

Model Handling and IBIS File Sizes –
Recent Experiences with new IBIS Files
Size/Complexity Evolution and Implications for Model
Users

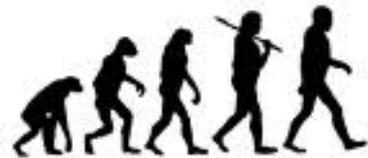
Hirohiko Matsuzawa
Zuken Inc – Yokohama/Japan
Ralf Brüning, Michael Schäder
Zuken EMC Technology Center
Paderborn/Germany

Asian IBIS Summit
Tokyo Japan
November 15, 2010

Introduction/Scope of Presentation



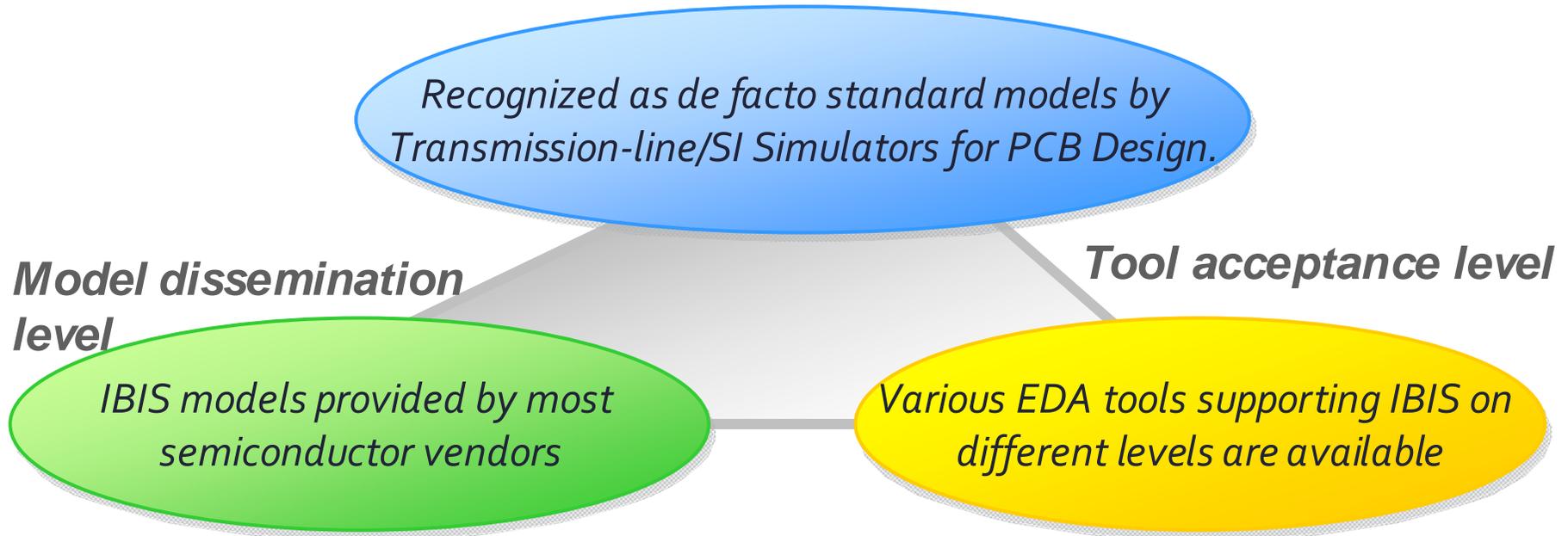
- In this presentation, Zuken would like to share experiences and observations made in the last 24 months on the file size and data amount of new IBIS files, which make model usage for the people in the CAD departments (which are often non IBIS experts) more difficult.
- This is meant as a proposal for discussion between IBIS model makers, CAD tool vendors and IBIS model users.
- Of course Zuken R+D recognizes technology evolution and that modern high sophisticated and complex silicon defines serious challenges to the model makes (and the model users) !



Recognition of IBIS Models



The recognition/perception of IBIS



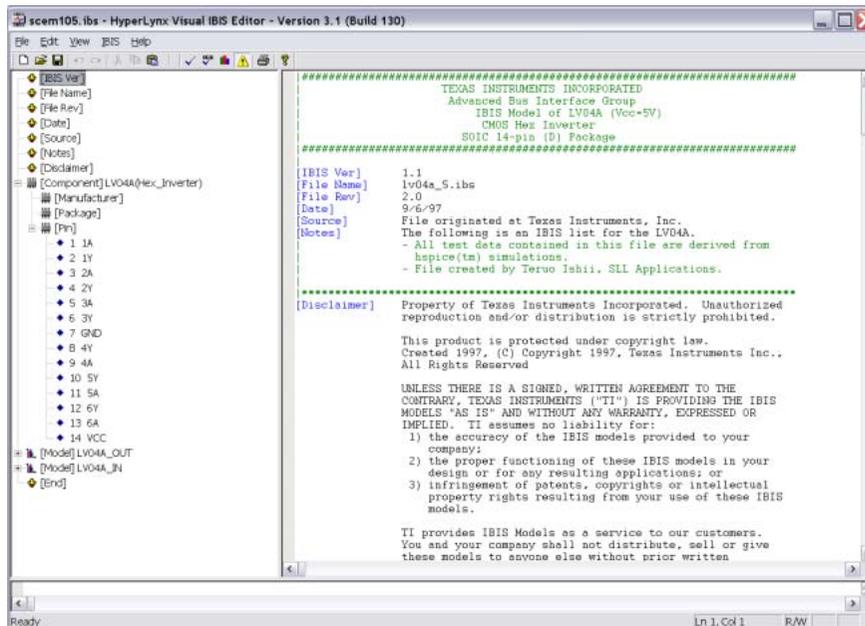
Looking overall, the IBIS environment is very well accepted (that's what we hoped to achieve in the early 90's...)

The Beginning ...



In the beginning IBIS has been defined when rather simple devices have been used on PCBs:

- SOP and PLCC packages, just very few models, simple RLC packages
- Later on processor models, more complex packages
- DIMM modules as EBDs with IBIS 3.x



IBIS Data Usage in today's EDA Tools



- EDA Tools either read IBIS at runtime, or they convert IBIS data into own model formats (aka import and conversion process) or instantiate such models (i.e. HSPICE)
- When using the „IBIS Golden Parser“ (to some extend most EDA tools utilize it), IBIS relevant data structures are created → huge operations in memory will take place

Chapter 2: HSPICE Netlist Commands
IBIS

.IBIS

Provides IBIS functionality by specifying an IBIS file and component and optional keywords.

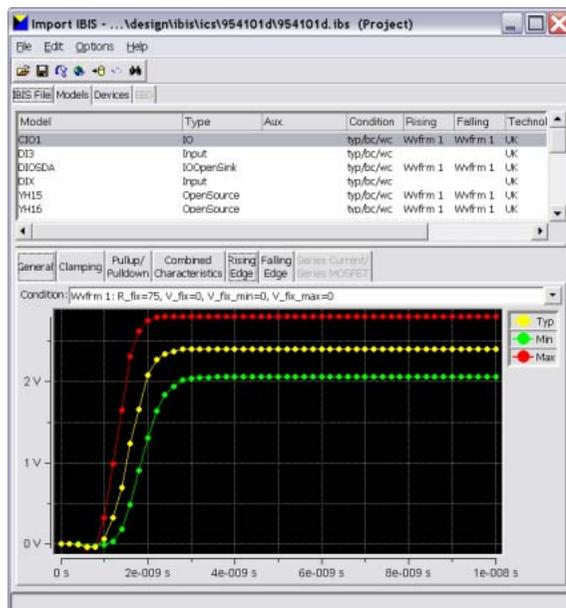
Syntax

```
.IBIS 'ibis_name'
+ file = 'ibis_file_name'
+ component='component_name' [time_control=0|1]
+ [mod_sel='se1=mod1,se2=mod2,...']
+ [package = 0|1|2|3] [pkgfile='pkg_file_name']
+ [typ={typ|min|max}]
+ [nowarn]
+ ...
```

Arguments

Keyword	Description
ibis_name	Instance name of this ibis command
file	Name of ibis (.ibs) file
component or cname	Component name
time_control	Invokes an HSPICE time-control algorithm to achieve greater accuracy for high speed digital signal buffers <ul style="list-style-type: none"> 0 (default): Time step algorithm will not take effect 1: Launches time-step algorithm
mod_sel	Assigns special model for model selector, here model selector can be used for series model. If model selector is used for a pin of a component, but mod_sel is not set in the .ibis command, then the first model under the corresponding [Model Selector] will be selected as default.

HSPICE® Reference Manual: Commands and Control Options
A-2007.09 79



The 'Select IC Model' dialog box shows information on the selected device: Library: 90c031m.ibs, Device: DS90C031TM, Signal: DOUT4+, Pin: 13. It includes a list of libraries and devices to select from, and a 'Select by' section with options for Pin and Signal.

New Problems (1): IBIS file size grows significantly

- Some recent samples:
 - Altera Stratix III:
 - 68.1 MB file size
 - 2359 (generic) pins, 7077 models PLUS model selector variants (ODT variation)
 - Passed IBISCHK (2 minutes on a fast PC), but some IBIS freeware tools (editors) have problems
 - For the end user the excellent XLS documentations allows model selection (but: manual process, error prone)
 - Lattice sc2.ibis
 - More then 80 MB
 - 2163 (generic) pins
 - Very ,special' model naming scheme

NAMING CONVENTION

The IBIS [Model] header is limited by the specification to a total of 20 characters. With such a set of characters available for naming models it becomes important to attempt to meaningfully encode the IO standards so they fit within the twenty character limit. It would seem that twenty characters would provide room enough for describing IO's. However, the ELD IO structure continues to grow more and more complex. The complexity is making the twenty characters insufficiently descriptive. In order to overcome this issue the naming convention described below is implemented to resolve the issue.

The twenty character space is managed as follows:
 bbbvvvdddprugtcoixx

b = standard
 v = voltage (x.xx)
 s = slew code
 d = drive (xx.x me)
 p = pullup code
 r = series resistor
 u = terminate to v
 g = terminate to g
 t = terminate to t
 c = common mode te
 o = diff resistor
 i = diff resistor
 x = reserved

[Pin]	signal_name	model_name	R_f
1	ag1330axxxxxxxxxain	ag1330axxxxxxxxxain	
2	ag1330axxxbxxxxxxxxain	ag1330axxxbxxxxxxxxain	
3	ag1330axxxcxxxxxxxxain	ag1330axxxcxxxxxxxxain	
4	ag1330axxxexxxxxxxxain	ag1330axxxexxxxxxxxain	
5	ag1330fxxxxxxxxxxxxaio	ag1330fxxxxxxxxxxxxaio	
6	ag1330fxxxxxxxxxxxxaou	ag1330fxxxxxxxxxxxxaou	
7	ag1330fxxxbaaaaaaio	ag1330fxxxbaaaaaaio	
8	ag1330fxxxbaaaaaaou	ag1330fxxxbaaaaaaou	
9	ag1330fxxxcaaaaaaio	ag1330fxxxcaaaaaaio	
10	ag1330fxxxcaaaaaaou	ag1330fxxxcaaaaaaou	
11	ag1330fxxxexxxxxxxxxaio	ag1330fxxxexxxxxxxxxaio	
12	ag1330fxxxexxxxxxxxxaou	ag1330fxxxexxxxxxxxxaou	

Lattice Semiconductor Corporation

Home | Products | Solutions | Support | Documents | Downloads | Sales | Store | About Us

Home > Products > FPGA > LatticeSC/M > PURESPEED I/O Technology > PURESPEED I/O Technology IBIS Models

PURESPEED I/O Technology IBIS Models

To access individual documents listed below, click on the document title. Click on any column header to sort the list of available resources.

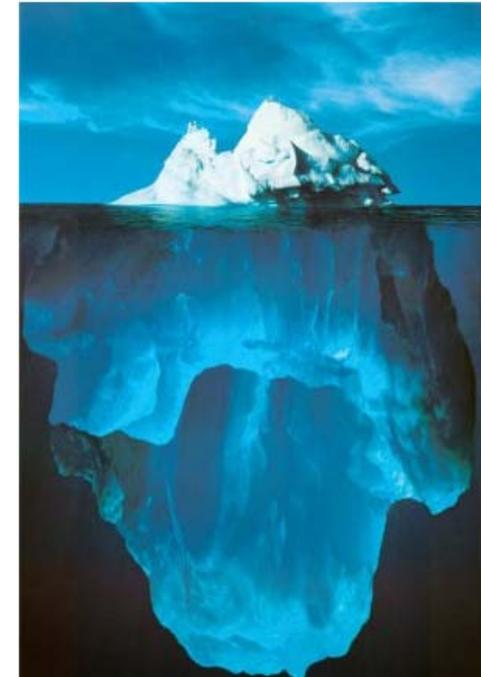
Icon Key:
R Restricted access (account Sign-in required)

Type	Title	Version	Date	Format	Size
R IBIS Models	[IBIS] LatticeSC IBIS Model	2.0	Feb 2009	IBS	78.5 MB

New Problems (2): Massive use of Model Selector Statements

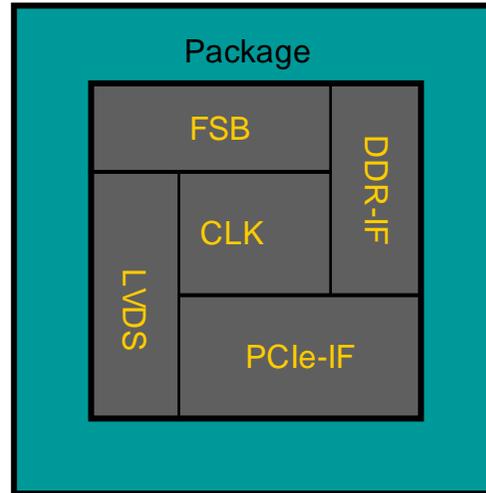
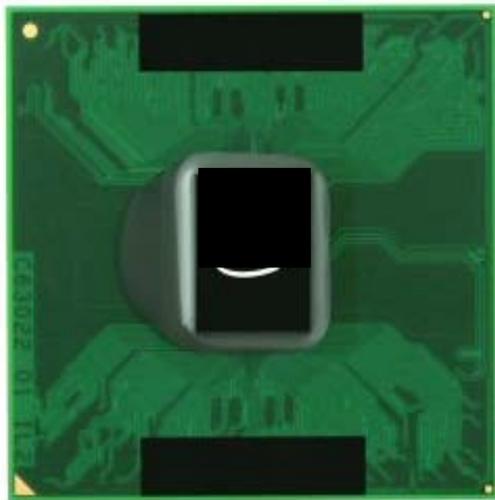
- Up to 30 model selectors are used in some cases
 - Issues are often not visible directly to model users
 - Samples:
 - Micron/Samsung EBDs
 - FPGA technology models

```
[Model Selector] DQ
DQ_FULL_533      Full-Strength IO Driver with no ODT
DQ_FULL_ODT50_533 Full-Strength IO Driver with 50 Ohm ODT Submodel
DQ_FULL_ODT75_533 Full-Strength IO Driver with 75 Ohm ODT Submodel
DQ_FULL_ODT150_533 Full-Strength IO Driver with 150 Ohm ODT Submodel
DQ_HALF_533      Reduced-Strength IO Driver with no ODT
DQ_HALF_ODT50_533 Reduced-Strength IO Driver with 50 Ohm ODT Submodel
DQ_HALF_ODT75_533 Reduced-Strength IO Driver with 75 Ohm ODT Submodel
DQ_HALF_ODT150_533 Reduced-Strength IO Driver with 150 Ohm ODT Submodel
DQ_FULL_800      Full-Strength IO Driver with no ODT
DQ_FULL_ODT50_800 Full-Strength IO Driver with 50 Ohm ODT Submodel
DQ_FULL_ODT75_800 Full-Strength IO Driver with 75 Ohm ODT Submodel
DQ_FULL_ODT150_800 Full-Strength IO Driver with 150 Ohm ODT Submodel
DQ_HALF_800      Reduced-Strength IO Driver with no ODT
DQ_HALF_ODT50_800 Reduced-Strength IO Driver with 50 Ohm ODT Submodel
DQ_HALF_ODT75_800 Reduced-Strength IO Driver with 75 Ohm ODT Submodel
DQ_HALF_ODT150_800 Reduced-Strength IO Driver with 150 Ohm ODT Submodel
```



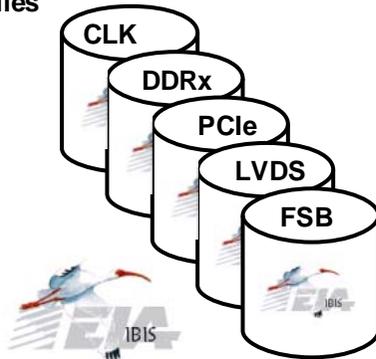
News Problems (3): Splitted IBIS files, seperate package models

Example: Large processor model



Manual work needed for using such models in SI simulation !!!

Various IBIS Files



Zuken Observations



- New devices are often modelled by really HUGE IBIS device descriptions with several thousands of models
- New DDRx memory modules utilize EBDs heavily (the re-appearance of the IBIS EBDs is potentially caused by the DDR DIMM modules)
- Package models are often missing, or implicit set to zero RLC values, sometimes an external package model is referenced in SPICE or S-Parameters syntax → further manual editing before simulation is required

⇒ IBIS will get less handy for model users

```
[Component]      Pentium_M_rev13
[Manufacturer]   Intel Corporation
|
[Package]
|
|      typ          min          max
R_pkg      0.0000hm      0.0000hm      0.0000hm
L_pkg      0.00H         0.00H         0.00H
C_pkg      0.00F         0.00F         0.00F
|
| User selects a package model by uncommenting one the following models.
[Package Model]  bnspkg_fsb_single_line
|[Package Model]  bnspkg_fsb_odd
|[Package Model]  bnspkg_fsb_even
|
| *****
```

Conclusion/Question



- The recent development on (some) IBIS device descriptions makes the model usage more complicated, especially for less experienced users.
- EDA-Tools can still handle such models, but if this development continues, IBIS data handling will become challenging.
- Manual editing of (sometimes several) model files will lower down the convenience and acceptance of using IBIS and therefore may harm the future progress/usage of the IBIS standard.