De-emphasis Buffer Modeling
Issues with IBIS

Nanditha Rao
Intel Corporation
nanditha.p.rao@intel.com

Asian IBIS Summit
Tokyo, Japan
November 14, 2008
Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2008, Intel Corporation. All rights reserved.
Agenda

• Brief background
  – Differential buffer with de-emphasis
  – De-emphasis implementation in IBIS

• Different de-emphasis implementation
  – A different de-emphasis approach
  – Boost stimulus derivation
  – Example boost stimulus in SPICE

• Modeling the “new” boost
  – Issues with defining the boost
  – Single-ended or differential?
  – Verilog* AMS approach

• Comments and future work

*Other names and brands may be claimed as the property of others
Differential Buffer

• PCI Express* differential buffer example

Note: NBIAS and PBIAS are set to get 10mA from the driver
Differential Buffer with De-emphasis

- Adding a second buffer (de-emphasis/boost)- WIRED-OR
- Boost adds to the main during full-swing (101010) operation

In_p and In_n are main stimulus patterns
Boost_p = In_p
Boost_n = In_n

*Other names and brands may be claimed as the property of others
Differential Buffer with De-emphasis

During de-emphasized operation (continuous ones/zeros), opposite legs of the buffer are switched on (one of the methodologies).

Boost_p = \overline{\text{In}_p}

Boost_n = \overline{\text{In}_n
De-emphasis Implementation in IBIS

- Two differential buffers—main and boost are WIRED-OR
- Terminations and parasitics shouldn’t be duplicated
- Boost is driven by a stimulus derived from the main pattern depending on the de-emphasis logic
- $C_{\text{comp}}$ is split between main and boost in the ratio of number of driver slices (one of the ways)

$\text{Boost (t)} = \text{Inp (t-1)}$
De-emphasis - Different Approach

• During de-emphasized operation, the boost buffer is completely \textbf{shut off}. Load sees only main

You can no-longer tie these inputs together

*Other names and brands may be claimed as the property of others
Boost Stimulus... Some Derivations

Boost input stimulus will need to be different

<table>
<thead>
<tr>
<th>Inp</th>
<th>Boost_NDP</th>
<th>Boost_PDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Boost_NDP = Inp(t-1) . Inp(t)

Boost_PDP = Inp(t-1) . Inp(t)

Boost_PDP = Boost_PDP

Boost_NDN = Inn(t-1) . Inn(t)

Boost_PDN = Inn(t-1) . Inn(t)

Boost_PDN = Boost_PDN

Similar logic applies for ndn and pdn, except that they will follow Inn

*Other names and brands may be claimed as the property of others*
Example Boost Stimulus in SPICE

**************************************************** Logic for Boost_NDP****************************************************
E_inx    inpdelay 0 DELAY inp          0 TD='period/2'     ****Delay
E_intn   in2    0               vol='inV-V(inpdelay)'    **Inversion of the delay
**AND inp and in2
Gswitch  Boost_NDP inp VCR PWL(1) inp 0 0v,1e-6 vcc,10m
Gswitch1 Boost_NDP in2 VCR PWL(1) in2 0 0v,1e-6 vcc,10m

Rxxx  Boost_NDP 0 1E6   *** OUTPUT LOAD

****************************************************Logic for Boost_PDP****************************************************
E_inx1   inpdelay1 0 DELAY inp          0 TD='period/2'    **Delay
E_inn1   inpbar   0               vol='inV-V(inp)'      **Inversion

Gswitch2 inpbar VCR PWL(1) inpbar 0 0v,1e-6 vcc,10m
Gswitch3 inpbar inpdelay1 VCR PWL(1) inpdelay1 0 0v,1e-6 vcc,10m

E_boostp Boost_PDP 0      vol='inV-V(inpbar)'    **Inversion

Rxxx1 Boost_PDP 0 1E6   *** OUTPUT LOAD

Similarly for NDN and PDN
The ‘Working’ Stimulus

NDP=0 (OFF) during de-emphasis
N follows i/p during normal mode

PDP=1 (OFF) during de-emphasis
P follows i/p during normal mode

NDN=0 (OFF) during de-emphasis
N follows i/p during normal mode

PDN=1 (OFF) during de-emphasis
P follows i/p during normal mode

*Other names and brands may be claimed as the property of others
Modeling the ‘New’ Boost

- Approach 1: Define 2 B-elements for main and 1 B-element for each of pdn, pdp, ndp and ndn (modeled as differential buffer)

Doesn’t work as Boost == 2 differential buffers

*Other names and brands may be claimed as the property of others
Modeling the ‘New’ Boost

• Approach 2: Define either pdp and pdn or ndp and ndn (modeled as differential buffer)

**Differential Main buffer**

- Non-inverting
- Inverting

**Differential Boost buffer**

- Non-inverting
- Inverting

- Differential swing correlates well
- Mismatch on de-emphasis transition
- C_comp distribution gets tricky

Recall A.Muranyi’s

“Data Dependent Buffer Characteristics”

Jan 2003

*Other names and brands may be claimed as the property of others*
A Further Issue

However, single ended signals are incorrect as we ignored ndn and ndp

Transistor

IBIS

6-dB De-emphasis

- Half the driver is switched off during switch from transition to non-transition ==> Odd behavior in transistor
- Table-based IBIS cannot mimic this behavior

*Other names and brands may be claimed as the property of others
Earlier De-emp Approach

Current reverses through boost buffer during de-emphasis

Miscorrelation was not observed with the previous de-emp methodology

Fairly good C_comp distribution

*Other names and brands may be claimed as the property of others
Modeling the ‘New’ Boost

• Approach 3:
  – Model boost as single-ended buffer instead of differential
  – Define 4 B-elements- with pdp, pdn, ndp and ndn stimulus

*Other names and brands may be claimed as the property of others
**Verilog* AMS Model**

- **Approach 4:**
  - Create equation based Verilog*-AMS model for the differential buffer
  - Shows good correlation

![Graph depicting waveform comparison between Transistor and Verilog* A](image-url)
Comments and Future Work

- Different de-emphasis implementations will need different boost stimulus logic
- Correlation issues observed while modeling the new de-emphasis buffer using table-based ibis
  - Manual fine tuning could fix this, but tedious and not the best approach
- Verilog* AMS approach provides good correlation
- Alternatives for investigation:
  - Driver schedule
  - AMS approach

*Other names and brands may be claimed as the property of others