IBIS-ISS Package Modeling

Walter Katz

SiSoft

IBIS ATM

August 7, 2012

SiSoft™
We Are Signal Integrity

Overview

- General Functionality Requirements
- Example .ibs File
- Supply Node Requirements
- On-Die Model Requirements
- Pin and Die Pad Data
- Rambus Forked Clock
- One Victim Package Models
  - By Model_name
  - By Pin_name
  - Signal Coupling
  - Signal/Power Coupling
  - Passing Pullup and Pulldown voltages
  - NEXT and FEXT Coupling
  - 9 Corners
  - Tstonefile
- Full Interface Package Models
- On-Die Models
- Bank Models
- Conclusion

General Functionality Requirements

- Package and On-Die Models
  - Package Models are between Pins and Die Pads
  - On-Die Models are between Di Pads and Buffers
  - Model maker may include On-Die Redistribution Layer (RDL) in the Package Model
  - Model maker may include On-Die Equalization in the Buffer Model (e.g. T-Coil in AMI Tstonefile)
  - Care must be taken when generating Package, On-die and [Model] models to avoid double counting
- Power Modeling
- Victim Models
  - Uncoupled
  - Crosstalk
    - FEXT and NEXT
- Interface Models
- Bank Models

Victim Model Functionality

- Victim package models are models for a specific (victim) channel.
  - For an uncoupled package models there would be 2 ports for a single ended channel and 4 ports for a differential channel. For coupled package models there would be 2 (or 4) ports for the primary, or victim channel, and additional ports for the aggressor channels.
  - The victim channel must accurately (sufficient accuracy for users to make engineering decisions) represent the insertion loss and return loss of the victim channel.
  - The aggressor channels may be less accurate than the victim channel, but must have sufficient accuracy to predict crosstalk on the victim channel for users to make engineering decisions on the victim channel.
  - These package models represent a very narrow slice of the complete package, and therefore only one pin node and one die node is required for each supply voltages.
  - Package models are associated to specific pins either by Pin_name, or by Model_name.
  - The on-die power model can be represented by a Cdie capacitance and an Rdie resistance. More complicated on-die power models may be required.

FEXT and NEXT Crosstalk

- The aggressor channels will normally be FEXT (Far End Cross Talk), although they can also be NEXT (Near End Cross Talk) channels.
  - The purpose of the cross talk channels is to allow the user to determine crosstalk noise on the victim channel.
  - NEXT is everything that is not FEXT.
  - FEXT is normally defined as coupling between two channels that both channels have their Tx on the same chip, and have the same driver model, and both channels have their Rx on the same second chip and have the same receiver model.
  - Since the package does not know what is at the other end of a channel, then for the purposes of package models NEXT are channels that do not have the same model as the victim. A package model can have only one NEXT model, but can have several NEXT aggressor channels using that model.

Interface and Bank Functionality Requirements

- The interface package model has multiple channels
  - All channels are both Victim and Aggressor.
  - There is no concept of Victim or Aggressor.
  - Multiple supply pin and die nodes are allowed.
- Bank models are sections of a package that contain multiple interfaces
- Additional functionality is required for on-die signal and power IBIS-ISS subckts. Most examples demonstrate functionality required for the package models. All of the functionality for package models also applies to on-die models.

Example to Define IBIS 6.0 Package Modeling Functionality Requirements

- The following .ibs file examples demonstrate the features that I believe are required in the IBIS 6.0 IBIS-ISS Packaging Solution. I am using a possible syntax. The intent is not to propose a specific syntax, but to use a possible syntax to demonstrate the functionality that I believe is required.
- Items in RED highlight the feature being demonstrated.
- Each example represents actual package models currently supplied by IC Vendors or requested by IC Vendors and EDA Vendors. I can (but may not in some cases without permission) give the name of specific IC and EDA Vendors.
- New functionality required in the ".ibs" file uses traditional IBIS keyword formats.
- The data in the new IBIS-ISS package sections uses a Parameter Tree syntax.

Supply Voltages for [Model]s and Package Pullup and Pulldown Resistors

- Each Supply Voltage is defined by its Signal_name
- For Victim Models
  - There can only one die node for each Supply Voltage
  - There can only one pin node for each Supply Voltage
- For Interface Models
  - There may be multiple die nodes for each Supply Voltage
    - May be a subset of all die supply pads
  - There may be multiple pin nodes for each Supply Voltage
    - May be a subset of all pin supply pins
- Need Supply Voltages for package pullup and pulldown resistors
- Each [Model] requires a pullup and pulldown supply
  - Package models do a redistribution of power and ground between pins and die, and on-die models do another redistribution to buffers.
  - [Pin Mapping] has similar function.

Supply voltages for Models when using Victim Package models

```
[Model] DQ
Pullup_Signal_name      VDD
Pulldown_Signal_name    VSS
[Model] DQS
Pullup_Signal_name      VDD
Pulldown_Signal_name    VSS
[Model] Tx
Pullup_Signal_name      VQQ
Pulldown_Signal_name    VSS
[Model] Rx
Pullup_Signal_name      VQQ
Pulldown_Signal_name    VSS
```

Supply voltages. Used to define voltages applied to package which may be different than [Voltage Range] used to generate [Models]. Also required to supply voltages to packages for]ackage pullup and pulldown resistors.

```
[Supply Signal_name]
| Signal_name           Voltage
VDD                     2.0 1.8 2.2
VQQ                     1.0 .9 1.1
VSS                     0.0 0.0 0.0
[End Supply Signal_name]
```

On-Die Power Model Requirements for Victim Channels

Simple On-Die capacitance and resistance of supply nodes

[Cdie Signal_name]
| Signal_name Cdie_typ Cdie_min Cdie_max
VDD 1.6pF 1.4pF 1.8pF
VQQ 1.4pF 1.2pF 1.6pF
VSS 2pF 1.8pF 2.2pF
[End Cdie Signal_name]

[Rdie Signal_name]
| Signal_name Rdie Rdie_min Rdie_max
VDD 0.20 NA NA
VQQ 0.15 NA NA
VSS 0.10 NA NA
[End Rdie Signal_name]

Does the on-die power model need to be more complicated, e.g. multiple Rdie, Cdie (RC Ladder Circuit) or a subckt?

Are different on-die power models required when simulating a single interface, or a slice of an interfac?

Example .ibs File

[Component]
[IBIS-ISS Package] Feature_<A:Z>
[IBIS-ISS On-Die] Feature_<A:Z>
[Pin]

|Pin_name	Signal_name	Model
A1	DQ1	DQ
A2	DQ2	DQ
A3	DQ3	DQ
A4	DQ4	DQ
A5	DQ5	DQ
A6	DQ6	DQ
A7	DQ7	DQ
A8	DQ8	DQ
A9	DQS+	DQS
A10	DQS-	DQS
A11	Tx1+	Tx
A12	Tx1-	Tx
A13	Tx2+	Tx
A14	Tx2-	Tx
A15	Tx3+	Tx
A16	Tx3-	Tx
A17	Rx1+	Rx
A18	Rx1-	Rx
A19	Rx2+	Rx
A20	Rx2-	Rx
A21	Rx3+	Rx
A22	Rx3-	Rx
B1	VQQ	POWER
B2	VQQ	POWER
B3	VDD	POWER
B4	VDD	POWER
B5	VSS	GND
B6	VSS	GND

[Diff Pin]

A9	A10
A11	A12
A13	A14
A15	A16
A17	A18
A19	A20
A21	A22

[Model] DQ

Pullup_Signal_name	VDD
Pulldown_Signal_name	VSS

[Model] DQS

Pullup_Signal_name	VDD
Pulldown_Signal_name	VSS

[Model] Tx

Pullup_Signal_name	VQQ
Pulldown_Signal_name	VSS

[Model] Rx

Pullup_Signal_name	VQQ
Pulldown_Signal_name	VSS

Pins and Die Data

[Pin Die Data] | Pin_X, Pin_Y, Die_X , Die_Y are optional

Units Inch

|Pin_name	Die_name	Pin_X	Pin_Y	Die_X	Die_Y
A1	D.A1	0.0	0	.0	0
A2	D.A2	0.05	0	.015	0
A3	D.A3	0.1	0	.01	0
…					
A17	D.A17	0.3	0.1	.03	.01
A18	D.A18	0.35	0.1	.035	.01
A19	D.A19	0.4	0.1	.04	.01
A20	D.A20	0.45	.1	.045	.01
A21	D.A21	0.5	.1	.005	.01
A22	D.A22	0.55	.1	.05	.01
B1	NA	0.	.2		
B2	NA	0.05	.2		
B3	NA	0.15	.2		
B4	NA	0.2	.2		
B5	NA	0.25	.2		
B6	NA	0.3	.2		

[End Pin Die Data]

D.A1, D.A2, … are implicit die pad names.

Supply Die Data

[Supply Die Data] | Die_X, Die_Y are optional
Units Inch

Die_name	Signal_name	Die_X	Die_Y
S1	VQQ	0.0	.02
S2	VQQ	0.005	.02
S3	VQQ	0.01	.02
S4	VQQ	0.015	.02
S5	VQQ	0.02	.02
S6	VQQ	0.025	.02
S7	VDD	0.03	.02
S8	VDD	0.035	.02
S9	VDD	0.04	.02
S10	VDD	0.045	.02
S11	VDD	0.0	.025
T1	VSS	0.05	.025
T2	VSS	0.01	.025
T3	VSS	0.015	.025
T4	VSS	0.02	.025
T5	VSS	0.025	.025
T6	VSS	0.03	.025
T7	VSS	0.035	.025
T8	VSS	0.04	.025

[End Supply Die XY]

S1, S2, .., T1, T2, … are explicit die pad names.

Example IBIS-ISS Package Models Demonstrating Feature Requirements

- Each of the following slides demonstrate the data in a [IBIS-ISS Package] Feature_<A:Z> to [End IBIS-ISS Package] section.
- Rambus Forked Clock
- One Victim Package Models
  - By Model_name
  - By Pin_name
  - Signal Coupling
  - Signal/Power Coupling
  - Passing Pullup and Pulldown voltages
  - NEXT and FEXT Coupling
  - 9 Corners
  - Tstonefile
- Full Interface Package Models
- On-Die Models
  - Uncoupled
  - Coupled Signal and Power
- Bank Models

Rambus Forked Clock

xyz.ibs

```
[Component]
[IBIS-ISS Package] Forked.ipkg
[IBIS-ISS On-Die] On-Die.idie
[Pin]
|Pin_name    Signal_name Model
A1           CLKin+      CLK
A2           CLKin -     CLK
A3           CLKout +    CLK
A4           CLKout -    CLK
[Diff Pin]
A1           A2
A3           A4
[Pin Die Data]  |
|Pin_name    Die_name
A1           D.A1
A2           D.A2
A3           D.A1
A4           D.A2
| Note D.A1 and D.A2 are implicit Die_names
| Note D.A1 and D.A2 are explicit Die_names
[End Pin Die Data]
[Model] CLK
```

Forked.ipkg

```
(Package_models
      (Xclk
            (File          xclk.mod)
            (subckt        xclk)
            (Ports     A1 A2 A3 A4 D.A1 D.A2)))
```

On-Die.idie

```
(On-Die_model_Assignments
     ((Model_name CLK) (On-Die_model CLKx)))
(On-Die_models
     (CLKx  (Tstonefile  clkx.s4p)
     (Ports     Buffer+ Pad+ Buffer- Pad-))
```

xclk.mod

```
.subckt xclk inP inM outP outM dieP dieM
…
.ends xclk
```

By Model_name

```
(Package_model_Assignments
        (1  (Model_name DQ)             (Package_model pkg1))
        (2  (Model_name DQS)            (Package_model pkg2))
        (3  (Model_name Tx)             (Package_model pkg2))
        (4  (Model_name Rx)             (Package_model pkg2))
)
(Package_models
        (pkg1
                (File           pkg1.mod)
                (Subckt         (Corner pkg1_typ pkg1_min pkg1_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin Pad)   | Reserved port keywords
        )
        (pkg2
                (File           pkg2.mod)
                (Subckt         (Corner pkg2_typ pkg2_min pkg2_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin+ Pad+ Pin- Pad-)
        )
)
```

By Pin_name

```
(Package_model_Assignments
        (1  (Pin_name A1)    (Package_model pkg1)     (Length 11))
        (2  (Pin_name A2)    (Package_model pkg1)     (Length 12))
        (3  (Pin_name A3)    (Package_model pkg1)     (Length 14))
        (4  (Pin_name A4)    (Package_model pkg1)     (Length 14))
        (5  (Pin_name A5)    (Package_model pkg1)     (Length 11))
        (6  (Pin_name A6)    (Package_model pkg1)     (Length 21))
        (7  (Pin_name A7)    (Package_model pkg1)     (Length 18))
        (8  (Pin_name A8)    (Package_model pkg1)     (Length 11))
        (9  (Pin_name A9)    (Package_model pkg2)     (Length 15))
        (10  (Pin_name A11) (Package_model pkg2)     (Length 21))
        (11  (Pin_name A13) (Package_model pkg2)     (Length 21))
        (12  (Pin_name A15) (Package_model pkg2)     (Length 21))
        (13  (Pin_name A17) (Package_model pkg2)     (Length 11))
        (14  (Pin_name A19) (Package_model pkg2)     (Length 11))
        (15  (Pin_name A21) (Package_model pkg2)     (Length 11))
)
(Package_models
        | See previous slide
)
| Differential Pin_names only reference Active High Pin_name.
```

Signal Coupling

```
(Package_model_Assignments
        (1  (Model_name DQ)     (Package_model pkg1))
        (2  (Model_name DQS)    (Package_model pkg2))
        (3  (Model_name Tx)     (Package_model pkg3))
        (4  (Model_name Rx)     (Package_model pkg3))
)
(Package_models
        (pkg1
                (File           pkg1.mod)
                (Subckt         (Corner pkg1_typ pkg1_min pkg1_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin Pad Pin1 Pad1 Pin2 Pad2)
        )
        (pkg2
                (File           pkg2.mod)
                (Subckt         (Corner pkg2_typ pkg2_min pkg2_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin+ Pad+ Pin- Pad-)
        )
        (pkg3
                (File           pkg3.mod)
                (Subckt         (Corner pkg3_typ pkg3_min pkg3_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin+ Pad+ Pin- Pad- Pin1+ Pad1+ Pin1- Pad1- Pin2+ Pad2+ Pin2- Pad2-)
        )
)
```

Signal and Power Coupling

```
(Package_model_Assignments
        (1  (Model_name DQ)     (Package_model pkg1))
        (2  (Model_name DQS)    (Package_model pkg2))
        (3  (Model_name Tx)     (Package_model pkg3))
        (4  (Model_name Rx)     (Package_model pkg3))
)
(Package_models
        (pkg1
                (File           pkg1.mod)
                (Subckt         (Corner pkg1_typ pkg1_min pkg1_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin Pad Pin1 Pad1 Pin2 Pad2 Pad.VDD  Pin.VDD  Pad.VSS  Pin.VSS)
        )
        (pkg2
                (File           pkg2.mod)
                (Subckt         (Corner pkg2_typ pkg2_min pkg2_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin+ Pad+ Pin- Pad- Pad.VDD  Pin.VDD  Pad.VSS  Pin.VSS)
        )
        (pkg3
                (File           pkg3.mod)
                (Subckt         (Corner pkg3_typ pkg3_min pkg3_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin+ Pad+ Pin- Pad- Pin1+ Pad1+ Pin1- Pad1- Pin2+ Pad2+ Pin2- Pad2-
                                Pad.VQQ  Pin.VQQ  Pad.VSS  Pin.VSS)
        )
)
```

Pullup and Pulldown Voltages

```
(Package_model_Assignments
        (1  (Model_name DQ)             (Package_model pkg1))
        (2  (Model_name DQS)            (Package_model pkg2))
        (3  (Model_name Tx)             (Package_model pkg2))
        (4  (Model_name Rx)             (Package_model pkg2))
)
(Package_models
        (pkg1
                (File           pkg1.mod)
                (Subckt         (Corner pkg1_typ pkg1_min pkg1_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin Pad VDD VSS)
        )
        (pkg2
                (File           pkg2.mod)
                (Subckt         (Corner pkg2_typ pkg2_min pkg2_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin+ Pad+ Pin- Pad-)
        )
)
```

NEXT Coupling

```
(Package_model_Assignments
        (1  (Model_name DQ)     (Package_model pkg1)    (NEXT_Model_name DQS))
        (2  (Model_name DQS)    (Package_model pkg2))
        (3  (Model_name Tx)     (Package_model pkg3))
        (4  (Model_name Rx)     (Package_model pkg3))
)
(Package_models
        (pkg1
                (File           pkg1.mod)
                (Subckt         (Corner pkg1_typ pkg1_min pkg1_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin Pad Pin1 Pad1 Pin2 Pad2 NEXT_Pin3+ NEXT_Pin3+ NEXT_Pin3- NEXT_Pin3-)
        )
        (pkg2
                (File           pkg2.mod)
                (Subckt         (Corner pkg2_typ pkg2_min pkg2_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin+ Pad+ Pin- Pad-)
        )
        (pkg3
                (File           pkg3.mod)
                (Subckt         (Corner pkg3_typ pkg3_min pkg3_max))
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin+ Pad+ Pin- Pad- Pin1+ Pad1+ Pin1- Pad1- Pin2+ Pad2+ Pin2- Pad2-)
        )
)
```

>3 corners (9 in this example)

```
(Package_model_Assignments
        (1  (Model_name DQ)          (Package_model pkg1))
        (2  (Model_name DQS)         (Package_model pkg2))
        (3  (Model_name Tx)          (Package_model pkg2))
        (4  (Model_name Rx)          (Package_model pkg2))
)
(Package_models
        (pkg1
                (Xtalk          (List   typ min max))
                (Zo             (List   typ min max))
                (File           pkg1_{Xtalk}_{Zo}.mod)
                (Subckt         pkg1_{Xtalk}_{Zo})
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin Pad)
        )
        (pkg2
                (Xtalk          (List   typ min max))
                (Zo             (List   typ min max))
                (File           pkg2_{Xtalk}_{Zo}.mod)
                (Subckt         pkg2_{Xtalk}_{Zo})
                (Length         (Range 15 11 21))
                (Parameters     Length)
                (Ports          Pin+ Pad+ Pin- Pad-)
        )
)
```

Tstonefile Directly

```
(Package_model_Assignments
        (1  (Model_name DQ)             (Package_model pkg1))
        (2  (Model_name DQS)            (Package_model pkg2))
        (3  (Model_name Tx)             (Package_model pkg2))
        (4  (Model_name Rx)             (Package_model pkg2)))
(Package_models
        (pkg1
                (Tstonefile     (Corner pkg1_typ.s2p  pkg1_min.s2p  pkg1_max.s2p))
                (Ports          Pin Pad))
        (pkg2
                (Tstonefile     (Corner pkg2_typ.s4p  pkg2_min.s4p  pkg2_max.s4p))
                (Ports          Pin+ Pad+ Pin- Pad-)))
```

Note, (Tstonefile pkg2_typ.s4p) is equivalent to:

```
        (File pkg2_typ.mod)
        (Subckt pkg2_typ)
        pkg2_typ.mod
                .subckt pkg2_typ 1 2 3 4
                S1 1 2 3 4 0 mname= pkg2_typ
                .model pkg2_typ S N=24 Tstonefile= pkg2_typ.s4p
                .ends pkg2_typ
```

Interface Using Package “Quadrants”

```
(Package_models
    (DDR3
        (File       DDR3.mod)
        (Subckt     DDR3)
        (Ports      A1 A2 A3 A4 A5  A6 A7  A8 A9 A10
                    D.A1 D.A2 D.A3 D.A4 D.A5  D.A6 D.A7  D.A8 D.A9 D.A10
                    B1 B5 D.B1 D.B5))
    (SerDes
        (Tstonefile serdes.s24p)
        (Ports      A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22
                    D.A11 D.A12 D.A13 D.A14 D.A15 D. A16
                    D.A17 D.A18 D.A19 D.A20 D.A21 D.A22)))
```

Interface Using Full Package Models

```
(Package_models
    (DDR3 (Tstonefile      package.s65p)
        (Ports             A1 A2 A3 A4 A5  A6 A7  A8 A9 A10
                           D.A1 D.A2 D.A3 D.A4 D.A5  D.A6 D.A7  D.A8 D.A9 D.A10
                           D.S8 NC NC NC   B4 NC      | VDD
                           D.T3 NC NC NC   B5 NC      | VSS
                           NC NC NC NC     NC NC      | VQQ
                           NC NC NC NC NC NC NC NC NC NC NC NC   | A<11:22>
                           NC NC NC NC NC NC NC NC NC NC NC NC)  | D.A<11:22>
    ) ) | model contains 2 Pin ports, and 4 Pad ports for each supply
    (SerDes  (Tstonefile     package.s65p)
          (Ports             NC NC NC NC NC NC NC NC NC NC
                             NC NC NC NC NC NC NC NC NC NC
                             NC NC NC NC      NC NC    | VDD
                             NC NC D.T7 NC    NC B6    | VSS
                             NC D.S3 NC NC    B1 NC    | VQQ
                             A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22
                             D.A11 D.A12 D.A13 D.A14 D.A15 D.A16
                             D.A17 D.A18 D.A19 D.A20 D.A21 D.A22)))
```

Notes:

Tools may create DDR3.s24p and SerDes.28p from package.s65p. This is very important for large packages which could be multi Giga Byte sNp (N ~2000) Tstonefiles.

Touchstone 2.0 should be enhanced to support binary format organized by Sij instead of frequency to allow efficient random access IO to extract an snp from an sNp where n<<N.

For large sNp (e.g. s1000p) need a Sparse Ports method.

Interface Using Full Package Models, Sparse Port Order

```
(Package_models
    (DDR3 (Tstonefile              package.s1000p)
        (Sparse_Ports
            (A1 1) (A2 2) (A3 3) (A4 4) (A5 5) (A6 6) (A7 7) (A8 8) (A9 9) (A10 10)
            (D.A1 401)(D.A2 402)(D.A3 403)(D.A4 404)(D.A5 405)
            (D.A6 406)(D.A7 407)(D.A8 408)( D.A9 409)( D.A10 410)
            (D.S8 801)(B4 803)(D.T3 803)(B5 804)
        )
    )
)
```

Tool generates an s24p from ports 1 2 3 4 5 6 7 8 9 10 401 402 403 404 405 406 407 408 409 410 801 802 803 804 805 of s1000p.

Interface Using Separate Signal and Power Models

```
(Package_models
    (DDR3
        (Signal
            (Tstonefile DDR3_Signal.s20p)
            (Ports      A1 A2 A3 A4 A5  A6 A7  A8 A9 A10
                        D.A1 D.A2 D.A3 D.A4 D.A5  D.A6 D.A7  D.A8 D.A9 D.A10))
        ( Power
            (Tstonefile DDR3_Power.s4p)
            (Ports      D.S8 B4 D.T3 B5))
    )
    (SerDes
        (Signal
            (Tstonefile SerDes_Signal.s24p)
            (Ports      A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22
                        D.A11 D.A12 D.A13 D.A14 D.A15 D.A16
                        D.A17 D.A18 D.A19 D.A20 D.A21 D.A22))
        ( Power
            (Tstonefile SerDes_Power.s4p)
            (Ports      D.T7 B6 D.S3 B1 ))
    )
)
```

On-Die by Model, Uncoupled

```
(On-Die_model_Assignments
        (1  (Model_name DQ)                         (On-Die_model  DQ))
        (2  (Model_name DQS)            (On-Die_model  DQS))
        (3  (Model_name Tx)                         (On-Die_model  SerDesTx))
        (4  (Model_name Rx)                         (On-Die_model  SerDesRx))
)
(On-Die_models
        (DQ
                (Tstonefile     dq.s2p)
                (Ports          Pad Buffer))
        (DQS
                (Tstonefile     dqs.s4p)
                (Ports          Buffer+ Pad+ Buffer- Pad-))
        (SerDesTx
                (Tstonefile     serdestx.s4p)
                (Ports          Buffer+ Pad+ Buffer- Pad-))
        (SerDesRx
                (Tstonefile     serdesrx.s4p)
                (Ports          Buffer+ Pad+ Buffer- Pad-))
)
```

On-Die Models Signal and Power Coupling

```
(Package_model_Assignments
      (1  (Model_name DQ)                    (On-Die_model  DQ))
      (2  (Model_name DQS)                   (On-Die_model  DQS))
      (3  (Model_name Tx)                    (On-Die_model  SerDesTx))
      (4  (Model_name Rx)                    (On-Die_model  SerDesRx))
)
(Package_models
      (DQ
            (Tstonefile     dq.s10p)
            (Ports          Buffer Pad Buffer1 Pad1 Buffer2 Pad2
                            Pad.VDD  Buffer.VDD Pad.VSS  Buffer.VSS))
      (DQS
            (Tstonefile     dqs.s8p)
            (Ports          Buffer+ Pad+ Buffer- Pad- Pad2
                            Pad.VDD  Buffer.VDD Pad.VSS  Buffer.VSS))
      (SerDesTx
            (Tstonefile     serdestx.s16p)
            (Ports          Pin+ Pad+ Pin- Pad- Pin1+ Pad1+ Pin1- Pad1- Pin2+ Pad2+ Pin2- Pad2-
                            Pad.VQQ Pin.VQQ Pad.VSS Pin.VSS))
      (SerDesRx
            (Tstonefile     serdesrx.s16p)
            (Ports          Buffer+ Pad+ Buffer- Pad-
                            Pad.VQQ Pin.VQQ Pad.VSS Pin.VSS))
)
```

.ibs File with Multiple Banks

```
[Component]
[IBIS-ISS Package] Banks_pkg
[IBIS-ISS On-Die] Banks_die
[Pin]
|Pin_name      Signal_name     Model
A1             DQ1             DQ
A2             DQ2             DQ
…
A44            Rx3-            SataRx
B1             VQQ             POWER
…
B12            VSS             GND
[Diff Pin]
A9             A10
,,,
A43            A44
[Model] DQ
Pullup_Signal_name             VDD
[Model] DQS
Pullup_Signal_name             VDD
[Model] Tx
Pullup_Signal_name             VQQ
[Model] Rx
Pullup_Signal_name             VQQ
[Model] SataTx
Pullup_Signal_name             VQQ
[Model] SataRx
Pullup_Signal_name             VQQ
```

Multiple Banks, Package and On-Die

```
(Package_models
        (Bank1
            (Signal
                (Tstonefile Bank1_pkg.s44p)
                (Ports          A1 A2 … A22    D.A1 D.A2 … D.A22))
            ( Power
                (Tstonefile Bank1_pkg_pwr.s6p)
                (Ports          B1 B2 B3 B4    Pad.VDD Pad.VQQ)))
        (Bank2
            (Signal
                (Tstonefile Bank2_pkg.s44p)
                (Ports          A23 A24 … A44    D.A23 D.A24 … D.A44))
            ( Power
                (Tstonefile Bank2_pkg_pwr.s6p)
                (Ports          B6 B7 B8 B9 Pad.VDD Pad.VQQ))))
(On-Die_models
        (Bank1
            (Signal
                (Tstonefile Bank1_die.s44p)
                (Ports          D.A1 D.A2 … D.A22   Buffer.A1 Buffer.A2 … Buffer.A22 ))
            ( Power
                (Tstonefile Bank1_die_pwr.s4p)
                (Ports          Pad.VDD Pad.VSS Buffer.VDD Buffer.VQQ )))
        (Bank2
            (Signal
                (Tstonefile Bank2_die.s44p)
                (Ports          A23 A24 … A44    D.A23 D.A24 … D.A44   BufferA23 BufferA24 … BufferA44 ))
            ( Power
                (Tstonefile Bank2_pkg_pwr.s4p)
                (Ports          Pad.VDD Pad.VSS Buffer.VDD Buffer.VQQ))))
```

Conclusion

Does the functionality described in the presentation satisfy the needs of IC Vendors to deliver IBIS-ISS Package Models so that their customers can make engineering decisions?

- Is all of this functionality required?
- Is additional functionality required?