An algorithm to model over-clocking more accurately

IBIS Summit at DesignConEast 2003

Royal Plaza Hotel, Marlborough, MA

June 23, 2003

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

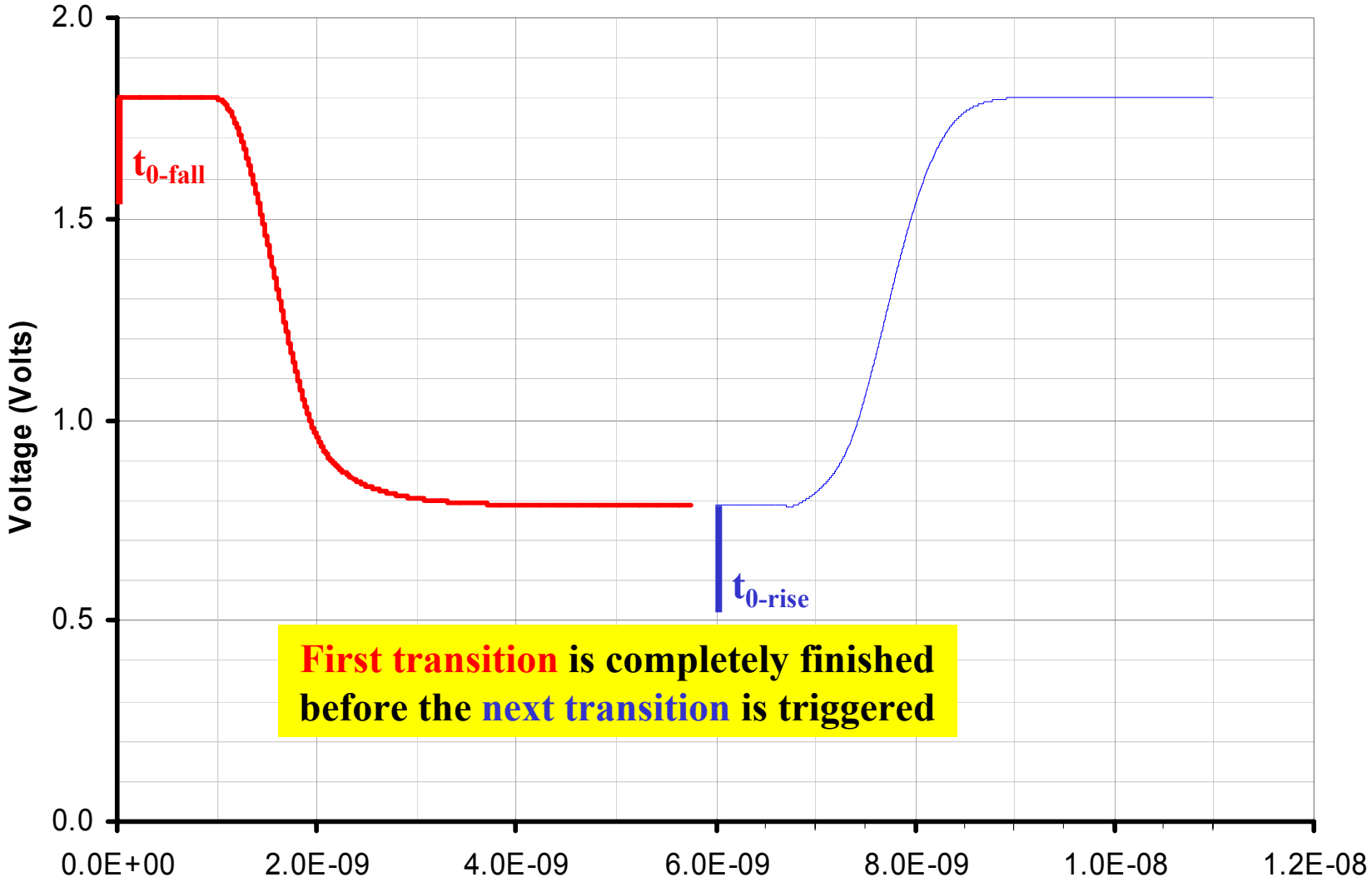
intel

CPD

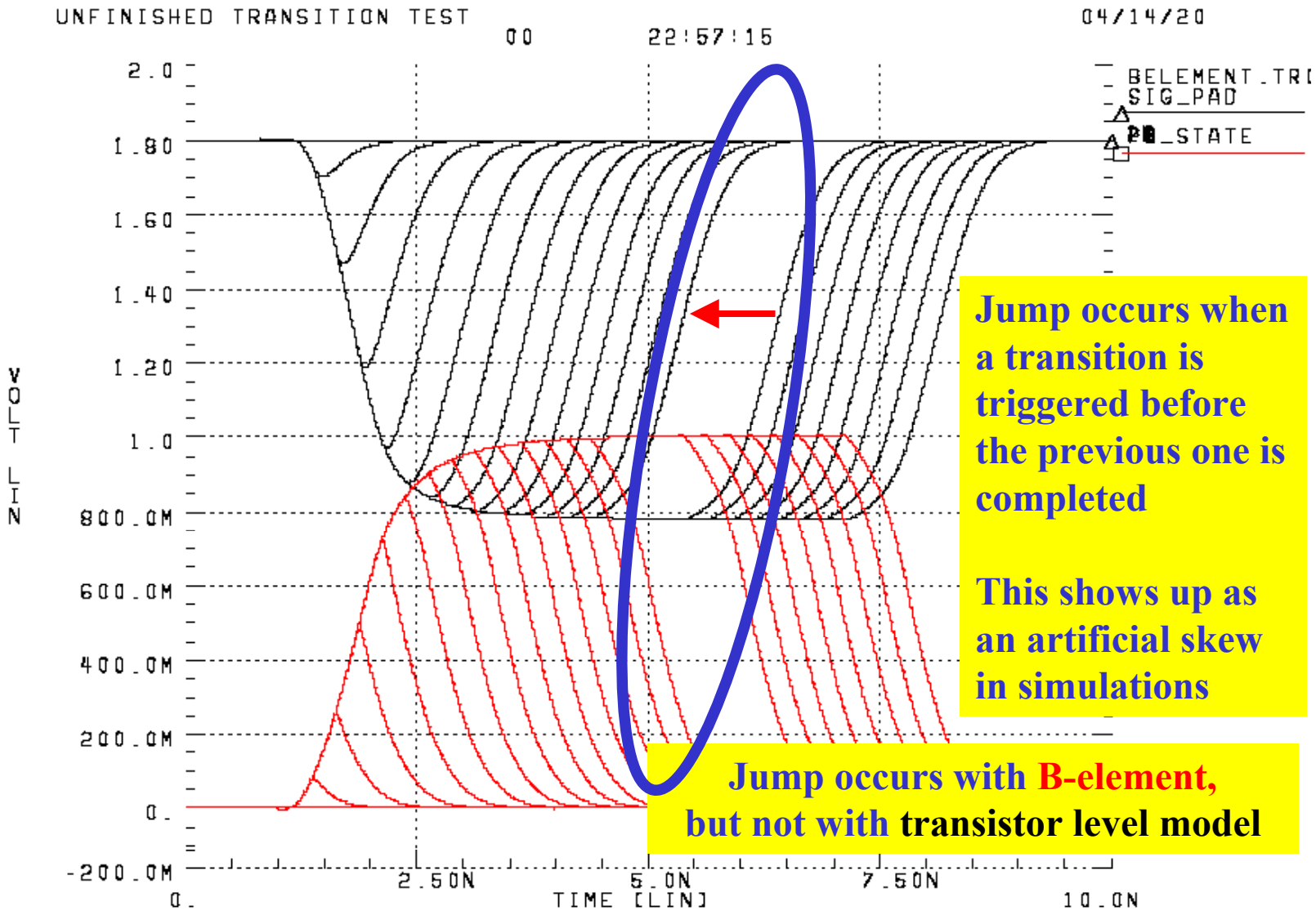
Outline

- Problem statement
- A possible algorithm idea (not good)
- A better algorithm idea
- SPICE waveforms to prove second idea

Normal operation



HSPICE's B-element in version 99.4



Jump occurs when a transition is triggered before the previous one is completed

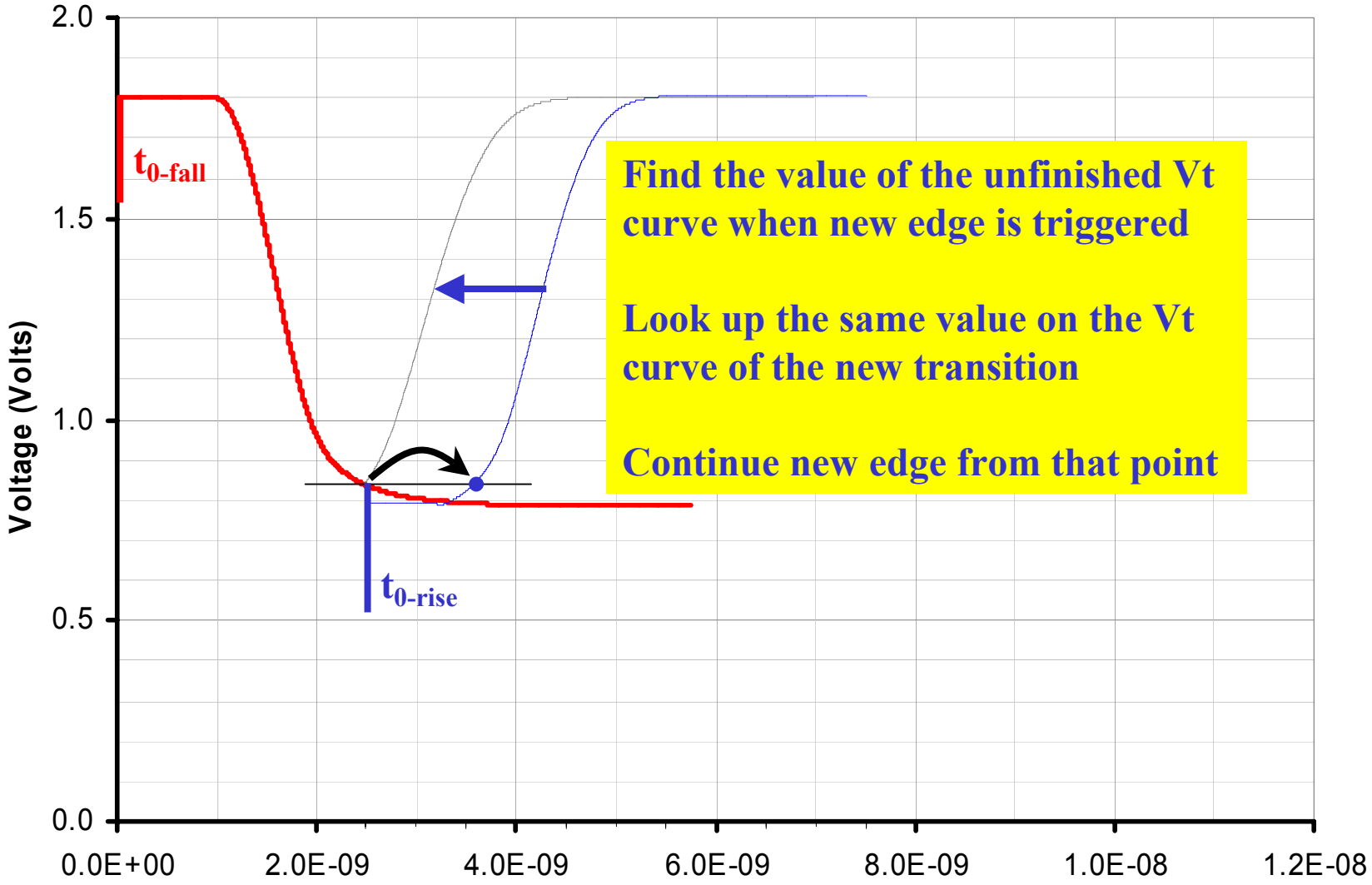
This shows up as an artificial skew in simulations

Jump occurs with B-element, but not with transistor level model

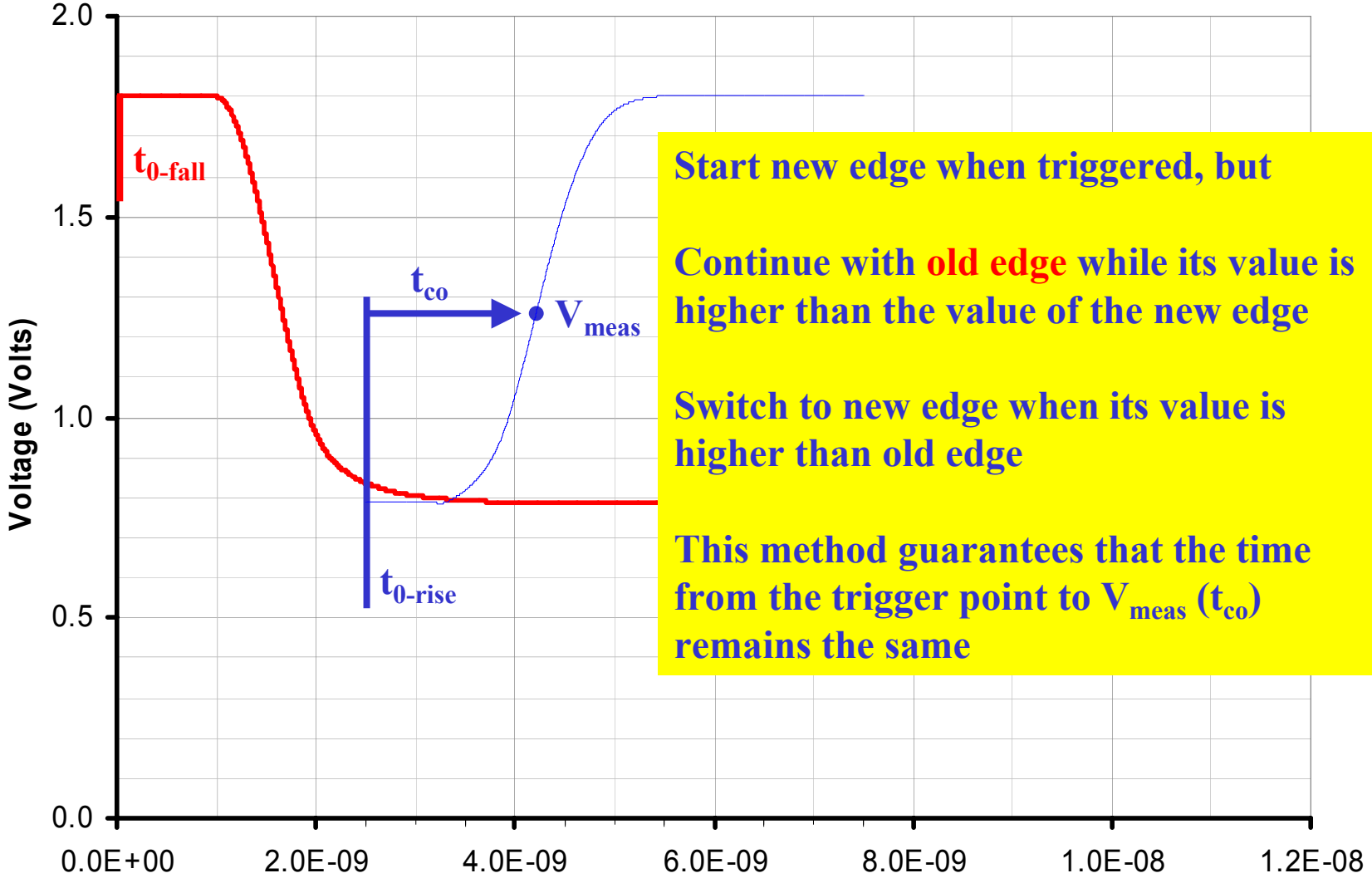
Disclaimer

- **Even though this problem was first discovered with HSPICE's B-element, this is not only an issue specific to HSPICE.**
- **The fundamental problem is that the IBIS specification was written with the assumption that a transition is always completed before a new transition begins.**
- **Since the IBIS specification does not address over-clocking, it is up to the tool vendor to deal with the situation when it arises.**
- **Each tool may have a different solution, some may seem to be correct (because they are not noticeable), but no one can really claim that they can reproduce the SPICE model's waveforms exactly when over-clocking occurs.**

Conceptual explanation of the algorithm



Arpad's first algorithm idea (don't implement it)



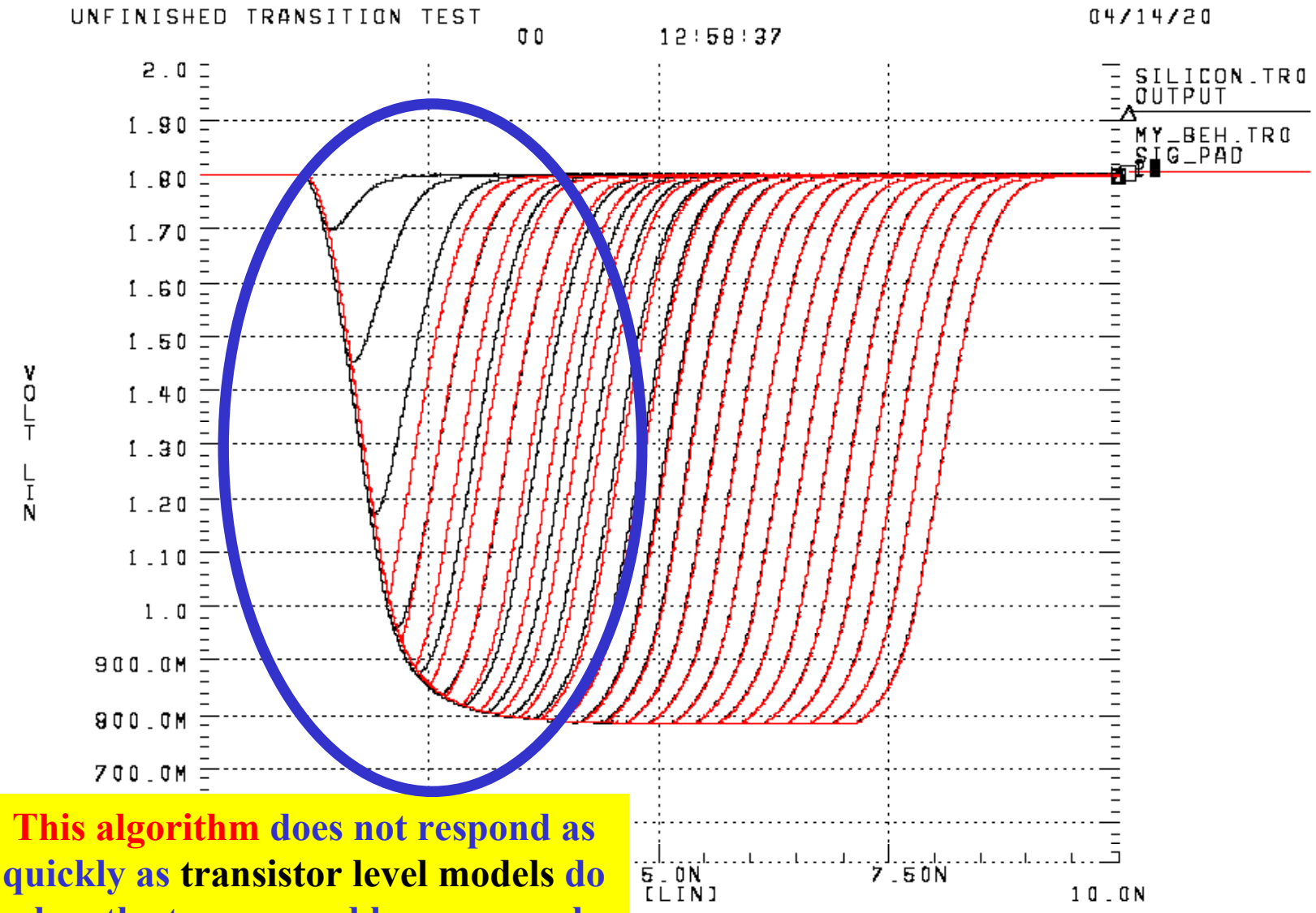
Start new edge when triggered, but

Continue with old edge while its value is higher than the value of the new edge

Switch to new edge when its value is higher than old edge

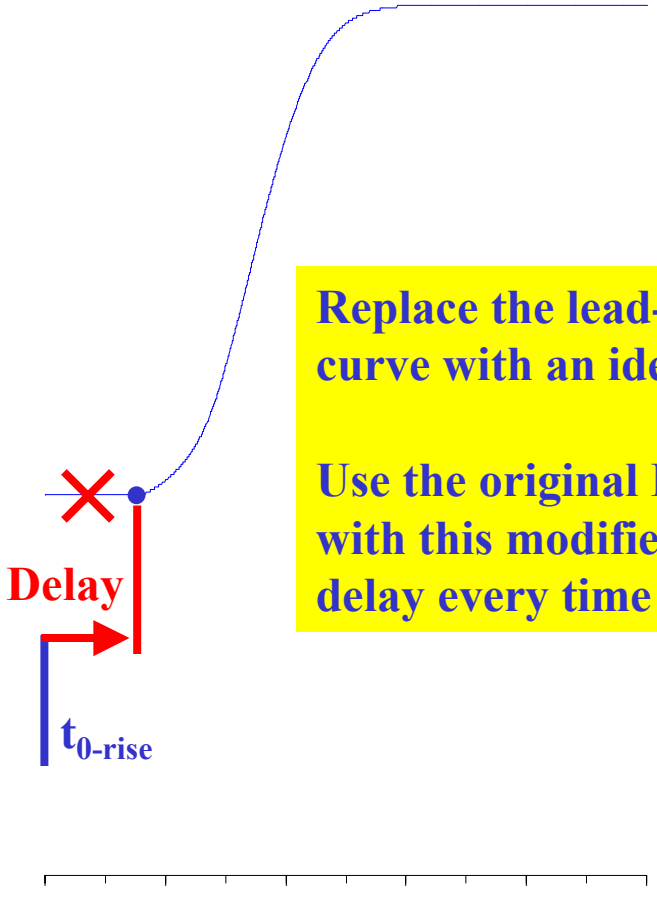
This method guarantees that the time from the trigger point to V_{meas} (t_{co}) remains the same

Arpad's first idea vs. transistor level model



This algorithm does not respond as quickly as transistor level models do when the turnaround happens early

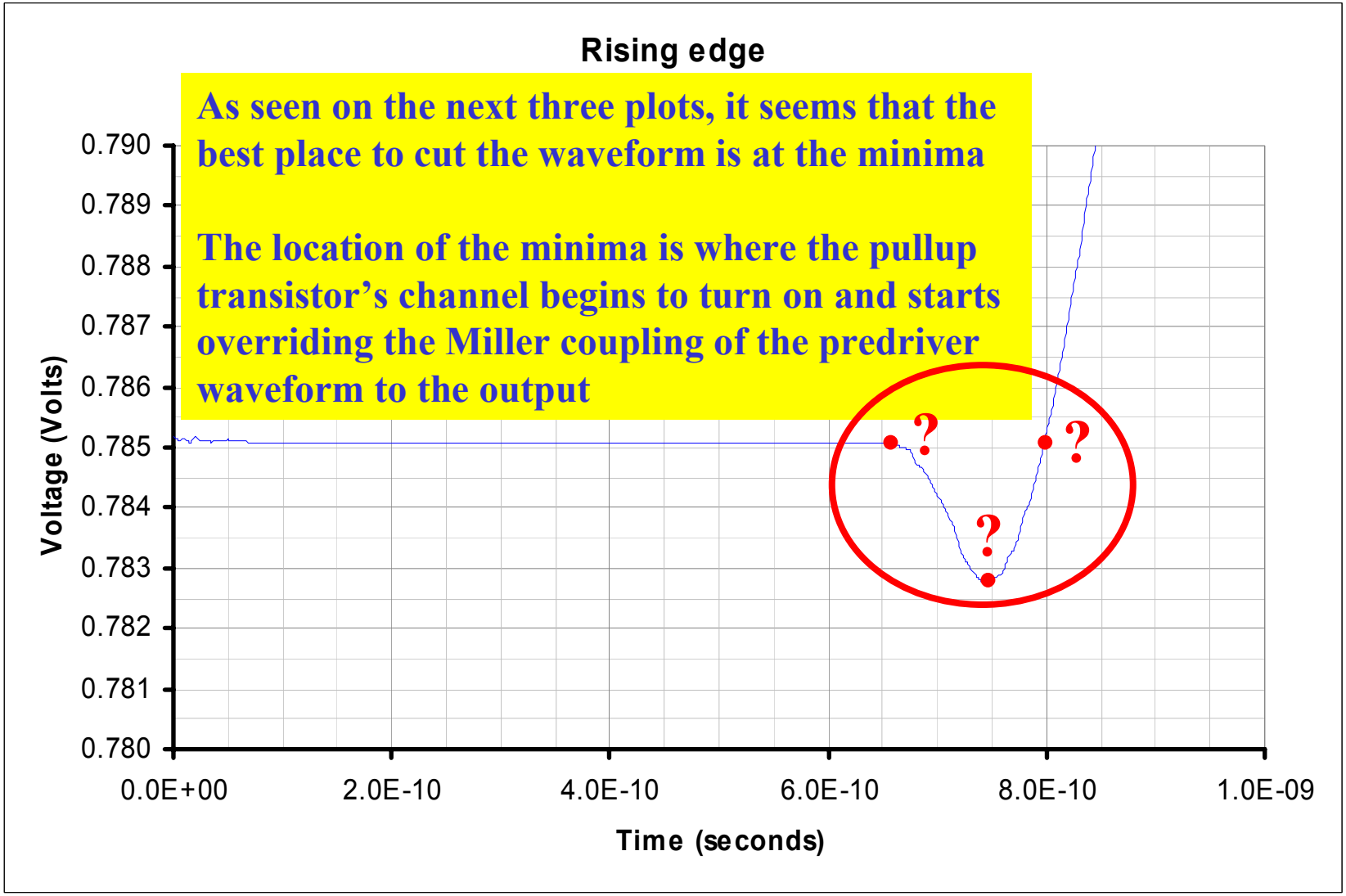
Arpad's second algorithm idea



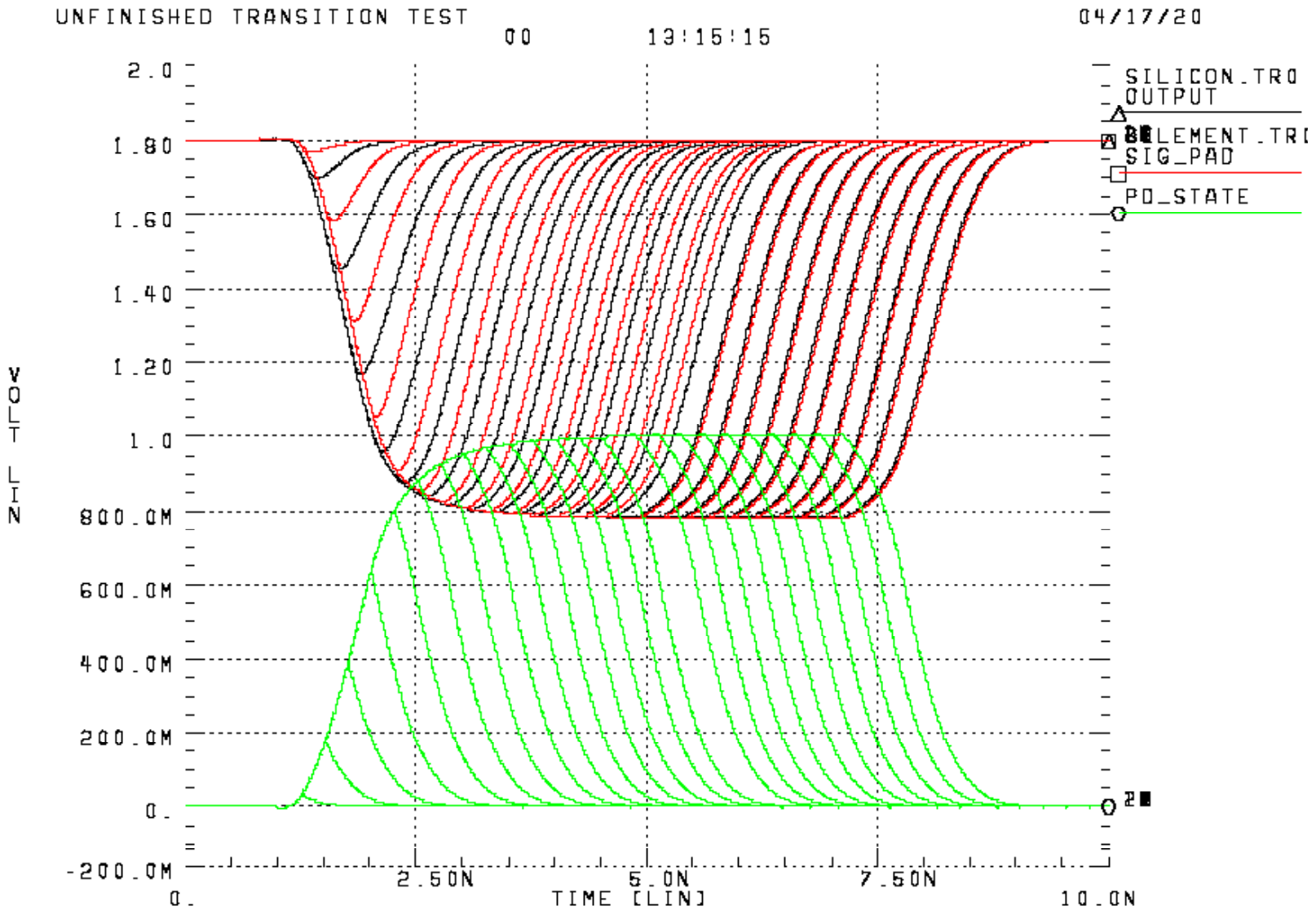
Replace the lead-in portion of the Vt curve with an ideal delay

Use the original B-element algorithm with this modified Vt curve, but add a delay every time it is triggered

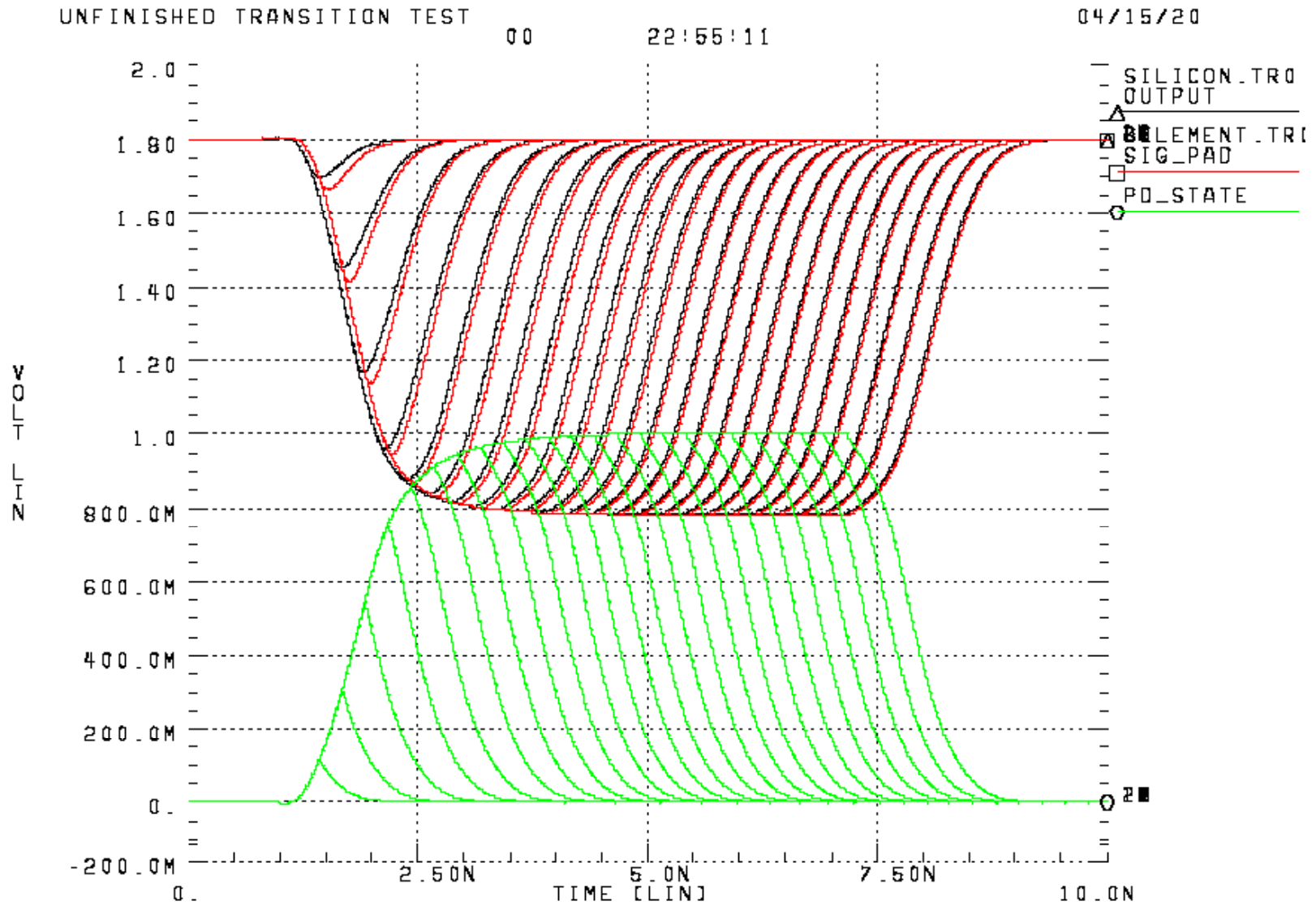
Where exactly should the Vt curve be cut?



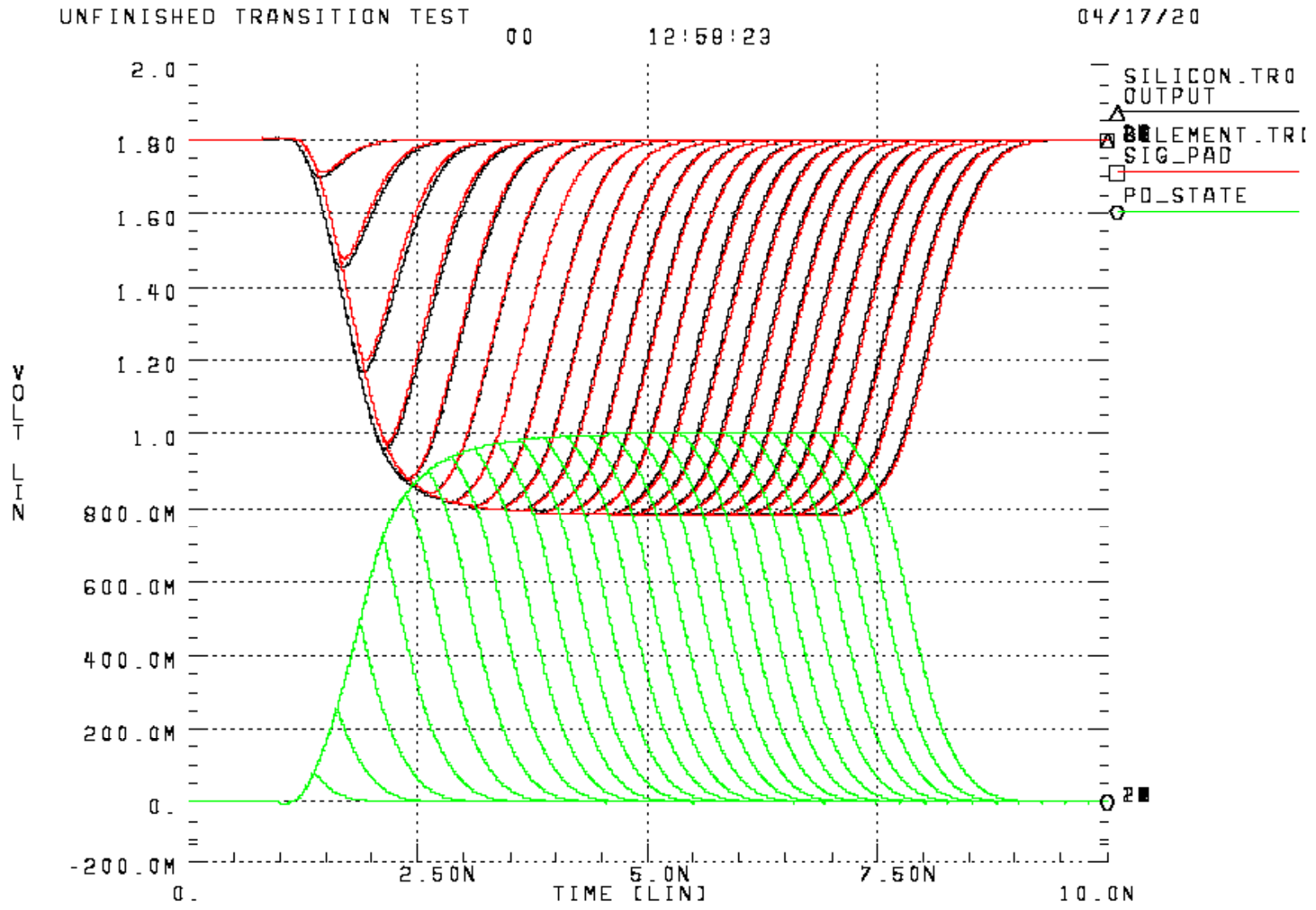
Vt curve cut at left side



Vt curve cut at right side



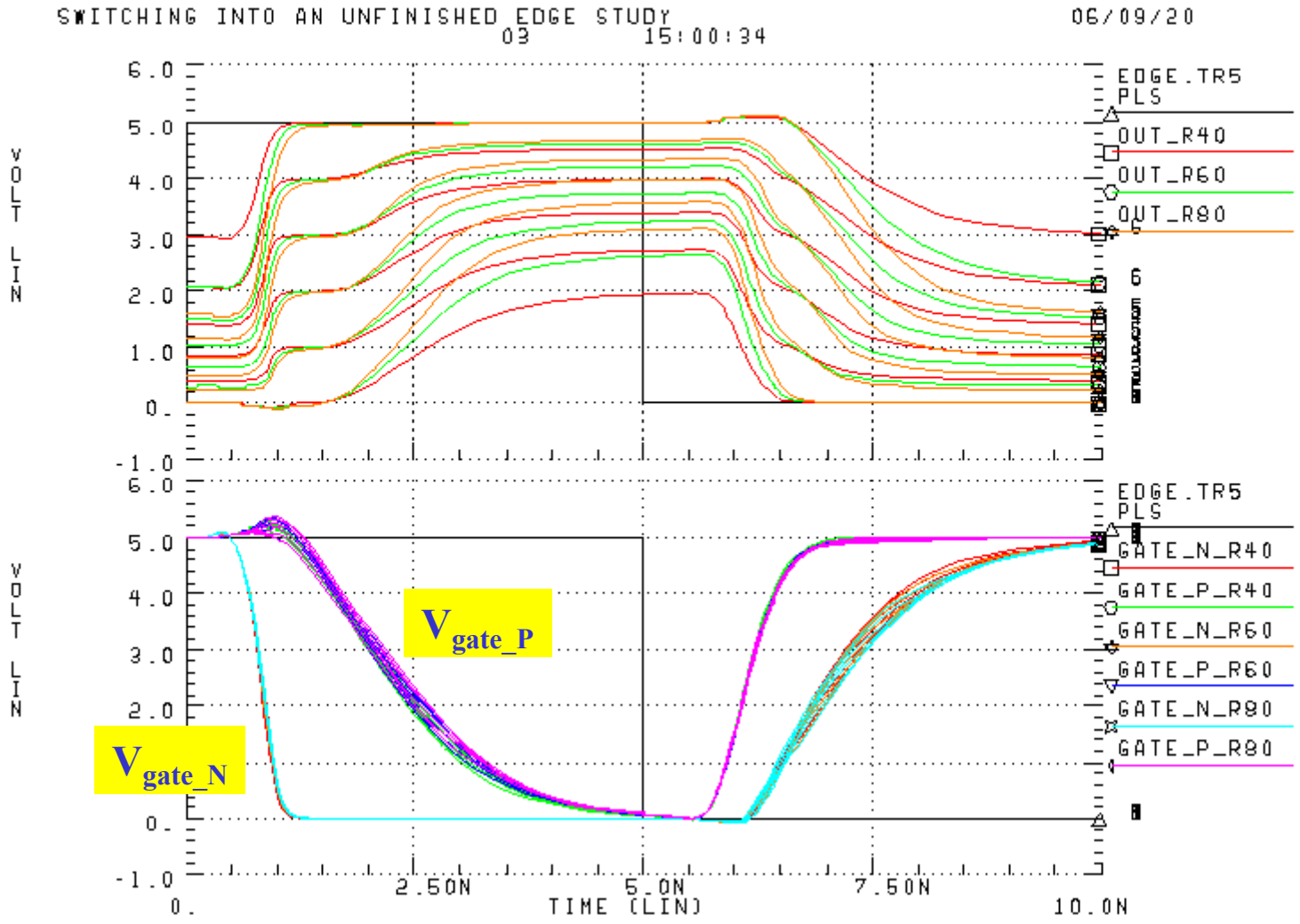
Vt curve cut at the minima yields best correlation



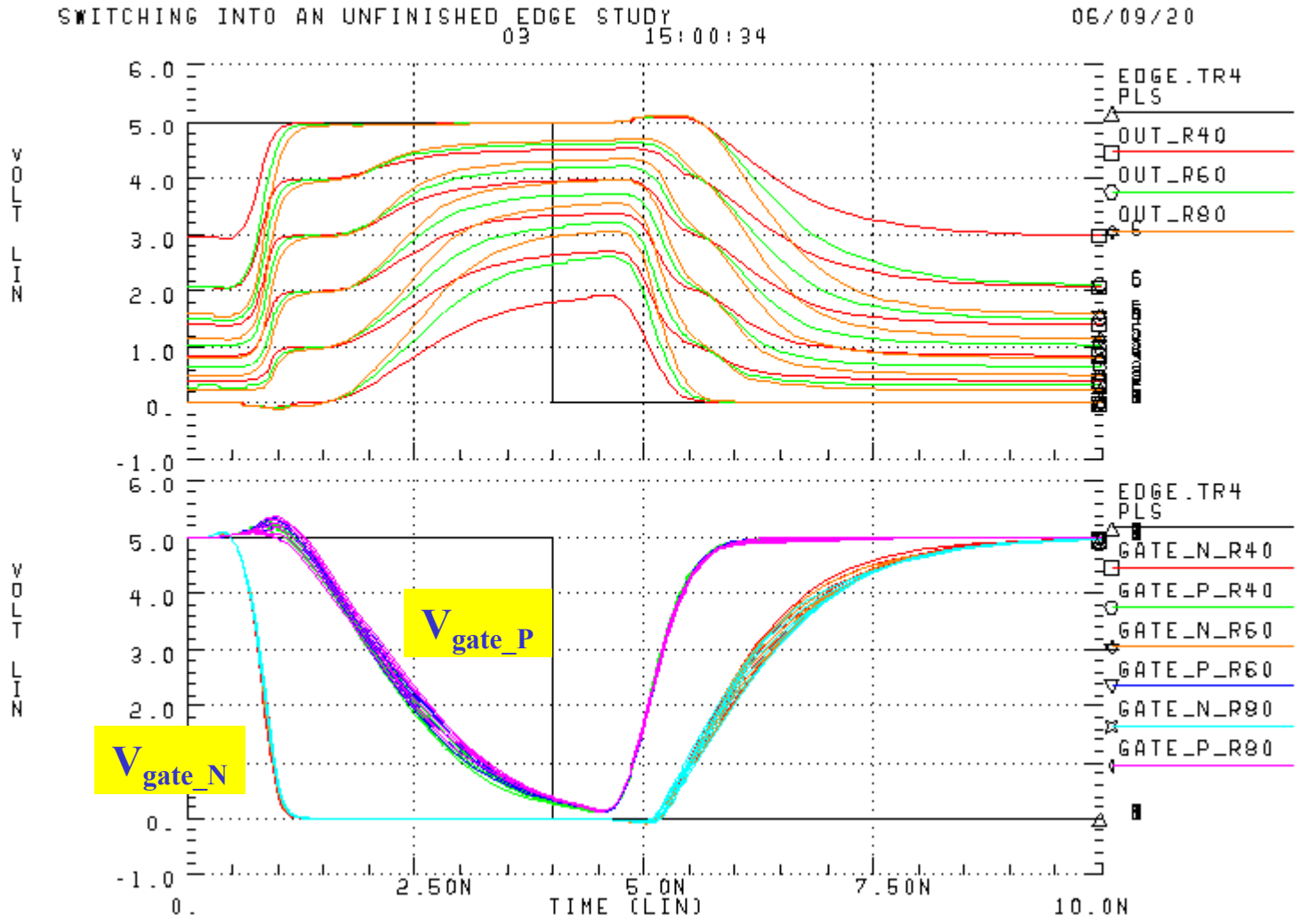
Looking inside the SPICE model for explanations

- We will look at two groups of six waveform sets.
  - Rising edge followed by a falling edge
  - Falling edge followed by a rising edge
- The second edge is brought closer to the first edge successively on each plot to watch the effects of over-clocking.
- The top plot shows waveforms on the output pad loaded by various $R_{fixture}$ $V_{fixture}$ combinations.
  - 40, 60, 80 Ω
  - 0, 1, 2, 3, 4, 5 V
- The bottom plot shows the pre-driver output waveforms for the N and P-channel output transistors (gate voltages).
- Watching the output and pre-driver waveforms relative to the stimulus pulse we can make some observations.
  - There is a constant delay from the pulse to the pre-driver waveform
  - This explains the need for a delay element to keep the duration of the lead in portion of the Vt curves (i.e. internal delay of buffer)
  - The pre-driver waveforms turn around from the point they reached when over-clocking occurs
  - This explains the need for “turning around” on the Vt curves also

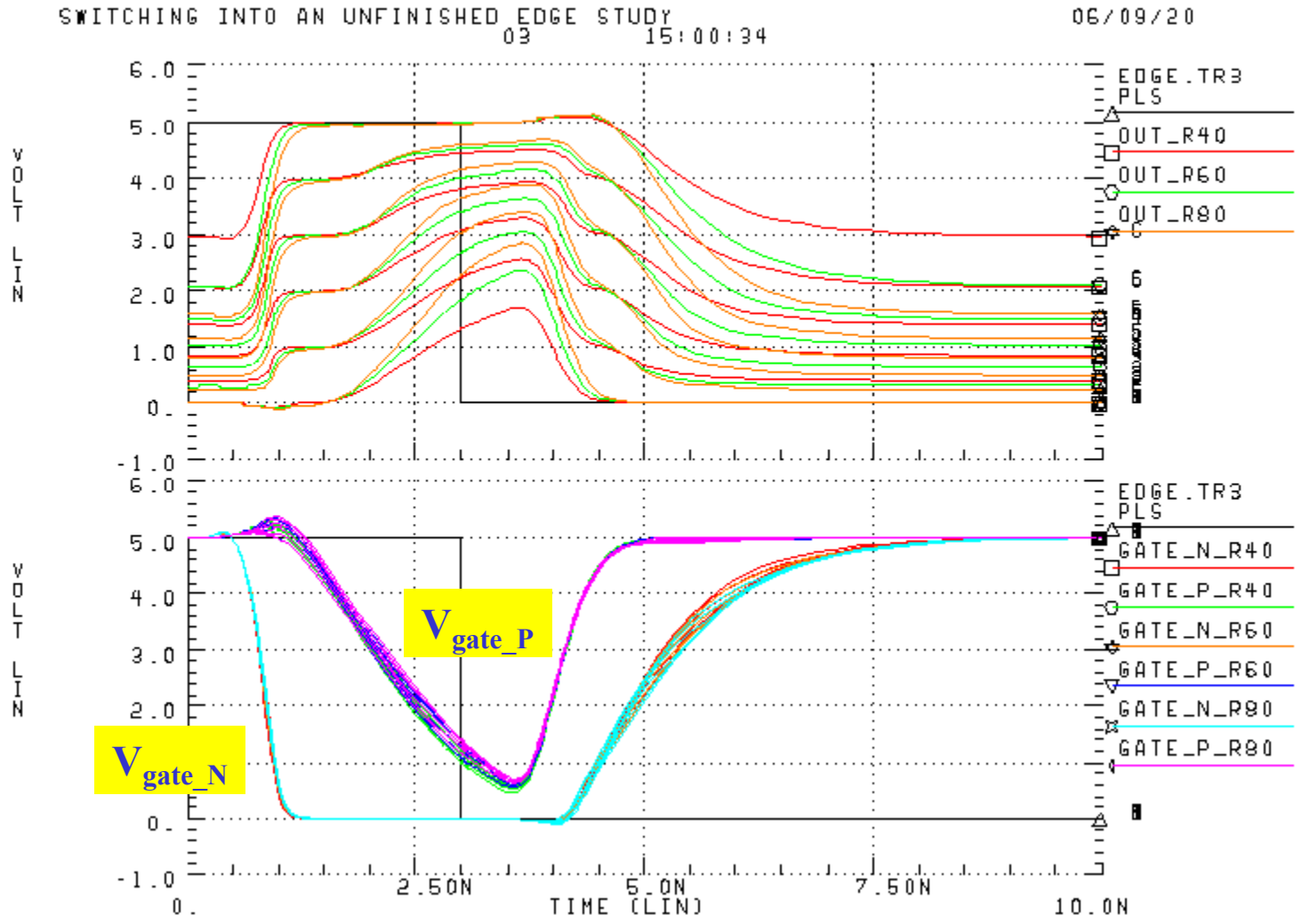
Pulse width = 5.0 ns



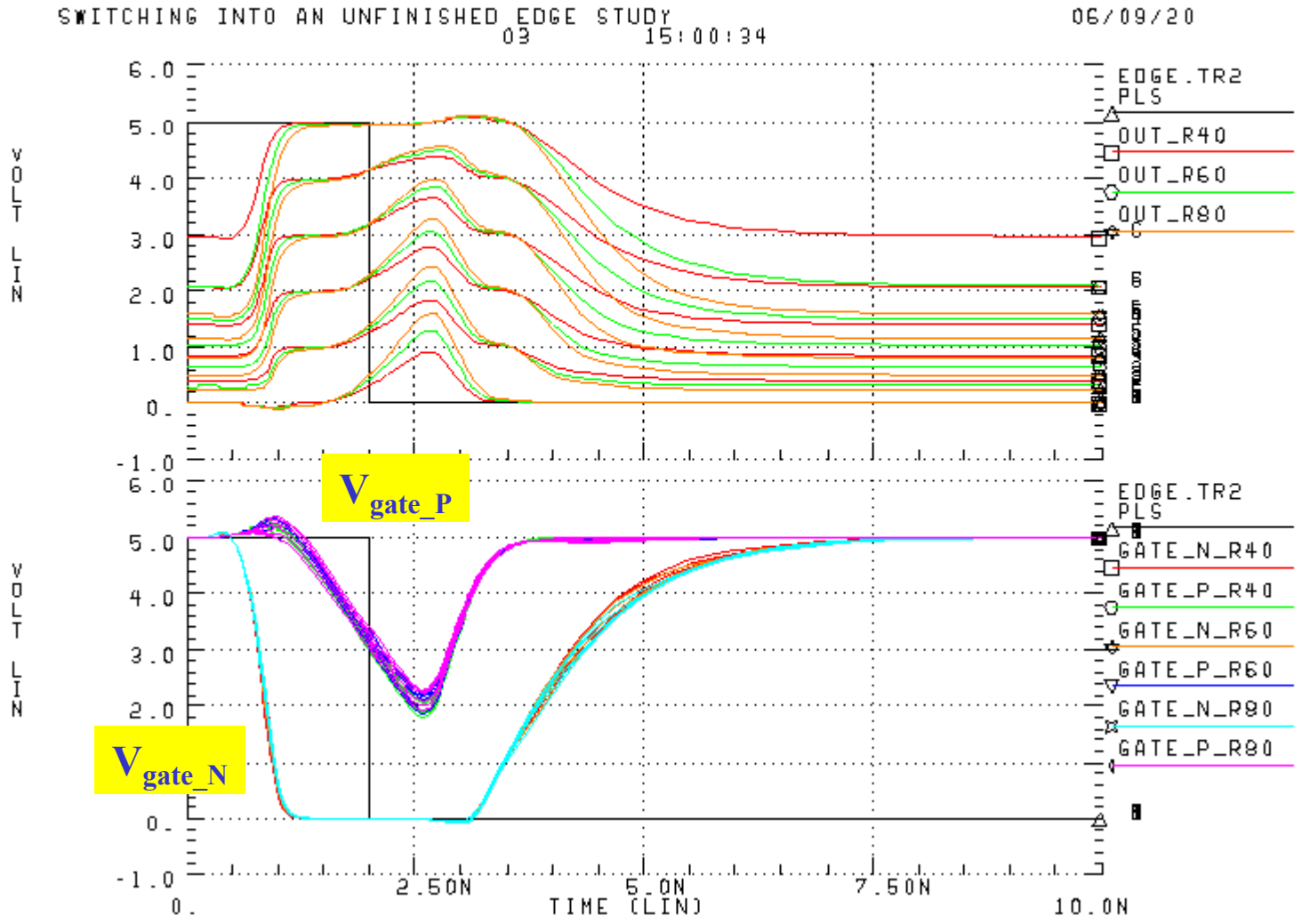
Pulse width = 4.0 ns



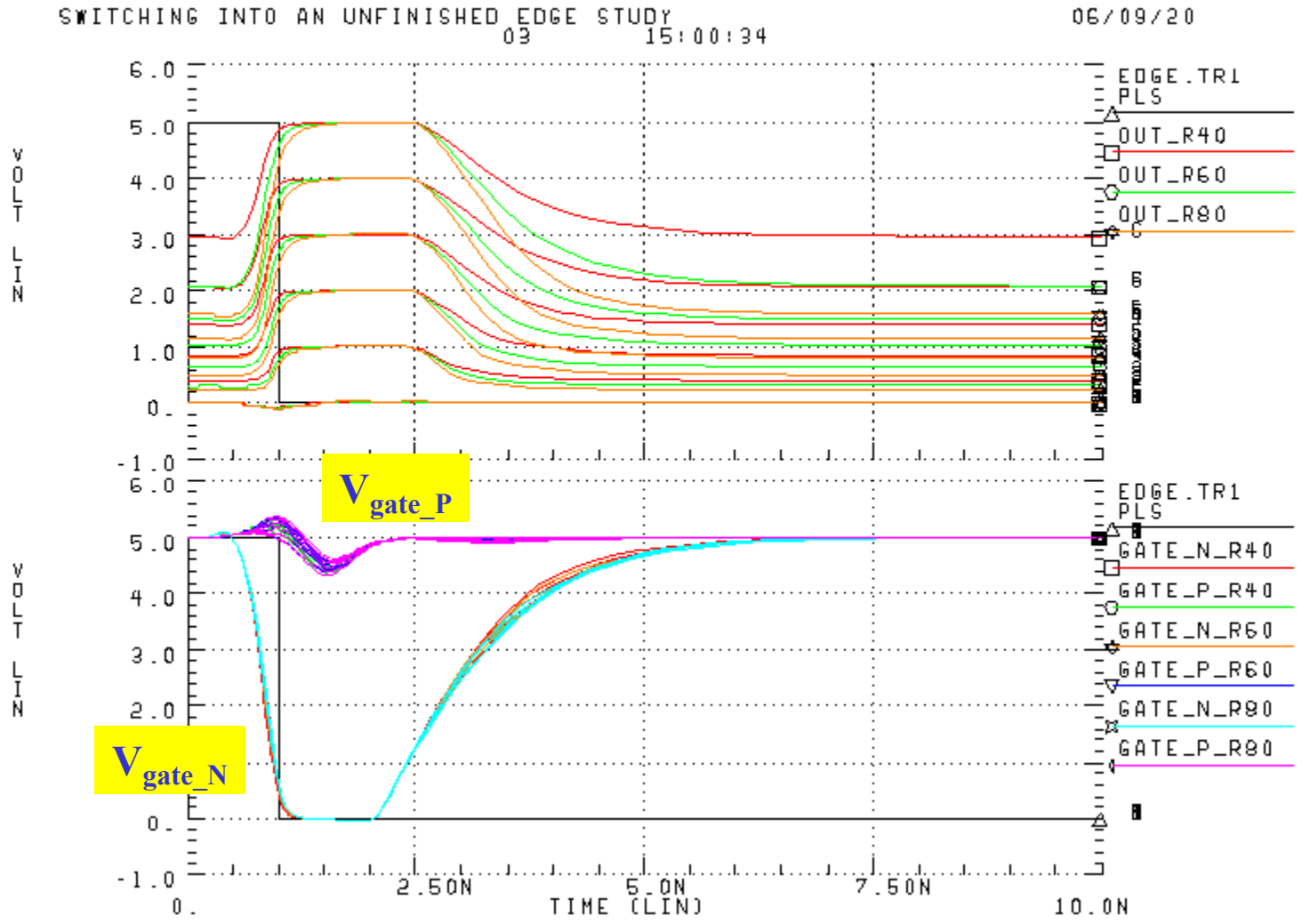
Pulse width = 3.0 ns



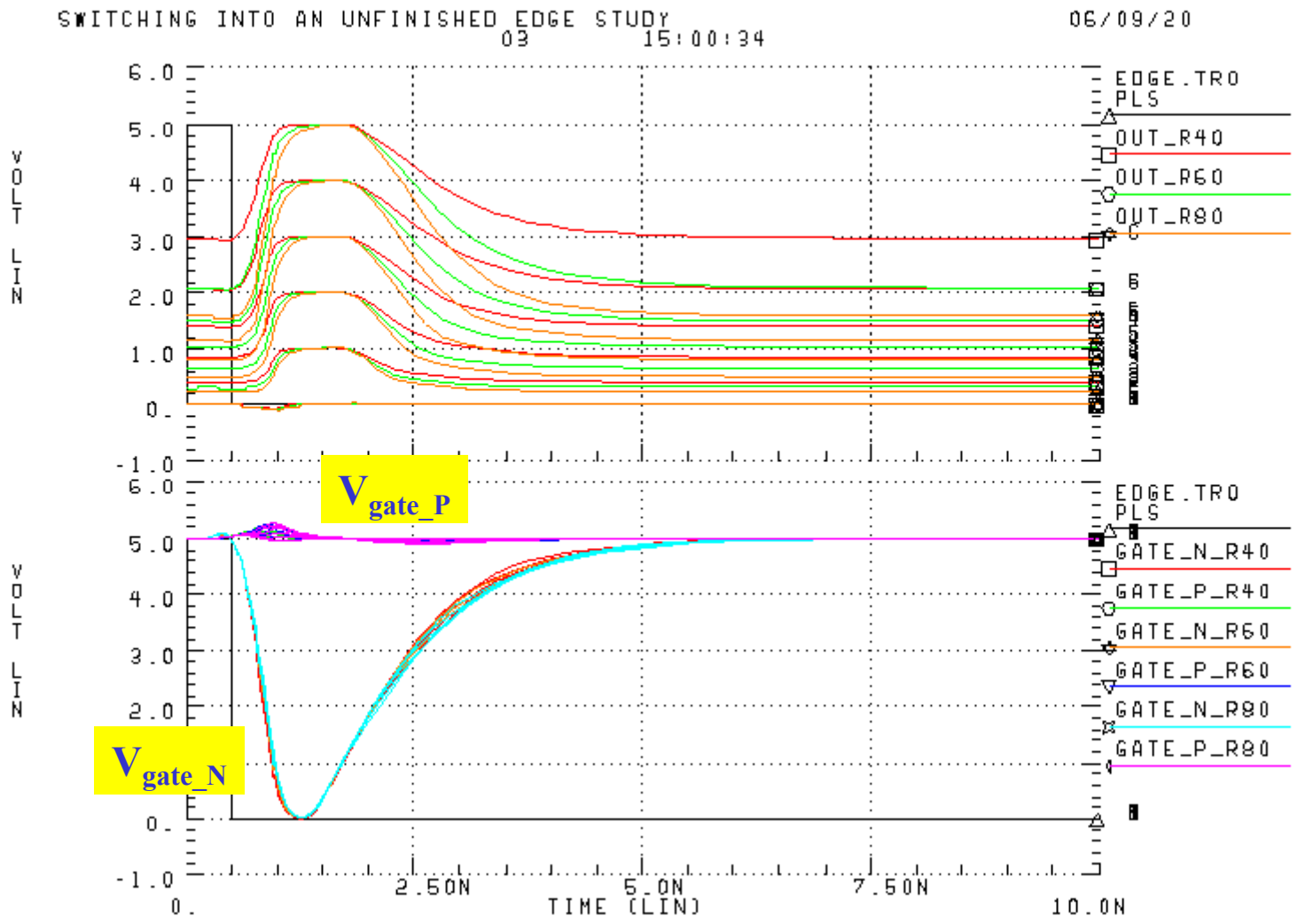
Pulse width = 2.0 ns



Pulse width = 1.0 ns



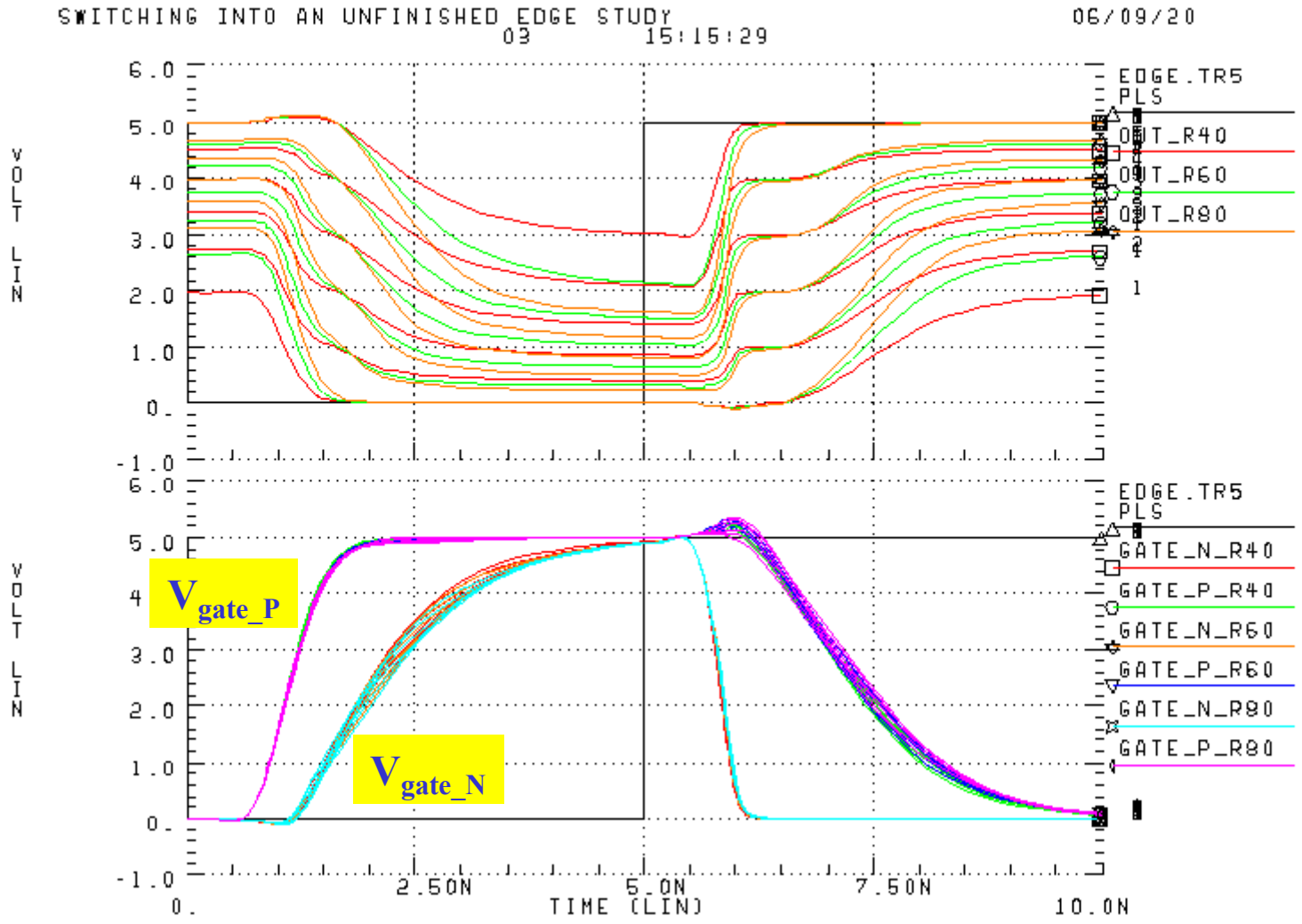
Pulse width = 0.5 ns



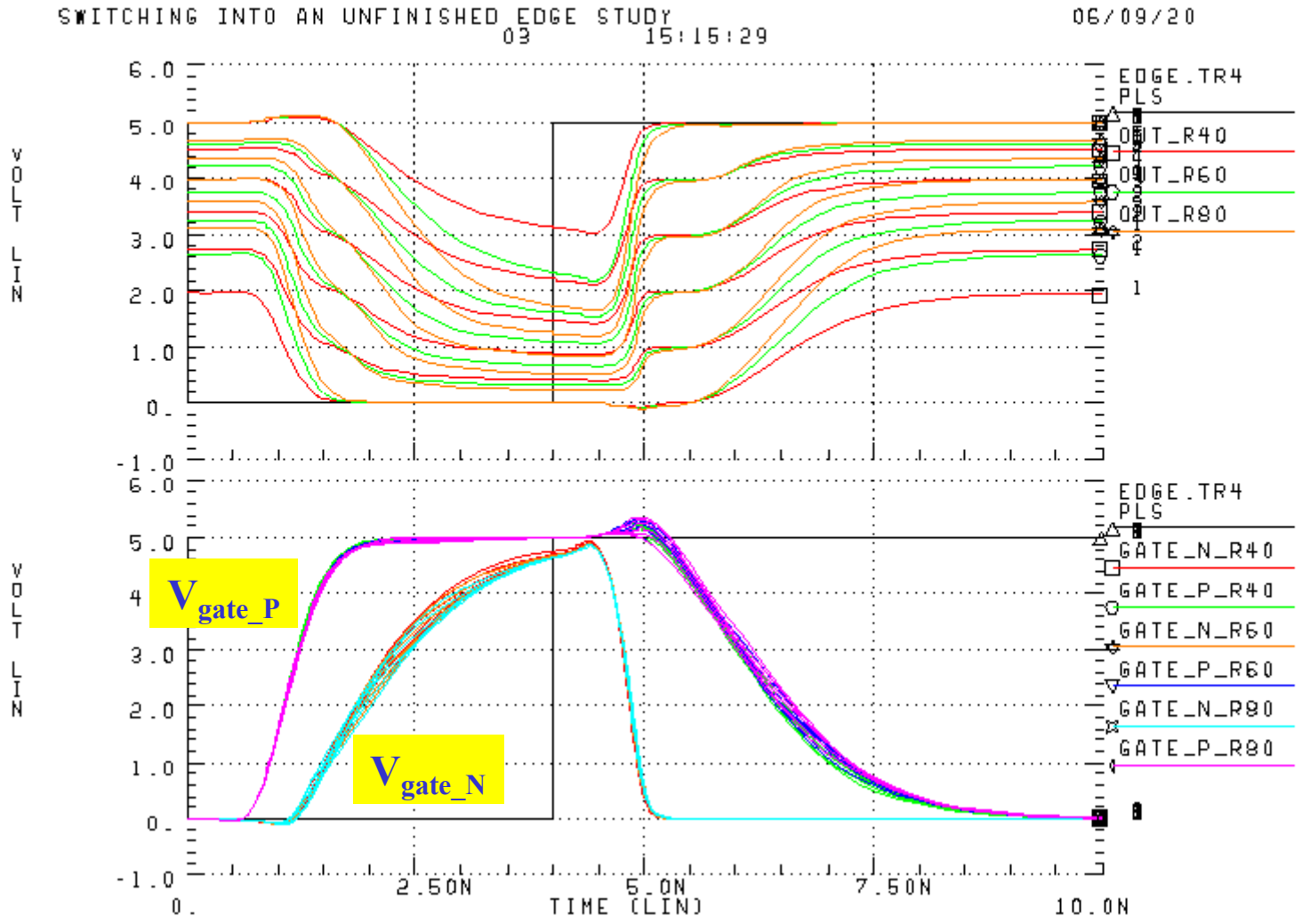
2nd set

- This page is put here only to serve as a separator page.

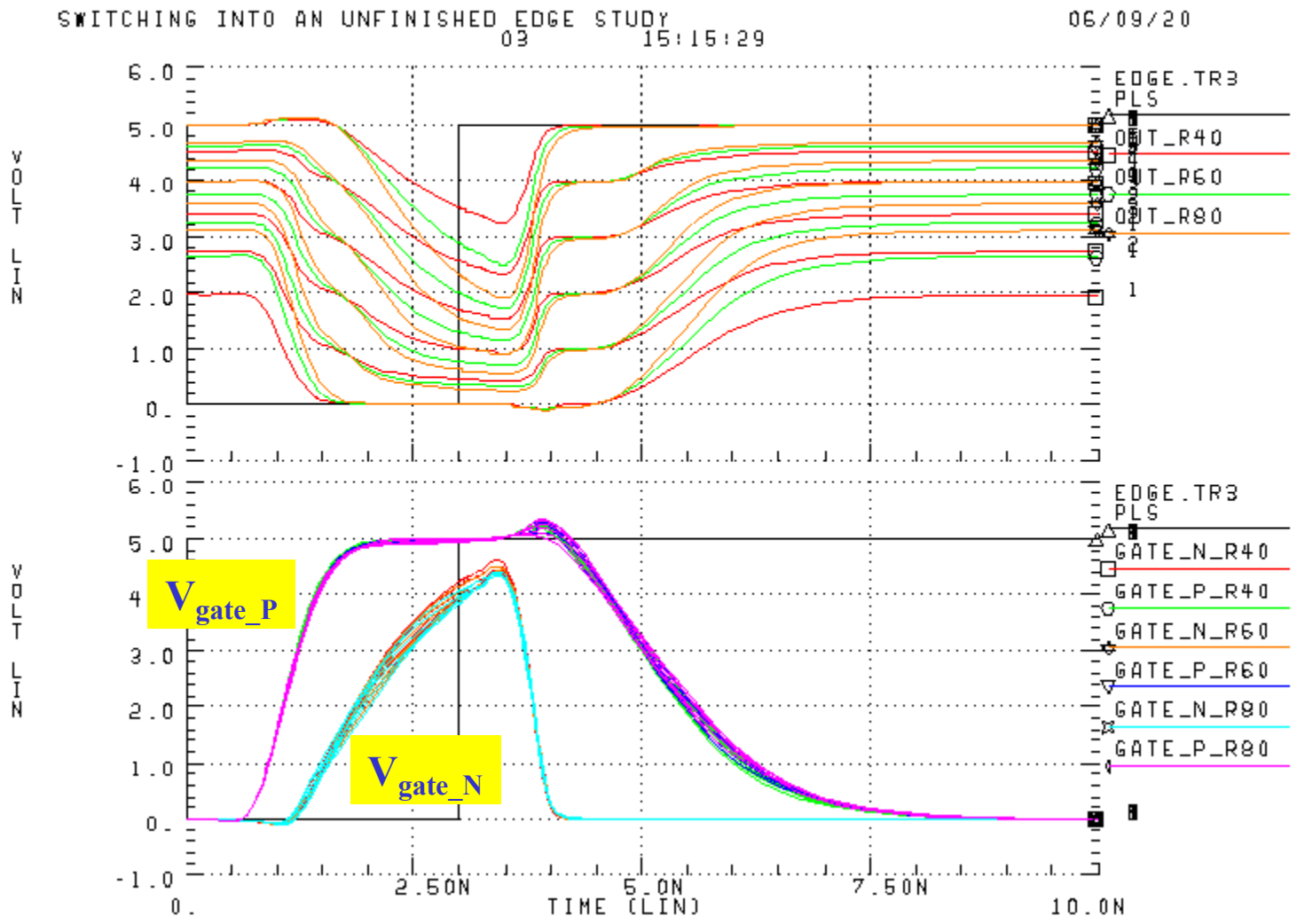
Pulse width = 5.0 ns



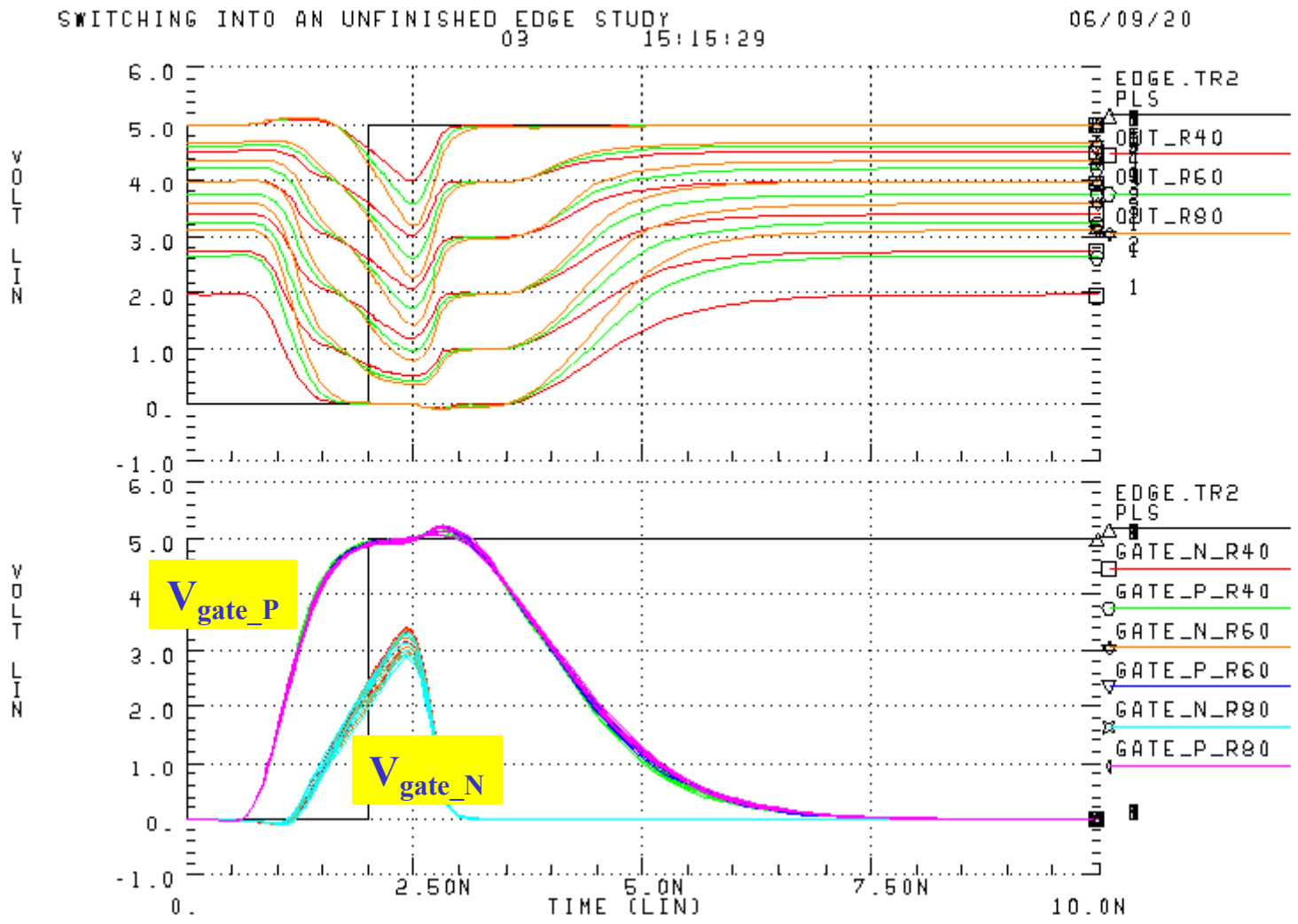
Pulse width = 4.0 ns



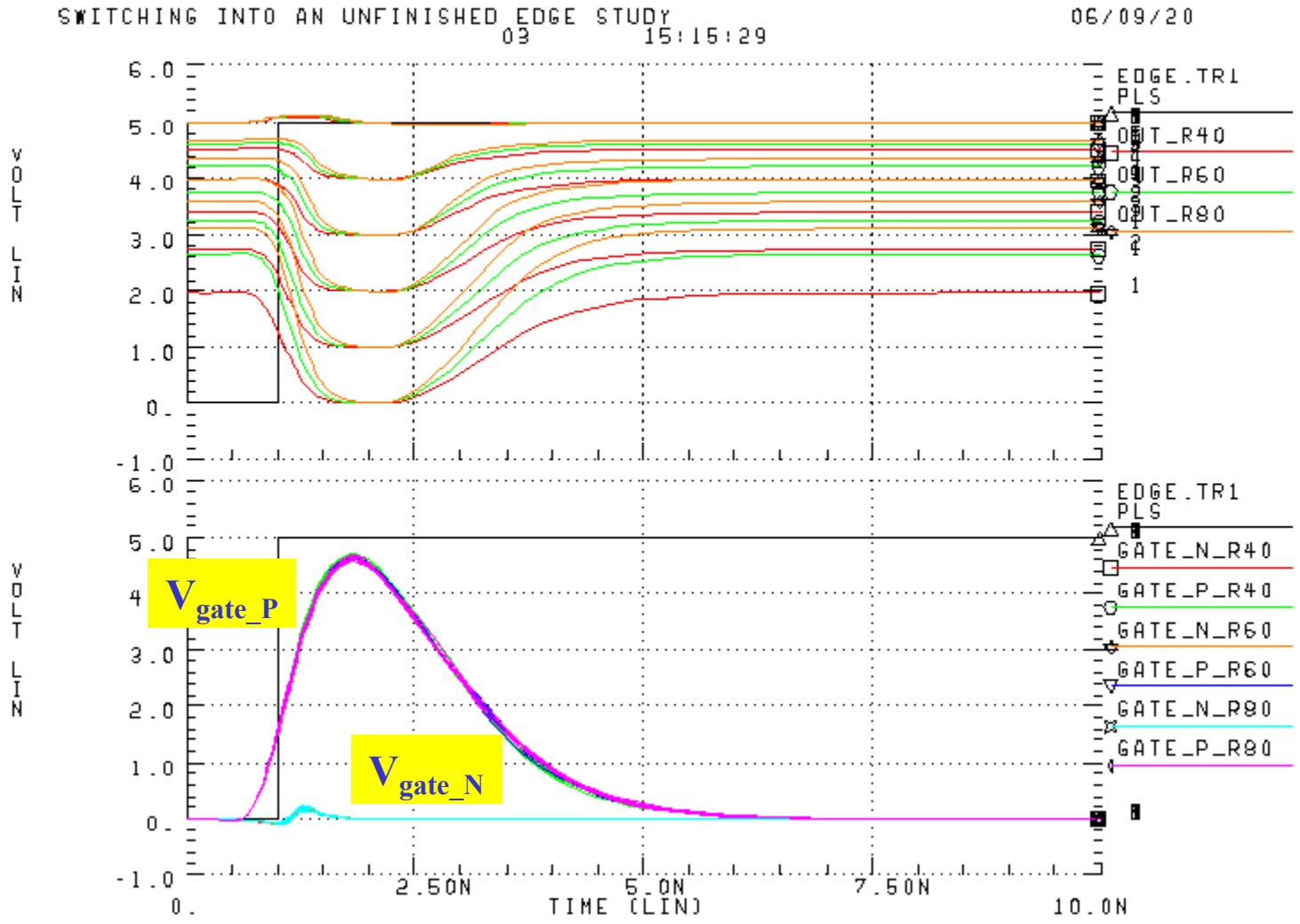
Pulse width = 3.0 ns



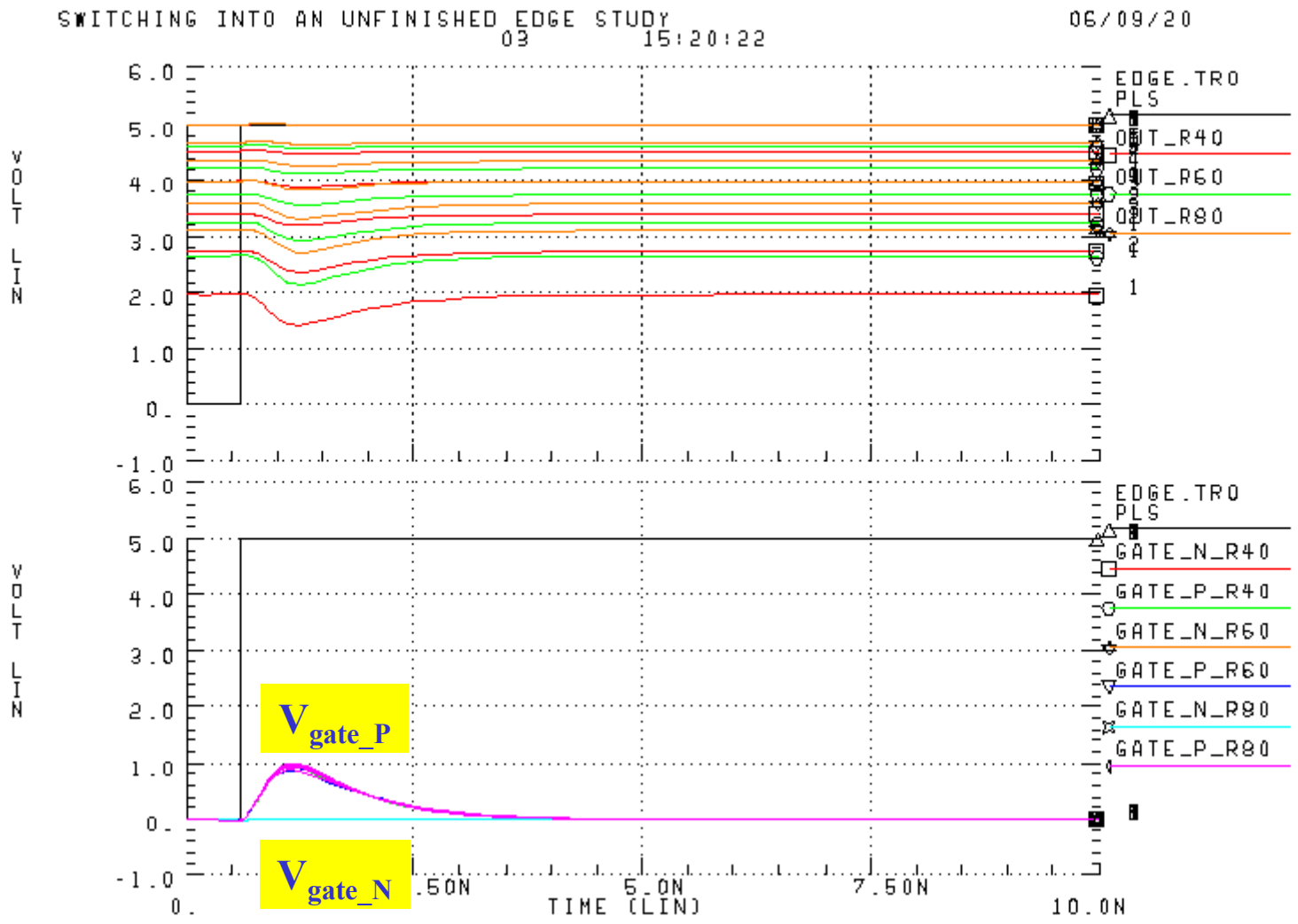
Pulse width = 2.0 ns



Pulse width = 1.0 ns



Pulse width = 0.6 ns



Conclusions

- **It may be possible to design an algorithm to handle over-clocking with existing IBIS data.**
  - **Looking at derivatives of the waveform could reveal the location where the output of the buffer really begins to switch**
  - **The shape of the Vt curve may not always be usable for this purpose as a general solution**
- **A new parameter representing the internal delay(s) of a buffer may be a better way to do it.**
  - **Need to develop a technique for model makers to measure this number**
  - **Need to add a new keyword or sub-parameter to the IBIS specification associated with the waveforms**