A VHDL-AMS true differential buffer model using IBIS v3.2 data

IBIS Summit at DesignCon 2004

Santa Clara Convention Center, CA

February 2, 2004

Arpad Muranyi

Signal Integrity Engineering

Intel Corporation

arpad.muranyi@intel.com

Outline

- Background
- Block diagram of VHDL-AMS model
- Code changes highlighted
- Correlation study
  - Waveform overlays with various loads
- Summary

Background

- **The VHDL-AMS differential buffer model introduced in this presentation is a simple modification of the VHDL-AMS single-ended buffer model introduced at the June 5 and June 23, 2003 IBIS Summits**
  http://www.eda.org/pub/ibis/summits/jun03b/muranyi1.pdf
- **This model also incorporates the modeling technique developed for differential buffers presented at the October 15, 2002 and October 21, 2003 IBIS Summits**
  http://www.eda.org/pub/ibis/summits/oct02/muranyi.pdf
  http://www.eda.org/pub/ibis/summits/oct03/muranyi.pdf
- **Main features:**
  - 1 digital input, 1 digital enable, 1 digital output (dummy for receiver out)
  - 4 analog supplies, 2 analog (differential) I/O ports
  - Uses normal IBIS data (I-V and V-t curve tables) for “common mode” component [Model]
  - Uses fitted coefficients (calculated from the I-V tables of the [Series MOSFET] model) for the “differential” component
  - Includes 4-way split C_comp plus C_diff
  - This presentation is accompanied by a VHDL-AMS file (IBIS_diff.vhd) which is made available freely to anyone interested
  - This is done to encourage the use of the *-AMS extensions of IBIS for improved behavioral modeling

Block diagram of the VHDL-AMS model

Library calls

Entity
- **generics** — Added C_diff, k0-k5 for I_diff
- **ports** — Added 2nd port for inverting I/O pad

Architecture
- **quantities**
- **signals**
- **functions**
  - **lookup**
  - **common length**
  - **common time**
  - **common wfm**
  - **coeff**

Doubled all quantities for 2nd port and added two more for the differential components

Processes
- **catch**
- **logic**

Break statements
Simultaneous equations to select coefficients
Simultaneous equations to calculate output currents

Doubled all equations for 2nd port and added two more for the differential components

VHDL-AMS implementation – changes (1)

```
entity IBIS_DIFF_IO is

  generic (C_comp       : real := 1.00e-12;  -- Default C_comp value and
           k_C_comp_pc : real := 0.25;       -- splitting coefficients
           k_C_comp_pu : real := 0.25;
           k_C_comp_pd : real := 0.25;
           k_C_comp_gc : real := 0.25;
           C_diff       : real := 50.0e-15;  -- Default C_diff value (50.0fF)

           ------------------------------------------------------------------
           -- [Pullup Reference] and [Pulldown Reference] values
           ------------------------------------------------------------------
           V_pu_ref : real := 1.8;
           V_pd_ref : real := 0.0;

           ------------------------------------------------------------------
           -- Coefficients of Idiff surface (from Matlab surface fitting)
           ------------------------------------------------------------------
           k0 : real := -6.503179353194756e-006;
           k1 : real :=  2.541816815085296e-003;
           k2 : real := -2.541334083148360e-003;
           k3 : real :=  2.809854297776799e-005;
           k4 : real := -4.580644144607367e-004;
           k5 : real :=  4.354430013260378e-004;

           R_diff : real := 700.0;  -- In case a linear resistor does the job

-----------------------------------------------------------------------------
-- Vectors of the IV curve tables
-----------------------------------------------------------------------------
```

VHDL-AMS implementation – changes (2)

```
--------------------------------------------------------------------------------
-- V_fixture and R_fixture values
--------------------------------------------------------------------------------
           Vfx_pu_on  : real := 1.0;
           Vfx_pu_off : real := 1.0;
           Vfx_pd_on  : real := 1.5;
           Vfx_pd_off : real := 1.5;

           Rfx_pu_on  : real := 50.0;
           Rfx_pu_off : real := 50.0;
           Rfx_pd_on  : real := 50.0;
           Rfx_pd_off : real := 50.0;
           -------------------------------------------------------------------
           Delta_t    : real := 1.0e-12);      -- This parameter
           -- determines what the maximum time delta will be between the
           -- points of the Vt curves and scaling coefficient curves
           -- after preprocessing the input data.
     -------------------------------------------------------------------------
     port (signal   In_D   : in   std_logic;
           signal   En_D   : in   std_logic;
           signal   Rcv_D  : out  std_logic;

           terminal IO_p   :      electrical;
           terminal IO_n   :      electrical;
           terminal PC_ref :      electrical;
           terminal PU_ref :      electrical;
           terminal PD_ref :      electrical;
           terminal GC_ref :      electrical);

end entity IBIS_DIFF_IO;
--==============================================================================
```

VHDL-AMS implementation – changes (3)

```
--============================================================================
architecture Diff_IO_2eq of IBIS_DIFF_IO is
  ----------------------------------------------------------------------------
  -- Common mode components for IV curves
  ----------------------------------------------------------------------------
  quantity  Vpc_p    across   Ipc_p    through  PC_ref  to  IO_p;
  quantity  Vpu_p    across   Ipu_p    through  PU_ref  to  IO_p;
  quantity  Vpd_p    across   Ipd_p    through  IO_p    to  PD_ref;
  quantity  Vgc_p    across   Igc_p    through  IO_p    to  GC_ref;

  quantity  Vpc_n    across   Ipc_n    through  PC_ref  to  IO_n;
  quantity  Vpu_n    across   Ipu_n    through  PU_ref  to  IO_n;
  quantity  Vpd_n    across   Ipd_n    through  IO_n    to  PD_ref;
  quantity  Vgc_n    across   Igc_n    through  IO_n    to  GC_ref;
  ----------------------------------------------------------------------------
  -- Common mode components for C_comp
  ----------------------------------------------------------------------------
  quantity  Vc_pc_p  across   Ic_pc_p  through  PC_ref  to  IO_p;
  quantity  Vc_pu_p  across   Ic_pu_p  through  PU_ref  to  IO_p;
  quantity  Vc_pd_p  across   Ic_pd_p  through  IO_p    to  PD_ref;
  quantity  Vc_gc_p  across   Ic_gc_p  through  IO_p    to  GC_ref;

  quantity  Vc_pc_n  across   Ic_pc_n  through  PC_ref  to  IO_n;
  quantity  Vc_pu_n  across   Ic_pu_n  through  PU_ref  to  IO_n;
  quantity  Vc_pd_n  across   Ic_pd_n  through  IO_n    to  PD_ref;
  quantity  Vc_gc_n  across   Ic_gc_n  through  IO_n    to  GC_ref;
  ----------------------------------------------------------------------------
  -- Differential IV surface and C_comp
  ----------------------------------------------------------------------------
  quantity  V_pn     across   I_pn     through  IO_p    to  IO_n;
  quantity  Vc_diff  across   Ic_diff  through  IO_p    to  IO_n;
  ----------------------------------------------------------------------------
```

VHDL-AMS implementation – changes (4)

```
  ---------------------------------------------------------------------------
  -- Various signals and quantities (for internal calculations)
  ---------------------------------------------------------------------------
  signal     pu_on       : std_logic := '0';
  signal     pu_off      : std_logic := '0';
  signal     pd_on       : std_logic := '0';
  signal     pd_off      : std_logic := '0';

  signal     State_D     : std_logic := 'U';

  signal     In_time     : real := 0.0;
  signal     En_time     : real := 0.0;
  signal     Event_time  : real := 0.0;

  quantity   k_pu_p      : real := 0.0;
  quantity   k_pd_p      : real := 0.0;
  quantity   k_pu_n      : real := 0.0;
  quantity   k_pd_n      : real := 0.0;
--=============================================================================
```

VHDL-AMS implementation – changes (5)

```
--============================================================================
  -- This section contains the simultaneous analog equations to find the
  -- appropriate scaling coefficients according to the state the buffer.
  ---------------------------------------------------------------------------
  if (Event_time = 0.0) use                              -- Initialization
    if (State_D = '1') use                               -- Start with the end of the
      k_pu_p == K_pu_on(K_pu_on'right);                  -- Vt curves for those which
      k_pd_p == K_pd_off(K_pd_off'right);                -- are fully on initially
      k_pd_n == K_pd_on(K_pd_on'right);
      k_pu_n == K_pu_off(K_pu_off'right);

    elsif (State_D = '0') use
      k_pd_p == K_pd_on(K_pd_on'right);
      k_pu_p == K_pu_off(K_pu_off'right);
      k_pu_n == K_pu_on(K_pu_on'right);
      k_pd_n == K_pd_off(K_pd_off'right);

    else
      k_pu_p == 0.0;
      k_pd_p == 0.0;
      k_pu_n == 0.0;
      k_pd_n == 0.0;
    end use;

  else                                  -- Look up coefficients in normal operation
    . . .
    . . .
    . . .
```

VHDL-AMS implementation – changes (6)

```
else                              -- Look up coefficients in normal operation

   if    (pu_on  = '1') use
     k_pu_p == Lookup("Vt", now - Event_time, K_pu_on,  T_common);
     k_pd_n == Lookup("Vt", now - Event_time, K_pd_on,  T_common);
   elsif (pu_off = '1') use
     k_pu_p == Lookup("Vt", now - Event_time, K_pu_off, T_common);
     k_pd_n == Lookup("Vt", now - Event_time, K_pd_off, T_common);
   else
     k_pu_p == K_pu_on(K_pu_on'left);
     k_pd_n == K_pd_on(K_pd_on'left);
   end use;


   if    (pd_on  = '1') use
     k_pd_p == Lookup("Vt", now - Event_time, K_pd_on,  T_common);
     k_pu_n == Lookup("Vt", now - Event_time, K_pu_on,  T_common);
   elsif (pd_off = '1') use
     k_pd_p == Lookup("Vt", now - Event_time, K_pd_off, T_common);
     k_pu_n == Lookup("Vt", now - Event_time, K_pu_off, T_common);
   else
     k_pd_p == K_pd_on(K_pd_on'left);
     k_pu_n == K_pu_on(K_pu_on'left);
   end use;

 end use;
--============================================================================
```

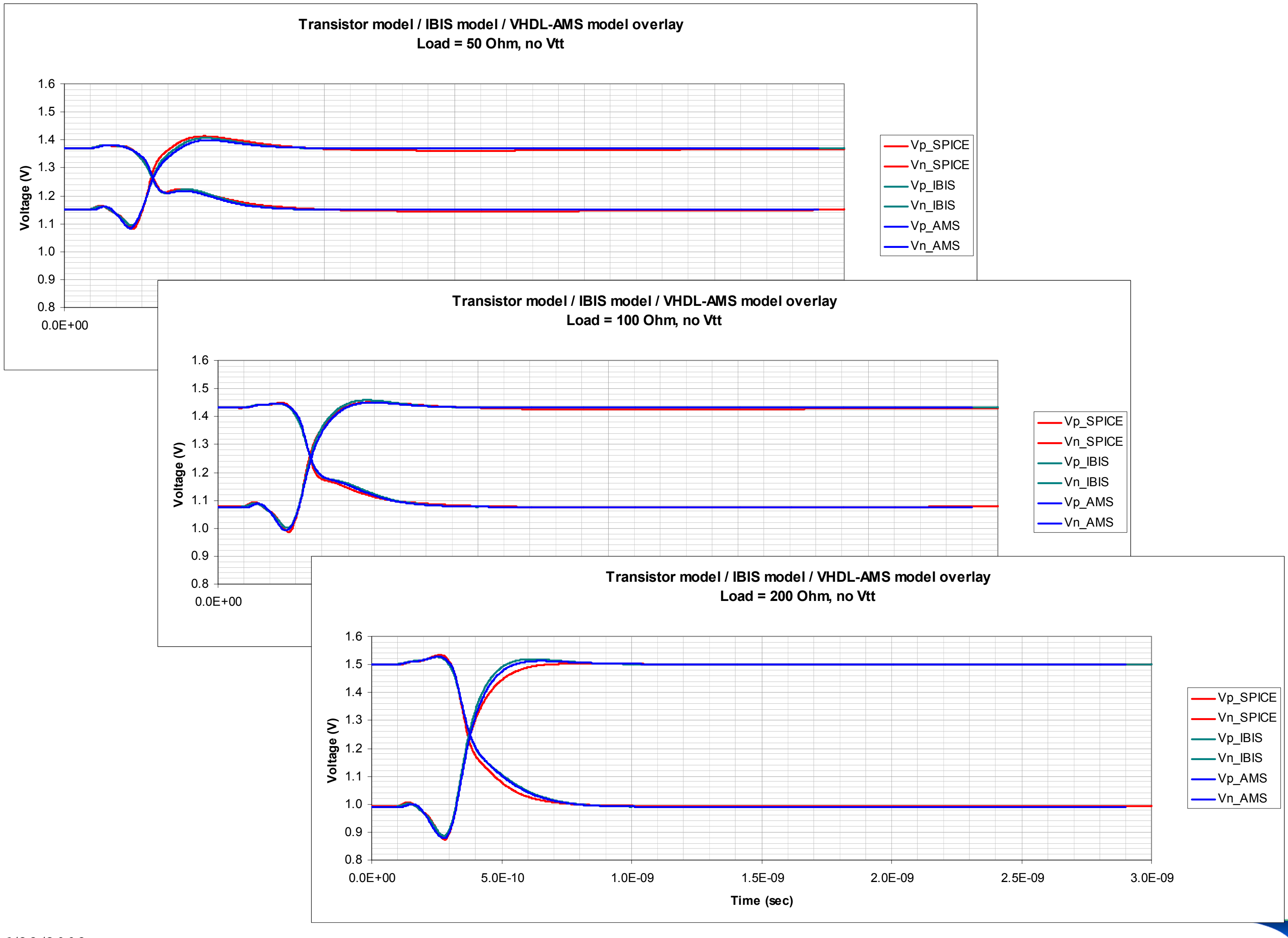
VHDL-AMS implementation – changes (7)

```
---------------------------------------------------------------------------
  -- Common mode components for IV curves
  ---------------------------------------------------------------------------
  Ipc_p   == -1.0          * Lookup("IV", Vpc_p, I_pc, V_pc);
  Ipu_p   == -1.0 * k_pu_p * Lookup("IV", Vpu_p, I_pu, V_pu);
  Ipd_p   ==        k_pd_p * Lookup("IV", Vpd_p, I_pd, V_pd);
  Igc_p   ==                 Lookup("IV", Vgc_p, I_gc, V_gc);

  Ipc_n   == -1.0          * Lookup("IV", Vpc_n, I_pc, V_pc);
  Ipu_n   == -1.0 * k_pu_n * Lookup("IV", Vpu_n, I_pu, V_pu);
  Ipd_n   ==        k_pd_n * Lookup("IV", Vpd_n, I_pd, V_pd);
  Igc_n   ==                 Lookup("IV", Vgc_n, I_gc, V_gc);
  ---------------------------------------------------------------------------
  -- Common mode components for C_comp
  ---------------------------------------------------------------------------
  Ic_pc_p == k_C_comp_pc * C_comp * Vc_pc_p'dot;
  Ic_pu_p == k_C_comp_pu * C_comp * Vc_pu_p'dot;
  Ic_pd_p == k_C_comp_pd * C_comp * Vc_pd_p'dot;
  Ic_gc_p == k_C_comp_gc * C_comp * Vc_gc_p'dot;

  Ic_pc_n == k_C_comp_pc * C_comp * Vc_pc_n'dot;
  Ic_pu_n == k_C_comp_pu * C_comp * Vc_pu_n'dot;
  Ic_pd_n == k_C_comp_pd * C_comp * Vc_pd_n'dot;
  Ic_gc_n == k_C_comp_gc * C_comp * Vc_gc_n'dot;
  ---------------------------------------------------------------------------
  -- Differential IV surface and C_comp
  ---------------------------------------------------------------------------
  I_pn    == k0 + k1*Vpd_p + k2*Vpd_n + k3*Vpd_p*Vpd_n + k4*(Vpd_p**2) + k5*(Vpd_n**2);
--  I_pn    == V_pn / R_diff;       -- In case a linear resistor does the job

  Ic_diff == C_diff * Vc_diff'dot;
--===========================================================================
```

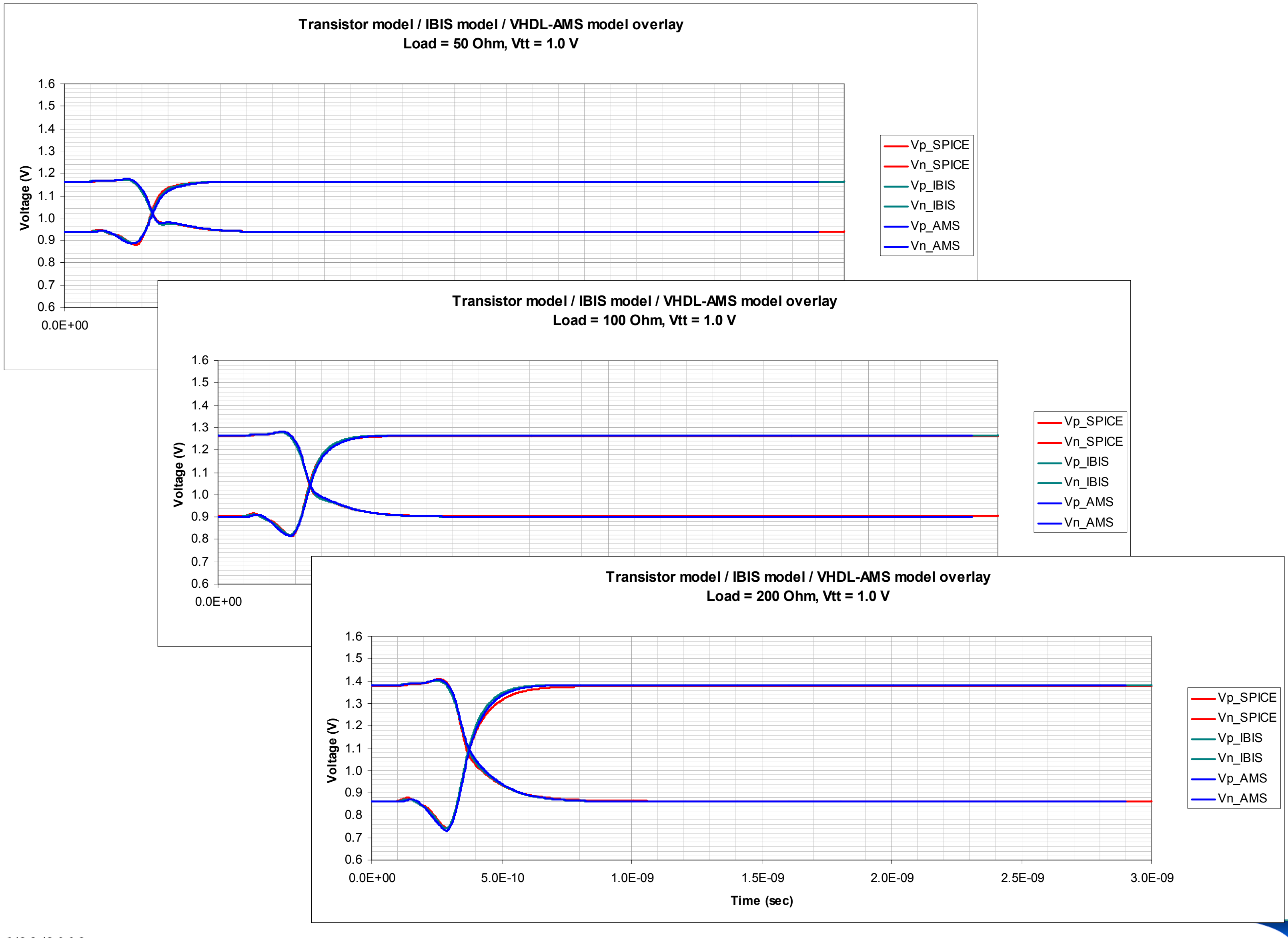
Correlation study

- **The same test cases which were used in the correlation work of the October 21, 2003 presentation were repeated here using the VHDL-AMS model**
  - 3 resistor values (50, 100, 200 Ω pin-to-pin)
  - 3 Vtt values (no source ≈ 1.25 V, 1.0 V, 1.5 V)
  - Open (no load condition)
- **The overlays on the following pages show the waveforms of the original transistor model, IBIS/HSPICE B-element model, and the VHDL-AMS model**

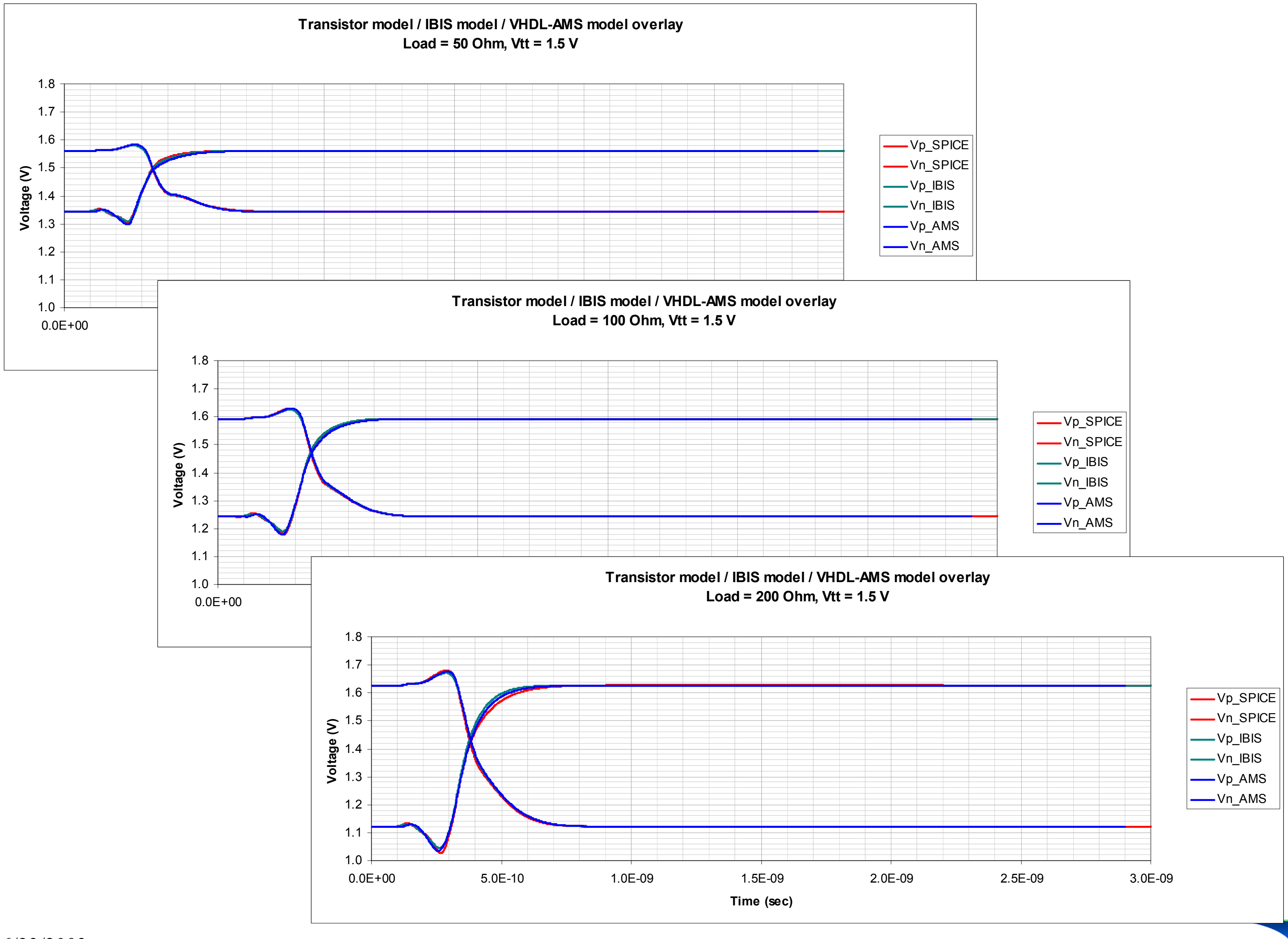
Correlation – no Vtt source



Correlation – Vtt = 1.0 V

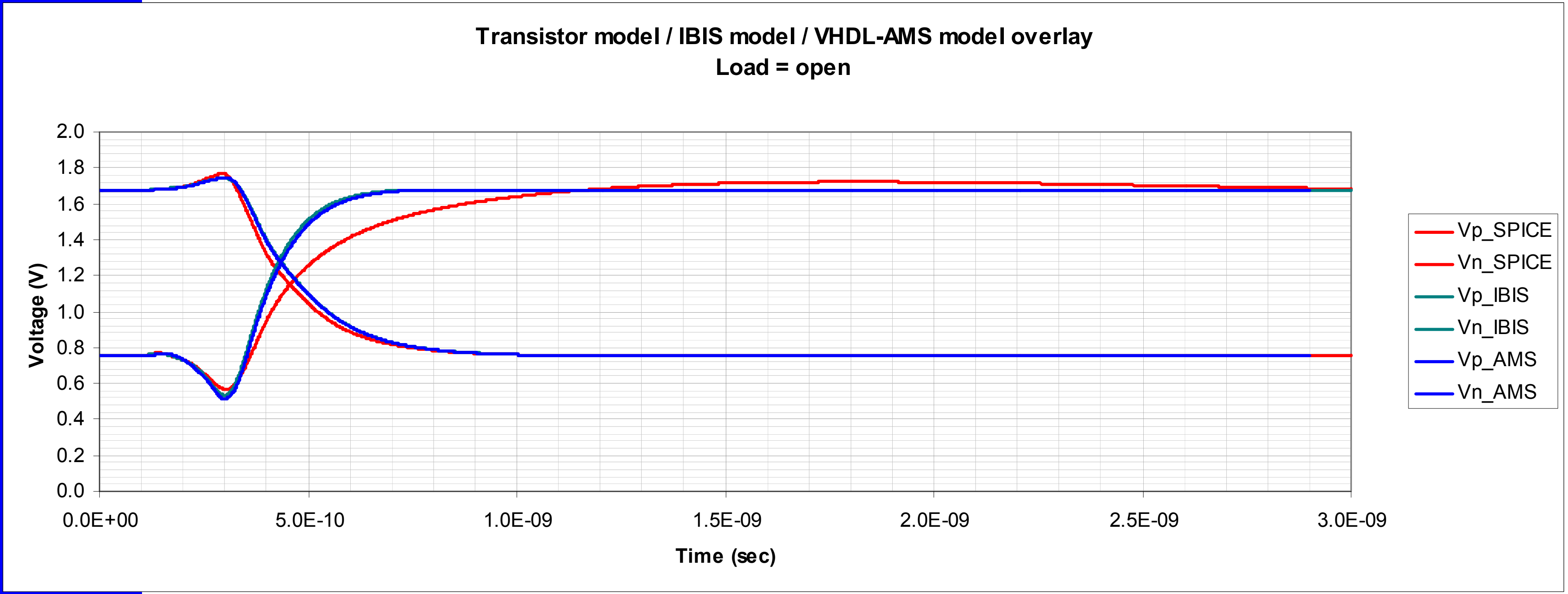


Correlation – Vtt = 1.5 V



6/23/2003
*Other brands and names are the property of their respective owners

Correlation – no load (open)



6/23/2003
*Other brands and names are the property of their respective owners

CPD

Summary

- **A modification of the basic VHDL-AMS I/O buffer model has been shown**
  - http://www.eda.org/pub/ibis/summits/feb04a/IBIS_diff.vhd
  - Feel free to download and use the file any way you want
- **The modified model is a “true differential” I/O model**
  - It has two analog I/O ports
  - This model can be used with [External Model] in IBIS v4.1 (not tested yet)
- **Good correlation with the original transistor model and the HSPICE B-element + [Series MOSFET] implementation**
  - The deviations are similar to the ones seen with the HSPICE B-element / [Series MOSFET] implementation
  - These discrepancies are due to higher order effects in the transistor model which are not captured in the IBIS data used with these behavioral models