Package and On-Die Interconnect .3

Walter Katz

IBIS ATM

September 24, 2013

SiSoft™
We Are Signal Integrity

Overview

- Package Model Requirements
- On-Die Model Requirements
- EMD like solution with various examples of package models
- EMD like solution with various examples of on-die models
- Cases Where Not a 1:1 Correspondence Between Pins and Die Pads
- Generating the s8p for Full DQ Bus from the s4p
- A Higher Level Abstraction Based on Identifying FEXT and NEXT

Package Model Requirement Survey

Note: sNp is a short hand notation that also applies to an N Terminal IBIS-ISS subckt.

- The package models for all 4 DQ are represented by a single s2p (DQ.s2p)
- There are 4 unique s2p for the four DQ (DQ0.s2p, DQ1.s2p, DQ2.s2p, DQ3.s2p)
- There is one s8p that represents the full package model (excluding power and GND) (DQ.s8p).
- There is one s12p that represents the full package model (including power and GND) (DQ.s12p).
- The package models for all 4 DQ are represented by a single s6p (DQ.s6p). Two of the ports represent the DQ as a victim. Four of the ports represent DQ aggressors.
- There are 4 unique s6p for the four DQ (DQ0.s6p, DQ1.s6p, DQ2.s6p, DQ3.s6p) Two of the ports represent the DQ as a victim. Four of the ports represent DQ aggressors, but do not know which DQ are aggressors.
- There are 4 unique s6p for the four DQ (DQ0.s6p, DQ1.s6p, DQ2.s6p, DQ3.s6p) Two of the ports represent the DQ as a victim. Four of the ports represent DQ aggressors, and the DQ aggressors are specifically known.

Package Model Requirement Response

- From one IC Vendor
  - There is one s8p that represents the full package model (excluding power and GND) (DQ.s8p).
  - There is one s10p that represents the full package model (including power) (DQ.s10p).
  - The package models for all 4 DQ are represented by a single s6p (DQ.s6p). Two of the ports represent the DQ as a victim. Four of the ports represent DQ aggressors.
- From a second IC Vendor
  - There are 4 unique s8p for the four DQ (DQ0.s10p, DQ1.s10p, DQ2.s10p, DQ3.s10p) Two of the ports represent the DQ as a victim. Four of the ports represent DQ aggressors, but do not know which DQ are aggressors. Four of the ports represent a DQS aggressor.
- From one EDA Vendor
  - Ditto on above plus the following:
  - The package models for all 4 DQ are represented by a single s2p (DQ.s2p)
  - There are 4 unique s2p for the four DQ (DQ0.s2p, DQ1.s2p, DQ2.s2p, DQ3.s2p)
  - There are 4 unique s6p for the four DQ (DQ0.s6p, DQ1.s6p, DQ2.s6p, DQ3.s6p) Two of the ports represent the DQ as a victim. Four of the ports represent DQ aggressors, but do not know which DQ are aggressors.

On-Die Interconnect Model Requirements

- Similar questions a for Package Model Requirements.
- No IC Vendor response to this inquiry, but previous discussion (both e-mail and in the package meetings) indicate a need for similar functionality.
- There will be a need to deal with more (or different number of) die pad supply pads then package pin supply pins.

EMD Like Solution for IBIS Component Packaging – IBIS File

- The IBIS File
  - [Component] sdram
  - [IBIS_ISS_Package] sdram.pkg
  - [PIN] signal_name model_name R_pin L_pin C_pin
  - A1 VDD POWER
  - A2 VSS GND
  - B7 DQ1 DQ
  - C2 DQ0 DQ
  - D3 DQ2 DQ
  - D7 DQ3 DQ

sdram.pkg 2 Terminal s2p or subckt for each Pin

```
(sdram
   (DQ0  | s2p for this specific pin (C2)
      (Tstonefile_File (Corner DQ0_typ.s2p DQ0_min.s2p DQ0_max.s2p))
      (Terminals
         (1 (Pin (Pin_Name C2)))
         (2 (Pad (Pad_Name C2)))
      )
    )
 (DQ1  | s2p for this specific pin (B7)
      (Tstonefile_File (Corner DQ1_typ.s2p DQ1_min.s2p DQ1_max.s2p))
      (Terminals (1 (Pin (Pin_Name B7)))(2 (Pad (Pad_Name B7))))
 (DQ2  | s2p for this specific pin (D3 )
      (Tstonefile_File (Corner DQ2_typ.s2p DQ2_min.s2p DQ2_max.s2p))
      (Terminals (1 (Pin (Pin_Name D3)))(2 (Pad (Pad_Name D3))))
 (DQ3  | 2 terminal subckt for this specific pin (B7)
      (IBIS_ISS_File (Value DQ3_typ.mod))
      (Subckt (Corner DQ3_typ DQ3_min DQ3_max))
      (Terminals (1 (Pin (Pin_Name D7)))(2 (Pad (Pad_Name D7))))
...
```

sdram.pkg 2 Terminal s2p or subckt for each Pin, one parameterized subckt

```
...
 (DQ0  | parameterized subckt for this specific pin (C2)
      (IBIS_ISS_File (Value DQ.mod))
      (Subckt (Value DQ))
      (Parameters
           (Framis (Corner 0 -1 1))
           (Length (Value .02)))
       (Terminals
         (1 (Pin (Pin_Name C2)))
         (2 (Pad (Pad_Name C2)))))
 (DQ1  | parameterized subckt for this specific pin (C2)
      (IBIS_ISS_File (Value DQ.mod))
      (Subckt (Value DQ))
      (Parameters
           (Framis (Corner 0 -1 1))
           (Length (Value .01)))
       (Terminals
         (1 (Pin (Pin_Name B7)))
         (2 (Pad (Pad_Name B7)))))
...
```

S2p for each Model

```
 ...
(DQ  | s2p for this specific model (DQ)
  (Tstonefile_File (Corner DQ_typ.s2p DQ_min.s2p DQ_max.s2p))
  (Terminals
     (1 (Pin (Model_Name DQ)(Connection 1)))
     (2 (Pad (Model_Name DQ)(Connection 1))))
))
...
```

s8p for full DQ bus

```
 ...
(DQall  | s8p for all DQ bus signal pins
  (Tstonefile_File (Corner DQall_typ.s8p DQall_min.s8p DQall_max.s8p))
  (Terminals
     (1 (Pin (Pin_Name C2)(Connection 1)))
     (2 (Pin (Pin_Name B7)(Connection 2)))
     (3 (Pin (Pin_Name D3)(Connection 3)))
     (4 (Pin (Pin_Name D7)(Connection 4)))
     (5 (Pad (Pad_Name C2)(Connection 1)))
     (6 (Pad (Pad_Name B7)(Connection 2)))
     (7 (Pad (Pad_Name D3)(Connection 3)))
     (8 (Pad (Pad_Name D7)(Connection 4)))
))
...
```

S10p for full DQ bus and Vdd

```
 ...
(DQvdd  | s10p for all DQ bus signal pins
  (Tstonefile_File (Corner DQvdd_typ.s10p DQvdd_min.s10p DQvdd_max.s10p))
  (Terminals
     (1 (Pin (Pin_Name C2)(Connection 1)))
     (2 (Pin (Pin_Name B7)(Connection 2)))
     (3 (Pin (Pin_Name D3)(Connection 3)))
     (4 (Pin (Pin_Name D7)(Connection 4)))
     (5 (Pad (Pad_Name C2)(Connection 1)))
     (6 (Pad (Pad_Name B7)(Connection 2)))
     (7 (Pad (Pad_Name D3)(Connection 3)))
     (8 (Pad (Pad_Name D7)(Connection 4)))
     (9 (Pin (Pin_Name A1)(Connection 5)))
     (10 (Pad (Pad_Name A1)(Connection 5)))
))
...
```

S10p for DQ0 and Vdd

```
 ...
(DQ0vdd  | s10p for DQ0 signal and vdd
  (Tstonefile_File (Corner DQvdd_typ.s10p DQvdd_min.s10p DQvdd_max.s10p))
  (Terminals
     (1 (Pin (Pin_Name C2)(Connection 1)))
     (5 (Pad (Pad_Name C2)(Connection 1)))
     (9 (Pin (Pin_Name A1)(Connection 2)))
     (10 (Pad (Pad_Name A1)(Connection 2)))
))
...
```

s4p for any DQ and one aggressor

```
 ...
(DQ_1ag | s4p for any DQ signal and one worst case aggressor
  (Tstonefile_File (Corner DQ_1ag_typ.s4p DQ_1ag_min.s4p DQ_1ag_max.s4p))
  (Terminals
     (1 (Pin (Model_Name DQ)(Victim 1)))
     (2 (Pad (Model_Name DQ)(Victim 1)))
     (3 (Pin (Model_Name DQ)(Aggressor 2)))
     (4 (Pad (Model_Name DQ)(Aggressor 2)))
))
...
(DQ0_1ag | s4p for DQ0 signal and one worst case aggressor
 (Tstonefile_File (Corner DQ0_1ag_typ.s4p DQ0_1ag_min.s4p DQ0_1ag_max.s4p))
 (Terminals
     (1 (Pin (Pin_Name C2)(Victim 1)))
     (2 (Pad (Pad_Name C2)(Victim 1)))
     (3 (Pin (Model_Name DQ)(Aggressor 2)))
     (4 (Pad (Model_Name DQ)(Aggressor 2)))
))
...
```

On Die Models

The IBIS File

[Component] sdram
[IBIS_ISS_Package] sdram.pkg
[IBIS_ISS_On-Die] sdram.ondie

[PIN]	signal_name	model_name	R_pin	L_pin	C_pin
A1	VDD	POWER			
A2	VSS	GND			
B7	DQ1	DQ			
C2	DQ0	DQ			
D3	DQ2	DQ			
D7	DQ3	DQ			

sdram.ondie

- Start with sdram.pkg as an example
- Change all occurrences of (Pad and (Pad_Name to (Buffer and (Buffer_Name
- Change all occurrences of (Pin and (Pin_Name to (Pad and (Pad _Name
- sNp and subckt names will of course be different

Cases Where There is No 1:1 Correspondence Between Pins and Die Pads

- If there are more (or less) Die Pads than Pins the terminal records in these package model examples need for each additional Die Pad a Pad_Name of the Die Pad in the record:
  - `(<terminal#> (Pad (Pad_Name <pad_name>))`
- Examples where there is no 1:1 Correspondence
  - Different number of Power Pins and Power Die Pads
  - Fly By Clock
    - 2 Pins connected to the same Die Pad
  - Stacked Memory
  - One Pin goes to two different instances of the same model name.

Enhancements Needed for IBIS Component

- Enhancements to the IBIS component section have be proposed and discussed in the IBIS Interconnect Committee.
  - A new [Die Pad] section listing for each Die Pad
    - Pad_name
    - Signal Name
    - Model Name
  - If two Pins have the same Pin_name and same or different Signal_names then some indication that they represent one pin connected to two Die Pad instances.
  - If two Pins have different Pin_names, but the same Signal_name and Model_name, then some indication that they represent two pins connected to a single instance of the same Model_name
  - A method of indicating that a Pin is actually connected to <number of stacked model_name instances>.

Generating the s8p for Full DQ Bus from the s4p Limited Model

- The last IBIS-ATM meeting was focused on generating a full bus model (Dqall, s8p) from a simple single aggressor model (DQ1ag).
- The issue here is that typically there are two strong aggressors for a victim and two or more weaker aggressor models.
- This will required further discussion.
  - Near (strong) aggressors and far (weak) aggressors.
  - Sliding model.
  - Sparse Matrix sNp.

A Higher Level Abstraction Based on Identifying FEXT and NEXT

- We will need more clarification from Scott on what he intended here.
- One application of this abstraction is to define a package model for a bus containing both Tx and Rx buffers.
- The next slides shows that this can be done, if we choose to have the IBIS Component package solution to support this abstraction.
- Example, IBIS Component contains 96 Tx and Rx differential buffers: defining model with one victim and two aggressors. Easily extended to additional aggressors.

The Package Model – Method 1

```
(Rx_Next_Fext
 (Tstonefile (Value Rx_Next_Fext.s12p)
 (Terminals
   (1(Pin(Model_Name Rx)(Polarity Non_Inv)(Aggressor 1))
   (2(Pin(Model_Name Rx)(Polarity Inv)(Aggressor 1))
   (3(Pad(Model_Name Rx)(Polarity Non_Inv)(Aggressor 1))
   (4(Pad(Model_Name Rx)(Polarity Inv)(Aggressor 1))
   (5(Pin(Model_Name Rx)(Polarity Non_Inv)(Victim 1))
   (6(Pin(Model_Name Rx)(Polarity Inv)(Victim 1))
   (7(Pad(Model_Name Rx)(Polarity Non_Inv)(Victim 1))
   (8(Pad(Model_Name Rx)(Polarity Inv)(Victim 1))
   (9(Pin(Model_Name Tx)(Polarity Non_Inv)(Aggressor 2))
   (10(Pin(Model_Name Tx)(Polarity Inv)(Aggressor 2))
   (11(Pad(Model_Name Tx)(Polarity Non_Inv)(Aggressor 2))
   (12(Pad(Model_Name Tx)(Polarity Inv)(Aggressor 2))))
```

The Package Model – Method 2

```
(Next_Fext
 (Tstonefile (Value Next_Fext.s12p)
 (Model_Names (Value Tx Rx))
 (Terminals
  (1(Pin(Direction Fext)(Polarity Non_Inv)(Aggressor 1))
  (2(Pin(Direction Fext)(Polarity Inv)(Aggressor 1))
  (3(Pad(Direction Fext)(Polarity Non_Inv)(Aggressor 1))
  (4(Pad(Direction Fext)(Polarity Inv)(Aggressor 1))
  (5(Pin(Direction Fext)(Polarity Non_Inv)(Victim 1))
  (6(Pin(Direction Fext)(Polarity Inv)(Victim 1))
  (7(Pad(Direction Fext)(Polarity Non_Inv)(Victim 1))
  (8(Pad(Direction Fext)(Polarity Inv)(Victim 1))
  (9(Pin(Direction Next)(Polarity Non_Inv)(Aggressor 2))
  (10(Pin(Direction Next)(Polarity Inv)(Aggressor 2))
  (11(Pad(Direction Next)(Polarity Non_Inv)(Aggressor 2))
  (12(Pad(Direction Next)(Polarity Inv)(Aggressor 2))))
```