Efficient Techniques for the Capacitance Extraction of Chip-Scale VLSI Interconnects Using Floating Random Walk Algorithm

Chao Zhang, Wenjian Yu

Department of Computer Science & Technology,
Tsinghua University, Beijing, China

Jan. 24, 2014

Outline

- Background
- Floating random walk algorithm
- The Gaussian surface for full-net extraction
- Parallel space management techniques
- Numerical results
- Conclusions

Background

- Capacitance extraction methods
- Boundary / Finite element method
  - Accurate but slow.
- Pattern matching method
  - Fast but inaccurate, able to extract full-chip.
- Floating random walk (FRW)
  - Has tunable accuracy and is scalable to large cases.
  - [Solid-State Electron'1992], [Batterywala'2005], [1], [ICCAD'2009], [SIMPAT'2013], [TCAD'2013].

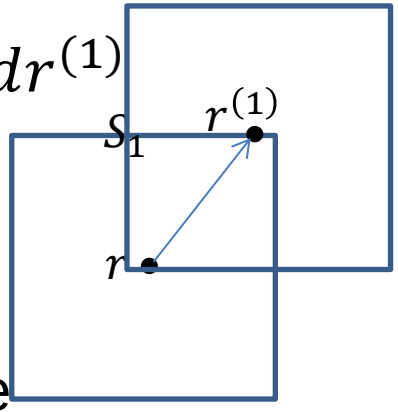
[1] M. Kamon and R. Iverson, "High-accuracy parasitic extraction,"in EDA for IC Implementation, Circuit Design, and Process Technology, CRC Press, Boca Raton, FL, 2006.

Background

- Shortages of FRW algorithm in existing literatures:
  - Cannot extract a full-net with complicated geometries.
  - Cannot handle very large number of conductor blocks.
- The contributions of this paper:
  - Use virtual Gaussian surface sampling technique to support complicated full-net extraction.
  - Improve the Octree based space management structure to organize all blocks in a chip.
  - Propose an efficient FRW algorithm for full-chip extraction.

Floating random walk algorithm

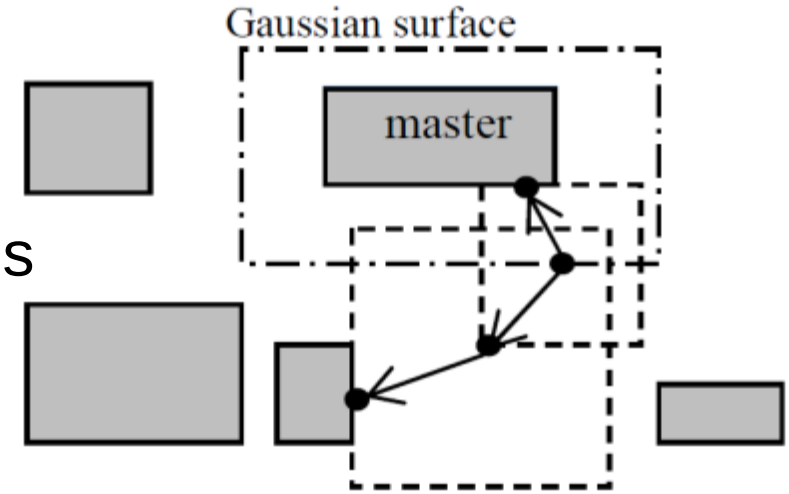
- Electric potential $\varphi(r) = \int_{S_1} P_1\left(r, r^{(1)}\right)\varphi\left(r^{(1)}\right) dr^{(1)}$
  - Use Monte-Carlo integrals:
    - $\varphi(r) = \frac{1}{N}\sum_{i=1}^{N} \varphi(r_i)$, r_i is the ith sample.
    - The samples are randomly selected on surface S_1 according to probability density function (PDF) P_1.



- When $\varphi(r^{(1)})$ is unknown, use nested integrals:
  - $\varphi(r) = \int_{S_1} P_1\left(r, r^{(1)}\right) \int_{S_2} P_1\left(r^{(1)}, r^{(2)}\right) \ldots \int_{S_k} P_k\left(r^{(k-1)}, r^{(k)}\right)\varphi\left(r^{(k)}\right) dr^{(k)} \ldots dr^{(2)}\, dr^{(1)}$
- FRW is the spatial sampling procedure.

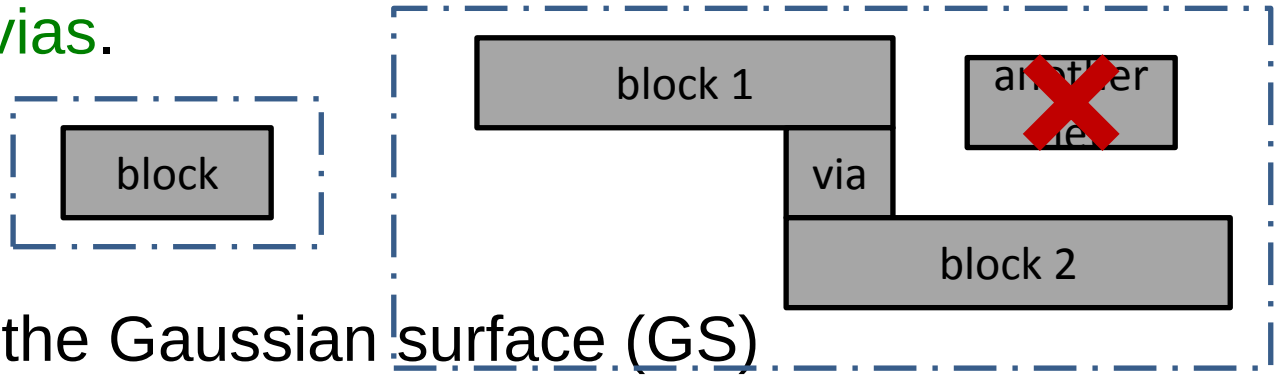
Floating random walk algorithm

- Capacitance $\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \Rightarrow Q_1 = C_{11}V_1 + C_{12}V_2$
- Gauss's law $Q_i = \int_{G_i} \varepsilon(r)E(r)\,dr = \int_{G_i} \varepsilon(r)\nabla\varphi(r)\,dr =$ $\int_{G_i} \varepsilon(r) \int_{S_1} \omega\left(r, r^{(1)}\right) P\left(r, r^{(1)}\right) \varphi\left(r^{(1)}\right) dr^{(1)} dr$
- A 2D example with 2 walks
  - Maximal cube transition domain suits to Manhattan IC geometries



The Gaussian surface for full-net extraction

- To calculate the delay of a full signal path
  - The interconnect wires in a net should be segmented and converted to a distributed RC network.
  - The FRW algorithm should be enhanced to consider the existence of vias.



- Challenge:
  - Constructing the Gaussian surface (GS) for a net with complicated geometries of blocks and vias.
  - The cuboid Gaussian surface is no longer an option.

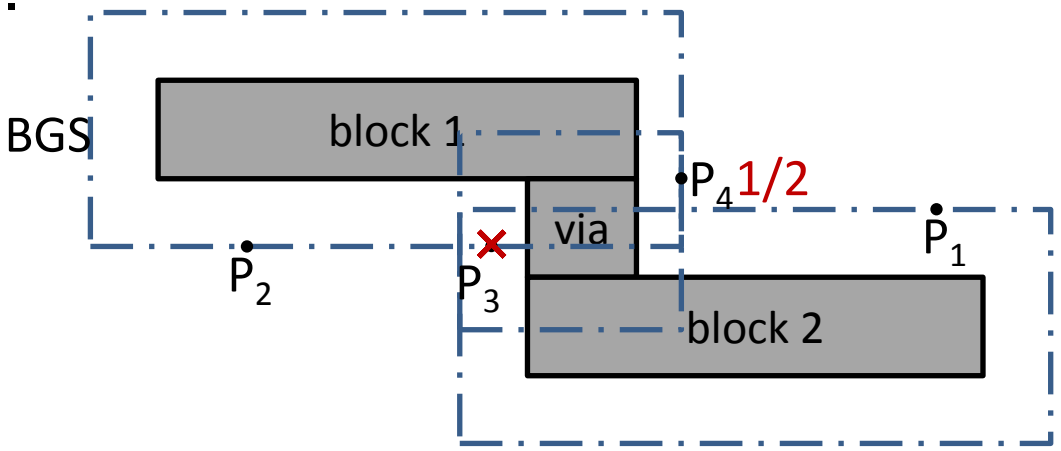
The Gaussian surface for full-net extraction

- Straightforward idea
  - Generate the Gaussian surface for each block (BGS).
  - Then calculate the envelope of them.

- Getting the envelope is complex and unnecessary.
  - Instead of the geometric representation of the GS, FRW algorithm only requires randomly sampling on it.

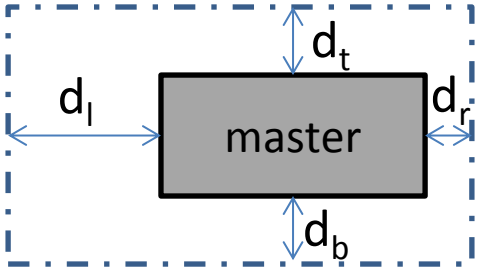
The virtual Gaussian surface sampling technique

- Directly sample on all the BGSs uniformly.
- Only the samples on the envelope are valid.
- The required sampling may not be uniform.
  - To use importance sampling with PDF $p(r)$, each valid sample should only be accepted with probability $p(r)/U$, $U \geq \max\big(p(r)\big)$.



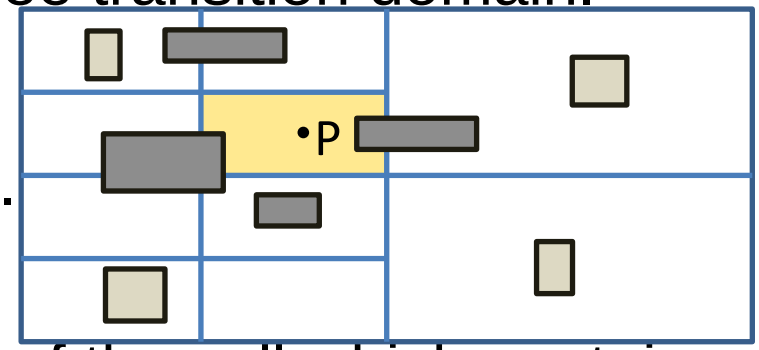
The placement of Gaussian surface

- Generate cuboid BGS for a single block:
  - Ignore other blocks in same net.
  - In each direction, find the nearest block, and select a position between them as one face of the BGS.
- The position affects the performance.
  - At the middle.
  - Equal distance in each direction.
- A general strategy to optimize:
  - Use scale factor to limit the variety in different directions.
  - max(d_t, d_b, d_l, d_r) / min(d_t, d_b, d_l, d_r) ≤ scale factor.



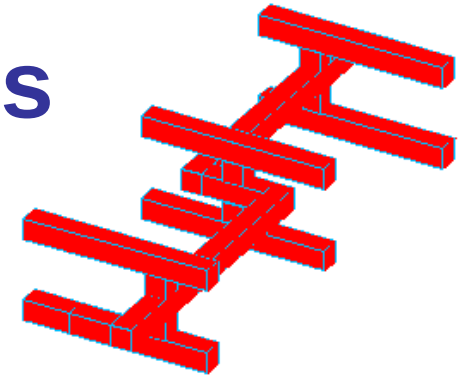
Parallel space management techniques

- Nearest block from a point is queried every hop.
  - To get the maximal conductor-free transition domain.
- Octree
  - Space is divided into small cells.
  - Each cell has a candidate list.
  - Only blocks in the candidate list of the cell which contains the query point are tested.

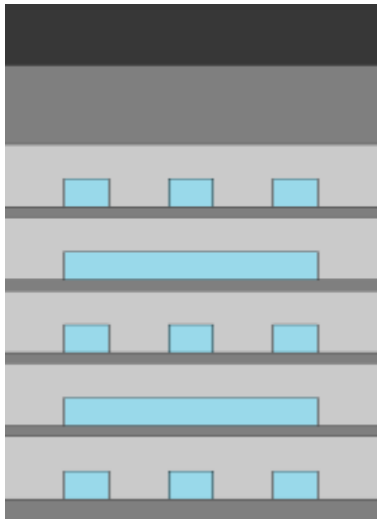


- Construction is very slow. 37062 blocks take half an hour.
  - Each block is compared with all existing candidates. $O(n^2)$
  - Accelerate with pruning skills and parallelization.

Numerical results

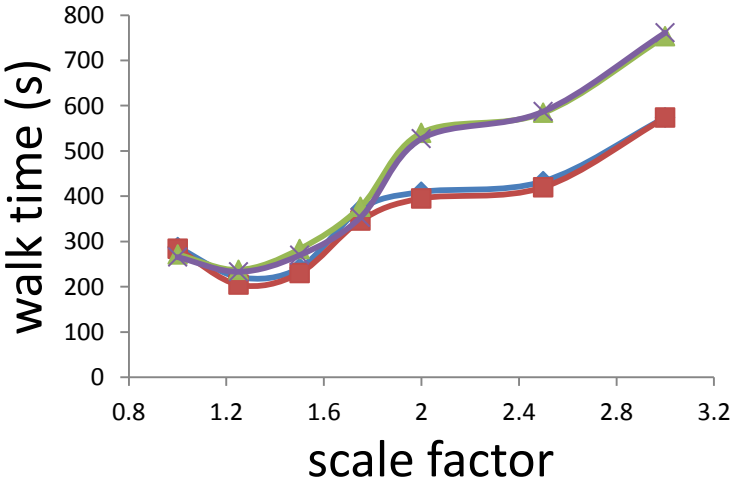


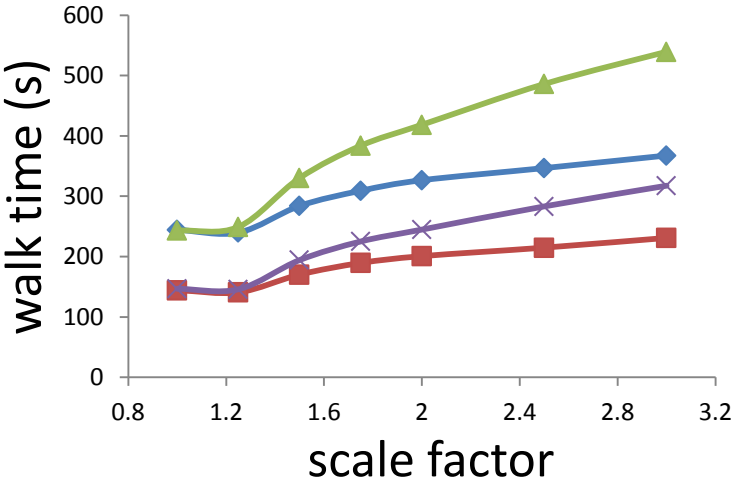
- 4 multi-dielectric cases
  - Case1: small & complex, to verify the correctness.
  - Case2: real design FreeCPU, 3037 nets and 37062 blocks, under 180nm technology.
  - Case3: synthetic, 101596 blocks, 45nm tech.
  - Case4: similar to Case2, larger, half million blocks.
- Criterion for FRW: $1\text{-}\sigma < 1\%$.



Numerical results

- Different scale factors and sampling PDFs.
  - With Case2 & Case3





- Best efficiency:
  - Scale factor = 1.25, $p(r) \propto \varepsilon(r)$ (the red line).
  - Case specific, try with low accuracy.

Numerical results

- The speed up brought by pruning skills in Octree construction
  - In unit of second.

Case	#block	Previous work	With the pruning	speedup
2	37062	1757.9	0.53	3316
3	101595	16595.6	3.12	5319
4	484441	> 2 days	12.27	--

- The parallel construction can further bring 4.4 times speedup with 8 threads.

Numerical results

- Full-chip extraction
  - The enhanced FRW algorithm with proposed techniques is able to extract a full-chip and produce all distributed capacitances of all net.
- For Case2:
  - 8-thread multi-core parallelization.
  - In 35.2 minutes, averagely 0.7 second per net.

Conclusions

- FRW algorithm is a good candidate for achieving both accuracy and scalability.
  - The computational time weakly depends on problem size.
- An efficient FRW algorithm for full-chip capacitance extraction is presented.
  - The VGSS enables full-net extraction and cooperates well with importance sampling.
  - The scale factor strategy can optimize the position of Gaussian surface.
  - The improved Octree is able to organize millions of conductor blocks.

Thank you!