Improving Power Supply Induced Jitter Simulation Accuracy

[1]Chulsoon Hwang, [1]Yin Sun, [2]Zhiping Yang, [3]Randy Wolff, and [4]Bob Ross

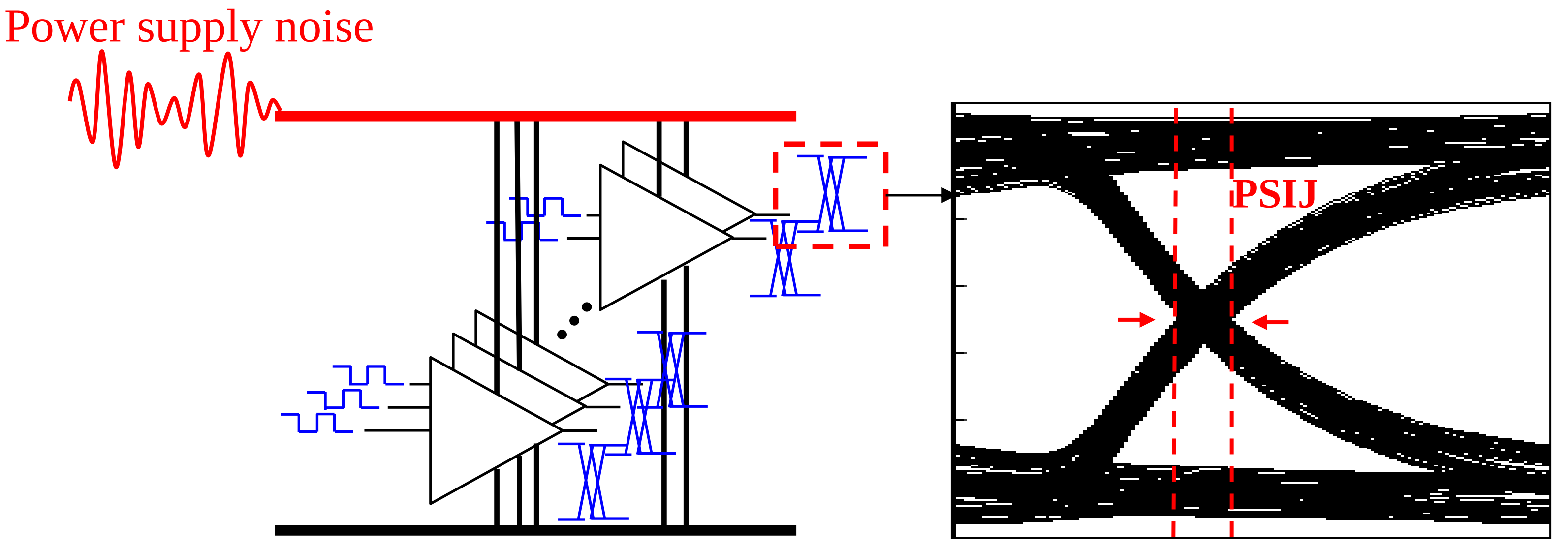
[1]Missouri University od Science and Technology
[2]Google
[3]Micron
[4]Teraspeed

Acknowledgement

- Materials are reorganized and modified from the previous presentation

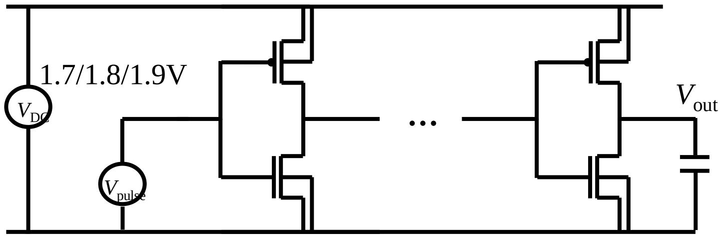
"Improving Power Supply Induced Jitter Simulation Accuracy for IBIS Model", IBIS Summit at IEEE Virtual Symposium on EMC+SIPI, 2020.

Power Supply Induced Jitter (PSIJ)



Limitation of IBIS Model

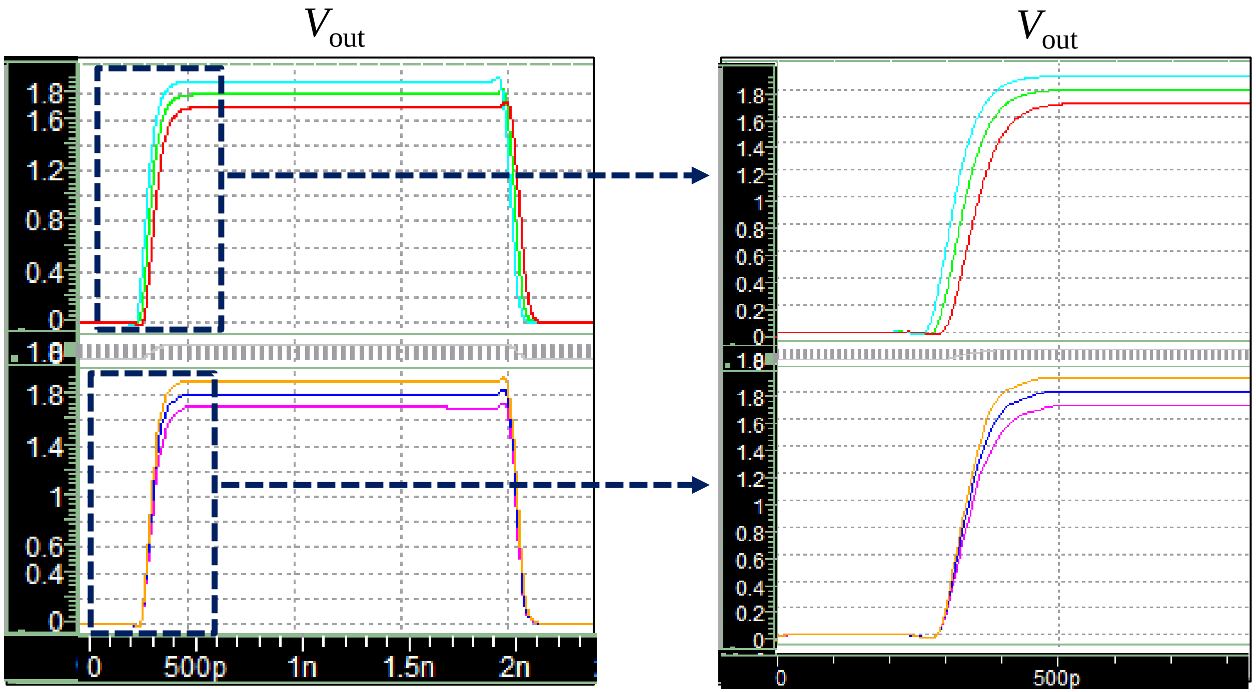
- The most of PSIJ is caused by the internal buffer delay rather than the last stage buffer
- Current IBIS cannot account for the delay change happened in the internal buffer



- Example: an inverter chain output
- VDD: 1.7/1.8/1.9V

Spice Results

Power-aware IBIS model Results
(ver5.1,generated with an EDA tool)



Power-Aware IBIS Model

Power-aware IBIS model considers gate modulation effect, ratio modification on Ku, Kd based on power rail voltage value

Gate Modulation Coefficients

The ST "Gate Modulation" solution is based on the introduction of two coefficients, one for the Pullup and one for the Pulldown stage, which modulate properly the IBIS standard current (I_IBIS-STD) when a bouncing noise occurs on the power and ground nodes

I(Vgs, Vds) = Kssn(Vgs,Vds)*I(Vgs=VDD,Vds)

Effective SPICE current — IBIS standard current

I_effective = Kssn(Vgs,Vds)*I_IBIS-STD

$$K_d(t) I_{pd} -> K_{sspd}(V_{pd}) K_d(t) I_{pd}$$

$$K_u(t) I_{pu} -> K_{sspu}(V_{pu}) K_u(t) I_{pu}$$

$$K_{sspd}(V_{pd}) = \frac{V_{pd}}{I_{sspd}(0)}$$

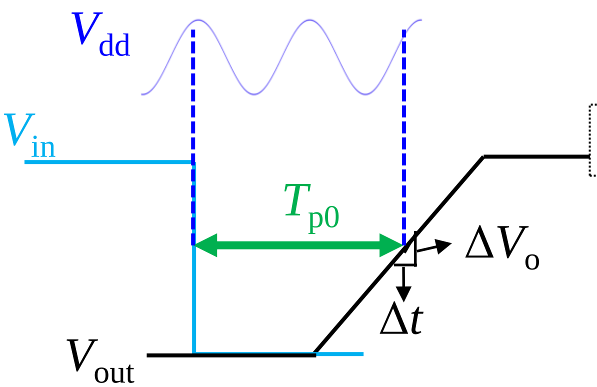
$$K_{sspu}(V_{pu}) = \frac{V_{pu}}{I_{sspu}(0)}$$

Source: "BIRD 98 and ST 'Gate Modulation' Convergence", IBIS Open Forum Teleconference, Jan. 27th, 2007

The ratio modification Ksspd, Ksspu on Ku, Kd is only a function of V_{pd} or V_{pu}, it cannot reflect the effect of power rail voltage noise during the propagation delay

PSIJ Mechanism

Averaged power supply noise effect during T_{p0}



Instantaneous timing variation

$$\frac{PSRR(\omega)\cdot e^{j\omega t}}{Slope}$$

Time averaged

$$\text{Jitter Sensitivity}(\omega)=\frac{1}{T_{p0}}\int_0^{T_{p0}}\frac{PSRR(\omega)\cdot e^{j\omega t}}{Slope}dt$$

$$=\int_0^{T_{p0}}\frac{T_{pd\max}-T_{pd\min}}{V_{dd\max}-V_{dd\min}}PSRR'(\omega)\cdot\frac{e^{j\omega t}}{T_{p0}}dt$$

①　②　③

[1] $$\text{Jitter Sensitivity}(\omega)=\frac{T_{pd\max}-T_{pd\min}}{V_{dd\max}-V_{dd\min}}\,PSRR'(\omega)\,e^{j\pi fT_{p0}}\,\mathrm{sinc}\left(\pi fT_{p0}\right)$$

Underlaying mechanism

① jitter sensitivity @ DC

② frequency dependency due to PSRR

③ time averaged power noise

IBIS implementation perspective

① propagation delay min/typ/max variations

② assuming frequency invariant (limitation)

③ time averaging term in Ku & Kd

[1] Y. Sun, J. Lee, and C. Hwang, “A Generalized Power Supply Induced Jitter Sensitivity Analysis Method Based on Power Supply Rejection Ratio Response” submitted to *IEEE Trans. on Very Large Scale Integration Systems* (under review).

Ku(t) & Kd(t) Modification (An Example)

Modify Ku(t), Kd(t) as a function of **time averaged** power rail voltage Vcc(t); introduce correction coefficient B and A as a function of **time**

$$K_u(t) = K_{u0}(t) + B_u(t) \cdot \left[\frac{\int_0^{T_{switch}} V_{cc}(t)}{T_{switch}} - V_{cc0} \right] + A_u(t) \left[\frac{\int_0^{T_{switch}} V_{cc}(t)}{T_{switch}} - V_{cc0} \right]^2$$

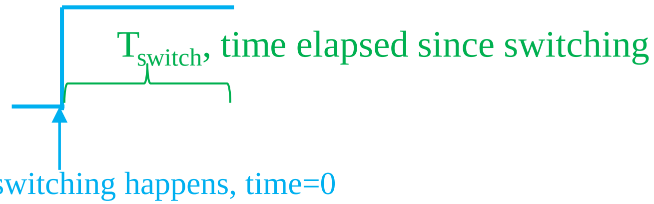
$$K_d(t) = K_{d0}(t) + B_d(t) \cdot \left[\frac{\int_0^{T_{switch}} V_{cc}(t)}{T_{switch}} - V_{cc0} \right] + A_d(t) \left[\frac{\int_0^{T_{switch}} V_{cc}(t)}{T_{switch}} - V_{cc0} \right]^2$$

Ku, Kd under nominal Vcc (typical)

Linear fitting coefficient

Quadratic fitting coefficient

Averaged Vcc(t) after the switching event happens;



Previous method on modification of Ku, Kd does not consider the time averaged effect;

Source: Behavioral modeling of jitter due to power supply noise for input/output buffers (US Patent 9842177B1)

Coefficients Extraction

Extraction of Au(t) and Bu(t) from Ku(t) under different Vcc

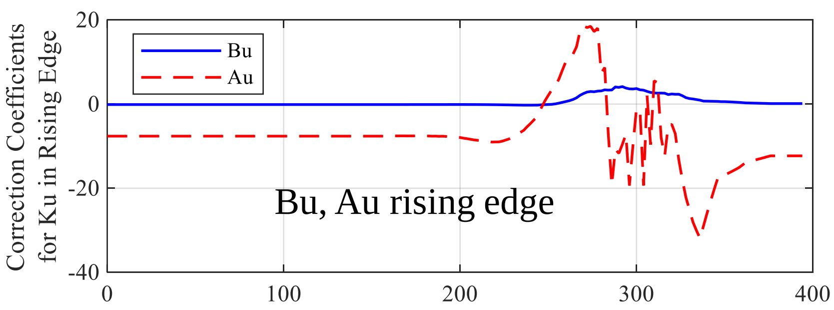
$$K_{u\_\max}(t) = K_{u0}(t) + B_u(t)(V_{cc\_max} - V_{cc0}) + A_u(t)(V_{cc\_max} - V_{cc0})^2$$

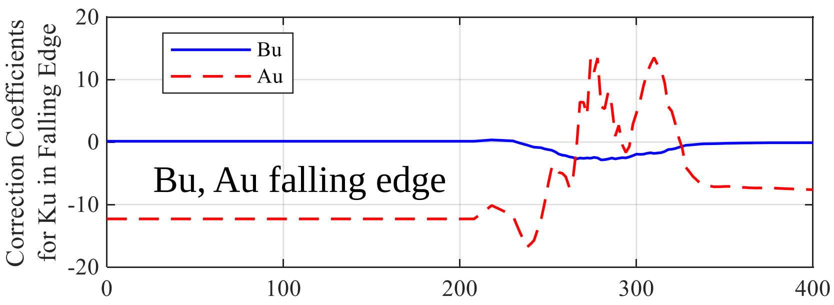
$$K_{u\_\min}(t) = K_{u0}(t) + B_u(t)(V_{cc\_\min} - V_{cc0}) + A_u(t)(V_{cc\_\min} - V_{cc0})^2$$

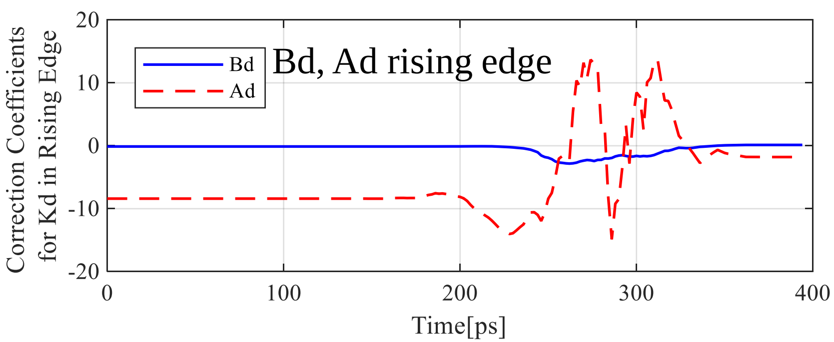
2 equations, 2 unknowns algorithm
=> Bu(t), Au(t)

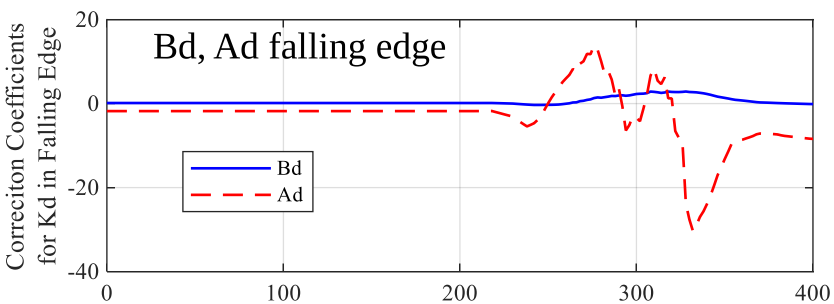
Bd(t), Ad(t) can be obtained similarly

V_{cc_max}=1.9V, V_{cc0}=1.8V, V_{cc_min}=1.7V



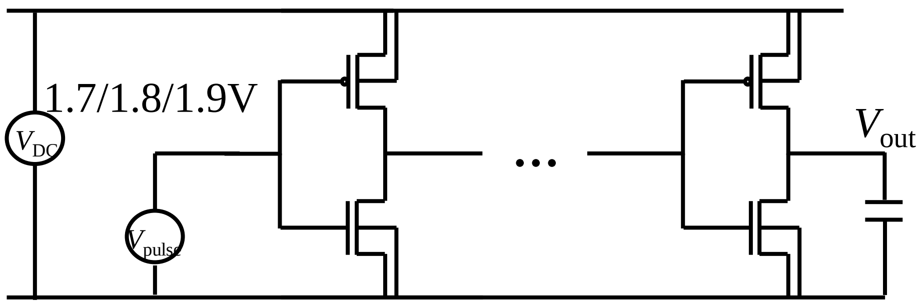


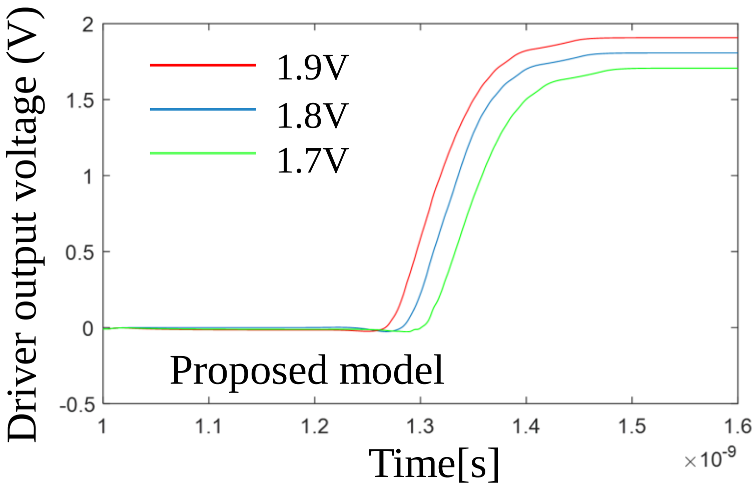


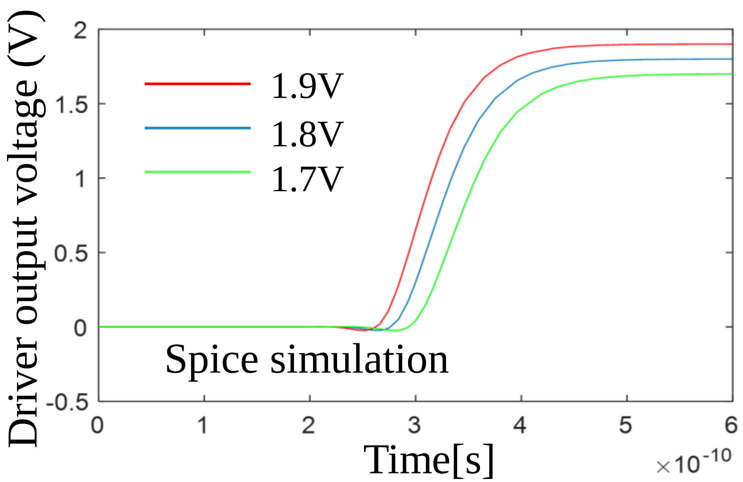


Model Validation – Case 1

1. Vcc 1.7/1.8/1.9V respectively

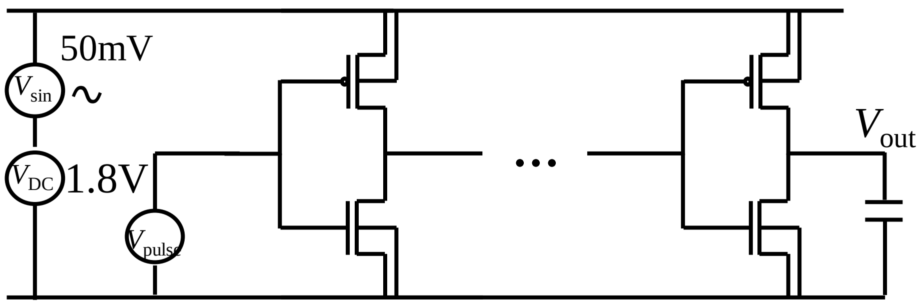


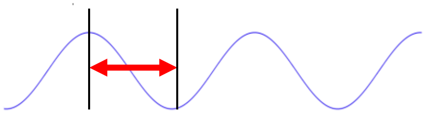




Model Validation – Case 2

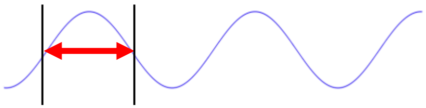
2. Vcc have very low frequency noise

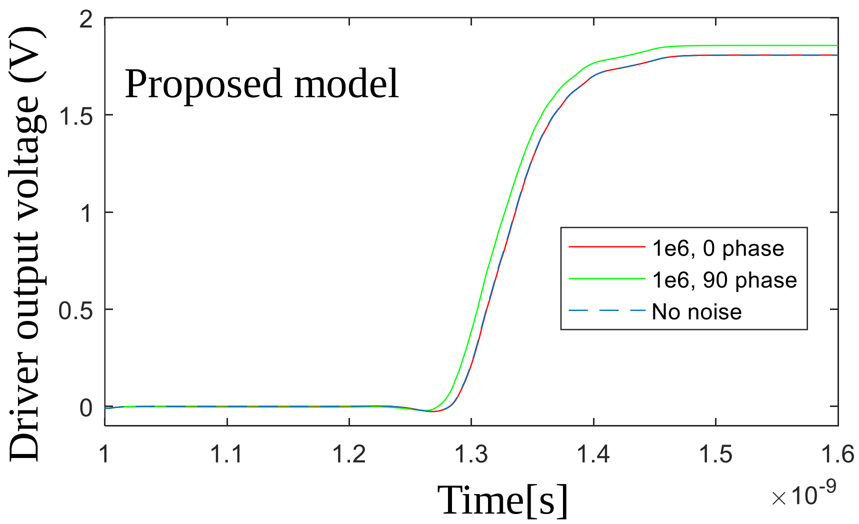


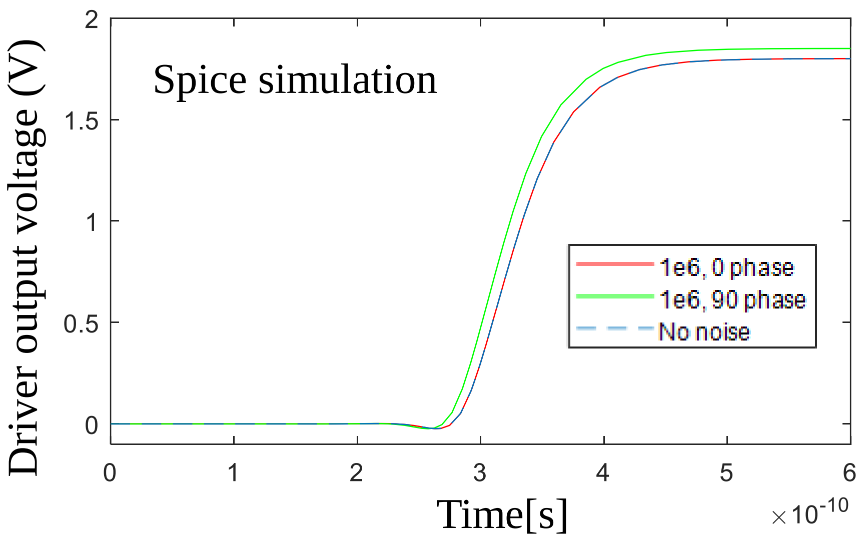


Vcc=1.8V+0.05*sin(2*pi*1e6)

Vcc=1.8V+0.05*sin(2*pi*1e6+pi/2)

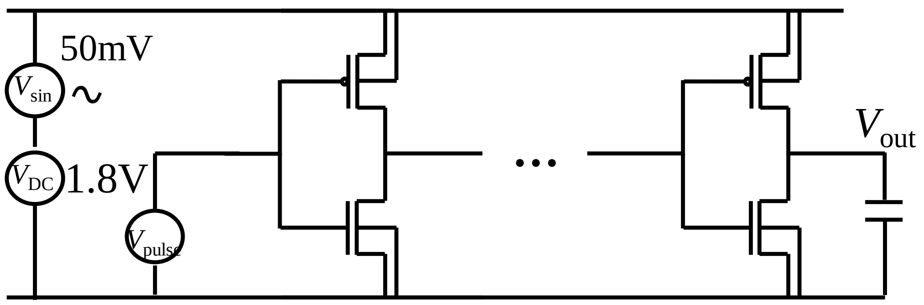






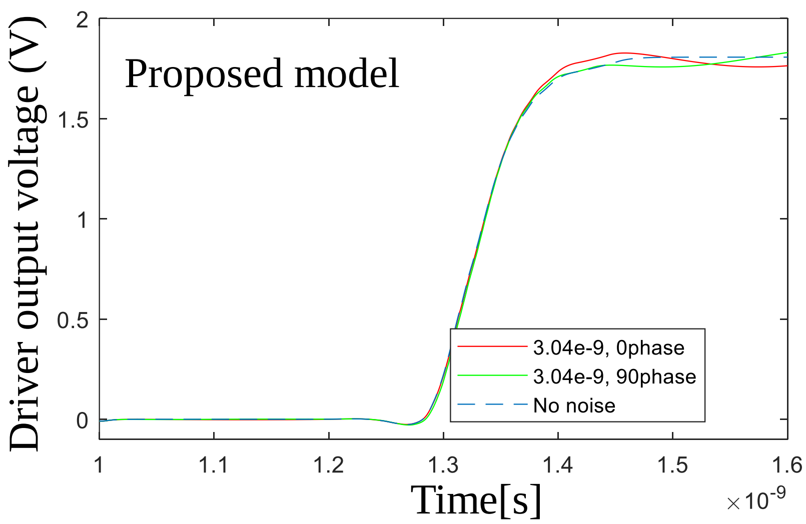
Model Validation – Case 3

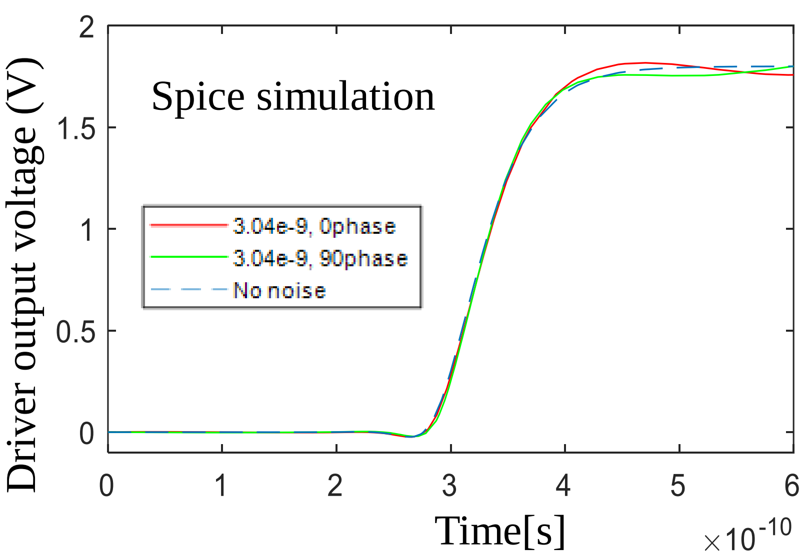
3. Vcc have noise with frequency corresponds to propagation delay (329ps)



Vcc=1.8V+0.05*sin(2*pi*3.04e9)

Vcc=1.8V+0.05*sin(2*pi*3.04e9+pi/2)





BIRD

Key issue: how to include propagation delay min/typ/max variations

1) Include a specific (and meaningful) time 0 in waveforms: [Rising Waveform] [Falling Waveforms]

- BIRD 68.1 proposed a common timing reference for all waveforms to provide accurate duty cycles but didn’t state where the time reference should be set

- From Randy: we would need a way to tell the software to start the waveform tables at a specific time aligned with a time reference inside of the SPICE subcircuit. It might make it difficult for model makers to use existing model creation software.

2) Provide a list of delay values measured from a simulation: [Initial Delay] or a new keyword.