IBIS Modeling Using Latency Insertion Method (LIM)

José E. Schutt-Ainé

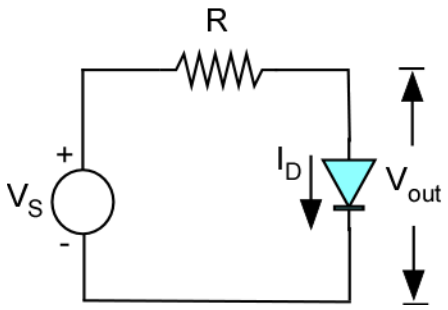
University of Illinois at Urbana-Champaign

Jilin Tan, Ping Liu, Feras Al-Hawari, Ambrish Varma

Cadence Design Systems

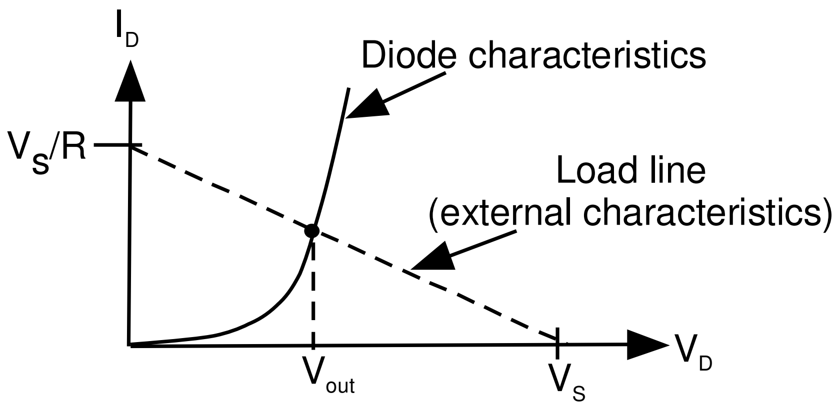
16th IEEE Workshop on Signal and Power Integrity
May, 2012
Sorrento, Italy

Nonlinear Circuit



How do we solve a simple diode circuit problem?

Graphical method…



… solve transcendental equations

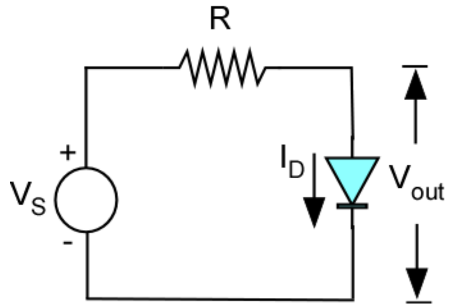
$$V_{out} = V_D$$

$$I_D = I_S\left(e^{V_D/V_T} - 1\right)$$

$$V_S = RI_D + V_D = RI_D(V_D) + V_D$$

Diode Circuit – Iterative Method

… or use the Newton-Raphson method…



Use: $x_{k+1} = x_k - \left[f'(x_k) \right]^{-1} f(x_k)$

$$V_{out} = V_D$$

$$x^{(k+1)} = x^{(k)} - \left[f'(x^{(k)}) \right]^{-1} f(x^{(k)})$$

$$f(V_D) = \frac{V_D - V_S}{R} + I_S \left(e^{V_D/V_T} - 1 \right) = 0 \qquad f'(V_D) = \frac{1}{R} + \frac{I_S}{V_T} e^{V_D/V_T}$$

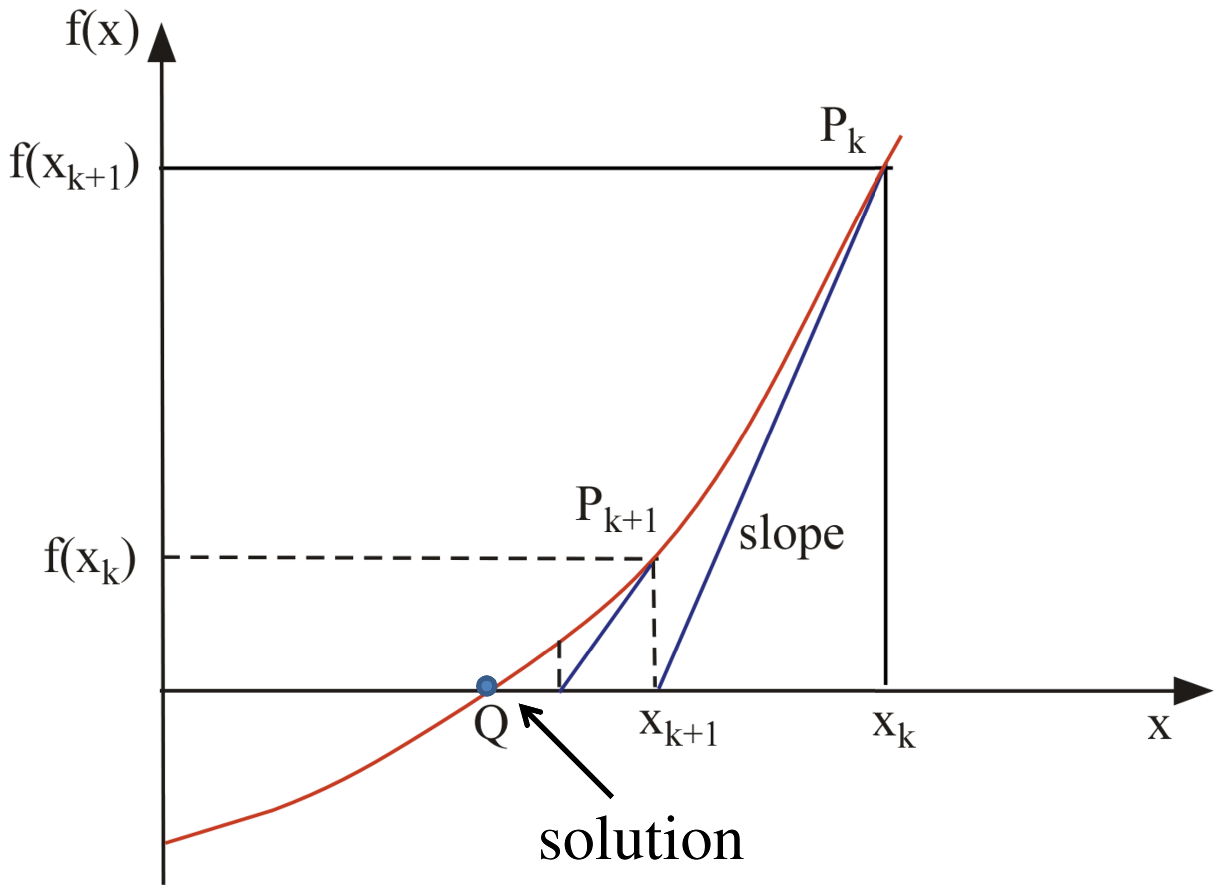
$$V_D^{(k+1)} = V_D^{(k)} - \frac{\dfrac{V_D^{(k)} - V_S}{R} + I_S \left(e^{V_D^{(k)}/V_T} - 1 \right)}{\dfrac{1}{R} + \dfrac{I_S}{V_T} e^{V_D^{(k)}/V_T}}$$

Where $V_D^{(k)}$ is the value of V_D at the kth iteration

Procedure is repeated until convergence to final (true) value of V_D which is the solution. Rate of convergence is quadratic.

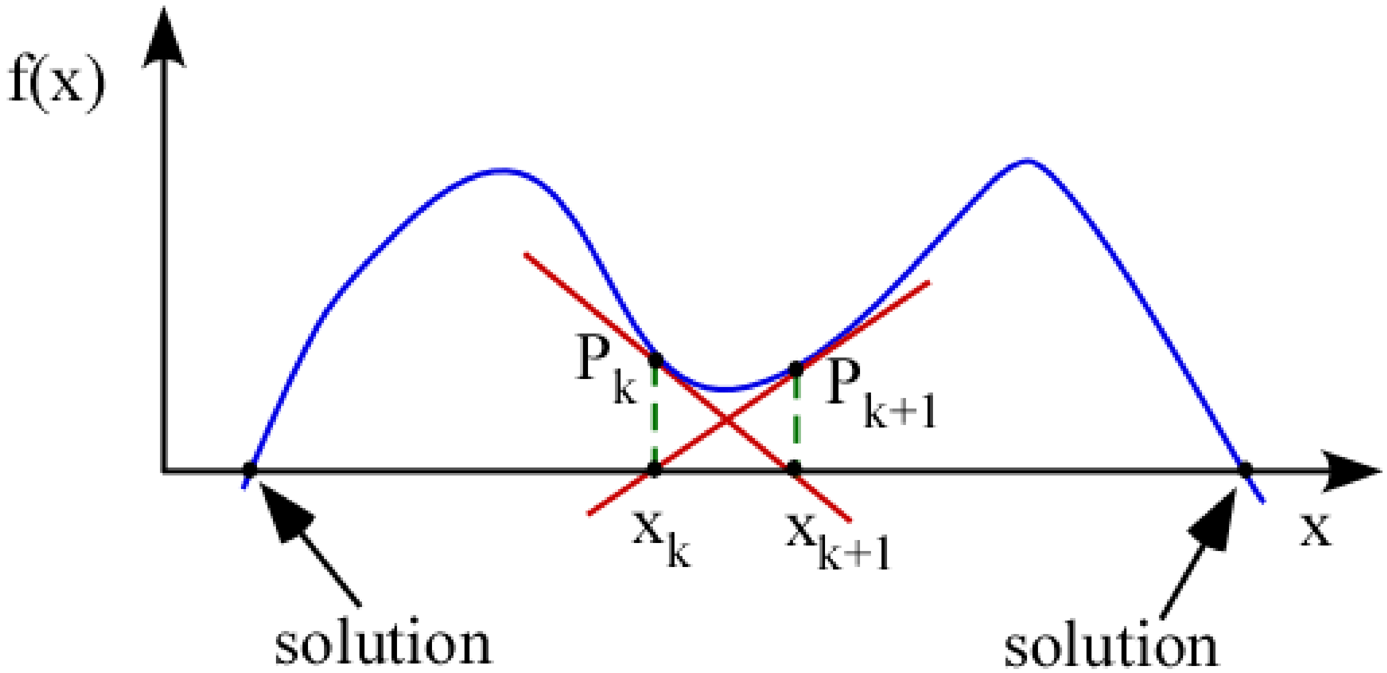
Newton Raphson Method

(Graphical Interpretation)



Newton Raphson Method

If initial guess is not close enough, NR will lock into oscillations and solution will not converge



Limitations

- IBIS data can be unpredictable
- Transient response requires solution of nonlinear system
- Most simulators use Newton-Raphson (NR) technique combined with modified nodal analysis (MNA)
- NR may not converge
- NR may slow down simulation

Why LIM?

- LIM does not iterate on nonlinear problems
- There is no convergence issue
- MNA has <u>super-linear</u> numerical complexity
- LIM has <u>linear</u> numerical complexity
- LIM uses no matrix formulation
- LIM has no matrix ill-conditioning problems
- LIM is much faster than MNA for large circuits

Latency Insertion Method**

Each branch must have an inductor*

Each node must have a shunt capacitor*

Express branch current in terms of <u>history</u> of adjacent node voltages

Express node voltage in terms of <u>history</u> of adjacent branch currents

**** If branch or node has no inductor or capacitor, insert one with very small value***

** J. E. Schutt-Ainé, "Latency Insertion Method for the Fast Transient Simulation of Large Networks," IEEE Trans. Circuit Syst., vol. 48, pp. 81-89, January 2001.

LIM: Leapfrog Method

$$
\begin{tikzcd}[column sep=small]
I_{ij}^{n} \arrow[dr, red] & & I_{ij}^{n+1} \arrow[dr, red] & & I_{ij}^{n+2} \arrow[dr, red] & \\
& V_i^{n+1/2} \arrow[ur, blue] & & V_i^{n+3/2} \arrow[ur, blue] & & V_i^{n+5/2}
\end{tikzcd}
$$

$\longrightarrow$ *n: time*

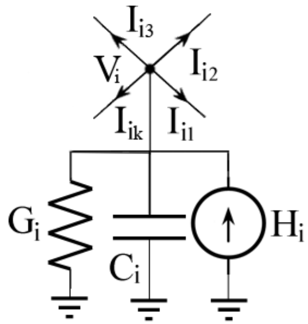
$$V_i^{n+1/2} = \frac{\dfrac{C_i V_i^{n-1/2}}{\Delta t} + H_i^n - \sum_{k=1}^{N_a} I_{ik}^n}{\dfrac{C_i}{\Delta t} + G_i}$$

$$I_{ij}^{n+1} = I_{ij}^n + \frac{\Delta t}{L_{ij}}\left(V_i^{n+1/2} - V_j^{n+1/2} - R_{ij} I_{ij}^n\right)$$

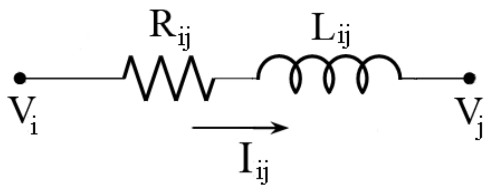
Leapfrog method achieves second-order accuracy, i.e., error is proportional to Δt^2

LIM Algorithm

- **Represents network as a grid of nodes and branches**



Node structure



Branch structure

- **Discretizes Kirchhoff's current and voltage equations**

$$V_i^{n+1/2} = \frac{\frac{C_i V_i^{n-1/2}}{\Delta t} + H_i^n - \sum_{k=1}^{N_a} I_{ik}^n}{\frac{C_i}{\Delta t} + G_i} \qquad I_{ij}^{n+1} = I_{ij}^n + \frac{\Delta t}{L_{ij}}\left(V_i^{n+1/2} - V_j^{n+1/2} - R_{ij} I_{ij}^n\right)$$

- **Uses "leapfrog" scheme to solve for node voltages and branch currents**
- **Presence of reactive elements is required to generate latency**

LIM is Easy to Code…

time loop

For time=1, N_t

branch loop

For branch =1, N_b

Update current as per Equation:

$$I_{ij}^{n+1} = I_{ij}^{n} + \frac{\Delta t}{L_{ij}}\left(V_i^{n+1/2} - V_j^{n+1/2} - R_{ij} I_{ij}^{n} + E_{ij}^{n+1/2}\right)$$

Next branch;

node loop

For node=1, N_n

Update voltage as per Equation

$$V_i^{n+1/2} = \frac{\dfrac{C_i V_i^{n-1/2}}{\Delta t} + H_i^n - \sum_{k=1}^{M_i} I_{ik}^n - \sum_{q=1}^{N} J_q^n}{\dfrac{C_i}{\Delta t} + G_i}$$

Next node;

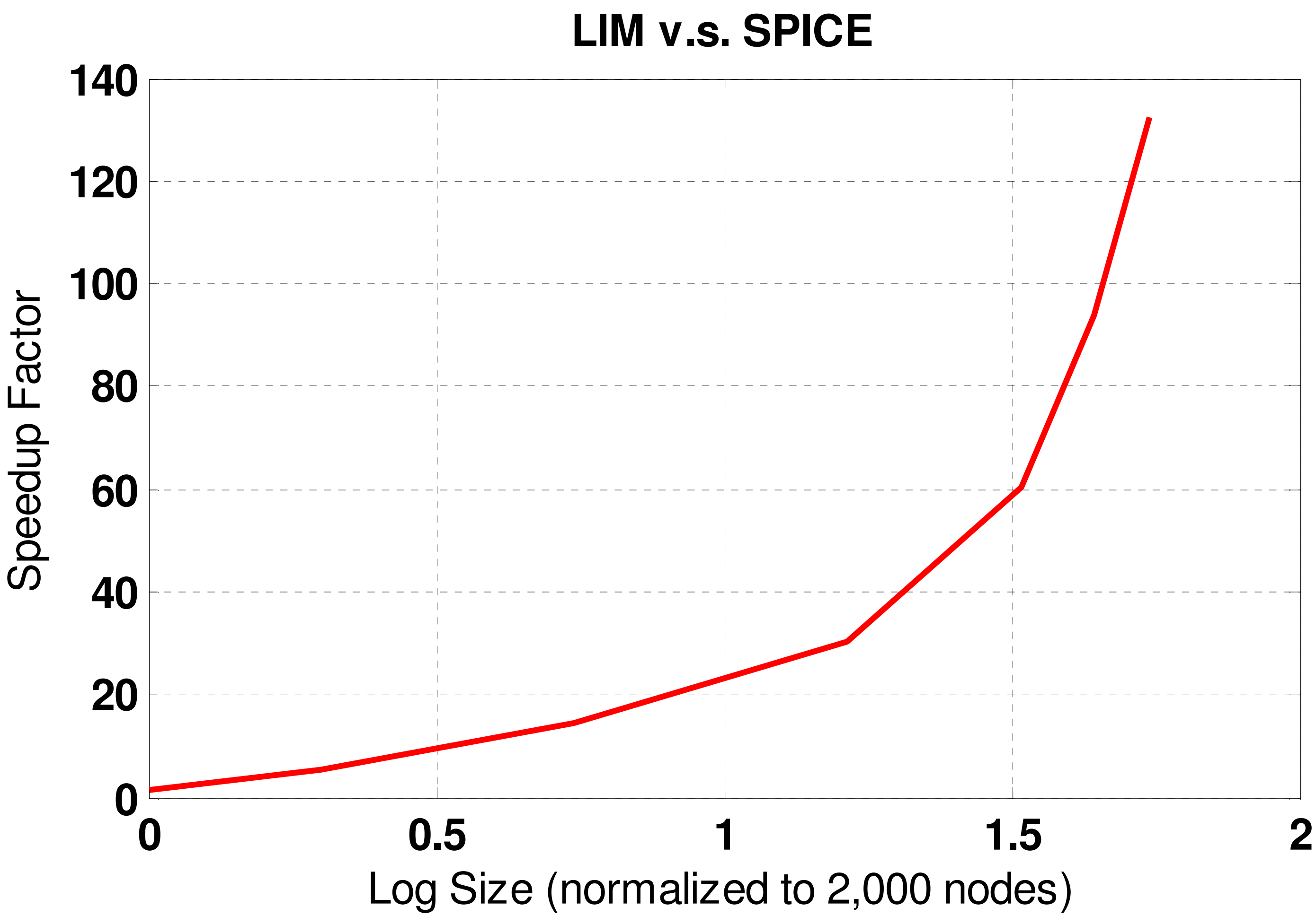
Next time;

LIM is Fast…

Comparison of runtime for LIM and SPICE per 100 time steps.

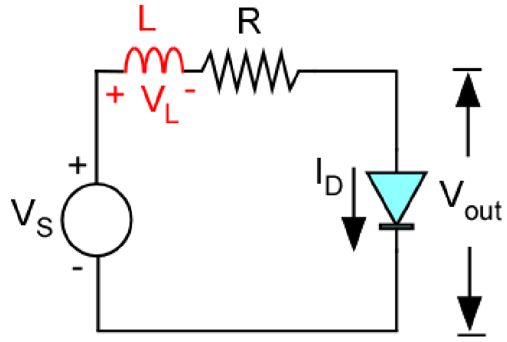
Circuit	# nodes	# MOSFET	SPICE	LIM
ADDER	36	62	0.0058 s	0.0005 s
VOTER	1709	4243	0.369 s	0.041 s
RAM CKT	4850	13880	1.94 s	0.184 s

… and gets faster as circuit size increases



LIM has NO Convergence Issues

Introduce latency in diode circuit through a small inductor L



… then use colocation and leapfrog:

$$V_D \rightarrow V_D^n, V_D^{n+1}, V_D^{n+2}, \ldots$$

$$I_D \rightarrow I_D^{n-1/2}, I_D^{n+1/2}, I_D^{n+3/2}, \ldots$$

with $V_L^n = L \dfrac{I^{n+1/2} - I^{n-1/2}}{\Delta t}$

…and if time steps are sufficiently small,

Current solution

$$I_D^{n+1/2} = I_S \left(e^{V_D^n / V_T} - 1 \right)$$

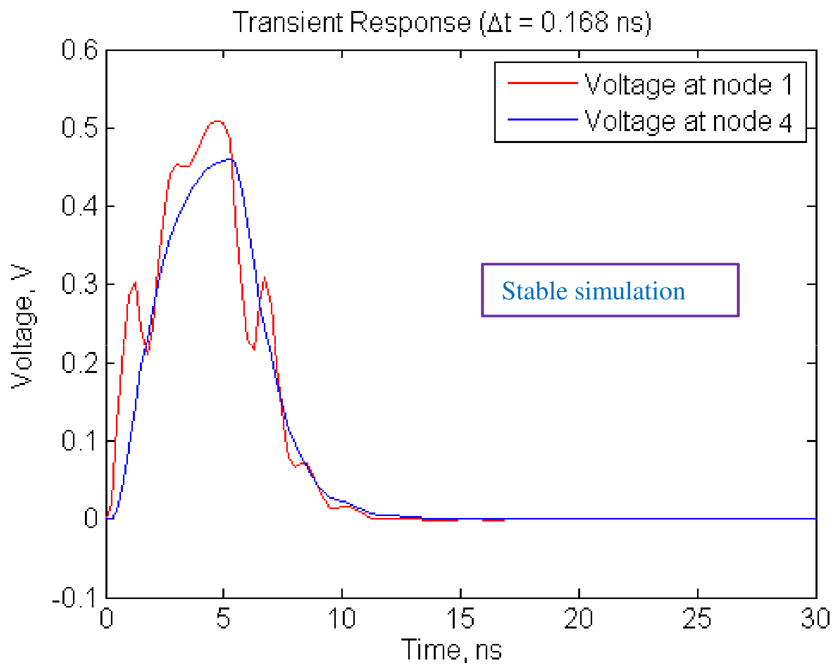
or

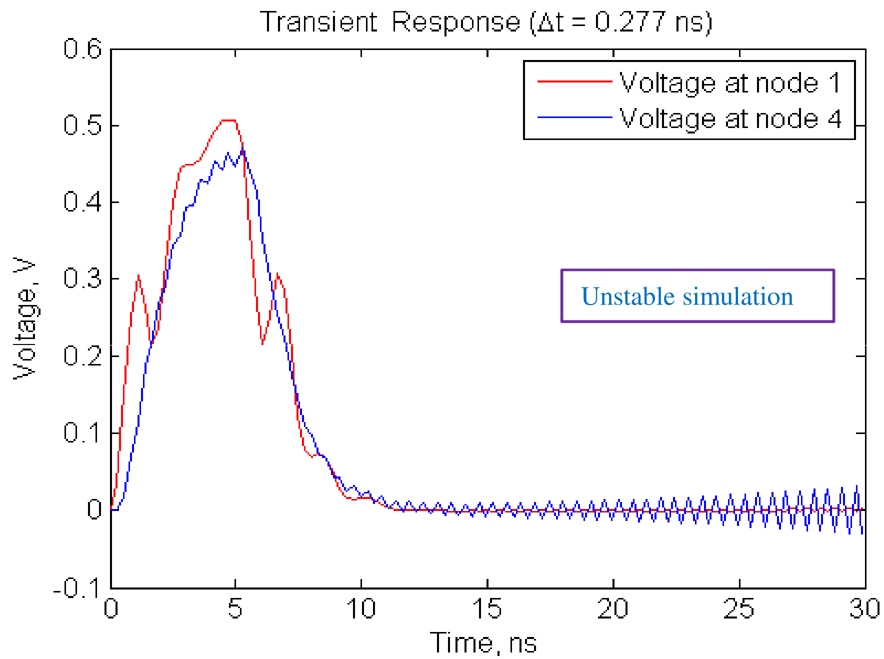
Voltage solution

$$V_D^{n+1} = V_T \ln \left(\frac{I_D^{n+1/2}}{I_S} + 1 \right)$$

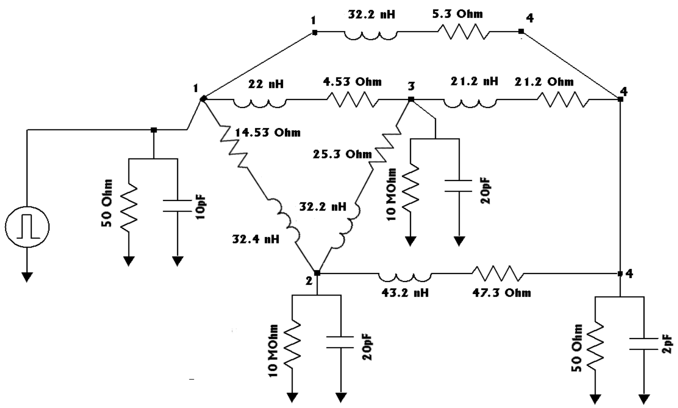
Explicit !

LIM Suffers from Stability Issues …





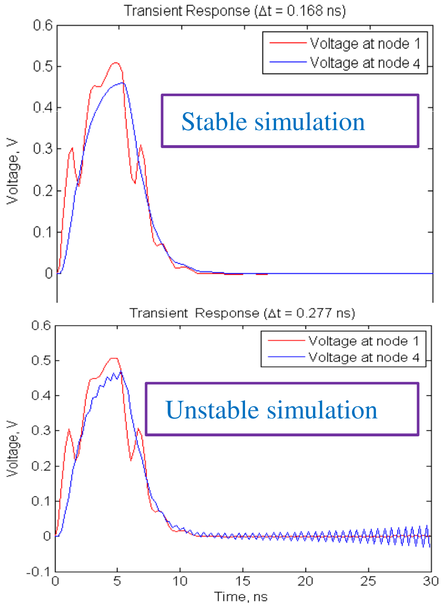
...but they can be controlled...

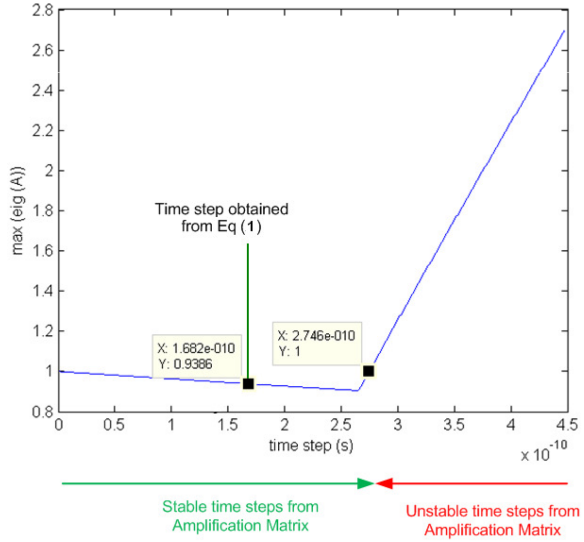


- The LIM algorithm is conditionally stable
- No faster than speed of light propagation

- The <u>Amplification Matrix</u> [5]
  - Eigenvalues must be less than unity for the system to be stable
  - A-matrix provides the <u>exact limit</u> on the size of the time step

$$\boxed{\begin{bmatrix} \mathbf{v}^{n+1/2} \\ \mathbf{i}^{n+1} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{v}^{n-1/2} \\ \mathbf{i}^{n} \end{bmatrix}}$$





- The <u>Energy Method</u> [6]

$$\boxed{\Delta t \le \sqrt{2}\,\min_{i=1}^{N_n}\left(\sqrt{\frac{C_i}{N_b^i}\min_{p=1}^{N_b^i}(L_{i,p})}\right)}$$

  - Smallest LC product in the circuit
  - Provides a <u>sufficient</u> condition on Δt
  - Easy to use in our simulations

Application of LIM to IBIS

- IBIS Data Processing
- Ku/Kd Extraction
- IBIS Standard Formulation
- LIM-IBIS Formulation
- LIM-IBIS Solutions
- Extension to Gate Modulation Effects
- Conclusion and Future Work

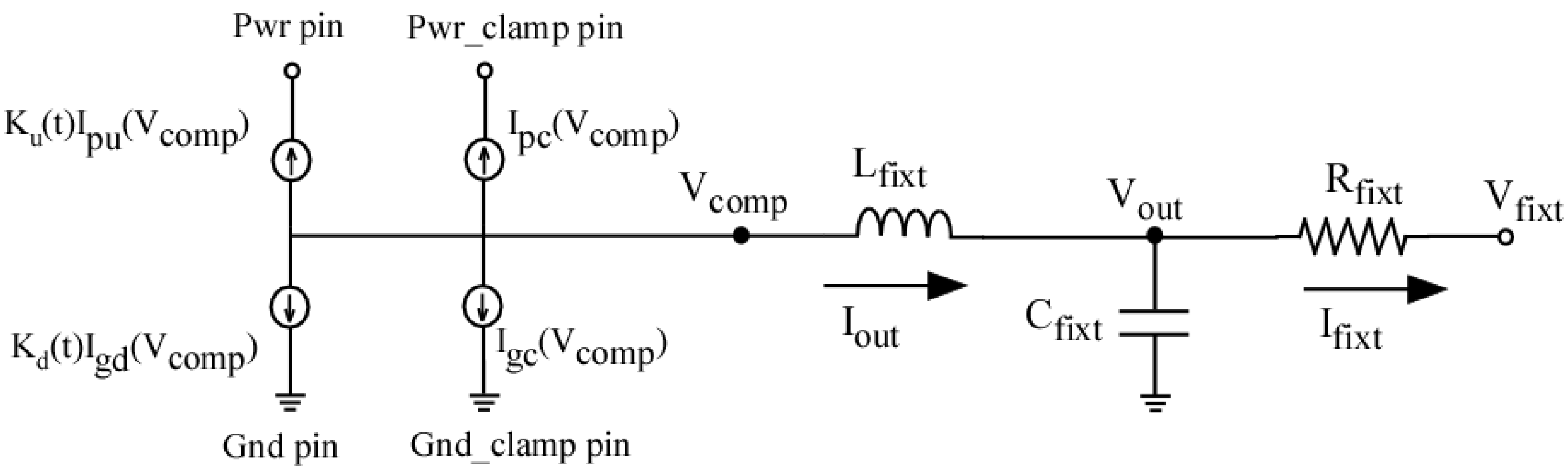
IBIS Data Processing

1. Arrange static IV data
2. Pulldown data (current vs voltage) ➔ I_{pd}, m_{pd} points
3. Pullup data (current vs voltage) ➔ I_{pu}, m_{pu} points
4. Ground clamp data (current vs voltage) ➔ I_{gc} , m_{gc} points
5. Power clamp data (current vs voltage) ➔ I_{pc} , m_{pc} points

IBIS Data Processing

- Next Get VT data. VT data is presented as: Rising waveform:
- Voltage versus time for V_{fix} low → V_{R1} , m_{r1} points
- Voltage versus time for V_{fix} high → V_{R2} , m_{r2} points: Falling waveform:
- Voltage versus time for V_{fix} low → V_{F1} , m_{f1} points
- Voltage versus time for V_{fix} high → V_{F2} , m_{f2} points

IBIS Ku/Kd Extraction*



$$I_{fixt} = \frac{V_{out} - V_{fixt}}{R_{fixt}}$$

$$I_{cap} = C_{fixt} \frac{V_{out}(t) - V_{out}(t - \Delta t)}{\Delta t}$$

$$V_{comp} = L_{fixt} \frac{\Delta I_{out}}{\Delta t} + V_{out}$$

$$I_{out} = I_{cap} + I_{fixt}$$

Ku/Kd Extraction

We need to extract K_u and K_d

Procedure is well documented*

- **Pick value V_{comp1}**
- **Find closest corresponding currents in static IV data**
- **Set them as I_{pd1}, I_{pu1}, I_{gc1} and I_{pc1}**
- **Pick value V_{comp2}**
- **Find closest corresponding currents in static IV data**
- **Set them as I_{pd2}, I_{pu2}, I_{gc2} and I_{pc2}**

* Ying Wang, Han Ngee Tan "The Development of Analog SPICE Behavioral Model Based on IBIS Model", Proceedings of the Ninth Great Lakes Symposium on VLSI, GLS '99.

IBIS Circuit Analysis

2 equations, two unknown system

$$-I_{out1} = K_u I_{pu1} + K_d I_{pd1} + I_{pc1} + I_{gc1}$$

$$-I_{out2} = K_u I_{pu2} + K_d I_{pd2} + I_{pc2} + I_{gc2}$$

Rearrange as:

$$K_u I_{pu1} + K_d I_{pd1} = -I_{out1} - I_{pc1} - I_{gc1} = I_{RHS1}$$

$$K_u I_{pu2} + K_d I_{pd2} = -I_{out2} - I_{pc2} - I_{gc2} = I_{RHS2}$$

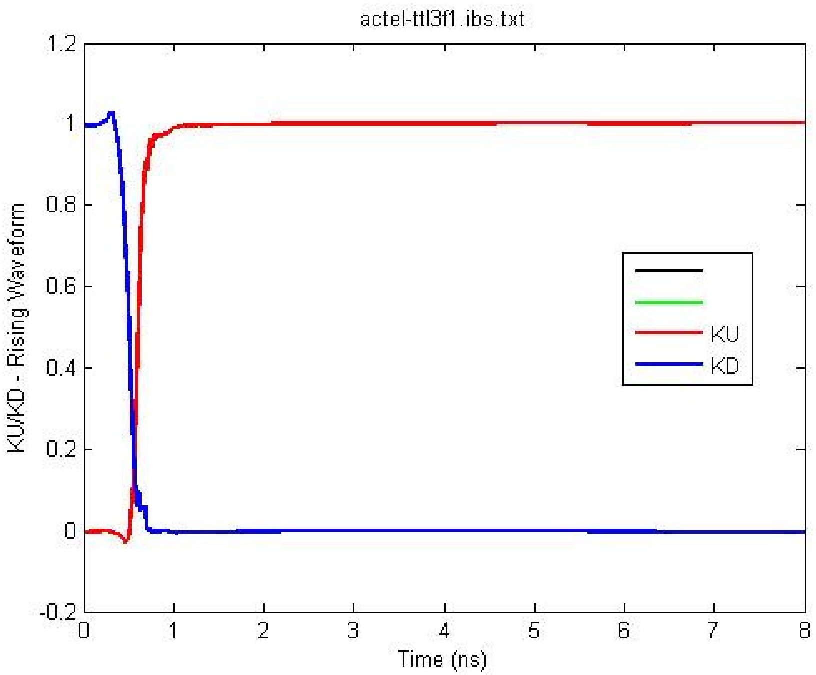
or

$$\begin{bmatrix} I_{pu1} & I_{pd1} \\ I_{pu2} & I_{pd2} \end{bmatrix} \begin{bmatrix} K_u \\ K_d \end{bmatrix} = \begin{bmatrix} I_{RHS1} \\ I_{RHS2} \end{bmatrix}$$

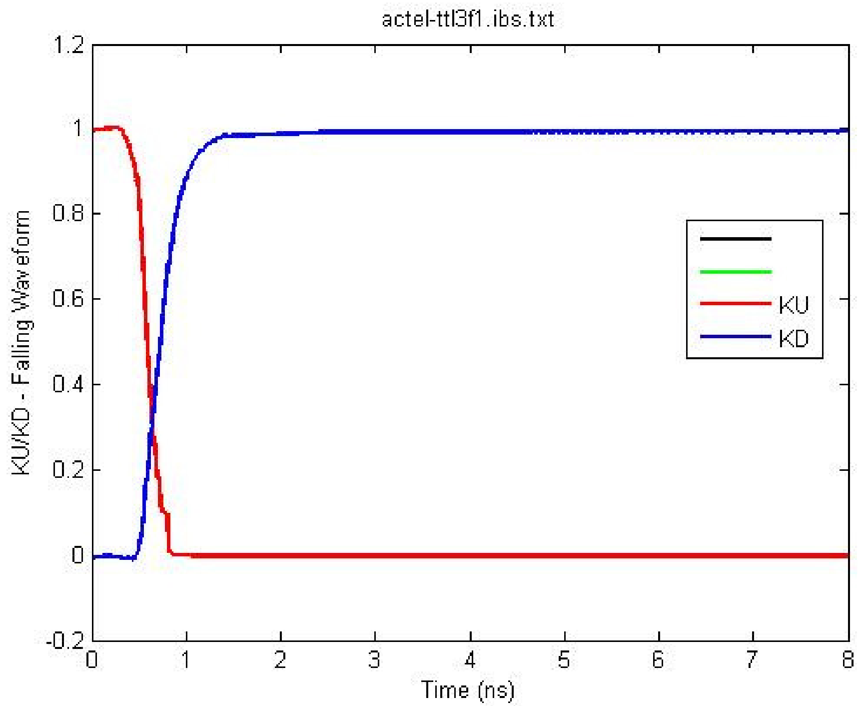
Solve for K_u and K_d

Example of Ku and Kd

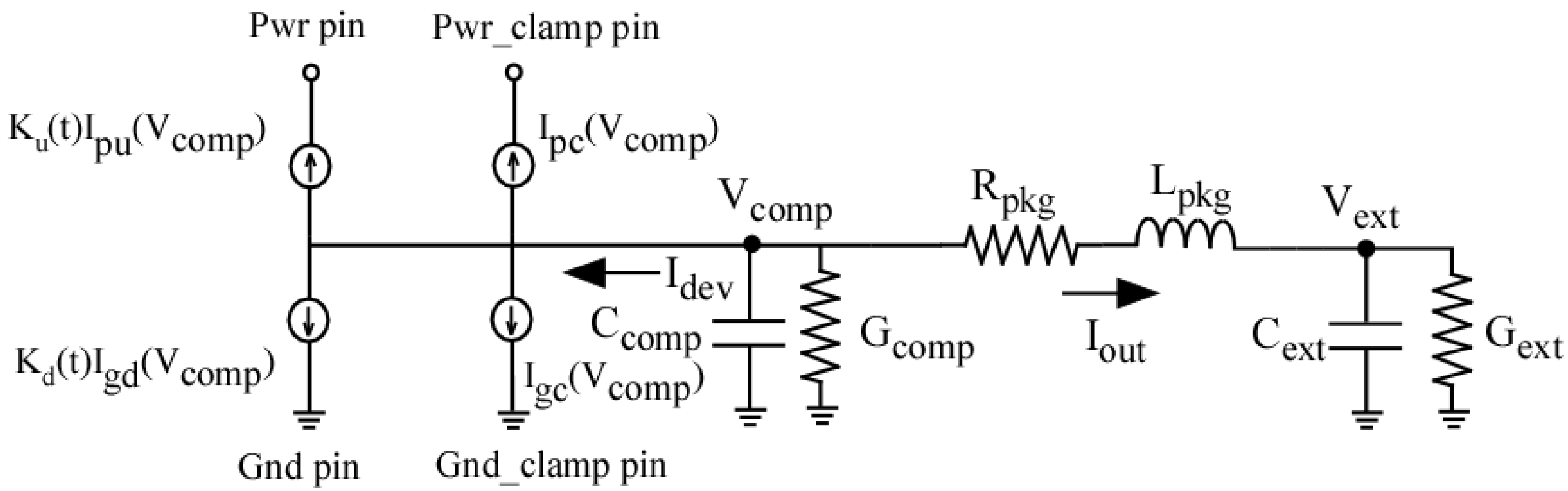
Rising Waveform



Falling Waveform



IBIS Simulations

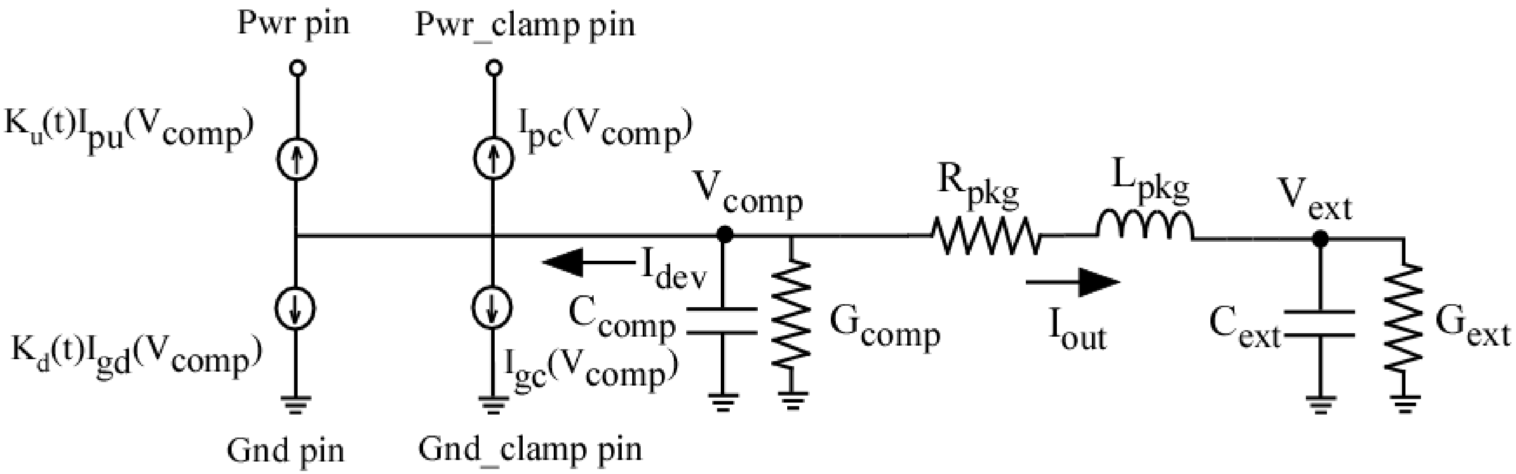


$$K_u I_{pu}\left(V_{comp}\right)+K_d I_{pd}\left(V_{comp}\right)+I_{pc}\left(V_{comp}\right)+I_{gc}\left(V_{comp}\right)$$

$$+I_{out}\left(V_{comp}\right)+I_{C_{comp}}\left(V_{comp}\right)+I_{G_{comp}}\left(V_{comp}\right)=0$$

Nonlinear system ➔ use Newton-Raphson

...or Better: Use a LIM Formulation



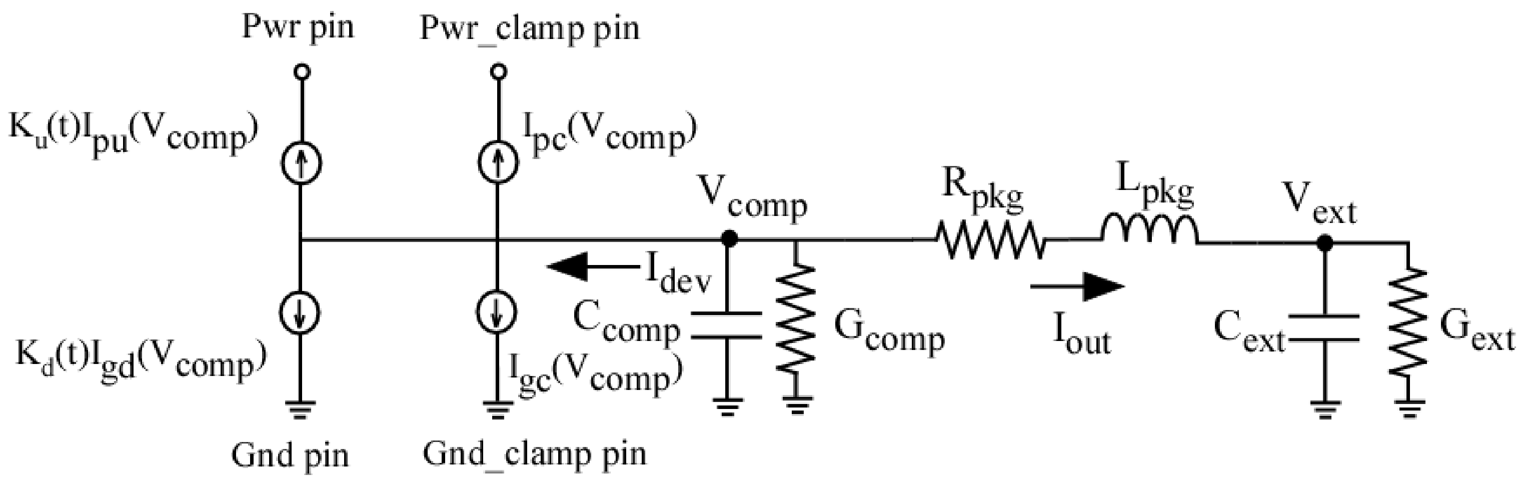
$$C_{ext}\frac{\left(V_{ext}^{n+1/2}-V_{ext}^{n-1/2}\right)}{\Delta t}+\frac{G_{ext}}{2}\left(V_{ext}^{n+1/2}+V_{ext}^{n-1/2}\right)=I_{out}^{n}$$

$$V_{ext}^{n+1/2}=\frac{I_{out}^{n}+\left(\frac{C_{ext}}{\Delta t}-\frac{G_{ext}}{2}\right)V_{ext}^{n-1/2}}{\left(\frac{C_{ext}}{\Delta t}+\frac{G_{ext}}{2}\right)}$$

$$V_{comp}^{n+1/2}-V_{ext}^{n+1/2}=L_{pkg}\frac{\left(I_{out}^{n+1}-I_{out}^{n}\right)}{\Delta t}+\frac{R_{pkg}}{2}\left(I_{out}^{n+1}+I_{out}^{n}\right)$$

$$I_{out}^{n+1}=\frac{\left(V_{comp}^{n+1/2}-V_{ext}^{n+1/2}\right)+I_{out}^{n}\left(\frac{L_{pkg}}{\Delta t}-\frac{R_{pkg}}{2}\right)}{\left(\frac{L_{pkg}}{\Delta t}+\frac{R_{pkg}}{2}\right)}$$

IBIS-LIM Solution



$$C_{comp}\frac{\left(V_{comp}^{n+1/2}-V_{comp}^{n-1/2}\right)}{\Delta t}+\frac{G_{comp}}{2}\left(V_{comp}^{n+1/2}+V_{comp}^{n-1/2}\right)=-I_{out}^{n}-I_{dev}^{n}$$

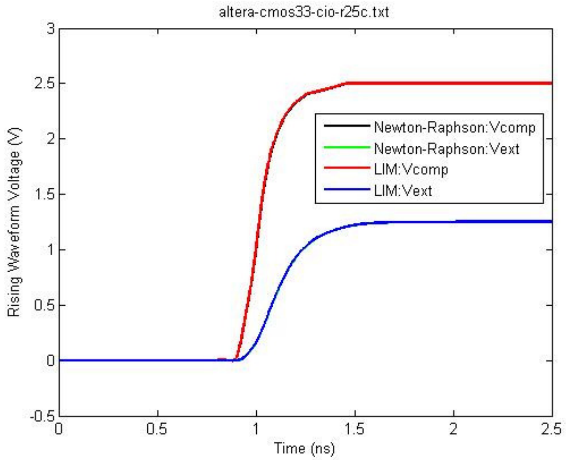
$$V_{comp}^{n+1/2}=\frac{-I_{out}^{n}-I_{dev}^{n}+\left(\frac{C_{comp}}{\Delta t}-\frac{G_{comp}}{2}\right)V_{comp}^{n-1/2}}{\left(\frac{C_{comp}}{\Delta t}+\frac{G_{comp}}{2}\right)}$$

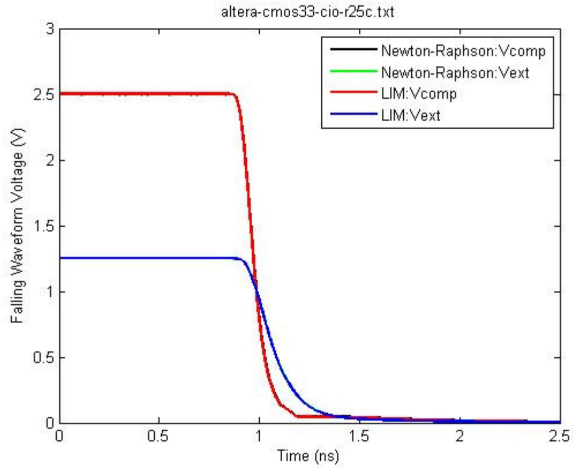
Explicit equations

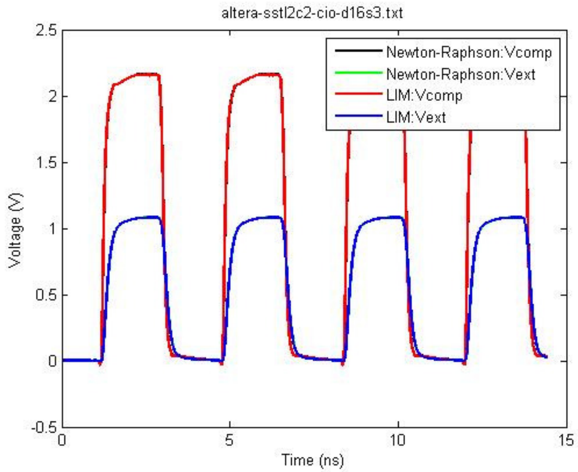
$$I_{dev}^{n}=K_u I_{pu}\left(V_{comp}\right)+K_d I_{pd}\left(V_{comp}\right)+I_{pc}\left(V_{comp}\right)+I_{gc}\left(V_{comp}\right)$$

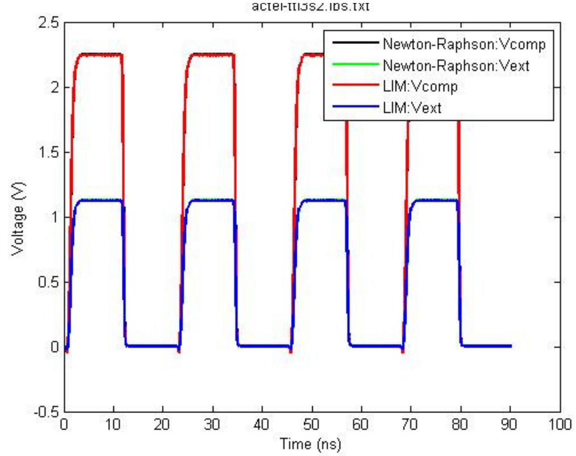
Transient Simulation Examples

NR and LIM give same results…





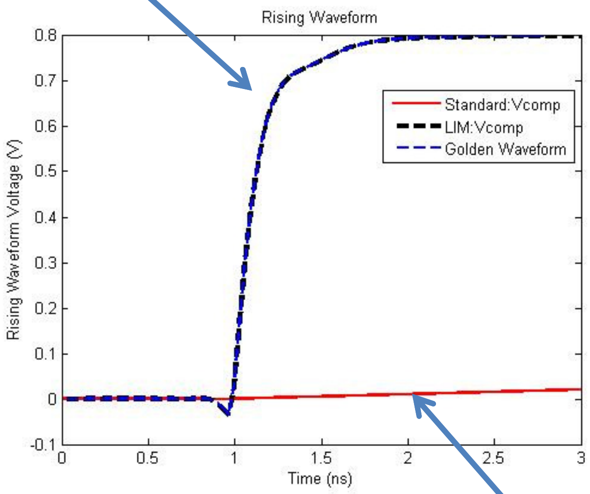




Transient Simulation Examples

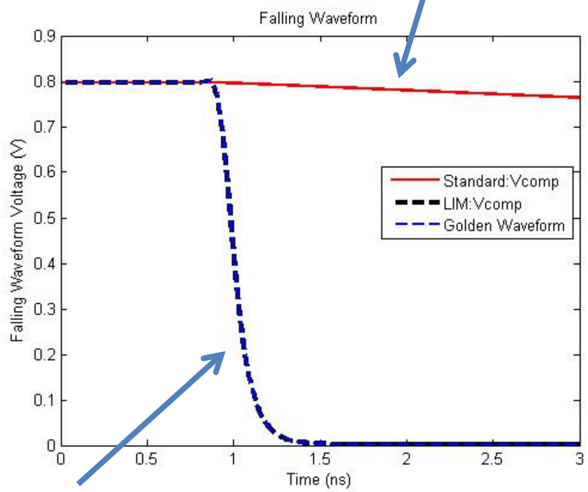
… in some cases Newton-Raphson fails to converge…

LIM



NR: failed convergence

NR: failed convergence



LIM

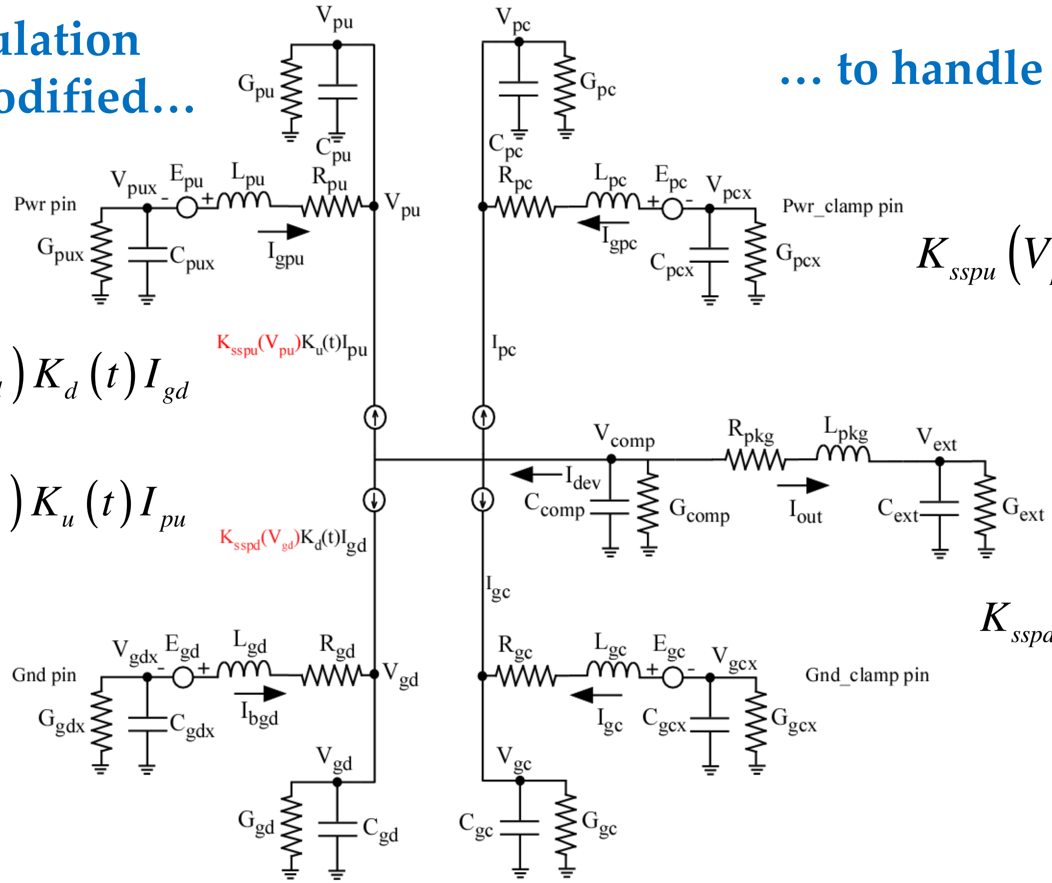
Handling Gate Modulation Effects (BIRD 98.3)

LIM-IBIS formulation can easily be modified…

… to handle SSN

$$K_d(t) I_{gd} \rightarrow K_{sspd}(V_{gd}) K_d(t) I_{gd}$$

$$K_u(t) I_{pu} \rightarrow K_{sspu}(V_{pu}) K_u(t) I_{pu}$$



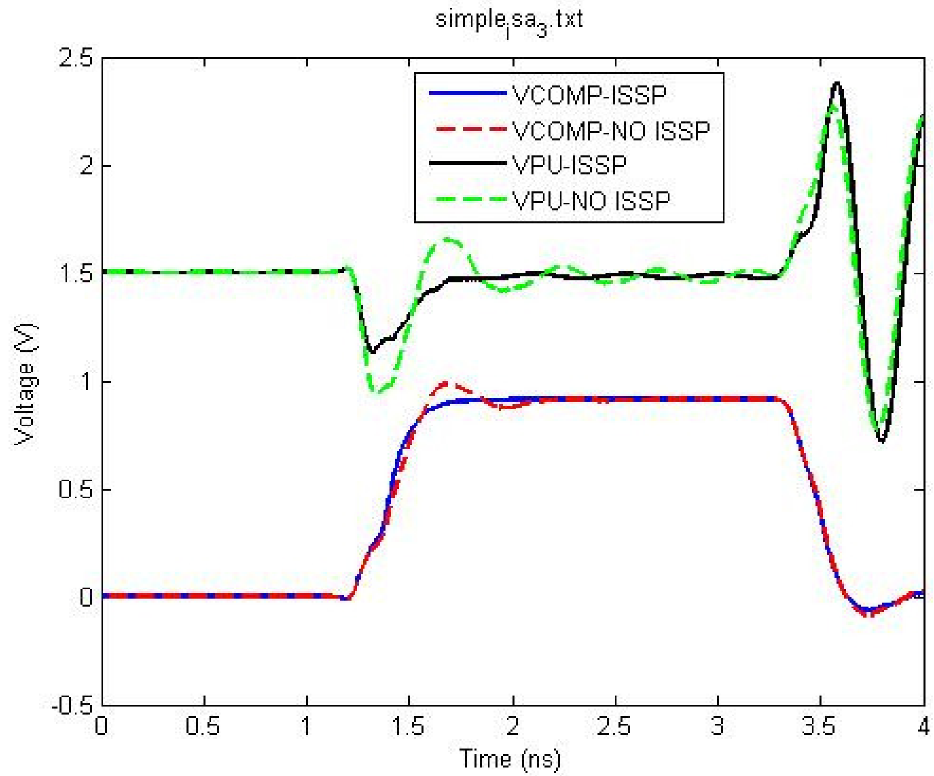
$$K_{sspu}(V_{pu}) = \frac{I_{sspu}(V_{pu})}{I_{sspu}(0)}$$

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

Gate Modulation Effects (BIRD 98.3)

Large power supply inductance Small decoupling capacitance

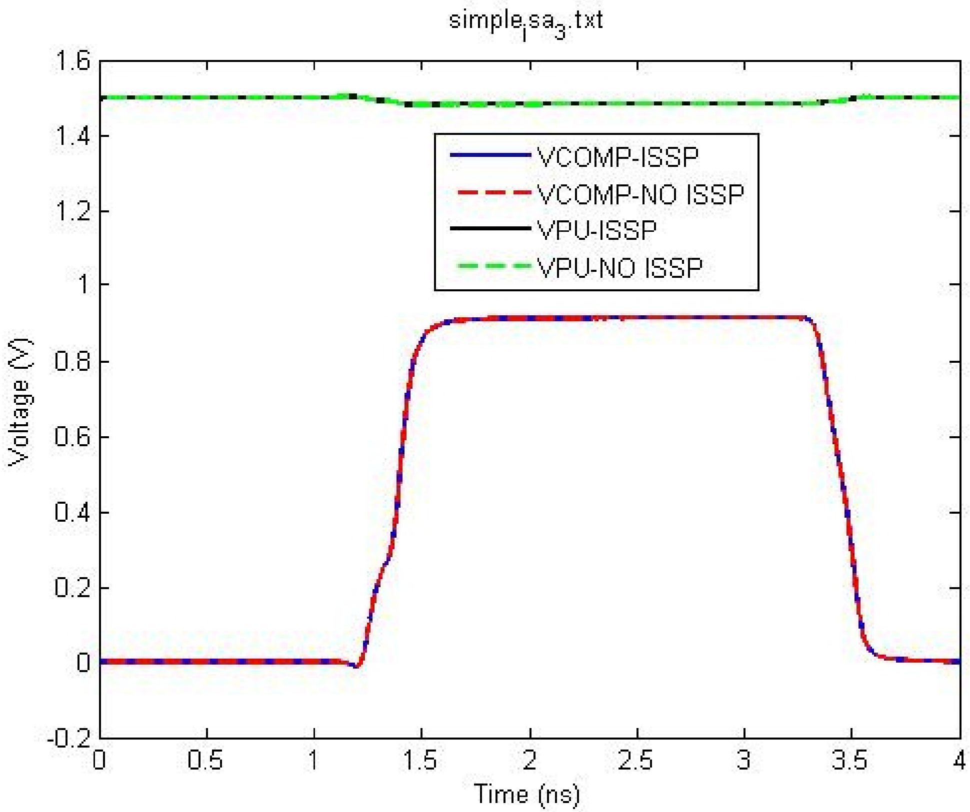
L_{pu} = 5 nH
C_{pu} = 0.001 nF



Gate Modulation Effects (BIRD 98.3)

Small power supply inductance Large decoupling capacitance

L_{pu} = 0.005 nH
C_{pu} = 0.1 nF



Conclusions

- We demonstrated that LIM can be used to simulate IBIS-based circuits with optimum accuracy.
- Because of the inserted latency, LIM does not use an iterative scheme to solve nonlinear equations and thus does not suffer from convergence problems
- LIM-based simulations were successful in instances where the traditional Newton-Raphson technique failed to provide a solution
- LIM is expected to be several orders of magnitude faster for large circuits containing a multitude of IBIS models.

References

[1] J. E. Schutt-Ainé, "Latency Insertion Method for the Fast Transient Simulation of Large Networks," IEEE Trans. Circuit Syst., vol. 48, pp. 81-89, January 2001.

[2] Dmitri Klokotov, and José Schutt-Ainé, “Transient Simulation of Lossy Interconnects using the Latency Insertion Method (LIM)”, Proceedings of the 17th Topical Meeting on Electrical Performance of Electronic Packaging (EPEP), pp. 251-254, San Jose, CA, October 2008.

[3] José Schutt-Ainé, "Stability Analysis of the Latency Insertion Method Using a Block Matrix Formulation", Proceedings of EDAPS-08, Seoul, South Korea, December 2008.

[4] J. Schutt-Aine, D. Klokotov, P. Goh, Jilin Tan, F. Al-Hawari, Ping Liu, Wenliang Dai, "Application of the latency insertion method to circuits with blackbox macromodel representation," Proceedings of the 11th Electronics Packaging Technology Conference (EPTC), pp. 92-95, Singapore, December 2009.

[5] P. Goh, J. Schutt-Aine, D. Klokotov, J. Tan, P. Liu, W. Dai, and F. Al-Hawari, “Partitioned latency insertion method with a generalized stability criteria,” *IEEE Trans. on Comp., Packaging and Manufacturing Tech.,* vol .1, no. 9, Sept. 2011.

[6] S. Lalgudi and M. Swaminathan, “Analytical stability condition of the latency insertion method for nonuniform GLC circuits,” *IEEE Trans. on Circuits and Systems*, vol. 55, no. 9, Sept. 2008.

[7] Ying Wang, Han Ngee Tan "The Development of Analog SPICE Behavioral Model Based on IBIS Model", Proceedings of the Ninth Great Lakes Symposium on VLSI, GLS '99.

[8] P. Tehrani, Y. Chen, and J. Fang, "Extraction of Transient Behavioral Model of Digital I/O Buffers from IBIS," Proceedings of 1996 ECTC Conference, pp1009-1015.