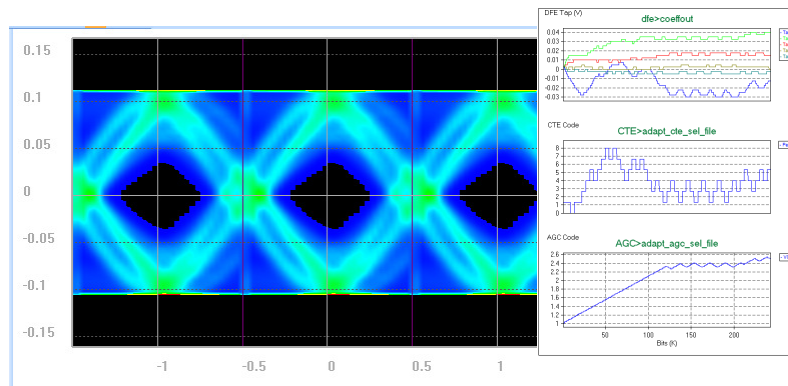
Ignoring the First 150k Bits

- Default was to ignore the first 40k bits
- Eliminates the noise from before coefficients converged
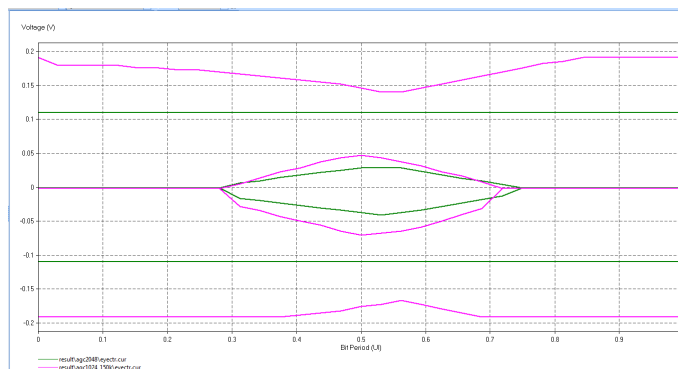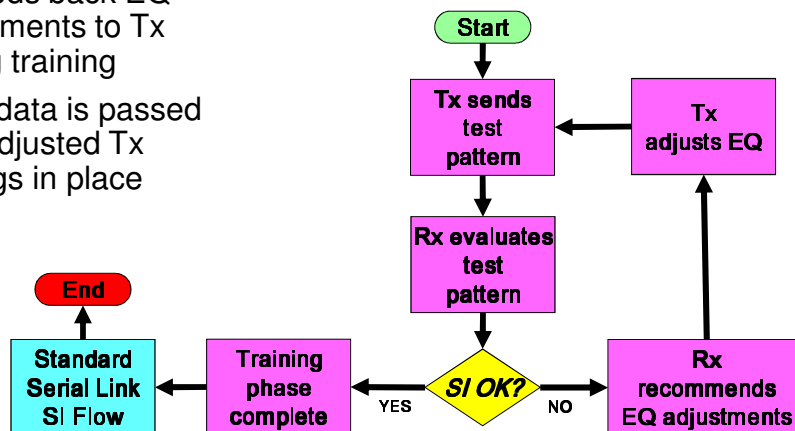- Very important to be able to visualize how the adaptation is converging
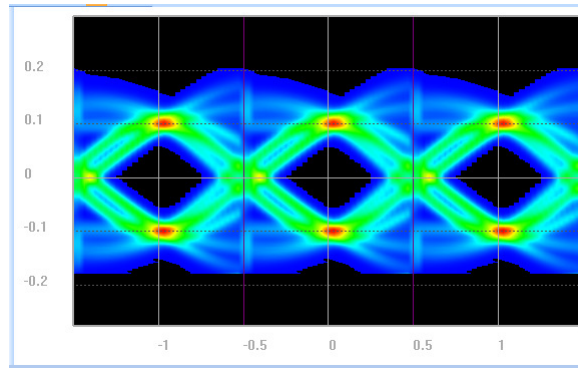
cādence®

Original Eye Contour vs. Final

- Adjusting AGC adaptation time and Ignore_Bits made a significant difference in eye height

cādence®

7

Agenda

- [External Model]
- AMI equalization adaptation
- ***Backchannel training***
- Applying IBIS-AMI to DDR applications

cadence®

Backchannel Training (IBIS BIRD 147, in IBIS 7.0)

- Rx feeds back EQ adjustments to Tx during training
- Then data is passed with adjusted Tx settings in place

Start

Tx sends test pattern → Tx adjusts EQ

Rx evaluates test pattern

End

Standard Serial Link SI Flow ← Training phase complete ← SI OK? — YES

SI OK? — NO → Rx recommends EQ adjustments

cadence®

8

With and Without Backchannel

- Backchannel turned down Tx FFE settings somewhat
- Leaves more "heavy lifting" to Rx and its advanced adaptation
- Improves overall signal quality significantly
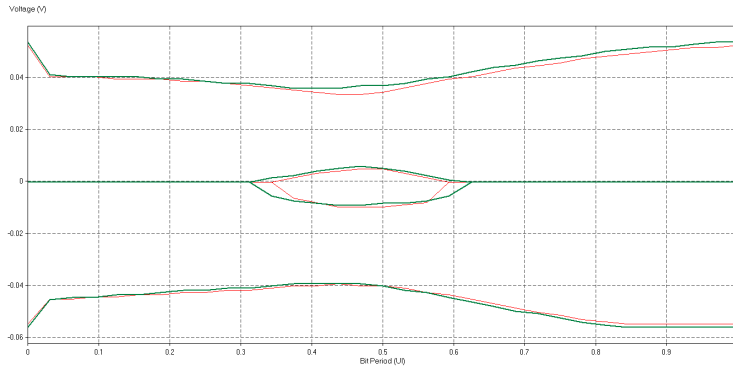
cādence®

Agenda

- [External Model]
- AMI equalization adaptation
- Backchannel training
- ***Applying IBIS-AMI to DDR applications***

cādence®

9

DDR4 Brought Some New Requirements

- Specified DQ mask compliance checking at a particular BER
- BER analysis requires extrapolation (bathtubs)
- Extrapolation requires a lot of traffic to be passed (need a lot of samples)
  - Channel simulation can be applied
- Started to see equalization used at Controller side
  - AMI modeling can be applied
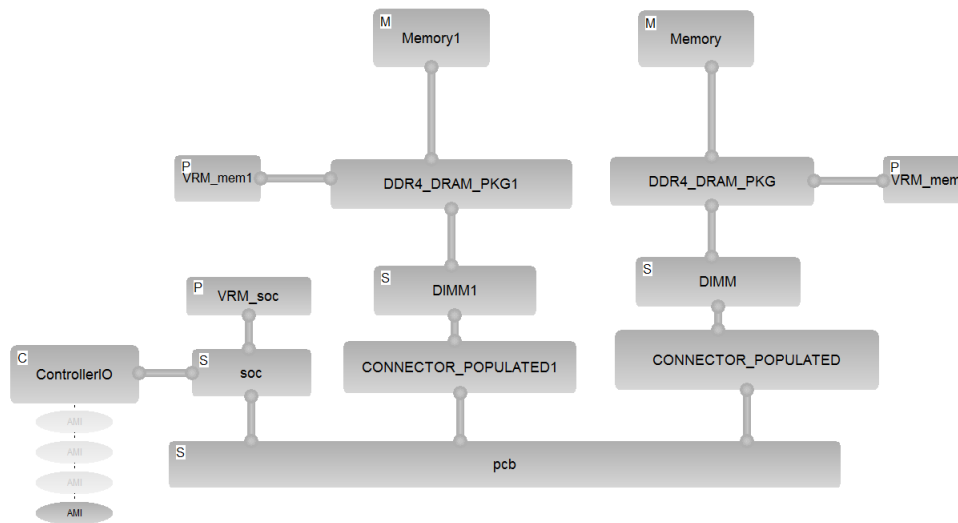- Worked with IP division to develop AMI model for DDR4 IP
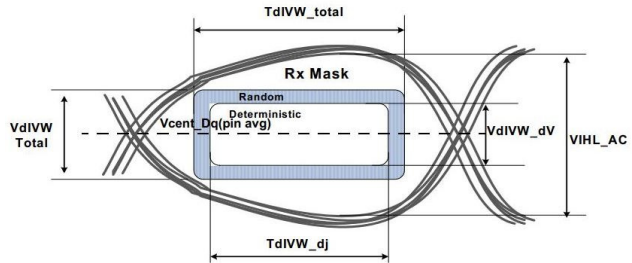
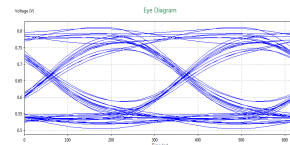cādence®

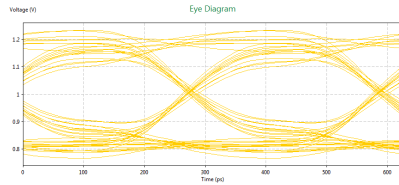# Simulation Testbench

cādence®
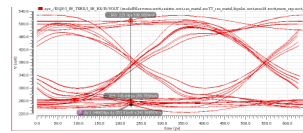
10

CTLE Correlation: 3200Mbps

Input of receiver @pad   Output of CTLE

IBIS-AMI
channel sim

Transistor-
Level
circuit sim
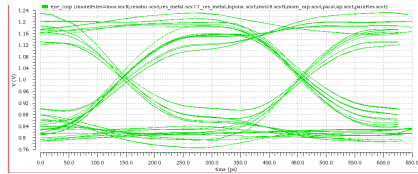
cādence®
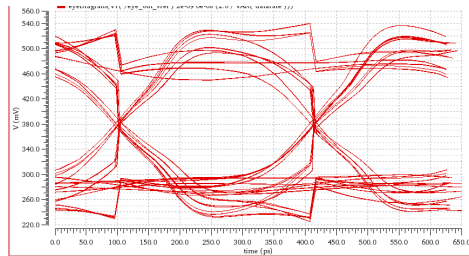
CTLE + DFE Correlation: 3200Mbps

Transistor-
Level
circuit sim
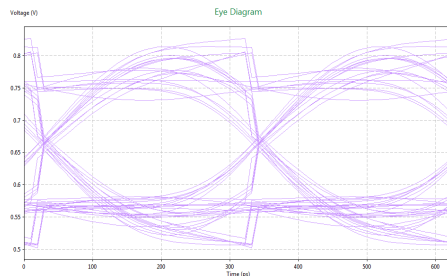
IBIS-AMI
channel sim

cādence®

11

Summary

- External Model syntax can be very useful for pre-design modeling, when detailed IBIS models are not available, and has had some recent additions in capability
- Building IBIS-AMI models is not the obstacle it used to be
- Adaptive equalization often has interplay between multiple sub-modules in real devices, and therefore also in AMI models
- If adaptive, understand if your EQ coefficients converge during simulation
- Backchannel training enables interplay between the Tx and Rx in simulation, and can produce more realistic results for devices that use backchannel
- Channel simulation and AMI modeling has been successfully applied to DDR4 IP (and more of this is expected with DDR5)

cādence®

cādence®

12

Comparison of Time Domain and Statistical IBIS-AMI Analyses

Mike LaBonte

SiSoft

Asian IBIS Summit 2017

Taipei, ROC

November 15, 2017

SiSoft™
We Are Signal Integrity

---

9 Combinations of TX and RX Model Types

- AMI file has:
  - GetWave_Exists = True   Best for bit-by-bit simulation
  - Init_Returns_Impulse = True   Best for statistical analysis
 - 3 types: "Init-only", "GetWave-only", "Dual"
  - 3 TX * 3 RX = 9 combinations

Case #	TX			RX			Convolution Input	Statistical	Time Domain
	Getwave Exists	Init_Returns_Impulse	Meaning	Getwave Exists	Init_Returns_Impulse	Meaning			
1	FALSE	TRUE	Init-Only	FALSE	TRUE	Init-Only	3	OK	Static TX EQ, Static RX Eq
2	FALSE	TRUE	Init-Only	TRUE	FALSE	Getwave-Only	1 or 2	No RX EQ	Static TX EQ, Dynamic RX Eq
3	FALSE	TRUE	Init-Only	TRUE	TRUE	Dual	2	OK	Static TX EQ, Dynamic RX Eq
4	TRUE	FALSE	Getwave-Only	FALSE	TRUE	Init-Only	3	No TX EQ	Dynamic TX EQ, Static RX Eq
5	TRUE	FALSE	Getwave-Only	TRUE	FALSE	Getwave-Only	1,2,or 3	No TX or RX EQ	Dynamic TX EQ, Dynamic RX EQ
6	TRUE	FALSE	Getwave-Only	TRUE	TRUE	Dual	1	No TX EQ	Dynamic TX EQ, Dynamic RX EQ
7	TRUE	TRUE	Dual	FALSE	TRUE	Init-Only	iFFT(FFT(3)/FFT(2))	OK	Dynamic TX EQ, Static RX Eq
8	TRUE	TRUE	Dual	TRUE	FALSE	Getwave-Only	1	No RX EQ	Dynamic TX EQ, Dynamic RX EQ
9	TRUE	TRUE	Dual	TRUE	TRUE	Dual	1	OK	Dynamic TX EQ, Dynamic RX EQ

SiSoft™
We Are Signal Integrity

Simulation limitations

🟩	Correct equalization of TX and RX modeled
🟩	Correct equalization of TX and RX modeled: Assumes no adaptation in TX
🟧	Assumes Static RX Equalization is a good representation of the RX: No adaptation
🟨	Assumes Static RX EQ is a good representation of the RX: No Adaptation, Requires advanced math capabilities in Simulator
🟥	Equalization data is missing

Case #	TX	RX	Statistical	Time Domain
1	Init Model Only	Init Model Only	OK	Static TX EQ, Static RX EQ
2	Init Model Only	Getwave Model Only	No RX EQ	Static TX EQ, Dynamic RX EQ
3	Init Model Only	Dual Model	OK	Static TX EQ, Dynamic RX EQ
4	Getwave Model Only	Init Model Only	No TX EQ	Dynamic TX EQ, Static RX EQ
5	Getwave Model Only	Getwave Model Only	No TX or RX EQ	Dynamic TX EQ, Dynamic RX EQ
6	Getwave Model Only	Dual Model	No TX EQ	Dynamic TX EQ, Dynamic RX EQ
7	Dual Model	Init Model Only	OK	Dynamic TX EQ, Static RX EQ
8	Dual Model	Getwave Model Only	No RX EQ	Dynamic TX EQ, Dynamic RX EQ
9	Dual Model	Dual Model	OK	Dynamic TX EQ, Dynamic RX EQ

Best Option

SiSoft
We Are Signal Integrity

Time-Domain Simulation

- **Inputs:**
  - Channel and buffer Impulse responses
  - User-defined input stimulus
  - Algorithmic models (AMI_GetWave)

- **Analysis Method:**
  - Waveform processing & convolution

- **Outputs:**
  - Bit pattern waveforms
  - Persistent eye diagrams
  - Eye height / width measurements
  - Eye contours @ probabilities
  - Equalized / unequalized responses

Analog Channel

Stimulus

Impulse Response

User Settings

Time-Domain Engine

TX
AMI_Getwave
AMI_Init

RX
AMI_Getwave
AMI_Init

SiSoft
We Are Signal Integrity

Statistical Simulation

- **Inputs:**
  - Analog channel impulse response
  - User selections for model parameters
  - Algorithmic models (==AMI_Init / impulse response processing==)

- **Analysis Method:**
  - Convolution engine (pulse response)

- **Outputs:**
  - Statistical eye diagrams
  - Eye height / width measurements
  - Eye contours @ probabilities
  - Equalized / unequalized responses

Analog Channel

Impulse Response   User Settings

Statistical
Engine

TX
AMI_Init

RX
AMI_Init
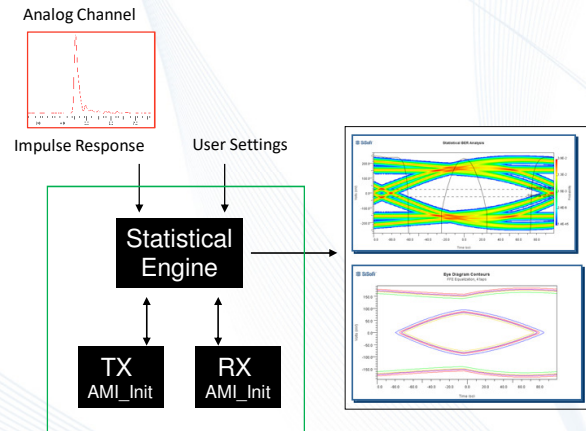
Nov 2017   IBIS-AMI Analysis Comparison   5

SiSoft
We Are Signal Integrity

Which IBIS-AMI Model Type is Best?

- Need to evaluate suitability for modeling:

  - Impairments: The factors that harm the signal

    - Mostly in the channel

    - Statistical analysis has advantages

  - Corrective measures: Signal improvements

    - Mostly inside the SerDes

    - Time domain has advantages

Nov 2017   IBIS-AMI Analysis Comparison   6

SiSoft
We Are Signal Integrity

Impairments To Be Modeled

Amplitude Impairment	Physical Cause
Inter-symbol interference (ISI)	Signal distortion (linear and nonlinear)
Crosstalk	Electromagnetic coupling in passive interconnect
Receiver sensitivity	Low signal amplitude causes decision latch to fail clock-data timing
Additive White Gaussian Noise (AWGN)	Shot noise in receiver amplifiers

Clock Impairment	Physical Cause
Random Jitter (RJ)	a. Shot noise in oscillator gain element b. Power supply noise modulating gate delays
Duty Cycle Distortion (clock) (DCD)	For half rate clock, duration difference between positive and negative half cycles
Duty Cycle Distortion (data)	Difference between data rise and fall times
Sinusoidal Jitter (SJ)	Clock noise on power supply modulating gate delays

Nov 2017   IBIS-AMI Analysis Comparison   7

SiSoft™
We Are Signal Integrity

Corrective Measures To Be Modeled

Corrective Measure	Time Domain Behavior
TX FFE	May adapt in time domain, but this is rare
RX CTLE	Linear, time-invariant (LTI)
RX AGC	Adapts in time domain
RX Saturation	Not adaptive, but not time-invariant either
RX DFE	Adapts in time domain
Others…	

Nov 2017   IBIS-AMI Analysis Comparison   8

SiSoft™
We Are Signal Integrity

Inter-Symbol Interference (ISI) Impairments

SiSoft
We Are Signal Integrity

Step Response Analysis
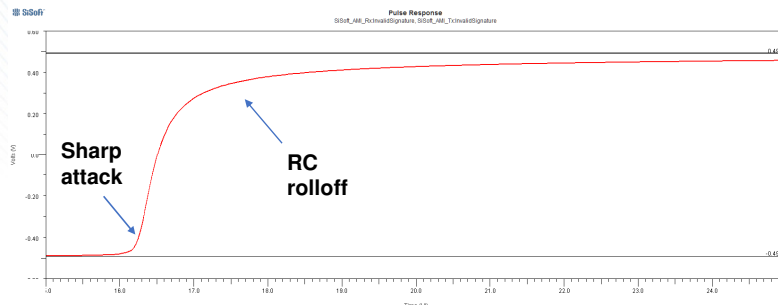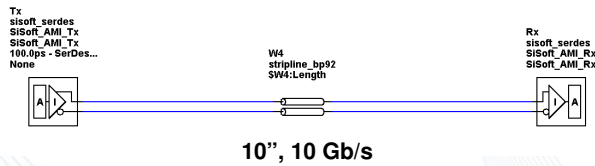
Tx
sisoft_serdes
SiSoft_AMI_Tx
SiSoft_AMI_Tx
100.0ps - SerDes...
None

W4
stripline_bp92
$W4:Length

Rx
sisoft_serdes
SiSoft_AMI_Rx
SiSoft_AMI_Rx

10", 10 Gb/s

Sharp attack

RC rolloff

SiSoft
We Are Signal Integrity

Pulse vs. Step Responses

10", 10 Gb/s
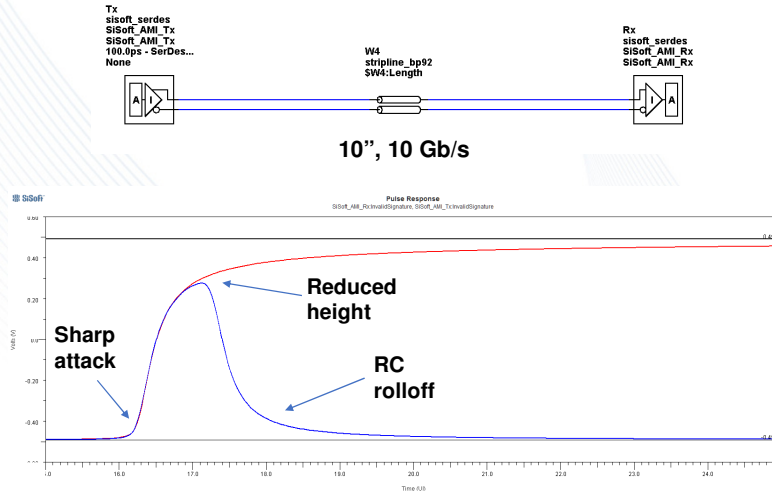
Channel Pulse Response
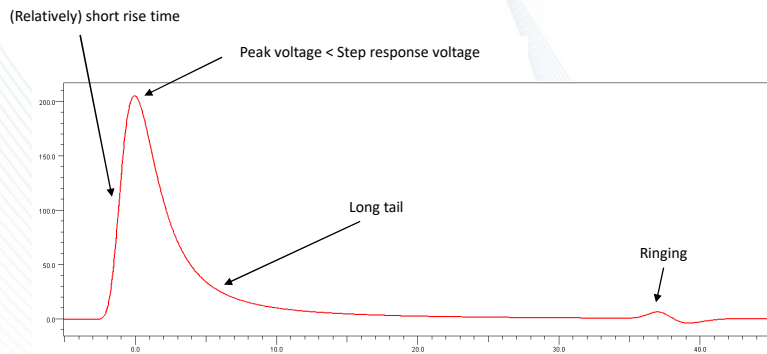
- Requires accurate Tx/Rx analog models to correctly predict ringing impairment due to reflections

Aligned Pulse Response and ISI

Hula hoop algorithm determines clock sampling time and main cursor height.
This is the maximum possible inner eye height.

Voltages at these points subtract from
the eye height at the sampling point.

$Inner\ Eye\ Height =$

$main\_cursor - \Sigma\ |ISI\_voltages|$

24 UI
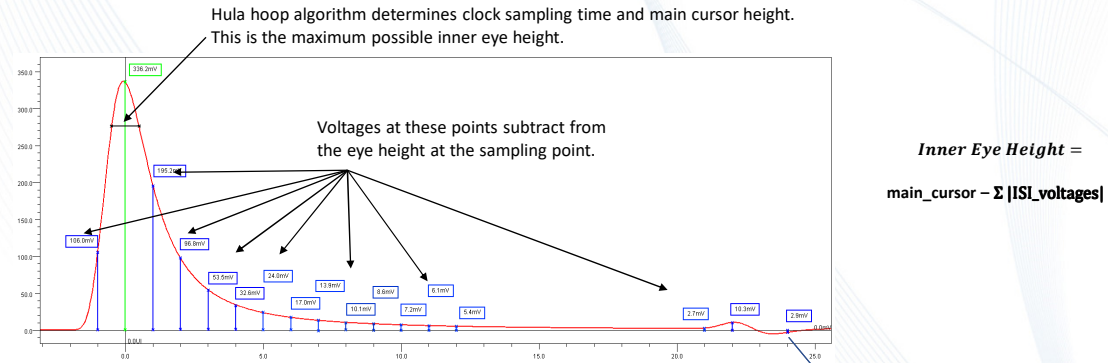
- Voltage and time scales show ISI contributions
- Useful in evaluating EQ & predicting eye opening

Nov 2017   IBIS-AMI Analysis Comparison   13

SiSoft
We Are Signal Integrity

Statistical ISI Inner Eye Quick Calculation

Prediction: 580mV
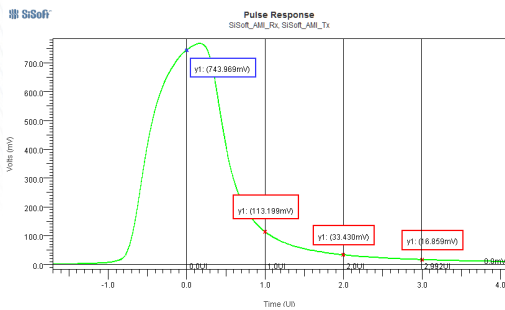
Simulated Actual: 550mV

$Inner\ Eye\ Height = main\_cursor - \Sigma\ |ISI\_voltages|$
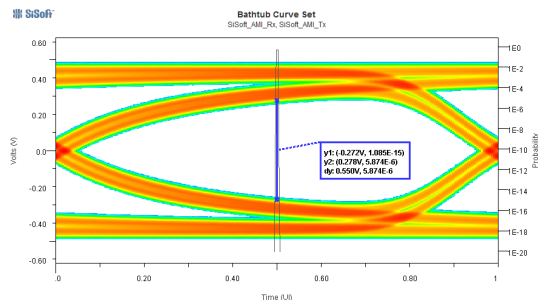
A quick calculation gets us close, but small amounts of energy in the tail add up

Nov 2017   IBIS-AMI Analysis Comparison   14

SiSoft
We Are Signal Integrity

Time Domain ISI

- Time domain waveform from impulse response
- Bit pattern modulated
- Linear superposition
- LTI assumed
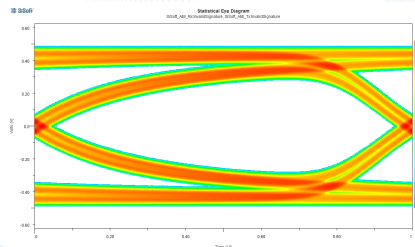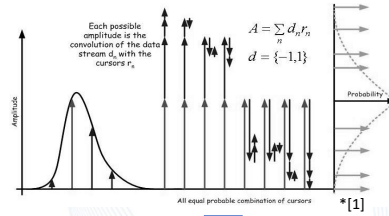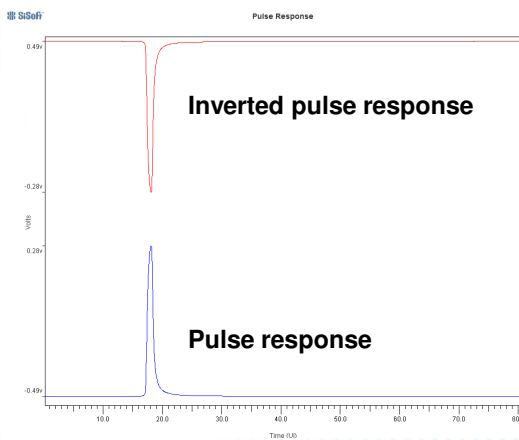
Example bit pattern: 11010
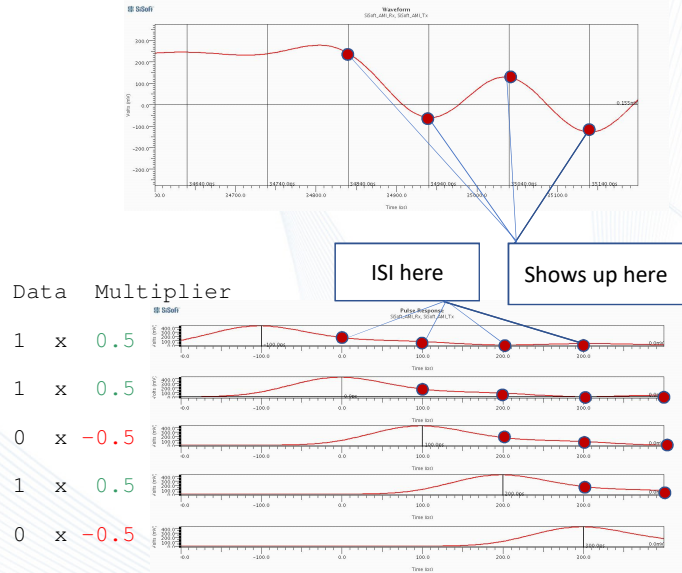
ISI here | Shows up here

```
Data  Multiplier
1   x   0.5
1   x   0.5
0   x  -0.5
1   x   0.5
0   x  -0.5
```

Nov 2017 · IBIS-AMI Analysis Comparison · 15

SiSoft
We Are Signal Integrity

Statistical ISI

Inverted pulse response

Pulse response

$$A = \sum_n d_n r_n$$
$$d = \{-1, 1\}$$

Each possible amplitude is the convolution of the data stream d_n with the cursors r_n

All equal probable combination of cursors

*[1]

Nov 2017 · IBIS-AMI Analysis Comparison · 16

SiSoft
We Are Signal Integrity

All Possible LTI Combinations Evaluated

Each possible amplitude is the convolution of the data stream d_n with the cursors r_n

$$A = \sum_n d_n r_n$$

$$d = \{-1, 1\}$$

Amplitude

Probability

*[1]

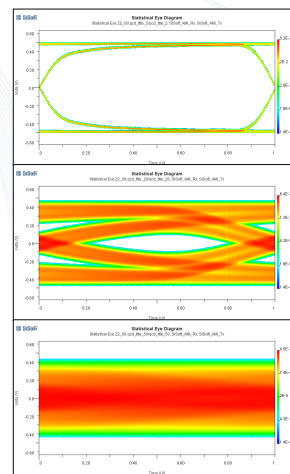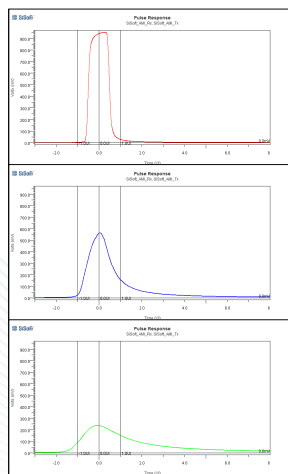All equal probable combination of cursors

SiSoft™
We Are Signal Integrity
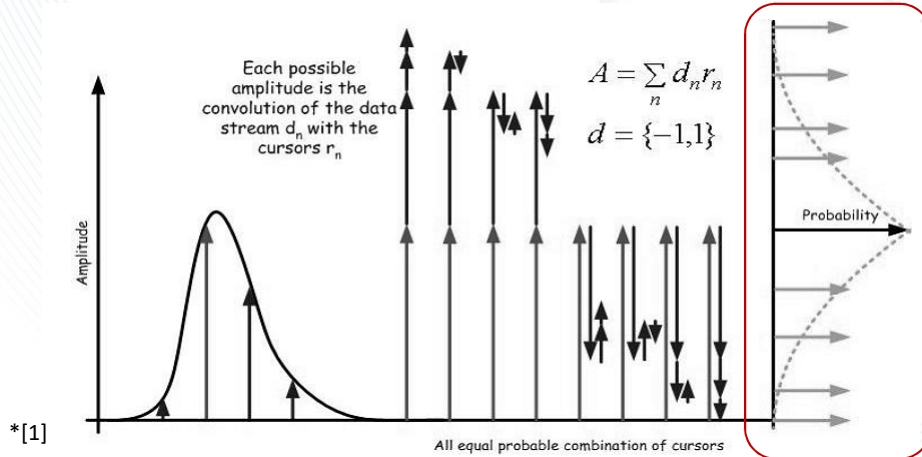
Channels, Pulses and Statistical Eyes

Short channel, Minimal ISI

Medium channel, Moderate ISI

Long channel, Extreme ISI

SiSoft™
We Are Signal Integrity

Accounting for All ISI Scenarios

- A 28Gbps link may have a bit every 0.2 inches
- Many bits can be on the channel at once
- With reflections that number is multiplied
- Required impulse response may be many UI in length
- <u>The bit pattern affects how these interact</u>

To completely model all possible ISI scenarios we must try every possible bit pattern for the number of UI needed to capture all significant ISI
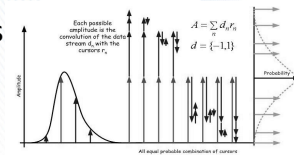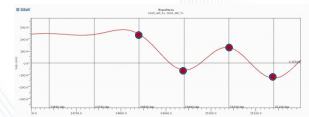
Nov 2017　　　　　IBIS-AMI Analysis Comparison　　　　　19

SiSoft
We Are Signal Integrity

Can We Account for All ISI Scenarios?

- Theoretically need to try 2^N patterns, where N is the number of UI before ISI becomes insignificant
  - Example: 24 UI NRZ impulse response must simulate 2^{24} = 16,777,216 patterns, each 24 UI in length, total of 402,653,184 bit computations
- Time domain simulation
  - N-length patterns tested sequentially
  - PRBS helps reduce redundancies
  - Often able to simulate only a fraction of cursor combinations
- Statistical analysis
  - Able to directly calculate all 2^N cursor combinations
  - Efficient computation of channel response, not a circuit
  - May still have a practical upper limit for N

Nov 2017　　　　　IBIS-AMI Analysis Comparison　　　　　20

SiSoft
We Are Signal Integrity

Jitter and Noise Impairments

SiSoft™
We Are Signal Integrity

Jitter and Noise in IBIS-AMI

- IBIS 6.1 provides multiple TX & RX impairments

- TX jitter directly modulates the TX output
  - Simulators jitter the stimulus pattern sent to the TX in time domain simulations
  - Statistical analysis convolves jitter with eye diagram

- RX jitter affects recovered clock behavior
  - Simulators combine jitter data with clock information returned by the RX
  - Statistical analysis convolves jitter with eye diagram

- RX noise affects sampling latch data input

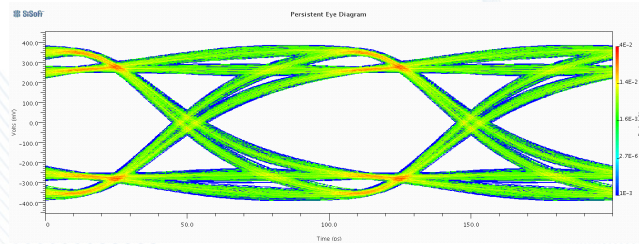- **Jitter and noise are handled by the simulator, not by the models**

SiSoft™
We Are Signal Integrity
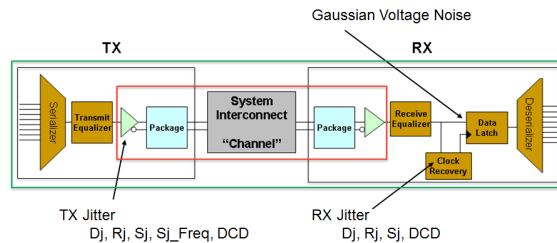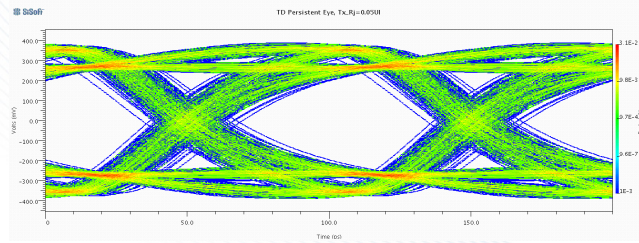
Time Domain Eyes With and Without Tx Jitter

Random Jitter
(Tx_Rj) = 0

Only Impairment is
Inter-Symbol
Interference (ISI)

1e-3

Tx_Rj = 0.05UI

ISI + Jitter

1e-3

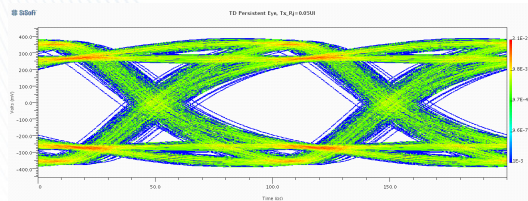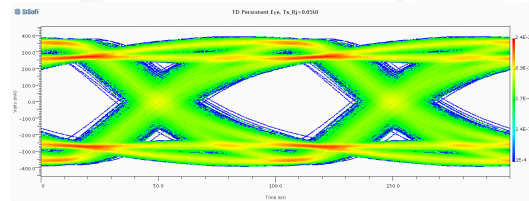Nov 2017　　　　　　　IBIS-AMI Analysis Comparison　　　　　　　23

SiSoft
We Are Signal Integrity
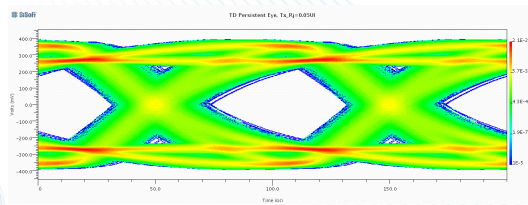

Time Domain: How Many Bits to Simulate?

1,000 UI

10,000 UI

100,000 UI

1,000,000 UI

Nov 2017　　　　　　　IBIS-AMI Analysis Comparison　　　　　　　24

SiSoft
We Are Signal Integrity
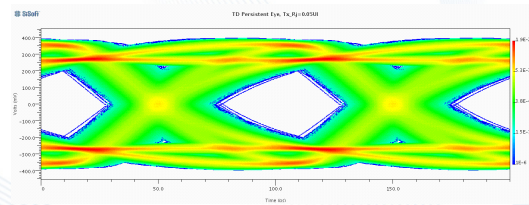
What Maximum BER Can We Tolerate?

- IEEE-802.3bj-KR4 FEC on   1e-5
- IEEE-802.3bj-KR4 FEC off   **1e-12**   if low latency required
- OIF-CEI-56G FEC on   1e-4
- OIF-CEI-56G FEC off   **1e-20**
- PCIe-G3   **1e-12**
- PCIe-G4   **1e-12**
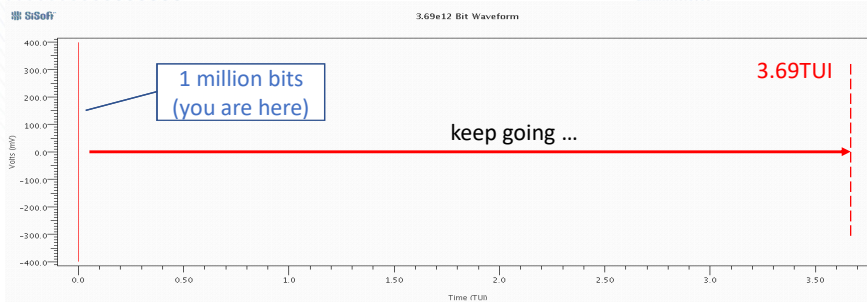- DDR4   **1e-12**   eye mask rules
- DDR5   TBD

Nov 2017   IBIS-AMI Analysis Comparison   25

SiSoft™
We Are Signal Integrity

How Many Error-Free Bits for 1e-12 BER?

It's Not 1e12

Confidence Level	90%	95%	99%
Maximum BER	1e-12	1e-12	1e-12
Error-free Bits Simulated *[2]	3.00e12	3.69e12	5.30e12

Nov 2017   IBIS-AMI Analysis Comparison   26

SiSoft™
We Are Signal Integrity

Statistical Eye With ISI and Jitter

BER = 1e-6
Contour

BER = 1e-12
Contour

BER = 6.44e-21
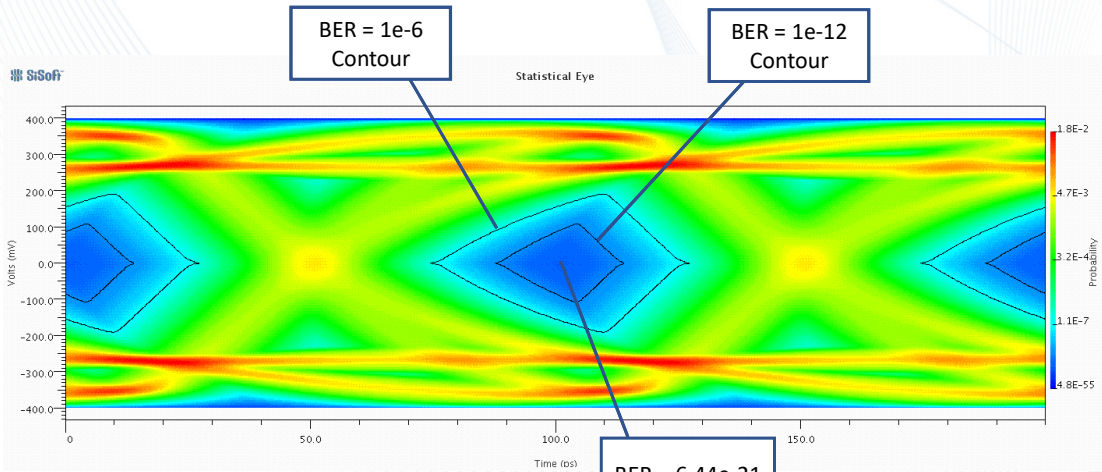
Nov 2017   IBIS-AMI Analysis Comparison   27

We Are Signal Integrity

Tx Corrective Measures

Nov 2017   IBIS-AMI Analysis Comparison   28

We Are Signal Integrity

# Desired Pulse Response for Low ISI

Pulse Response

Sampling clock position

Pulse energy should be confined here

**Any energy here causes
Inter-Symbol Interference (ISI)**

Nov 2017 IBIS-AMI Analysis Comparison 29

SiSoft
We Are Signal Integrity
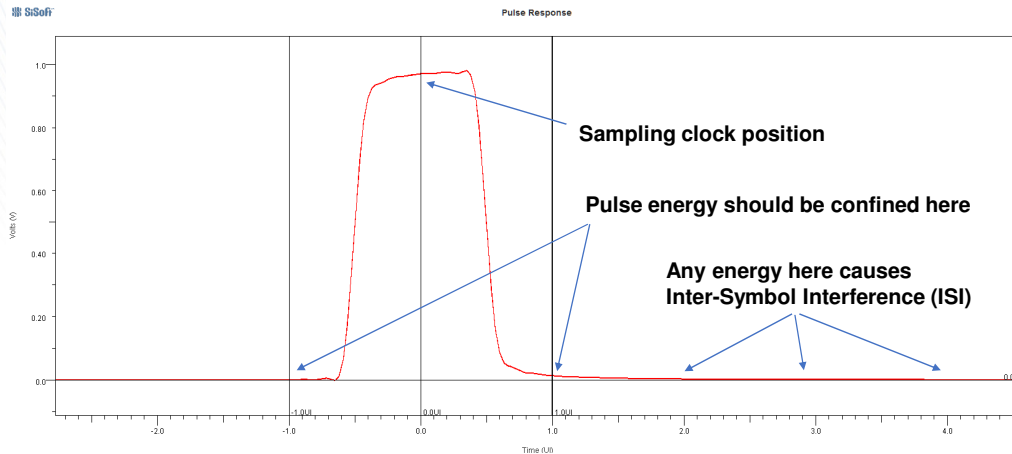
Tx Feed-Forward Equalization (FFE)

- Usually implemented as taps spaced at the signal data rate

- Can precede the signal (pre-cursor), follow the signal (post-cursor), or both

- Typical configuration is 1 pre-cursor, 2 post-cursor taps

$x[n]$ → z^{-1} → z^{-1} → ⋯ → z^{-1}

b_0 b_1 b_2 b_N

Σ → Σ → ⋯ → Σ → $y[n]$
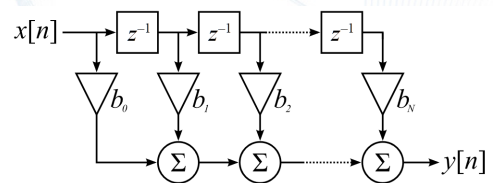
Nov 2017 IBIS-AMI Analysis Comparison 30

SiSoft
We Are Signal Integrity

TX FFE Equalization (1st post-cursor)

- Goal: boost high frequency content
- Transition occurs at full strength, then driver "pulls back" for subsequent bits
- TX EQ is often referred to as de-emphasis
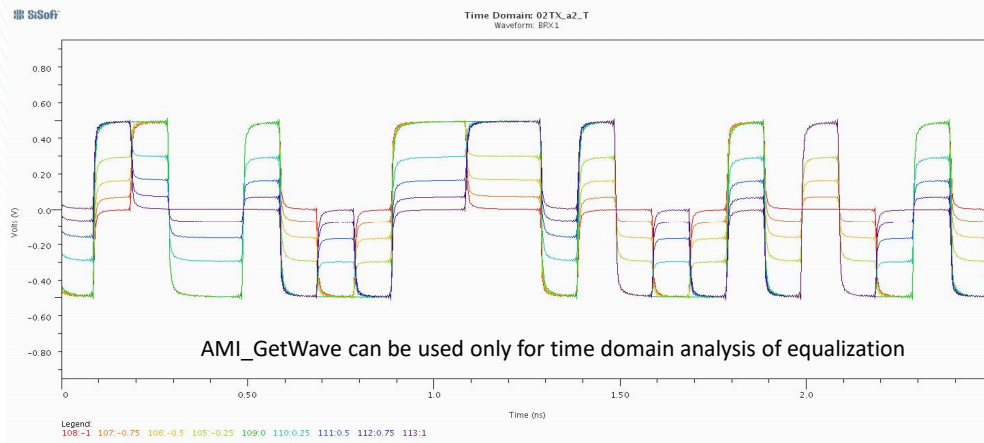- TX EQ always reduces the energy sent into the channel
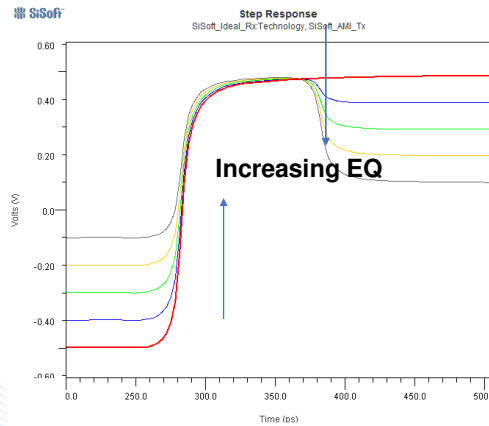
SiSoft
We Are Signal Integrity

AMI_GetWave Models Can Process Equalization Directly in Time Domain

AMI_GetWave can be used only for time domain analysis of equalization

SiSoft
We Are Signal Integrity

AMI_Init Can Return Impulse Response for Equalization

Assuming an LTI system, the impulse response can be used for both statistical and time domain analysis of equalization
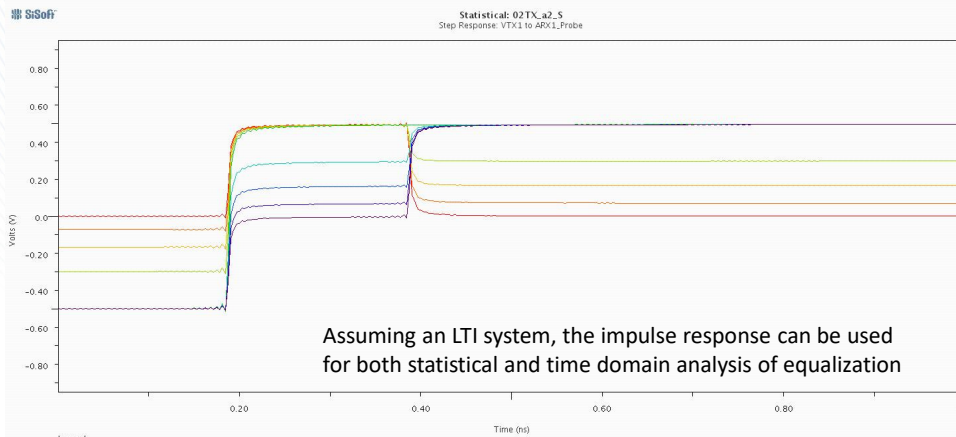
Nov 2017   IBIS-AMI Analysis Comparison   33

EQ Example: 20 inch channel, 10 Gb/s
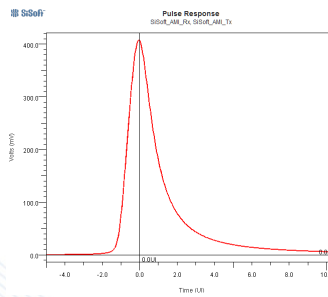
15.3 dB loss   **12+ bits of ISI**   **No EQ = No eye**
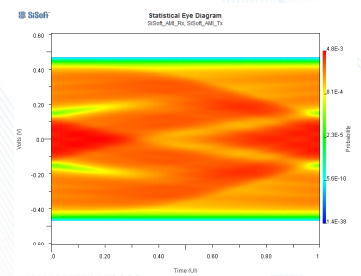
Nov 2017   IBIS-AMI Analysis Comparison   34

Sweeping the 1st Post-cursor Pulse Response

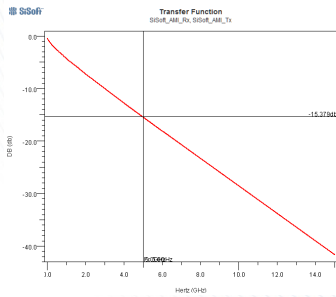| Case	Cursor	1st Post
1	1.0	0.0
2	0.9	-0.1
3	0.8	-0.2
4	0.7	-0.3
5	0.6	-0.4

- Which case will give us the best eye?
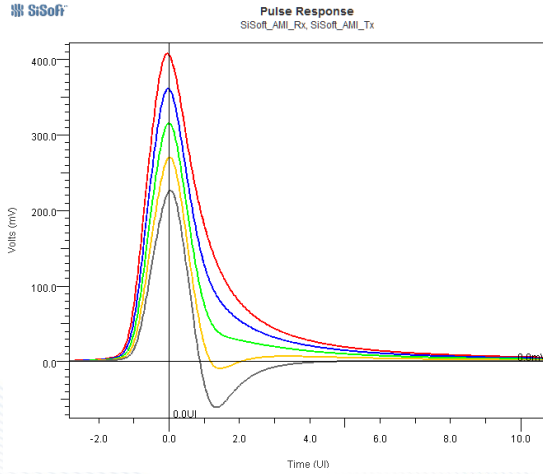
Nov 2017 IBIS-AMI Analysis Comparison 35

Using Pulse Responses to Find TX Equalization

Row	Tx:tap_filter.0	Tx:tap_filter.1	Stat Eye Height (V)
1	1	0	0
2	.9	-.1	0
3	.8	-.2	0.0706985
4	.7	-.3	0.166147
5	.6	-.4	0.126204
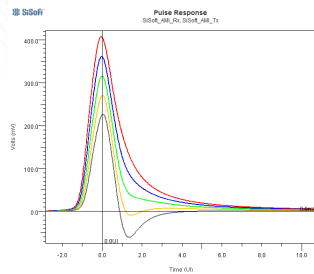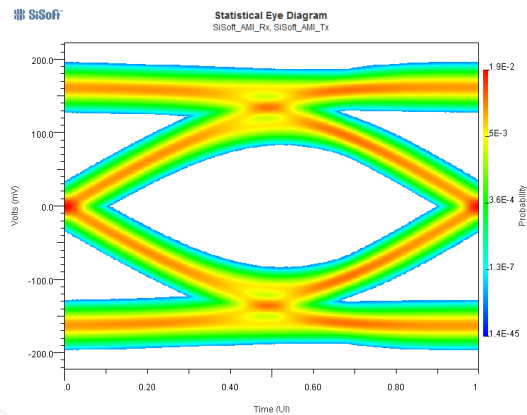
Full Time Domain analysis not required

Nov 2017 IBIS-AMI Analysis Comparison 36

AMI_GetWave Can Also Model Time-Variant Effects

RX Decision Feedback Equalizer (DFE) taps

- Adaptive corrections
  - DFE
  - CTLE
  - AGC
- Non-Linear Impairments
  - Saturation
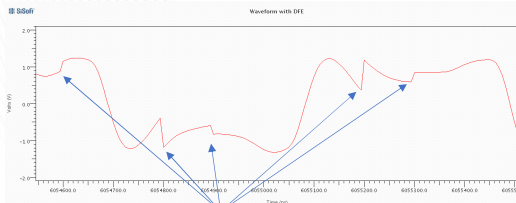
RX DFE action visible in eye diagram
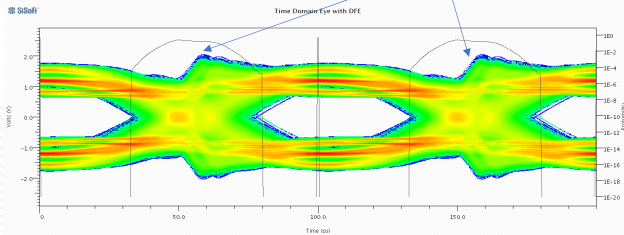
Nov 2017   IBIS-AMI Analysis Comparison   37

Using Both Time Domain and Statistical Analysis

- No single analysis method models all impairments and all corrective measures well enough

- Many helpful techniques, eg.:
  - Statistical extrapolation of time domain
  - Get adapted settings from time domain and apply to statistical (can reduce Ignore_Bits)
  - Approximate adapted DFE in RX AMI_Init
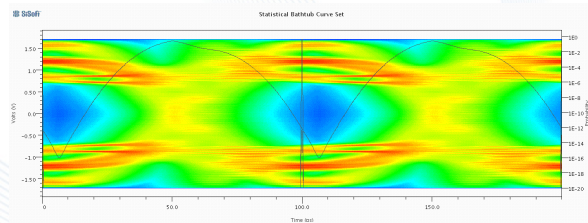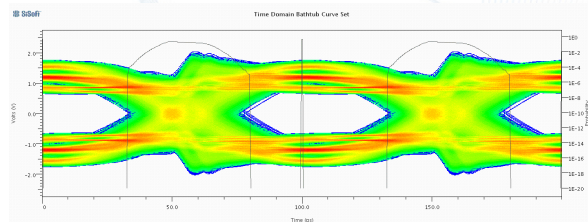
Nov 2017   IBIS-AMI Analysis Comparison   38

Conclusions

- IBIS-AMI time domain simulation with AMI_GetWave can precisely model non-linear effects such as DFE and saturation.
  - But it can be impossible to simulate enough bits in time domain to prove the low BER requirements of some technologies.
- IBIS-AMI statistical analysis can quickly evaluate very low BER.
  - But it can not precisely model time-variant effects such as DFE and saturation.
- It is good practice to use both analysis methods.

Nov 2017   IBIS-AMI Analysis Comparison   39

:::: SiSoft™
We Are Signal Integrity

Thank You

- Much content copied from:
  - *Pragmatic Signal Integrity Boot Camp*
    - Donald Telian, SiGuys
    - Michael Steinberger, SiSoft
    - Tripp Worrell, SiSoft
    - Todd Westerhoff, SiSoft
    - Graham Kus, SiSoft
    - Eric Brock, SiSoft
  - DesignCon 2017, Santa Clara, CA
- References
  [1] Anthony Sanders, Mike Resso, John D'Ambrosia, *Channel Compliance Testing Utilizing Novel Statistical Eye Methodology*, DesignCon 2004
  [2] Jeruchim, Michel C., Philip Balaban, and K. Sam Shanmugan, *Simulation of Communication Systems*, Second Edition, New York, Kluwer Academic/Plenum, 2000

Nov 2017   IBIS-AMI Analysis Comparison   40

:::: SiSoft™
We Are Signal Integrity

Characterizing and Modeling of a Linear CTE

Skipper Liang
Asian IBIS Summit
Taipei, ROC
November 15, 2017

cādence®

To Divide a RX Circuit:

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

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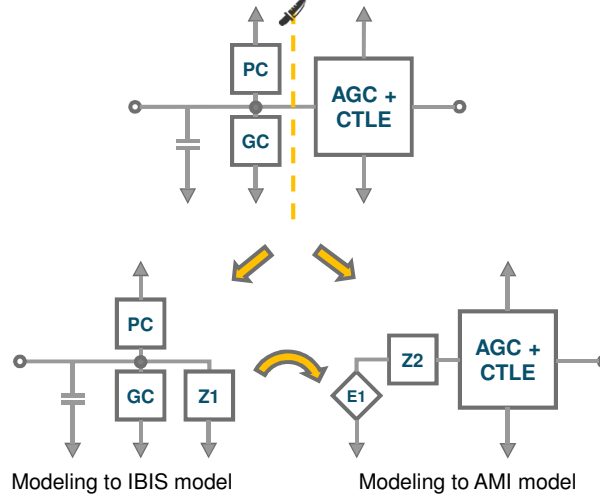
1. Question 1: What's the value of **Z1** and **Z2**?
2. Question 2: What if the whole design is described in **an encrypted netlist**?

cādence®

IBIS model
- generated from AMI generation tools

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

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

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

   The circuit should be a <u>RX</u> one composed of <u>linear</u> components.

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

   Detailed in the following pages

cādence®

Thevenin's Theorem

Thévenin's theorem
From Wikipedia, the free encyclopedia

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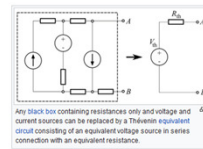
- Any linear electrical network with voltage and current sources and only resistances can be replaced at terminals A-B by an equivalent voltage source V_{th} in series connection with an equivalent resistance R_{th}.
- The equivalent voltage V_{th} is the voltage obtained at terminals A-B of the network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

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

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

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

In short:

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

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

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

cādence®

Thevenin Equivalence

Typically, an RX test topology
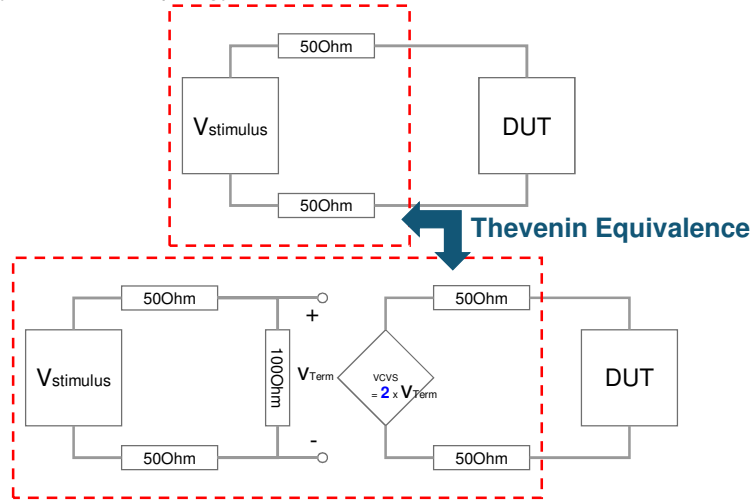
50Ohm

$V_{stimulus}$

50Ohm

DUT

Thevenin Equivalence

50Ohm

$V_{stimulus}$

100Ohm

V_{Term}

+

−

50Ohm

VCVS = **2** x V_{Term}

50Ohm

DUT

50Ohm

cādence

Thevenin Equivalence(Cont'd)

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


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

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

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

```
R100 pos neg 100

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

R50p intp intp_c 50
R50n intn intn_c 50

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

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

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

50Ohm

$V_{stimulus}$

pos

DUT

rx_comp_p

rx_comp_m

neg

50Ohm

50Ohm

pos

+

$V_{stimulus}$

100Ohm

V_{Term}

neg

−

50Ohm

intp

intn

VCVS = **2** x V_{Term}

50Ohm

Intp_c

50Ohm

Intn_c

DUT

rx_comp_p

rx_comp_m

cādence

Thevenin Equivalence(Cont'd)

Hard to tell the difference
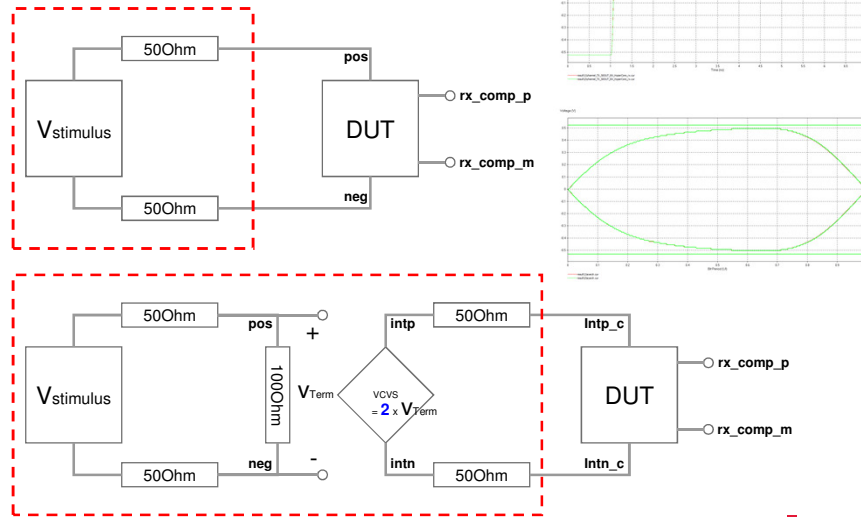
cādence®

Alternative RX

Alternative RX

Source Stimulus   Dummy Buffer   Treat as an "**Whole DUT**"

Modeling in **IBIS**   Modeling in **AMI**
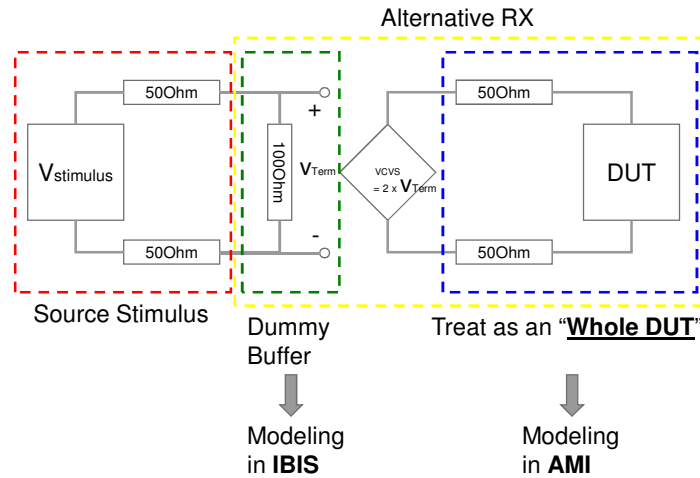
cādence®

IBIS of the Dummy Buffer

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

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

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

cdns_tx_rx.ibs
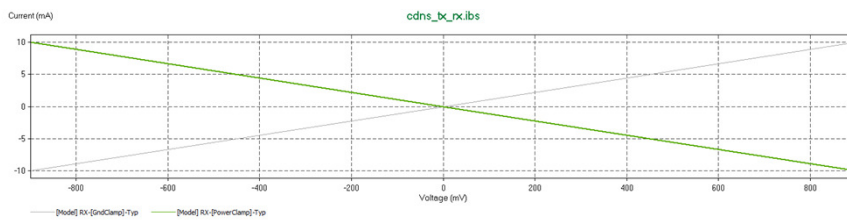
cādence®

Alternative RX for a **90Ohm** terminator

Alternative RX

Source Stimulus   Dummy Buffer   Treat as an "**Whole DUT**"

Modeling in **IBIS**   Modeling in **AMI**

Or……

cādence®

IBIS model of 100Ohm terminator

Alternative RX

Source Stimulus

Dummy Buffer

Modeling in **IBIS**
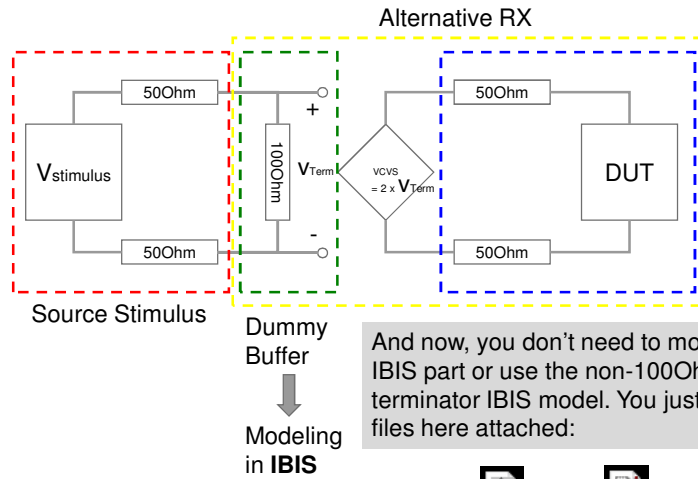
And now, you don't need to model the IBIS part or use the non-100Ohm terminator IBIS model. You just use the files here attached:
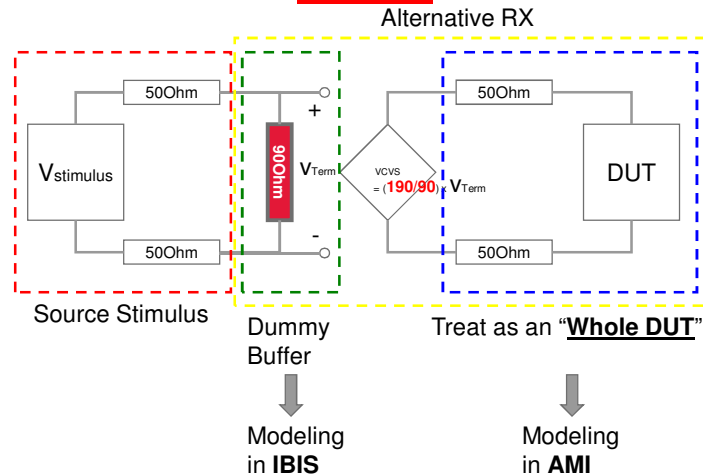
rx100term.ibs   R_100.ckt

IBIS model of 100Ohm terminator (Cont'd)

The content of these files:

rx100term.ibs

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

R_100.ckt

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

Characterizing:

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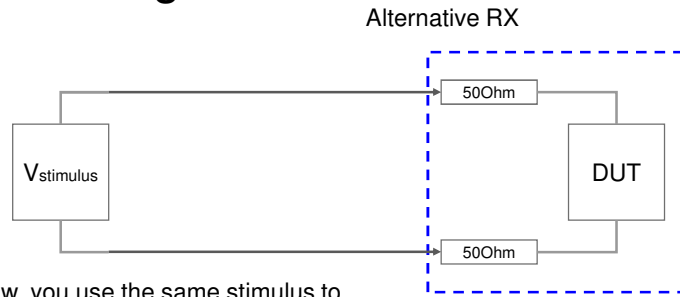
Now, you use the same stimulus to characterize the circuit inside the Blue Dashed Box, that is:

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

Which we use to characterize the channel.

Treat as an "**Whole Equalizer**"

Modeling in **AMI**

cādence®

Characterizing - Normalizing:

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

cādence®

Characterizing – Normalizing (Cont'd):

Beware!!:

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

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

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

0.57v

1.14v
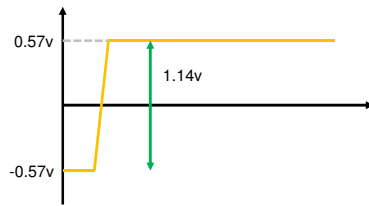
-0.57v

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

cādence

Characterizing – Normalizing (Cont'd):

The answer is:

Alternative RX

| 50Ohm | | + | | | 50Ohm | |

$V_{stimulus}$ 100Ohm V_{Term} VCVS   **2** x V_{Term} DUT

Source Stimulus
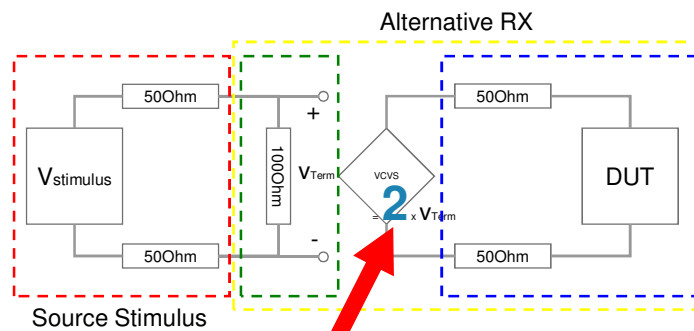
HERE!!

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

cādence

Correlation – Channel Analysis
between the IBIS-AMI and the Transistor Netlist
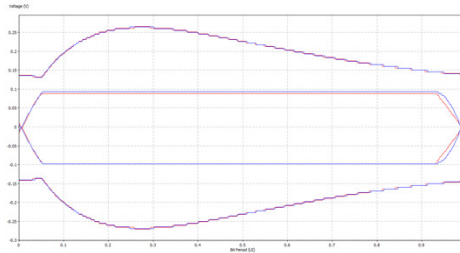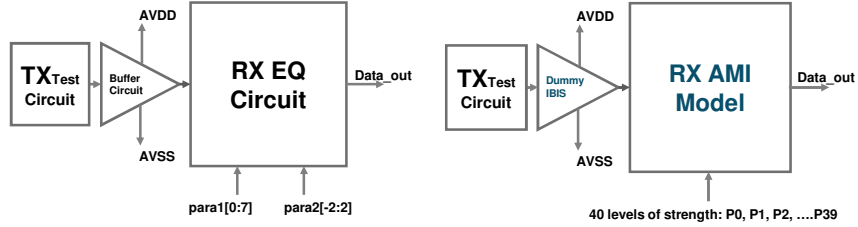
Red: Eye Contour of Transistor Netlist under Channel Analysis
Blue: Eye Contour of IBIS-AMI model under Channel Analysis

Correlation – Mid Length Channel

Correlation – Mid Length Channel (Probe 1)

Correlation – Mid Length Channel (Probe 2)

Correlation – Long Length Channel

Transient Analysis:
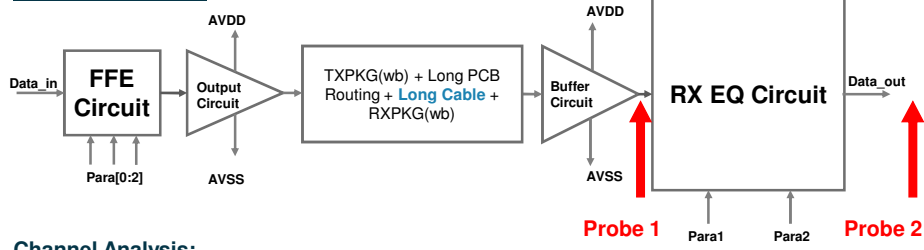
Channel Analysis:
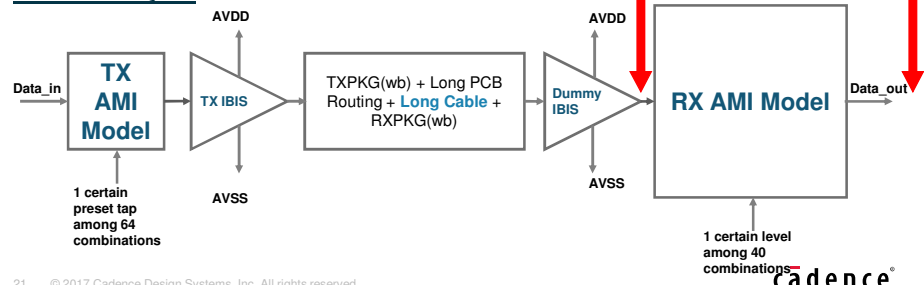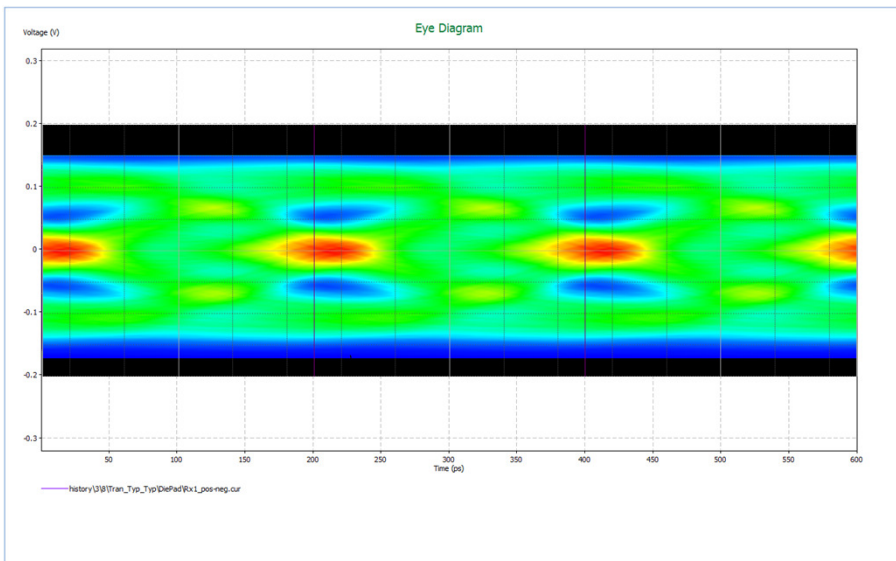
Correlation – Long Length Channel (Probe 1)

Correlation – Long Length Channel (Probe 2)
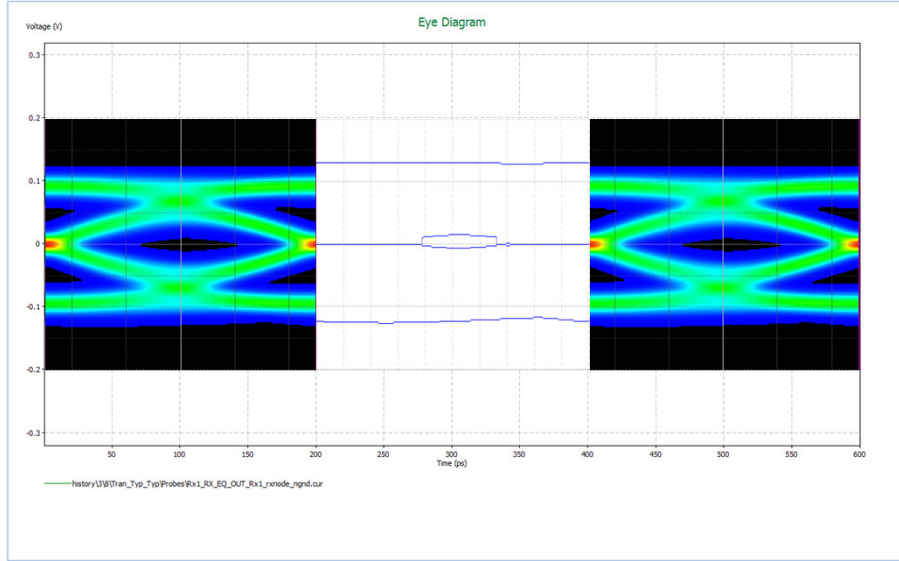
Limitation

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

Or in short, a CT**L**E which satisfies:
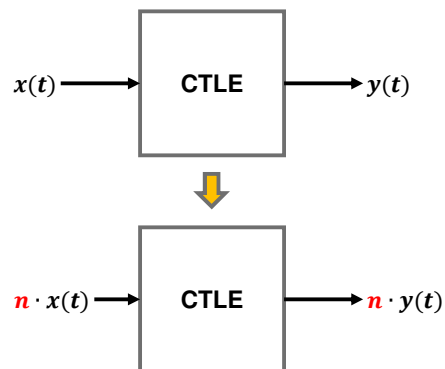
Limitation (Cont'd)

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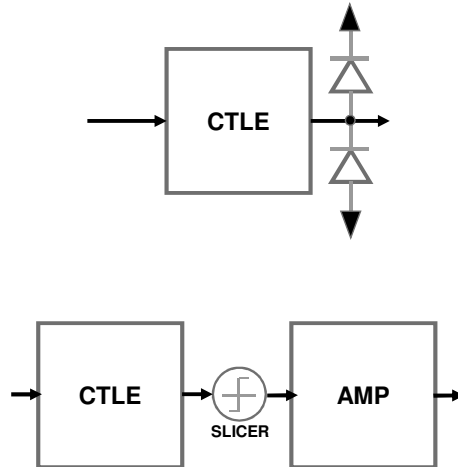
cādence

Conclusion

- Cutting/Dividing the whole design is a necessary process during RX IBIS-AMI modeling. This slides provides a method which guarantees the combination of the sub-designs so-divided is exactly equivalent to the original whole design.

- Also, the method in this slides benefits modelers that they will no longer need to model a RX IBIS model. A dummy IBIS will be used for all cases while the buffer characteristics has been modeled into the AMI model. No cutting/dividing is needed any more.

- This slides provides a method to generate IBIS-AMI simply by characterizing the V/T of the netlist – away more accurate than generating IBIS-AMI by inputting parameters values.

- However, this method is only valid while being applied to a purely linear equalizer, that is, there exists a purely linear relationship between the input and output of the equalizer.

- What else? Can a TX FFE be modeled simply by characterizing? Can an non-linear RX CTE be modeled simply by characterizing?

cādence

Correlation – Mid Length Channel

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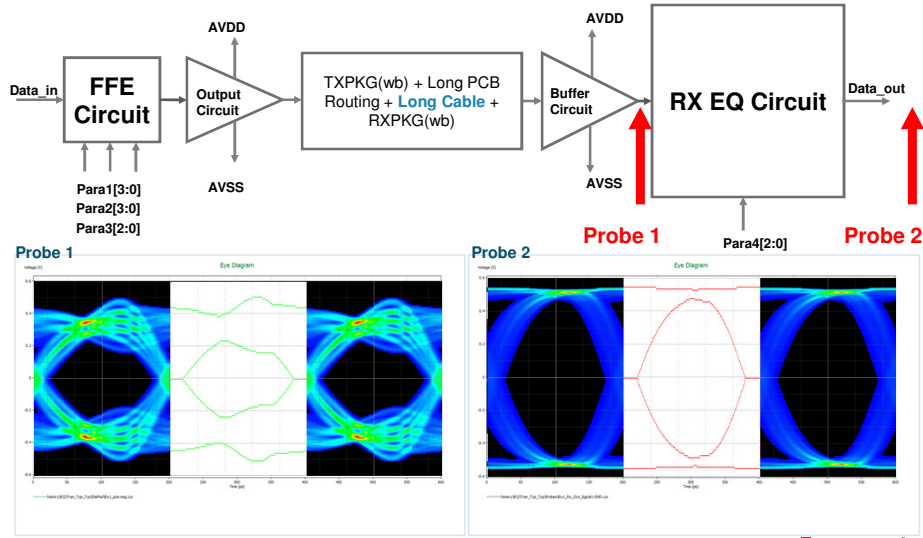
cādence

Correlation – Long Length Channel

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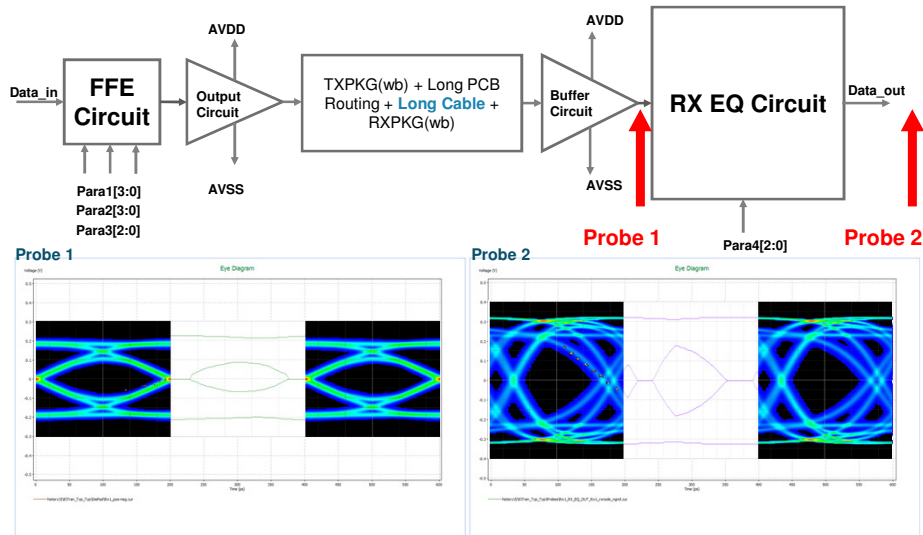
cādence

See you on IBIS Summit 2018

cadence®

Using DATA Files for IBIS-AMI Models

Lance Wang

Asian IBIS Summit
Taipei, Taiwan
November 15th, 2017

Outline

This was originally presented in 2017 DesignCon IBIS Summit, and it is with updates.

- Motivation
- Platform/OS dependent DLL, SO
- Using DATA Files for AMI models
- The advantages using DATA Files for AMI models
- Test case for AMI models with DATA Files
- Summary

Special thanks to ANSYS for providing some images in this presentation

2

Motivation

When we make and use AMI models:

- Who makes AMI executable:
  - Did I write the code that is compatible with all platform/OSs and compilers?
- Who puts everything together into an IBIS file:
  - Did I mess up 64bits and 32bits dll/so files in the IBIS file?
- Who uses AMI models:
  - Why doesn't this AMI model support my OS? Did I do anything wrong with them?

3

Motivation

- Of course, everyone will blame EDA vendors

4

Motivation

Can we make everyone a LITTLE happier?

5

Platform/OS/Compiler Dependent

- We have so many platform/OS/Compilers to cover:
  - Windows, Linux, Unix, Mac OS, etc.
  - WinXP, Win8, Win10, Ubuntu, Debian, Fedora, CentOS, Red Hat, etc. Solaris, Solaris 10, HP-UX, OSx, etc.
  - Visual Studio 8 – 11, CC, gcc, etc.
- We also have 32 bit, 64 bit and maybe 128 bit in the future to cover.

This is not all!

6

Platform/OS/Compiler Dependent

- This makes Designers/Modelers have many headaches:
  - Can my programmed source code be compiled successfully in all platforms?
  - How do I compile and debug my code in all platforms and compilers?
  - Do I need to support all of them?

7

Using DATA Files for AMI models

AMI standard functions

DATA process functions

DATA File process functions

DATA File

DLL/SO File

8
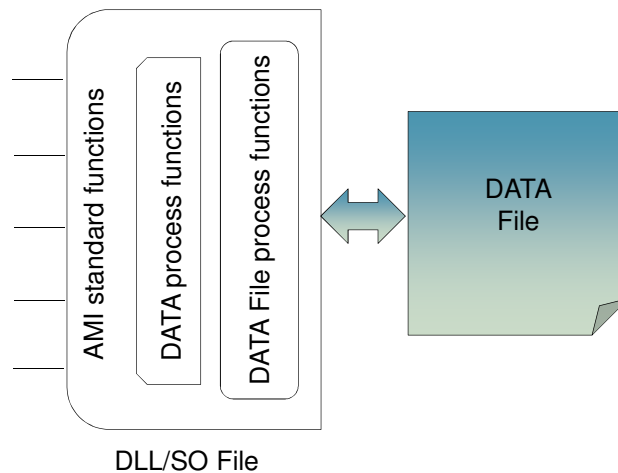
Using DATA Files for AMI models

- In DLL/SO
  - AMI standard functions
    - AMI_Init(), AMI_GetWave(), AMI_Close(). Etc.
  - DATA process functions
    - Process the DATA for algorithms
    - Take inputs from AMI standard functions
    - Apply algorithms
    - Put results back to AMI standard functions
  - DATA File process functions
    - Read DATA File
    - Decryption (Optional)

9

Using DATA Files for AMI models

- In DATA File
  - Software code, and/or
  - Behavioral data, and/or
  - Parameters, or/and
  - Other data
  - Encrypted contents (Optional)

10

Using DATA Files for AMI models

- The DLL/SO file
  - Could be made by professional software programmers for model vendors or EDA vendors
  - One time job! It can be used for many different DATA Files
- The DATA File
  - Can be created by Designers or Modelers
  - No compilation needed. Platform/OS/Compiler/Bits independent.

© 2006-2017 IO Methodology Inc.   11

The advantages using DATA Files for AMI models

- DATA File is Platform/OS/Compiler/Bit INDEPENDENT
- Although we will still have to build Platform/OS/Compiler/Bit dependent DLL/SO files to fulfill current IBIS specification, they only need to be built ONCE for model vendor and/or EDA vendor since it can be used for different DATA Files
- The DATA File will be easier to build. And it can be used for different data types that the Model or EDA vendor defines
- The DATA can be securely encrypted with advanced encryption algorithms. It is even more secured than software executables.

© 2006-2017 IO Methodology Inc.   12

Generic Test case for AMI models with DATA Files
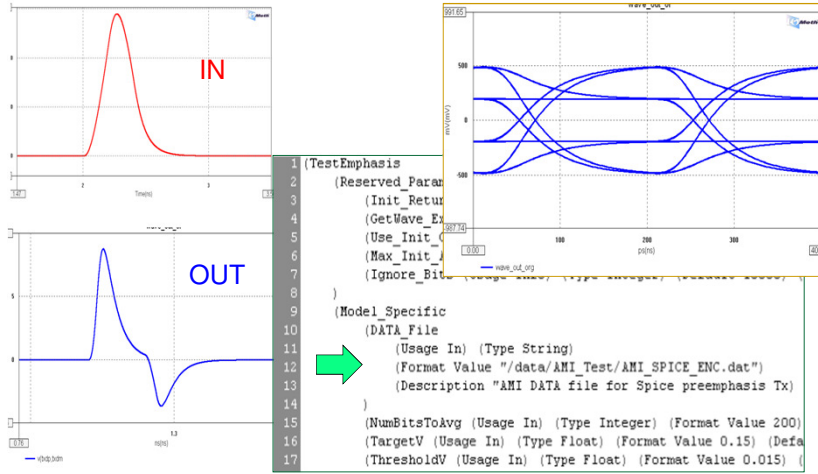
IN

OUT

```
1  (TestEmphasis
2     (Reserved_Param
3        (Init_Retu
4        (GetWave_Ex
5        (Use_Init_
6        (Max_Init_
7        (Ignore_Bit
8     )
9     (Model_Specific
10       (DATA_File
11          (Usage In) (Type String)
12          (Format Value "/data/AMI_Test/AMI_SPICE_ENC.dat")
13          (Description "AMI DATA file for Spice preemphasis Tx)
14       )
15       (NumBitsToAvg (Usage In) (Type Integer) (Format Value 200)
16       (TargetV (Usage In) (Type Float) (Format Value 0.15) (Defa
17       (ThresholdV (Usage In) (Type Float) (Format Value 0.015) (
```

13

USB3 TX Test case for AMI models with DATA Files

Channel S-parameters

USB 3 Tx   Dummy AMI Rx

io   Port1   Port3   io

Port2   Port4

inv_io   inv_io

ID=32   ID=30
output   input
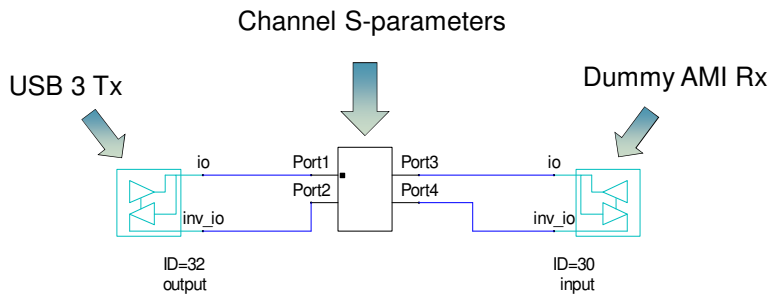
14

USB3 TX Test case for AMI models with DATA Files – AMP 105 No EQ Applied

Impulses

Bathtub

Steps
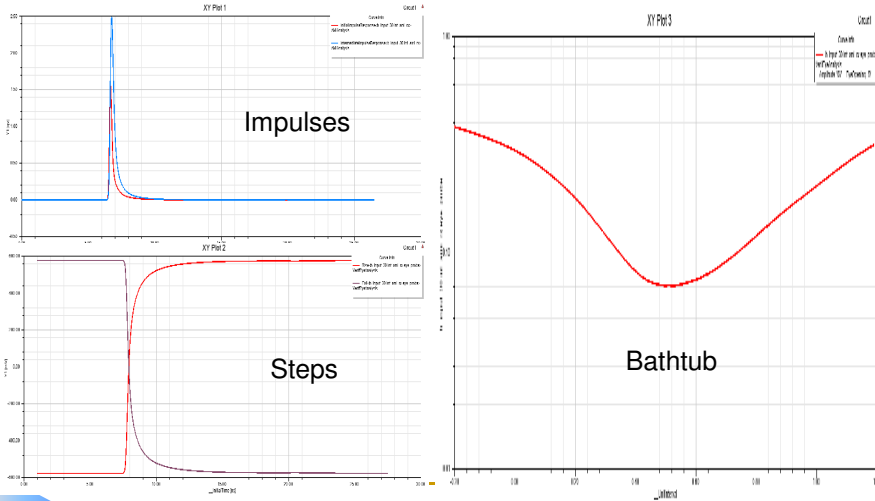
15

USB3 TX Test case for AMI models with DATA Files – AMP 105 No EQ Applied

16

USB3 TX Test case for AMI models with DATA Files – AMP 105 EQ 30 Applied

Impulses

Steps

Bathtub

© 2006-2017 IO Methodology Inc.

17

USB3 TX Test case for AMI models with DATA Files – AMP 105 EQ 30 Applied
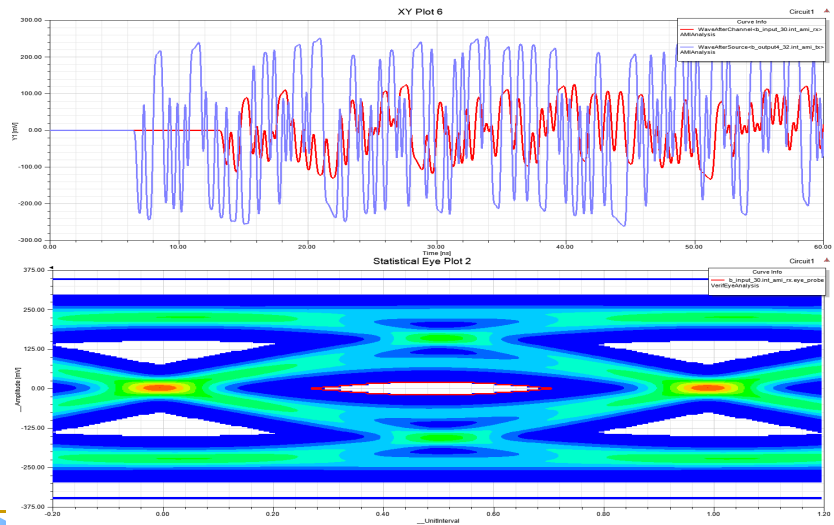
© 2006-2017 IO Methodology Inc.

18

Summary

- This presentation introduces a DATA File method for IBIS AMI models
  - The DATA File is easier to be created by modeler
  - The DATA File can be very secure through advanced encryption technologies
  - The DLL/SO will only be made by software professional ONCE and used for different DATA files
- This method is currently supported by IBIS specification and works with multiple EDA AMI tools

© 2006-2017 IO Methodology Inc.   19

IOMeth™

The Modeling Specialist

Developing the methodologies to make IBIS modeling easy and accurate

http://www.iometh.com

© 2006-2017 IO Methodology Inc.   20

IBIS-AMI Modeling Using Scripts and Spice Models

Asian IBIS Summit
Taipei, Taiwan
November 15th, 2017
(Previously presented October 18th, November 13th, 2017)

Wei-hsing Huang, SPISim
Wei-hsing.Huang@spisim.com

1

Agenda:

- Motivation
- Background
- IBIS-AMI Modeling Flow
- Modeling with Scripts
- Modeling with Spice models
- Summary
- Q & A

2

Motivation

- Channel analysis usually requires IBIS-AMI:
  - For internal analysis and/or external model release

- AMI Modeling is technically challenging
  - Requires cross domain expertise
  - Take longer to ramp-up and develop comparing to IBIS

- Can we lower the AMI modeling barriers?
  - Use scripting languages
  - Use existing spice models

3

Background 1/3

- Channel analysis: [1]
  - Mostly have stages beyond traditional IBIS (e.g. Tx/Rx EQ)
  - Analyis methodologies [2]
    - Statistical: for LTI (Linear Time Invariant) circuit, using superposition
    - Time-domain: for NLTV (Non-Linear, Time Variant) circuit, using convolution

4

Background 2/3

- AMI Model: [3]
  - Includes an .ibs, .ami and .dll/.so files
  - .ibs specifies .ami and .dll/.so files
  - .ami is a plain text file

```
[Algorithmic Model]
| The Model_type for the associated [Model] must be "I/O"
| "I/O_open_drain", "I/O_open_sink", "I/O_open_source", or "I/O_ECL".

Executable_Tx Windows_VisualStudio_32 tx_getwave.dll tx_getwave_params.ami
Executable_Tx Solaris_cc_32 libtx_getwave.so tx_getwave_params.ami
|
Executable_Rx Windows_VisualStudio_32 rx_getwave.dll rx_getwave_params.ami
Executable_Rx Solaris_cc_32 libtx_getwave.so rx_getwave_params.ami
|
[End Algorithmic Model]
```

This part is for simulator

This part is for AMI model, passed into .dll/.so as name-value pairs

```
(Rx_model
  (Reserved_Parameters
    (Resolve_Exists (Usage Info) (Type Boolean) (Value True)
      (Description "Indicates whether the executable model implements
        AMI_Resolve."))
    (Model_Name (Usage In) (Type String) (Value "ignore_me")
      (Description "IBIS model name"))
    (Rx_Receiver_Sensitivity (Usage Out) (Type Float) (Range 0.0 0.0 0.01)
      (Description "Value depends on OP_mode and data rate"))    ...

  (Model_Specific
    (Tstonefile (Usage Dep) (Type String) (Value "ignore_me.s4p")
      (Description "Rx analog model. Value depends on OP_mode"))
    (my_corner (Usage In) (Type String) (Corner "Typ" "Min" "Max")
      (Description "Informs the executable model what corner is selected by
        user"))
    (OP_mode (Usage In) (Type Integer) (List 0 1 2 3)
      (Description "Operation mode"))
    ...
  )
)
```

```
/* Init: Initialize data structure
void *modPtr = (void*) *modIMem;
IBIS_AMI_API long AMI_Init(double *htInput,
                           long    rowSize,
                           long    numAggr,
                           double  sampInt,
                           double  bitTime,
                           char    *inpParm,
                           char    **outParm,
                           void    *modIMem,
                           char    **message)
// Parse the parameter string
void *modPtr = (void*) *modIMem;
if (!modPtr && *inpParm) {
  /* parse "filtered" Model_Specific info
     original AMI file to see difference.
     Reserved_Parameters parsed by simulat
  modIPtr = (*modIMem) = SPI_AMI_Parse(inpP
}
return (modIPtr? SPI_AMI_Init(htInput, row
                              sampInt, bit
                              outParm, modI
```

Test Visualizer
Expression: inpParm
Value:
(Rx_model(Tstonefile "ignore_me.s4p")
(my_corner "Type")...

5

Background 2/3

- AMI Model: [3]
  - Includes an .ibs, .ami and .dll/.so files
  - .ibs specifies .ami and .dll/.so files
  - .ami is a plain text file

```
[Algorithmic Model]
| The Model_type for the associated [Model] must be "I/O"
| "I/O_open_drain", "I/O_open_sink", "I/O_open_source", or "I/O_ECL".

Executable_Tx Windows_VisualStudio_32 tx_getwave.dll tx_getwave_params.ami
Executable_Tx Solaris_cc_32 libtx_getwave.so tx_getwave_params.ami
|
Executable_Rx Windows_VisualStudio_32 rx_getwave.dll rx_getwave_params.ami
Executable_Rx Solaris_cc_32 libtx_getwave.so rx_getwave_params.ami
|
[End Algorithmic Model]
```

This part is for simulator

This part is for AMI model, passed into .dll/.so as name-value pairs

```
(Rx_model
  (Reserved_Parameters
    (Resolve_Exists (Usage Info) (Type Boolean) (Value True)
      (Description "Indicates whether the executable model implements
        AMI_Resolve."))
    (Model_Name (Usage In) (Type String) (Value "ignore_me")
      (Description "IBIS model name"))
    (Rx_Receiver_Sensitivity (Usage Out) (Type Float) (Range 0.0 0.0 0.01)
      (Description "Value depends on OP_mode and data rate"))    ...

  (Model_Specific
    (Tstonefile (Usage Dep) (Type String) (Value "ignore_me.s4p")
      (Description "Rx analog model. Value depends on OP_mode"))
    (my_corner (Usage In) (Type String) (Corner "Typ" "Min" "Max")
      (Description "Informs the executable model what corner is selected by
        user"))
    (OP_mode (Usage In) (Type Integer) (List 0 1 2 3)
      (Description "Operation mode"))
    ...
  )
)
```
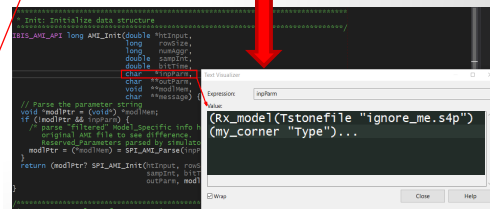
Test Visualizer
Expression: inpParm
Value:
(Rx_model(Tstonefile "ignore_me.s4p")
(my_corner "Type")...

5

Background 3/3

- AMI Model:
  - .dll/.so may implements these API functions
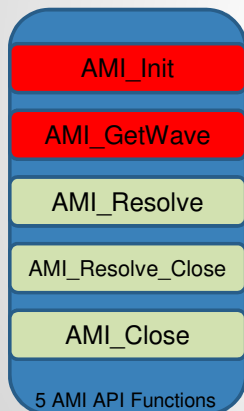
| AMI_Init |
For LTI processing, when AMI_GetWave does not exist

| AMI_GetWave |
For LTI or NLTV processing, when AMI_GetWave does exist

| AMI_Resolve |

These arrays serve as both waveform data input and output!

| AMI_Resolve_Close |

| AMI_Close |

5 AMI API Functions

```
long AMI_Init (double *impulse_matrix,
               long number_of_rows,
               long aggressors,
               double sample_interval,
               double bit_time,
               char **AMI_parameters_in,
               char **AMI_parameters_out,
               void **AMI_memory_handle,
               char **msg)
```

```
long AMI_GetWave (double *wave,
                  long wave_size,
                  double *clock_times,
                  char **AMI_parameters_out,
                  void *AMI_memory)
```

6

AMI Modeling Flow

- ## Identify model behavior(s)
  - o From mathematical equation, simulation or measurements

- ## Code the behaviors and IBIS-AMI API
  - o API implemented MUST follow IBIS spec. and in C
  - o Core processing can be in written other languages

- ## Compile and link as .dll or .so
  - o Check library dependencies, different OS bits & linux distros

  With Script and/or Spice models for core processing, this AMI model is very reuseable!

7

Modeling with Scripts: Flow

Your Existing Link Tool

AMI_Init
- Save data passed from Link Tool to a plain text file
- Your AMI_Init Script: Only deal with plain text file(s)
- Read data to form response and pass back to Link Tool

AMI_GetWave [If Get Called]
- Save data passed from Link Tool to a plain text file
- Your AMI_GetWave Script: Only deal with plain text file(s)
- Read data to form response and pass back to Link Tool

AMI_Close
- Your AMI_Close Script: actually is not needed

A Proxy Class

Similar flow for AMI_Resolve and AMI_Resolve_Close   [4]

Script path and arguments are passed via .ami file

8

Modeling with Scripts: Example

Store calling arguments into text file

Call external scripts for processing

Parse resulting text file and return
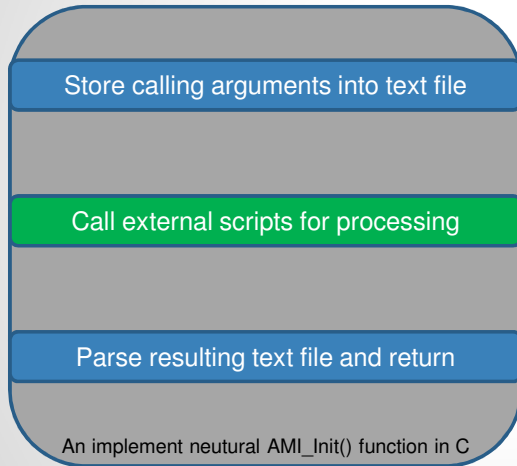
An implement neutral AMI_Init() function in C

```
clear
clc
%% file to load data from
inpFile = 'AMI_Init_Inp.txt';
%% file to save data to
outFile = 'AMI_Init_Out.txt';

%%
%% Parse waveform data passed from the simulator
%%
wave = parseInput(inpFile);

%%
%% Perform AMI_Init using Matlab
%%
sample_per_bit = floor(bit_time / sample_interval);
preTap  = -0.05;
posTap  = -0.1;
mainSig = 1 - abs(preTap) - abs(posTap);
waveInp = wave;
ht = [[1 zeros(1, sample_per_bit-1)] * preTap ...
      [1 zeros(1, sample_per_bit-1)] * mainSig ...
      [1 zeros(1, sample_per_bit-1)] * posTap];
out=conv(wave, ht);
wave = out(1:size(wave, 2));

%%
%% Store waveform data to return to the simulator
%%
storeOutput(wave, outFile)
```

9

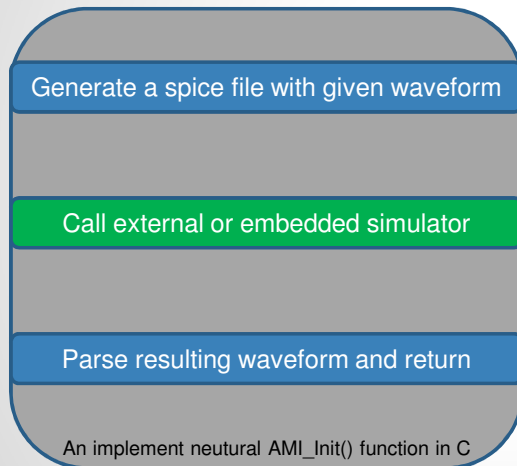# Modeling with Scripts: Consideration

- Performance and distribuability:
  - o Intepretor performance.
  - o Redistributable (license)?
  - o Does it require model user to install intrepreter?

- Consider Python! [5]
  - o SciPy, NumPy etc for numerical analysis.
  - o Embedded python: a single zip file together with AMI models.
  - o Performance and extendability.

10

Modeling with Spice: Concept

- ## Dynamically generated PWL inputs:
  - High-Z assumption
  - Simulate
    - Circuit may need to provide GND.
  - Retrieve waveform and return

No need to code circuit behavior!

11

Modeling with Spice: Flow

Your Existing Link Tool		
	AMI_Init	Translate data from Link Tool into PWL sources
		Spice executable or API simulation
		Retrieve simulation waveform for Link Tool
	AMI_GetWave [If Get Called]	Translate data from Link Tool into PWL sources
		Spice executable or API simulation
		Retrieve simulation waveform for Link Tool
	AMI_Close	Release simulator resource if needed
	A Proxy Class	

12

Modeling with Spice: Example

Generate a spice file with given waveform

Call external or embedded simulator

Parse resulting waveform and return

An implement neutral AMI_Init() function in C

```
A Spice AMI deck
.tran 1e-11 2.001e-08
.probe VOUT=par('V(OUTP)-V(OUTN)')
.probe VINP=par('V(INPP)-V(INPN)')
.option noinit acct
.option rshunt=1E12
.option method=gear
.option RELTOL=0.01
.option ABSTOL=1N VNTOL=1M
.option ITL4=500
* The following two lines are for HSpice compatibility
.OPTION POST_VERSION=9601
.OPTION PROBE POST
* INPUT
VINP INPP INPN PWL(
+ 0 0.5
+ 1e-11 0.5
...
+ 1.998e-08 -0.499034
+ 1.999e-08 -0.499035
+ 2e-08 -0.499035)
* AMI Subckt
.INC D:\WorkDir\CTLE.sp
XAMI INPP INPN OUTP OUTN RC_CTLE   CTLE_R1=0.01234 CTLE_C1=5.678E-12
* High Z Load
RHIZ OUTP OUTN 1E6
.end
```

13

Modeling with Spice: Consideration

- Performance and distribuability:
  - o Availability of device models?
  - o Redistributable (license)?
  - o Does model user need specific simulator?

- Consider open source simulator!
  - o NgSpice, QUCS etc all supports API/Shared library [6]
  - o The AMI model is basically a circuit simulator
    - ▪ Implement once, use many times!
  - o Performance vs Accuracy

14

Summary:

- AMI model using scripts and spice circuit:
  - Doable! (Has been implemented! Example included.)
  - Can reduce AMI modeling time significantly
  - Can serve as an intermediate step toward full C/C++ implementation.

- Considerations:
  - Performance:
    - Not a concern if only AMI_Init is needed (called only once)
  - Model release:
    - Can the model be distributed and used easily?
  - A simple wrapper IBIS-AMI model is needed as a proxy.

15

References:

1. **High-speed Links Circuits and Systems** http://www.ece.tamu.edu/~spalermo/ecen720.html
2. **Simulating High-Speed Serial Channels with IBIS-AMI Models**
   http://literature.cdn.keysight.com/litweb/pdf/5990-9111EN.pdf?id=2095655
3. **IBIS V6.1 Spec. Section 10** http://ibis.org/ver6.1/
4. **AMI Analysis Using a Proxy Class** http://ibis.org/summits/feb17/
5. **Embedding python in another application** https://docs.python.org/3/extending/embedding.html
6. **NgSPice as a shared library** http://ngspice.sourceforge.net/shared.html

16