Functional SI Simulation using IBIS 4.1 and HDL models

John Doe 1, Acme Widgets, jd@acme.com
John Doe 2, Acme Widgets, jd@acme.com
John Doe 3, Acme Widgets, jd@acme.com
John Doe 4, Acme Widgets, jd@acme.com

ABSTRACT
The Multi-Lingual AMS extensions of IBIS 4.1 provide significant new capabilities that can be used to implement many novel enhancements to SI verification techniques and applications. This paper will describe one such application that combines HDL functional models and IBIS I/O models of a CPU chipset and DDR-SDRAM memory to perform a thorough Signal Integrity verification of a complete Chipset to Memory interface in a single simulation.

Introduction

Just at the time it is becoming critical to perform a thorough SI analysis on a dynamic memory interface design, it is becoming increasingly difficult to perform such an analysis. High signal edge-rates, advanced techniques such as dynamic termination, complex requirements such as slew-derated timing, and critical pattern-dependant crosstalk issues are presenting significant challenges to today's designer of DDR2 and DDR3 memory interfaces. This paper will demonstrate an analysis technique using the latest SI standards that has the capability to provide a thorough automated DDR2 analysis in one simulation.

This technique utilizes the new multi-lingual capability of IBIS 4.1. It combines the functional HDL test code and IBIS I/O models, as provided by the manufacturers, to build IBIS 4.1 models that can be used in a complete functional SI analysis. Once the models are assembled, importing and assigning models is straight forward, and running the simulation involves very few steps.

The functional HDL code includes timing checks and the IBIS models contain basic electrical checks. Using the AMS extensions to augment the HDL and IBIS features, a functional SI simulation can automatically deal with the following:

- electrical violations such as overshoot/undershoot
- timing violations such as setup and hold time
- inclusion of crosstalk effects between address, data, command and control signals
- potential pattern-dependant crosstalk between multiple victim and aggressor traces
- slew-dependant timing considerations
- dynamic termination considerations
- automatic eye measurements

Current DRAM SI Verification Technique

Currently, SI and Timing of DDR memory interfaces are verified with a time-consuming, mostly manual procedure. Individual simulations are run on each net, and waveforms are analyzed using a mixture of hand measurements, tool-generated measurements, and visual inspection. Eye diagrams are typically the most dynamic technique for evaluating signal integrity and timing characteristics of an individual net. Measurements on the eye-diagrams reveal eye aperture widths, slew rates, and overshoot/undershoot characteristics. These parameters indicate waveform quality and can be used in a margin analysis to determine whether the system is timed appropriately.

All conditions under which the timing of the memory is determined are absolute worst-case conditions. If the memory still passes with timing margin (however small), it may be reasonably assumed that the module will perform to specifications in the real world. However, this method does not give an accurate representation of how the memory interface is actually performing. This worst-case prediction does not verify every net and is subject to simulation setup errors. Owing to the worst-case prediction of all parameters applied to a single net, the margin analysis method described above may predict a failure, when in fact the actual design is sound. Currently, there are very few tools that can analyze the timing for each individual net. This is a severe limitation as clock frequencies increase. Using a methodology that estimates an overall worst-case scenario can lead to a functional design being rejected due to the limitations of the tools available.

IBIS 4.1 and AMS hold great promise to significantly automate this process and to provide more direct and accurate indication of the performance of the memory interface.

IBIS 4.1 Enabling Technology

Through the late 1990s and early 2000s, it became increasingly clear to the IBIS committee that continuing to add features to the traditional IBIS template was not the best way to serve the SI community. It was becoming more difficult to add features for the latest technology, and the cycle of committee-ratification EDA-vendor-adoption was becoming intolerably long. For those reasons, the IBIS committee decided to make a fundamental change in the IBIS standard by adopting

support for standard AMS modeling languages. Unlike the fixed template of the traditional IBIS model, an AMS modeling language provides much greater flexibility. If a new feature is required to accommodate the latest technology, the modeler simply adds the new feature to the model. This eliminates the need to propose an enhancement to the template model, wait for the committee to ratify the changes and then wait for the EDA vendors to adopt the change. By simply coding the new feature into the AMS model, it immediately becomes a supported feature of any IBIS 4.1 compliant SI simulator.

IBIS 4.1 and AMS have opened the door to many novel applications such as the functional SI simulation described in this paper. Other applications include automated measurements such as jitter and eye opening for SERDES interfaces, fast configurable AMS alternatives to traditional transistor-level SPICE models, DFE and Clock/Data recover PLL models.

Figure 1 shows how IBIS 4.1 compares to traditional IBIS. While at first glance, the IBIS 4.1 multi-lingual wrapper may seem like an unnecessary step, it is actually a very useful complement to the AMS model. While the model provides the traditional electrical simulation information, the IBIS wrapper provides all the physical information not available to the user of the raw model. This physical information greatly simplifies integration of the device into the overall simulation. It also provides measurement criteria for automated testing and other information that makes the model easier to use..

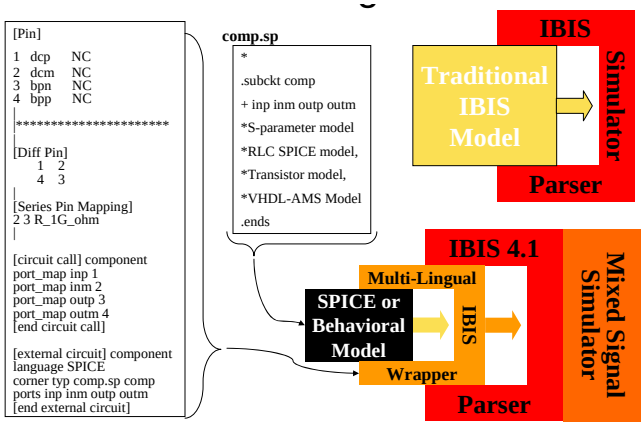

Figure 1. Traditional IBIS 3.x and IBIS 4.1

Figure 2 is an actual excerpt from one of the two IBIS files created for this functional SI simulation. The top section (down to the **EXTERNAL MODEL** divider) shows the traditional IBIS component, package parasitics and component pinout information. The next section (between the **EXTERNAL MODEL** and **TABLE MODEL** dividers) shows the new IBIS 4.1 syntax. The [Node Declarations] section declares all of the nodes used as connections internal to the IBIS model. The [Circuit Call] section connects some of these nodes and [Component] pins to external HDL models. The HDL models themselves are declared through the [External Circuit] section. The [External Circuit] keyword indicates that this is an external multi-lingual model. The "Language VHDL-AMS" statement indicates that the underlying model is in IEEE 1076.1 VHDL-AMS syntax. Since VHDL-AMS is a pure superset of VHDL, the VHDL code for the chipset and memory modules can be simulated in a VHDL-AMS simulator. The IBIS "corner" statement has been enhanced to include the names of the files where the VHDL source code resides and the names of the entity-architecture pair to use. The last section (below the **TABLE MODEL** divider) is a traditional IBIS [Model] statement that defines the I/O table model for the dq_full buffer.

```
|*****************************************************************
|   46v64m16.ibs * IBIS 4.1 Model
| 1Gb DDR SDRAM - Die Revision "A"
|
| Part Number        VDD/VDDQ            Architecture        Package
|   46V64M16TG       2.5V/2.5V            64M x 16           66-pin TSOP
|*****************************************************************
|
[IBIS Ver]     4.1
[File name]      46v64m16.ibs
[File Rev]     2.3
[Date]         09/22/2005
...

|*****************************************************************
|                      COMPONENT:   46V64M16TG
|*****************************************************************
|
[Component]         46V64M16TG
[Manufacturer]
|
[Package]   | TSOP package
|              typ              min              max
R_pkg          0.036            0.025            0.046
L_pkg          4.35nH           2.73nH           5.97nH
C_pkg          1.00pF           0.72pF           1.29pF
|
[Pin]   signal_name   model_name   R_pin     L_pin          C_pin
|
1       VDD           POWER         0.034     5.02nH         4.29pF |POWER
5       CAS#          NC            0.031     3.71nH         0.92pF |inbuff
4       RAS#          NC            0.032     3.71nH         0.95pF |inbuff
7       CS#           NC            0.033     3.43nH         0.95pF |inbuff
...
|
|**********************EXTERNAL MODEL*********************************
[node declarations]
DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ14 DQ15
UDQS LDQS UDM LDM DQE A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13
BA0 BA1 RAS_N CAS_N WE_N CS_N CKE CLK CLK_N GND VDD TEST
|
|Instantiate the VHDL Model
[circuit call] MT46V64M16
Port_map DQ0    DQ0
Port_map DQ1    DQ1
Port_map RAS_N RAS_N
Port_map CAS_N CAS_N
...
[end circuit call]
|
| Instantiation the I/O model
[circuit call]          dq_full
Port_map A_signal       1
Port_map D_receive      DQ0_in
Port_map D_Drive        DQ0_out
port_map D_Enable       DQ_OE
Port_map A_gcref        GND
Port_map A_pcref        VDD
[end circuit call]
|
...
|Declare the VHDL DRAM model
[external circuit] MT46V64M16
Language VHDL-AMS
Corner Typ mt46v64m16.vhd MT46V64M16(behave)
Ports DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ14
Ports LDQS UDQS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 BA0 BA1
Ports CLK CLK_N CKE CS_N RAS_N CAS_N WE_N UDM LDM DQE
[end external circuit]
|
...
|
|**********************TABLE MODELS***********************************
|
[Model]         dq_full
Model_type      I/O
|
Vinl = 940.000mV
Vinh = 1.560V
C_comp                        3.460pF          3.190pF          3.730pF
|
[Model Spec]
...
```

Figure 2. Excerpt of an IBIS 4.1 model used in this Functional SI example

Functional SI Analysis

Figure 3 shows a block diagram of a functional SI test system. This illustrates the connectivity of the HDL and IBIS models to the physical board layout using the IBIS 4.1 constructs.

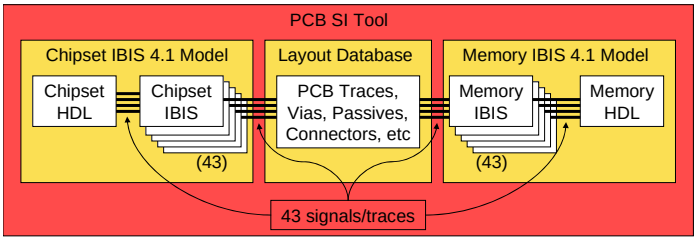

Figure 3. Block Diagram of Functional SI Simulation

As mentioned earlier, once the model is in IBIS 4.1 format, using it in an IBIS 4.1 SI simulator becomes routine. Figure 4 shows the layout of a memory module analyzed with this functional SI technique. The traces that are part of the simulation are shown highlighted in white. Once the simulation is initiated, the SI tool automatically assembles the following items into a netlist and invokes a mixed signal simulation.

- Chipset HDL Model
- Chipset IBIS Models
- Chipset Package Models
- PCB Stub and VIA Models
- PCB Trace Models
- PCB Resistor Pack Models
- PCB Capacitor Models
- PCB Connector Models
- Memory Package Models
- Memory IBIS Models
- Memory HDL Model

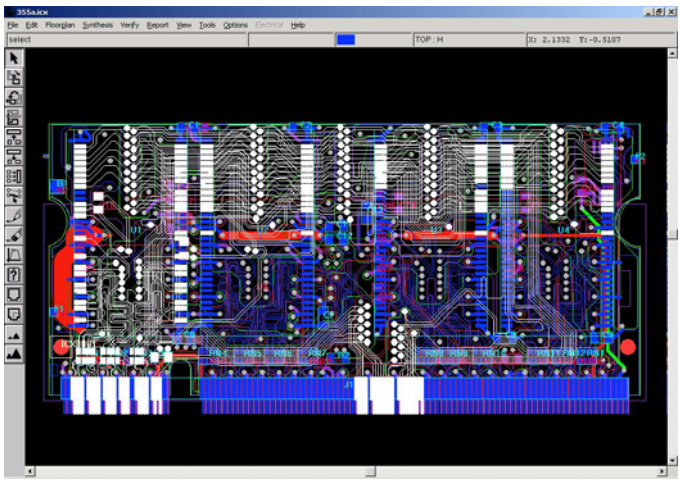

Figure 4. Section of PCB Memory Module layout with simulated traces highlighted in white

Functional SI Verification Results

Since functional SI analysis uses the digital code from the chipset testbench, the system stimulus is automatically generated. Likewise, since the memory responds to commands from the chipset as it would in a real system, the data patterns originating from the memory are completely automated as well. Also, the chipset and memory both use digital HDL that contains timing checks, so any timing violation caused by PCB effects will be flagged. Figure 5 shows an example of this (created when some traces were extended past their critical length).

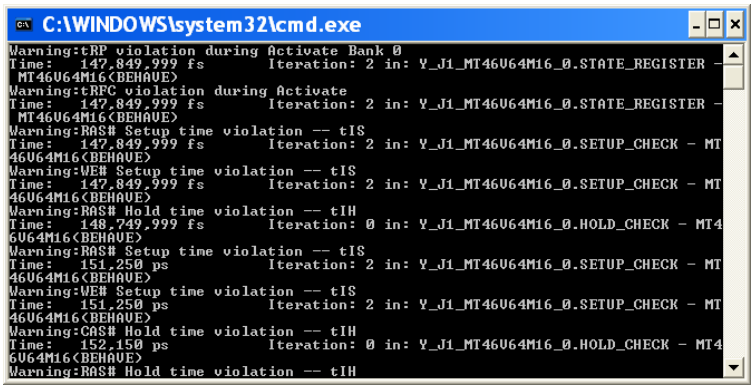

Figure 5. Output screen showing timing error messages

Figure 6 shows the analog address, clock and data signals at the memory die during a burst write from the chipset. Notice that all the traces are evaluated simultaneously, using pseudo-random address and data patterns. Alternatively, incrementing address and data, or walking ones patterns could also be easily accommodated. These results were obtained after roughly 3 minutes of simulation.

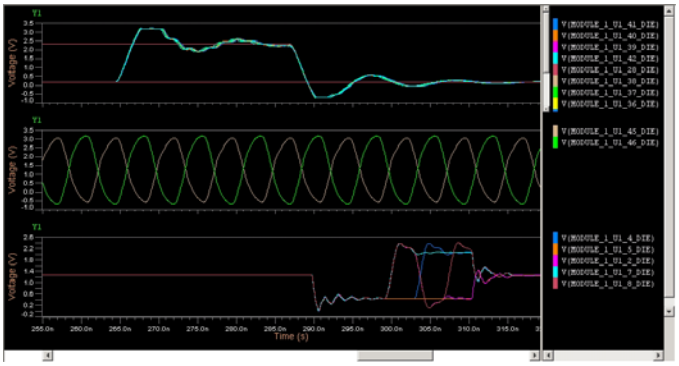

Figure 6. Analog Simulation Results of Address, Clock, and Data

Figure 7 shows the digital address and clock signals at the memory die during a burst write from the chipset.

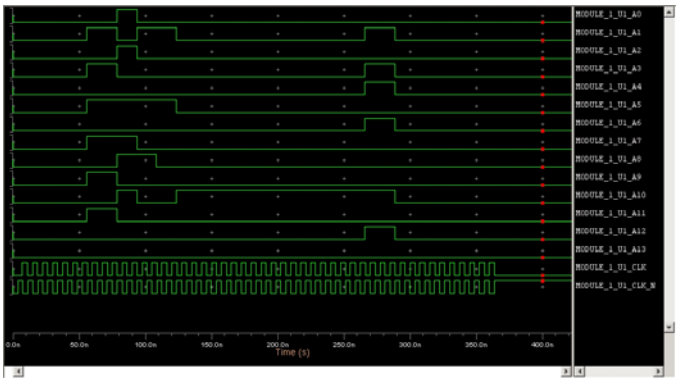

Figure 7. Digital Simulation Results of Address and Clock

Figure 8 shows the combined analog and digital address signals at the memory die during a burst write from the chipset. Combination of the analog and digital waveforms provides a useful visualization of the logical analog equivalent.

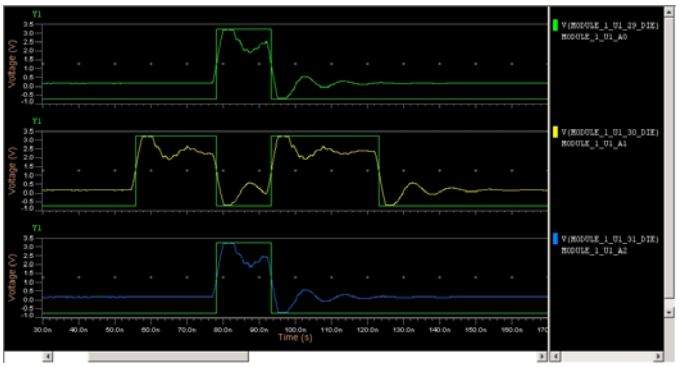

Figure 8. Analog and Digital Simulation Results of Address

Improvements and Future Development

Setting up this verification technique required instantiating a [Model] in a [Circuit Call] statement. This is specifically not allowed by the IBIS 4.1 specification (but was fortunately allowed by our test simulator). Removal of this exclusion from the IBIS specification would simplify creation of these models.

Also, the chipset and memory HDL code had to be enhanced to include address and data patterns that were more appropriate for SI analysis.

The simulation could be enhanced further by exercising all other signal lines in the system. This would exercise the entire memory bus, and all signals visible in Figure 4 would be highlighted.

Due to time limitations, dynamic ODT and slew-dependant timing were not modeled in this example. But, it should be no problem to do so with the AMS modeling language.

Crosstalk analysis could also be enhanced in this application by replacing the PCB physical information with an 86-port S-parameter model.

Summary and Conclusion

This application shows an example of using IBIS 4.1 and AMS to simplify and improve the electrical and timing analysis of complex CPU to Memory interfaces. This technique provides an industry standard method to accommodate the complexities of modern SDRAM interface design including:

- High edge and clock rates
- Dynamic On-Die Termination
- Slew dependant timing
- Traditional and pattern-dependant crosstalk
- Electrical violations such as excessive overshoot or insufficient slew rate
- Timing violations which include degradation due to PCB and crosstalk effects.
- Automated eye measurement
- Automated pass/fail determination

Furthermore, the verification is performed on a bit by bit basis yielding much more realistic results over the more pessimistic eye-diagram post-processing techniques.

This application shows just one of the many ways IBIS 4.1 and AMS can improve the usability, capability and productivity of SI analysis. Many future applications will benefit from the combination of IBIS 4.1 and AMS.