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 worstcase 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 EDAvendor-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 multilingual 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 entityarchitecture 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.

46*64ml6.ibs * IBIS 4.1 Model 1Gb DUR SDRAM - Due Revision 'A* Part Number VD/VDOV 46%64Ml6TG 2.5V/2.5V 64H × 16 [BIS Ver] 4.1 [File name] 46v64ml6.ibs [File name] 46v64ml6TG Manufacturer] 09/22/2005 ************************************	****	****		******	*****
1 40% AND STATES 640 x 16 65-pin TSO 1 1015 4:0% AND STATES 640 x 16 65-pin TSO 1 101 109/22/2005 109/22/2005 109/22/2005 1 COMPONENT: 460% AND STATES 0.046 1 COMPONENT: 460% AND STATES 0.046 1 Upp min max R_pkg 0.036 0.025 0.046 L_pkg 1.00F 0.72PF 1.29PF Pin signal_name model_name R_pin L_pin 1 VDD POVER 0.034 5.02nH 4.29PF 1 VDD POVER 0.031 3.71nH 0.92pF 1 VDD POVER 0.033 3.43nH 0.95pF 1nb 1 VDD POVER 0.034 5.02nH 4.23pF 100 1 VDD POVER 0.033 .43nH 0.95pF 1nb 1 VDD POVER 0.033 .43nH 0.95pF 1nb 1 CS# NC	46v64m16.ibs 1Gb DDR SDRAM - Part Number	* IBIS 4.1 Mode - Die Revision ' VDD/VDDO	el 'A" Arc	hitecture	Package
The set of	46V64M16TG	2.57/2.57	64	$M \ge 16$	66-pin TSOP
COMPONENT: 46V64M16TG Kanufacturer] Package] TSOP package 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 [PO 5 CAS# NC 0.031 3.71nH 0.92pF inb 7 CS# NC 0.033 3.43nH 0.95pF inb 7 CS# NC 0.031 0.000 DQ10 DQ11 DQ12 DQ13 DQ14 DQ15 1 Nstantiate the VHDL Model Corcuit call) MT46V64M16 Port_map DQ DQ0 Port_map Asignal 1 Fort.map CAS_N CAS_N Fort_map Laster GND Fort.map D_preceive DQ0_in Fort.map A_signal 1 Fort.map A_signal 1 Fort.map A_signal 1 Fort.map A_signal 1 Fort.map A_signal 1 Fort.map D_preceive DQ0_int Fort.map A_signal 1 Fort.map A_signal 1 Fort.signal A_signal A_s	IBIS Ver] 4 File name] File Rev] 2.3 Date] 092	6v64m16.ibs 22/2005			
Component] 46V64M16TG (Manufacturer) (Package] TSOP package 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 [PO 5 CAS# NC 0.033 3.71nH 0.92pF inb 4 RAS# NC 0.033 3.71nH 0.92pF inb 7 CS# NC 0.033 3.43nH 0.95pF inb 7 CS# NC 0.033 3.43nH 0.95pF inb 1 To CS# NC 0.033 3.43nH 0.95pF [ND 1 To CS# NC 0.033 3.43nH 0.95pF] 1 To CS# NC 0.031 0.11 DOL 2013 DO14 DO15 1 To CS# NC 0.032 0.11 DO12 DO13 DO14 DO15 1 To CS# NC 0.032 0.11 A12 A13 A4 5.64 A7 A8 9A 10.11 A12 A13 1 Ratatiate the VHDL Model 1 Circuit call] MT46V64M16 Port_map DO1 DO1 Port_map CAS_N CAS_N 1 To Tamp Drive DO0 1 Port_map Aggref GND 1 Port_map Aggref GND 1 Ports L00 DO1 DO2 DO3 DO4 DO5 DO6 DO7 DO8 DO9 DO10 DO11 DO12 DO13 DO1 1 Ports DO3 DO3 A4 A5 A6 A7 A8 A9 10 A11 A12 A13 BA0 BA1 1 Port SLOSUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports DO0 DO1 DO2 DO3 DO4 DO5 DO6 DO7 DO8 DO9 DO10 DO11 DO12 DO13 DO1 1 Port SLOSUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1 Port A1 A12 A13 BA0 BA1 Ports L00 SUBOS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 1	**************	**************************************	•********** VT: 46V64 •****	*************** M16TG *************	***************************************
Package] TSOP package min max pkg 0.036 0.025 0.046 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 Pinb 2 VDD POWER 0.031 3.71nH 0.32pF Pinb 4 RAS# NC 0.033 3.43nH 0.95pF Pinb 5 CAS# NC 0.033 3.43nH 0.95pF Pinb 600 D02 D03 D04 D05 D06 D07 D08 D09 D010 D011 D012 D013 D014 D015 D004 D02 D03 D04 D05 D06 D07 D08 D09 D010 D011 D012 D013 D014 D015 700 D10 D02 D03 D04 D05 D06 D07 D08 D09 D010 D011 D012 D013 D014 D015 D005 D07 map D01 D01 Port_map D01 D01 Port_map D40 D00 D00 D00 D01 D01 Port_map D45 N RAS N Port_map D5 D00_in Port_map D45 N RAS N Port_map D45 N RAS N Port_map D 5 D00_in Port_map D4 D00 IN Portes D00 D01 D01 D01 D012	Component] Manufacturer]	46V64M16TG			
Lpkg 0.036 0.025 0.046 _pkg 1.35nH 2.73nH 5.97nH _pkg 1.00pF 0.172pF 1.23pF [Pin] signal_name model_name R_pin L_pin C_pin VDD POWER 0.034 5.02nH 4.29pF Finb AGS# NC 0.032 3.71nH 0.95pF Finb ARAS# NC 0.033 3.43nH 0.95pF Finb NG D0032 D04 D05 D06 D07 D09 D01 D012 D013 D014 D015 NGO D01 D02 D03 D04 D05 D06 D07 D09 D09 D010 D011 D012 D013 D014 D015 D012 D012 D013 D014 D015 Sort_map D01 D01 D02 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 SA0 BA1 RAS_N RAS_N Sort_map DAS N RAS_N N RAS_N N RAS_N SACT_MAS Sort_map D01 D01 Sort_map CAS_N RAS_N Sort_map CAS_N RAS_N Sort_map CAS_N RAS_N Sort_map D_Braceive D00_oit Sort_map D_Braceive D00_oit Sort_map D_Braceive D00	Package] TSO	package			
Pin] signal_name model_name R_pin L_pin C_pin VDD POWER 0.034 5.02nH 4.29pF [PO CAS# NC 0.031 3.71nH 0.92pF [inb RAS# NC 0.032 3.71nH 0.95pF [inb rest NC 0.032 3.71nH 0.95pF [inb node declarations] 0.032 3.43nH 0.95pF [inb node declarations] 0.02002 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ14 DQ15 DQ14 DQ2 DQ3 DQ4 DQ5 NQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ14 DQ15 DQ5 DQ5 UD4 UDH DD4 DQ6 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 SA0 BA1 RAS_N CAS_N WE_N CS_N CKE CLK CLK_N GND VDD TEST Instantiate the VHDL Model circuit call] Id_full Ort_map D1 D01 Ortmap D1 D01 Ort_map A_signal 1 - - Ort_map A_goref OND DO - Ort_map A_goref OND OND - Ort_map A_poref VDD - - - Ort_map A_goref VDD <td< td=""><td>typ 2_pkg 0.0 2_pkg 4.0 2_pkg 1.0</td><td>o m: 036 0 95nH 2 00pF 0</td><td>n 025 73nH 72pF</td><td>max 0.046 5.97nH 1.29pF</td><td></td></td<>	typ 2_pkg 0.0 2_pkg 4.0 2_pkg 1.0	o m: 036 0 95nH 2 00pF 0	n 025 73nH 72pF	max 0.046 5.97nH 1.29pF	
VDD POWER 0.034 5.02nH 4.29pF [PO CAS# NC 0.031 3.71nH 0.92pF]nb RAS# NC 0.032 3.71nH 0.95pF]nb CS# NC 0.033 3.43nH 0.95pF]nb CS# NC 0.032 024 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ14 DQ15 DQ5 DDQ5 DDQ5 UDM LDM DQE A0 A1 A2 A3 A4 A5 A6 A7 A8 A3 A10 A11 A12 A13 A0 BA1 RAS_N CAS_N WE_N CS_N CKE CLK CLK_N GND VDD TEST Instantiate the VHDI Model circuit call] M146V64M16 Vort_map DQ0 DQ0 Vort_map RAS_N RAS_N Vort_map CAS_N CAS_N N Vort_map CAS_N CAS_N N Vort_map D_receive DQ0_in Vort_map D_receive DQ0_in Vort_map A_signal 1 Vort_map A_goref GND Vort_map A_goref GND Vort_map A_goref GND Vort_map A_goref GND Vorts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Vort_map A_goref GND Vort_map A_goref GND Vort_map A_goref GND Vorts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Vorts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Vorts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Vorts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Vorts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Vorts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Vorts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Vorts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Vorts CLK_N CKE CS_N RAS_N CAS_N WE_N UDM LDM DQE end external circuit] *********************************	Pin] signal_na	ame model_name	e R_pin	L_pin	C_pin
<pre>************************************</pre>	VDD CAS# RAS# CS#	POWER NC NC NC	0.034 0.031 0.032 0.033	5.02nH 3.71nH 3.71nH 3.43nH	4.29pF POWEH 0.92pF inbuff 0.95pF inbuff 0.95pF inbuff
<pre>node declarations] 000 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ14 DQ15 DQS LDQS UDM LDM DQE A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A0 BA1 RAS_N CAS_N WE_N CS_N CKE CLK CLK_N GND VDD TEST Instantiate the VHDI Model circuit call) MT46V64M16 fort_map DQ1 DQ1 fort_map CAS_N CAS_N end circuit call] Instantiation the I/O model circuit call] dq_full fort_map A_signal 1 fort_map A_signal 1 fort_map A_signal 1 fort_map A_sreef GND fort_map A_greef WDD end circuit call] Declare the VHDI DRAM model external circuit] MT46V64M16 anguage VHDL-AMS former Typ mt46v64m16.vdh MT46V64M16(behave) forts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 forts LDQS UDQS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 BA0 BA1 forts LDQS UDQS A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 BA0 BA1 forts CLK_LN CKE CS_N RAS_N WE_N UDM LDM DQE end external circuit] *********************************</pre>	*****	*******EXTERNAI	. MODEL****	****	*****
Instantiate the VHDL Model circuit call] MT46V64M16 Ort_map D00 D00 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 D00_in Port_map D_receive D00_out port_map D_receive D00_out port_map A_gcref GND Port_map A_gcref VDD end circuit call] Declare the VHDL DRAM model external circuit] MT46V64M16 anguage VHDL-AMS ports D00 D01 D02 D03 D04 D05 D06 D07 D08 D09 D010 D011 D012 D013 D01 Ports D00 D01 D02 D03 D04 D05 D06 D07 D08 D09 D010 D011 D012 D013 D01 Ports D00 D01 D02 D03 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 D0E end external circuit] ********************************	node declaration 000 DQ1 DQ2 DQ3 I IDQS LDQS UDM LD1 3A0 BA1 RAS_N CAS	NS] DQ4 DQ5 DQ6 DQ7 I DQE A0 A1 A2 A S_N WE_N CS_N CI	DQ8 DQ9 DQ 43 A4 A5 A6 (E CLK CLK_	010 DQ11 DQ12 A7 A8 A9 A10 N GND VDD TES	DQ13 DQ14 DQ15 A11 A12 A13 T
<pre>ind circuit call] Instantiation the I/O model circuit call] dq_full fort_map A_signal 1 fort_map D_receive DQ0_out cort_map D_Enable DQ_Oet ort_map A_pcref GND cort_map A_pcref VDD end circuit call] Declare the VHDL DRAM model external circuit] MT46V64M16 anguage VHDL-AMS corner Typ mt46v64m16.vhd MT46V64M16(behave) forts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 forts DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 forts DQ0 SUDQS AD A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 BA0 BA1 forts CIK CIK_N CKE CS_N RAS_N CAS_N WE_N UDM IDM DQE end external circuit] ********************************</pre>	Instantiate the circuit call] M Port_map DQ1 D(Port_map DQ1 D(Port_map RAS_N Ri Port_map CAS_N Ci	VHDL Model 746V64M16 20 15_N 15_N			
Instantiation the I/O model circuit call] dq_full Port_map A signal 1 Port_map D_receive DQ0_in Port_map D_Drable DQ_OE Port_map A_gcref GND Port_map A_gcref VDD end circuit call] Declare the VHDL DRAM model external circuit] MT46V64M16 anguage VHDL-AMS Orner Typ mt46v64m16.vhd MT46V64M16(behave) Ports DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Ports DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ1 Ports DQ0 UDQ A DA AA A	end circuit call	.]			
Declare the VHDL DRAM model external circuit] MT46V64M16 anguage VHDL-AMS Corner Typ mt46v64m16.vhd MT46V64M16(behave) Model] dq_full fodel_type I/O Vinh = 940.000mV Corner Typ MC40pF 3.730pF	Instantiation circuit call] ort_map A_signa ort_map D_receir ort_map D_Drive ort_map D_Enable ort_map A_gcref ort_map A_gcref	the I/O model dq_full dq_full dq_full dq_full dq_full DQ0_out bQ0_out GND VDD			
**********************************	Declare the VHDI external circui anguage VHDL-AMS orner Typ mt46vi "orts DQO DQ1 DQ2 Ports LDQS UDQS d "orts CLK CLK_N (end external cir	DRAM model] MT46V64M16 4m16.vhd MT46V6 DQ3 DQ4 DQ5 DQ 0 A1 A2 A3 A4 KE CS_N RAS_N (rouit]	54M16(behav 26 DQ7 DQ8 35 A6 A7 A8 CAS_N WE_N	70) DQ9 DQ10 DQ11 8 A9 A10 A11 A UDM LDM DQE	DQ12 DQ13 DQ14 12 A13 BAO BA1
Model] dq_full fodel_type I/O 7inl = 940.000mV 7inh = 1.560V 3.460pF [3.190pF 3.730pF _comp 3.460pF [3.190pF 3.730pF	************	*******TABLE MO)DELS*****	*****	*****
/inl = 940.000mV /inh = 1.560V comp 3.460pF β.190pF 3.730pF	Model] da fodel_type I/	∟full ′0			
Model Speci	'inl = 940.000mV 'inh = 1.560V comp	3.4	50pF	β.190pF	3.730pF

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

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).

C:\WINDOWS\system32\cmd.exe - 🗆 🗙
Warning:tRP violation during Activate Bank 0
Time: 147,849,999 fs Iteration: 2 in: Y_J1_MT46U64M16_0.STATE_REGISTER -
MT46V64M16(BEHAVE)
Warning:tRFC violation during Activate
Time: 147,849,999 fs Iteration: 2 in: Y_J1_MT46V64M16_0.STATE_REGISTER -
MT46V64M16(BEHAVE)
Warning:RAS#_Setup_time_violation tIS
Time: 147,849,999 Fs Iteration: 2 in: Y_J1_MT46064M16_0.SETUP_CHECK - MT
46V64M16(BEHAUE)
Warning:WEH Setup time violation tis
IIME: 147,849,999 fs Iteration: 2 in: Y_JI_MI46064MI6_0.SEIUP_CHECK - MI
46064716(BERHOE)
WARNING-MASH NOID THE VIOLATION TH Time: 140 740 909 Fe
Wayner BOCH Setup time uiglation tIS
Time: 151 250 ve Ite violation: 2 in: V II MT46164M16 0 SETUP CHECK - MT
Warning: WE# Setup time violation tIS
Time: 151.250 ps Iteration: 2 in: Y J1 MT46U64M16 0.SETUP CHECK - MT
46U64M16(BEHAUE)
Warning:CAS# Hold time violation tIH
Time: 152,150 ps Iteration: 0 in: Y_J1_MT46U64M16_0.HOLD_CHECK - MT4
6U64M16 <behaue></behaue>
Warning:RAS# Hold time violation tIH

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.

Functional SI Verification Results

•	+		+					MCOULE_1_U1_A0
· · · · ·	· _			•	•			MCOULE_1_U1_A1
•								HCOULE_1_U1_A2
								HCOULE_1_U1_A3
								MCOULE_1_U1_A4
								M0001E_1_01_A5
					•			M0001E_1_01_A6
								MCOULE_1_U1_A7
								MC00LE_1_01_A8
								MODULE_1_U1_A9
								H000LE_1_U1_A10
								HC00LE_1_01_A11
				•	•			H00ULE_1_U1_A12
								HCOULE_1_U1_A13
ากกกกกกกก	านขามามา	เกกสกกเ	ากกกกกก	กกกกก	กกกลกกก	Innnn		WCDOLK_1_U1_CTK
								HCDOLE_1_U1_CLK_N
	100.00	950.04	200.04	240.00	300.04	350.00	400.05	-
			TRADE IN CO.	and the second sec	the second se	and the second se		

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 slewdependant 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.