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

[**Invited Paper**]

Wenjian Yu, Mingye Song, and Zhezhao Xu

BNRist, Dept. Computer Science and Tech., Tsinghua University, Beijing 100084, China.
Email: yu-wj@tsinghua.edu.cn, songmy16@mails.tsinghua.edu.cn, zhezhaoxu@gmail.com

Abstract

Capacitance extraction for interconnect parasitics modeling is the basis of accurate circuit simulation and physical verification for quality integrated circuit (IC) design. Due to its scalability and reliability, the floating random walk (FRW) based solver has been widely used for the capacitance extraction. In this paper, the recently developed techniques on distributed parallel FRW solver and for handling complex floating metals are presented. With them, the efficiency and accuracy of capacitance extraction/simulation for very large-scale integration (VLSI) circuit design are remarkably improved.

1. Introduction

Accurate capacitance modeling and simulation with three-dimensional (3-D) field solver has been applied to the design of advanced integrated circuits (ICs) [1, 2]. It enables device/interconnect capacitance extraction necessary for the verification of circuit performance metrics, and provides a design validation tool for on-chip capacitor structures. There are three main kinds of methods for 3-D capacitance solver: domain discretization method, boundary element method, and the floating random walk (FRW) method [3-6]. Compared to the others, the FRW method is more stable in accuracy and scalable to large cases, due to its nature of a discretization-free method. And, as a Monte Carlo method, it's embarrassingly parallelizable. This makes the FRW based capacitance solver popular nowadays [2], since parallel-computing facilities are easily available. Recently, the FRW method was extended to efficiently handle cylindrical inter-tier-vias in 3-D ICs [7] and the non-Manhattan conductors in packaging and touchscreen [8]. For the variation-aware capacitance modeling [9], a technique based on the FRW method was proposed [10]. Combined with a Markov-chain random walk approach, it was also made to efficiently extract the capacitances of circuits with IP protected or cyclic substructures [11].

A lot of work has been devoted to the FRW method to improve its efficiency. Most of them consider the multi-threaded parallel implementation on a single computer [4, 12]. To tackle very large computing task, the parallel computing using GPU or multiple computers should be considered [6, 13, 14]. Although the approach based on GPU can be more energy-efficient, its implementation requires more labor on software development and maintenance. In contrast, the distributed parallel FRW algorithm on computer cluster or the *cloud* is more feasible. This has been evidenced by the industrial practice of multi-net or full-chip extraction using FRW solver. Its basic idea is to assign subsets of nets to each machine and then collect their results on a host machine. This approach with coarse-grained workload distribution is not efficient enough, and not suitable for the extraction of a single net. For the capacitance extraction in VLSI design, the parallel FRW solver handling millions of conductor blocks is of concern. Besides, as complex floating metals exist for the on-chip decoupling capacitor, how to accurately simulate it is also a challenge.

In this paper, we first present efficient distributed FRW techniques for large-scale VLSI capacitance extraction, after a brief review of the FRW method. Then, the technique accurately simulating the structure with complex floating metals is presented, which adapts to the validation of metal-insulator-metal (MIM) capacitors.

2. Floating Random Walk Based Capacitance Solver

The FRW method is originated from expressing the electric potential of a point $\boldsymbol{r}$ as an integral of the potential on surface S enclosing $\boldsymbol{r}$ [3, 4, 15]:

$$\phi(\boldsymbol{r}) = \oint_S P(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \phi(\boldsymbol{r}^{(1)}) \mathrm{d}\boldsymbol{r}^{(1)} , \quad (1)$$

where $P(\boldsymbol{r}, \boldsymbol{r}^{(1)})$ is called surface Green's function and can be regarded as a probability density function and $\mathrm{d}\boldsymbol{r}^{(1)}$ denotes infinitesimal area at position $\boldsymbol{r}^{(1)}$. Therefore, $\phi(\boldsymbol{r})$ is the statistical mean of $\phi(\boldsymbol{r}^{(1)})$, and can be calculated with a Monte Carlo procedure sampling S. The domain enclosed by S is called transition domain.

In capacitance extraction, the coupling capacitances between a conductor i and other conductors are wanted. With the FRW method, a Gaussian surface G_i is constructed to enclose conductor i (see Fig. 1). According to the Gauss theorem, the charge of conductor i

$$Q_i = \oint_{G_i} \varepsilon(\boldsymbol{r}) g \int_{S^{(1)}} \omega(\boldsymbol{r}, \boldsymbol{r}^{(1)}) q(\boldsymbol{r}, \boldsymbol{r}^{(1)}) \phi(\boldsymbol{r}^{(1)}) \mathrm{d}\boldsymbol{r}^{(1)} \mathrm{d}\boldsymbol{r} , \quad (2)$$

where $\varepsilon(\boldsymbol{r})$ is the dielectric permittivity at point $\boldsymbol{r}$, $q(\boldsymbol{r}, \boldsymbol{r}^{(1)})$ is the probability density function for sampling on $S^{(1)}$, the surface of a transition domain. g is a constant, which satisfies $\oint_{G_i} \varepsilon(\boldsymbol{r}) g \mathrm{d}\boldsymbol{r} = 1$. $q(\boldsymbol{r}, \boldsymbol{r}^{(1)})$ may be different from $P(\boldsymbol{r}, \boldsymbol{r}^{(1)})$, and $\omega(\boldsymbol{r}, \boldsymbol{r}^{(1)})$ is the weight value [4]. Thus, Q_i can be estimated as the statistical mean of sampled values on G_i, which is further the mean of sampled potentials on $S^{(1)}$ multiplying the weight value. If the

This work was supported by National Natural Science Foundation of China (No. 61422402).

978-1-5386-4441-6/18/$31.00 ©2018 IEEE

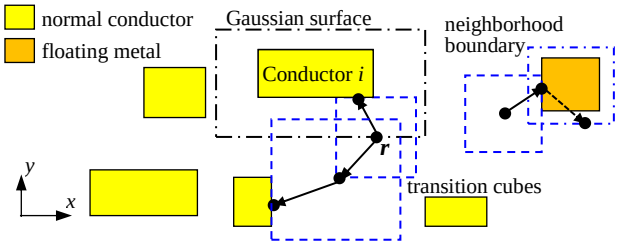

Figure 1. Two random walks from ***r*** in the FRW method, and a random walk encountering a floating metal (2-D top view).

sampled potential is unknown, this spatial sampling procedure will repeat until a point with known potential is obtained (e.g., on conductor surface). It forms a floating random walk (FRW) including a sequence of hops. Each hop is from the center of a transition domain to its boundary. With a number of such walks, the statistical mean of the weight values for the walks terminating at conductor j approximates the coupling capacitance C_{ij} (if $j \neq i$), or the self-capacitance C_{ii}.

For the structure with floating dummies, the FRW method was modified in [16] to calculate the equivalent capacitances among normal conductors. As shown in Fig. 1, the random walk does not terminate at a floating metal, and instead continues with the next position randomly selected on its neighborhood boundary. The neighborhood boundary is the uniform inflation of the cuboid dummy fill. Although this approach demonstrated good accuracy for structures including floating dummies, it could induce large systematic error for structure with general-shape floating metals, like the MIM capacitor.

3. Distributed Parallel FRW Algorithms

In this section, we first present a parallel space management technique, which is necessary for handling large-scale interconnect structures. Then, the efficient distributed parallel FRW procedure is presented.

3.1 Distributed space management construction

The cubic transition domain is widely adopted because it fits well the Manhattan-shaped interconnects in VLSI circuit. The space management technique [4, 12] is for quickly finding the nearest conductor for constructing the transition cube, and required for simulating a case with thousands of conductor blocks. Its basic idea is to construct a spatial data structure (like Octree, grid, or their hybrid) storing the local conductor information, and then with it the size of transition cube can be quickly calculated. However, only a multi-thread parallel algorithm exists for generating the spatial data. Notice for a large-scale case, the construction of space management data costs more time than the FRW procedure.

To develop a distributed algorithm for constructing space management data, we adopt the uniform grid as the spatial structure as it leads to simpler inter-process communication. Fig. 2 demonstrates a grid with candidate list structure. Every grid cell contains a candidate list that consists of the conductor blocks which can be

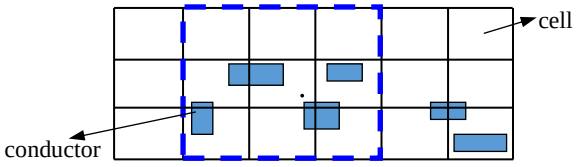

Figure 2. A uniform grid structure. The blue dashed box encloses nearby cells of the dot's owner cell (2-D top view).

the nearest to the cell's inside points. Generating these candidate lists are the major work of space management construction, for which conductor blocks in nearby cells are inserted into the candidate list after checking domination relationship. As the candidate lists are independent to each other, their generation can be distributed to processes. After all processes finish the work, the results are sent among processes to build the whole spatial data structure for each process.

The flowchart of the distributed space management construction is shown in Fig. 3, which is implemented with message passing interface (MPI). A well designed communication scheme is crucial to its efficiency. Two concerns are how to communicate between processes and what to send. We use *MPI_Allgather* command for the communication, while a compressed data structure is designed for the information of candidate lists. Therefore, high parallel efficiency can be achieved [17].

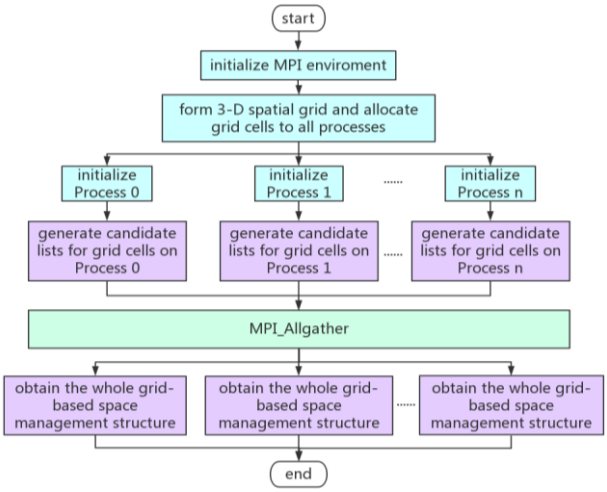

Figure 3. The distributed space management construction.

3.2 Distributed FRW procedure

The random walk procedure in the FRW method is suitable for parallel computing, as the random walks are independent to each other. An existing distributed FRW algorithm sends intermediate data from each process to process 0 after executing every m walks [14]. It still has large communication overhead. From [4], we see that the estimated error of capacitance is inversely proportional to the square root of the number of walks. It means if we evenly divide the walks to processes, it's equivalent to assigning a new termination criterion to each process:

$$\delta' = \sqrt{m_{proc}} \cdot \delta, \qquad (3)$$

where m_{proc} is the number of processes and δ is the specified accuracy criterion of the FRW algorithm (the program stops when capacitance error becomes less than δ). Therefore, we can just set the termination criterion (3) to each process at the beginning, and then no more communication among processes is needed. The flow-chart of this distributed FRW procedure is as Fig. 4.

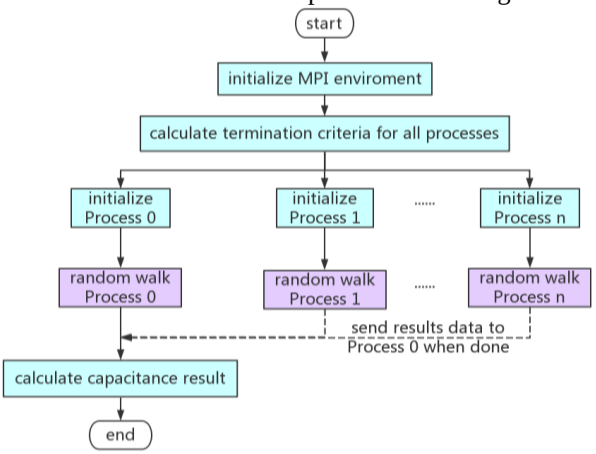

Figure 4. The distributed parallel FRW procedure.

This distributed algorithm reduces the communication cost to the least, and thus makes large acceleration. To handle the situation where the machines in the cluster have different performance, a comprehensive formula for assigning the termination criterion is also derived [17]. Another merit of this algorithm is that it ensures the reproducibility of capacitance result. The reproducibility has drawn a lot of attention in the community of parallel computing [18], but is not addressed in the research of parallel FRW solver. Actually it's a practical request from customers. With fixed random number seeds and this static workload distribution, we're able to make this parallel FRW solver 100% reproducing same capacitance results when the simulated cases, the termination criterion and the number of processes do not vary. At the same time, the accuracy and efficiency is not degraded.

3.3 Experimental results

The above techniques have been implemented into RWCap [4], and tested on a computer cluster. Each node of the cluster includes 12-core Intel Xeon X5670 CPU at 2.93GHz and 32GB memory, and the nodes are connected with infiniband QDR network. Two VLSI design cases, one containing 484,441 conductor blocks and the other 2,302,995 blocks, are tested. The runtimes of space management construction are listed in Table 1.

Table 1. The runtimes of the parallel space management construction for two test cases from VLSI circuits.

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

From the table we see that with 60 processes the cost of constructing the space management can be reduced from 824 seconds to 22 seconds, meaning **37.4X** speedup. We also set 0.5% 1-σ error criterion on self-capacitance, and extract a single net. With 60 processes the cost of FRW procedure for Case 2 is reduced by **39X**, from 189 seconds to 4.9 seconds. This validates the high efficiency of the proposed distributed FRW algorithms.

4. Accurate Treatment of Complex Floating Metals

In this section, we derive a theoretically rigorous approach (based on electric neutrality) for handling floating metals in the FRW method. Then, experiments are presented to demonstrate its advantage.

For each floating metal F, the electric neutrality means its charge equals to zero:

$$Q(F) = \oint_{\Gamma_f} \varepsilon(\boldsymbol{r}) \frac{\partial \phi(\boldsymbol{r})}{\partial n(\boldsymbol{r})} \mathrm{d}\boldsymbol{r} = 0 \quad , \tag{4}$$

where Γ_f is a neighborhood boundary around F, $\varepsilon(\boldsymbol{r})$ is the dielectric permittivity at point $\boldsymbol{r}$, and $\partial\phi(\boldsymbol{r})/\partial n(\boldsymbol{r})$ is the outer normal electric field intensity. Approximating the partial derivative in (4) with the central difference formula, we obtain

$$0 \approx \oint_{\Gamma_f} \varepsilon(\boldsymbol{r}) \frac{\phi(\boldsymbol{r}_{\text{out}}) - \phi(\boldsymbol{r}_{\text{in}})}{2s(\boldsymbol{r})} \mathrm{d}\boldsymbol{r} , \tag{5}$$

where $\boldsymbol{r}_{\text{in}}$ and $\boldsymbol{r}_{\text{out}}$ are two points such that the line connecting them intersects Γ_f perpendicularly at $\boldsymbol{r}$. And, $\boldsymbol{r}_{\text{in}}$ and $\boldsymbol{r}_{\text{out}}$ have the same distance $s(\boldsymbol{r})$ to point $\boldsymbol{r}$ (see Fig. 5). If $\boldsymbol{r}_{\text{in}}$ is on F such that $\phi(\boldsymbol{r}_{\text{in}})$ equals to $\phi(F)$, we have:

$$\left(\oint_{\Gamma_f} \frac{\varepsilon(\boldsymbol{r})}{s(\boldsymbol{r})} \mathrm{d}\boldsymbol{r}\right) \phi(F) \approx \oint_{\Gamma_f} \frac{\varepsilon(\boldsymbol{r})}{s(\boldsymbol{r})} \phi(\boldsymbol{r}_{out}) \mathrm{d}\boldsymbol{r} , \quad \text{or} \tag{6}$$

$$\phi(F) \approx \oint_{\Gamma_f} P_F(\boldsymbol{r}) \phi(\boldsymbol{r}_{out}) \mathrm{d}\boldsymbol{r} = \oint_{\Gamma_f} \frac{\varepsilon(\boldsymbol{r})}{K \cdot s(\boldsymbol{r})} \phi(\boldsymbol{r}_{out}) \mathrm{d}\boldsymbol{r} , \tag{7}$$

$$\text{where} \quad P_F(\boldsymbol{r}) = \frac{\varepsilon(\boldsymbol{r})}{K \cdot s(\boldsymbol{r})} \quad \text{and} \quad K = \oint_{\Gamma_f} \frac{\varepsilon(\boldsymbol{r})}{s(\boldsymbol{r})} \mathrm{d}\boldsymbol{r} . \tag{8}$$

Obviously, $P_F(\boldsymbol{r})$ is a probability density function, and (7) implies a random transition scheme from the floating metal F. This is different from the method in [16]. We do not assume an integral of electric potential on the neighborhood boundary Γ_f. Instead, the integral in (7) involves the electric potential on possible locations of $\boldsymbol{r}_{\text{out}}$ which constitute an unclosed *sampling surface*.

To ensure the existence of $\boldsymbol{r}_{\text{in}}$ on F, we construct the neighborhood boundary Γ_f for a cuboid-shape floating

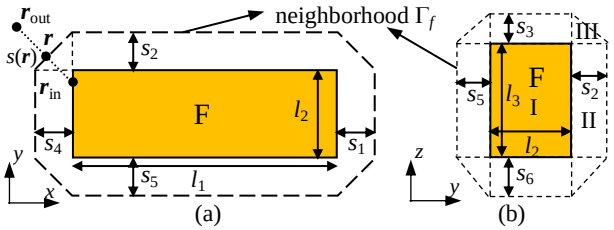

Figure 5. The central difference formula based approach with a special neighborhood boundary. (a) top view; (b) side view.

metal with dimensions $l_1 \times l_2 \times l_3$, as that shown in Fig. 5. We first translate each face of F outward a distance and then connect these translated faces with bevel faces. The resulted Γ_f includes 26 faces: 6 faces obtained by translating the faces of F (denoted by type I), 12 faces each connecting two adjacent type-I faces (denoted by type II), and 8 faces each connecting three type-II faces (denoted by type III). So far, we have derived a new approach for handling cuboid-shape floating metal. It is based on (5) rigorously with the 2nd-order central differentiation formula. Thus, it has less systematic error than the method in [16]. Further, we derive the integrals of $P_F(\boldsymbol{r})$ on the three types of faces in Γ_f. Then, we can perform random sampling on Γ_f according to $P_F(\boldsymbol{r})$ in (8) and the rejection sampling technique. For each sample, the next random-walk position $\boldsymbol{r}_{\text{out}}$ is finally obtained. This forms a new transition scheme applied while the random walk encounters a floating metal, and results in a new modified FRW method for capacitance simulation.

The MIM capacitor often involves floating metals in multi-rectangle shape (see Fig. 6). Eq. (7) still holds and Γ_f can be constructed similarly. To simplify its construction and sampling, we consider a multi-rectangle metal as a combination of multiple blocks. Then, we just construct Γ_f for each block, and then make sampling on them and apply the rejection sampling technique [12]. This correctly produces sampling points, and enables the random transitions starting from a multi-rectangle floating metal. This approach can also be extended to handle structures with multi-dielectric environment [19].

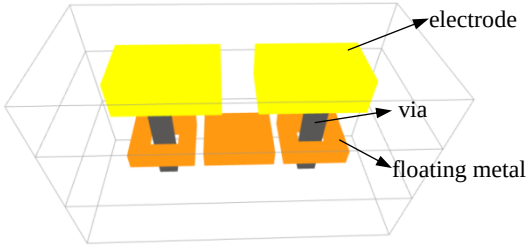

Figure 6. An MIM capacitor avoiding dielectric breakdown.

Three capacitor structures with floating metals (Case 3~5) are tested. Case 5 is just like that in Fig. 6. The proposed approach is compared with that in [16]. For the both, we make each face of floating metal have same distance d to the neighborhood boundary. The calculated capacitances for varied d values are shown in Fig. 7(a)~(c). To suppress the stochastic error, a small 0.05% 1-σ error is set as the termination criterion. And, the golden values from Raphael are also plotted. From them we see with the approach in [16] and its extension the systematic error can be very large, while the proposed approach keeps reasonable accuracy even for the largest d. It brings up to 5X reduction on the systematic error. We draw Fig. 7(d) to reflect how d affects the runtime of the approaches and their errors. It reveals that they have almost same runtime-vs-d trend, but different errors.

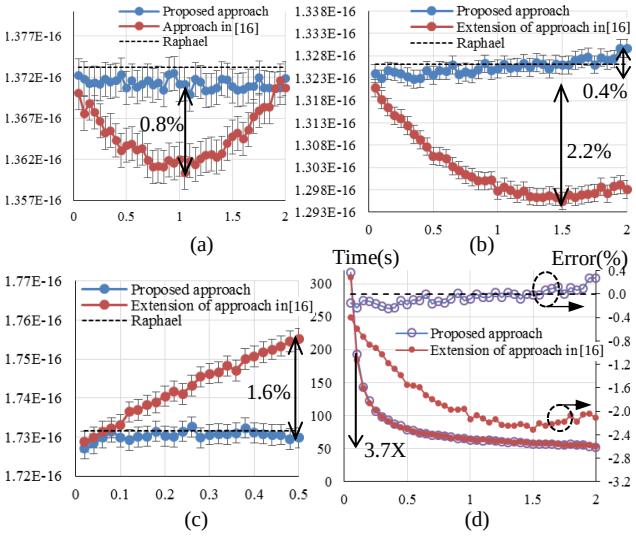

Figure 7. (a)~(c) Capacitance (with ±3σ error bars) versus the d value for Case 3~5. (d) Runtime and error trends for Case 4.

From it we also see that, to achieve 0.5% systematic error the proposed approach can be over 3.7X faster than the extension of the approach in [16].

5. 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.

References

[1] W. Yu and X. Wang, Advanced Field-Solver Techniques for RC Extraction of Integrated Circuits, Springer Inc., 2014.
[2] T. Dillinger, https://www.semiwiki.com/forum/content/7154-field-solver-parasitic-extraction-goes-mainstream.html, 2017
[3] Y. Le Coz and R. B. Iverson, Solid State Electron., 35, p. 1005 (1992).
[4] W. Yu, H. Zhuang, C. Zhang, et al., IEEE Trans. Comput.-Aided Design, 32, p. 353 (2013).
[5] G. Rollins, Rapid3D 20X performance improvement, 2010.
[6] N. D. Arora, S. Worley, et al., IEEE EDSSC, p. 459, 2015
[7] C. Zhang, W. Yu, Q. Wang, et al., IEEE Trans. Comput.-Aided Design, 34, p. 1977 (2015).
[8] Z. Xu, C. Zhang and W. Yu, IEEE Trans. Comput.-Aided Design, 36, p. 120 (2017).
[9] W. Yu, Q. Zhang, Z. Ye, et al., Microelectronics Reliability, 52, p. 704 (2012).
[10] P. Maffezzoni, Z. Zhang, et al., IEEE Trans. Comput.-Aided Design, *accepted* (2018).
[11] W. Yu, B. Zhang, et al., IEEE DATE, p. 1225 (2016).
[12] C. Zhang and W. Yu, IEEE ASP-DAC, p. 756 (2014).
[13] K. Zhai, W. Yu and H. Zhuang, IEEE DATE, p. 1661 (2013).
[14] Z. Xu, W. Yu, et al., ACM GLSVLSI, p. 99 (2016).
[15] W. Yu, IEEE ICSICT, p. 227 (2016).
[16] S. Batterywala, R. Ananthakrishna, et al., IEEE VLSID, p. 129 (2006).
[17] M. Song, Z. Xu, et al., ACM GLSVLSI, p. 189 (2018).
[18] N. Revol and P. Theveny, IEEE Trans. Computer, 63, p. 1915 (2014)
[19] W. Yu, Z. Xu, B. Li, et al., IEEE Trans. Comput.-Aided Design, 37, p. 1711 (2018).