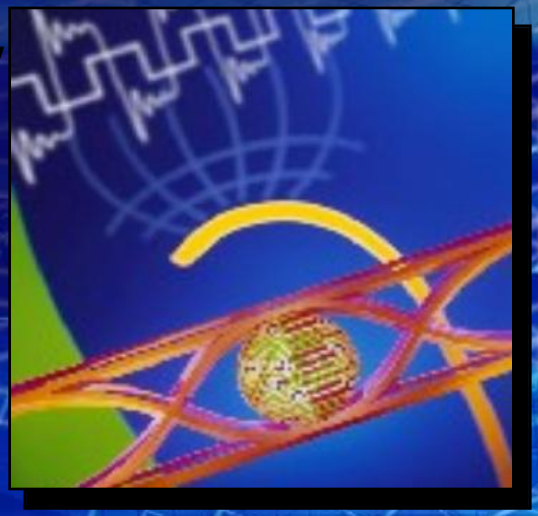
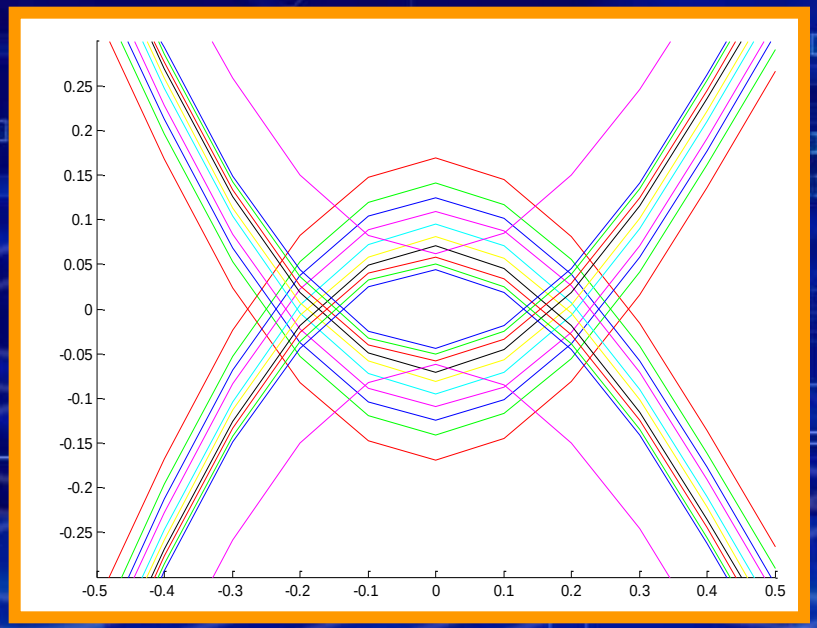
IBIS-AMI Analog Modeling and Much Needed Improvements for IBIS



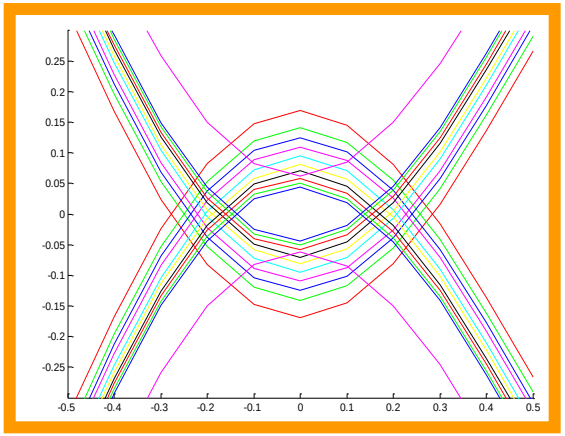
IBIS Summit, DesignCon,
February 3, 2011
Santa Clara, CA



Arpad Muranyi

Mentor Graphics®

IBIS-AMI Analog Modeling and Much Needed Improvements for IBIS



IBIS Summit, DesignCon,
February 3, 2011
Santa Clara, CA

1. My plea from DesignCon 2010
2. The golden opportunity and complete solution
3. The IBIS-AMI analog model
4. IBIS-AMI analog model examples with IBIS-ISS
5. A simple package model example with IBIS-ISS
6. Backup - An example for Figure 12 using IBIS-ISS

My plea from DesignCon 2010

This was the last slide in my DesignCon 2010 presentation
http://www.eda.org/ibis/summits/feb10/muranyi2.pdf

- **AMI gave new life to IBIS, but we still need to address some serious shortcomings in the "legacy" portions of the IBIS specification to make AMI work without any EDA tool dependent, proprietary modeling solutions**
- **This situation needs immediate attention, and urgent solutions**
- **Let's address these problems in a timely manner and provide a *complete* IBIS-AMI modeling solution in IBIS v5.2**

The golden opportunity

- **Version 1.0 of the IBIS Interconnect SPICE Subcircuit Specification (IBIS-ISS) has been submitted to the IBIS Open Forum today**
  - **IBIS-ISS standardizes the passive elements of HSPICE ® with permission from Synopsys**
  - **IBIS-ISS contains R, L, C, W-elements, S-parameter support, some controlled sources, etc…**
  - **it supports subcircuits with parameters, string parameters (for S-parameter file names, for example), etc…**
- **IBIS-ISS is a subcircuit definition language, not a complete SPICE language**
  - **IBIS-ISS subcircuits are instantiated by a top level netlist or higher level subcircuits**
  - **simulator controls (.options) or post processing statements (.print), etc… are not included**
- **IBIS already supports subcircuits under [External ***]**
  - **but Berkeley-SPICE is severely limited, useless for most modern modeling needs**
  - **adding IBIS-ISS to [External ***] would improve the buffer modeling capabilities of IBIS**
  - **adding IBIS-ISS to [Define Package Model] would improve the package modeling of IBIS**

The AMI aspect

- **Algorithmic modeling relies on accurate channel characterization simulations using IBIS models**

```
| GENERAL ASSUMPTIONS:
|
| This proposal breaks SERDES device modeling into two parts - electrical
| and algorithmic.  The combination of the transmitter's analog back-end, the
| serial channel and the receiver's analog front-end are assumed to be linear
| and time invariant.  There is no limitation that the equalization has to be
| linear and time invariant.  The "analog" portion of the channel is
| characterized by means of an impulse response leveraging the pre-existing
| IBIS standard for device models.
```

- **The Opal document (and BIRD 122) proposes a solution for the analog buffer modeling problems, but strictly for AMI use**
  - **normal IBIS buffer and package modeling are not addressed by these proposals**
- **IBIS-ISS can describe everything that is being proposed in Opal and BIRD 122**

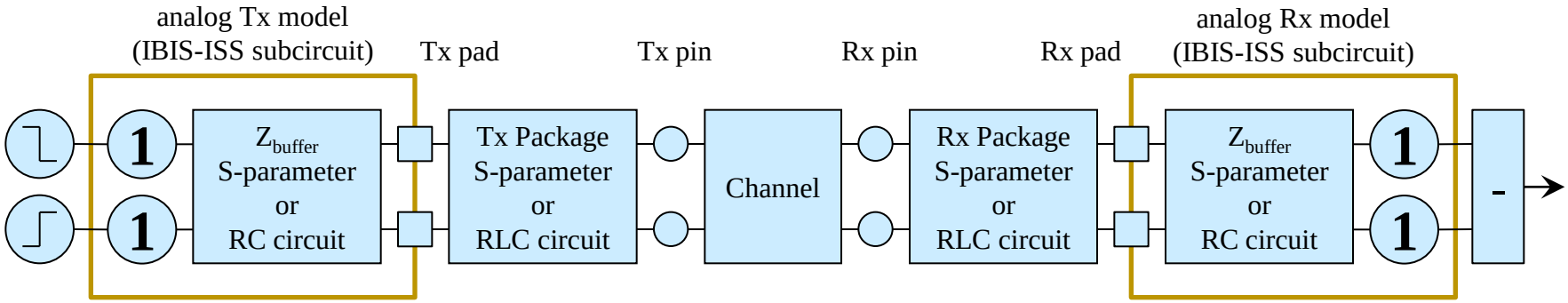
The complete solution

- **Linking IBIS-ISS with IBIS could provide significant improvements for legacy IBIS modeling as well as for IBIS-AMI analog buffer modeling**
- **BIRD 116 proposes the addition of IBIS-ISS to the existing IBIS [External Model] and [External Circuit] keywords**
  - **[External Model] is more suitable for AMI purposes because**
    - a direct link to AMI parameters can be established through the [Algorithmic Model] keyword
    - it supports true differential buffer modeling
    - works together with [Model Selector]
- **BIRD 117 and 118 propose additional flexibilities for parameterizing the [External ***] keywords**
  - **this is useful for passing AMI parameters from the .ami file to the IBIS-ISS subcircuit**
- **BIRD 125 proposes the addition of IBIS-ISS to the [Define Package Model] keyword**

A short IBIS-AMI refresher

- **IBIS-AMI simulations consist of three main pieces**
  1) **An analog model of the channel**
     - **interconnect (PCB traces, connectors, cables, etc…)**
       - these come in the form of S-parameter files, W-elements, RLC circuits
     - **analog models of the Tx and Rx devices and their package models**
       - these would theoretically come in .ibs and .pkg files
     - **the result of the analog simulation is used to generate the impulse response of the channel which is the basic input to the AMI models**
  2) **The algorithmic Tx and Rx models**
     - **these are the executable binary (.dll) files**
  3) **The algorithmic model parameter file**
     - **these are the .ami files**

The AMI analog model



PWL voltage sources in EDA tool (parameters in .ami file)

infinite input impedance must be guaranteed by an ideal isolation amplifier in the netlist or an ideal isolation amplifier incorporated within the Touchstone S-parameter data (model parameters may be defined in the .ami file)

zero output impedance must be guaranteed by an ideal isolation amplifier in the netlist or an ideal isolation amplifier incorporated within the Touchstone S-parameter data (model parameters may be defined in the .ami file)

difference waveform and IR are generated by the EDA tool for AMI simulations

Some technical details

- **The PWL stimulus voltage sources are NOT part of the analog model, these will be provided by the EDA tool**
  - their parameters (Voh, Vol, tr, tf) are defined in the .ami file using predefined Reserved_Parameters (Usage Info)
- **The analog Tx models must have an infinite input impedance, and the analog Rx models must have a zero output impedance at the algorithmic/analog boundary**
  - this may be achieved by using an ideal isolation amplifier voltage controlled voltage source (E-element) in the circuit topology, or by incorporating an isolation amplifier within a Touchstone S-parameter file that models this boundary
  - model parameters may be defined in the .ami file (as Reserved or Model Specific parameters)
- **The difference waveform is generated by the EDA tool from the differential output of the Rx model**
- **This difference waveform is further processed by the EDA tool to generate the channel's impulse response for the algorithmic model**

True differential models in IBIS

IMPORTANT: All true differential models under [External Model] assume single-ended digital port connections (D_drive, D_enable, D_receive).

The [Diff Pin] keyword is still required within the same [Component] definition when [External Model] describes a true differential buffer. The [Model] names or [Model Selector] names referenced by the pair of pins listed in an entry of the [Diff Pin] MUST be the same.

The D_to_A or A_to_D adapters used for SPICE, Verilog-A(MS) or VHDL-A(MS) files may be set up to control or respond to true differential ports. An example is shown below.

A differential stimulus may be generated by placing two D_to_A adapters between D_drive and the [External Model]

The A_to_D converter may be connected to the output of the Rx buffer if it is a 4-port model

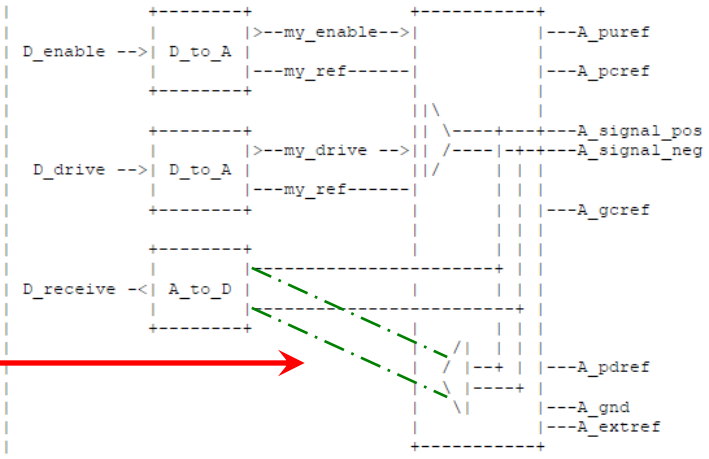


Figure 11: Example SPICE, Verilog-A(MS) or VHDL-A(MS) implementation of a true differential buffer

In EDA tool **In [External Model]**

S-parameter Tx model with [External Model] & IBIS-ISS

Legacy IBIS file:

```
|------------------------------------------------------------
| Example of an analog AMI Tx model using [External Model] and
| an IBIS-ISS S-parameter:
|------------------------------------------------------------
[Model] ISS_Diff_Tx
Model_type Output_diff
Rref_diff = 100
|
[Voltage Range] 1.0   NA   NA
|
[Ramp]
dV/dt_r   0.6/40p   NA   NA
dV/dt_f   0.6/40p   NA   NA
|
[External Model]
Language ISS
|
| Corner corner_name file_name circuit_name (.subckt name)
Corner     Typ          AMIdriver.cir    AMI_Sdrv
|
| List of parameters
Parameters  TSFile = AMIfile(Tstonefile)
|
| List of converter parameters
Converter_Parameters  VloP =  AMIfile(Vol)
Converter_Parameters  VhiP =  AMIfile(Voh)
Converter_Parameters  VloN =  AMIfile(Voh)
Converter_Parameters  VhiN =  AMIfile(Vol)
Converter_Parameters  Tfa  =  AMIfile(Trf)
Converter_Parameters  Tri  =  AMIfile(Trf)
|
| Ports List of port names (in same order as in SPICE)
Ports  A_signal_pos  A_signal_neg  my_driveP  my_driveN  A_gnd
|
| D_to_A d_port port1 port2 vlow vhigh trise tfall corner_name
D_to_A  D_drive  my_driveP  my_ref  VloP  VhiP  Tfa  Tri  Typ
D_to_A  D_drive  my_driveN  my_ref  VloN  VhiN  Tfa  Tri  Typ
|
[End External Model]
|
[Algorithmic Model]
Executable  Windows_VisualStudio_32  MentorTx.dll  MentorTx.ami
[End Algorithmic Model]
```

EDA tool GUI lets user select "one of many". Dependency Table may also affect what this GUI does.

Tx.ami file:

```
(Tstonefile (Usage Info)(Type String)
   (Corner "NC.s4p" "WC.s4p" "BC.s4p")
   (Description "Driver on-die S-parameter file")
)
(Voh (Usage Info)(Value 0.9)(Type Float)
   (Description "Output open circuit high voltage")
)
(Vol (Usage Info)(Value 0.0)(Type Float)
   (Description "Output open circuit low voltage")
)
(Trf (Usage Info)(Value 40e-12)(Type Float)
   (Description "20%-80% output rise time")
)
```

IBIS-ISS subcircuit:

```
*******************************************************************************
.SUBCKT  AMI_Sdrv  A_signal_pos  A_signal_neg  my_driveP  my_driveN  my_ref
+ TSFile="TouchstoneFileName.s4p"

Sdriver  my_driveP  A_signal_pos  my_driveN  A_signal_neg  my_ref
+ MNAME=TSFile
+ [FBASE = base_frequency] [FMAX=maximum_frequency]

*******************************************************************************
.ends
```

RC Rx model with [External Model] & IBIS-ISS

Legacy IBIS file:

```
|-------------------------------------------------------------
| Example of an analog AMI Rx model using [External Model] and
| an IBIS-ISS circuit:
|-------------------------------------------------------------
[Model] ISS_Diff_Rx
Model_type Input_diff
|
[Voltage Range] 1.0   NA   NA
|
[External Model]
Language ISS
|
| Corner corner_name file_name circuit_name (.subckt name)
Corner     Typ          AMIreceiver.cir    AMI_RCrcv
|
| List of parameters
Parameters   Rt_H  Rt_L    = AMIfile(Rt)
Parameters   Rd            = AMIfile(Rd)
Parameters   Cc_H  Cc_L    = AMIfile(Cc)
Parameters   Cd  Vt
|
| List of converter parameters
Converter_Parameters  Vlo = -0.05
Converter_Parameters  Vhi =  0.05
|
| Ports List of port names (in same order as in SPICE)
Ports  A_signal_pos  A_signal_neg  my_rcv_H  my_rcv_L  A_gnd
|
| D_to_A d_port port1 port2 vlow vhigh corner_name
A_to_D  D_receive  my_rcv_H      my_rcv_L      Vlo  Vhi  Typ
|A_to_D  D_receive  A_signal_pos  A_signal_neg  Vlo  Vhi  Typ
|
[End External Model]
|
[Algorithmic Model]
Executable  Windows_VisualStudio_32  MentorTx.dll  MentorRx.ami
[End Algorithmic Model]
|
```

Rx.ami file:

```
(Rt (Usage Info)(Value 47.75)(Type Float)
   (Description "Single-ended termination resistance")
)
(Rd (Usage Info)(Value 99.75)(Type Float)
   (Description "Differential termination resistance")
)
(Cc (Usage Info)(Value 0.5e-12)(Type Float)(Default 0.5e-12)
   (Description "Input Capacitance")
)
```

IBIS-ISS subcircuit:

```
*******************************************************************************
.SUBCKT  AMI_RCrcv  A_signal_pos  A_signal_neg  my_rcv_H  my_rcv_L  my_ref
+ Rt_H   = 1e+6
+ Rt_L   = 1e+6
+ Rd     = 1e+6
+ Cc_H   = 0
+ Cc_L   = 0
+ Cd     = 0
+ Vt     = 0

Cc_H  A_signal_pos  my_ref         C=Cc_H
Cc_L  A_signal_neg  my_ref         C=Cc_L
Cd    A_signal_pos  A_signal_neg   C=Cd
Rt_H  A_signal_pos  my_vtt         R=Rt_H
Rt_L  A_signal_neg  my_vtt         R=Rt_L
Rd    A_signal_pos  A_signal_neg   R=Rd

Vvtt    my_vtt    my_ref  DC=Vt
E_H     my_rcv_H  my_ref  VCVS  A_signal_pos  my_ref  1
E_L     my_rcv_L  my_ref  VCVS  A_signal_neg  my_ref  1
*******************************************************************************
.ends
```

Package modeling with IBIS-ISS (a simple example)

Legacy IBIS file:

```
|-------------------------------------------------------------
| Example of an IBIS model using an IBIS-ISS package model
|-------------------------------------------------------------
|
[Pin] signal_name  model_name
  1   …            …
  2   …            …
  3   …            …
  4   …            …
  5   Channel_1P   ISS_Diff_Tx
  6   Channel_1N   ISS_Diff_Tx
  7   Channel_2P   ISS_Diff_Tx
  8   Channel_2N   ISS_Diff_Tx
  9   …            …
 10   …            …
|
[Package Model]  A_4_pin_pkg_model
|…
|…
```

Matched by name

IBIS-ISS subcircuit:

```
*************************************************************
.SUBCKT  S_pkg  P1  P2  P3  P4  P5  P6  P7  P8
+ TSFile="TouchstoneFileName.s8p"

Sdriver  P1  P2  P3  P4  P5  P6  P7  P8
+ MNAME=TSFile
+ [FBASE = base_frequency] [FMAX=maximum_frequency]

*************************************************************
.ends
```

Matched by position

IBIS .pkg file:

```
|-------------------------------------------------------------
| This example implements a package model using an IBIS-ISS
| subcircuit.
|-------------------------------------------------------------
|
[Define Package Model]  A_4_pin_pkg_model
[Manufacturer]          Noname Company, Inc.
[OEM]                   Another Noname Package Company, Inc.
[Description]           4-pin …
[Number Of Pins]  4
|
[Pin Numbers]
  5   DiePortName = IDP_5
  6   DiePortName = IDP_6
  7   DiePortName = IDP_7
  8   DiePortName = IDP_8
|
[Package Circuit]
Language IBIS-ISS
|
| Corner corner_name file_name circuit_name (.subckt name)
Corner  Typ  PackageModel.spi  S_pkg
|
| Parameters List of parameters
Parameters  TSFile = "My_TstoneFile.s8p"
|
| Ports are in same order as defined in SPICE
Ports  5      6      7      8
Ports  IDP_8  IDP_7  IDP_6  IDP_5
|
[End Package Circuit]
[End Package Model]
```

Declaration of Implicit die ports

Matched by name

Backup

Figure 12 (pg. 136) in IBIS v5.0

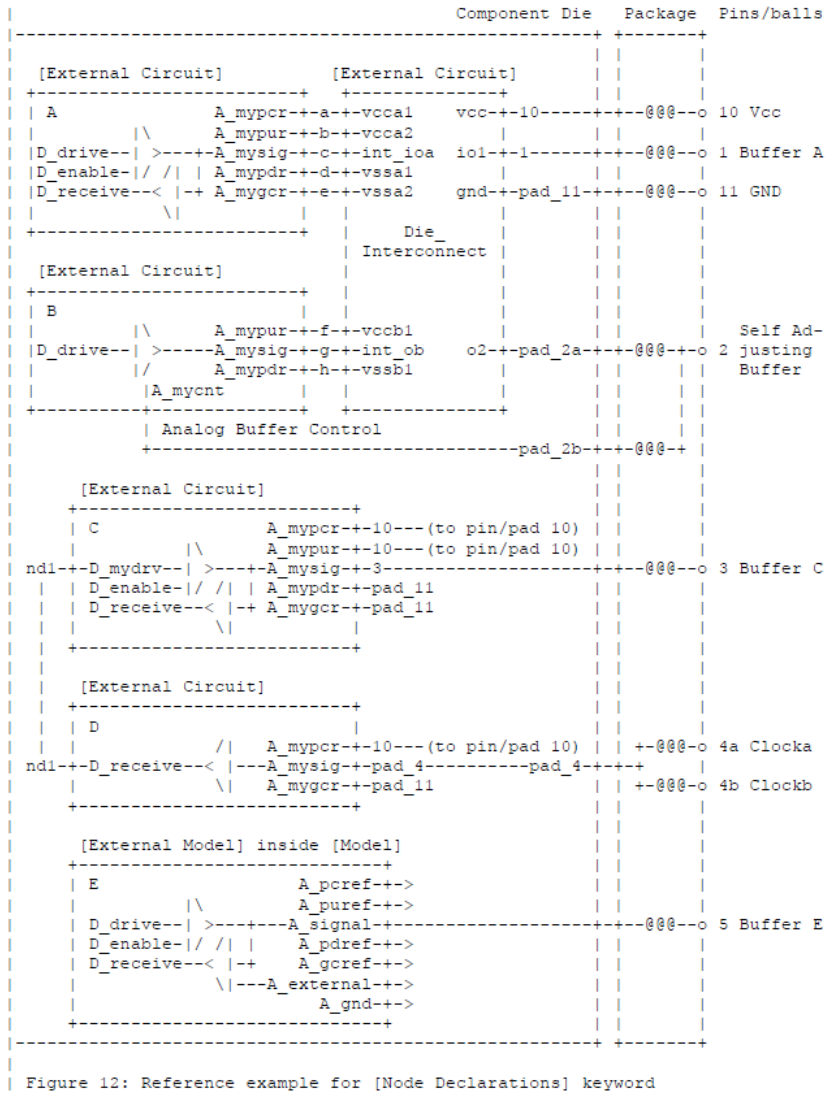


Figure 12 implemented with IBIS-ISS

Legacy IBIS file:

```
|-------------------------------------------------------------
| Example of an IBIS model using an IBIS-ISS package model.
| This example implements a package call for the drawing in
| Fig. 12 on pg. 136 of the IBIS v5.0 specification.
|-------------------------------------------------------------
|
[Pin] signal_name  model_name
 10   Vcc          POWER
  1   A0           CIRCUITCALL
 11   GND          GND
  2   CAS0         CIRCUITCALL
  3   A1           CIRCUITCALL
  4a  Clk_A        CIRCUITCALL
  4b  Clk_B        CIRCUITCALL
  5   A2           Buffer_E
|
[Node Declarations]
| Die nodes:
a  b  c  d  e  f  g  h  nd1    | List of die nodes
|
| Die pads:
pad_2a  pad_2b  pad_4  pad_11  | List of die pads
|
[End Node Declarations]
|
[Package Model]  QS-SMT-cer-8-pin-pkgs
|…
```

IBIS-ISS subcircuit:

```
**************************************************************
.SUBCKT  S_pkg  P1  P2   P3   P4   P5   P6   P7   P8
+               P9  P10  P11  P12  P13  P14  P15  P16
+ TSFile="TouchstoneFileName.s16p"

Sdriver  P1  P2   P3   P4   P5   P6   P7   P8
+        P9  P10  P11  P12  P13  P14  P15  P16
+ MNAME=TSFile
+ [FBASE = base_frequency] [FMAX=maximum_frequency]

**************************************************************
.ends
```

IBIS .pkg file:

```
|-------------------------------------------------------------
| This example implements a package model using an IBIS-ISS
| subcircuit for the drawing in Fig. 12 on pg. 136 of the
| IBIS v5.0 specification.
|-------------------------------------------------------------
|
[Define Package Model]  QS-SMT-cer-8-pin-pkgs
[Manufacturer]          Quality Semiconductors Ltd.
[OEM]                   Acme Package Co.
[Description]           8-Pin ceramic SMT package
[Number Of Pins]  8
|
[Pin Numbers]
 10   DiePort = IDP_10
  1   DiePort = IDP_1
 11   DiePort = pad_11
  2   DiePort = pad_2a
  2   DiePort = pad_2b
  3   DiePort = IDP_3
  4a  DiePort = pad_4
  4b  DiePort = pad_4
  5   DiePort = IDP_5
|
[Package Circuit]
Language IBIS-ISS
|
| Corner corner_name file_name circuit_name (.subckt name)
Corner  Typ  PackageModel.spi  S_pkg
|
| Parameters List of parameters
Parameters  TSFile = "My_TstoneFile.s16p"
|
| Ports are in same order as defined in SPICE
Ports  10  1  11  2  3  4a  4b  5
Ports  IDP_5  pad_4  IDP_3  pad_2b
Ports  pad_2a  pad_11  IDP_1  IDP_10
|
[End Package Circuit]
[End Package Model]
```

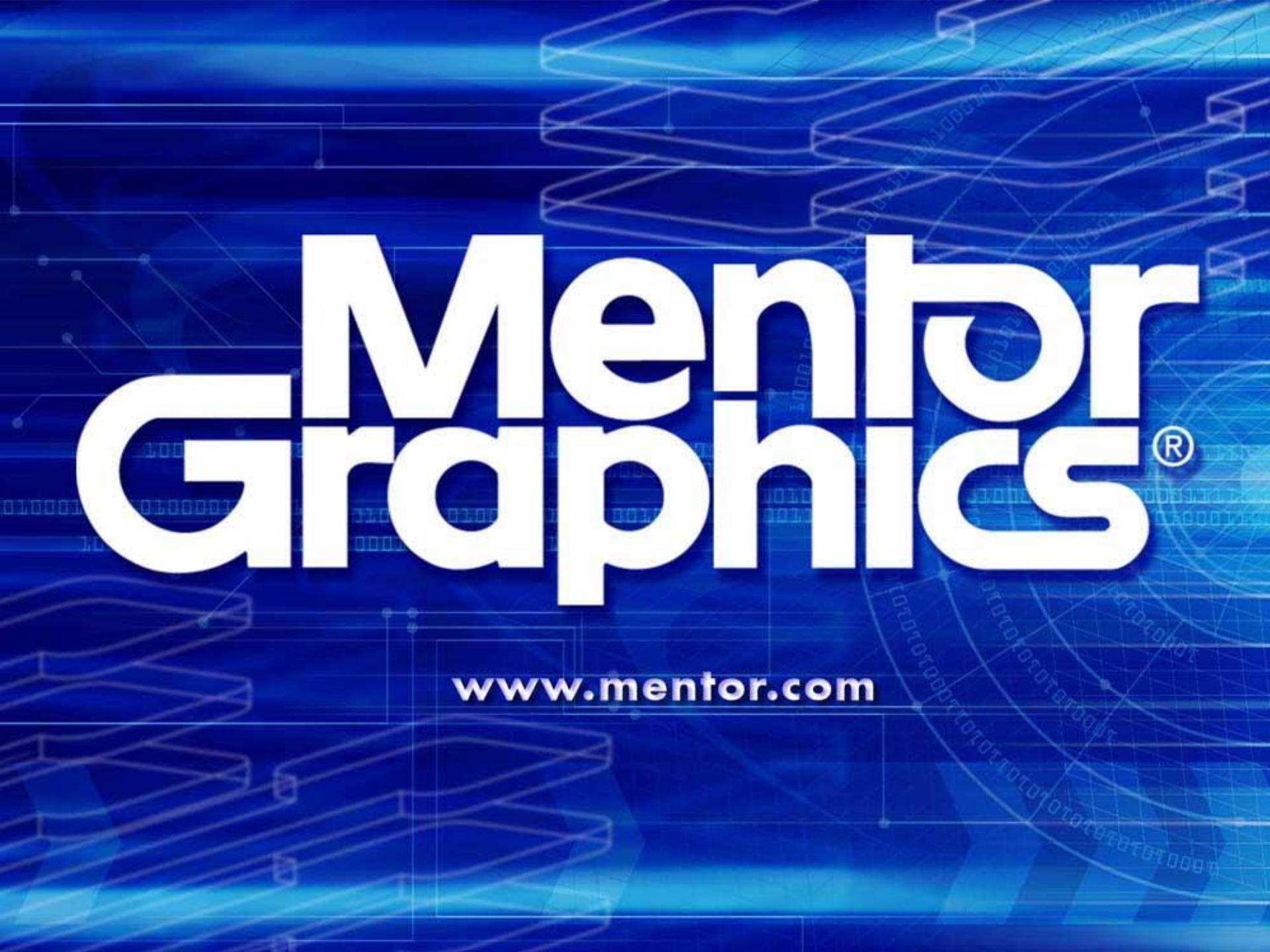
Matched by name

Implicit die ports (in blue)

Explicit die ports (in red) are declared under [Node Declarations]

Matched by position

Matched by name


Mentor Graphics®
www.mentor.com