More experiences
with IBIS-AMI models

Eckhard Lenski
European IBIS Summit
Sorrento, Italy
May 16, 2012
More experiences with IBIS-AMI models

Why IBIS-AMI

Differences IBIS vs IBIS-AMI

Basics IBIS-AMI

IBIS-AMI models

summary
Higher datarates

Classic SI
- Rise fall times
- Overshoots

BER, ISI
- UI
- Eye opening
- Jitter
Higher datarates

10Gbps and more

- PCI-Express 2, 3, 4
- Serial ATA 2, 3
- USB 3
- XAUI
- CEI 11G
- Emerging 25 / 40 / 100 Gbps standards

Above 1GHz

- ISI (Inter Symbol Interference)
- Attenuation
- Jitter
- Influence of vias
- Influence of connectors
## Analysis time

### Estimated time for determination of BER

Eg 6.375Gbps

<table>
<thead>
<tr>
<th>BER</th>
<th>Measurement</th>
<th>Spice</th>
<th>Ibis-ami</th>
</tr>
</thead>
<tbody>
<tr>
<td>3e-12</td>
<td>1 minute</td>
<td>hour</td>
<td>seconds</td>
</tr>
<tr>
<td>1e-15</td>
<td>2 days</td>
<td>days</td>
<td>a few minutes</td>
</tr>
<tr>
<td>6e-18</td>
<td>1 year</td>
<td>more days</td>
<td>More minutes</td>
</tr>
</tbody>
</table>

Source: measurement data from *Bogatin Altera high speed seminar 2006*
More experiences with IBIS-AMI models

Why IBIS-AMI

Differences IBIS vs IBIS-AMI

IBIS-AMI models

Basics IBIS-AMI

summary
IBIS

Driver
- U-I curves
- U-t curves

Receiver
- U-I curves
- Params (vih, vinl)
## Modeling of high speed features

<table>
<thead>
<tr>
<th>Tx &amp; Rx with</th>
<th>In IBIS</th>
<th>IBIS-AMI</th>
<th>Algorithmic Modeling Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>equalization</td>
<td>-</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Pre-emphasis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 tap</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3 tap</td>
<td>-</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FFE</td>
<td>-</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DFE</td>
<td>-</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>jitter</td>
<td>-</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
IBIS-Ami (Algorithmic modeling interface)

Driver
- FFE
- Swing
- Taps / Pre-emphasis

Receiver
- DFE
- Taps
- Jitter
- CDR
- Equalization

Algorithmic Tx model

Classic ibis Tx model

Algorithmic Rx model

Classic ibis Rx model
platform

IBIS
- Runs on every platform
- Same model

IBIS-AMI
- Platform dependant
  - Linux
  - windows
- 32 / 64 bit
- different model
  - Model.so (linux)
  - Model.dll (windows)
- documentation file necessary
More experiences with IBIS-AMI models

Summary

IBIS-AMI models

Basics IBIS-AMI

Why IBIS-AMI

Differences IBIS vs IBIS-AMI
IBIS-AMI keyword

[Model] Tx
- Model_type Open_sink
- [Voltage Range] 1.2 1.14 1.26
- C_comp 1.00p 0.95p 1.05p
- [Pullup]

[Algorithmic Model]
- Executable Windows_VisualStudio_32 NSNTx.dll NSNTx.ami
- Executable Linux_gcc3.2.3_32 NSNlibTx.so NSNlibTx.ami
- [End Algorithmic Model]
Executable
Windows_VisualStudio_32
NSNTx.dll
NSNTx.ami

- Executable
- Windows_VisualStudio_32
- NSNTx.dll
- NSNTx.ami

- Subparameter
- Platform_Compiler_Bits
- name of shared library file
- Description file for params
IBIS-AMI principle

Driver

RX
Analog
Algorithmic

RX
Algorithmic

Algorithmic Rx model

TX
Algorithmic

TX
Analog

Classic ibis Tx model

PWB + Conn. + Via

Z0

Algorithmic Tx model

PKG

PKG

Public
Signal processing in the driver before the final output stage

- equalization/pre-emphasis (FFE feed forward equalizer)
- encoding (8b/10b)
Signal processing in the receiver after the first input stage

equalization (e.g. DFE decision feedback equalizer)
clock recovery
Impulse response representing the analog channel

Driver (TX) final output stage
Channel
Receiver (RX) input stage

Channel must be LTI (Linear and Time Invariant)
IBIS-AMI channel elements

**Elements**
- Tx analog
- Rx analog
- Pkg
- Channel, T-line
- Via
- Connector

**Models**
- Classic ibis driver
- Classic ibis receiver
- S-params
- W-element, S-params
- S-params
- S-params
Modeling of high speed features

Question

- Which simulation
  - used the good
  - used the bad
- S parameter model?
# IBIS-AMI driver / receiver elements

## TX
- AMI_INIT
- AMI_Getwave
- AMI_close

## RX
- AMI_INIT
- AMI_Getwave
- AMI_close

## TX/RX Ami
- **AMI_INIT**
  - Used for statistical analysis, setting for getwave parameters, LTI
- **AMI_Getwave**
  - Timing analysis, non LTI behavior
- **AMI_close**
  - Clear memory, etc.
IBIS-AMI parameters

General parameters
- Init_Returns_Impulse
- Getwave_Exists
- ( Use_Init_Outputs )
- Ignore_Bits
- Max_Init_Aggressors
- ......

Specific model parameters
- Tap settings
- Equalization settings
- Driver strength
- PVT conditions
- Gain
- ......
Impulse response

- Step response

- Impulse response

- Each topology has its own impulse response
Selection of model specific parameters

Specific model parameters

- Tap 0
  - Value: 1

- Tap 1
  - Increment: 0, -2, 3, 0.2

- Equalization settings
  - Range: 0, 0.5, 1

- Driver strength
  - list: 200mv, 400mv, 600mv, 800mv
IBIS-Ami first results

Model settings:
- Swing — 0 ... 63
- pretap — 0 ... 32
- posttap — 0 ... 63
IBIS-AMI first results

Eye diagram

BER

Public
[Model] ALPHA

- Model_type: Open_sink

- [Voltage Range] 0.03V  0.02V  0.04V

- C_comp 1.00p  0.95p  1.05p

- [Pullup]

- [Algorithmic Model]
  - Executable Windows_VisualStudio_32 NSNTx.dll NSNTx.ami
  - Executable Linux_gcc3.2.3_32 NSNlibTx.so NSNTx.ami

- [End Algorithmic Model]
[Model] BETA

- Model_type: Output
- [Voltage Range] 1.8V  1.75V  1.85V
- C_comp 0.80p  0.65p  0.95p
- [Pullup]

- [Algorithmic Model]
  - Executable Windows_VisualStudio_32 NSNTx.dll NSNTx.ami
  - Executable Linux_gcc3.2.3_32 NSNlibTx.so NSNTx.ami
- [End Algorithmic Model]

- Model needs mathematical program, but with specific release

- Model runs on windows XP, but not on Win7
[Model] GAMMA

- Model_type Open_sink
- [Voltage Range] 1.0V 0.92V 1.08V
- C_comp 0.50p 0.45p 0.55p
- [Pullup]

- [Algorithmic Model]
  - Executable Windows_VisualStudio_32 NSNTx.dll NSNTx.ami
  - Executable Linux_gcc3.2.3_32 NSNlibTx.so NSNTx.ami
- [End Algorithmic Model]

- Request for package models
- Answer: package model is inside dll,

- According IBIS-AMI spec this is not valid
- Package is a part of the channel
[Model]  DELTA  “not completed”  .ami  file

- Model_type: Open_sink
- [Voltage Range]: 0.03V  0.02V  0.04V
- C_comp: 1.00p  0.95p  1.05p
- [Pullup]

[Algorithmic Model]
- Executable: Windows_VisualStudio_32  NSNTx.dll  NSNTx.ami
- Executable: Linux_gcc3.2.3_32  NSNlibTx.so  NSNTx.ami

**IBIS-Ami file model10G_file.ami**

(Model_Specific)

- (DRIVE (Usage In) (Type Integer)
- (Format Value 0 ) (Default 0) (Description “DRIVE”)

**IBIS-AMI description file model10G-ami.pdf**

(DRIVE)  TX Swing  TX Pre-emphasis

<table>
<thead>
<tr>
<th>0</th>
<th>200mv</th>
<th>3db</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>400mv</td>
<td>5db</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>800mv</td>
<td>8db</td>
</tr>
</tbody>
</table>
[Model]  DELTA    upgraded    .ami file

- Model_type: Open_sink
- Voltage Range: 0.03V 0.02V 0.04V
- C_comp: 1.00p 0.95p 1.05p
- Pullup

[Algorithmic Model]
- Executable: Windows_VisualStudio_32 NSNTx.dll NSNTx.ami
- Executable: Linux_gcc3.2.3_32 NSNlibTx.so NSNTx.ami

IBIS-Ami file  model10G_file.ami

(Model_Specific)

(DRIVE (Usage In) (Type Integer)

(Format List 0 10 ... 80 ) (Default 0) (Description "DRIVE")

IBIS-AMI  description file  model10G-ami.pdf

(DRIVE) TX Swing TX Pre-emphasis

0 200mv 3db
10 400mv 5db

...........

80 800mv 8db
[Model]  EPSILON & [Model]  OMEGA

- **Model_type**  Open_sink

- **[Algorithmic Model]**

  - **Executable**  Linux gcc3.2.3_32  NSNlibTx.so
    - **NSNTx.ami**
    - **[End Algorithmic Model]**

- **Model_type**  Input

- **[Algorithmic Model]**

  - **Executable**  Windows VisualStudio_32  NSNlibRx.dll
    - **NSNRx.ami**
    - **[End Algorithmic Model]**
[Model] ZETA

- **Model_type** Open_sink
- **[Voltage Range]** 0.03V 0.02V 0.04V
- **C_comp** 1.00p 0.95p 1.05p

- **[Model Selector]**
  - Swing10
  - .......
  - Swing630

- **[Algorithmic Model]**
  - **Executable** Windows_VisualStudio_32 NSNTx.dll NSNTx.ami
    - (TX_swng **(Range 310 10 630)** (Usage In)(Type Integer)(Default 310)
      (Description "Transmitter swing"))

- **Executable** Linux_gcc3.2.3_32 NSNlibTx.so NSNTx.ami

**ibis file** Nsn.ibs

**ibis ami file** NsnTx.ami

Values must be aligned
IBISCHK5 results in 2 errors:

- E4630 - Parameter Format is mandatory for Rx_Clock_PDF
- E4630 - Parameter Description is mandatory for Rx_Clock_PDF

Executable Windows_VisualStudio_32 NSNTx.dll NSNTx.ami

(Reserved_Parameters
(Rx_Clock_PDF (Usage Info) (Type Float) (Format Table
  (Labels Row_No Time_UI Density)
    (-3 -1.2 0.1)
    (3 1.17E-001 0.01)
  ) | End Table
  ) | End Rx_Clock_PDF
 ) | End Reserved_Parameters
(Reserved_Parameters

(Rx_Clock_PDF (Usage Info) (Type Float) (Format Table

(Labels Row_No Time_UI Density)

(-30  -1.17E-001  0)
( 30  1.17E-001  0)
)

(Description "Receiver clock pdf is in table format."

)

) | End Rx_Clock_PDF

) | End Reserved_Parameters

- E4630 - Parameter Format is mandatory for Rx_Clock_PDF

- E4630 - Parameter Description is mandatory for Rx_Clock_PDF

Bug in ibis parser

Error in ami file
[Model] THETA  no documentation supported

IBIS-Ami file  model8G_file.ami

(Model_Specific
 (EQUAL (Usage In) (Type Integer) (Format List 0 ..... 8 ) (Default 0)

Labels “ 0 001  0.0db “

......

“ 8 110   9.5db : max “)

(Description “EQUAL  post tap settings ”)

---

Data sheet model8G.txt

Equalization post-tap :

0001  no equalization

........

0110  75%

---

Public
More experiences with IBIS-AMI models

summary

IBIS-AMI models

Basics IBIS-AMI

Differences IBIS vs IBIS-AMI

Why IBIS-AMI
Summary

- For > 5Gbps, IBIS, SPICE is too slow
- IBIS-AMI is a good solution for BER < 1e-12
- Increasing model support from IC-vendors and Tool vendors
- IBIS forum is working on enhancements
- IBIS-AMI is challenging
Thank you

Questions?