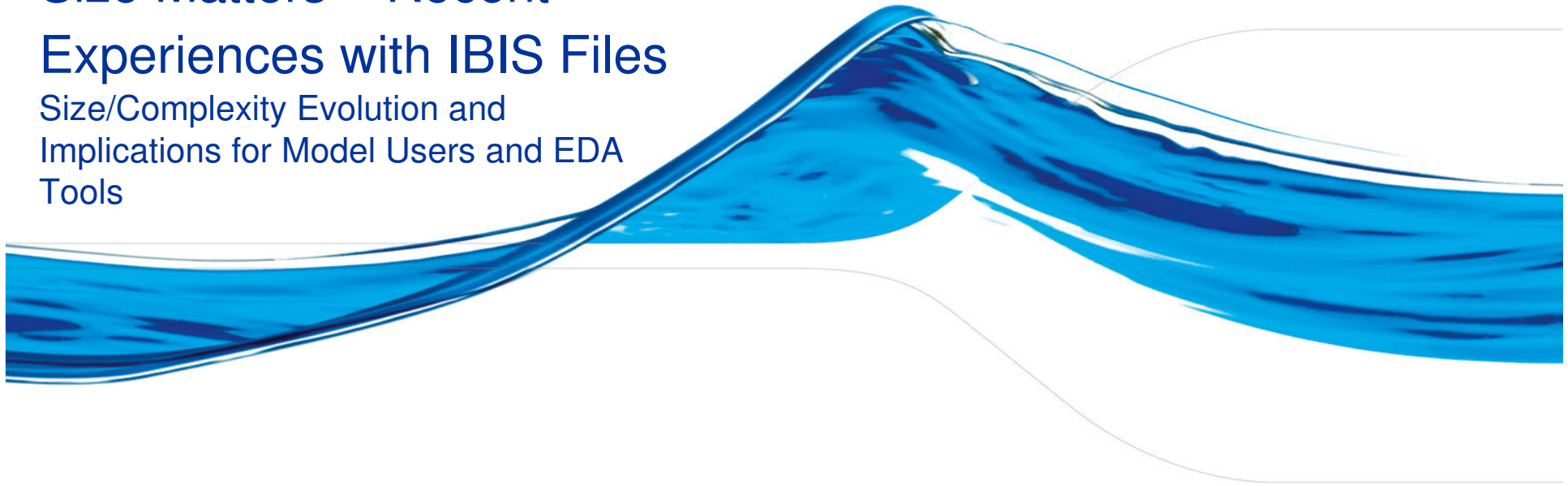


Size Matters – Recent Experiences with IBIS Files

Size/Complexity Evolution and
Implications for Model Users and EDA
Tools

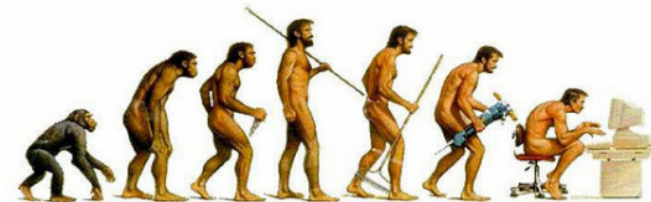


Ralf Brüning, Michael Schäder
Zuken EMC Technology Center
Paderborn - Germany

Preface



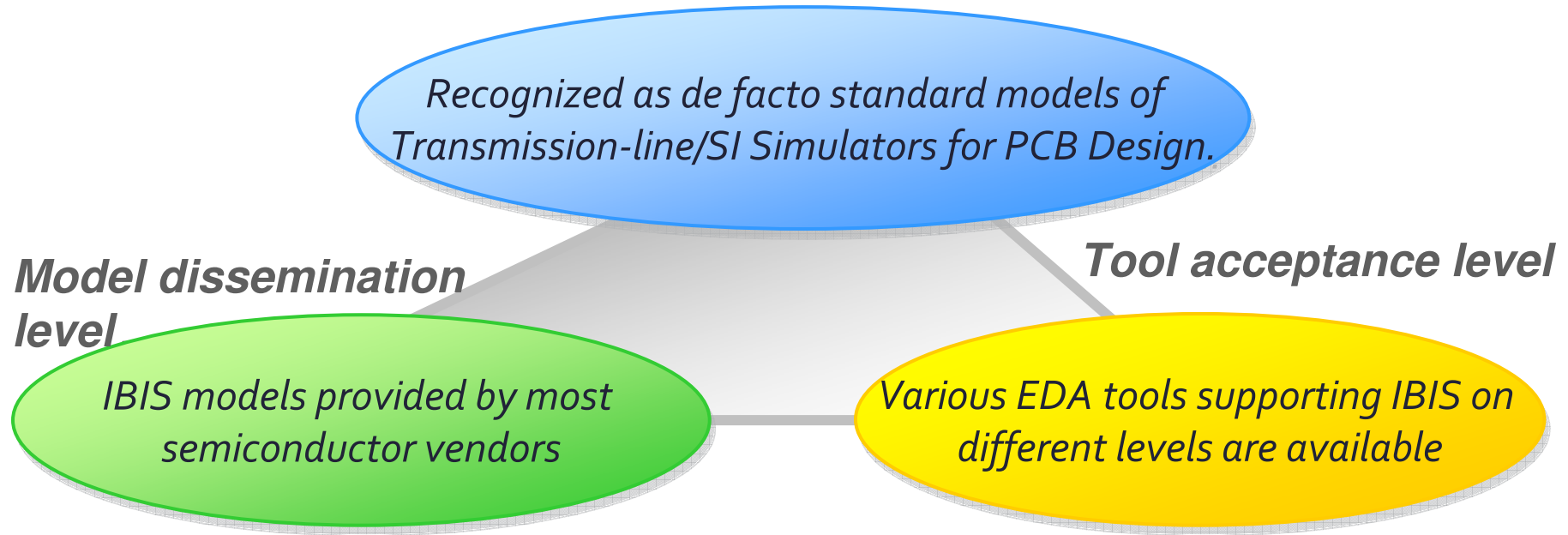
1. This is NOT one of these *,in the good old days‘* presentation.
2. We like to share experiences and observation we made in the last 12 months which make model usage for the people in the CAD departments (which are often not IBIS experts) more difficult.
3. This is meant as a proposal for discussion between model makers, tool vendors and model users.
4. Of course we recognize technology evolution and that modern high sophisticated and complex silicon defines serious challenges to the model makes (and the model users) !



Recognition of IBIS



The recognition/perception of IBIS

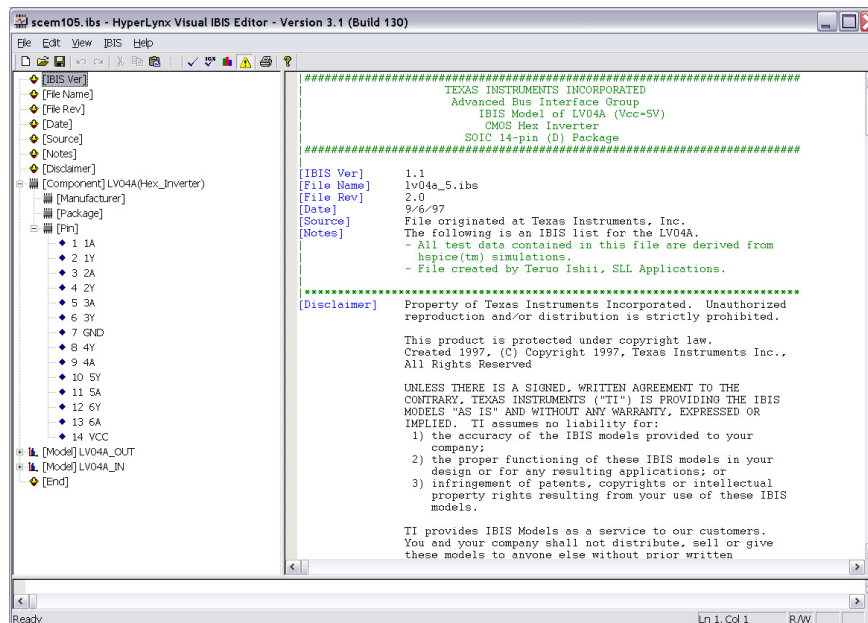


*Looking overall, IBIS environment is very well accepted
(thats what we hope to archieve in the early 90ies...)*

The Beginning ...

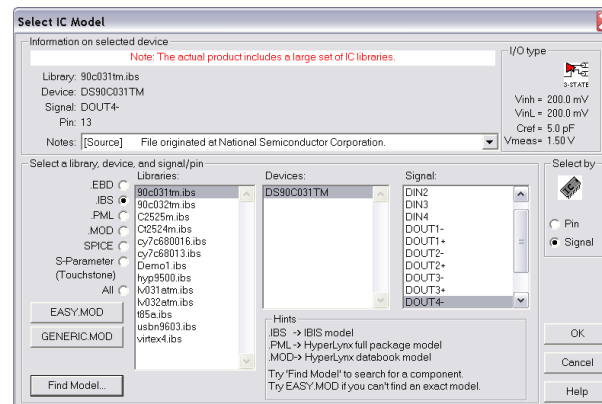
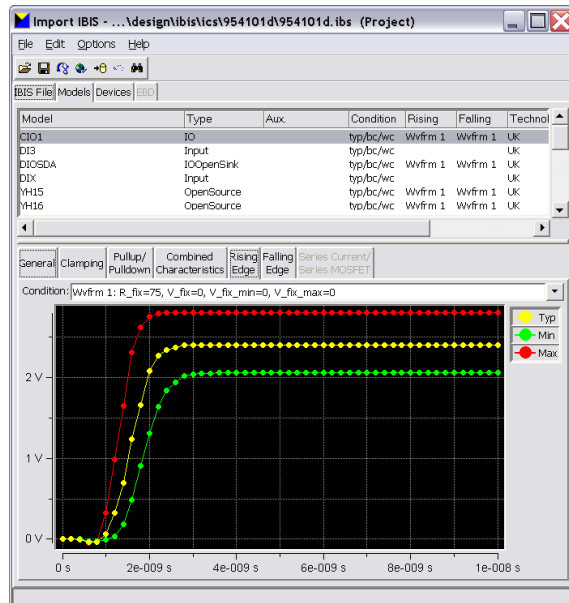
IBIS in the beginning has been defined when we had rather simple devices:

- SOP and PLCC packages, 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 into own model formats (also import and conversion process) or instantiate such models (i.e. HSPICE)
- When using the golden parser (to some extent most EDA tools utilize it) IBIS relevant data structures are created → huge operations in memory will take place



Chapter 2: HSPICE Netlist Commands
.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='sel1=mod1, sel2=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 effect1: 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 Problem (1): IBIS file sizes explode



- Some recent samples:
 - Altera Stratix III:
 - 68.1 MB file size
 - 2359 (generic) pins, 7077 model PLUS model selector variants (OCT variation)
 - Passed IBISCHK (2 minutes on a fast PC) , but some IBIS freeware (editor) have problems
 - For the end user 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

```
MS-DOS
C:\User\Zuken\IBIS>c:\bin\time.exe ibischk4.exe c:\user\Zuken\IBIS\Models\Altera
\stratix3.ibs
IBISCHK4 V4.2.2

Checking c:\user\Zuken\IBIS\Models\Altera\stratix3.ibs for IBIS 4.2 Compatibility...

Errors : 0

File Passed

real    1m54.00s
user    0m00.01s
sys     0m00.00s

C:\User\Zuken\IBIS>
```

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 PLD 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:
bbbvvsddprugtcioixx

b = standard
v = voltage (x.xx V)

x.xx ma)
R
code
assistance code
to vcc code
to gnd code
to vtt code
mode termination mode
istor code
istor current code

[Pin]	signal_name	model_name	R
1	ag1330axxxxxxxxxain	ag1330axxxxxxxxxain	
2	ag1330axxxbaaaaaaaaaain	ag1330axxxbaaaaaaaaaain	
3	ag1330axxxcaaaaaaaaaain	ag1330axxxcaaaaaaaaaain	
4	ag1330axxxeaaaaaaaaain	ag1330axxxeaaaaaaaaain	
5	ag1330fxxxxaaaaaaaaaio	ag1330fxxxxaaaaaaaaaio	
6	ag1330fxxxxaaaaaaaaaou	ag1330fxxxxaaaaaaaaaou	
7	ag1330fxxxbaaaaaaaaaaio	ag1330fxxxbaaaaaaaaaaio	
8	ag1330fxxxbaaaaaaaaaaou	ag1330fxxxbaaaaaaaaaaou	
9	ag1330fxxxcaaaaaaaaaaio	ag1330fxxxcaaaaaaaaaaio	
10	ag1330fxxxcaaaaaaaaaaou	ag1330fxxxcaaaaaaaaaaou	
11	ag1330fxxxeaaaaaaaaaio	ag1330fxxxeaaaaaaaaaio	
12	ag1330fxxxeaaaaaaaaaou	ag1330fxxxeaaaaaaaaaou	

The Problem (2): Use of Model Selector Statements get a massive Commodity

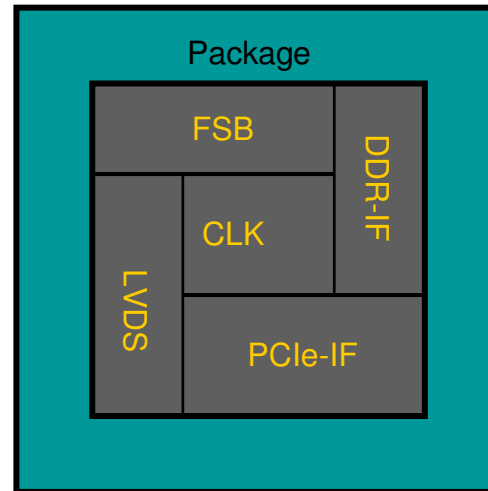
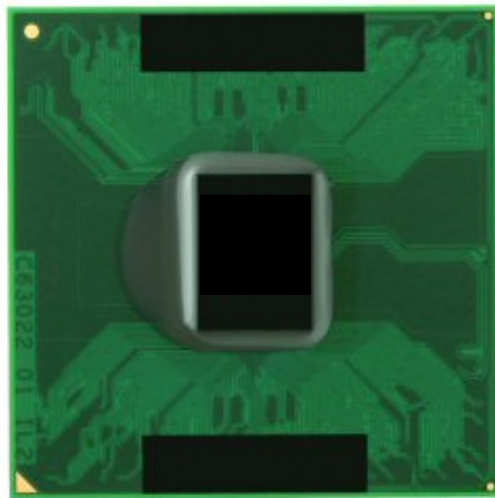
- Up to 30 model selectors in some cases, problem often not direct visible to model users
 - Micron/Samsung EBDs
 - FPGAs

```
[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
```



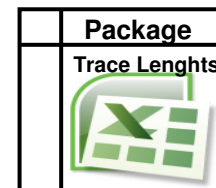
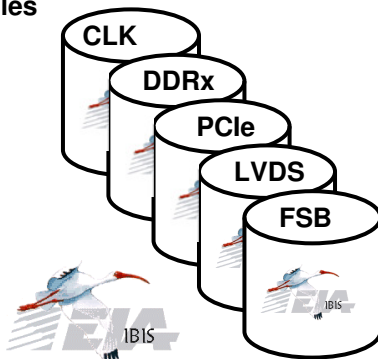
The Problem (3): Splitted IBIS files, Seperate package (here large processor model)



Hand Work needed !!!



Various IBIS Files



Observations



- New devices are often modeled by really HUGE IBIS device descriptions with thousands of models
- New memory modules utilizes heavily EBDs (the renaissance of the EBDs, caused by DDRs modules)
- Package models often missing, or implicit set to zero and external package model is referred in SPICE or S-Parameters synthax → often further handwork

⇒ IBIS gets less handy for model users (then intended ?)

```
[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 uncommentting one the following models.
[Package Model]  bnspkg_fsb_single_line
[Package Model]  bnspkg_fsb_odd
[Package Model]  bnspkg_fsb_even
|
| *****
```

Conclusion



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