IBIS & ICM Interfacing: A New Proposal

Michael Mirmak
Signal Integrity Engineering
Intel Corp.

Chair, EIA IBIS Open Forum
June 11, 2005

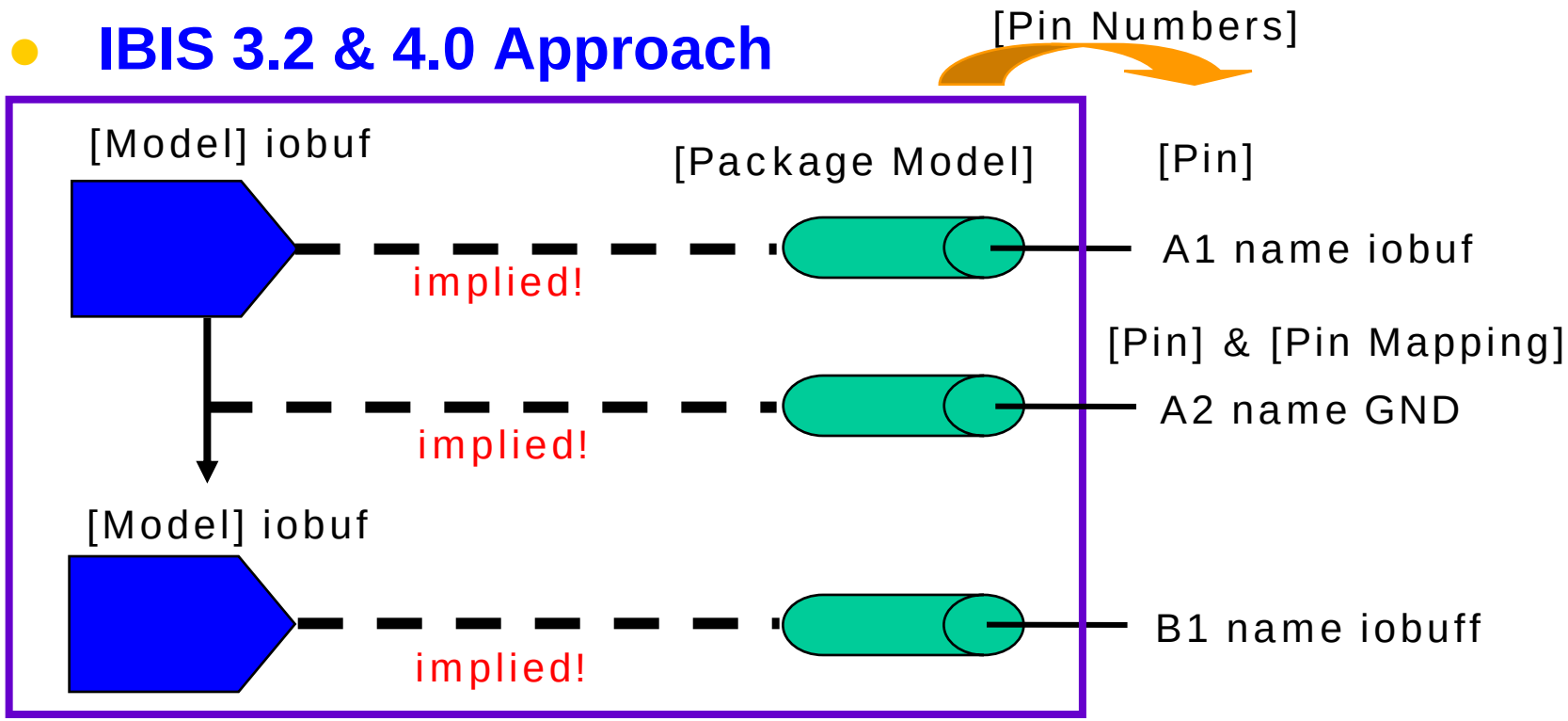
http://www-fmec.fm.intel.com/sie

IBIS & ICM Today

- **Big demand for ICM as IBIS package technology**
- **ICM can describe almost any interconnect**
  - **ICM today includes**
    - **interconnect RLGC or S-parameter characteristics**
    - **coupling, if present, between interconnect segments**
    - **pin (port) end-points and names**
  - **ICM does not involve IBIS in any way today**
    - **Therefore, ICM does not include connections between IBIS [Model], IBIS [Pin] and ICM end-points**
- **IBIS has a number of limits**
  - **1-to-1 connections from package to pin assumed**
  - **Why?**
  - **[Pin] implicitly defines both pin AND buffer instance**

Package Modeling Today

- **IBIS 3.2 & 4.0 Approach**



- **If [Pin] and [Pin Numbers] use the same values…**
  - **Tools assume connections corresponding to values**
  - **Tools infer connections between [Model] and package**
  - **[Pin Mapping] can map supplies to package pins**

IBIS & ICM

- **Could we use ICM to describe IBIS packages?**
  - **No changes required to ICM**
  - **IBIS would require several significant changes**
- **IBIS support for ICM needs to cover…**
  - **Expanding [Package Model] to reference ICM files**
    - **ICM under [Define Package Model] within IBIS files (?)**
  - **[Model], [External Model] and [External Circuit] cases**
    - **Should ICM be limited to only multi-lingual models?**

IBIS & ICM

- **What interfacing options require new syntax?**

1. **IBIS 3.2/4.0 + ICM**
   - **Key decision: expand IBIS beyond 1-to-1 [Pin] connections OR limit ICM to only 1-to-1 paths here**

2. **IBIS 4.1 + [External Model]**
   - **Should be nearly identical to IBIS 3.2/4.0 treatment**
   - **Again, should single path be kept as a limiter?**

3. **IBIS 4.1 + [External Circuit]**
   - **New syntax required for arbitrary ports**

[External Circuit]

- **Linking ICM to IBIS [E. Circuit]**
  - **Use [Node Declarations] to list internal ICM map pin names**

IBIS

```
|****************************************
[Node Declarations]
|Die pads OR PIN NAMES
A1, A2, A3, A4
buff1, buff2, buff3, buff4
[End Node Declarations]
|****************************************
```

Both sides of ICM interconnect are mapped

Only downsides:
Names must be matched;
arbitrary packages not reusable

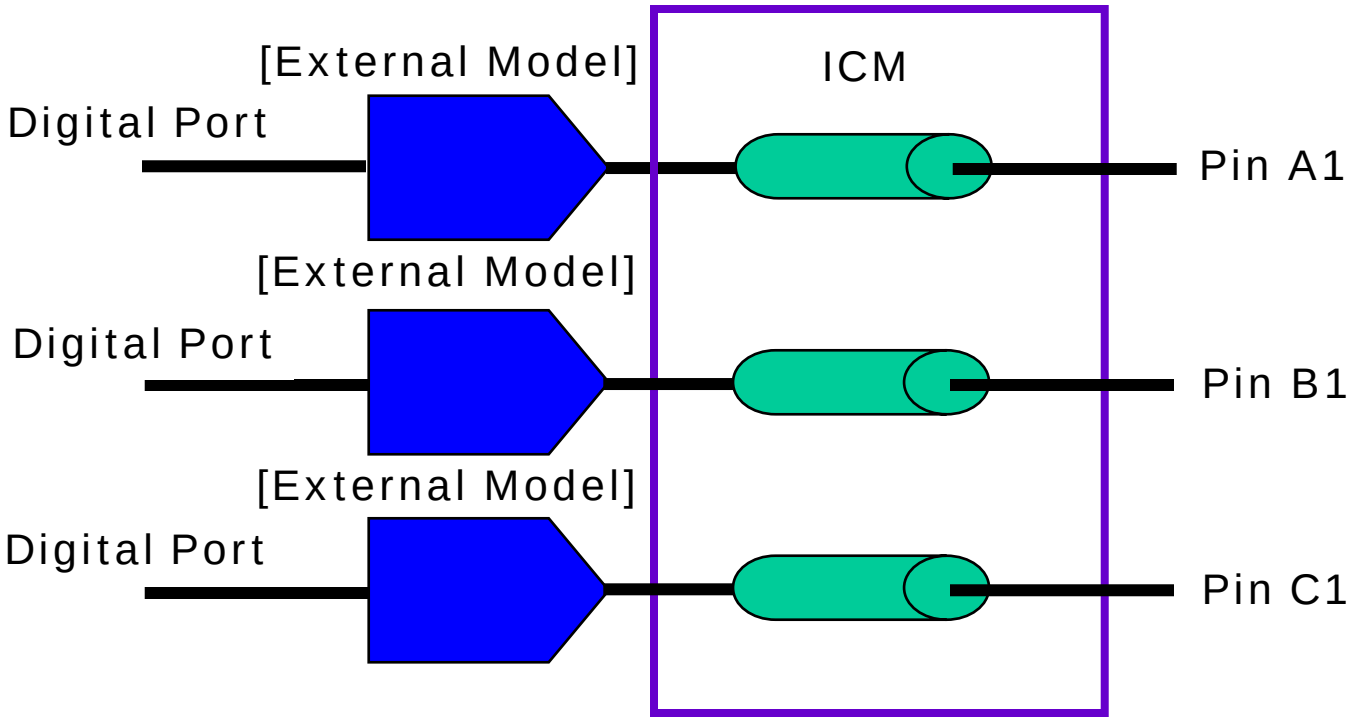
ICM (IIRD8)

```
[ICM Pin Map] Example1_external
Pin_order Row_ordered
Num_of_columns = 4
Num_of_rows = 1
Pin_list
|Pin Name
A1  AD2
A2  AD5
A3  AD7
A4  GND
```

```
[ICM Pin Map] Example1_internal
Pin_order Row_ordered
Num_of_columns = 4
Num_of_rows = 1
Pin_list
|Pin Name
buff1       AD2
buff2       AD5
buff3       AD7
buff4       GND
```

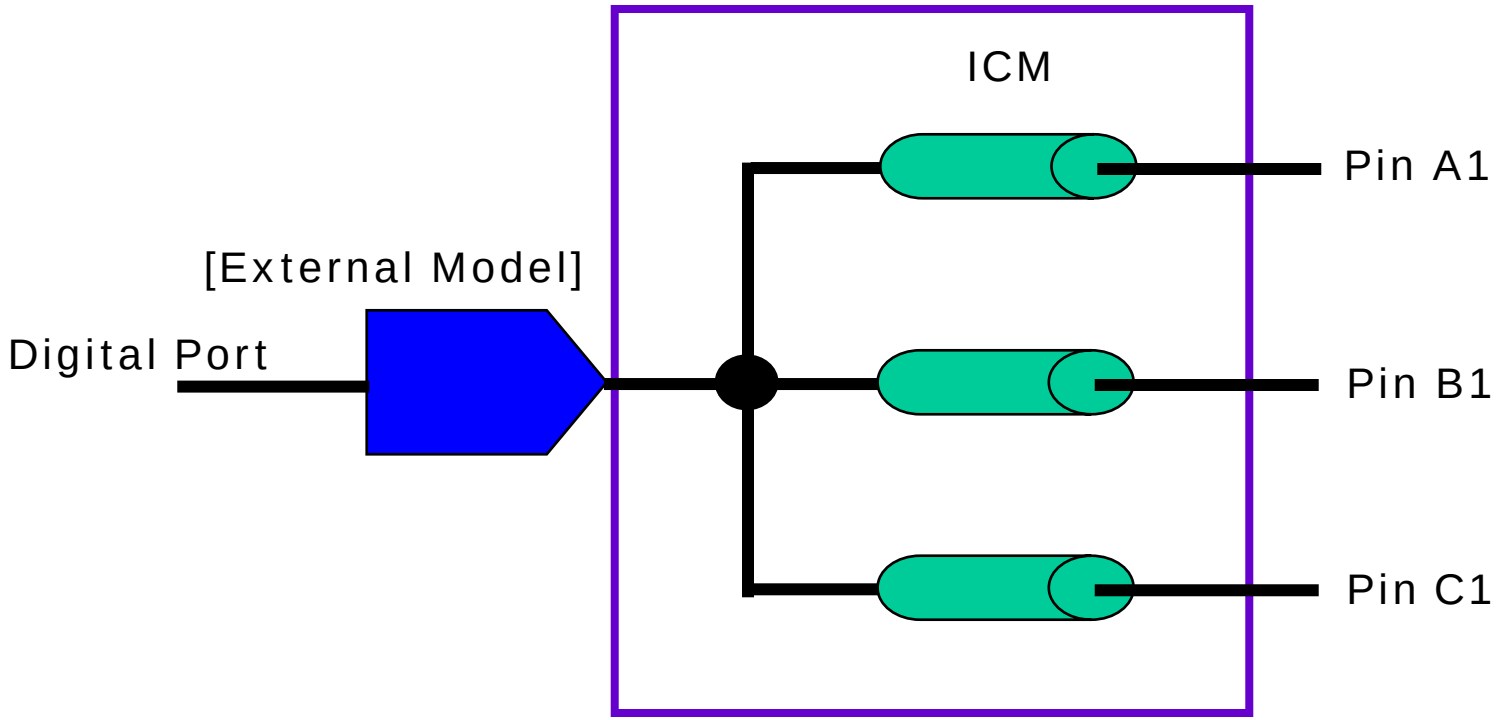
Four Cases

- **We must handle these four cases to be complete**
- **Case 1 – ICM expresses coupling**



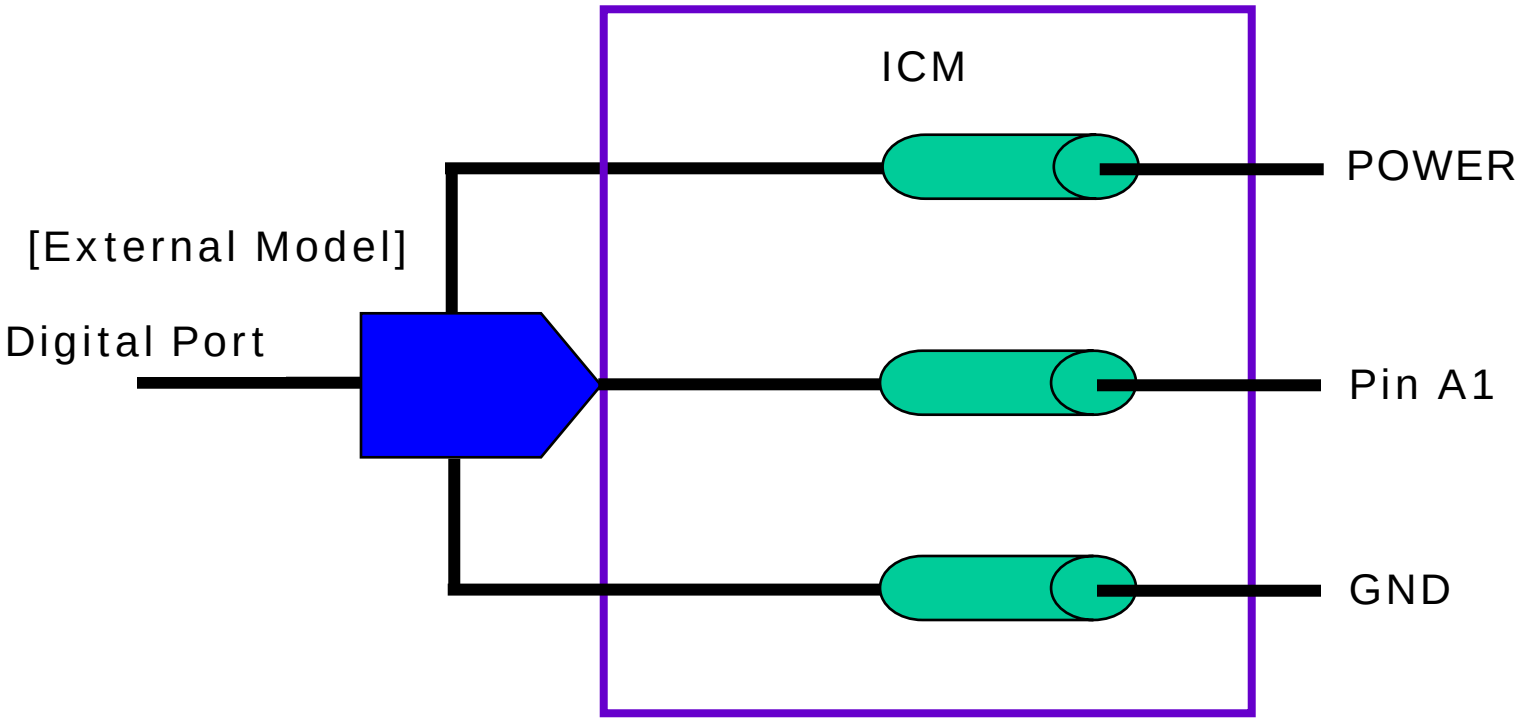
Four Cases

- **Case 2 – Wired-or or “mux” connections**
  - **Multiple pins, single [Model]**



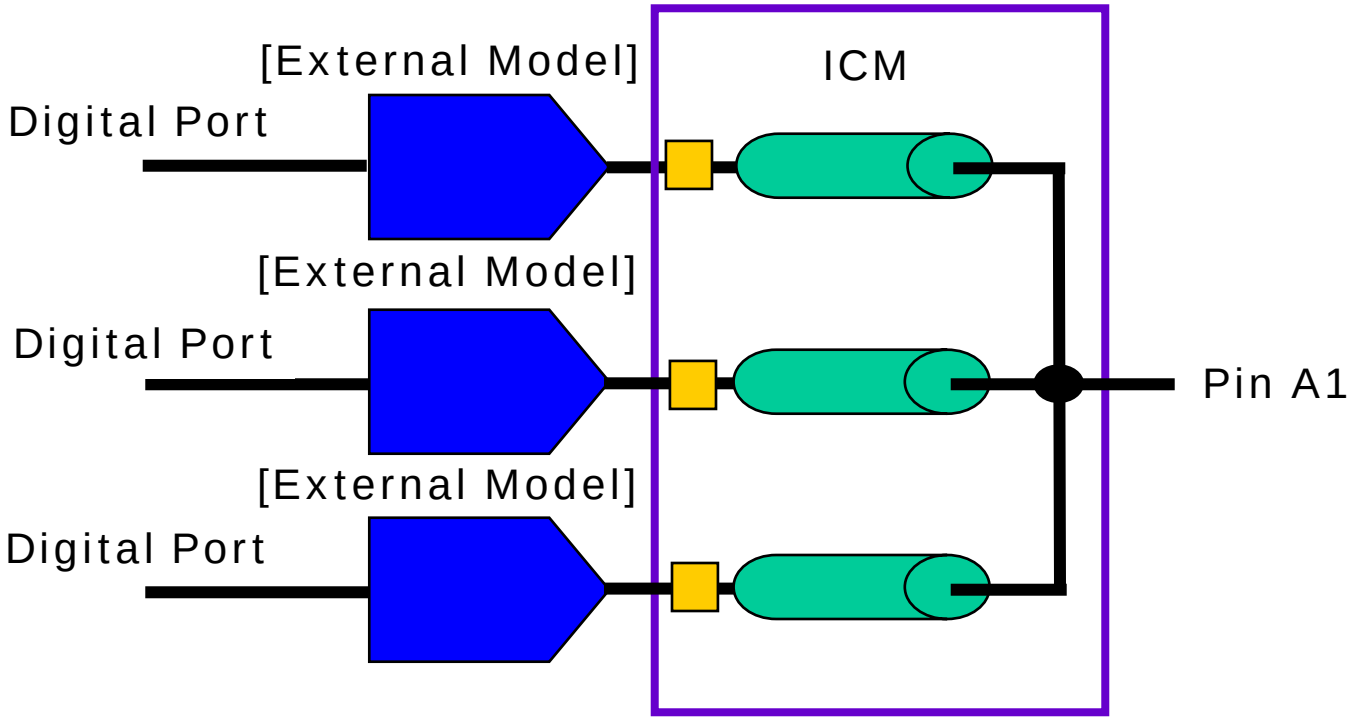
Four Cases

- **Case 3 – Coupling & power distribution**
  - **Single model, single signal pin**
  - **No different than Case 2, from [Pin] perspective**



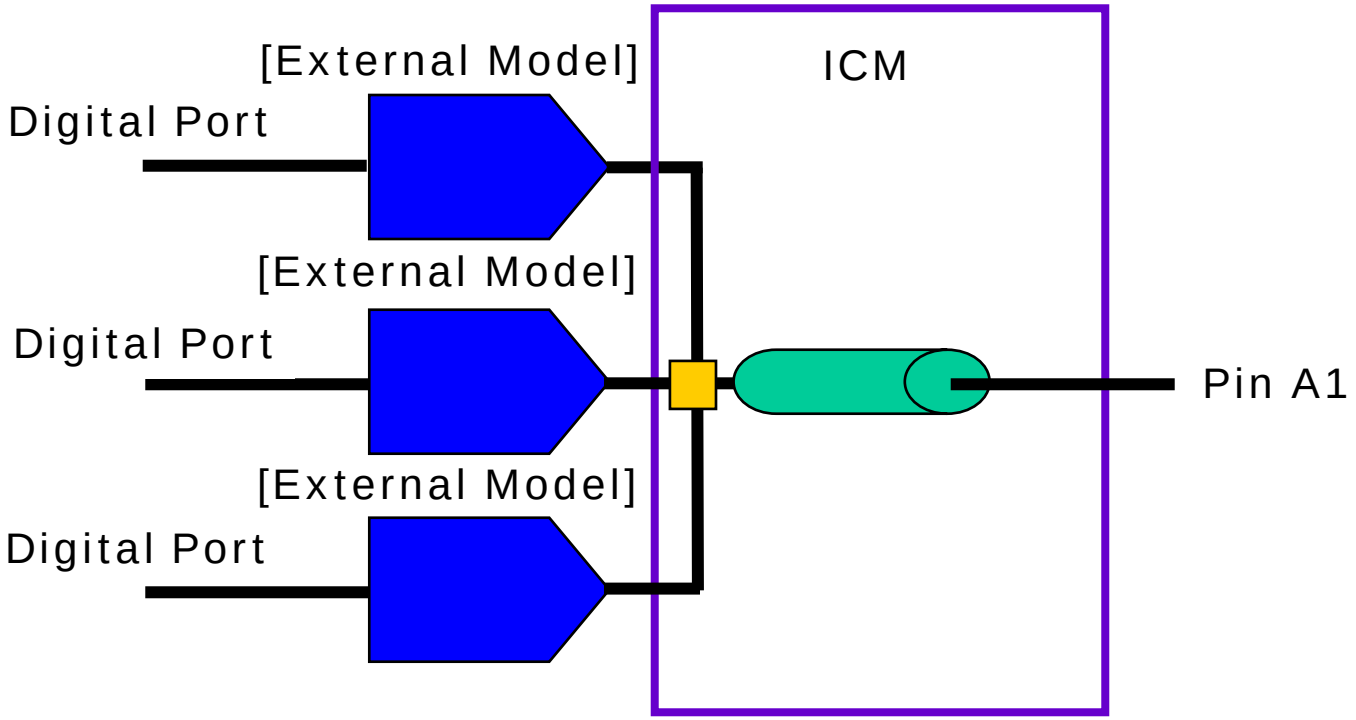
Four Cases

- **Case 4 – Wired-or or “mux” connection**
  - **Single pin, multiple [Model]s**



Four Cases

- **Case 4a – Wired-or or “mux” connection**
  - **Single pin, multiple [Model]s**
  - ***Variation: where are the package t-lines linked?***



Case 4 is the Biggest Problem!

- **Why is Case 4 problematic?**
- **[Pin]**
  - **The [Pin] keyword explicitly creates pins AND implicitly instantiates individual [Model] buffers**
  - **Each buffer has an individual die pad**
  - **Case 4 only has one [Pin] listing – one buffer, one pad!**
- **To function correctly with ICM, IBIS needs to enable separate instantiation of pins and buffers**

Expansion of [Pin]

- **Development from an idea by A. Muranyi**
- **Expand [Pin] to:**
  - **Permit reuse of [Pin] “number” in first column over more than one line**
  - **Add a column which names buffer for shared buffer instances!**
- **Concept:**
  - **Naming a [Pin] twice permits linking the same pin to multiple buffers**
  - **The “buffer_name” column gives us an explicit buffer instance identifier**
  - **Dot notation would add buffer name at end of full name**
  - **Unique pins still assume 1-1 association with buffers**
  - **Enables tools to track electrical association!**

Expansion of [Pin]

- **New [Pin] syntax examples**

```
[Pin] signal_name  model_name  R_pin  L_pin  C_pin  instance_name
A1    Power        POWER
A2    Ground       GND
A3    AD1          IO
A4    CLK          Input       NA     NA     NA     Input1
A4    CLK          Input       NA     NA     NA     Input2
|
| Pin A4 above is named CLK and is tied to TWO die pads/buffers.
| One is called Input1 and the other is called Input2.
| Both refer to the same [Model] in the IBIS file.
|
| The version below is also legal:

[Pin] signal_name  model_name  R_pin  L_pin  C_pin  instance_name
A1    Power        POWER
A2    Ground       GND
A3    AD1          IO
A4    CONTROL      Input       NA     NA     NA     InputControl
A4    CONTROL      Output      NA     NA     NA     OutputControl
|
| Pin A4 above is named CONTROL and is intended for use as a feedback buffer
| control.  This pin is tied to TWO die pads/buffers.  The buffers use different
| [Model]s in the IBIS file and are named InputControl and OutputControl.
```

Expansion of [Pin]

- **Multiple pins, same buffer**

```
[Pin] signal_name  model_name  R_pin  L_pin  C_pin  instance_name
A1    Power        POWER
A2    Ground       GND
A3    AD1          IO
A4    CLK1         Output       NA     NA     NA     DRIVERA
A5    CLK2         Output       NA     NA     NA     DRIVERA
|
|
| Pins A4 and A5 are physically separate pins, with distinct names. However,
| they are intended to be driven by the same buffer (a ganged output)
| In this case, the same model_name is used (same buffer design is used for both
| pins AND the same instance_name is used for each.  This shows that the same
| buffer instance is used for both pins, and not two instances of the same
| buffer design.
```

Pulling ICM in...

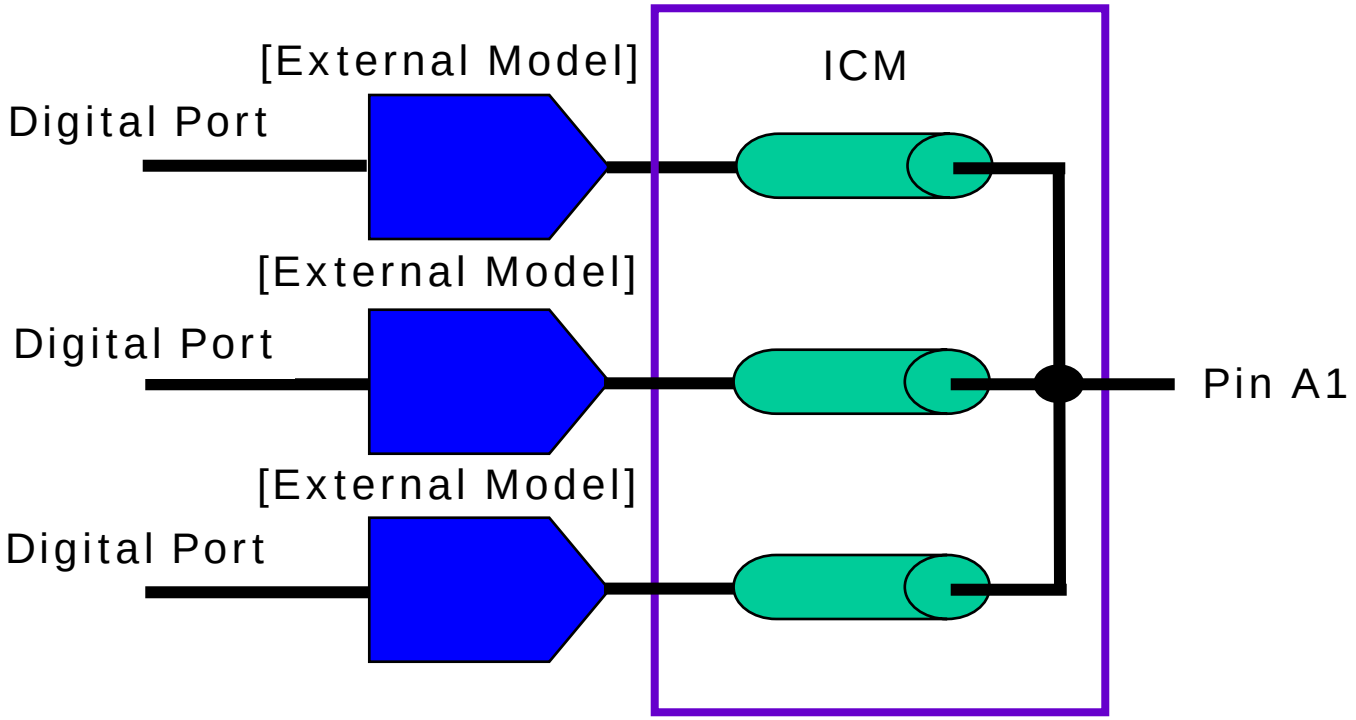
- **New proposal from Arpad Muranyi**
  - **Concept: assume 3.2 die pad names = 4.1 port names**
    - [Model] ports are implicitly defined in 4.1
    - Just make A_signal, A_puref, A_pdref, etc. accessible for 3.2 models
    - Instantiation is by component, pin name (one pin, one model)
  - **"Dot" syntax for names, tying ports to pins to nodes**
    - Use explicit names in the ICM file
    - Example:
      - *Component.pin_name in ICM on pinlist side*
      - *Component.pin_name.port_name on die side*
    - Resembles existing tool approach, to some degree
  - **Analog port names appear in ICM pin, node lists**
    - Dangling nodes? OK!
    - All connections are explicit (no tree path in this scheme)
    - Digital ports disallowed
  - **Advantages**
    - Can use current [Package Model] syntax
    - Can use ICM file just as we use PKG file
    - Permits power, ground path modeling
  - **Disadvantages**
    - Do we need ICM/IBIS parser integration?
    - [Pin Mapping] could potentially conflict

Curing Case 4

- **How would Case 4 be used with the new [Pin]?**
  - **Recall that we use "dot" syntax**
    - **Explicit, unique names for buffer die pad**
    - **Names tie ports to pins to nodes**
    - **FOR MULTIPLE PIN/BUFFER CONNECTIONS**
      - instance_name = unique buffer instance name
    - **Example:**
      - Component.pin_name in ICM on pinlist side
      - Component.pin_name.instance_name.port_name on die side
    - **FOR 1-1 CONNECTIONS**
      - Pin_name = unique buffer instance name
    - **Example:**
      - Component.pin_name in ICM on pinlist side
      - Component.pin_name.(noentry).port_name on die side
      - EDA Vendors: would the above work?

Four Cases

- **Case 4 – ICM expresses wired-or or “mux”**
  - **Single pin, multiple [Model]s**



Summary of Spec Changes

- **To implement ICM to IBIS link**
  - **Update [Pin] with instance_name column**
    - **Include new usage rules for multiple pins/multiple bfrs**
  - **Update [Pin Numbers] (IBIS PKG) for Buffers**
  - **Update [Package Model] to accept .icm files**
    - **How to refer to specific model in ICM file?**
    - **Should IBIS include dot notation names?**
- **This does not replace Circuit Call**
  - **New [Pin] only instantiates multiple model connections for “native” IBIS**
  - **Circuit Call connections need further work**
- **Issues to consider**
  - **Parsing: How to check dot notation between ICM and IBIS?**
  - **Conflicts with [Pin Mapping], [Series Pin Mapping], [Diff Pin]?**

BACKUP

[Pin] Rules

- **New syntax language rules for specification**

Rules: Each shared or repeated entry under column 1 of [Pin] must have the same signal_name as all other pins using the same column 1 entry. Example (1) shows such a case.

Any [Pin] which uses the same column 1 entry as another pin MUST have instance_name filled out, even if the model_name is different. Otherwise, an error is flagged, as in IBIS version 1.1 through 4.1.

R_pin, L_pin and C_pin entries for duplicated pins may be filled out numerically or may be NA. If instance_name is used, the R_pin, L_pin and C_pin columns MUST contain entries. Rules regarding precedence of [Package Model] and [Package] over [Pin] parasitic entries still apply even when instance_name is present.

No entry in the instance_name field is permitted for [Pin]s of type POWER, GND, NA or CIRCUITCALL.

Identical instance_name entries for different [Pin]s (different column 1 entries) with identical model_name entries are permitted. This signifies multiple pins connected to the same buffer instance. Example (3) shows such a case.

Package Modeling Today

```
|*****************************************************************
[IBIS Ver] 3.2
[File name] example.ibs
{...}
[Component] Example_chip
{...}
[Package Model] simple_package
|*****************************************************************
[Pin] signal_name  model_name   R_pin   L_pin  C_pin
1     IO1          io_buffer
2     IO2          io_buffer
3     IO3          io_buffer
|*****************************************************************
[Model]     io_buffer
Model_type  I/O
{...}
|*****************************************************************
|
[Define Package Model] simple_package
[Number of Pins] 3
|
[Pin Numbers]
A1 Len=1.2 L=2.0n C=0.5p R=0.05/
B1 Len=1.2 L=2.0n C=0.5p R=0.05/
C1 Len=1.2 L=2.0n C=0.5p R=0.05/
|
[End Package Model]
[End]
|*****************************************************************
```

Header

Pin/Model assignment

Model definition

Package Model definition/assignment

Package Modeling Today

- **A Few Oddities**
  - **Package Pin attachment**

"A package stub description starts at the connection to the die and ends at the point at which the package pin interfaces with the board or substrate the IC package is mounted on."

```
A1 Len=0 L=1.2n/ Len=1.2 L=2.0n C=0.5p/
Len=0 L=2.0n C=1.0p/
```

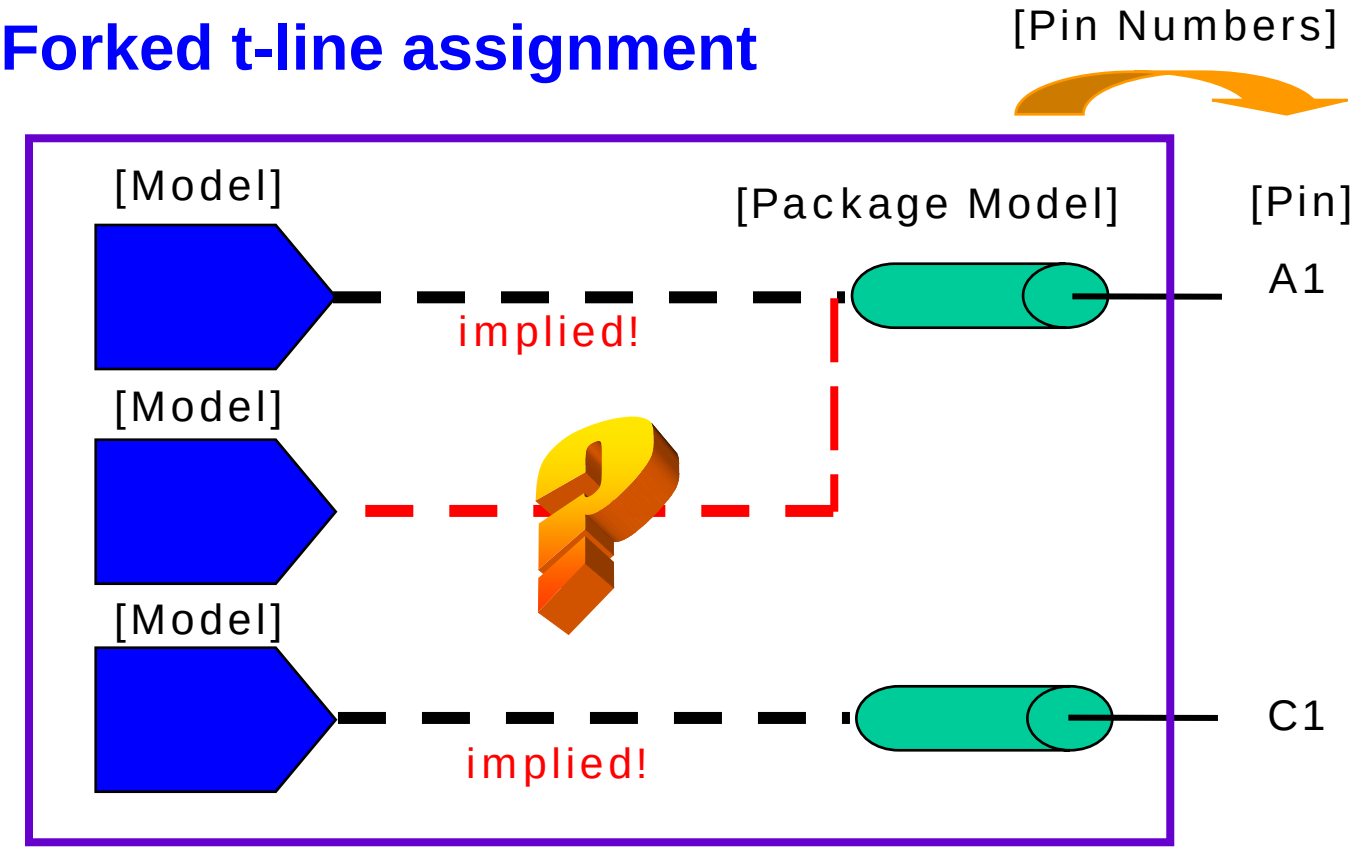
Pin is here!

  - **Package Pins vs. Fork/Endfork**

"The package pin is connected to the last section of a package stub description not surrounded by a Fork/Endfork statements."

What about this?

- **Forked t-line assignment**



- **This structure *cannot* be described using IBIS 3.2/4.0**
  - **A fork can only end as an unterminated stub**