IBIS-AMI Analog Modeling and Much Needed Improvements for IBIS

IBIS Summit, DesignCon, February 3, 2011
Santa Clara, CA

Arpad Muranyi

© Mentor Graphics Corp., 2011, Reuse by written permission only. All rights reserved.
IBIS-AMI Analog Modeling and Much Needed Improvements for IBIS

IBIS Summit, DesignCon, February 3, 2011
Santa Clara, CA

1. My plea from DesignCon 2010
2. The golden opportunity and complete solution
3. The IBIS-AMI analog model
4. IBIS-AMI analog model examples with IBIS-ISS
5. A simple package model example with IBIS-ISS
6. Backup - An example for Figure 12 using IBIS-ISS
My plea from DesignCon 2010

This was the last slide in my DesignCon 2010 presentation

- AMI gave new life to IBIS, but we still need to address some serious shortcomings in the “legacy” portions of the IBIS specification to make AMI work without any EDA tool dependent, proprietary modeling solutions
- This situation needs immediate attention, and urgent solutions
- Let’s address these problems in a timely manner and provide a complete IBIS-AMI modeling solution in IBIS v5.2
The golden opportunity

- Version 1.0 of the IBIS Interconnect SPICE Subcircuit Specification (IBIS-ISS) has been submitted to the IBIS Open Forum today
  - IBIS-ISS standardizes the passive elements of HSPICE® with permission from Synopsys
  - IBIS-ISS contains R, L, C, W-elements, S-parameter support, some controlled sources, etc…
  - it supports subcircuits with parameters, string parameters (for S-parameter file names, for example), etc…

- IBIS-ISS is a subcircuit definition language, not a complete SPICE language
  - IBIS-ISS subcircuits are instantiated by a top level netlist or higher level subcircuits
  - simulator controls (.options) or post processing statements (.print), etc… are not included

- IBIS already supports subcircuits under [External ***]
  - but Berkeley-SPICE is severely limited, useless for most modern modeling needs
  - adding IBIS-ISS to [External ***] would improve the buffer modeling capabilities of IBIS
  - adding IBIS-ISS to [Define Package Model] would improve the package modeling of IBIS
The AMI aspect

- Algorithmic modeling relies on accurate channel characterization simulations using IBIS models

```
GENERAL ASSUMPTIONS:
This proposal breaks SERDES device modeling into two parts - electrical and algorithmic. The combination of the transmitter’s analog back-end, the serial channel and the receiver’s analog front-end are assumed to be linear and time invariant. There is no limitation that the equalization has to be linear and time invariant. The “analog” portion of the channel is characterized by means of an impulse response leveraging the pre-existing IBIS standard for device models.
```

- The Opal document (and BIRD 122) proposes a solution for the analog buffer modeling problems, but strictly for AMI use
  - normal IBIS buffer and package modeling are not addressed by these proposals

- IBIS-ISS can describe everything that is being proposed in Opal and BIRD 122
The complete solution

- Linking IBIS-ISS with IBIS could provide significant improvements for legacy IBIS modeling as well as for IBIS-AMI analog buffer modeling
- **BIRD 116** proposes the addition of IBIS-ISS to the existing IBIS [External Model] and [External Circuit] keywords
  - [External Model] is more suitable for AMI purposes because
    - a direct link to AMI parameters can be established through the [Algorithmic Model] keyword
    - it supports true differential buffer modeling
    - works together with [Model Selector]
- **BIRD 117** and **118** propose additional flexibilities for parameterizing the [External *] keywords
  - this is useful for passing AMI parameters from the .ami file to the IBIS-ISS subcircuit
- **BIRD 125** proposes the addition of IBIS-ISS to the [Define Package Model] keyword
A short IBIS-AMI refresher

IBIS-AMI simulations consist of three main pieces

1) An analog model of the channel
   - interconnect (PCB traces, connectors, cables, etc…)
     ■ these come in the form of S-parameter files, W-elements, RLC circuits
   - analog models of the Tx and Rx devices and their package models
     ■ these would theoretically come in .ibs and .pkg files
   - the result of the analog simulation is used to generate the impulse response of the channel which is the basic input to the AMI models

2) The algorithmic Tx and Rx models
   - these are the executable binary (.dll) files

3) The algorithmic model parameter file
   - these are the .ami files
The AMI analog model

finite input impedance must be guaranteed by an ideal isolation amplifier in the netlist or an ideal isolation amplifier incorporated within the Touchstone S-parameter data (model parameters may be defined in the .ami file)

zero output impedance must be guaranteed by an ideal isolation amplifier in the netlist or an ideal isolation amplifier incorporated within the Touchstone S-parameter data (model parameters may be defined in the .ami file)

PWL voltage sources in EDA tool (parameters in .ami file)
Some technical details

- The PWL stimulus voltage sources are NOT part of the analog model, these will be provided by the EDA tool
  - their parameters (Voh, Vol, tr, tf) are defined in the .ami file using predefined Reserved_Parameters (Usage Info)
- The analog Tx models must have an infinite input impedance, and the analog Rx models must have a zero output impedance at the algorithmic/analog boundary
  - this may be achieved by using an ideal isolation amplifier voltage controlled voltage source (E-element) in the circuit topology, or by incorporating an isolation amplifier within a Touchstone S-parameter file that models this boundary
  - model parameters may be defined in the .ami file (as Reserved or Model Specific parameters)
- The difference waveform is generated by the EDA tool from the differential output of the Rx model
- This difference waveform is further processed by the EDA tool to generate the channel’s impulse response for the algorithmic model
True differential models in IBIS

A differential stimulus may be generated by placing two D_to_A adapters between D_drive and the [External Model].

The A_to_D converter may be connected to the output of the Rx buffer if it is a 4-port model.

Figure 11: Example SPICE, Verilog-A(AMS) or VHDL-A(AMS) implementation of a true differential buffer

In EDA tool
In [External Model]
**Legacy IBIS file:**

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example of an analog AMI Tx model using [External Model] and an IBIS-ISS S-parameter:</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
</tr>
</tbody>
</table>

[Model] ISS_Diff_Tx
Model_type Output_diff
Rref_diff = 100

[Voltage Range] 1.0 NA NA

[Ramp]
\[dV/dt_r\] 0.6/40p NA NA
\[dV/dt_f\] 0.6/40p NA NA

[External Model]
Language ISS

<table>
<thead>
<tr>
<th>Corner</th>
<th>corner_name</th>
<th>file_name</th>
<th>circuit_name (.subckt name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typ</td>
<td>AMIdriver.cir AMI_Sdrv</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters TSFile = AMIfile(Tstonefile)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List of converter parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converter_Parameters VloP AMIfile(Vol)</td>
</tr>
<tr>
<td>Converter_Parameters VhiP AMIfile(Voh)</td>
</tr>
<tr>
<td>Converter_Parameters VloN AMIfile(Voh)</td>
</tr>
<tr>
<td>Converter_Parameters VhiN AMIfile(Voh)</td>
</tr>
<tr>
<td>Converter_Parameters Tfa AMIfile(Trf)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ports List of port names (in same order as in SPICE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_signal_pos A_signal_neg my_driveP my_driveN my_ref A gnd</td>
</tr>
<tr>
<td>D_to_A D_port port1 port2 vlow vhigh tri fall corner name</td>
</tr>
<tr>
<td>D_to_A D_drive my_driveP my_ref VloP VhiP Tfa Tri Typ</td>
</tr>
<tr>
<td>D_to_A D_drive my_driveN my_ref VloN VhiN Tfa Tri Typ</td>
</tr>
</tbody>
</table>

[End External Model]

[Algorithmic Model]
Executable Windows_VisualStudio_32 MentorTx.dll MentorTx.ami

[End Algorithmic Model]

---

**EDA tool GUI lets user select “one of many”. Dependency Table may also affect what this GUI does.**

**Tx.ami file:**

```plaintext
(Tstonefile Usage Info)(Type String)
(Corner "NC.s4p" "WC.s4p" "BC.s4p")
(Description "Driver on-die S-parameter file")

(Voh Usage Info)(Value 0.9)(Type Float)
(Description "Output open circuit high voltage")

(Vol Usage Info)(Value 0.0)(Type Float)
(Description "Output open circuit low voltage")

(Trf Usage Info)(Value 40e-12)(Type Float)
(Description "20%-80% output rise time")
```

**IBIS-ISS subcircuit:**

```plaintext
*******************************************************************************
.SUBCKT AMI_Sdrv A_signal_pos A_signal_neg my_driveP my_driveN my_ref +
     TSFile="TouchstoneFileName.s4p"
Sdriver my_driveP A_signal_pos A_signal_neg my_driveP my_driveN my_ref +
     MNAME=TSFile +
     [FBASE = base_frequency] [FMAX=maximum_frequency]
*******************************************************************************
.ends
```

---

**IBIS-AMI Analog Modeling and Much Needed Improvements for IBIS**
**Legacy IBIS file:**

---

Example of an analog AMI Rx model using [External Model] and an IBIS-ISS circuit:

**[Model] ISS_Diff_Rx**

Model type Input_diff

Voltage Range 1.0 NA NA

**[External Model]**

Language ISS

Corner corner_name file_name circuit_name (.subckt name)

<table>
<thead>
<tr>
<th>Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rt_H</td>
<td>1e+6</td>
</tr>
<tr>
<td>Rt_L</td>
<td>1e+6</td>
</tr>
<tr>
<td>Rd</td>
<td>1e+6</td>
</tr>
<tr>
<td>Cc_H</td>
<td>0</td>
</tr>
<tr>
<td>Cc_L</td>
<td>0</td>
</tr>
<tr>
<td>Cd</td>
<td>0</td>
</tr>
<tr>
<td>Vt</td>
<td>0</td>
</tr>
</tbody>
</table>

**Converter Parameters**

Vlo = -0.05
Vhi = 0.05

<table>
<thead>
<tr>
<th>Ports List of port names (in same order as in SPICE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_signal_pos A_signal_neg my_rcv_H my_rcv_L A_gnd</td>
</tr>
<tr>
<td>D_to A_d port port1 port2 vlow vhigh corner_name</td>
</tr>
<tr>
<td>A_to D D_receive my_rcv_H my_rcv_L Vlo Vhi Typ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>End External Model</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>[Algorithmic Model]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executable Windows_VisualStudio_32 MentorTx.dll MentorRx.ami</td>
</tr>
</tbody>
</table>

| [End Algorithmic Model] |

**Rx.ami file:**

(Rt (Usage Info) (Value 47.75) (Type Float) (Description "Single-ended termination resistance")
(Rd (Usage Info) (Value 99.75) (Type Float) (Description "Differential termination resistance")
(Cc (Usage Info) (Value 0.5e-12) (Type Float) (Default 0.5e-12) (Description "Input Capacitance")

**IBIS-ISS subcircuit:**

*******************************************************************************
.SUBCKT AMI_RCrcv A_signal_pos A_signal_neg my_rcv_H my_rcv_L my_ref
  + Rt_H = 1e+6
  + Rt_L = 1e+6
  + Rd = 1e+6
  + Cc_H = 0
  + Cc_L = 0
  + Cd = 0
  + Vt = 0

Cc_H A_signal_pos my_ref C=Cc_H
Cc_L A_signal_neg my_ref C=Cc_L
Cd A_signal_pos A_signal_neg C=Cd
Rt_H A_signal_pos my_vtt R=Rt_H
Rt_L A_signal_neg my_vtt R=Rt_L
Rd A_signal_pos A_signal_neg R=Rd
Vvtt my_vtt my_ref DC=Vt
E_H my_rcv_H my_ref VCVS A_signal_pos my_ref 1
E_L my_rcv_L my_ref VCVS A_signal_neg my_ref 1
*******************************************************************************
.ends
Package modeling with IBIS-ISS (a simple example)

Legacy IBIS file:

```plaintext
Example of an IBIS model using an IBIS-ISS package model

[Pin] signal_name model_name
  1   ...   ...
  2   ...   ...
  3   ...   ...
  4   ...   ...
  5   Channel_1P  ISS_Diff_Tx
  6   Channel_1N  ISS_Diff_Tx
  7   Channel_2P  ISS_Diff_Tx
  8   Channel_2N  ISS_Diff_Tx
  9   ...   ...
 10   ...   ...

[Package Model] A_4_pin_pkg_model
...```

IBIS-ISS subcircuit:

```plaintext
عص_LIBK_S_pkg P1 P2 P3 P4 P5 P6 P7 P8
+ TSFile="TouchstoneFileName.s8p"

Sdriven P1 P2 P3 P4 P5 P6 P7 P8
+ MNAME=TSFile
+ [FBASE = base_frequency] [FMAX=maximum_frequency]

.ends```

IBIS .pkg file:

```plaintext
This example implements a package model using an IBIS-ISS subcircuit.

[Define Package Model] A_4_pin_pkg_model
[Manufacturer] Noname Company, Inc.
[OEM] Another Noname Package Company, Inc.
[Description] 4-pin illustration package model
[Number Of Pins] 4

[Pin Numbers]
  5  DiePortName = IDP_5
  6  DiePortName = IDP_6
  7  DiePortName = IDP_7
  8  DiePortName = IDP_8

[Package Circuit]
Language IBIS-SS

  | Corner corner_name file_name circuit_name (.subckt name)
  | Corner Typ PackageModel.spi S_pkg
  | Parameters List of parameters
  | Parameters TSFile = “My_TstoneFile.s8p”
  | Ports are in same order as defined in SPICE
  | Ports  5  6  7  8
  | Ports  IDP_8  IDP_7  IDP_6  IDP_5

[End Package Circuit]
[End Package Model]```

Matched by name

Matched by position

Declaration of Implicit die ports
Backup
**IBIS-AMI Analog Modeling and Much Needed Improvements for IBIS**

---

**Figure 12 (pg. 136) in IBIS v5.0**

---

<table>
<thead>
<tr>
<th>Component Die</th>
<th>Package</th>
<th>Pins/Builds</th>
</tr>
</thead>
<tbody>
<tr>
<td>[External Circuit]</td>
<td>[External Circuit]</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A_my porous--a--vocal vcc--10---$88--0 10 Vcc</td>
<td></td>
</tr>
<tr>
<td>\</td>
<td>A_my porous--b--vocal</td>
<td></td>
</tr>
<tr>
<td>D_drive--</td>
<td>&gt;A_my sig--int sel 101--1---$88--0 1 Buffer A</td>
<td></td>
</tr>
<tr>
<td>D_enable--/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D_receive/&lt;</td>
<td></td>
<td>GND--pad_11--$88--0 11 GND</td>
</tr>
<tr>
<td>\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\</td>
<td></td>
<td>Interconnect</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>A_my porous--f--vccbl</td>
<td></td>
</tr>
<tr>
<td>\</td>
<td>A_my sig--g--int ob o2--pad_2a--$88--0 2 justify</td>
<td>Self Adjusting</td>
</tr>
<tr>
<td></td>
<td>A_my porous--h--vccbl</td>
<td>Buffer</td>
</tr>
<tr>
<td></td>
<td>A_my cnt</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Analog Buffer Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pad_2b--$88--0</td>
</tr>
<tr>
<td>[External Circuit]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A_my porous--10--(to pin/pad 10)</td>
<td></td>
</tr>
<tr>
<td>\</td>
<td>A_my porous--10--(to pin/pad 10)</td>
<td></td>
</tr>
<tr>
<td>ndi--D_mydrv--</td>
<td>&gt;A_my sig--3--$88--0 3 Buffer C</td>
<td></td>
</tr>
<tr>
<td>D_enable--/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D_receive/&lt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[External Circuit]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>A_my porous--10--(to pin/pad 10)</td>
<td>$88--0 4a Clocka</td>
</tr>
<tr>
<td>ndi--D_receive/&lt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A_my sig--pad_4--pad_1--$88--0 4b Clockb</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A_my porous--pad_11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[External Model] inside [Model]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>A_pcoref---&gt;</td>
<td></td>
</tr>
<tr>
<td>\</td>
<td>A_pcoref---&gt;</td>
<td></td>
</tr>
<tr>
<td>D_drive--</td>
<td>&gt;A_signal</td>
<td></td>
</tr>
<tr>
<td>D_enable--/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D_receive/&lt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Figure 12: Reference example for [Node Declarations] keyword

---

Mentor Graphics®
Figure 12 implemented with IBIS-ISS

Legacy IBIS file:

```plaintext
[Pin] signal_name model_name
10 Vcc POWER
 1 A0 CIRCUITCALL
11 GND GND
 2 CAS0 CIRCUITCALL
 3 A1 CIRCUITCALL
 4a Clk_A CIRCUITCALL
 4b Clk_B CIRCUITCALL
 5 A2 Buffer_E

[Node Declarations]
Die nodes:
  a  b  c  d  e  f  g  h  nd1
Die pads:
  pad_2a  pad_2b  pad_4  pad_11

[End Node Declarations]

[Package Model] QS-SMT-cer-8-pin-pkgs

Annotation: Explicit die ports are declared under [Node Declarations].
```

IBIS .pkg file:

```plaintext
[Define Package Model] QS-SMT-cer-8-pin-pkgs
[Manufacturer] Quality Semiconductors Ltd.
[OEM] Acme Package Co.
[Description] 8-Pin ceramic SMT package
[Number Of Pins] 8

[Pin Numbers]
10  DiePort = IDP_10
 1  DiePort = IDP_1
11  DiePort = pad_11
 2  DiePort = pad_2a
 2  DiePort = pad_2b
 3  DiePort = IDP_3
 4a DiePort = pad_4
 4b DiePort = pad_4
 5  DiePort = IDP_5

[Package Circuit] Language IBIS-ISS

-corner corner_name file_name circuit_name (.subckt name)
Corner Typ PackageModel.spi S_pkg

| Parameters List of parameters
Parameters TSFile = “My_TstoneFile.s16p”
| Ports are in same order as defined in SPICE
 Ports IDP_5 pad_4 IDP_3 pad_2b
  Ports pad_2a pad_11 IDP_1 IDP_10

[End Package Circuit]
[End Package Model]
```

Explicit die ports (in blue) are declared under [Node Declarations].

Matched by position

Matched by name

Implicit die ports (in blue)

Matched by name

Explicit die ports (in red)