# SASIMI: Sparsity-Aware Simulation of Interconnect-Dominated Circuits with Non-Linear Devices

J. Jain, S. Cauley, C.-K. Koh, and V. Balakrishnan

School of Electrical and Computer Engineering
Purdue University

PURDUE
UNIVERSITY

# Outline

- MNA Formulation
- Our Contribution
- RLP Formulation
- Computationally Efficient Implementation
  - Nonlinear System
  - Linear System
- Numerical Results
- Conclusion

# Outline

- MNA Formulation
- Our Contribution
- RLP Formulation
- Computationally Efficient Implementation
  - Nonlinear System
  - Linear System
- Numerical Results
- Conclusion

# MNA Formulation

$$\widetilde{G}x + \widetilde{C}\dot{x} = b$$

where

$$\widetilde{G} = \begin{bmatrix} \mathcal{G} & A_l^T \\ -A_l & 0 \end{bmatrix} \quad \widetilde{C} = \begin{bmatrix} \mathcal{C} & 0 \\ 0 & L \end{bmatrix} \quad x = \begin{bmatrix} v_n \\ i_l \end{bmatrix}$$

$$b = \begin{bmatrix} A_i^T I_s + I_{\mathrm{nl}} \\ 0 \end{bmatrix} \quad \mathcal{G} = A_g^T R^{-1} A_g \quad \mathcal{C} = A_c^T C A_c \quad I_{\mathrm{nl}} = f\left(v_n\right)$$

# MNA continued....

$$\frac{d}{dt}x(t)\Big|_{t=kh} \approx \frac{x^{k+1}-x^k}{h} \quad \text{and} \quad x^k \approx \frac{x^{k+1}+x^k}{2}$$

leads to following set of linear and non-linear equations

$$\left(\frac{\widetilde{G}}{2}+\frac{\widetilde{C}}{h}\right)x^{k+1} = -\left(\frac{\widetilde{G}}{2}-\frac{\widetilde{C}}{h}\right)x^k + \frac{b^{k+1}+b^k}{2}$$

$$I_{\mathrm{nl}}^{k+1} = f\left(v_n^{k+1}\right)$$

- Direct implementation $\Rightarrow O(pqn^3)$
  - p: number of simulation steps
  - q: number of Newton-Raphson iterations

# Outline

- MNA Formulation
- Our Contribution
- RLP Formulation
- Computationally Efficient Implementation
  - Nonlinear System
  - Linear System
- Numerical Results
- Conclusion

# Our Contribution

- Extension of the RLP Formulation (proposed in ICCAD 2004) to include non-linear devices, without sacrificing the computational benefits achieved due to sparsity of the linear system.

- Introduction of a novel preconditioner constructed based on the sparsity structure of the non-linear system.

# Outline

- MNA Formulation
- Our Contribution
- RLP Formulation
- Computationally Efficient Implementation
  - Nonlinear System
  - Linear System
- Numerical Results
- Conclusion

# RLP Formulation

- An alternative formulation of MNA equations

  Let

$$C = \left[\begin{array}{c|c} C_{cc} & C_{cv} \\ \hline C_{vc} & C_{vv} \end{array}\right] \quad A^T = \left[\begin{array}{c} A_1^T \\ \hline A_2^T \end{array}\right] \quad A_i^T = \left[\begin{array}{c} A_{i1}^T \\ \hline A_{i2}^T \end{array}\right]$$

- The above decomposition leads to two sets of equations

- Direct implementation termed as Exact-RLP algorithm

# Linear System

- $Ax = b$ with a constant, approximately sparse $A^{-1}$

$$\underbrace{\left(\frac{L}{h} + \frac{R}{2} + \frac{h}{4} A_1 P_{cc} A_1^T\right)}_{X} i_l^{k+1} = \underbrace{\left(\frac{L}{h} - \frac{R}{2} - \frac{h}{4} A_1 P_{cc} A_1^T\right)}_{Y} i_l^k$$

$$+ A_1 v_c^k + \frac{h}{4} A_1 P_{cc} A_{i1}^T \left(I_s^{k+1} + I_s^k\right) - A_1 P_{cc} C_{cv} \left(v_v^{k+1} - v_v^k\right)$$

$$+ \frac{A_2}{2} \left(v_v^{k+1} + v_v^k\right)$$

$$v_c^{k+1} = v_c^k - \frac{h}{2} P_{cc} A_2^T \left(i_l^{k+1} + i_l^k\right) + \frac{h}{2} P_{cc} A_{i1}^T \left(I_s^{k+1} + I_s^k\right)$$

$$- P_{cc} C_{cv} \left(v_v^{k+1} - v_v^k\right)$$

# NonLinear System

- $Ax = b$ with sparse time varying $A$ matrix

$$
\begin{aligned}
C_{vv} v_v^{k+1} &= C_{vv} v_v^k - \frac{h}{2} A_2^T \left( i_l^{k+1} + i_l^k \right) + \frac{h}{2} A_{i2}^T \left( I_s^{k+1} + I_s^k \right) \\
&- C_{vc} \left( v_c^{k+1} - v_c^k \right) + \frac{h}{2} \left( I_v^{k+1} + I_v^k \right)
\end{aligned}
$$

$$
I_v^{k+1} = f\left( v_v^{k+1} \right)
$$

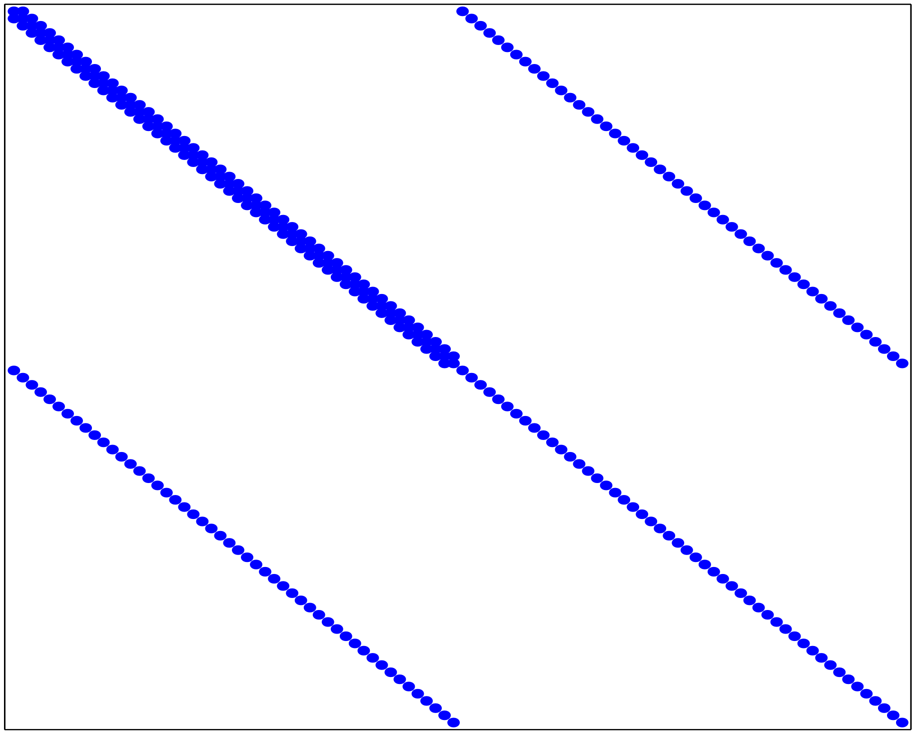
# Outline

- MNA Formulation
- Our Contribution
- RLP Formulation
- Computationally Efficient Implementation
  - Nonlinear System
  - Linear System
- Numerical Results
- Conclusion

# Solving sparse time-varying linear equations

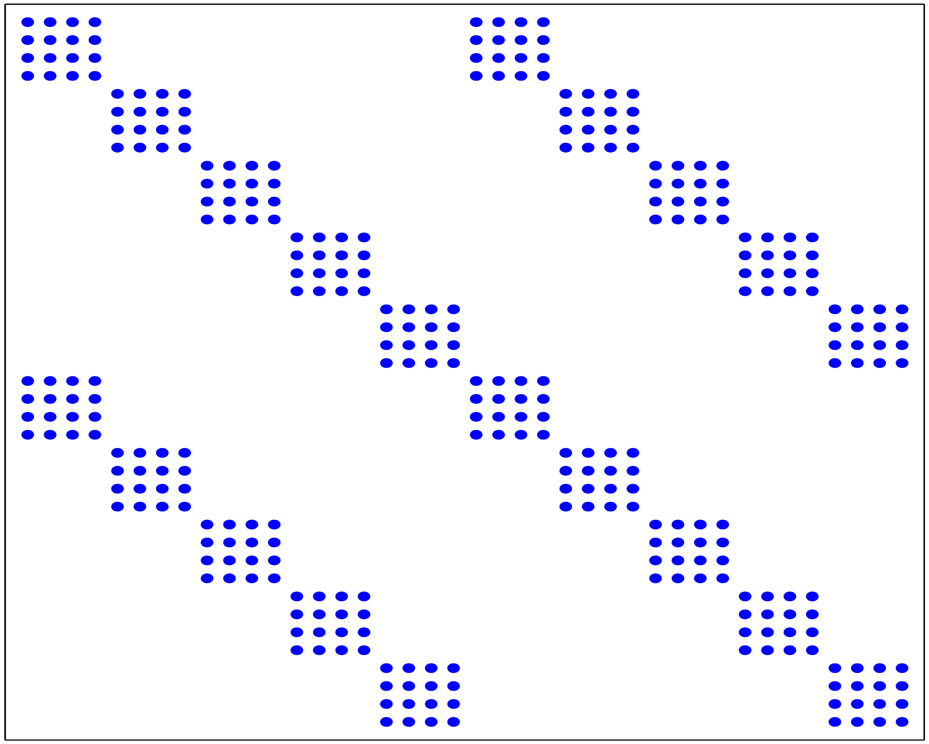
- Krylov subspace method work very well
- The choice of a preconditoner matrix, $M$, greatly affects convergence
- Properties of good preconditioner
  - $M^{-1}A \approx I$
  - Fast solution of $Mz = c$ for a general $c$

# Sparsity in $A$ (Circuit with inverters)



- More complicated circuit structures
  - distribute the entries around the diagonal and off-diagonal bands
  - lead to possibly more off diagonal bands

# Preconditioner Matrix



- Inverse of the preconditioner matrix can be computed efficiently in linear time
- Fast matrix-vector products, again in linear time

# Hadamard Product Representation

$$
B = \begin{pmatrix}
a_1 & -b_1 & & & \\
-b_1 & a_2 & -b_2 & & \\
 & \ddots & \ddots & \ddots & \\
 & & -b_{n-2} & a_{n-1} & -b_{n-1} \\
 & & & -b_{n-1} & a_n
\end{pmatrix}
$$

$$
B^{-1} = \underbrace{\begin{pmatrix}
u_1 & u_1 & \cdots & u_1 \\
u_1 & u_2 & \cdots & u_2 \\
\vdots & \vdots & \ddots & \vdots \\
u_1 & u_2 & \cdots & u_n
\end{pmatrix}}_{U} \circ \underbrace{\begin{pmatrix}
v_1 & v_2 & \cdots & v_n \\
v_2 & v_2 & \cdots & v_n \\
\vdots & \vdots & \ddots & \vdots \\
v_n & v_n & \cdots & v_n
\end{pmatrix}}_{V}
$$

- Simple example of a tridiagonal matrix

# Hadamard Product....

- Explicit formulae exist to compute the sequences $\{u\}$ , $\{v\}$ efficiently in $O(n)$ operations

- Matrix Vector products can be done in linear time

Matrix Vector product $y = B^{-1}c$ can be computed as follows

$$P_{u_i} = \sum_{j=1}^{i} u_j c_j \qquad P_{v_i} = \sum_{j=i}^{n} v_j c_j$$

$$y_1 = u_1 P_{v_1}$$

$$y_i = v_i P_{u_{i-1}} + u_i P_{v_i}$$

# Outline

- MNA Formulation
- Our Contribution
- RLP Formulation
- Computationally Efficient Implementation
  - Nonlinear System
  - Linear System
- Numerical Results
- Conclusion

# Solving $Ax = b$ with a constant, approximately sparse $A^{-1}$

Rewrite equations corresponding to the linear system

$$i_l^{k+1} = X^{-1} Y i_l^k + X^{-1} A_1 v_c^k + \frac{h}{4} X^{-1} A_1 P_{cc} A_{i1}^T \left( I_s^{k+1} + I_s^k \right)$$
$$- X^{-1} A_1 P_{cc} C_{cv} \left( v_v^{k+1} - v_v^k \right) + \frac{A_2}{2} \left( v_v^{k+1} + v_v^k \right)$$

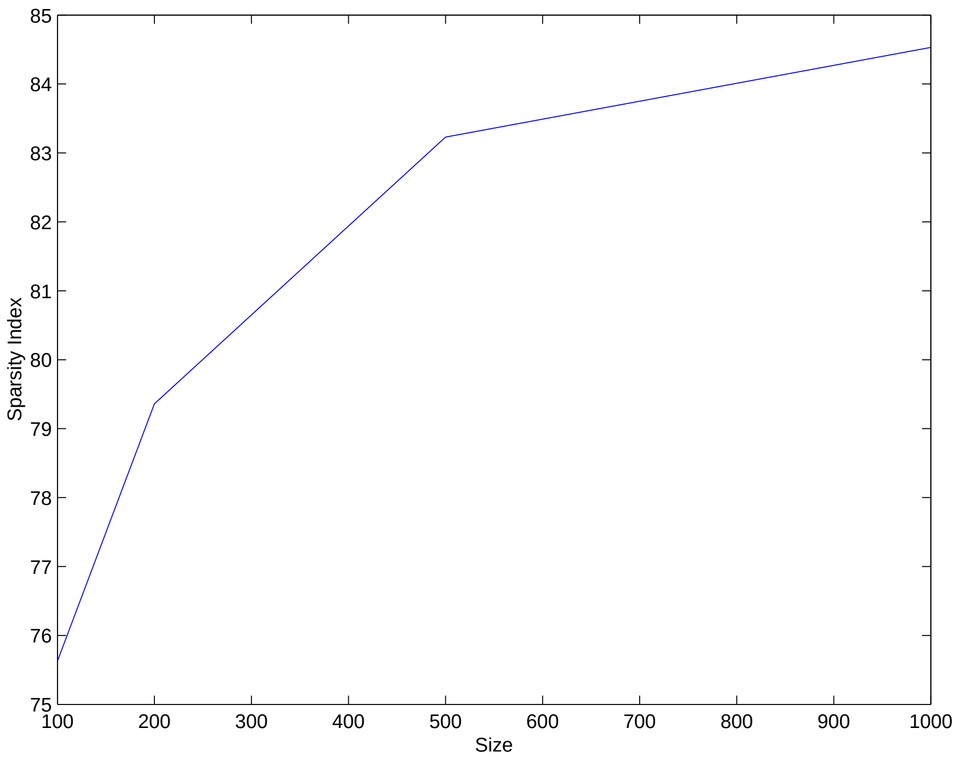
$$X^{-1} A_1 v_c^{k+1} = X^{-1} A_1 v_c^k - X^{-1} A_1 \frac{h}{2} P_{cc} A_2^T \left( i_l^{k+1} + i_l^k \right)$$
$$+ \frac{h}{2} X^{-1} A_1 P_{cc} A_{i1}^T \left( I_s^{k+1} + I_s^k \right) - X^{-1} A_1 P_{cc} C_{cv} \left( v_v^{k+1} - v_v^k \right)$$

# Advantages

- Sparse $X^{-1}$
  - Fast inverse of $X$
  - Sparse multiplications
- Time Complexity $O\left(pq\left(1-\nu\right)l^{2}\right)$
  - $\nu$ is the minimum of the sparsity indices of the matrices $X^{-1}Y$, $X^{-1}A_1$ and $X^{-1}A_1P_{cc}A_{i1}^{T}$
  - $l$ is the size of interconnect structure directly connected to nonlinear devices

# Sparsity in $X^{-1}Y$, $X^{-1}A_1$ and $X^{-1}A_1P_{cc}A_{i1}^T$

- System with parallel conductors driving bank of inverters



# Fast inverse of $X$

- Suppose sparsity pattern in $X^{-1}$ is known (ICCAD 2004)
- Manipulations of only a subset of the entries of the $X$ matrix can be used to compute the inverse matrix
- Let $X$ be a $5 * 5$ matrix
- $3$th row of $X^{-1}$ has the following form:

$$\begin{bmatrix} 0 & \star & \star & 0 & \star \end{bmatrix}$$

# Fast inverse ...

$$\begin{bmatrix} 0 & \star & \star & 0 & \star \end{bmatrix} \times \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

# Fast inverse ...

$$\begin{bmatrix} 0 & \star & \star & 0 & \star \end{bmatrix} \times \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

# Fast inverse ...

Hence the $3$rd row of $X^{-1}$ could be exactly computed from the second row of

$$\begin{bmatrix} X_{22} & X_{23} & X_{25} \\ X_{32} & X_{33} & X_{35} \\ X_{52} & X_{53} & X_{55} \end{bmatrix}^{-1}$$

- $\alpha_i$ nonzero entries in the $i$th row of $X^{-1}$
- In typical Interconnect structures $\alpha_i \ll n$
- Computation time for $X^{-1} \Rightarrow O\left(\sum_i \alpha_i^3\right) = O\left(n\right)$

# Outline

- MNA Formulation
- Our Contribution
- RLP Formulation
- Computationally Efficient Implementation
  - Nonlinear System
  - Linear System
- Numerical Results
- Conclusion

# Experimental set up

- Implemented SASIMI in C++
- Circuits consisting of busses with parallel conductors driving bank of inverters
- $1V$ Periodic square wave
- Time step $.15ps$

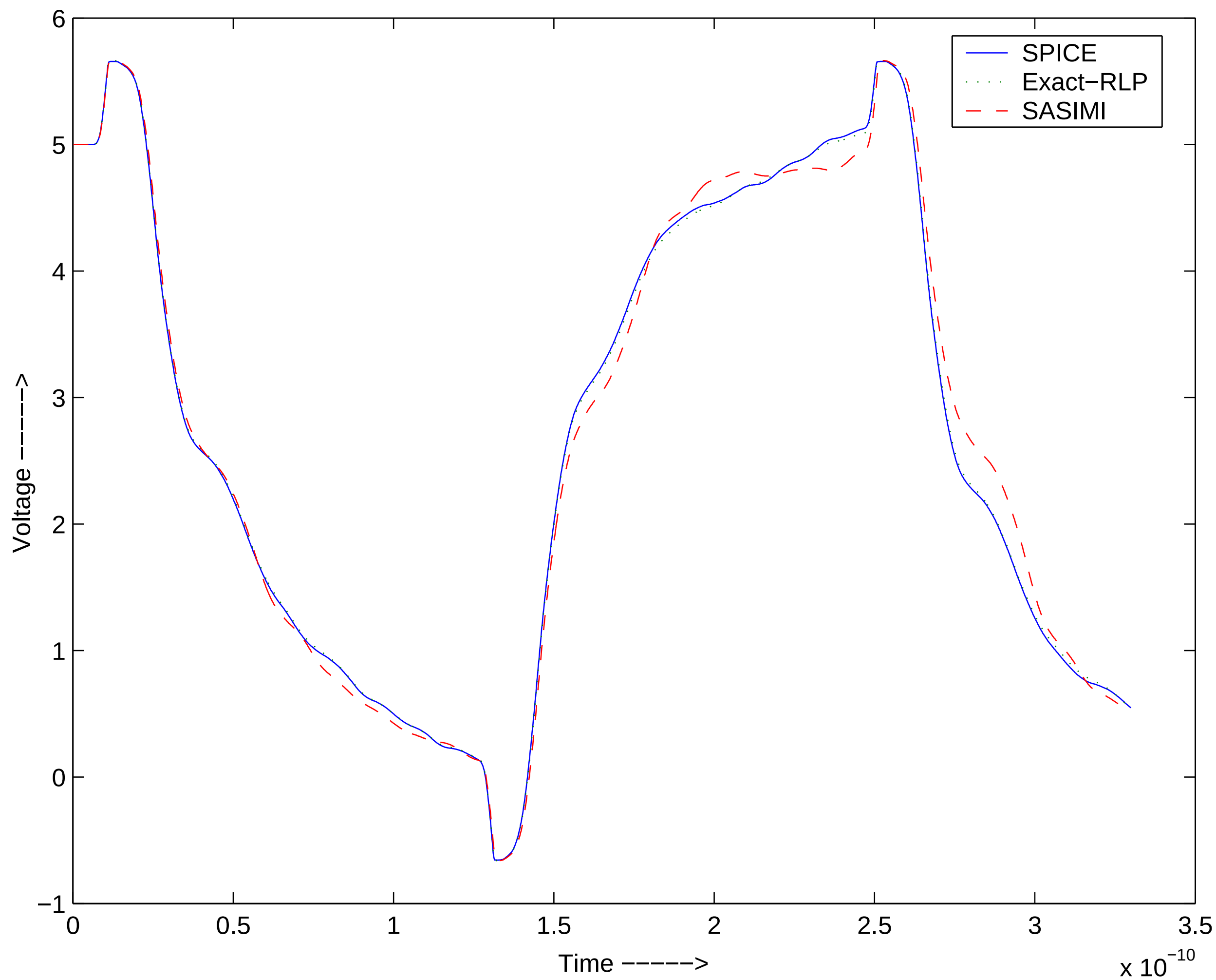
# Results: Run time comparison

| $\sigma$ | $\rho$=5 | | $\rho$=20 | | $\rho$=50 | |
|---|---|---|---|---|---|---|
| | SPICE | SASIMI | SPICE | SASIMI | SPICE | SASIMI |
| 100 | 11.96 | 1.26 | 13.73 | .21 | 13.54 | .12 |
| 200 | 100.25 | 2.68 | 68.72 | .28 | 67.68 | .22 |
| 500 | 3590.12 | 4.872 | 1919.21 | 3.01 | 1790.67 | 1.30 |
| 1000 | $>$12hrs | 22.71 | $>$10hrs | 16.49 | $>$10hrs | 15.20 |
| 2000 | $>$ 1day | 78.06 | $>$ 1day | 59.33 | $>$ 1day | 56.05 |

$$\rho := \frac{\#\ \text{Linear Elements}}{\#\ \text{NonLinear Elements}} \quad \sigma := \#\ \text{Linear Elements}$$

- SASIMI is about 1400 times faster than SPICE
- Percentage improvement increases with increase in $\rho$ and $\sigma$

# Results: Accuracy Comparison




Comparable performance from the point of view of accuracy

# Outline

- MNA Formulation
- Our Contribution
- RLP Formulation
- Computationally Efficient Implementation
  - Nonlinear System
  - Linear System
- Numerical Results
- Conclusion

# Conclusion

- SASIMI, a technique for simulation of large-scale Interconnect dominated VLSI circuits, has been proposed
  - Offers a potential of exploiting sparsity at a Simulation level
  - Computationally efficient
  - Preserves simulation accuracy