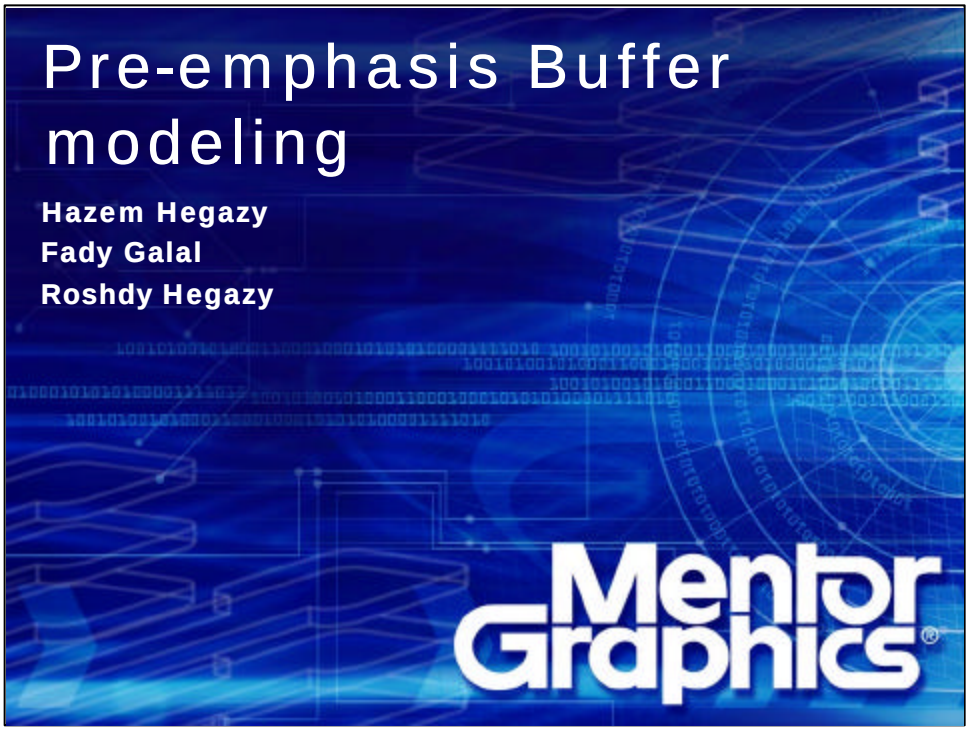


Speaker notes is present in this foils.

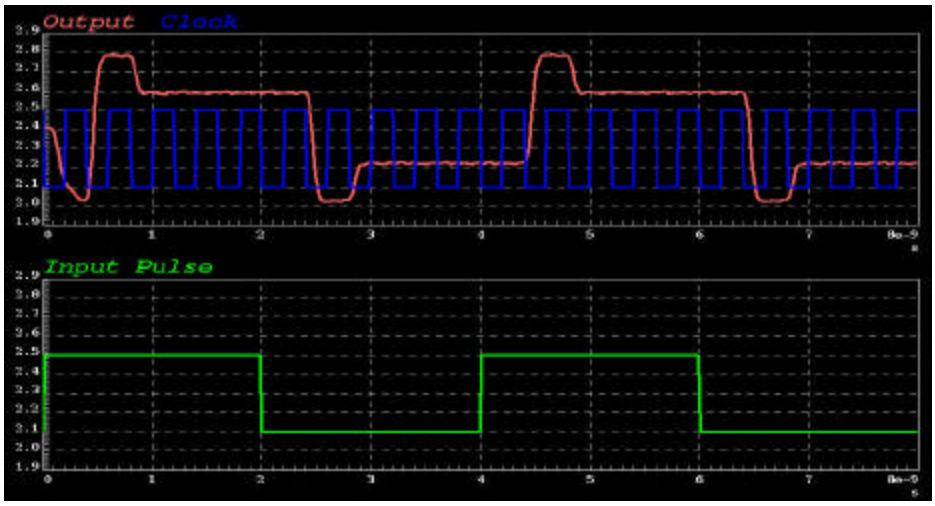
Agenda

- **Introduction**
- **DC through Transient simulation**
- **Modeling Technique (1)** **(One Model, No driver schedule)**
- **Modeling Technique (2)** **(4 Models with driver schedule)**
- **Summary**

Here is our agenda for today's presentation, We will introduce the pre-emphasis buffer behavior & what are the problems we face while modeling.

Then the work around for these problems followed by two techniques of modeling.

Pre-emphasis !!?

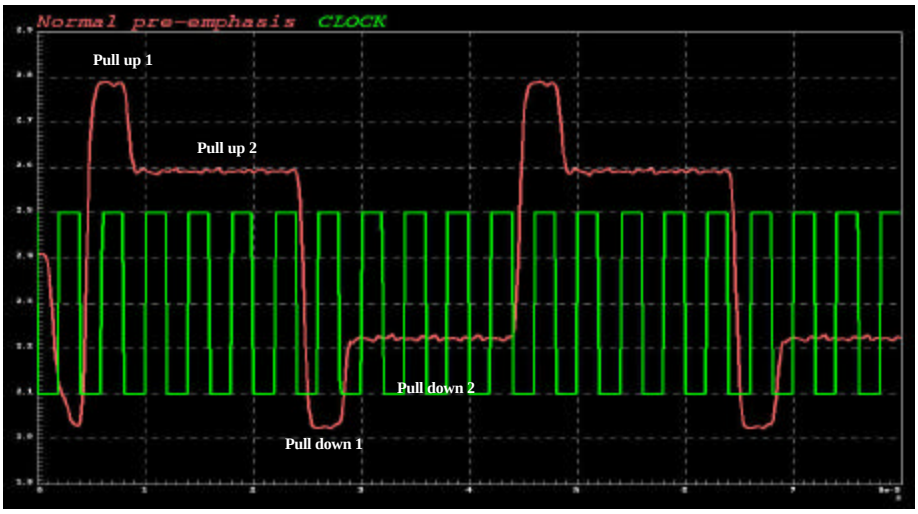


This slide shows 2 pulses of a pre-emphasis buffer.

As we can see that the waveform goes to the High-High level then after one clock period it goes to its final HIGH STATE at Medium-High level.

The pulse width/period are totally controlled by the input pulse, The clock only controls the period of the High-High & Low-Low levels.

Problems!!?

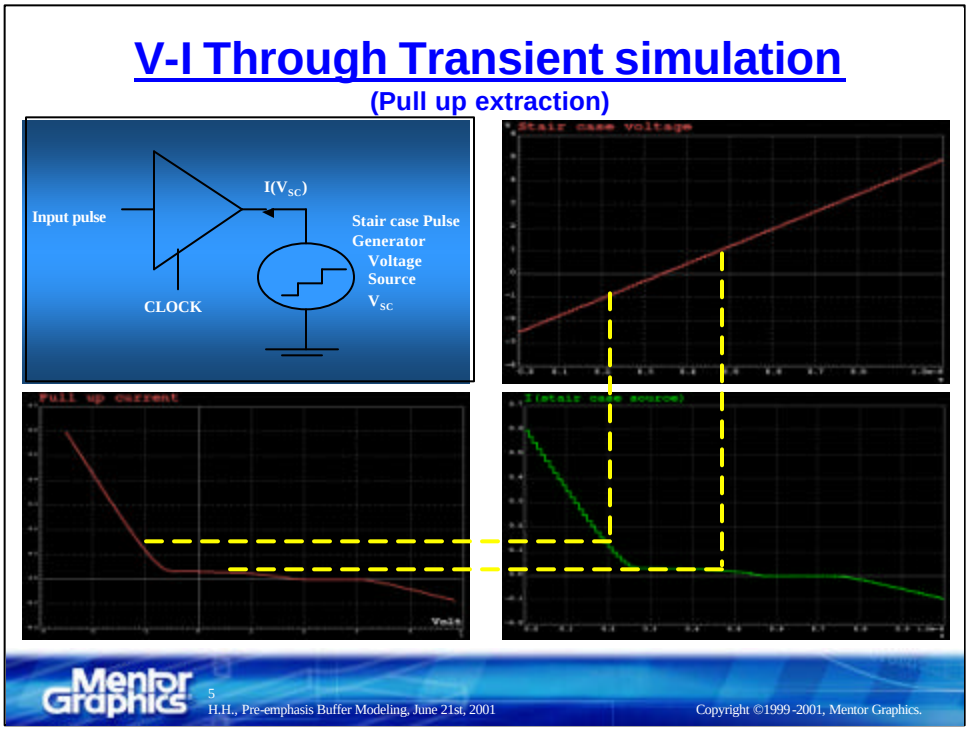


As usual there are some problems ??

-First, Notice the first rising edge has faulty DC level to begin from.

So, if we make normal DC sweep to get the pull up/down curves most probably we will get wrong DC curves.

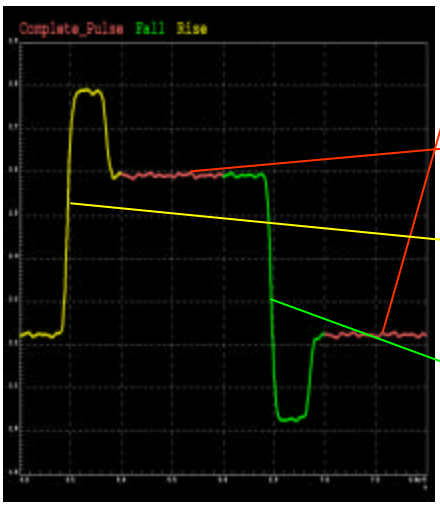
- Second, the Clock must be there in all simulations for proper operation of this kind of buffers. But, After one clock period the buffer will switch from High-High to Medium-High level. So, which DC will we get!!?



There is no other way to extract the DC curves unless the clock will be operating. So, what is the DC solution … it is the steady state one.

Then we can make the buffer operates normally (input pulse and normal clock signal) with our settings to have a very wide pulse width to assure steady state conditions. At the same moment we have a voltage source at the output PAD sweeping from –VDD to 2*VDD by stair case shape. Every stair will be at 0.1V apart from the previous one. With a long period between transition to reach steady state.

Modeling Technique (1)



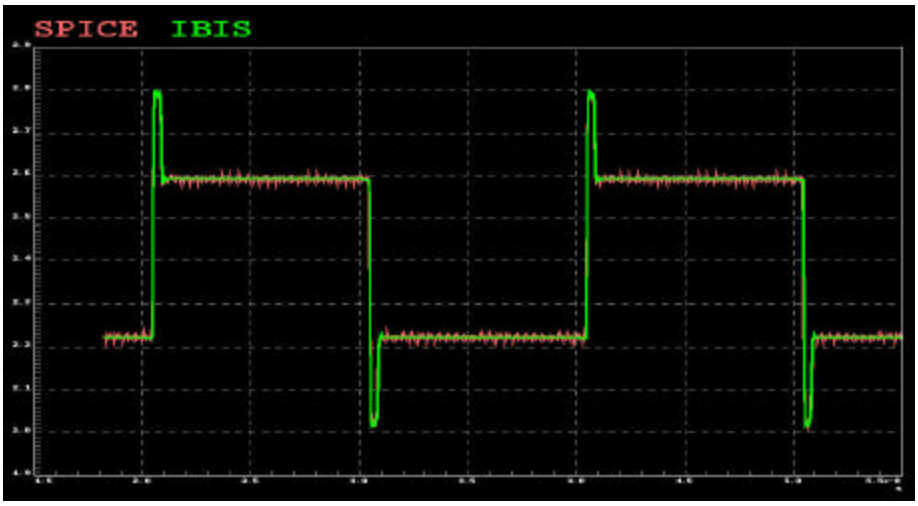

```
|
[Model]    example
..
..
[Pull down]          || Pull down 2
|
| voltage     I(typ)     I(min)       I(max)
V1            I1          ..            ..
..
..
[Pull up]            || Pull up 2
|
| voltage     I(typ)     I(min)       I(max)
V1            I1          ..            ..
..
..
[Rising Waveform]
R_fixture = 50
V_fixture = Vref
……
| time       V(typ)    V(min)      V(max)
|
t1                      V1             …      ….
…
[Falling Waveform]
R_fixture = 50
V_fixture = Vref
…..
| time       V(typ)    V(min)      V(max)
|
t1                      V1             …      ……
…….
|
[End]
```


Okay Now we can control the clock pulse width together with the input pulse to get the right DC curves we wants.

For the modeling technique(1) we will take Medium-High & Medium-Low levels to be our Pull up & pull down respectively.

The rising wave form in the IBIS file will be the yellow on and the falling waveform will be the green one to suit the DC currents extracted before.

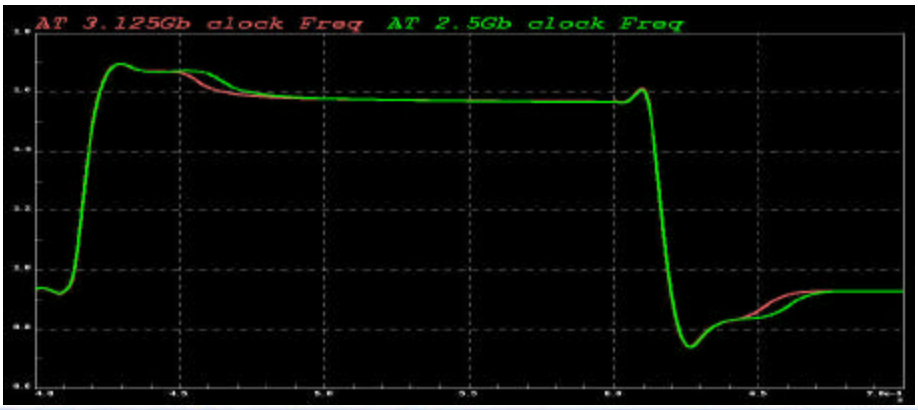
Validation



This is the validation (SPICE versus IBIS) using modeling technique (1).

Limitations

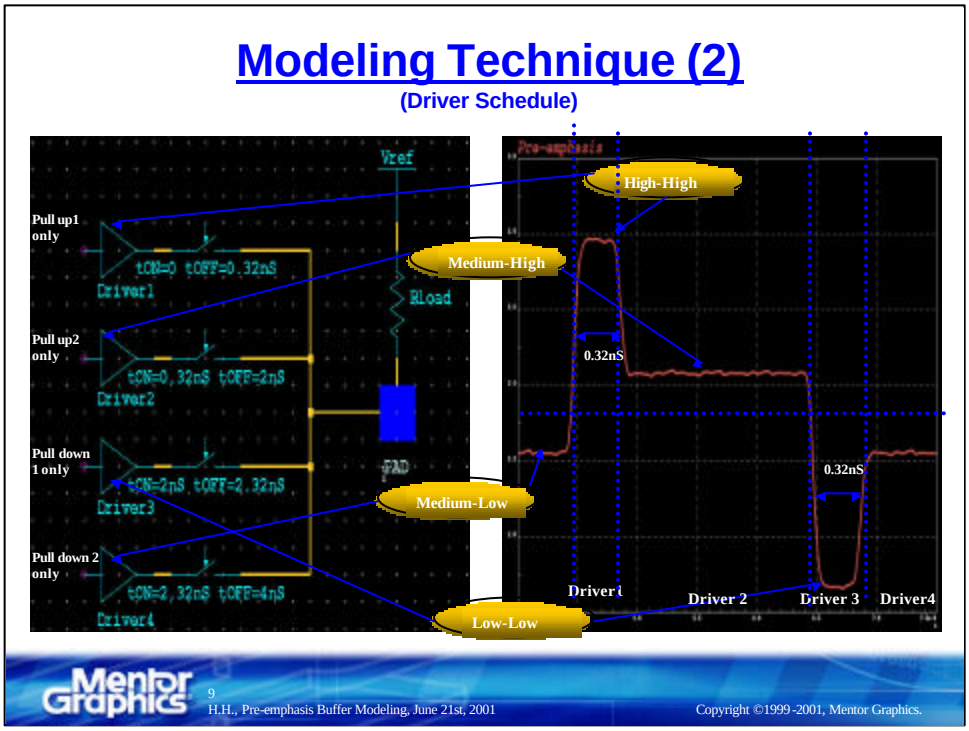
- **Non-monotonic wave forms(For some EDA tools)**
- **Single clock frequency operation**



Modeling technique (1) has two limitations:

-As the wave form will go first to High-High level then to the Medium-High level so, in some EDA tools these waveforms might be considered nonmonotonic.

- Some pre-emphasis buffers has clock frequency range so, it stays at the High-High level for one clock period then goes down to Medium-High level. By applying this technique we will have waveforms in the IBIS file represents only single frequency.



Here we came up with the modeling technique (2) by using driver schedule.

We will split the pre-emphasis into four buffers (depending on the number of dc levels in its waveform) each will represent one DC level.

Driver 1: has only pull up curve (pull down current is zero-open source-) to represents High-High level.

Driver 2: has only pull up curve (pull down current is zero-open source-) to represents Medium-High level.

Driver 3: has only pull down curve (pull up current is zero-open drain-) to represents Low-Low level.

Driver 4: has only pull down curve (pull up current is zero-open drain-) to represents Medium-Low level.

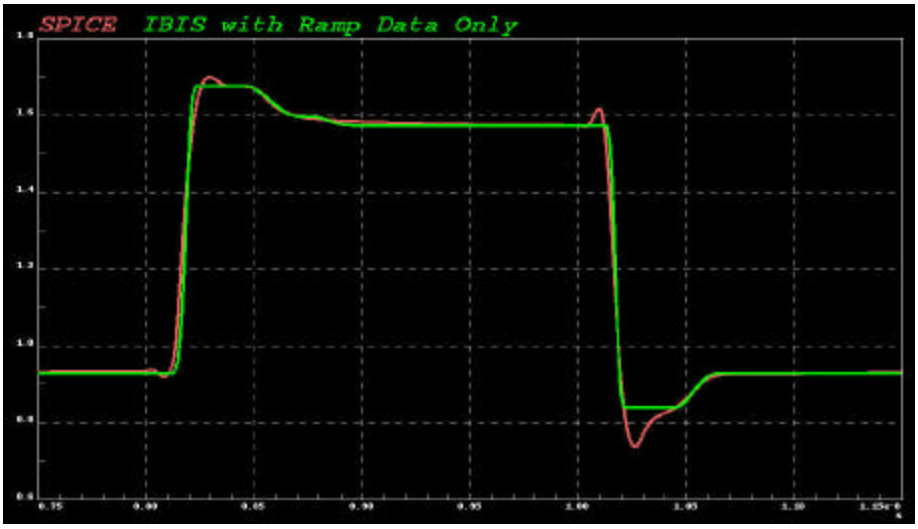
We switch between these four models in a way that suits the original wave form at whatever clock frequency.

Driver schedule keyword

- [Driver Schedule]
- |
- |Model_name Rise_on_dly Rise_off_dly Fall_on_dly Fall_off_dly
- |
- |
- High-High 0.0s 0.32ns NA NA
- Medium-High 0.32ns 2ns NA NA
- Low-Low NA NA 0.0ns 0.32ns
- Medium-Low NA NA 0.32ns 2ns

Note: The user should change the RED durations according to the operating frequency.(No need for re-modeling)

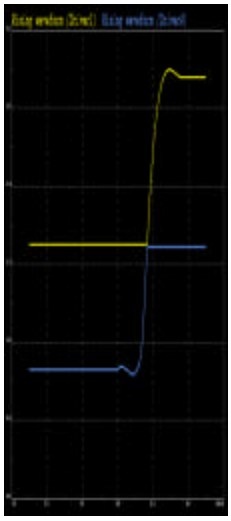
Validation (Ramp Data)



This is the validation by using the previous driver schedule keyword but with IBIS model contains only Ramp data.

You can see that it switches exactly with spice but the wave form has some differences.

Validation (Rising wave forms)



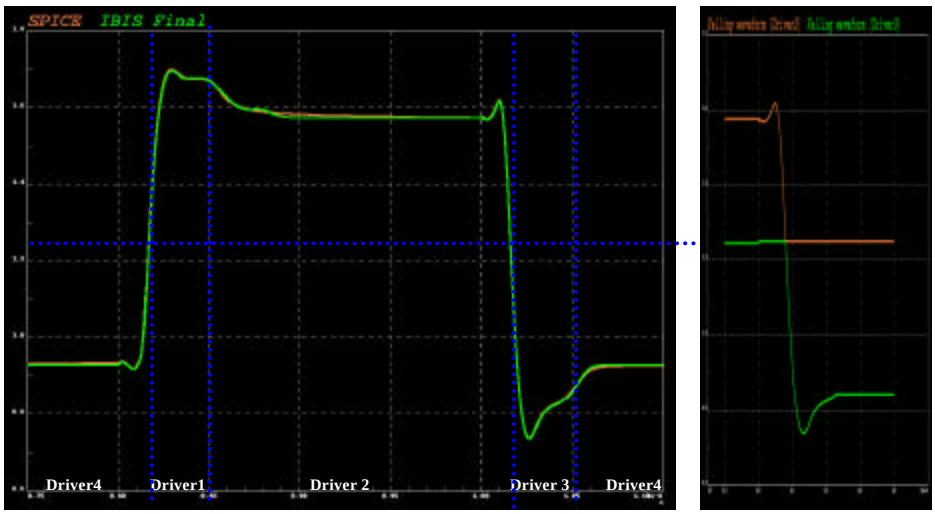
Driver4

Here we made an enhancement by adding rising wave forms into both driver 4 & driver 1 .

It is the same wave form in the spice but splitted at 1.25 (the load is 50 Ohm to 1.25 V Vref).

You can see the validation is getting better in the rising edge region and the falling edge is still the same as we haven't add falling wave form in both driver2 & driver3 yet.

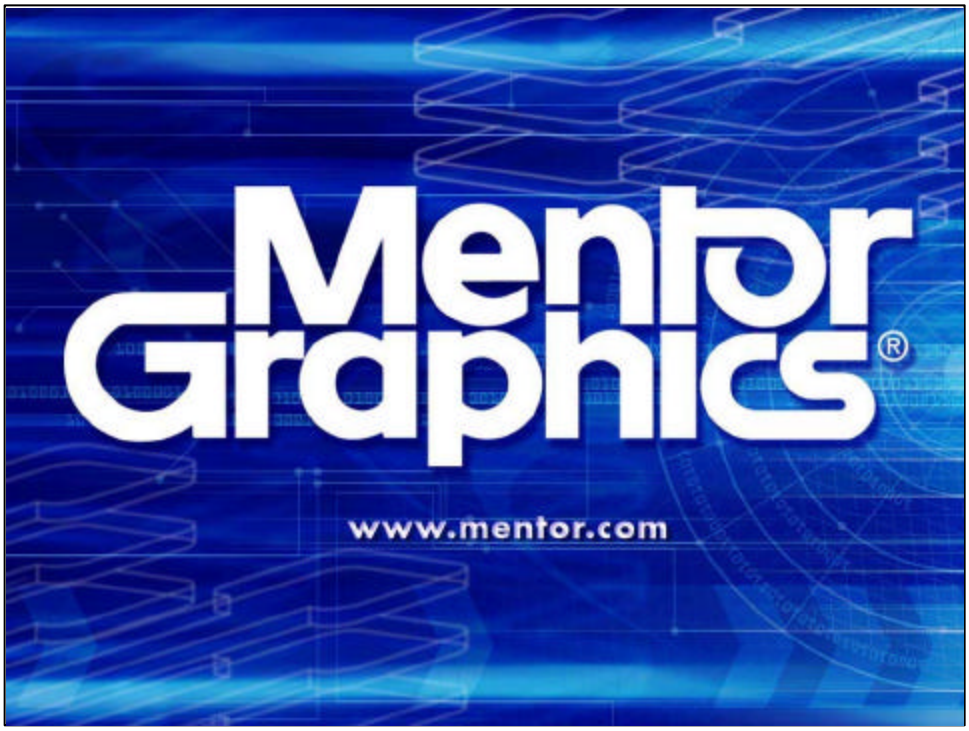
Validation (Falling waveforms)



Here we did the same for the falling wave form and our validatio n is excellent.

Summary

- **V-I curves Extraction from Transient simulation**
- **Modeling approach (1) for single frequency operation.** **(Changing the frequency needs remodeling)**
- **Driver Schedule Technique for frequency range operation.** **(Changing the frequency doesn't need remodeling)**


Mentor Graphics®
www.mentor.com