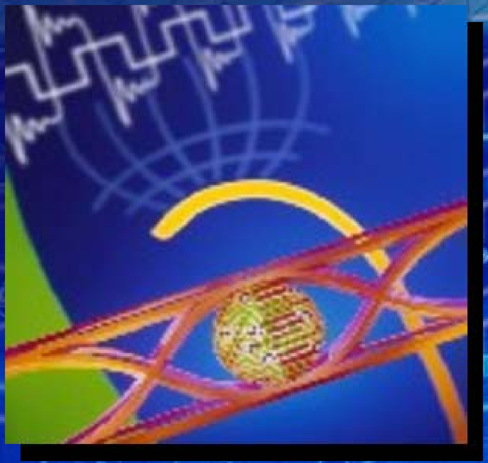
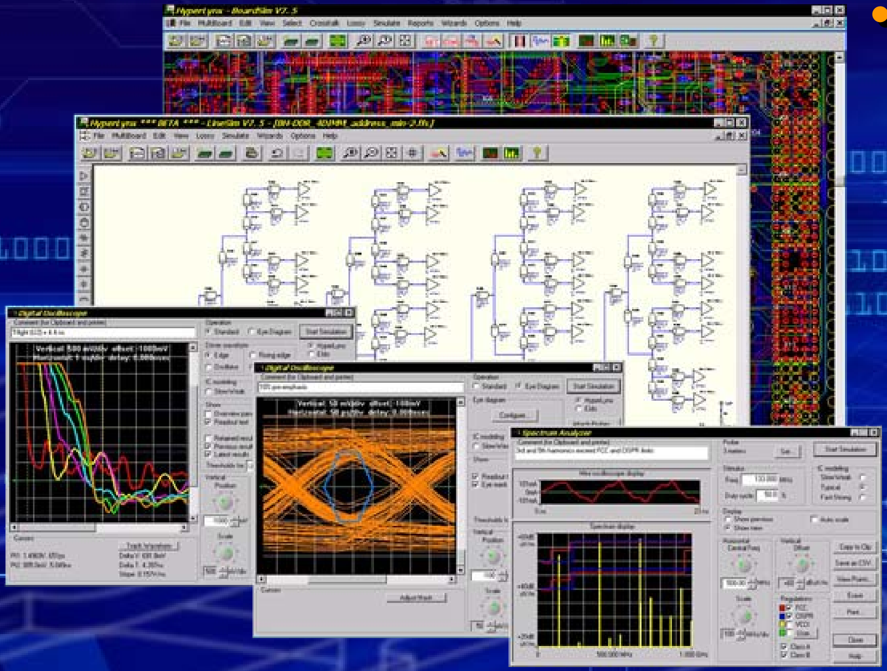
IBIS4.2 and VHDL-AMS for SERDES and DDR2 Analysis





Ian Dodd

Architect, High Speed Tools

Ian_dodd@mentor.com

Gary Pratt

Manager, High Speed Partnerships

gary_pratt@mentor.com

31st October 2006

© Mentor Graphics Corp., 2006,

Reuse by written permission only. All rights reserved.

IBIS 4.2 Multi-lingual Extensions

- Traditional IBIS lacks the ability to adequately model the behavior of devices used in state of the art communication channels:
  - Drivers with pre-compensation
  - Receivers with input slew rate sensitivity.
  - Phase locked loop clock and data recovery
  - Simple and adaptive equalization
  - Multi-level signaling

- Traditional IBIS also lacks the ability to adequately specify new measurements:
  - Differential overshoot
  - Eye masks

- IBIS 4.2 Multi-lingual Extensions can address both of these limitations

SPICE as an IBIS 4.2 Multi-lingual option

- Good supply of transistor level models for older devices
  - May be encrypted and therefore not portable between tools
- Poor standardization
  - Lots of proprietary primitives
- Extremely slow simulation
  - Particularly when using transistor level models
- Missing a high level view
  - Needed to effectively model complex digital logic
  - Needed to make complex measurements.

AMS as an IBIS 4.2 Multi-lingual option

- VHDL-AMS and Verilog-AMS International standards
  - IEEE and Accellera
- Fast.
  - Models are compiled to machine code just like built in primitives.
  - Digital content is handled in event driven kernel
- Flexible
  - Can provide both behavior and measurement.
- Accurate
  - Uses the same analog non-linear solver as SPICE

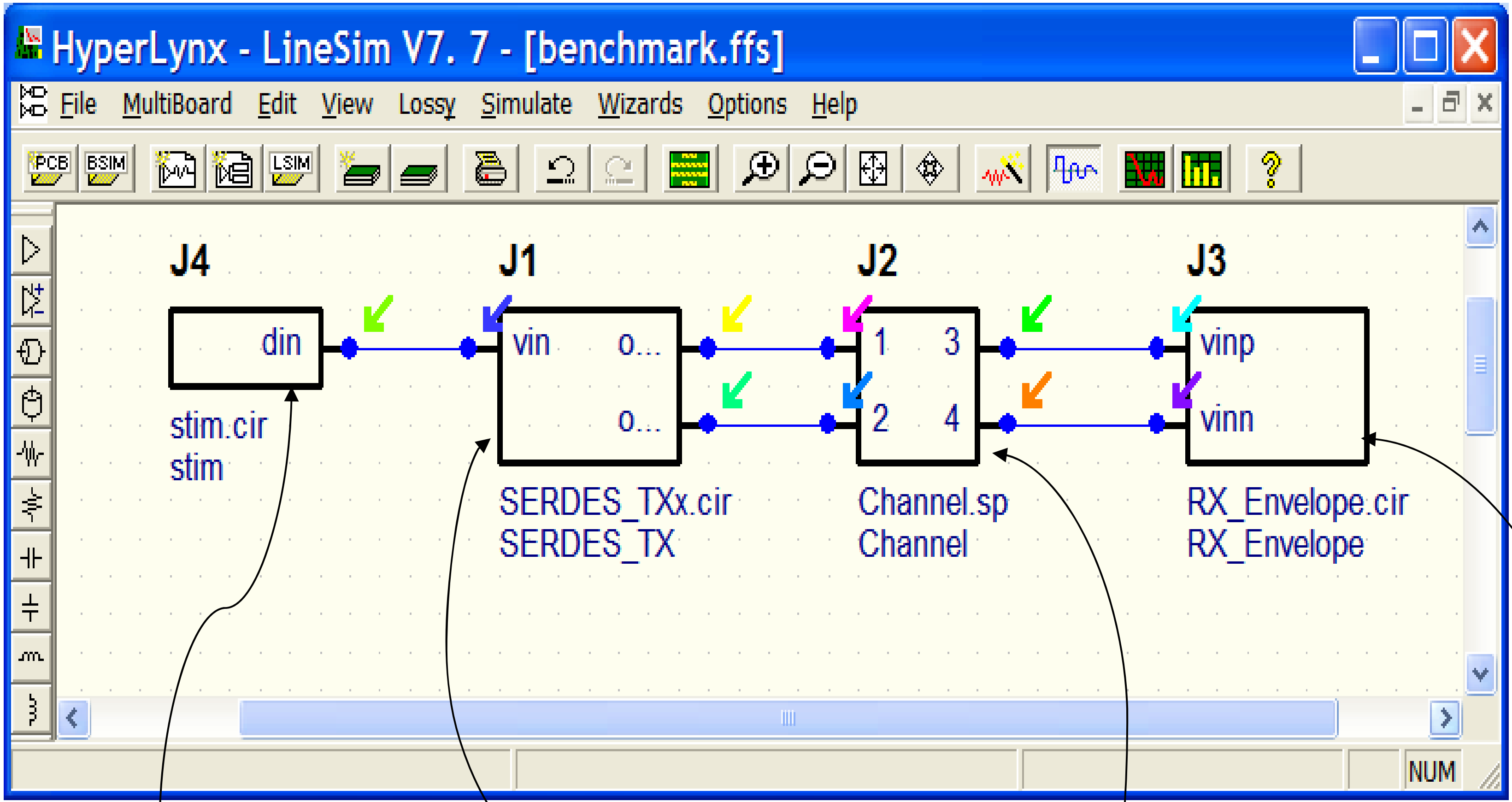
IBIS 4.2 Multi-lingual Case Studies

- The model maker and user can best decide whether it is best to create models using the IBIS 4.2 multi-lingual extensions utilizing SPICE or AMS
- The best solution for the SI Engineer may well be a tool that supports the mixing of both
- AMS provides some unique features so this presentation is going to provide two case studies that highlight these features.

AMS Case Study One
Full non-linear analysis of a SERDES channel

- Simulate to 10 million data bits
- Custom data pattern
- VHDL-AMS Driver with non-linear drive characteristics and pre-compensation
- Realistic S-Parameter model for packages, two connectors and backplane*
- VHDL-AMS receiver model with built in envelope recorder
- Simulations to be done on an average single processor notebook computer running Microsoft Windows
- Appropriate simulation time-step for accurate results

* As with previous examples used in presentations, this S-parameter model was provided by an independent third party and not optimized for simulation speed



AMS PRBS Generator

AMS SERDES TX

S-Parameter TX Package, Stub, Connector, Backplane, Connector, Stub, RX Package

AMS RX and Envelope Generator

The VHDL-AMS SERDES Transmitter Model

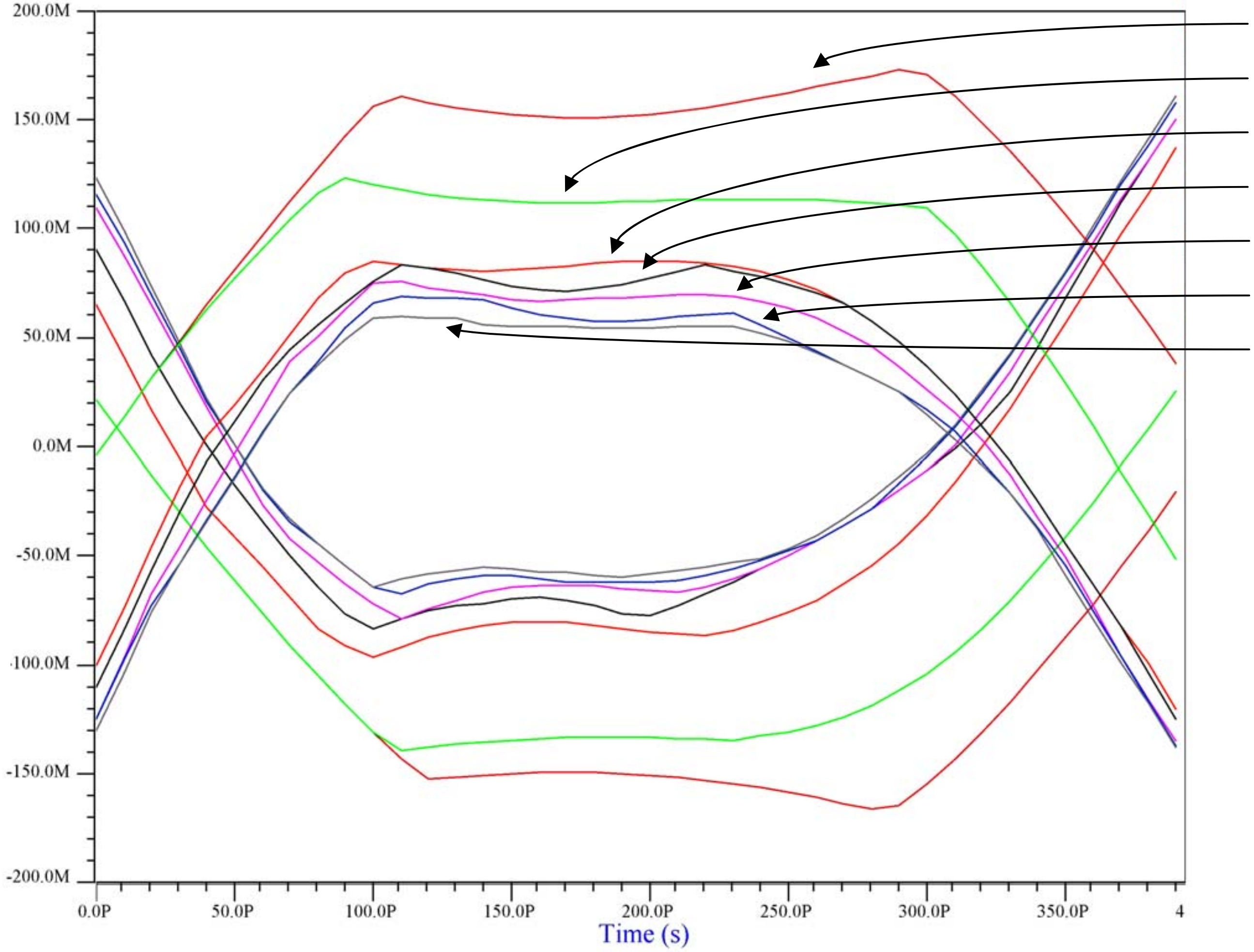
```
begin
  -- output the proper current based on the state of signal din,
  -- and values of constants Ipe and Imain
  if domain = quiescent_domain use  – if DC then
    itxp == Ipe/2.0; itxn == Ipe/2.0;     -- set both outputs to half
  elsif din='1' and din'delayed(bit) = '0' use
    itxp == Ipe; itxn == 0.0;              -- first pulse (txp positive)
  elsif din='1' and din'delayed(bit) = '1' use
    itxp == Imain; itxn == Ipe-Imain; -- normal pulse (txp positive)
  elsif din='0' and din'delayed(bit) = '1' use
    itxp == 0.0; itxn == Ipe;              -- first pulse (txn positive)
  elsif din='0' and din'delayed(bit) = '0' use
    itxp == Ipe-Imain; itxn == Imain; -- normal pulse (txn positive)
  end use;
  break on din, din'delayed(bit) ;   -- deal with the discontinuities

  -- P and N-side C_comp, R_term, Vdd
  i_r_term_p == (vtxp - Vdd)/R_term; i_c_comp_p == c_comp * vtxp'dot;
  i_r_term_n == (vtxn - Vdd)/R_term; i_c_comp_n == c_comp * vtxn'dot;

end architecture;
```

Results: Simulation to 10 Million Data Cycles

(All simulations completed overnight)



10 Cycles

100 Cycles

1,000 Cycles

10,0000 Cycles

100,000 Cycles

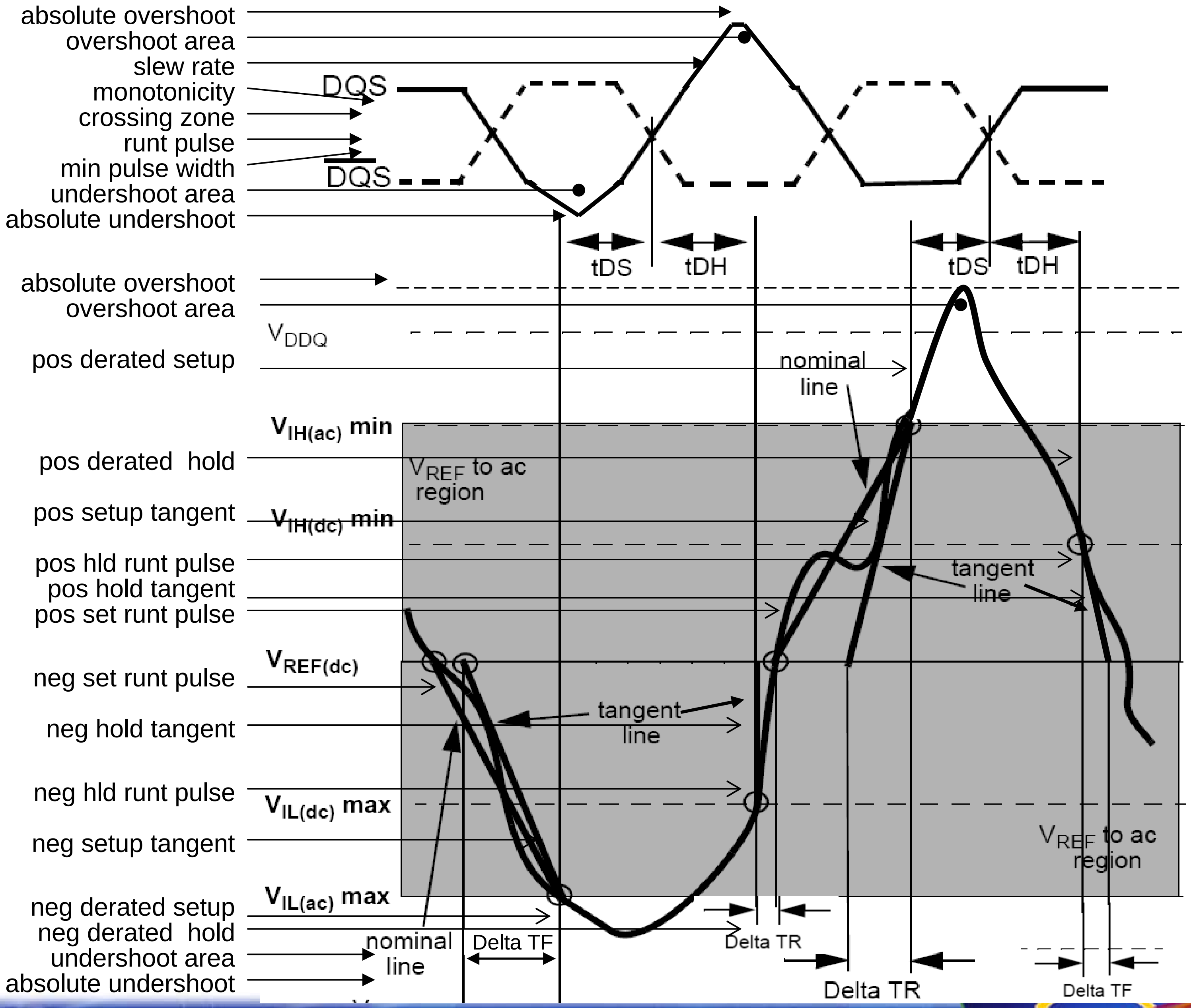
1,000,00 Cycles

10,000,000 Cycles

AMS Case Study Two
Automated DDR2 Measurements

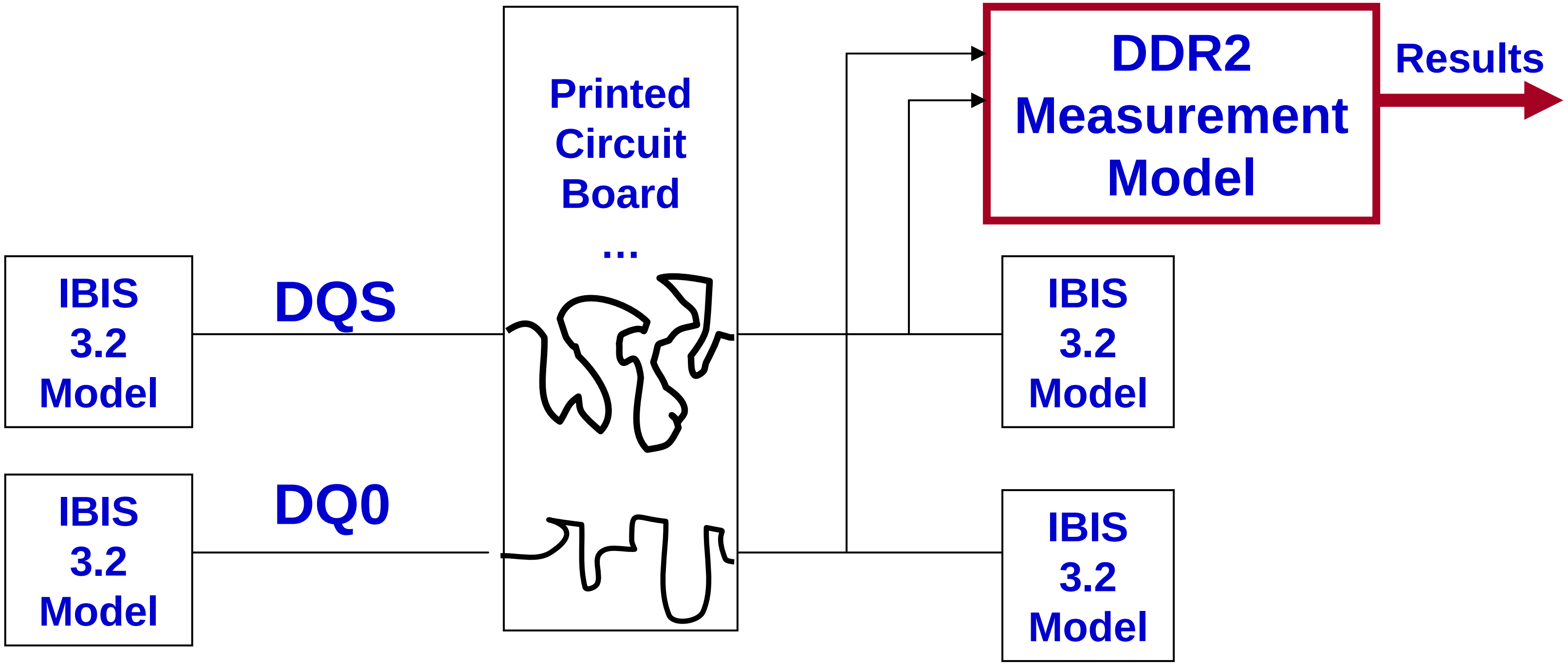
- Implement all measurements specified in the DDR2 datasheet in a VHDL-AMS model
- Utilize standard IBIS 3.2 driver and receiver models

DDR2 Electrical and Timing Constraints

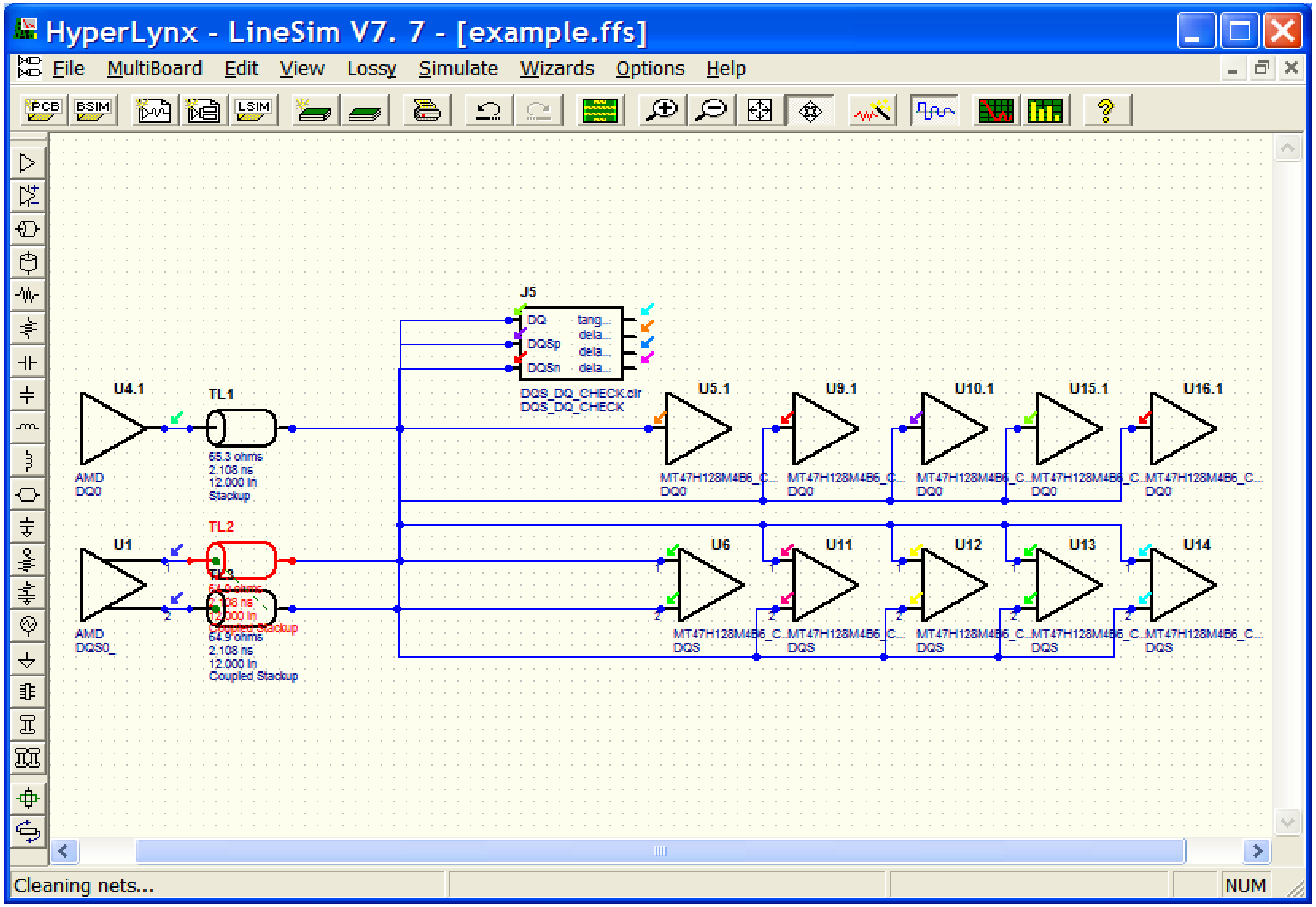

absolute overshoot
overshoot area
slew rate
monotonicity
crossing zone
runt pulse
min pulse width
undershoot area
absolute undershoot

absolute overshoot
overshoot area
pos derated setup
pos derated hold
pos setup tangent
pos hld runt pulse
pos hold tangent
pos set runt pulse
neg set runt pulse
neg hold tangent
neg hld runt pulse
neg setup tangent
neg derated setup
neg derated hold
undershoot area
absolute undershoot

IBIS 4.2 Measurement Model



Pre-layout analysis using the IBIS 4.2 Measurement Model



HyperLynx - LineSim V7. 7 - [example.ffs]

Cleaning nets...

TANGENT MEASUREMENT

Wait for vref crossing

Store data points

Wait for vix_ac cross

Calculate the slope from each point to the vix_ac crossing point

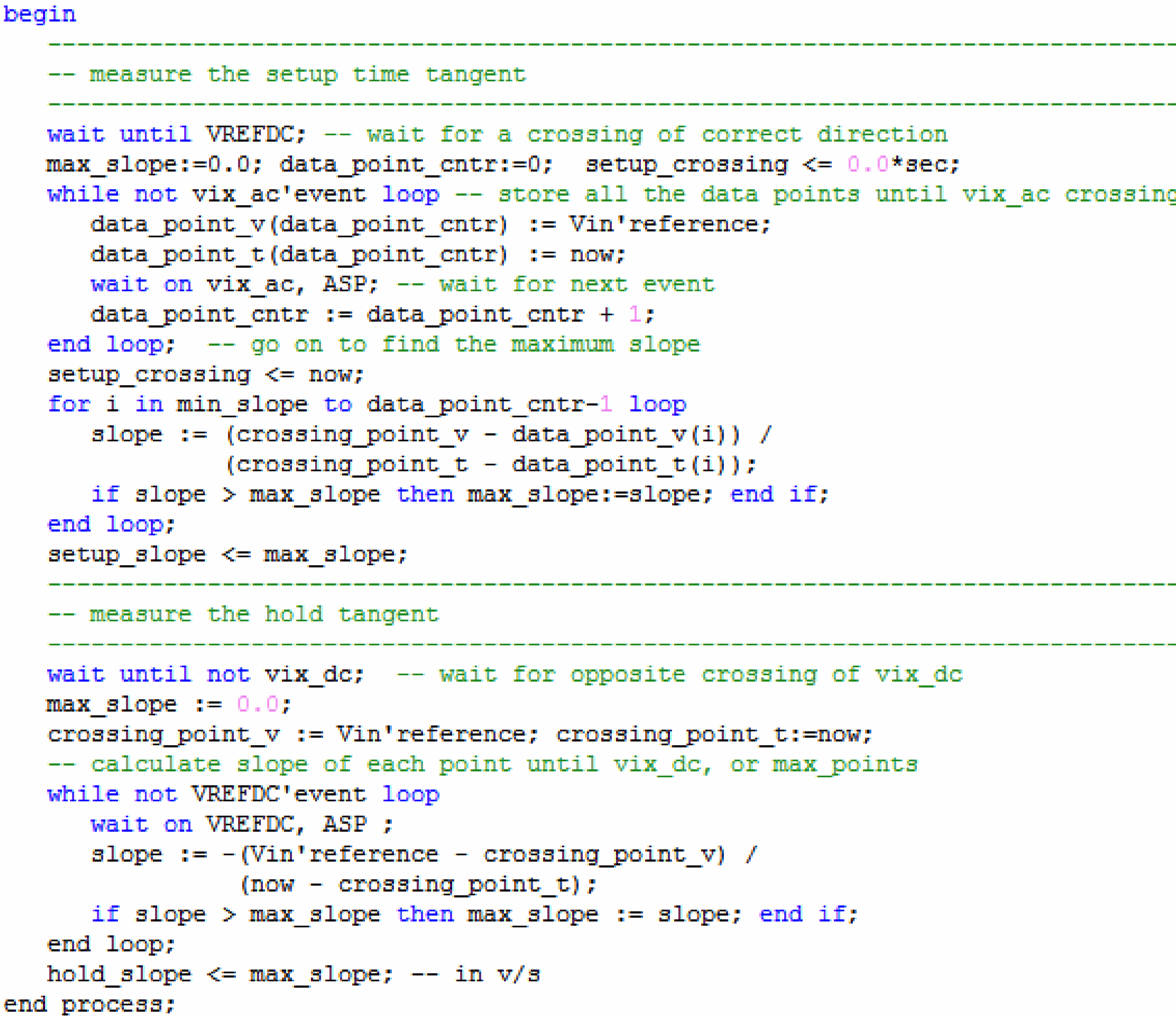
Return the maximum slope

Wait for vix_dc crossing

Calculate the slope from each subsequent point back to the vix_dc crossing

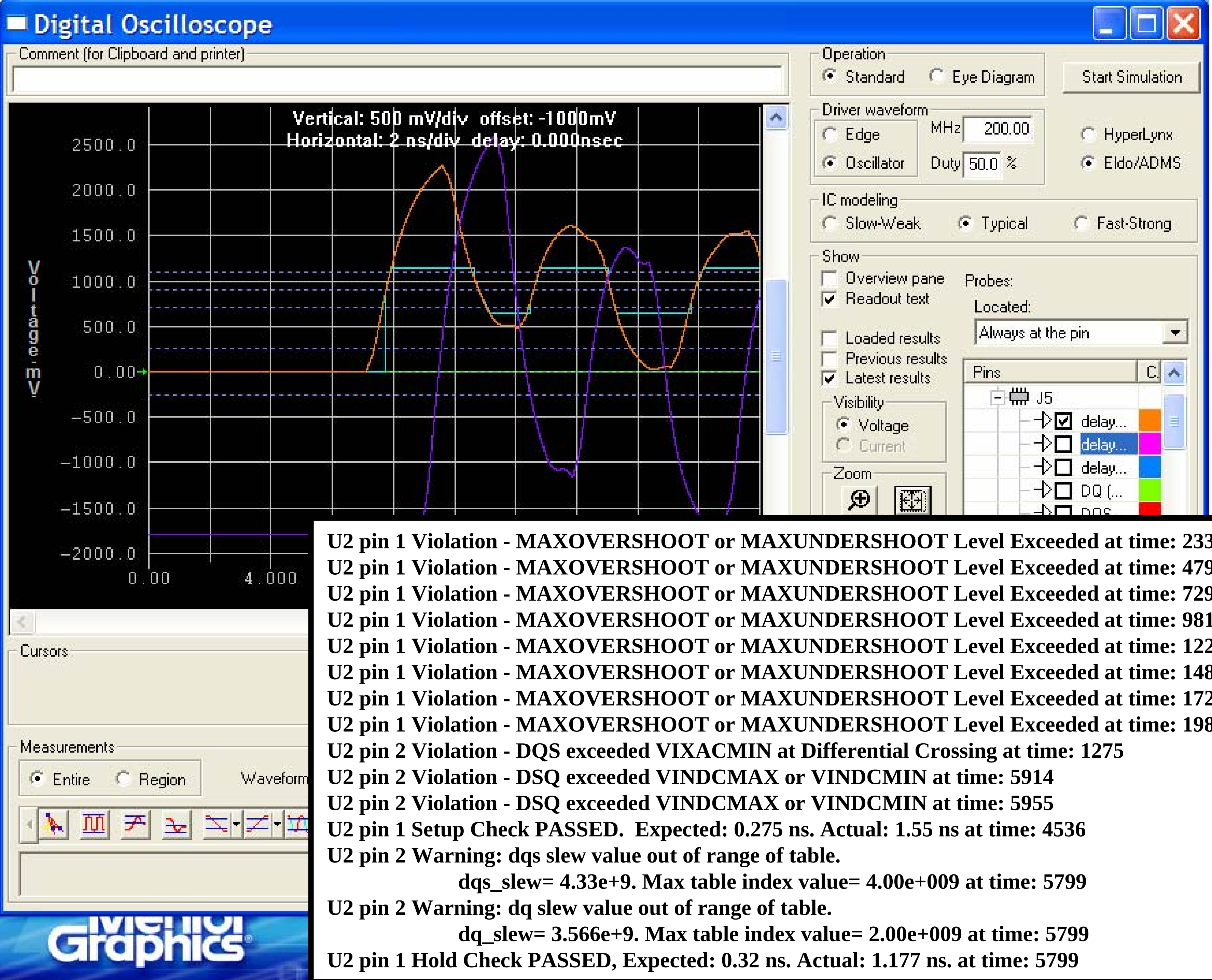
Wait for vref crossing

Return the max slope



```
begin
   ------------------------------------------------------------------------------
   -- measure the setup time tangent
   ------------------------------------------------------------------------------
   wait until VREFDC; -- wait for a crossing of correct direction
   max_slope:=0.0; data_point_cntr:=0;  setup_crossing <= 0.0*sec;
   while not vix_ac'event loop -- store all the data points until vix_ac crossing
      data_point_v(data_point_cntr) := Vin'reference;
      data_point_t(data_point_cntr) := now;
      wait on vix_ac, ASP; -- wait for next event
      data_point_cntr := data_point_cntr + 1;
   end loop;  -- go on to find the maximum slope
   setup_crossing <= now;
   for i in min_slope to data_point_cntr-1 loop
      slope := (crossing_point_v - data_point_v(i)) /
               (crossing_point_t - data_point_t(i));
      if slope > max_slope then max_slope:=slope; end if;
   end loop;
   setup_slope <= max_slope;
   ------------------------------------------------------------------------------
   -- measure the hold tangent
   ------------------------------------------------------------------------------
   wait until not vix_dc;  -- wait for opposite crossing of vix_dc
   max_slope := 0.0;
   crossing_point_v := Vin'reference; crossing_point_t:=now;
   -- calculate slope of each point until vix_dc, or max_points
   while not VREFDC'event loop
      wait on VREFDC, ASP ;
      slope := -(Vin'reference - crossing_point_v) /
                (now - crossing_point_t);
      if slope > max_slope then max_slope := slope; end if;
   end loop;
   hold_slope <= max_slope; -- in v/s
end process;
```

(error and exception handling removed for clarity)



Comment (for Clipboard and printer)

Operation: ● Standard ○ Eye Diagram

Driver waveform: ○ Edge ● Oscillator MHz 200.00 Duty 50.0 % ○ HyperLynx ● Eldo/ADMS

IC modeling: ○ Slow-Weak ● Typical ○ Fast-Strong

Show: □ Overview pane ☑ Readout text □ Loaded results □ Previous results ☑ Latest results

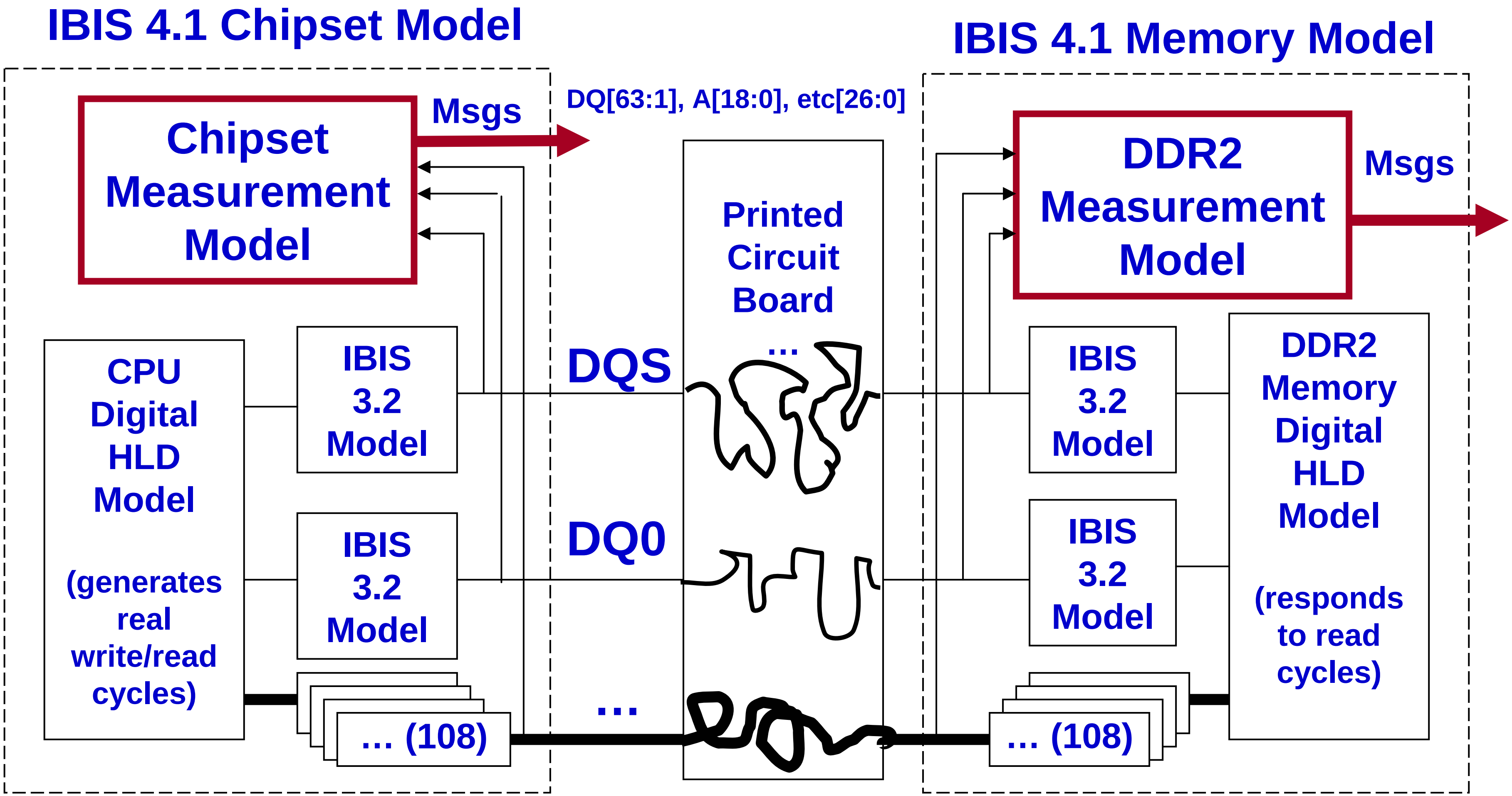
Visibility: ● Voltage ○ Current

Located: Always at the pin

Measurements: ● Entire ○ Region

U2 pin 1 Violation - MAXOVERSHOOT or MAXUNDERSHOOT Level Exceeded at time: 2337
U2 pin 1 Violation - MAXOVERSHOOT or MAXUNDERSHOOT Level Exceeded at time: 4790
U2 pin 1 Violation - MAXOVERSHOOT or MAXUNDERSHOOT Level Exceeded at time: 7290
U2 pin 1 Violation - MAXOVERSHOOT or MAXUNDERSHOOT Level Exceeded at time: 9815
U2 pin 1 Violation - MAXOVERSHOOT or MAXUNDERSHOOT Level Exceeded at time: 12294
U2 pin 1 Violation - MAXOVERSHOOT or MAXUNDERSHOOT Level Exceeded at time: 14816
U2 pin 1 Violation - MAXOVERSHOOT or MAXUNDERSHOOT Level Exceeded at time: 17293
U2 pin 1 Violation - MAXOVERSHOOT or MAXUNDERSHOOT Level Exceeded at time: 19816
U2 pin 2 Violation - DQS exceeded VIXACMIN at Differential Crossing at time: 1275
U2 pin 2 Violation - DSQ exceeded VINDCMAX or VINDCMIN at time: 5914
U2 pin 2 Violation - DSQ exceeded VINDCMAX or VINDCMIN at time: 5955
U2 pin 1 Setup Check PASSED. Expected: 0.275 ns. Actual: 1.55 ns at time: 4536
U2 pin 2 Warning: dqs slew value out of range of table.
            dqs_slew= 4.33e+9. Max table index value= 4.00e+009 at time: 5799
U2 pin 2 Warning: dq slew value out of range of table.
            dq_slew= 3.566e+9. Max table index value= 2.00e+009 at time: 5799
U2 pin 1 Hold Check PASSED, Expected: 0.32 ns. Actual: 1.177 ns. at time: 5799

Model for the Full DDR2 Channel Integrating both Behavior and Measurement *



* Note: This model is not in strictly IBIS 4.2 compliant because it uses an external circuit that references an IBIS 3.1 model

Special thanks to

Randy Wolff and his associates at Micron for assistance in developing DDR2 simulation and measurement models.

Mentor Graphics®

www.mentor.com