Practical Issues in Enabling a Corporate IBIS Library

IBIS Summit / DesignCon West 2005
Todd Westerhoff
Cisco Systems, Inc.
Signal Integrity: Trouble in Paradise

- Many IBIS models have significant quality problems
- Many semiconductor vendors lack expertise to create high quality models
- SI specialists waste time addressing quality issues
Bringing SI to the Masses

- Most hardware designers lack skills to find/resolve model problems
- SI libraries are where CAD libraries were 10-15 years ago
  - Ad-hoc library organization is becoming untenable
  - A more centralized library strategy is needed
Corporate SI Library Goals

- Help move SI analysis from the realm of the specialist to the engineer’s desktop:
  - Centralized, high quality SI library maintained by experts
  - Minimal time spent debugging / correcting models
  - General purpose library structure (not dictated by the needs of a specific toolset)
  - Simplified process to get design databases ready to simulate
Key Concepts

- Use “incoming inspection process” to identify problem models early
- Organize library for data management, map to structure needed by EDA tools
- Unified model naming convention allows users to find and select components easily
- Automate as much of the overall process as possible
Incoming Inspection Process

- Flag model problems BEFORE they get used for design work
  - Identify models that are poorly constructed or self-inconsistent
  - Develop a checklist for identifying common modeling problems
  - Re-examine models that have caused problems in past projects
  - Incorporate Industry Standards (IBIS Quality Checklist)
- Automate inspection process as much as possible
Driving IBIS Model Quality

- “Fix problem at the source” - drive quality issues back to device vendors through Component Management
- Use resources to support process improvement, instead of debugging models
Managing Library Data

- Organize data to allow developers and users to find the latest information
- Employ version control and revision histories
- Library / SI tool strategy should handle “second source” components
The Only Authority For a Part Type ...

... in the device library ...

... is the Corporate part number ...

... that points to the AVL list.
Library Directory Structure

Master Library

EDA Tool View

- Organize master library by Corporate part number
- Create different library “views” for EDA tools
  ➢ Meets goal of tool independent design
- Allows EDA tools to operate efficiently
Naming Strategy

- IBIS models from different semiconductor vendors will have no unified file, component or buffer naming strategy
- Depending on the EDA tools and models used, name collisions are possible
- Users assigning models interactively will benefit from a consistent naming approach
How To Organize IBIS Model Data?

Library directory structure

15-2201-01/
...
15-2236-01/
  TI/
   IBIS/
   sn74lvt244apw.ibs
   sn74lvth244apw.ibs
  PHL/
   IBIS/
   74lvt244apw.ibs
  FSI/
   IBIS/
   74lvth244mtc.ibs

} By Corporate part number
  }
  By vendor
    }
    By model type
    }
    By vendor part number
However ....

... Some IBIS simulators see only the [Component] and [Model] names from the .ibs files

Library directory structure

15-2236-01/
  HSD/
    TI/
      IBIS/
        sn74lvt244apw.ibs
        sn74lvth244apw.ibs
    PHL/
      IBIS/
        74lvt244apw.ibs
    FSI/
      IBIS/
        74lvth244mtc.ibs

... [Component]  LVT244B_DW
  ... [Model]  LVT244B_IN_33
  ... [Model]  LVT244B_OUT_33
  ...
  ti_sn74lvt244apw.ibs

... [Component]  LVT244B_DW
  ... [Model]  LVT244B_IN_33
  ... [Model]  LVT244B_OUT_33
  ...
  fsi_74lvth244mtc.ibs

Which is Which?

Device  LVT244B_DW
Input  LVT244B_IN_33
Output  LVT244B_OUT_33

Simulation library
For SI Libraries to Work Properly …

… [Component] names must be unique across the entire device library

Library directory structure:

- 15-2236-01/
  - HSD/
    - TI/
      - IBIS/
        - sn74lvt244apw.ibs
        - sn74lvth244apw.ibs
      - PHL/
        - IBIS/
          - 74lvt244apw.ibs
      - FSI/
        - IBIS/
          - 74lvth244mtc.ibs

One strategy: redefine IBIS [Component] to a 2/3 character vendor ID + vendor part number

Devices:
- TISN74LVT244APW
- TISN74LVTH244APW
- PHL_74LVT244APW
- FSI_74LVTH244MTC

Simulation library
An IBIS Naming Convention

[Component] declaration is 2/3 character vendor identifier + number for the device “function”

Multiple [Component] declarations can reside in the same IBIS file

Buffer model names end in “…IN, _OUT or _BI” according to their type

File name is 2/3 character vendor identifier + part number for the main device described in the model
What Users Want …

PCB Database

Central Simulation Library

Then a Miracle Occurs

Simulatable Database
The User’s Question …

1. Do we have a simulation model for this part?

2. What on earth is a NT3_PE_BGA1627-01, HSINK-VDD1P5A?
... The Answers

1. Maybe ...

2. The concatenation of the DEVICE and VALUE properties for this component
... But One Part Can Have Multiple Models!

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Part Number</th>
<th>MCN Pending</th>
<th>Qualification Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEXAS INSTRUMENTS</td>
<td>SN74LVTH244APW</td>
<td>No</td>
<td>Qualified</td>
</tr>
<tr>
<td>PHILIPS SEMICONDUCTOR</td>
<td>74LV244APW</td>
<td>Yes</td>
<td>Qualified</td>
</tr>
<tr>
<td>TEXAS INSTRUMENTS</td>
<td>SN74LVTH244APW</td>
<td>No</td>
<td>Qualified</td>
</tr>
<tr>
<td>FAIRCHILD SEMICONDUCTOR</td>
<td>74LVTH244MTC</td>
<td>No</td>
<td>Qualified</td>
</tr>
</tbody>
</table>

No of records retrieved: 4

... most EDA toolsets have no provision for this
The Role of Scriptware

- Map parts to simulation models
- Assign models for “second source” components
- Identify missing simulation models
Conclusion

- Bringing “SI to the masses” will require a centralized library strategy
  - Use incoming inspection to identify model problems early
  - Library can be organized for management, mapped to views that EDA tools require
  - Unified model naming strategy provides consistent component/buffer names to users
  - Automating SI model assignment eases the user’s burden and manages “second source” simulation models
… Sometimes Miracles DO Happen!