Introduction to the IBIS Macro Model Library

Arpad Muranyi
Signal Integrity Engineering

February 9, 2006
IBIS macro modeling concept

IBIS File

[External Circuit] or [External Model] call macro model templates

Macro model templates call building blocks from standard library

Building blocks are written using the analog only features of the *-AMS languages, and can be substituted with native SPICE elements in SPICE tools if necessary

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

- Three resistors
  - R, VCR, CCR
- Three capacitors
  - C, VCC, CCC
- Four inductors
  - L, VCL, CCL, K
- 22 voltage and 22 current sources, including
  - Delay, Min, Max, Abs, Sum, Mult, Div, PWL
- An ideal T-line
- Eight IBIS buffers
  - Input, Output, IO, 3-state, Opensource, IO_opensource, Opensink, IO_opensink
Philosophy of the test suite

- Assumption:
  - HSPICE* with Verilog-A option installed, and/or
  - SMASH* installed (for the VHDL-A(MS) examples)

- All examples are ready to go
  - No editing required, just simulate and look at the waveforms
  - The examples are simple, just enough to show the concept
  - No attempt was made to show all possibilities
  - The data files contain very short data tables so the waveforms may not all be smooth and rounded
  - Some lines are commented out in the VHDL-A(MS) building blocks to allow them work in the evaluation version of SMASH (Seduction)

- A PERL script has been developed by Paul Fernando (NCSU) to extract and reformat the data from IBIS files so that the library building blocks can read it

*Other brands and names are the property of their respective owners
Parameter data file format

```
`define IO_data \
C_comp(5.0p), \ 
.kC_comp_pc(0.25), \ 
.kC_comp_pu(0.25), \ 
.kC_comp_pd(0.25), \ 
.kC_comp_gc(0.25), \ 
.IVpc_length(4), \ 
.Ipc_data((-0.08, 0.00, 0.00, 1.00)), \ 
.Vpc_data((-5.00, -1.00, 5.00, 10.00)), \ 
.IVpu_length(4), \ 
.Ipu_data((-0.10, 0.00, -0.10, -0.20)), \ 
.Vpu_data((-5.00, 0.00, 5.00, 10.00)), \ 
.IVpd_length(4), \ 
.Ipd_data((-0.10, 0.00, 0.10, 0.20)), \ 
.Vpd_data((-5.00, 0.00, 5.00, 10.00)), \ 
.IVgc_length(4), \ 
.Igc_data((-0.08, 0.00, 0.00, 0.00)), \ 
.Vgc_data((-5.00, -1.00, 5.00, 10.00)), \ 

... 
```

VHDL-A(MS)  Verilog-A(MS)

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

- **Top level SPICE files**
  - Include the simulation control statements
  - Contain simple stimulus source(s), load(s) and a call to the macro model netlist (i.e. template)
  - Equivalent of the SI simulator tool’s IBIS environment with an IBIS file using an [External Model] or [External Circuit] statement

- **“Macro model template”-s**
  - Contain Verilog-A(MS) or VHDL-A(MS) netlists to instantiate the building blocks from the library
  - Show how to pass parameter values into the instances
  - Equivalent of a “complicated buffer” macro model

- **Model library file**
  - Contains the various building blocks of “primitives”
File system of the Verilog-A(MS) distribution for HSPICE

- Installation directory of your choice
  - One .sp and .va file per library building block
  - The .sp file contains the simulation control and stimulus statements and the call to the .va netlist (or the “template” file)

- Macro_lib subdirectory
  - Contains two mandatory files and the library file
    - constants.vams
    - disciplines.vams
    - IBIS_macro_library.va
  - plus a small collection of parameter data files (*.dat)
    - these files contain IBIS buffer parameters, such as C_comp, I-V and V-t tables, or
    - appropriate data tables for the PWL sources
File system of the Verilog-A(MS) distribution for HSPICE
File system of the Verilog-A(MS) distribution for HSPICE
File system of the VHDL-A(MS) distribution for SMASH

- Installation directory of your choice
  - One .nsx and .pat file per library building block
  - The .pat (pattern) file contains the simulation control statements
  - The .nsx file contains both the top level SPICE and the VHDL-AMS netlist of the “template”

- Macro_lib subdirectory
  - Contains the library and a function file
    - IBIS_macro_library.vhd
    - MacroLib_functions.vhd
  - plus a collection of parameter data files (*.txt)
    - these files contain IBIS buffer parameters, such as C_comp, I-V and V-t tables, or
    - appropriate data tables for the PWL sources

*Other brands and names are the property of their respective owners
File system of the VHDL-A(MS) distribution for SMASH
File system of the VHDL-A(AMS) distribution for SMASH
The “Compiled” subdirectory

- The Macro_lib directory of the VHDL-A(MS) distribution includes a directory called “Compiled”
  - This contains a compiled version of the library and all of its functions as a convenience to speed up testing
  - You don’t have to use it, but it can save you time
  - The “Copy_lib.bat” file copy the compilation into the work area of each library building block example
  - (Copying is faster than compiling it 63 times)
  - The “Exclude_files.txt” file will prevent the duplication of files which do not need to be copied

- This is a SMASH specific step, other tools may deal with the WORK and user library hierarchy in a different manner
File system of the VHDL-A(EMS) distribution for SMASH

*Other brands and names are the property of their respective owners
Resistor example - simulation code

```vhdl
library IEEE, MacroLib;
use IEEE.ELECTRICAL_SYSTEMS.all;

entity IBIS_R_VHDL is
    port (terminal Node_p, Node_n : electrical);
end entity IBIS_R_VHDL;

architecture Simple_test of IBIS_R_VHDL is
begin

IBIS_R1: entity MacroLib.IBIS_R(Ideal)
    generic map ( Rval => 50.0,
                  Scale => 2.0
    )
    port map ( P => Node_p,
                N => Node_n );
end architecture Simple_test;
```

- **macro model netlist**
- **library call parameters**
- **nodes**
- **control statements**
Resistor example - library code

```vhdl
library IEEE;
use IEEE.electrical_systems.all;

--==================================
entity IBIS_R is
    generic (Rval : real := 1.0;
              Scale : real := 1.0);
    port (terminal p, n : electrical);
end entity IBIS_R;

--==================================
architecture ideal of IBIS_R is
    quantity Vout across Iout through p to n;
begin
    Vout <= Scale * Rval * Iout;
end architecture ideal;
```

```verilog
//==================================
module IBIS_R (p, n);
    electrical p, n;
    branch (p, n) Out;
    parameter real Rval = 1.0;
    parameter real Scale = 1.0;
    analog begin
        V(Out) <+ Scale * Rval * I(Out);
    end
endmodule
```

VHDL-A(AMS)  Verilog-A(AMS)
Resistor example - waveforms
VCC example - simulation code

```vhdl
library IEEE, MacroLib;
use IEEE.ELECTRICAL_SYSTEMS.all;

entity IBIS_VCC_VHDL is
  port (terminal Node_p, Node_n,
        Ctrl_p, Ctrl_n : electrical);
end entity IBIS_VCC_VHDL;

architecture Simple_test of IBIS_VCC_VHDL is
begin
  IBIS_Cl : entity MacroLib.IBIS_VCC(IBIS)
    generic map ( VD => 0.5,
                  Scale => 2.0e-12 )
    port map ( P => Node_p,
               N => Node_n,
               PS => Ctrl_p,
               NS => Ctrl_n );
end architecture Simple_test;
```
VCC example - waveforms

![Graph showing waveforms](image-url)
IBIS_IO buffer example - code

macro model
netlist

library call
parameters

nodes

control statements
IBIS_IO buffer example - waveforms
Verilog-A(MS) waveforms of IBIS.IO
A differential pre/de-emphasis buffer

- a circuit netlist serves as the macro model, instantiating
  - four Verilog-A or VHDL-AMS IBIS I/O buffer models,
  - an inverter,
  - two ideal delays, and
  - eight current sources to scale the Boost buffer's current

Diagram borrowed from M. Mirmak
Test Verilog-A "Macro Model Template" in HSPICE
******************************************************************************
.TRAN 5.0ps 150.0ns
.OPTIONS POST=1 POST_VERSION=9007 PROBE
.hd1 ".\PreDeMacro.va"
******************************************************************************
.PROBE TRAN
+ Pls  = V(Pls)
+ OutP  = V(Out_p)
+ OutN  = V(Out_n)
******************************************************************************
Vvcc  Vcc  0  DC= 5.0
* This source represents a 111000111000 pattern
Vplos Pls  0  PULSE  (1.0 0.0 1.0ns 1.0ps 1.0ps 30.0ns 60.0ns)
******************************************************************************
X1  Pls  Out_p  Out_n  Vcc  Vcc  0  0  Vcc  PreDeMacro
*  In  Out_p  Out_n  PC  PU  PD  GC  En
*
Rld1  Out_p  Vtt  R= 50.0
Rld2  Out_n  Vtt  R= 50.0
******************************************************************************
.END
******************************************************************************
Pre-emphasis buffer - macro model

`include "constants.vams"
`include "disciplines.vams"
`include ".\AMS_files\IBIS_macro_library.vams"

module PreDeMacro (Ind, IOp, ion, PCreF, PueRef, PDef, GCRef, EnD);
    input    Ind, EnD;
    electrical Ind, EnD;
    inout    IOp, ion, PCreF, PueRef, PDef, GCRef;
    electrical IOp, ion, PCreF, PueRef, PDef, GCRef,

electrical InNM, InPB, InNB, Dref;
electrical PueRefPB, PDefPB, PCrefPB, GCrefPB;
electrical PueRefNB, PDefNB, PCrefNB, GCrefNB;
electrical RcvPM, RcvNM, RcvPB, RcvNB;

parameter real BitDelay = 10.0e-9;
parameter real ScaleBoost = -0.5;

`include "No_ODT_IO_data.dat"

IBIS_IO #('IO_data) PosM (PueRef, PDef, IOp, ion, Ind, EnD, RcvPM, PCref, GCref);
IBIS_IO #('IO_data) NegM (PueRef, PDef, IOp, ion, InNM, EnD, RcvNM, PCref, GCref);
IBIS_IO #('IO_data) PosB (PueRefPB, PDefPB, IOp, ion, Ind, RcvPB, PCrefPB, GCrefPB);
IBIS_IO #('IO_data) NegB (PueRefNB, PDefNB, IOp, ion, InNB, RcvNB, PCrefNB, GCrefNB);

IBIS_V  #(.Vdc(1.0))  Ddigl (Dref, PDef);
IBIS_VCVS Invl (Dref, InNM, Ind, PDef);
IBIS_VCVS_DELAY #(.TD(BitDelay)) Dlly1 (InNB, PDef, Ind, PDef);
IBIS_VCVS_DELAY #(.TD(BitDelay)) Dlly2 (InPB, PDef, InNM, PDef);

IBIS_CCCS #(.Scale(ScaleBoost)) IpCP (PCref, IOp, PCref, PCrefPB);
IBIS_CCCS #(.Scale(ScaleBoost)) IpUP (PueRef, IOp, PueRef, PueRefPB);
IBIS_CCCS #(.Scale(ScaleBoost)) IpDP (PDef, IOp, PDef, PDefPB);
IBIS_CCCS #(.Scale(ScaleBoost)) IgCP (GCref, IOp, GCref, GCrefPB);
IBIS_CCCS #(.Scale(ScaleBoost)) IpCN (PCref, IOn, PCref, PCrefNB);
IBIS_CCCS #(.Scale(ScaleBoost)) IpUN (PueRef, IOn, PueRef, PueRefNB);
IBIS_CCCS #(.Scale(ScaleBoost)) IpDN (PDef, IOn, PDef, PDefNB);
IBIS_CCCS #(.Scale(ScaleBoost)) IgCN (GCref, IOn, GCref, GCrefNB);

endmodule
Pre-emphasis buffer - waveforms
Wrap up

- Links to the two test suites including the most current versions of the macro model library
  
  http://www.eda.org/pub/ibis/macromodel_wip/Macro_Lib_VHDL_SMASH_2006_02_01.zip

- Please try it out and provide feedback, that is the only way this effort can be made useful!

- Lots of capabilities and features could still be added to the library, but we need to know what is needed, and what is practical, etc…

- In case you need help to find SMASH…


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