Modeling Pre/de-emphasis buffers with [Driver Schedule]

IBIS Summit at DesignCon 2005

Santa Clara Convention Center, CA

January 31, 2005

Arpad Muranyi
Signal Integrity Engineering
Intel Corporation
arpad.muranyi@intel.com

Background

- **The [Driver Schedule] keyword was first introduced in the IBIS 3.2 specification (ratified August 1999)**
  - It was intended to be used to model multi-staged slew rate controlled buffers, such as the GTL buffers used in Intel's microprocessors
  - The specification viewed this keyword as a temporary solution

```
|               Note: In a future release, the [Driver Schedule] keyword may
|               be replaced by a newer method of specification that is
|               consistent with some other planned extensions.  However, the
|               [Driver Schedule] syntax will continue to be supported.
```

- **Four clarification BIRDs have been written since the keyword first appeared:**

• BIRD52	"[Driver Schedule] Clarifications"	July 17, 1998
• BIRD58.3	"Driver Schedule Keyword Clarification"	May 28, 1999
• BIRD84.1	"Driver Schedule Clarifications"	December 5, 2003
• BIRD88.3	"Driver schedule initialization"	July 16, 2004

- **The last two BIRDs were prompted by questions raised by tool vendors still trying (or just starting) to implement the keyword**
- **Perspective: Pre/de-emphasis buffers started to appear around 2001**

Basic operation of the [Drive Schedule] keyword

- **The [Driver Schedule] keyword can have one or more [Model] references, each of which has its own set of delay parameters**
  - This was designed to model the sequence in which the various stages of a multi-staged, slew rate controlled buffer are turned on or off with respect to the stimulus signal
  - Later on this keyword was also used to model the so-called P-kickers (pullup devices which are turned on for a short amount of time during a rising edge transition)
- **Each of the models referenced by a [Driver Schedule] keyword is stimulated by a common stimulus signal**
- **The delay parameters of the keyword define simple timing relationships between the referenced [Model]'s pullup and/or pulldown structures with respect to the stimulus**

The meaning of the [Driver Schedule] parameters

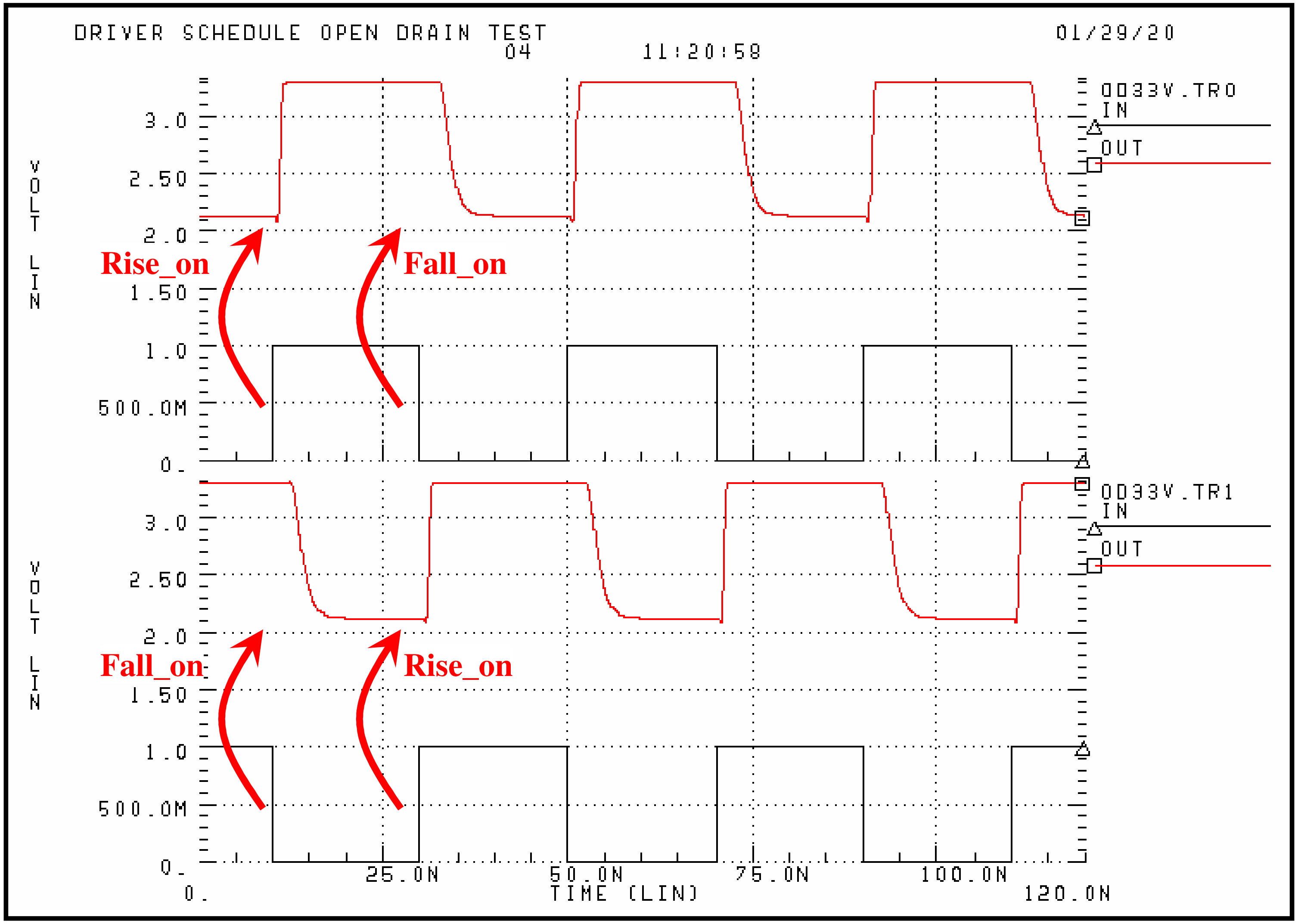
Assuming non-inverting polarity

- **Rise_on_dly:**
  - Triggered by rising edge stimulus
    - Pullup ON
    - Pulldown OFF
- **Rise_off_dly:**
  - Triggered by rising edge stimulus
    - Pullup OFF
    - Pulldown ON
- **Fall_on_dly:**
  - Triggered by falling edge stimulus
    - Pulldown ON
    - Pullup OFF
- **Fall_off_dly:**
  - Triggered by falling edge stimulus
    - Pulldown OFF
    - Pullup ON

Whichever comes first

“NA” means “not available” (i.e. without any information we can’t do anything) which is *not the same* as a zero delay

Rise_on_dly	Rise_off_dly	Fall_on_dly	Fall_off_dly
0.0ns	NA	0.0ns	NA



Rise_on

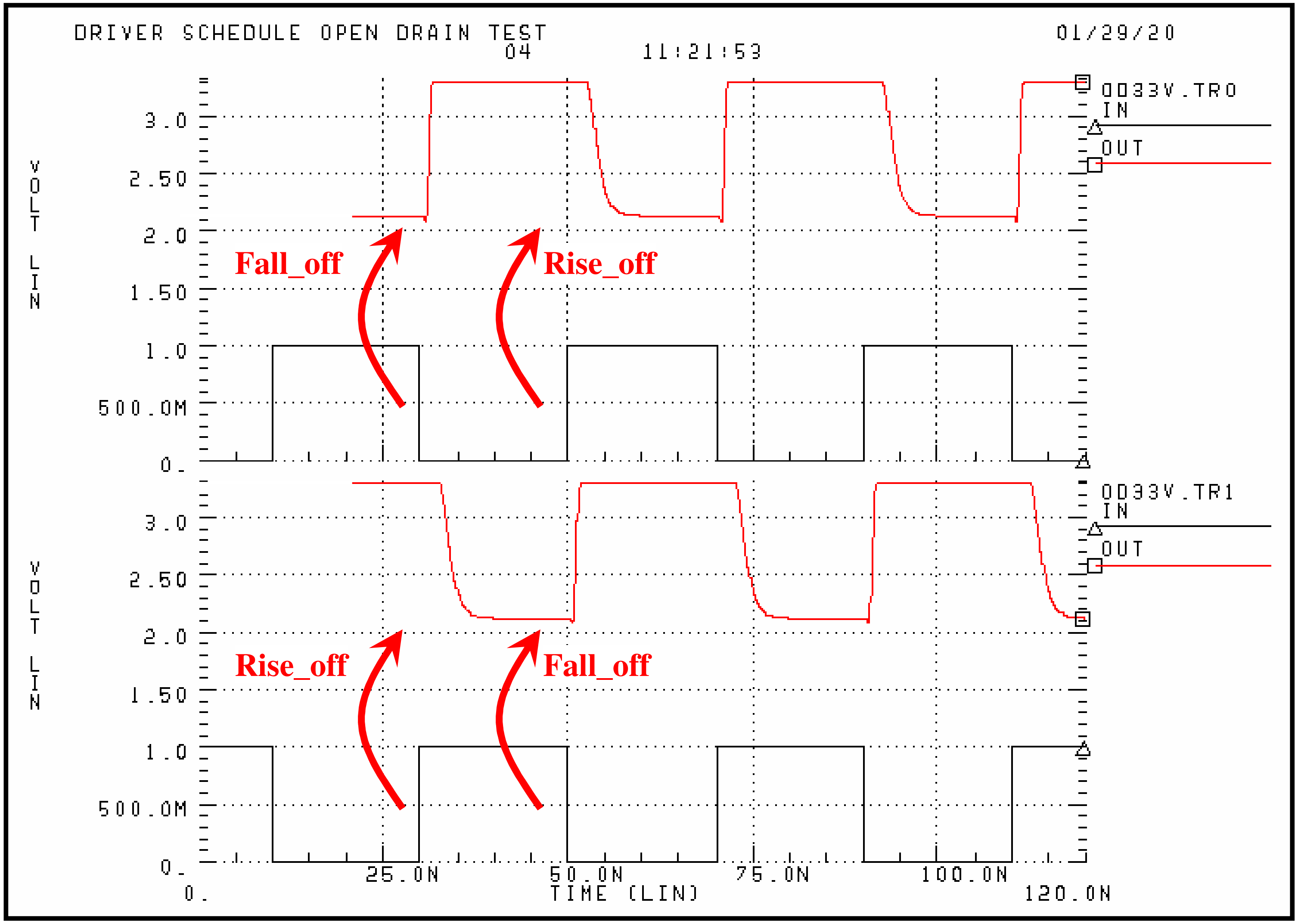
Fall_on

Fall_on

Rise_on

An open-drain scheduled buffer with zero ON delays = normal buffer operation

Rise_on_dly	Rise_off_dly	Fall_on_dly	Fall_off_dly
NA	0.0ns	NA	0.0ns



An open-drain scheduled buffer with zero OFF delays = inverted buffer

Pre/de-emphasis buffer review

In most of the current two-tap designs the "emphasis stimulus pattern" is a one bit delayed and inverted copy of the "input stimulus pattern"

This is not necessarily true for all pre/de-emphasis buffer designs. The delay may not be a one bit duration in each design, and multi-tap configurations would usually have a more complicated stimulus logic.

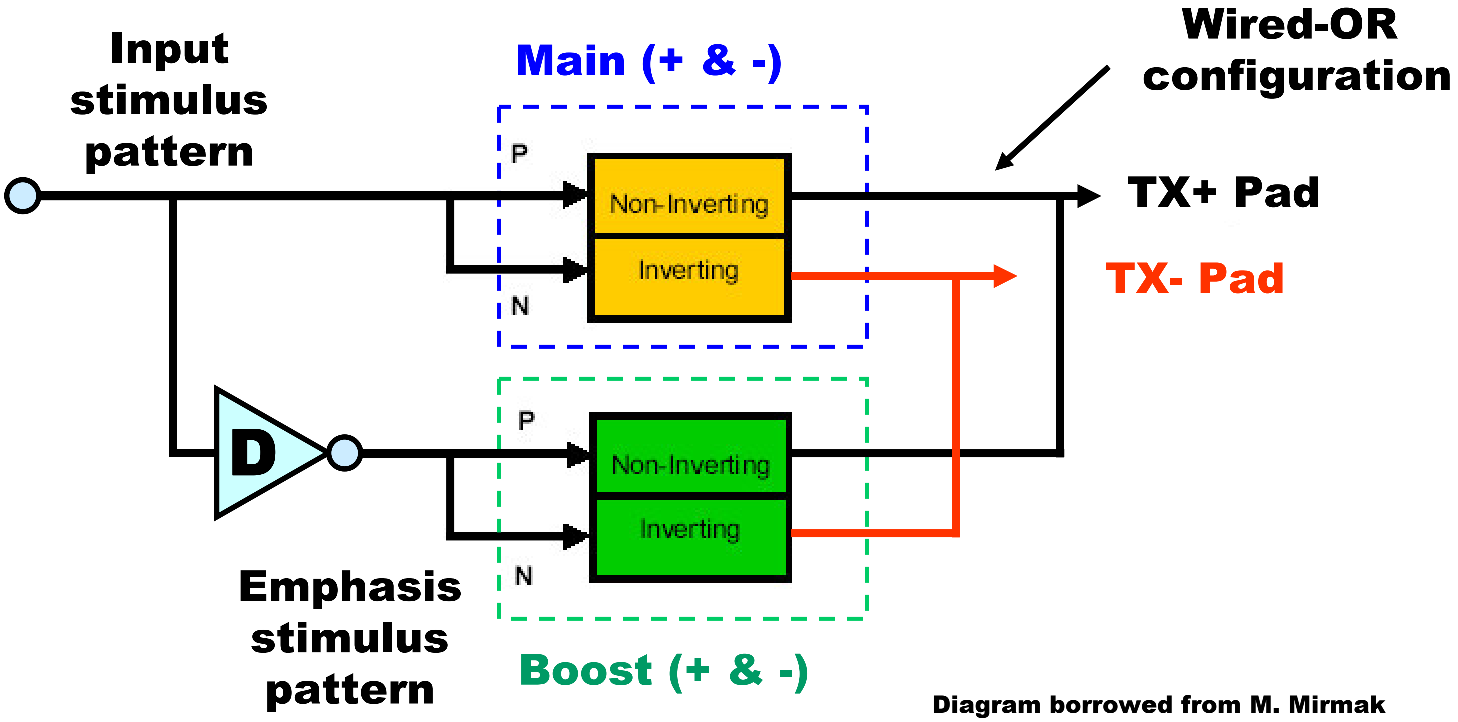


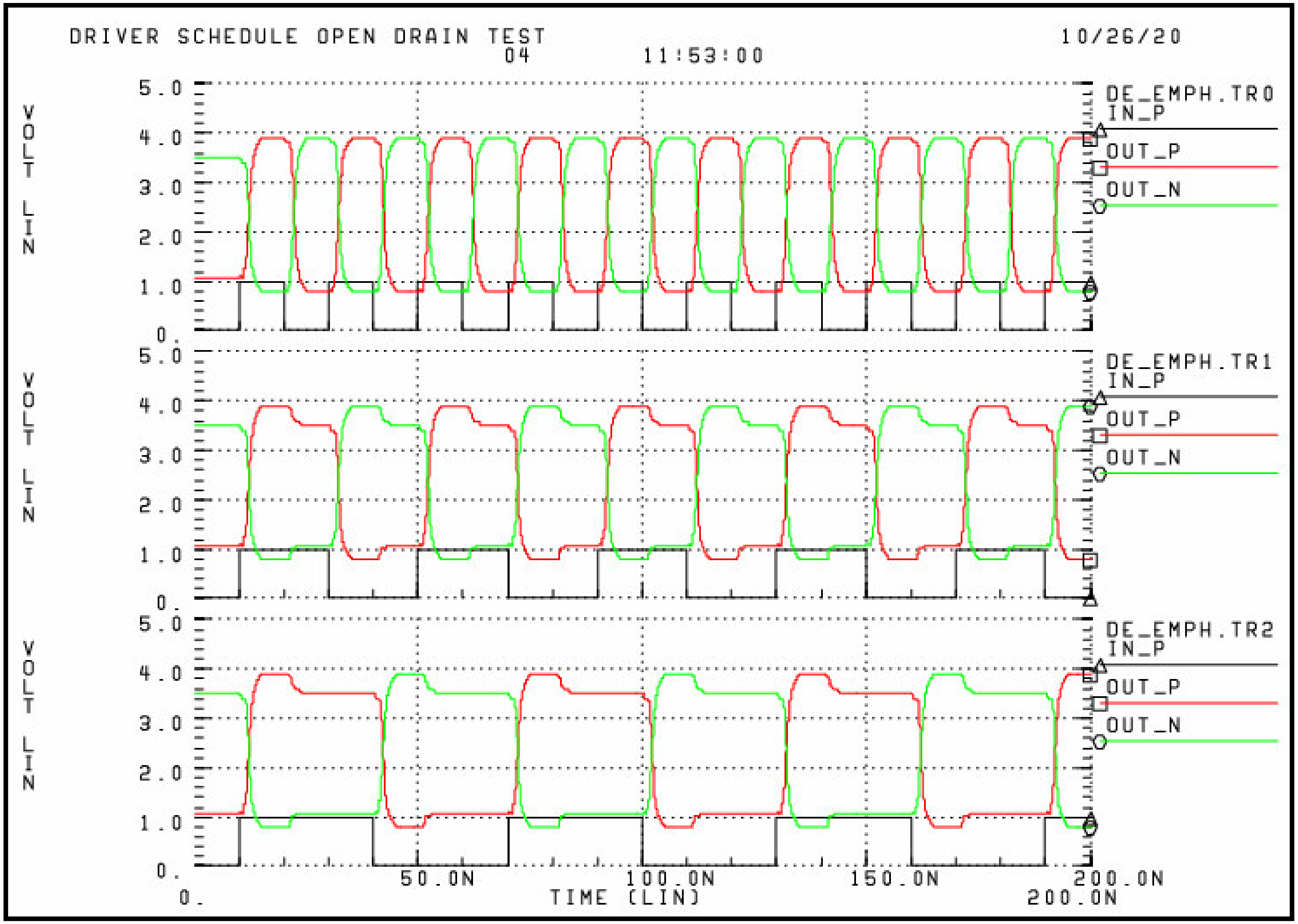
Diagram borrowed from M. Mirmak

The IBIS implementation

- **One way of implementing this in IBIS is to make two [Model]s for the Main and Boost buffers and tell the user of the model to wire-or them in the schematics of the simulation, and set up the appropriate stimulus**
  - To a novice user it may be difficult to set up the one bit delayed, inverted version of the input signal
  - This may be also be difficult in some tools which do not provide (easy) access to the stimulus signals
- **However, the [Driver Schedule] keyword provides a more user friendly solution without requiring any of these manual interventions:**

- **The Main buffer needs two zero ON delay parameters**
  - This is an equivalent of the normal, straight through operation of the Main buffer
- **The Boost buffer needs two OFF delay parameters with delay values equal to the width of a bit**
  - This is an equivalent of a one bit delayed, inverted operation of the Boost buffer

Simulation results with the [Driver Schedule] keyword



Stimulus: 0101010…(top), 0110011…(middle), 0111000111…(bottom)

Why didn’t we think about this before?

- **We did, but…**
- **There were discrepancies in the interpretation of the [Driver Schedule] keyword, because the specification was vague**
- **Now that the dust has settled with all the [Driver Schedule] BIRDs, we can finally count on predictable results between simulation tools**
- **Some tool vendors were slow implementing this keyword**
  - Perhaps the note in the IBIS specification about the temporary nature of this keyword made them feel that this was not an important feature(?)

Pros and cons

+ This technique provides a way to model gigabit pre/de-emphasis buffers with native (legacy) IBIS

- Fewer transistor level (SPICE) models will need to be released to customers
- Extends the life of legacy IBIS before requiring the IBIS v4.1 language extensions
- IBIS is a tool independent modeling standard
- IBIS simulations are faster than SPICE

+ Using [Driver Schedule] results in a more compact model

- Eliminates the need for connecting two separate [Model]s by hand in the schematics, one for the Main and one for the Boost portion of the buffer
- This is a major improvement over existing practices

− Some issues still remain with this technique

- Since legacy IBIS does not have provisions for clocked buffers, this model doesn’t have a clock input, consequently the delay parameter is “hard coded” and will need to be changed manually in the IBIS file for every clock frequency and simulation corner
- The [Driver Schedule] delay parameters do not have typ., min., max. corners
- Obtaining separate [Model] data for the Main and Boost buffers may still require the editing of the SPICE netlist
- There are a few questions around proper handling of C_comp

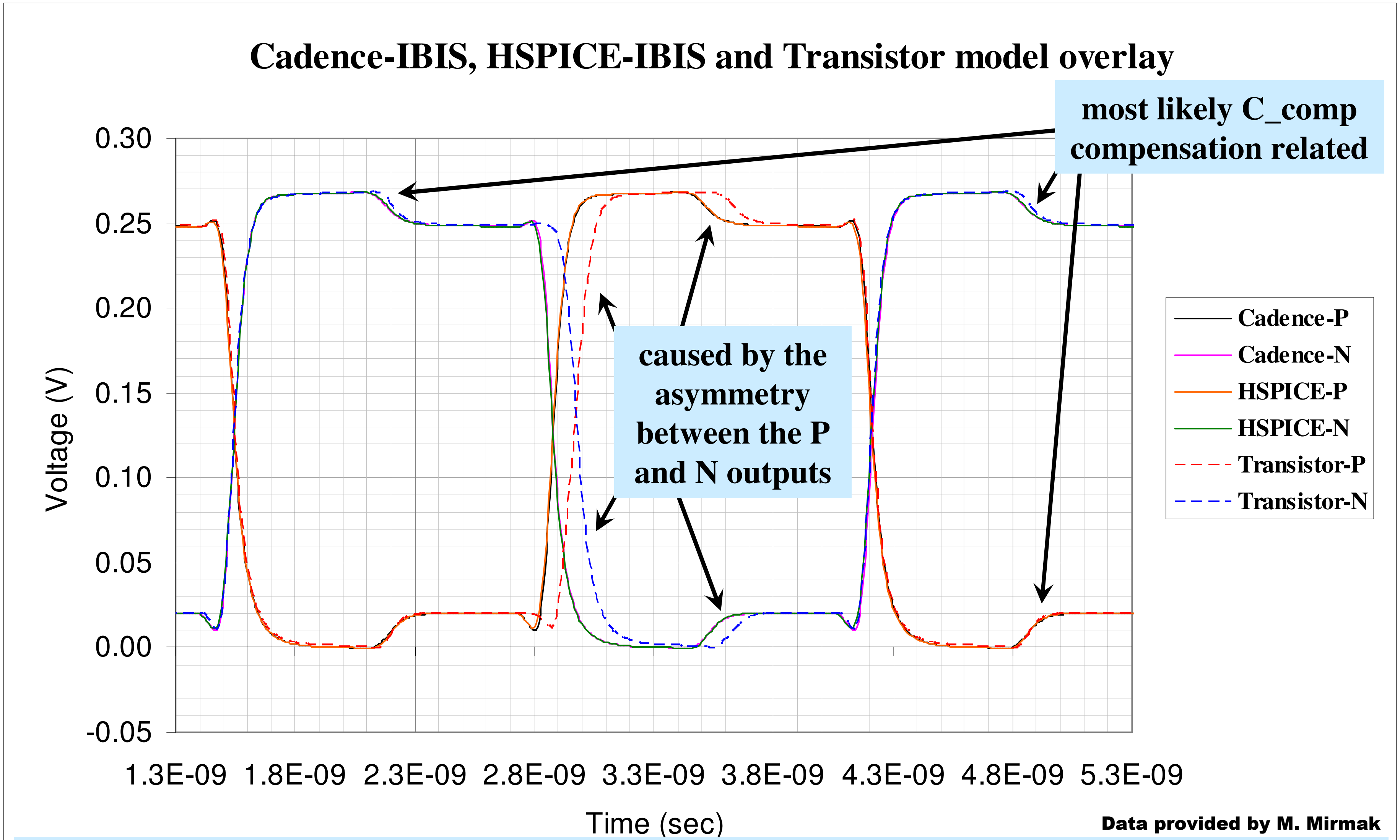
How to make Main and Boost buffers

- **Generate data for the Main buffer**
  - Disable the Boost buffer using control signals if available, or by editing the netlist
  - Do not disconnect the Boost buffer from the pad, its capacitive load must be there to obtain the correct waveforms for the Main buffer
- **Generate data for the Boost buffer**
  - Disable the Main buffer using control signals if available, or by editing the netlist
  - Do not disconnect the Main buffer from the pad, its capacitive load must be there to obtain the correct waveforms for the Boost buffer
- **Generate a "top level" [Model]**
  - Put the on-die resistors (and ODT if applicable) into the clamp I-V tables
  - Die capacitance (C_comp) and all other usual keywords and parameters go here
  - Add [Driver Schedule] to "instantiate" and "schedule" the Main and Boost [Model]s
    - Use two zeroes for the ON delay parameters for the Main buffer
    - Use two OFF delay parameters equal to the bit width time for the Boost buffer
- **Make three IBIS [Model]s for the above**
- **Be careful what you put in the "top level" and the "scheduled" [Model]s**
  - Scheduled [Model]s use: [Pulldown], [Pulldown Reference], [Pullup], [Pullup Reference], [Voltage Range], [Ramp], [Rising Waveform] and [Falling Waveform], as applicable
  - Everything else goes into the top level model
  - It is legal and some times useful to have more data in either one of the [Model]s, but they will be ignored by the simulator
  - There are some parser issues when certain data are missing (warning and error messages)

C_comp issues

- **The IBIS specification says that C_comp should be placed into the “top level” model and should represent the total buffer capacitance**
- **This is easy for the model maker, but tool vendors need to answer some difficult questions:**
  - How is the C_comp compensation done?
    - independently, inside the Main and Boost [Model]s?
    - collectively?
  - If independently, how is the capacitive loading effect of the neighboring model(s) accounted for in the compensation algorithm?
  - How is the total C_comp divided between the Main and Boost buffers?
  - Is the C_comp compensation correct for each transition?
    - strong to strong bit
    - strong to weak bit
    - weak to strong bit
- **More C_comp related information:**
  - http://www.eda.org/pub/ibis/summits/apr04/mirmak2.pdf
  - http://www.eda.org/pub/ibis/summits/oct04/mirmak2.pdf
- **A constant C_comp value may not be accurate enough at GHz speeds**
  - Frequency and/or voltage dependence may be important, which can only be modeled with the IBIS v4.1 language extensions

Correlation with a *symmetric* differential IBIS model



most likely C_comp compensation related

caused by the asymmetry between the P and N outputs

Data provided by M. Mirmak

The "asymmetry" problem can be fixed by making two different [Model]s for the P and N outputs

Correlation with an *asymmetric* differential IBIS model

- Due to lack of time this page will be completed after the IBIS Summit and uploaded to the website
- The goal is to show that the large miscorrelation can be eliminated using by making two non-identical [Model]s for the P and the N outputs

Summary

- **The clarification BIRDs made the implementation of [Driver Schedule] more predictable and consistent between simulators**
- **The [Driver Schedule] keyword is useful to describe the timing relationships between the building blocks of Pre/De-emphasis buffers**
- **The usage of Pre/De-emphasis buffers made with the [Driver Schedule] keyword is easier**
- **This extends the life of legacy IBIS**
- **Some issues and limitations still exist**
- **The issues and limitations will most likely be solved by writing models with the IBIS v4.1 language extensions**

For information on the aspects of differential buffer modeling please refer to the presentation at
http://www.eda.org/pub/ibis/summits/oct03/muranyi.pdf