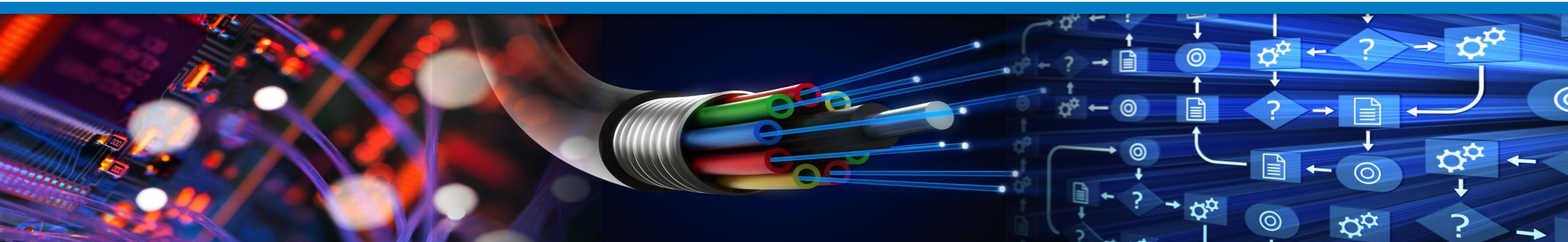
Initial Delay Issues in Analog IBIS Buffers

what it is – how it affects – how to address – and more

Michael Schäder • Mariusz Fąferko • Amir Wallrabenstein
EMC Technology Center
Paderborn Germany

European IBIS Summit
May 11, 2016
Torino Italy



Outline

- Introduction
- Initial delay in analog IBIS buffer models
- Switching characterization of analog IBIS buffers
- Manual trimming of V-T curves
- Automatic initial delay removal
- Automatic k-table trimming approach
- IBIS Standard 6.1, BIRD 168.1 and BIRD177 [Initial Delay]
- Summary

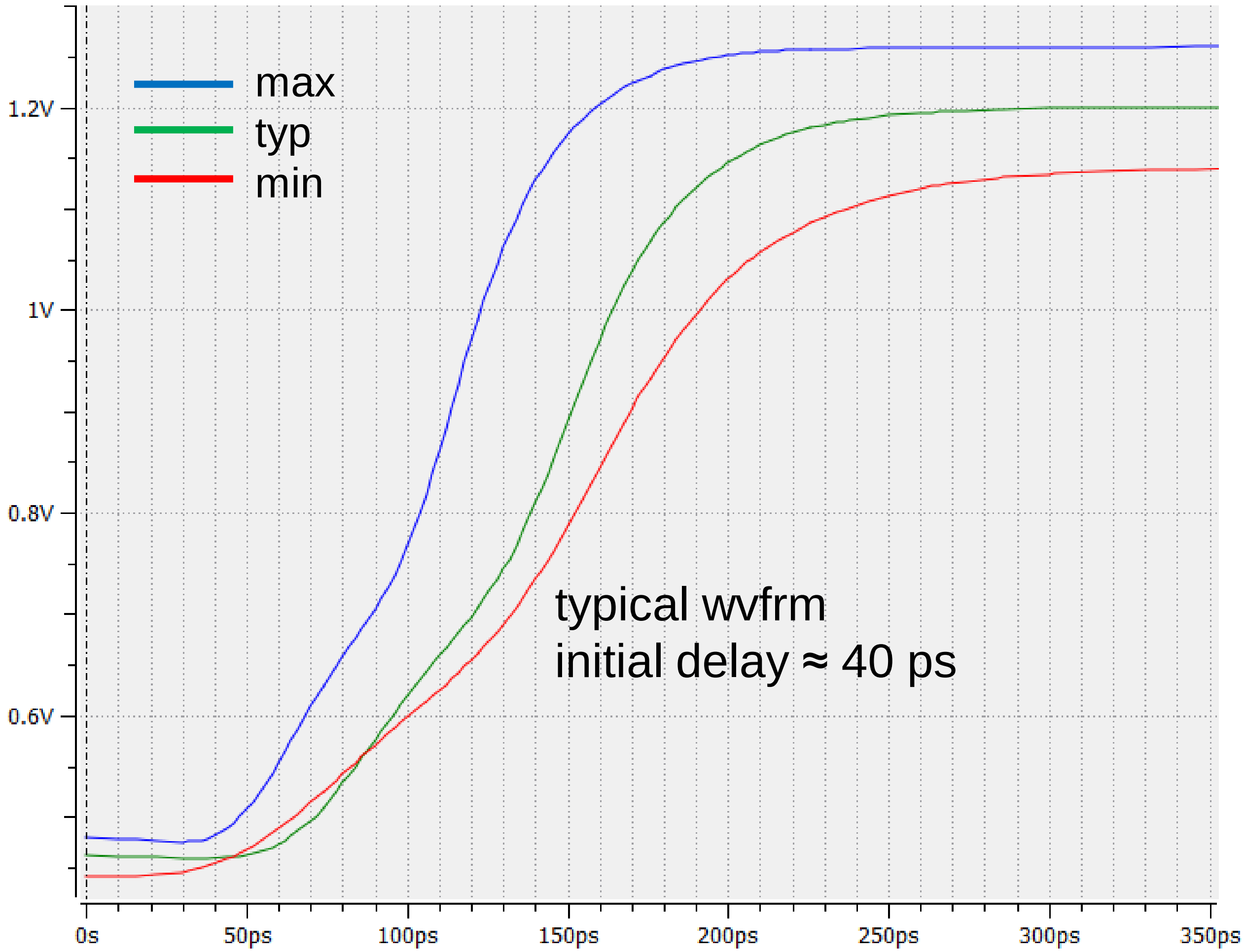
© Zuken

Introduction

- Initial delay and overclocking have been discussed for many years
  - Arpad Muranyi – IBIS Summit at DAC 2003, Anaheim
    How to model the switching into an unfinished edge problem
  - Lance Wang – IBIS Summit at DesignCon 2007, Santa Clara and at DAC 2007, San Diego
    Initial Time Delay Issue in IBIS VT Curves
    More on Initial Delay Issues
  - Yingxin Sun and Raymond Y. Chen – IBIS Summit Taipei 2013
    An Advanced Behavioral Buffer Model With Over-Clocking Solution
  - Xuefeng Chen – IBIS Summit Shanghai 2014
    An Effective Solution to Simulate Composite Current When Over-clocking
- Still quite relevant today
  - Almost all buffers include some extra delay in their time based characteristics
    [Rising Waveform]s, [Falling Waveform]s, and [Composite Current]s
  - Initial delay reduces possible buffer speed
  - Initial delay is often the reason for overclocking in simulation

Initial delay in analog IBIS buffer models

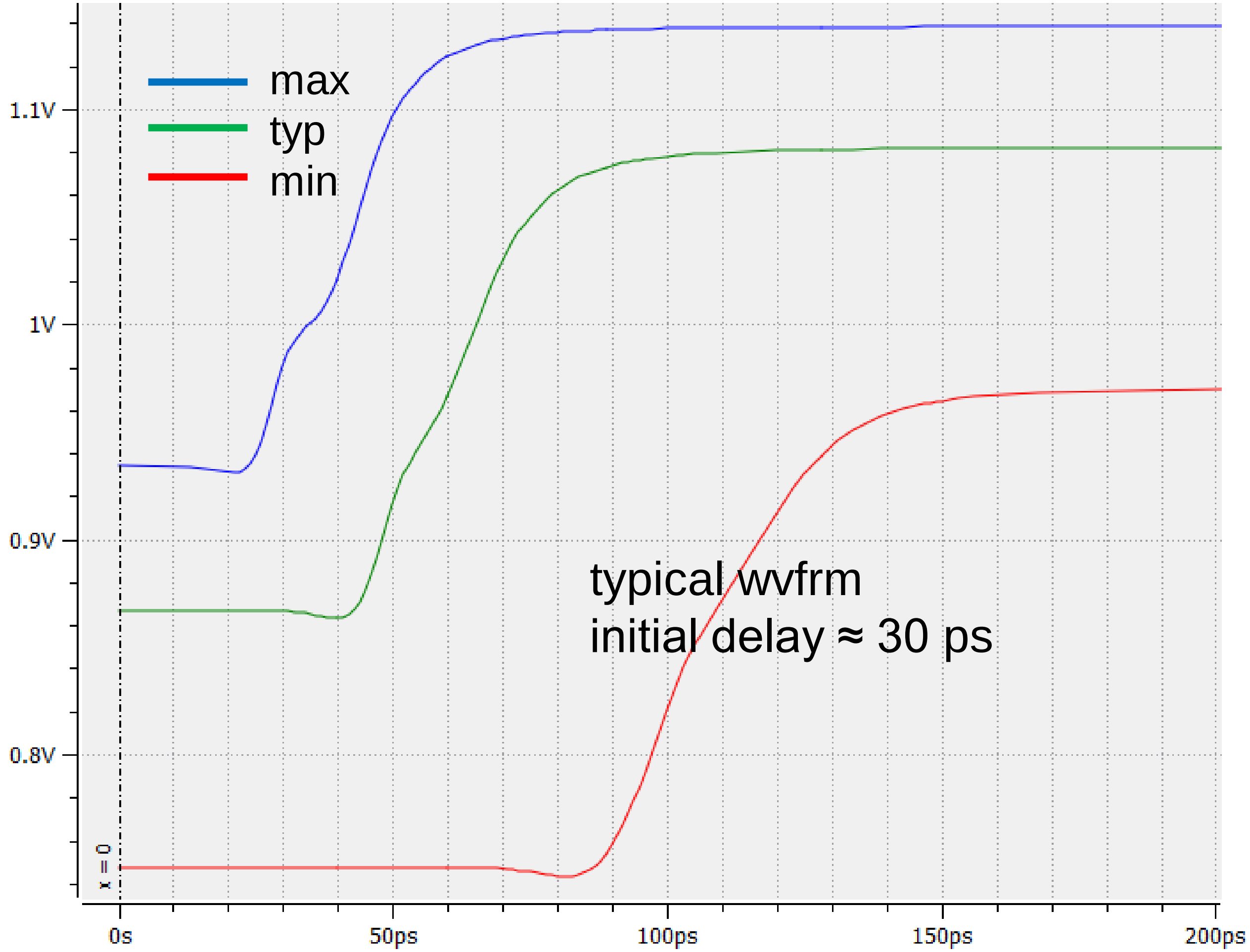
- Almost all models show initial delay in their time based characteristics (1)



typical wvfrm
initial delay ≈ 40 ps

Initial delay in analog IBIS buffer models

- Almost all models show initial delay in their time based characteristics (2)

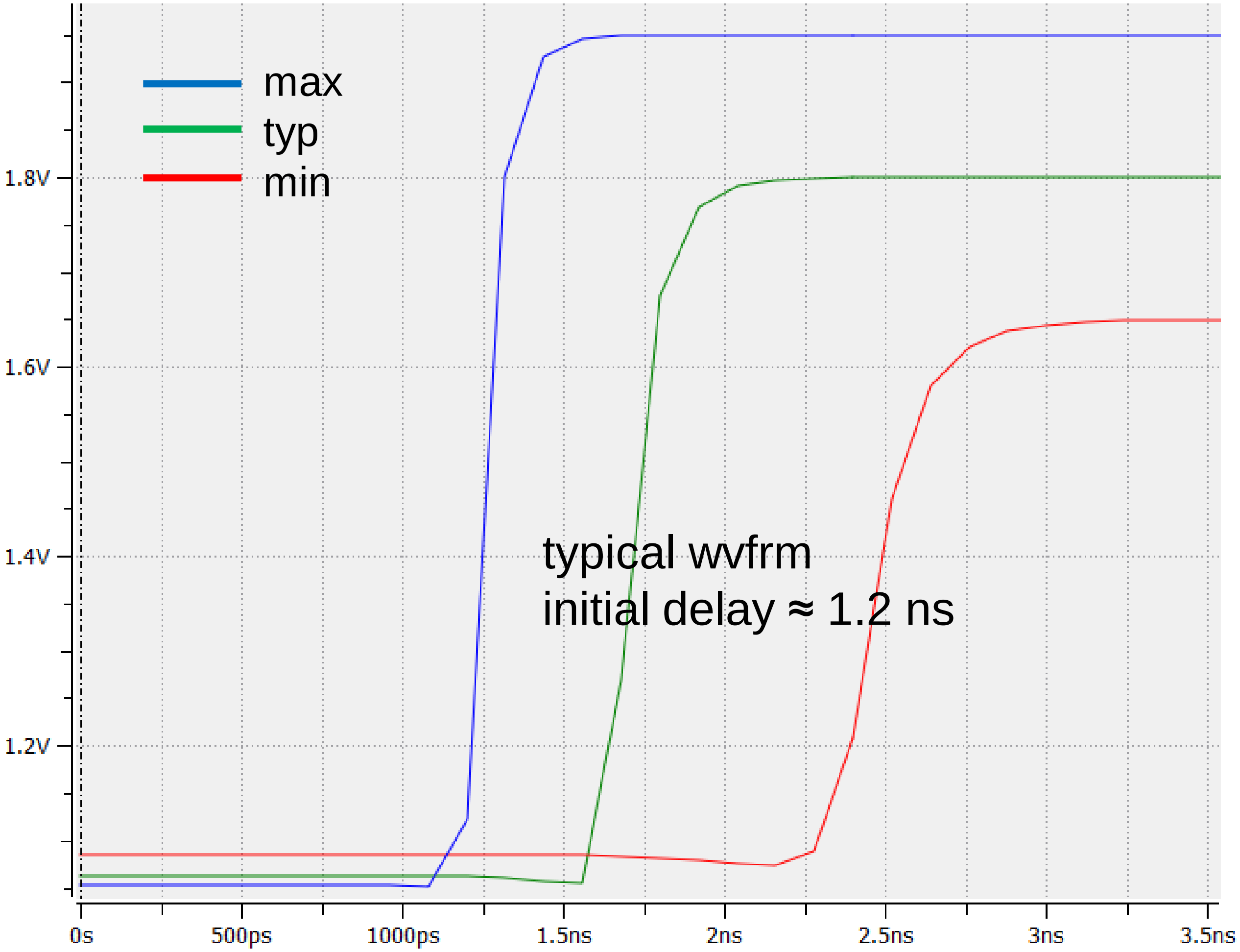


typical wvfrm
initial delay ≈ 30 ps

© Zuken

Initial delay in analog IBIS buffer models

- Almost all models show initial delay in their time based characteristics (3)



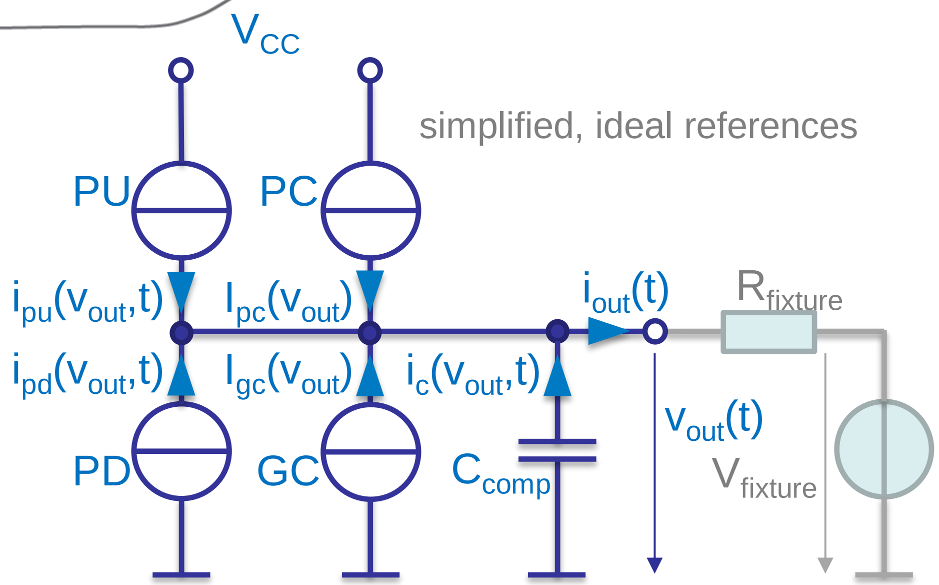
typical wvfrm
initial delay ≈ 1.2 ns

© Zuken

Switching characterization of analog IBIS buffers

Kirchhoff’s Current Law (KCL) for the output node

$i_{out}(t) = i_{PU}(v_{out}, t) + i_{PD}(v_{out}, t) + I_{PC}(v_{out}) + I_{GC}(v_{out}) + i_C(v_{out}, t)$



With time dependent weighting factors $k_U(t)$, $k_D(t)$ (general IBIS output equation)

$i_{out}(t) = k_U(t) \cdot i_{PU}(v_{out}) + k_D(t) \cdot i_{PD}(v_{out}) + I_{PC}(v_{out}) + I_{GC}(v_{out}) + i_C(v_{out}, t)$

Rising edge: start values $k_U(t) = 0; k_D(t) = 1$ 1 EQ / 2 UK

final values $k_U(t) = 1; k_D(t) = 0$

One [Rising Waveform] and one [Falling Waveform]

$k_D(t) = 1 - k_U(t);$ $k_U(t) = 1 - k_D(t);$ ideally complementary PU / PD transition

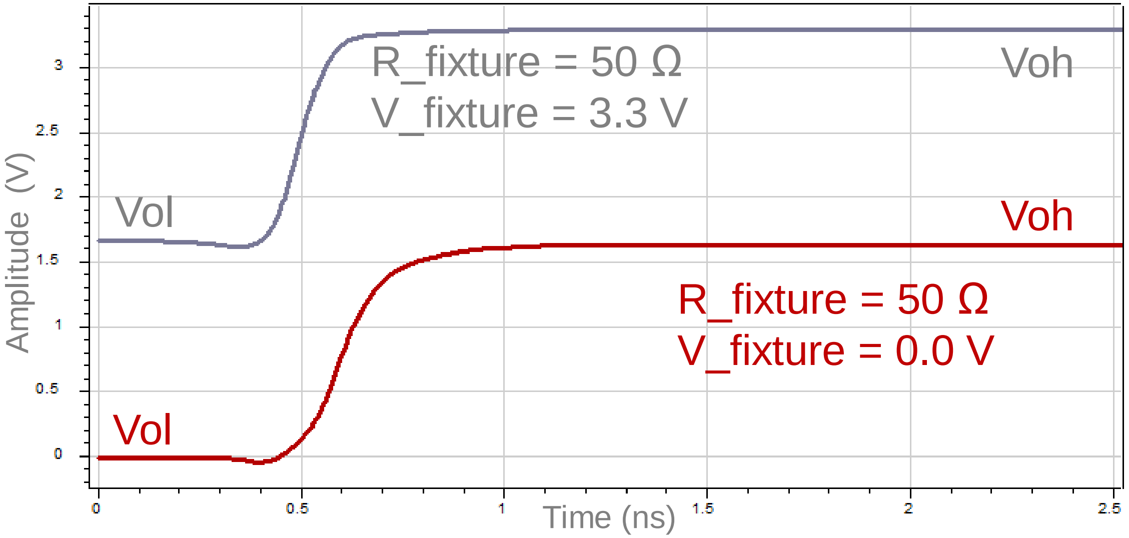
Two [Rising Waveform]s and two [Falling Waveform]s

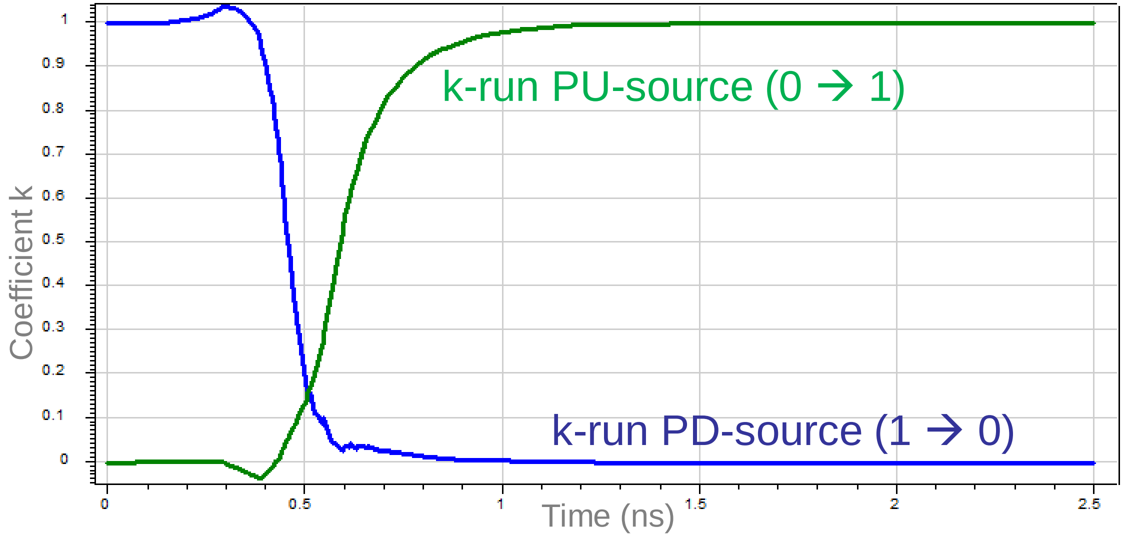
$i_{out1}(t) = k_U(t) \cdot i_{PU}(v_{out1}) + k_D(t) \cdot i_{PD}(v_{out1}) + I_{PC}(v_{out1}) + I_{GC}(v_{out1}) + i_C(v_{out1}, t)$ 2 EQ / 2 UK

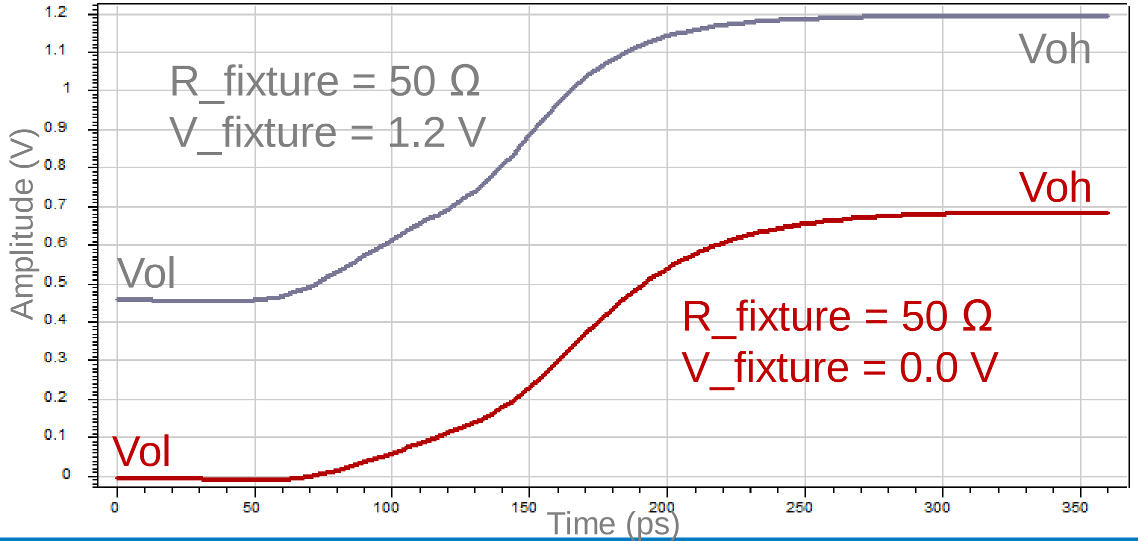
$i_{out2}(t) = k_U(t) \cdot i_{PU}(v_{out2}) + k_D(t) \cdot i_{PD}(v_{out2}) + I_{PC}(v_{out2}) + I_{GC}(v_{out2}) + i_C(v_{out2}, t)$

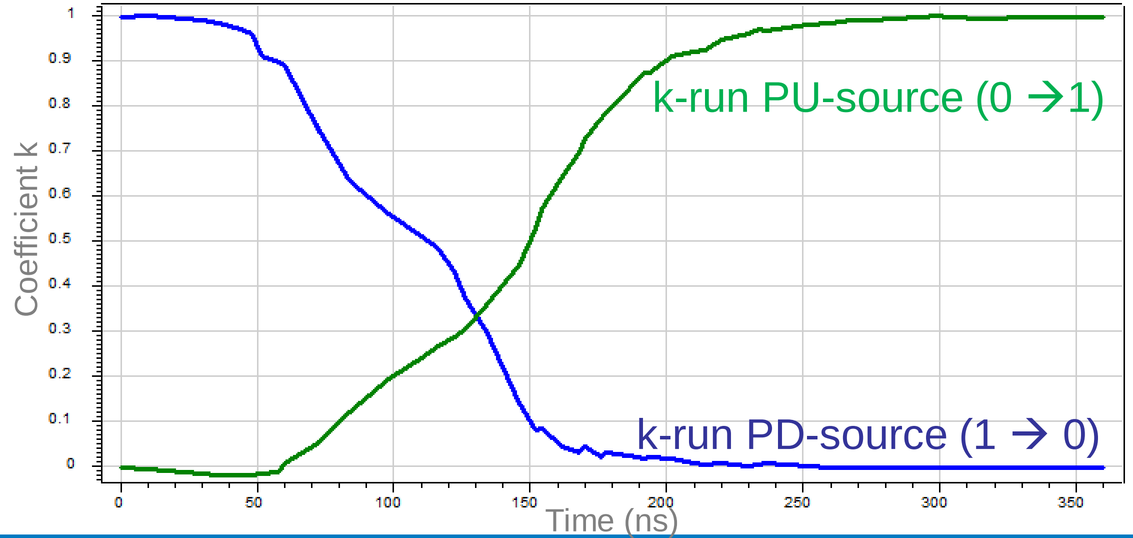
Switching characterization of analog IBIS buffers cont’d

- [Rising Waveform]s and resulting PU and PD switching coefficients (k-tables)









© Zuken

Switching characterization of analog IBIS buffers cont’d

- Initial delay in [Rising Waveform] and [Falling Waveform] appears in k-tables too
  - k-table initial delay is very similar but not necessarily equal to initial waveform delay
- k-tables perform transition from *static* level to *static* level
  - That is the intrinsic switching information in an IBIS model; nothing less, nothing more
- Overclocking happens when an opposite transition is initiated unless the *static* levels are reached (PU / PD fully on / off resp.)
  - Switching into an unfinished edge
  - This can be detected at simulation time
- Any initial delay removal impacts the buffer behavior
  - Rising/falling edge shape
  - DC-levels
  - Duty cycle (if waveform time correlation is broken)

Visual / manual trimming of V/t curves

- t_{tr} = 0.7 ns → t_{pulse} = 1.4 ns → ≈ 700 MHz

  initial delay ≈ 1.0 ns

  (t_{tr} = 1.7 ns → t_{pulse} = 3.4 ns → ≈ 350 MHz)

- t_{tr} = 1.2 ns → t_{pulse} = 2.4 ns → ≈ 400 MHz

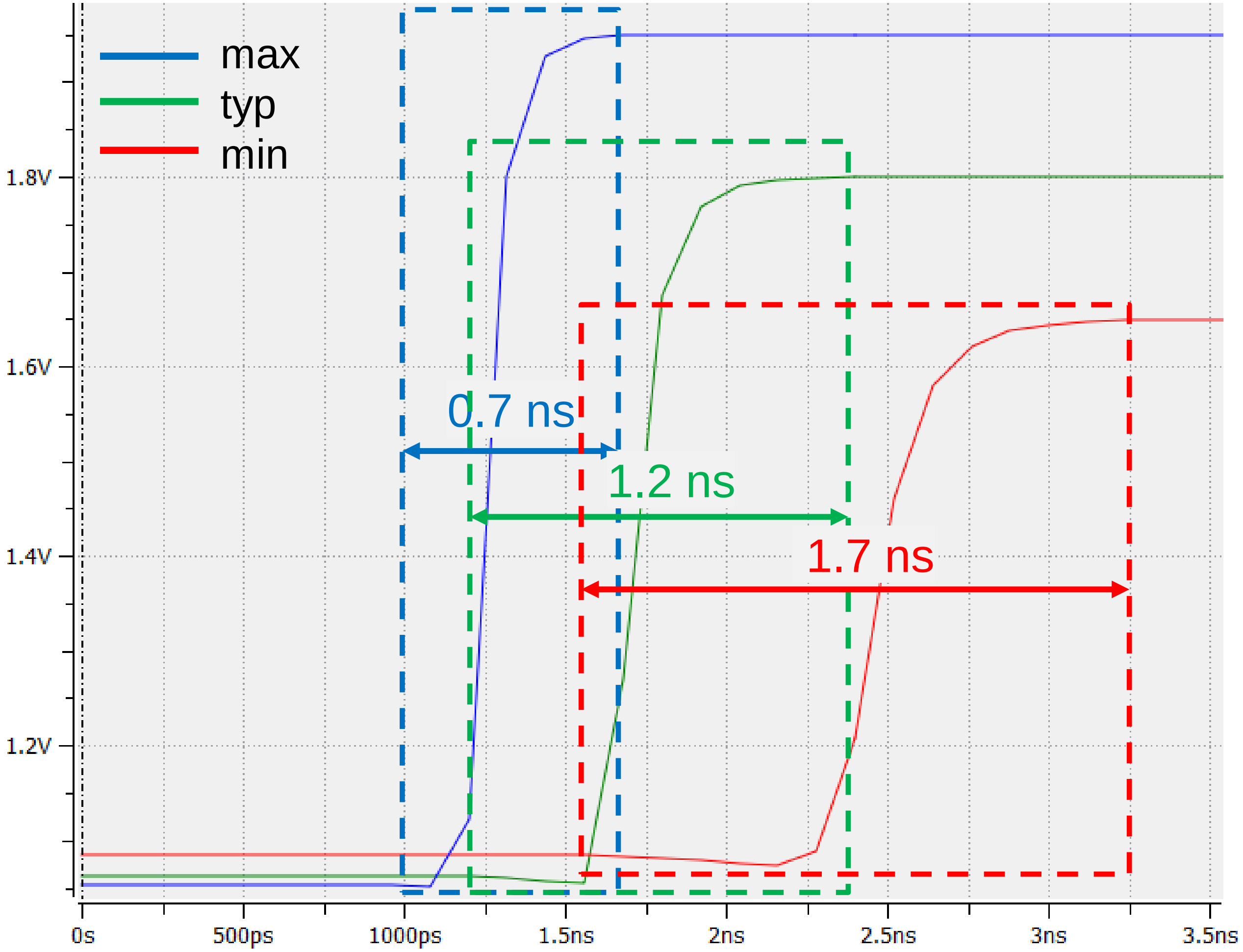
  initial delay ≈ 1.3 ns

  (t_{tr} = 2.5 ns → t_{pulse} = 5.0 ns → ≈ 200 MHz)

- t_{tr} = 1.7 ns → t_{pulse} = 3.4 ns → ≈ 290 MHz

  initial delay ≈ 1.6 ns

  (t_{tr} = 3.3 ns → t_{pulse} = 6.6 ns → ≈ 150 MHz)



assumption: symmetrical switching, no [Composite Current]

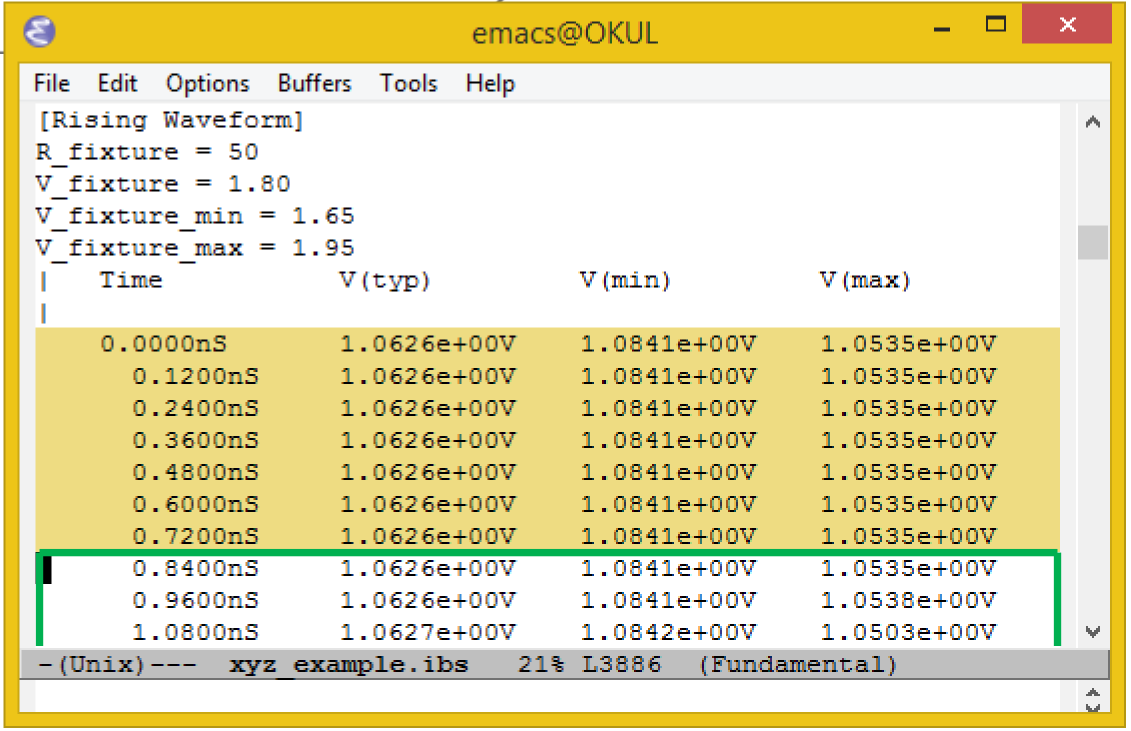
Initial delay halve possible buffer speed

© Zuken

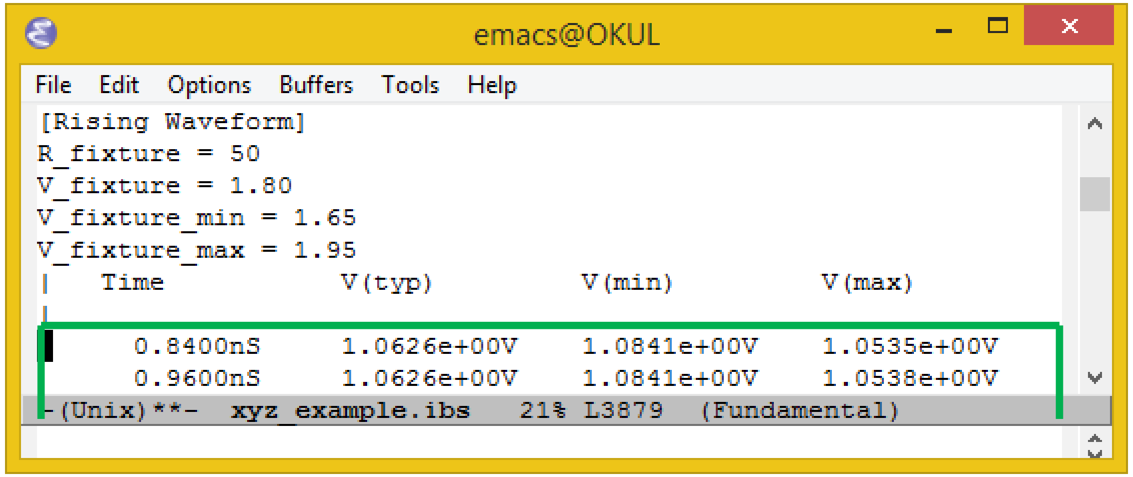
Manual trimming of V/t curves cont’d

- V-T (and I-T) curves are correlated in time
  - according to the IBIS Standard
- Associated [Rising/Falling Waveform]s have to be trimmed equally
  - It might be required to trim all waveforms (typ / min / max) of a model equally
  - Special care is required if [Composite Current] curves are present
- Depending on the target simulator time values need not to be shifted
  - Initial delay removal for all three rows can be done in one chunk in an ASCII-editor easily

Simple with visual inspection and little attention



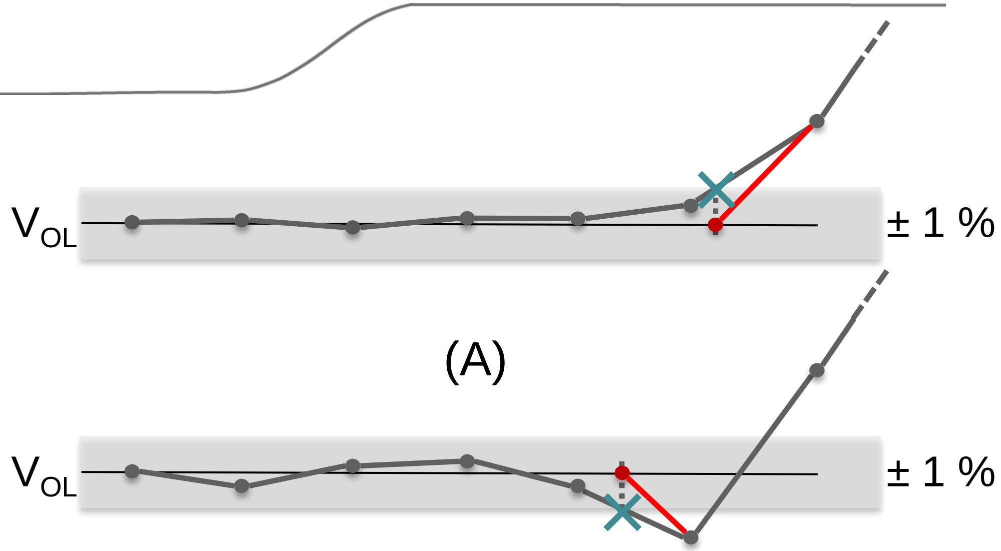
```
[Rising Waveform]
R_fixture = 50
V_fixture = 1.80
V_fixture_min = 1.65
V_fixture_max = 1.95
|   Time               V(typ)         V(min)         V(max)
|
    0.0000nS           1.0626e+00V    1.0841e+00V    1.0535e+00V
      0.1200nS         1.0626e+00V    1.0841e+00V    1.0535e+00V
      0.2400nS         1.0626e+00V    1.0841e+00V    1.0535e+00V
      0.3600nS         1.0626e+00V    1.0841e+00V    1.0535e+00V
      0.4800nS         1.0626e+00V    1.0841e+00V    1.0535e+00V
      0.6000nS         1.0626e+00V    1.0841e+00V    1.0535e+00V
      0.7200nS         1.0626e+00V    1.0841e+00V    1.0535e+00V
      0.8400nS         1.0626e+00V    1.0841e+00V    1.0535e+00V
      0.9600nS         1.0626e+00V    1.0841e+00V    1.0538e+00V
      1.0800nS         1.0627e+00V    1.0842e+00V    1.0503e+00V
-(Unix)---  xyz_example.ibs   21% L3886  (Fundamental)
```

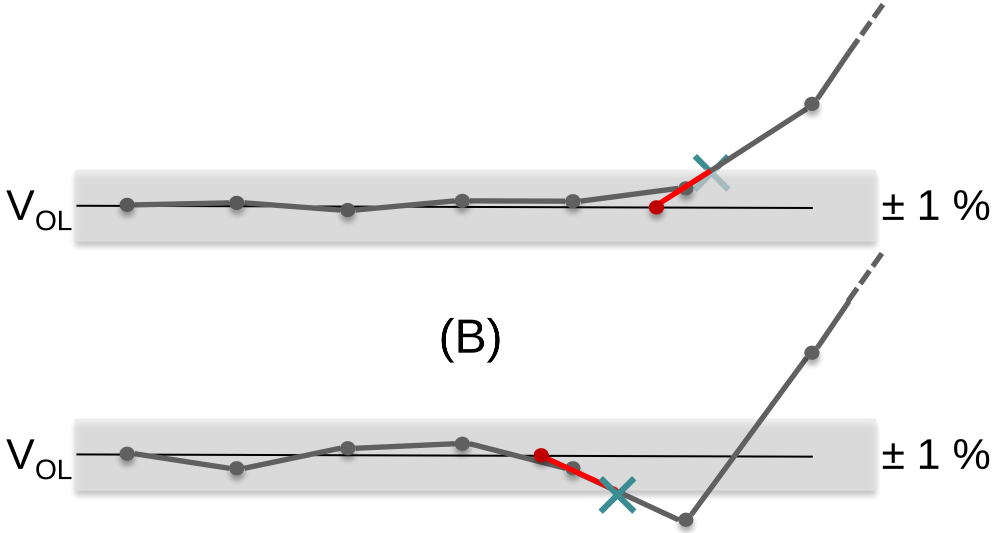


```
[Rising Waveform]
R_fixture = 50
V_fixture = 1.80
V_fixture_min = 1.65
V_fixture_max = 1.95
|   Time               V(typ)         V(min)         V(max)
|
      0.8400nS         1.0626e+00V    1.0841e+00V    1.0535e+00V
      0.9600nS         1.0626e+00V    1.0841e+00V    1.0538e+00V
-(Unix)**-  xyz_example.ibs   21% L3879  (Fundamental)
```

Automatic waveform trimming approaches (A) / (B)

- Define tolerance range of e.g. ± 1 % of total swing
  - $V_{oh} – V_{ol}$, or difference of end / start voltages
  - Individual tolerances per waveform
  - Noise / rippling may require increasing tolerance
- Find intersection of tolerance range with waveform
  - Start from the beginning and find first intersection, or
  - go backwards from e.g. 0.5 total swing to last intersection
- Restore correct DC-level
  - Set voltage at intersection to DC-level (A)
  - Extrapolate backward / forward to DC-level (B)
- Do this for all relevant waveforms and keep individual initial delays per waveform
- Determine the minimum from the individual initial delays
- Cut-off the min. initial delay from all relevant waveforms



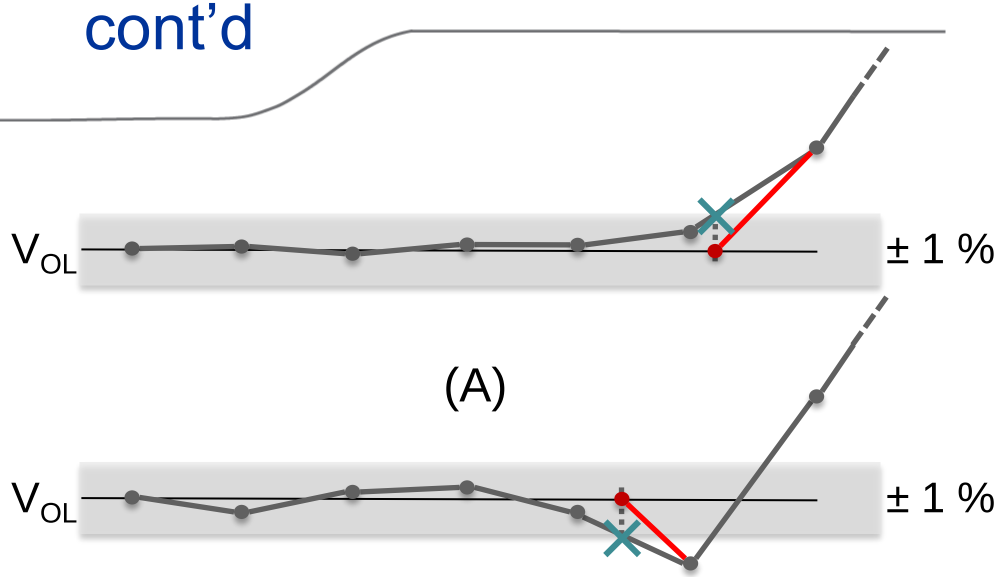


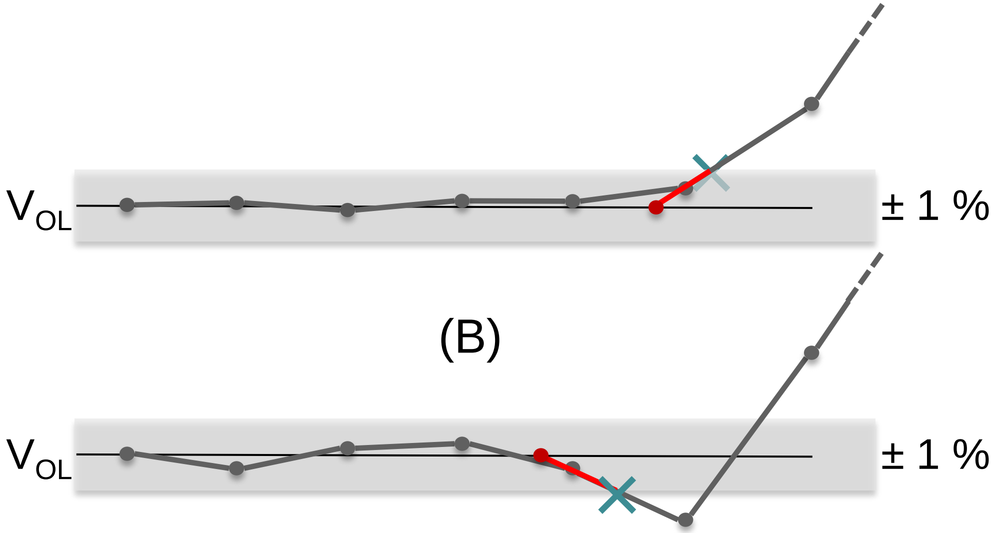
© Zuken

Automatic waveform trimming approaches (A) / (B) cont'd

- Remarks to approach (A)
  - Simple and inexpensive
  - (More) noticeable impact on rising / falling edge shape
  - Impact increases with higher tolerances
  - Effective initial delay detection / removal
  - Best applied on waveforms with negligible noise / rippling
- Remarks to approach B
  - Less simple and little more expensive
  - Reduced impact on rising / falling edge shape
  - Impact increases with higher tolerances
  - Less effective initial delay detection / removal with small tolerances

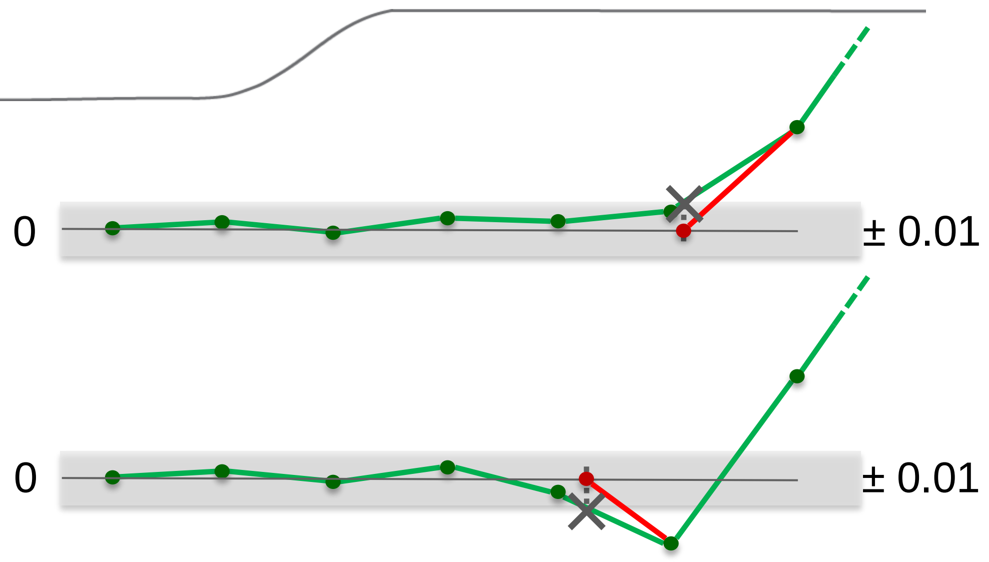
Both approaches are sensitive to waveform specific tolerances





© Zuken

Automatic k-table trimming approach



- Similar to waveform trimming approach (A) before
  - Define tolerance range of e.g. ± 0.01
    - Equals ± 1 % of static k-levels
  - Tolerance is waveform independent
  - Noise / rippling may require increasing tolerance

- Remarks
  - Simple and inexpensive
  - Works on effective switching directly (k-tables)
  - Works well in many known cases with 1 % tolerance
    - Negligible impact on rising / falling edge shape
  - Effective initial delay detection / removal

© Zuken

Initial delay / overclocking in simulation

- Initial delay removal should be reported to the user
  - Example message in simulation window, log-file, etc.

```
Initial delay: model “xyz_example_typ_out” – 1.098e-009 s removed
```

- Overclocking detection
  - Example message in simulation window, log-file, etc.

```
Overclocking: model “xyz_example_typ_out” – @ 1.250e-09 s, ratio 1.699
```

IBIS Standard 6.1, BIRD168.1 and 177 [Initial Delay]

- BIRD 168.1 – *Handling of Overclocking Caused by Delay in Waveform Tables*
  - Some EDA vendors allow user-specified truncation of initial delay in the V-t tables and/or in the Composite Current tables. Other EDA vendors may use arbitrary “windowing” techniques while some may not allow any similar actions.
    This leads to inconsistent simulation results across different EDA tools.
    The authors of this BIRD believe that only IBIS model developers have the appropriate knowledge to specify the exact amount of the initial delay that can and in fact should be properly removed from the waveform tables. […]
    This BIRD does not address true overclocking where the user attempts simulation of a buffer at switching rates exceeding hardware capabilities.
- BIRD 177 – *[Initial Delay] keyword for Submodels and Driver Schedules*
  - BIRD 168.1 introduced the [Initial Delay] keyword. Its scope is defined within the [Model] keyword. However, there is a question whether there is any inheritance within multi-stage drivers or for submodels. This BIRD clarifies this issue.

© Zuken

IBIS Standard 6.1 [Initial Delay]

- Keyword: [Initial Delay]
  - Allowed under [Model] and [Add Submodel] when V-T or I-T tables are present

```
[Initial Delay] | This keyword provides information on removable delay(s)
| time table     typy        min        max
V-T            0.20e-9   0.22e-9   0.18e-9
I-T            0.05e-9      NA         NA
```

- Allows model-makers and (experienced) users to treat initial delay
  - ibischk6 checks waveforms vs. I-V tables

```
WARNING - Model xyz_example_out: The [Rising Waveform] with [R_fixture]=50 Ohms
      and [V_fixture_max]=1.155V has MAX column DC endpoints of  0.60V and
      1.12v, but an equivalent load applied to the model's I-V tables yields
      different voltages ( 0.61V and  1.12V), a difference of  2.74% and 0.09%,
      respectively.
```

  - ibschk6 should test DC endpoints at [Initial Delay] time → Parser enhancement or bug

© Zuken

Summary

- Initial delay is common reason for overclocking of buffers in simulation
  - Review models before simulation with respect to intended signal speed
  - Ask for model update
  - Consider countermeasure
- Manual waveform trimming is often possible
- EDA tools may provide automatic initial delay removal
  - Check your tool for its capabilities
  - Make tests to verify the approach(es)
- IBIS 6.1 has [Initial Delay] keyword for [Model] and [Add Submodel]
  - Check that your tool already supports IBIS 6.1
  - Ask your model maker to use it
  - Use with care to treat initial delay / overclocking
    - Less error prone than manual IBIS-file modification
    - Transparent, consistent, and tool independent initial delay treatment

© Zuken