Recent Advance on Floating Random Walk Based Capacitance Solver for VLSI Circuit Design

Wenjian Yu, Mingye Song, and Zhezhao Xu

Department of Computer Science & Technology
Tsinghua University
Beijing, China

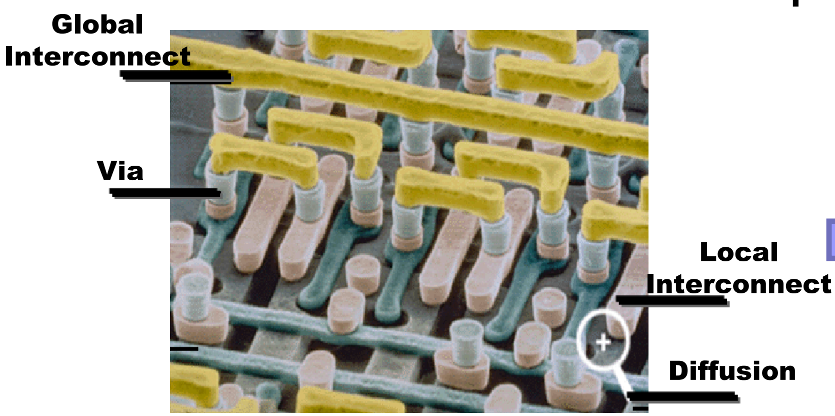
Nov. 3, 2018

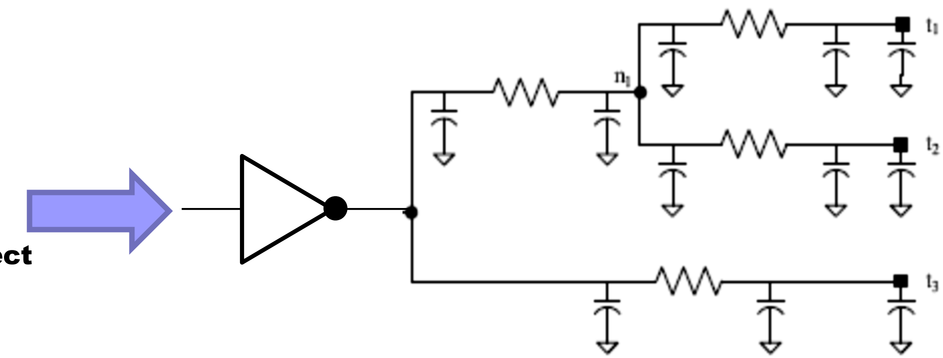
Outline

- Background
- The Floating Random Walk Method
- Distributed Parallel FRW Algorithms
- Accurate Treatment of Complex Floating Metals
- Conclusions

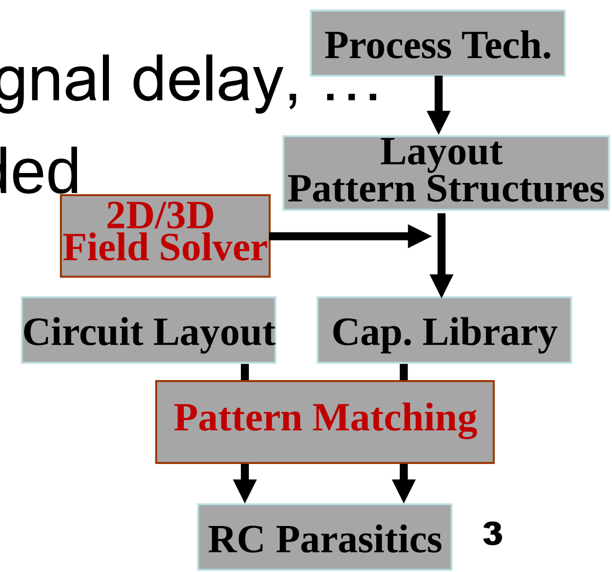
Background

- Accurate capacitance modeling in IC design
  - Device/interconnect capacitance extraction





  - For verifying performance metrics: signal delay, …
  - Accurate field solver is more demanded at advanced technology nodes
  - Also needed for validation of on-chip capacitor structures



Background

- 3-D capacitance field solver
$$\begin{cases} \nabla^2 \phi = 0 \\ C_{ij} = \int_{\Gamma_j} \varepsilon \dfrac{\partial \phi}{\partial \vec{n}} ds \end{cases}$$
- Methods for 3-D field solver
  - Finite difference/finite element method — Raphael, Q3D
    - Stable, versatile; slow
  - Boundary element method — FastCap, Act3D, QBEM/HBBEM
    - Fast, handle complex geometry;
    - Not scalable, need discretization (may affect accuracy)

  (Finite difference/finite element method and Boundary element method:) $$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

  - Floating random walk method — QuickCap/Rapid3D, RWCap
    - No discretization of problem domain (stable accuracy);
    - Scalable for large problem (low memory cost)
    - Embarrassingly parallel

FRW method in a recent *SemiWiki* article →

Field-Solver Parasitic Extraction Goes Mainstream
by Tom Dillinger
Published on 11-28-2017 10:00 AM
3 Comments

Background

- State-of-the-art of FRW based capacitance solver
  - Restrictions on geometry, slow speed (high accuracy)
  - In [1] and [2], extended to handle cylindrical inter-tier-vias in 3-D ICs and the non-Manhattan conductors
  - Combined with MCRW, efficiently extract capacitances of circuits with IP protected or cyclic substructures [3]
  - For efficient variation-aware capacitance modeling [4]
  - Parallel computing using GPUs [5, 6] *more labor on development & maintenance*
  - Distributed parallel FRW is more feasible [7] *Can be improved!*

Currently industry uses coarse-grained workload distribution

[1] C. Zhang, W. Yu, Q. Wang, et al., *IEEE Trans. Comput.-Aided Design*, 34, p. 1977 (2015)
[2] Z. Xu, C. Zhang and W. Yu, *IEEE Trans. Comput.-Aided Design*, 36, p. 120 (2017)
[3] W. Yu, B. Zhang, C. Zhang, et al., *IEEE DATE*, p. 1225 (2016)
[4] P. Maffezzoni, Z. Zhang, et al., *IEEE Trans. Comput.-Aided Design*, 37, p. 2180 (2018)
[5] N. D. Arora, S. Worley, et al., *IEEE EDSSC*, p. 459, 2015
[6] K. Zhai, W. Yu and H. Zhuang, *IEEE DATE*, p. 1661 (2013)
[7] Z. Xu, W. Yu, C. Zhang, et al., *Proc. ACM GLSVLSI*, p. 99 (2016)

Background

- Our recent work on FRW capacitance solver
  - Efficient distributed parallel FRW algorithm
    - Distributed space management construction (large structure)
    - Fine-grained workload distribution (extraction of single-net)
    - With 60 cores, runtime reduced from 824s to 22s (37X)
  - Accurate treatment of complex floating metals
    - A theoretically rigorous approach (b/o electric neutrality)
    - For validation of metal-insulator-metal (MIM) capacitors
    - 3.7X faster than existing approach while achieving same accuracy (i.e. 0.5% systematic error)

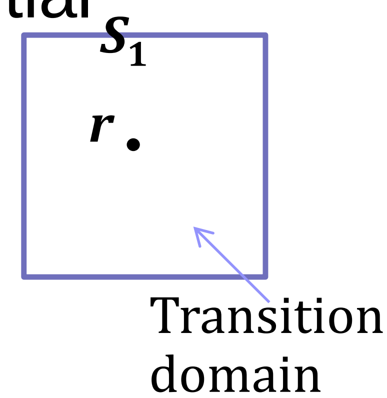
[1] M. Song, Z. Xu, W. Xue, W. Yu, *Proc. ACM GLSVLSI*, p. 189 (2018)
[2] W. Yu, Z. Xu, B. Li, C. Zhuo, *IEEE Trans. Comput.-Aided Design*, 37, p. 1711 (2018)

Outline

- Background
- The Floating Random Walk Method
- Distributed Parallel FRW Algorithms
- Accurate Treatment of Complex Floating Metals
- Conclusions

The Floating Random Walk Method

- The basics of FRW method
  - Integral formula for the electrostatic potential



$$\phi(\boldsymbol{r}) = \oint_{S_1} P_1(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \phi(\boldsymbol{r}^{(1)}) ds^{(1)}$$

P_1 is called surface Green's function, and can be regarded as a probability density function

  - Monte Carlo method: $\phi(\boldsymbol{r}) = \frac{1}{M} \sum_{m=1}^{M} \phi_m$

ϕ_m is the potential of a point on S_1, randomly sampled with P_1

  - What if ϕ_m is unknown? expand the integral recursively

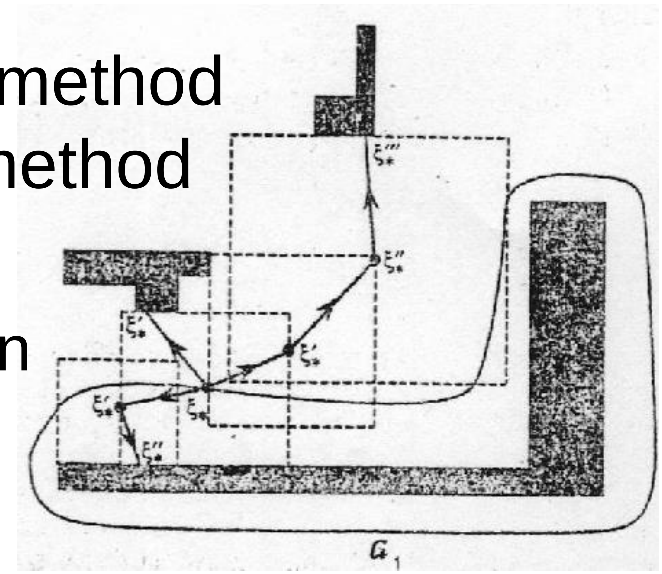
$$\phi(\boldsymbol{r}) = \oint_{S_1} P_1(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \oint_{S_2} P_1(\boldsymbol{r}^{(1)}, \boldsymbol{r}^{(2)}) \cdots$$

$$\oint_{S_k} P_1(\boldsymbol{r}^{(k-1)}, \boldsymbol{r}^{(k)}) \phi(\boldsymbol{r}^{(k)}) ds^{(k)} \cdots ds^{(2)} ds^{(1)}$$

This spatial sampling procedure is called **floating random walk**

The Floating Random Walk Method

- The Markov random process + MC method prove the correctness of the FRW method
- A 2-D example with 3 walks
  - Use maximal cubic transition domain
- How to calculate capacitances?



(picture from [1])

Definition: $$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{12} & C_{22} & C_{23} \\ C_{13} & C_{23} & C_{33} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

⇒ $\mathbf{Q_1 = C_{11} V_1 + C_{12} V_2 + C_{13} V_3}$

Integral for calculating charge (Gauss theorem)

$$Q_1 = \oint_{G_1} F(\mathbf{r}) \cdot \hat{n} \cdot \nabla \phi(\mathbf{r}) d\mathbf{r} = \oint_{G_1} F(\mathbf{r}) \cdot \hat{n} \cdot \nabla \oint_{S_1} P_1(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) ds^{(1)} ds$$

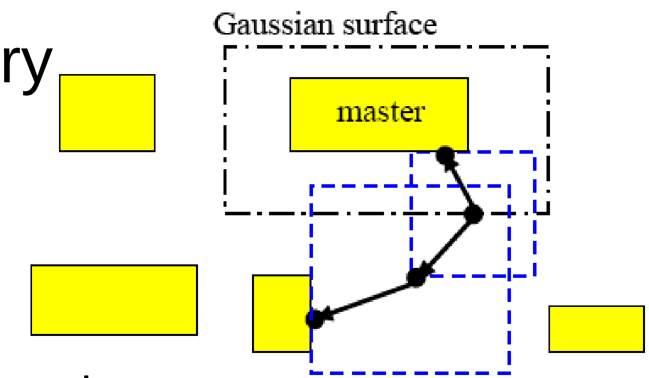
$$= \oint_{G_1} F(\mathbf{r}) g \oint_{S_1} P_1(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) \omega(\mathbf{r}, \mathbf{r}^{(1)}) ds^{(1)} ds$$

weight value, estimate of C_{11}, C_{12}, C_{13} coefficients

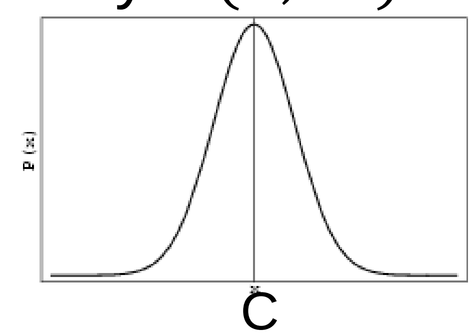
[1] Y. Le Coz and R. B. Iverson, *Solid State Electron.*, 35, p. 1005 (1992)

The Floating Random Walk Method

- The secrets of fast FRW solver for VLSI interconnects
  - Cubic transition domain fits geometry
  - Numerically pre-calculate transition probabilities and weight values
  - Importance sampling; placement of Gaussian surface; space management



- The MC procedure produces random value!
  - Capacitances obtained from many runs obey $N(C,\sigma^2)$
  - Std (1-σ error) of the normal distribution depicts the accuracy level of result
  - σ can be estimated in FRW, $\propto \frac{1}{\sqrt{N_{walk}}}$

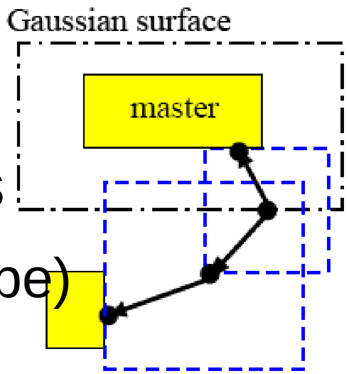


- Total runtime: $T_{total} = N_{walk} N_{hop} T_{hop}$

Outline

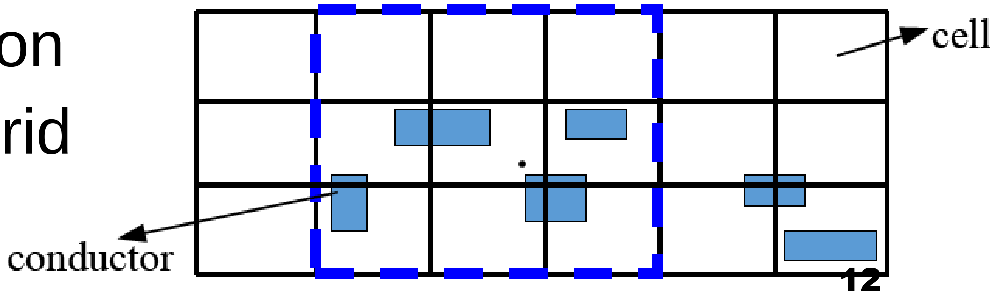
- Background
- The Floating Random Walk Method
- Distributed Parallel FRW Algorithms
- Accurate Treatment of Complex Floating Metals
- Conclusions

Distributed Parallel FRW Algorithms



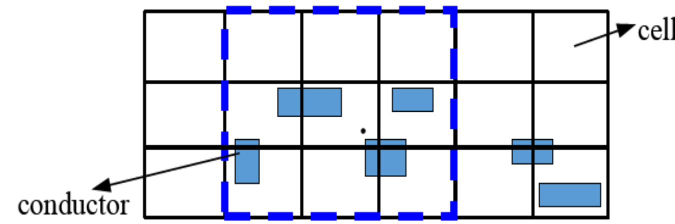
- Space management in the FRW
  - Required for case with many conductor blocks
  - Distance to the nearest conductor (transition cube)
  - Idea of space management
    - Construct a spatial structure (Octree, grid, or hybrid) storing the local conductor information
    - Then, the size of transition cube is quickly calculated
  - Space management construction costs more time!
- Distributed space management construction
  - Key: workload distribution
  - We adopt the uniform grid

Each cell contains a candidate list of nearest conductor blocks

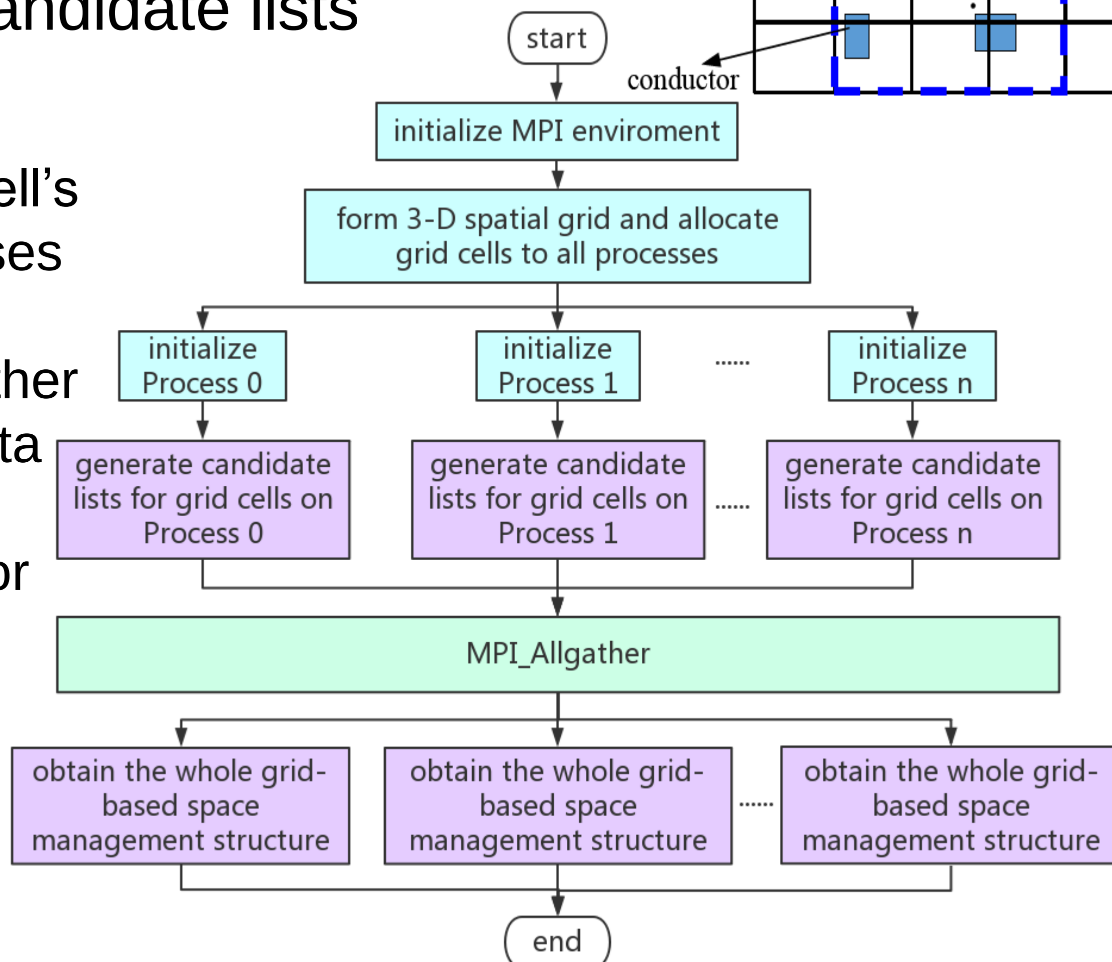


Distributed Parallel FRW Algorithms

- Distributed space management construction
  - Major work: domination check & generate candidate lists
  - Flowchart



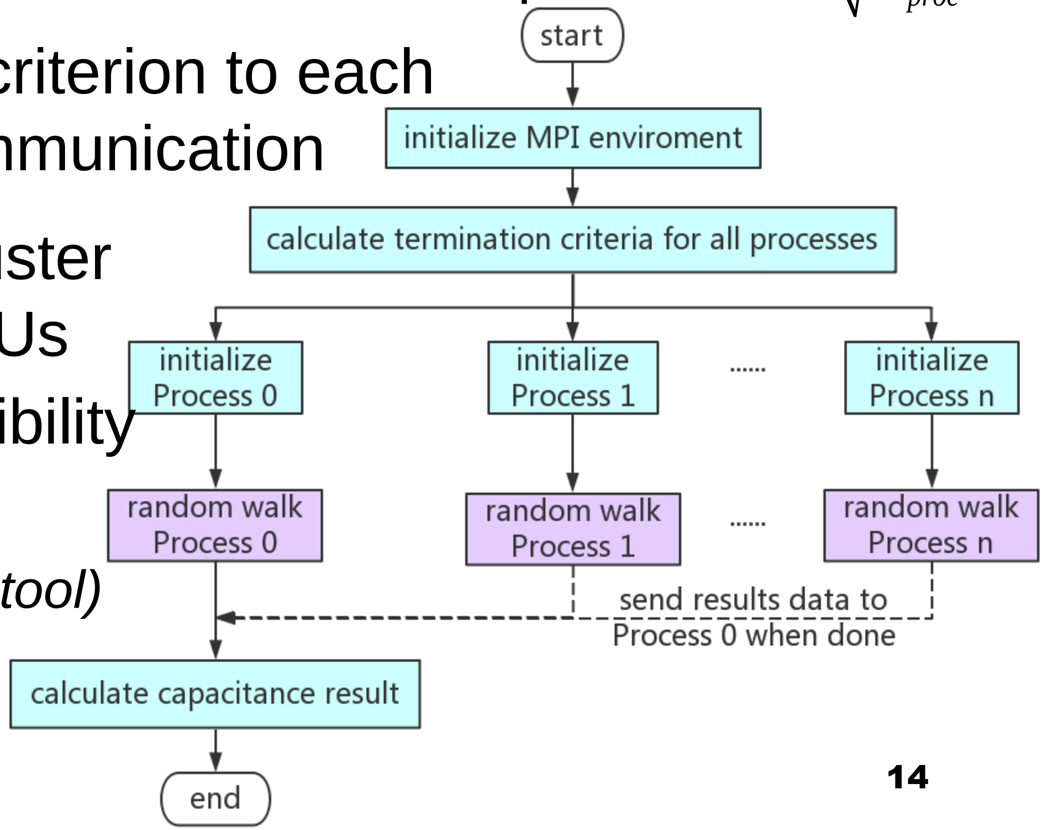
- Distribute the cell’s work to processes
- Use MPI_Allgather to exchange data so that whole structure built for each process
- Compressed data for CL



Distributed Parallel FRW Algorithms

- Distributed FRW procedure
  - An existing distributed approach sends intermediate data to process 0 after every m walks (~large communication)
  - Error $\sigma \propto \frac{1}{\sqrt{N_{walk}}}$ ⇨ Error achieved for each process: $\sigma' = \sqrt{m_{proc}} \cdot \sigma$
  - Idea: set termination criterion to each process; no more communication
  - Also applies to the cluster including different CPUs
  - Ensures the reproducibility of capacitance result

(a practical request for EDA tool)



Distributed Parallel FRW Algorithms

- Experimental results
  - Implemented in C++ and MPI, based on RWCap
  - Tested on a computer cluster with infiniband network
  - Case 1: 484,441 conductor blocks
  - Case 2: 2,302,995 conductor blocks
  - Extract a single net with 0.5% 1-σ error criterion on C_{self}

m_{proc}	Case 1		Case 2	
	time (s)	speedup	time (s)	speedup
1	52.1	1	824.1	1
12	6.55	8.0	76.7	10.7
36	3.15	16.6	29.9	27.6
60	2.27	23.0	22.1	37.4

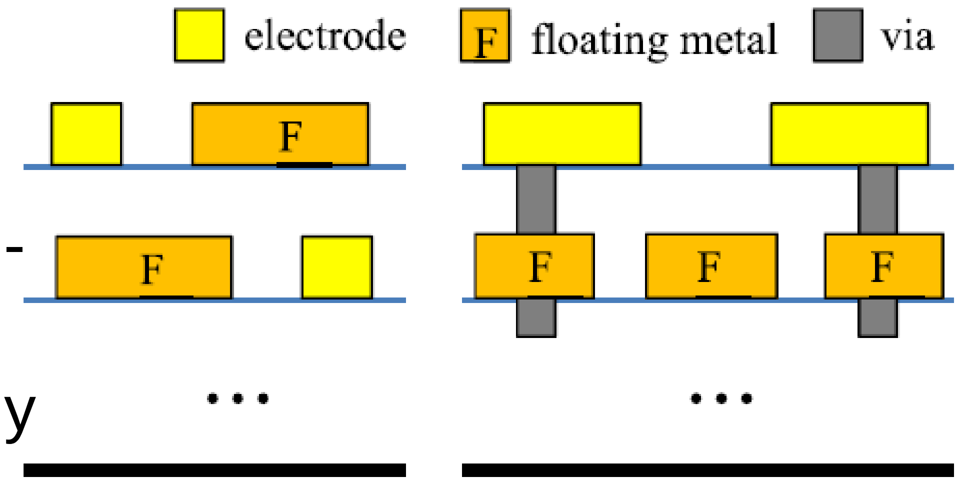
Runtime of FRW procedure for Case 2 is reduced from 189s to 4.9s (39X speedup)

Outline

- Background
- The Floating Random Walk Method
- Distributed Parallel FRW Algorithms
- Accurate Treatment of Complex Floating Metals
- Conclusions

Treatment of Complex Floating Metal

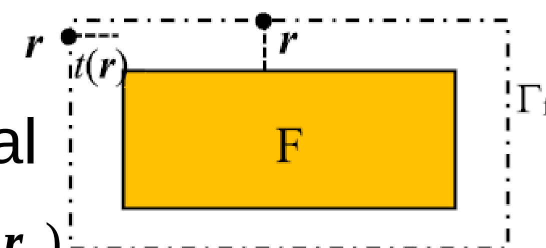
- Motivation
  - Floating metals with complex geometry exist in MIM capacitor for high-voltage circuit design
  - Its validation needs highly accurate simulation



(Cross-section view of two MIM capacitors)

  - Existing treatment for floating metal lacks theory basis; only accurate for square-shape metal fills

- The proposed approach
  - The electric neutrality of a floating metal



2nd-order central difference

$$Q(F) = \oiint_{\Gamma_f} \varepsilon(\mathbf{r}) \frac{\partial \phi(\mathbf{r})}{\partial n(\mathbf{r})} \mathrm{d}\mathbf{r} = 0 \;\Longrightarrow\; 0 \approx \oiint_{\Gamma_f} \varepsilon(\mathbf{r}) \frac{\phi(\mathbf{r}_{\mathrm{out}}) - \phi(\mathbf{r}_{\mathrm{in}})}{2s(\mathbf{r})} \mathrm{d}\mathbf{r}$$

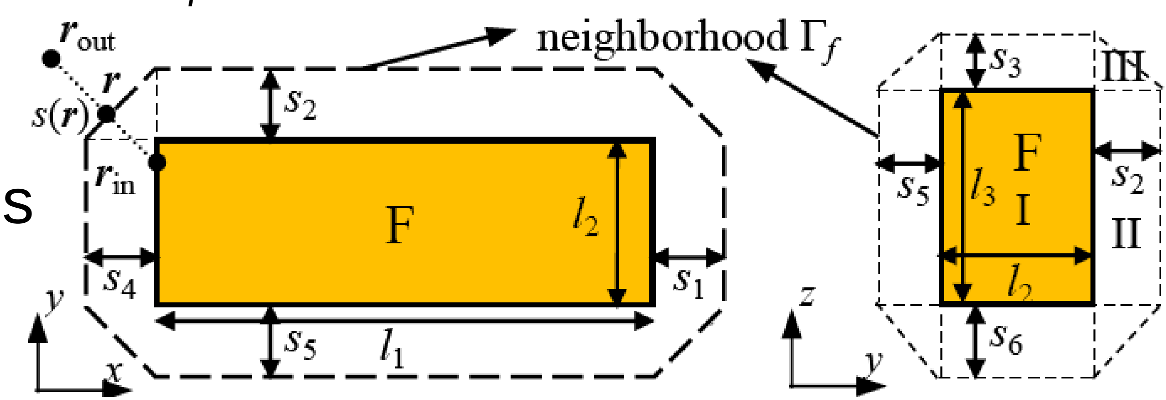
Treatment of Complex Floating Metal

- The proposed approach

$$0 \approx \oiint_{\Gamma_f} \varepsilon(\boldsymbol{r}) \frac{\phi(\boldsymbol{r}_{\text{out}}) - \phi(\boldsymbol{r}_{\text{in}})}{2s(\boldsymbol{r})} \mathrm{d}\boldsymbol{r} \Rightarrow (\oiint_{\Gamma_f} \frac{\varepsilon(\boldsymbol{r})}{s(\boldsymbol{r})} \mathrm{d}\boldsymbol{r})\phi(F) \approx \oiint_{\Gamma_f} \frac{\varepsilon(\boldsymbol{r})}{s(\boldsymbol{r})} \phi(\boldsymbol{r}_{out})\mathrm{d}\boldsymbol{r}$$

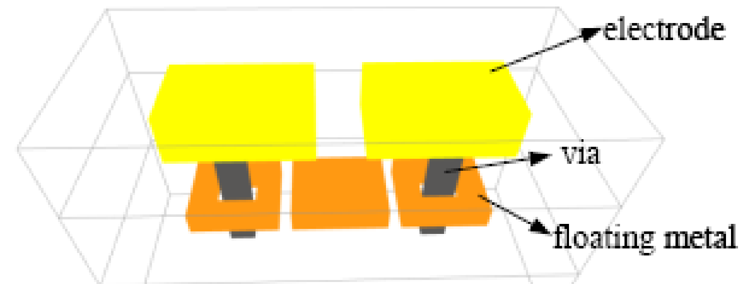
$$\Rightarrow \phi(F) \approx \oiint_{\Gamma_f} P_F(\boldsymbol{r})\phi(\boldsymbol{r}_{out})\mathrm{d}\boldsymbol{r} = \oiint_{\Gamma_f} \frac{\varepsilon(\boldsymbol{r})}{K \cdot s(\boldsymbol{r})} \phi(\boldsymbol{r}_{out})\mathrm{d}\boldsymbol{r}, \text{ where } K = \oiint_{\Gamma_f} \frac{\varepsilon(\boldsymbol{r})}{s(\boldsymbol{r})} \mathrm{d}\boldsymbol{r}$$

  - $P_F(\boldsymbol{r})$ is a probability density function
  - This implies a random transition scheme from the floating metal F to $\boldsymbol{r}_{out}$ (on a *sampling surface*)
  - To ensure $\boldsymbol{r}_{in}$'s on F, a Γ_f with 26 faces is assumed
  - We rigorously derives a new approach with less systematic error



Treatment of Complex Floating Metal

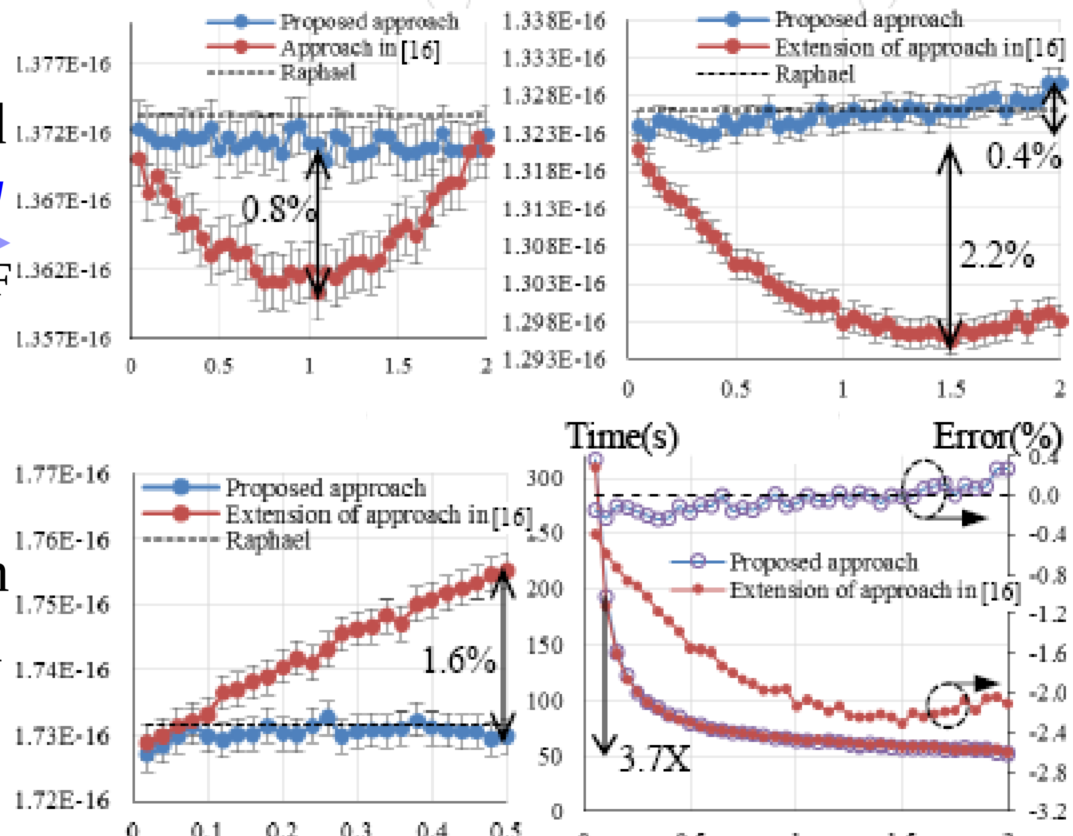
- Handle multi-rectangle shape
  - Construct Γ_f for each block + rejection sampling technique



- Experimental results
  - Three MIM cases are tested

capacitance (w/ ±3σ error) vs. *d*

d: distance from sampling surface to F



  - Take Raphael's as standard
  - Our approach is accurate even for the largest *d*
  - It brings up to 5X reduction of systematic error, or 3.7X speedup over the baseline

[16] S. Batterywala, R. Ananthakrishna, et al., *Proc. IEEE VLSID*, p. 129 (2006)

Outline

- Background
- The Floating Random Walk Method
- Distributed Parallel FRW Algorithms
- Accurate Treatment of Complex Floating Metals
- Conclusions

Conclusions

- Techniques for distributed parallel space management construction and FRW procedure are presented.
- They produce an efficient distributed FRW solver for VLSI capacitance extraction.
- An approach for handling complex floating metals is also presented, which makes the FRW solver capable of accurate on-chip capacitor simulation.

[1] M. Song, Z. Xu, W. Xue, W. Yu, *Proc. ACM GLSVLSI*, p. 189 (2018)
[2] W. Yu, Z. Xu, B. Li, C. Zhuo, *IEEE Trans. Comput.-Aided Design*, 37, p. 1711 (2018)

Thank You !

Wenjian Yu / Tsinghua University

yu-wj@tsinghua.edu.cn

http://numbda.cs.tsinghua.edu.cn