# IBIS & ICM Interfacing Options Alternative Proposals

# IBIS & ICM

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

1. **IBIS 3.2/4.0 + ICM**
   - **Are we willing to limit the ICM models here to single-path, pad-to-pin?**
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**

# Item (3)

No spec. changes!

- 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
```

# Items (1) and (2)

- **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**
- **Some of this can be cured in IBIS 5.0**

# IBIS & ICM Links

- **Linking ICM to IBIS [Model]**
  - This would cover [External Model] too
  - Ultimate issues: [Model] ports have no names in 3.2/4.0
    - D_drive, etc. aren't actually used except in 4.1 extensions
    - Power supply connections handled in [Pin Mapping]
  - Need way to instantiate [Model] separately from [Pin]
    - Careful! Could enable "floating" [Model]
- **Options:**
  - New IBIS reserved word to separate [Model] from [Pin]
    - Also a keyword; example: ICMLINK
    - Similar to CIRCUITCALL in [Pin]
  - [ICMLINK] would explicitly name
    - ICM node/pin map, reserved port name, [Pin] name if any
  - Extended to cover ICE models/[External Circuit]s?

# [ICM Link] Example

IBIS

```
[ICM Link] ICM_model_n
Signal_pin A1
Model_name B     1
|
| port           M_pin/node
A_si     l           N_A1
A_p    ef          CONN  owerpd
A_    ef           CONN_po    pu

M    el_name Buf2
|
|   rt           ICM_pin/node
A_   gnal2         CONN_A2
[En  ICM Link]
```

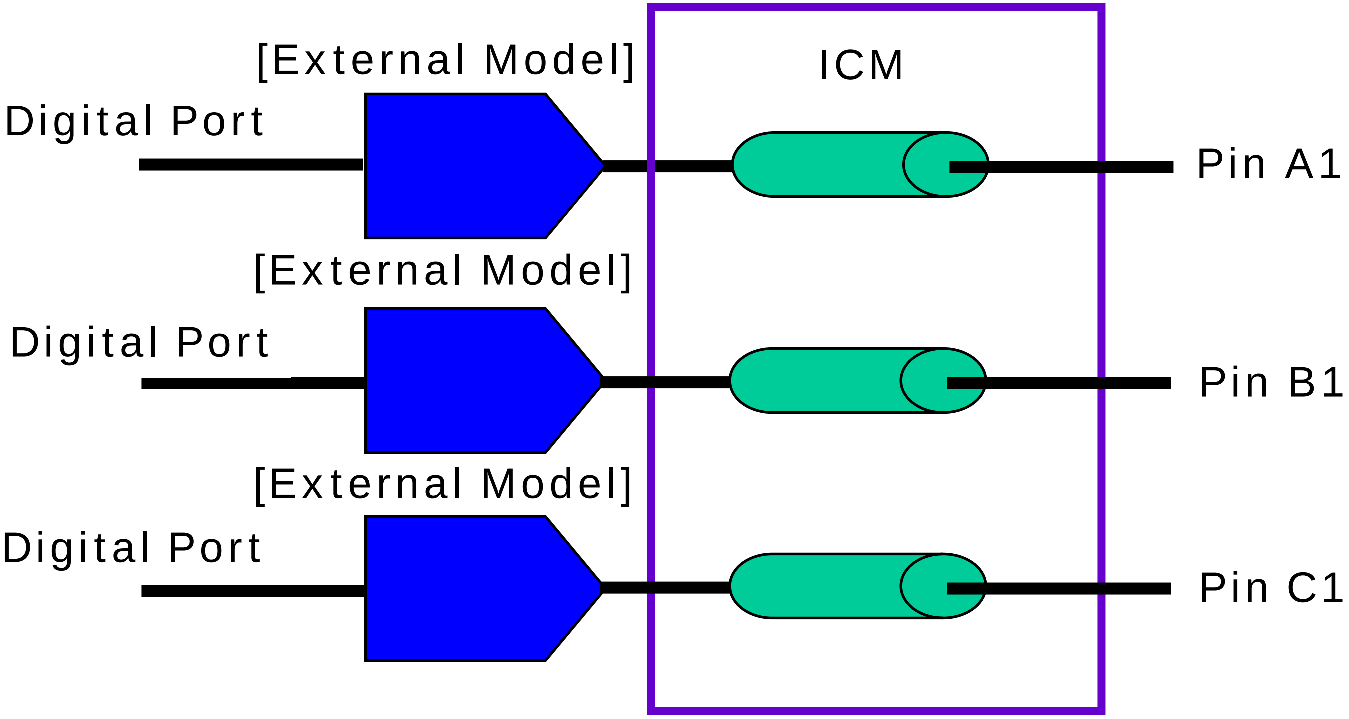
Assu      A1 is the only conne      n to the outside world

Stubbed model or mo       with no direct pin c        ection

- Format res      s two issues
  - Multiple [Model]      an now be linked with one ICM    kg
  - Stubs and dangling st     ctures can be included in ICM description without naming/connection in [ICM Link]
  - Permits instantiation of mu     ple      ces of the same [Model]
- Uglines
  - We have     st bypassed/r     placed [Pin]

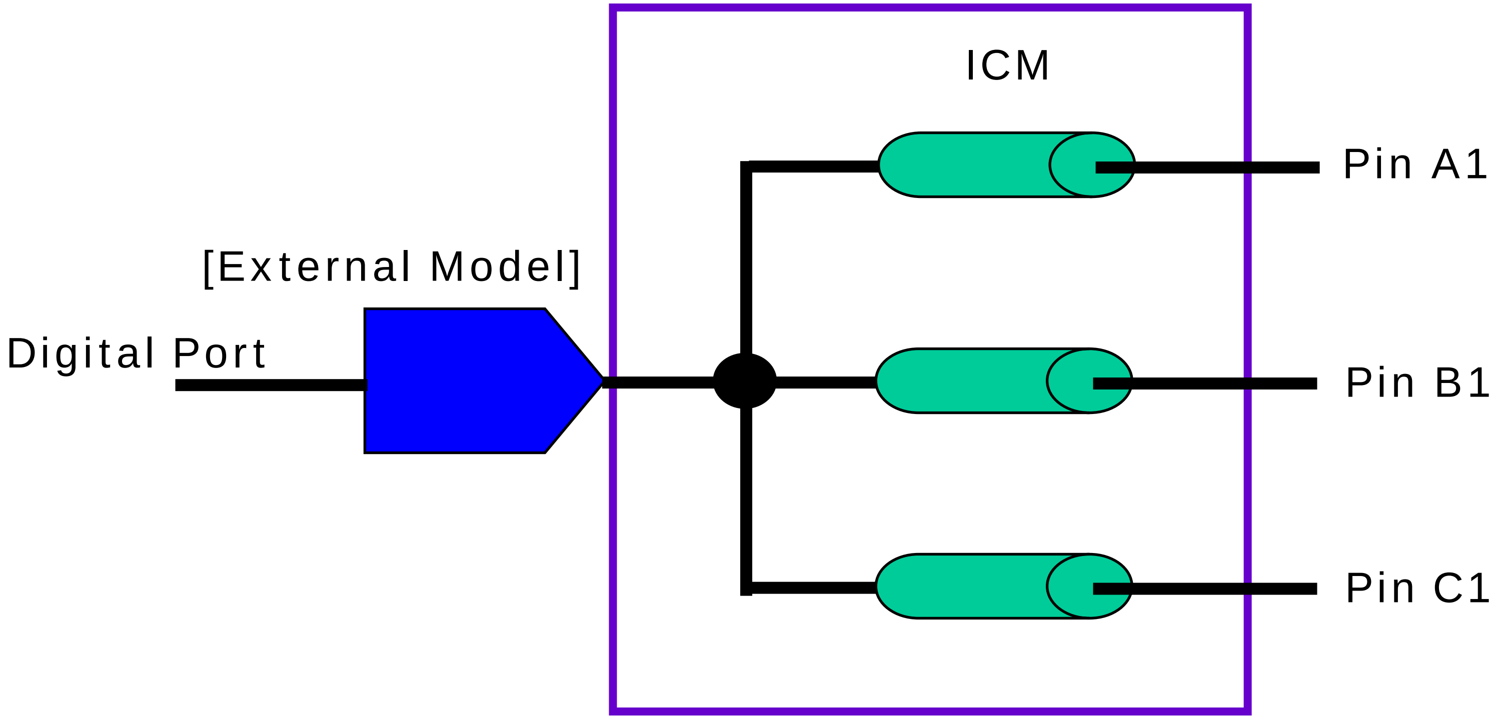
# Four Cases

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



# Four Cases

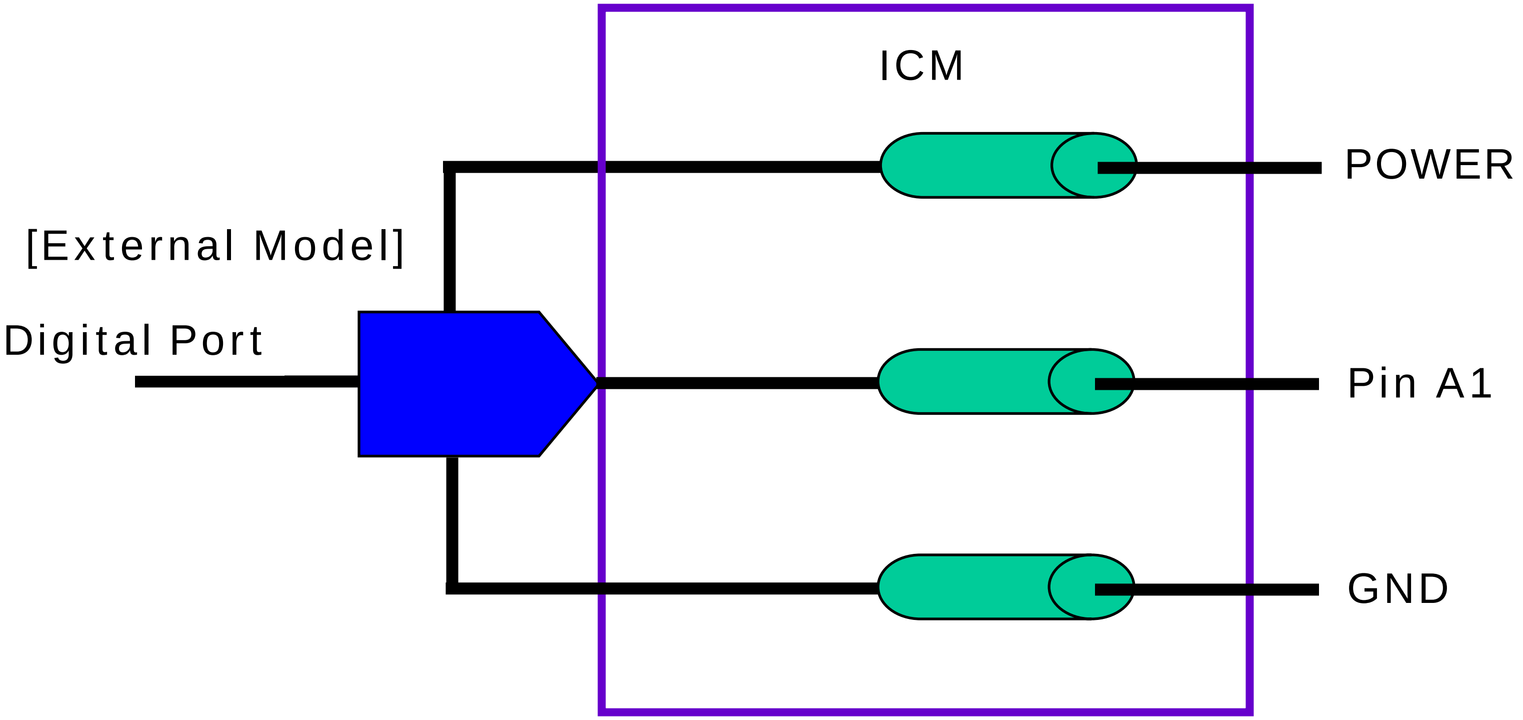
- **Case 2 – ICM expresses wired-or or “mux”**
  - **Multiple pins, single [Model]**



*Other brands and names are the property of their respective owners

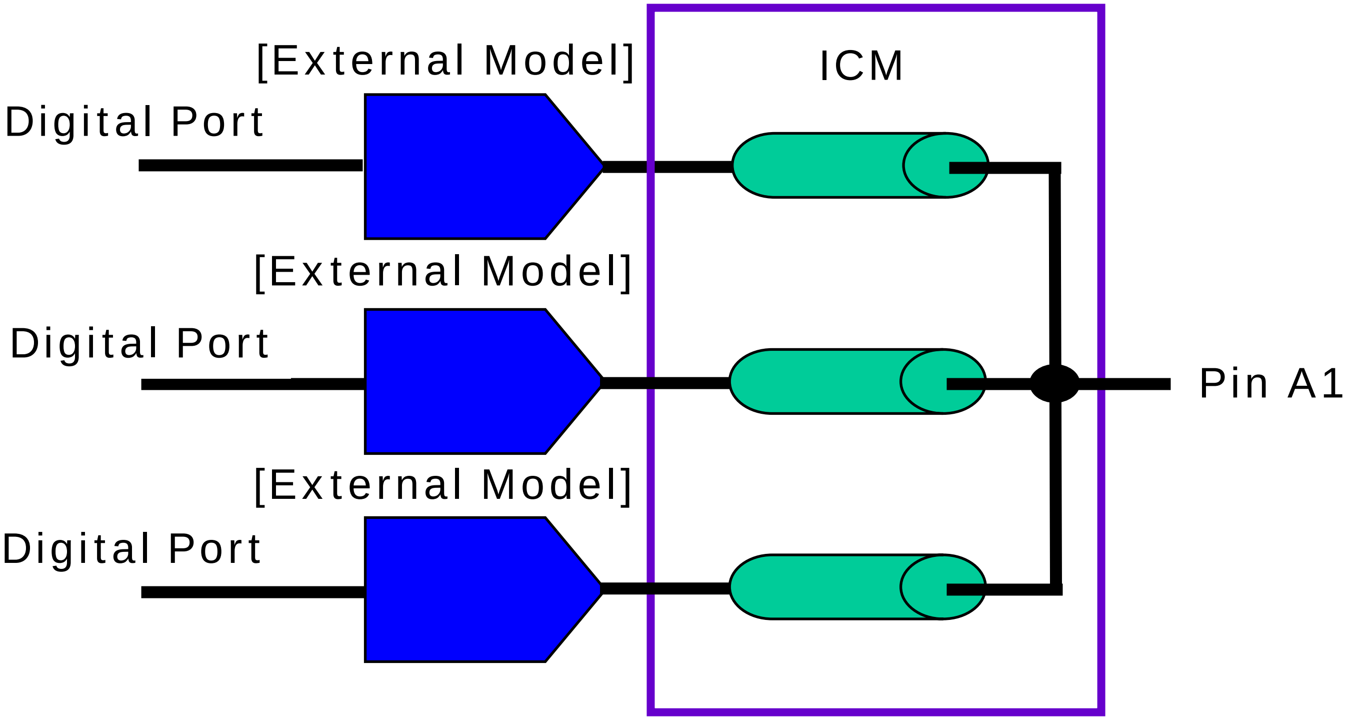
# Four Cases

- **Case 3 – ICM describes coupling & power distribution**
  - **Single model, single signal pin**



# Four Cases

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



# BACKUP

# Package Modeling Today

```
|**************************************************************************
[IBIS Ver] 3.2
[File name] example.ibs
{...}
[Component] Example_chip
{...}
[Package Model] simple_package
```

Header

```
|**************************************************************************
[Pin] signal_name  model_name   R_pin   L_pin  C_pin
1     IO1          io_buffer
2     IO2          io_buffer
3     IO3          io_buffer
```

Pin/Model assignment

```
|**************************************************************************
[Model]     io_buffer
Model_type  I/O
{...}
```

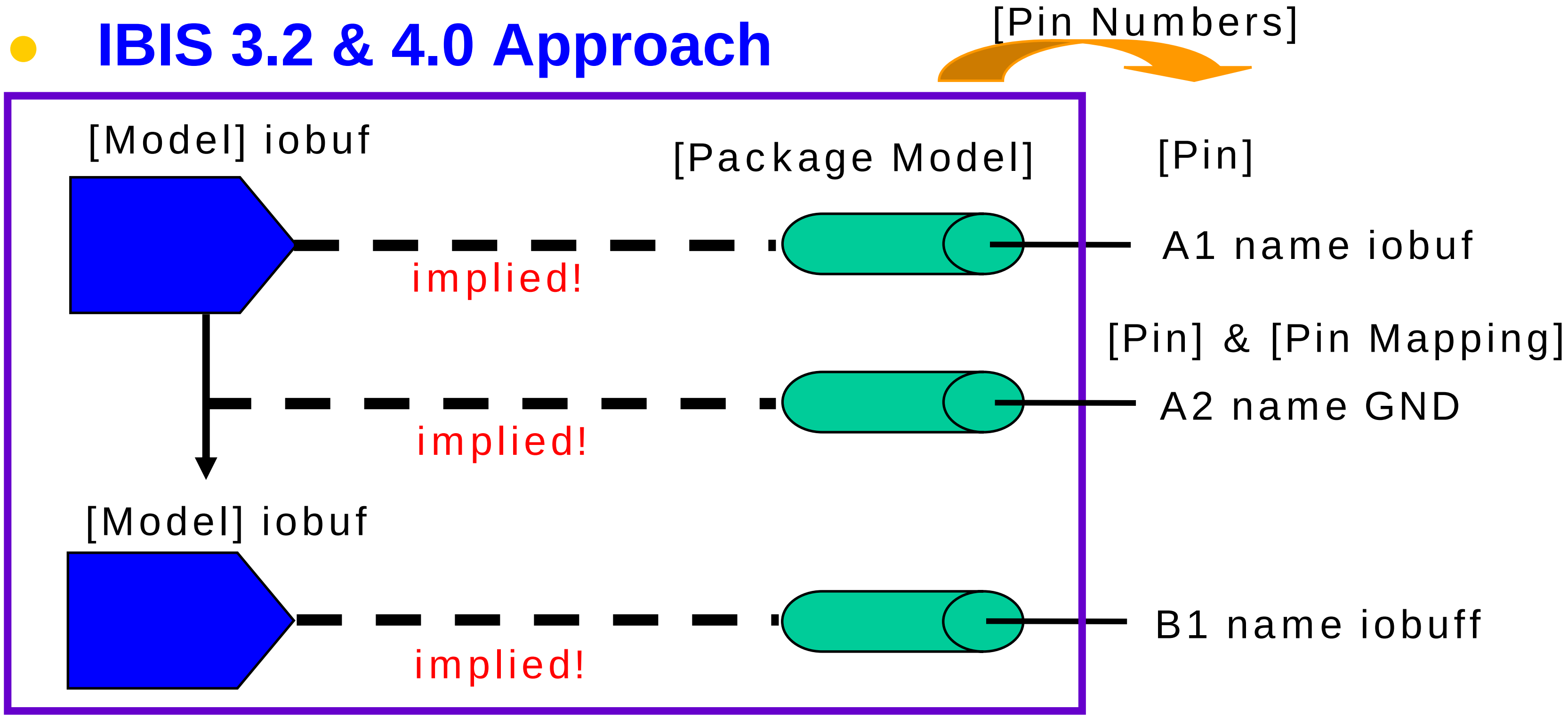
Model definition

```
|**************************************************************************
|
[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]
|**************************************************************************
```

Package Model definition/assignment

# 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**

# 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/
```

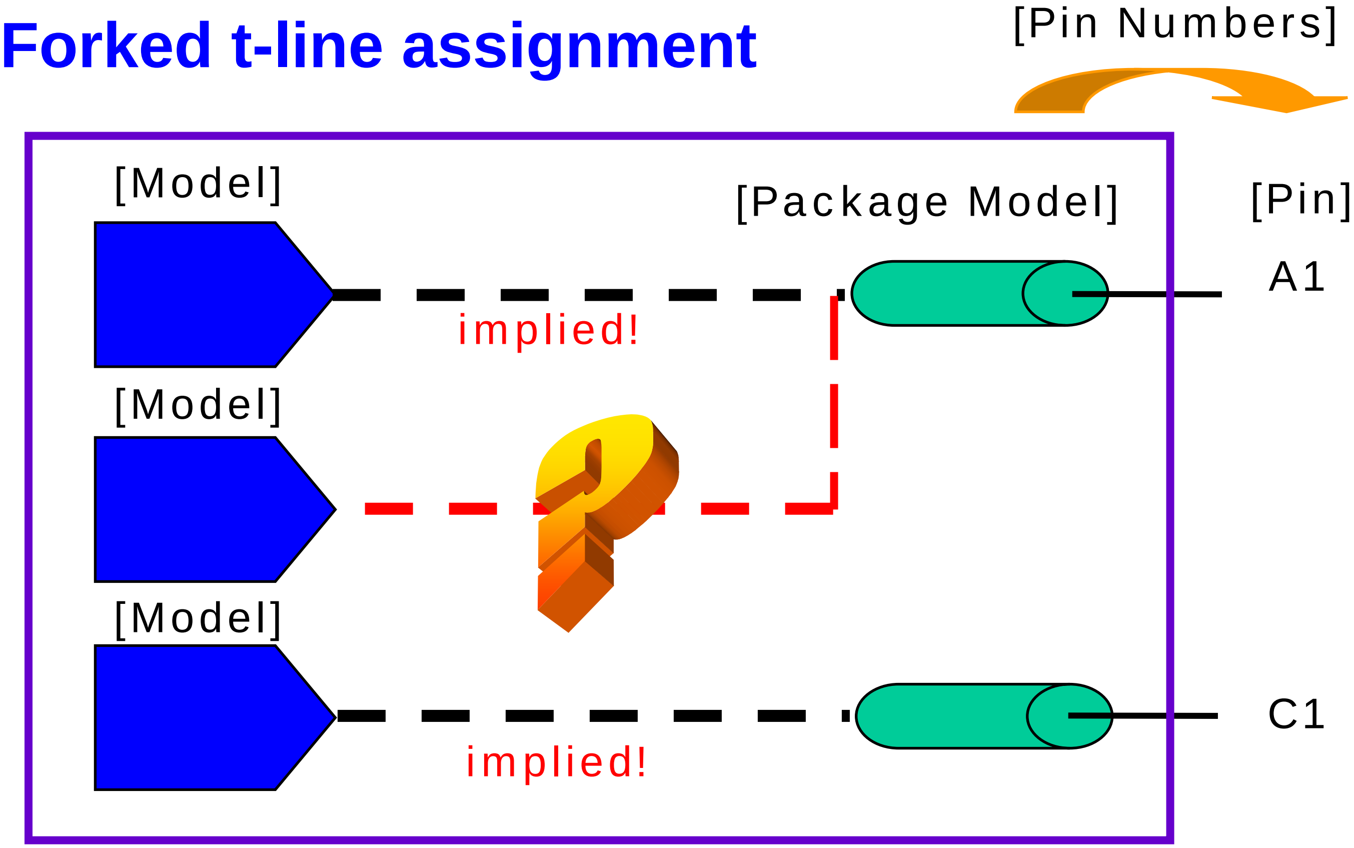
Pin is here!

  - **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**

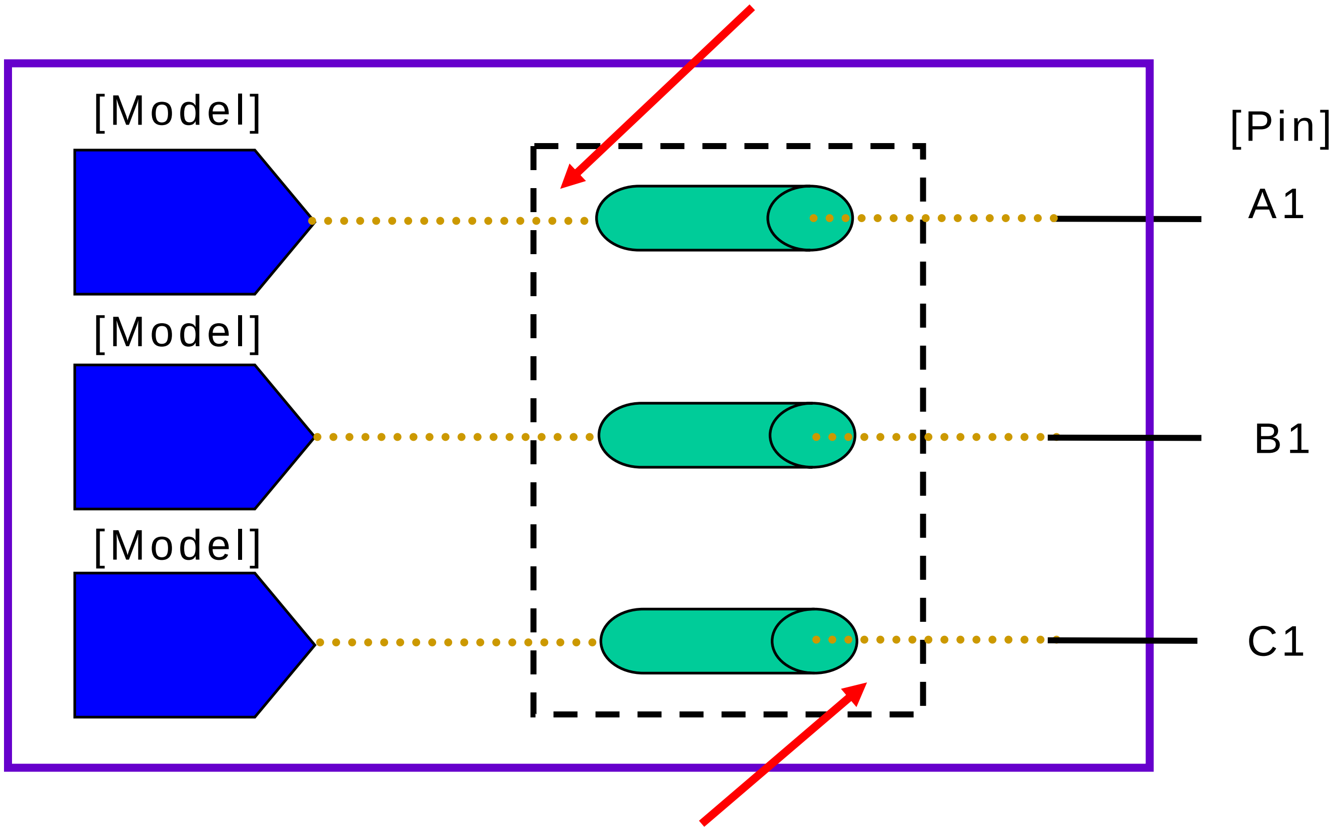


- **This structure *cannot* be described using IBIS 3.2/4.0**
  - **A fork can only end as an unterminated stub**

# What do we need?

- **The General Case...**

Need explicit link to [Model] instance



Need explicit link to [Pin] instance

# IBIS & ICM

- **How can we use ICM to describe packages?**
  - **ICM can describe…**
    - **interconnect RLGC or S-parameter characteristics**
    - **coupling, if present, between interconnect segments**
    - **pin (port) end-points and names**
  - **ICM does not describe…**
    - **connections between [Model], [Pin] and ICM end-points**
- **Changes Required**
  - **IBIS: need explicit link between [Model] and [Pin]**
    - **ICM can use node/pin map names from [Pin] listing**
    - **[Model] link options listed below**
  - **IBIS: explicit link between [E. Circuit] and [Pin]?**
    - **[Node Declarations]! See below**
  - **ICM: need differentiation between pin maps**
    - **Currently, same pin map may be used for all end-points**
    - **This is fixed in IIRD8 (Ross)**