



Update, Restart or Fork?

Michael Mirmak

Oct. 3, 2012

The Question...

Are useful package models so complex that traditional IBIS cannot be modified to support them?

- “Package” here includes on-die interconnect

In other words, to support packages, must we...

- only add/change keywords in IBIS 5.x/6.x
- invalidate existing keywords/relationships
- ... or start with a completely fresh approach?

Clarifying Questions

How many models are generated today using the original package/pin/buffer intent of IBIS?

How many models use & tools support keywords outside of [Model] scope, such as [Pin Mapping] and [External Model]?

How many existing “old” IBIS files are likely to be updated with *new* format package data but *without* buffer model changes?

- Assume “old” means IBIS *4.0* and below
- Does including IBIS *4.2* and below make a difference?

Rating the importance of existing structures & keywords is key to deciding next steps for IBIS

What Do We Need

References to pins

References to die pads (I/O)

References to buffer power connections (may involve multiple die pad connections)

Many-to-one pad-pin connections

One-to-many pad-in connections

Simple, easy-to-understand syntax

Minimal IBIS community costs/burdens

See previous IBIS-ATM meeting proposals and lists;
Strong agreement here

Pros/Cons Working List

Modify existing IBIS: Add keywords but preserving structure

- Examples: [Pin], [Model], [Component], [Pin Mapping], [Node Declarations] stand, add [Package Circuit]

Fork: Deprecate keywords and assumptions, while adding others

- Example: Remove one-to-one [Pin]/[Model] assumption, deprecate [Pin Mapping], add [Die Pads]...

“Blank Sheet of Paper”: no relationship to IBIS 5.1 assumed

- Examples: LISP-like tree hierarchy, separate model, package, component files

