

# An Efficient Tool For Extraction of Interconnect Models in Submicron Layouts

P. Maffezzoni, A. Brambilla, A. L. Lacaita

DEI, Politecnico di Milano, p.za L. Da Vinci 32, 20133 Milano, Italy

## 1 Introduction

During these last years the width of interconnects in silicon layouts has been reduced to less than  $0.25\mu m$ , the number of metal levels has been increased up to five and the contributions to parasitic capacitances has become dominant. This is why a renewed interest has been triggered on the development of improved extraction methods and recent literature reports a large number of proposals. Capacitance evaluation is a well known and studied problem which requires the solution of the Laplace's equation for the electrostatic potential. Many codes have been developed so far, based on finite element or finite difference methods, solving the Laplace's equation even in a 3D geometry; however they can be seldom applied to real life silicon layouts, since due to the layout geometrical complexity they easily run out of computer resources. The bottleneck is usually circumvented by avoiding the solution of Laplace's equation for the entire layout and trying to break the problem into many elementary geometries (sub-problems) or by not considering the real 3D geometry. Those are the so called "quasi 3D" or "2.5D" extractors. In this paper we present an alternative approach for parasitic extraction which solves the Laplace's equation considering the entire layout without making any geometrical simplification or breaking. It is based on an extended version of the Floating Random Walk algorithm (FRW) [2]. The use of FRW is not novel, but so far it has been limited to the extraction of the total capacitance of the interconnect. Here we show how to use FRW together with the Picard-Carson iterative procedure [1] in order to efficiently get a compact model of an interconnect.

## 2 The Picard-Carson method

Given the interconnect shown in fig. 1 of total length  $L$ , it is possible to represent its behavior at the input ( $x = 0$ ) and output ( $x = L$ ) terminals in the Laplace domain by means of the transmission matrix  $T$  relating the voltage  $V(L)$  and the  $I(L)$  current, measured at the end of the interconnect, with the corresponding values  $V(0)$ ,  $I(0)$  at the input

$$\begin{bmatrix} V(N) \\ I(N) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} V(0) \\ I(0) \end{bmatrix}$$

The  $A$ ,  $B$ ,  $C$ ,  $D$  entries can be computed using closed forms [1]:

$$\begin{aligned} A(s) &= I + A_1s + \dots + A_Ns^N + \dots & B(s) &= B_1 + B_2s + \dots + B_Ns^{N-1} + \dots \\ C(s) &= C_1s + C_2s^2 + \dots + C_Ns^N + \dots & D(s) &= I + D_1s + \dots + D_{N-1}s^{N-1} + \dots \end{aligned}$$

where  $A_i = \zeta^{(2i)}$ ,  $B_i = \zeta^{(2i-1)}$ ,  $C_i = \psi^{(2i-1)}$ ,  $D_i = \psi^{(2i)}$ . Saying  $c(x)$  and  $r(x)$  the line capacitance and resistance per unit length, the above terms are derived by means of the iterated integrals:

$$\zeta^{(h)}(x) = \int_0^x \psi^{(h-1)}(\tau)r(\tau)d\tau \quad ; \quad \psi^{(h)}(x) = s \cdot \int_0^x \zeta^{(h-1)}(\tau)c(\tau)d\tau \quad x \in [0, L]$$

where  $\zeta^{(0)}(x) = \psi^{(0)}(x) = 1$  represent the initial guess. The  $r(x)$  function is known, while  $c(x)$  is evaluated using the FRW algorithm, as summarized in the following section.

### 3 Parasitic capacitance extraction and FRW

The self capacitance of an electrode is equal to the electrical charge induced on it when its potential is raised to 1V and the other electrodes and ground plane are at 0V. The charge on the electrode is in turn equal to the flux of the electrical displacement vector through the  $\Sigma$  Gaussian surface contouring the interconnect. In this way the capacitance estimation is recasted in terms of evaluation of flux. The same approach can be adopted for the estimate of  $c(x)$ , which, in FRW, is done in a statistical way. Following a Monte Carlo procedure pick a point  $\mathbf{p}$  on the  $\Sigma$  Gaussian surface as shown in fig. 1 and start a random walk from it by choosing hops on maximum spheres that do not intersect other interconnects or ground plane. Selection of  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$ ,  $\mathbf{E}$  points (see fig. 2) is randomly done according to a probability density function related to the Green function that links potential or electric field displacement in the center of each sphere to potentials on its surface. A possible random walk that end on an electrode in the  $\mathbf{E}$  point is shown in fig. 2. The average value of the potentials collected on electrodes gives an estimation of the  $c(p)$  value.

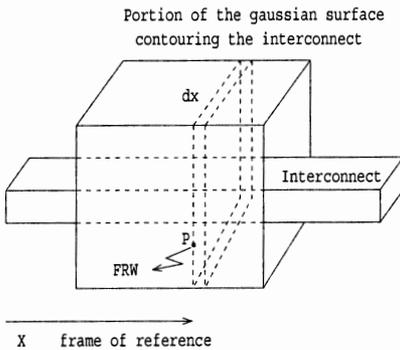


Fig. 1. A portion of the Gaussian surface through which the average value of the electric field  $E_\eta$  is estimated. Generic FRW is started from  $\mathbf{P}$  point.

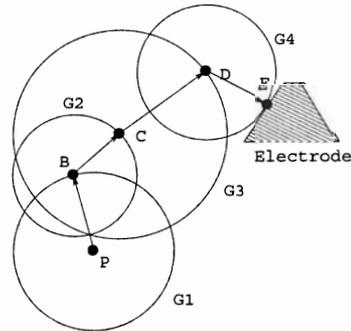


Fig. 2. A FRW walk composed of discrete hops; a walk ends when point (for example  $\mathbf{E}$ ) lies on the ground plane or on an interconnect.

It can be shown that *the recursive computation of the entries of the  $T$  matrix in done through this statistical sorting approaches, the exact values as the number of floating random walks goes to infinity.* In practice a high but finite number

of iterations (e.g. 10000) are performed till the estimated statistical error falls below a suitable relative value (e.g. 5%).

## 4 Simulation results

The efficient extraction of the lumped model and the statistical evaluation of the above integrals make the new algorithm very efficient and suitable for applications to industrial 3D layouts. Figs. 3, 4 show a performance comparison between SIMPLEX [3] and our approach, denoted to as FASTNET, in terms of CPU time and memory allocation. In both Figs. bars on the left refer to a simple cell embedded in the layout of Fig. 5. During parasitic extraction of this cell, FASTNET allocates more memory with respect to SIMPLEX due to the overhead for storing the layout structure. Bars on the right in the same Figs. show the performance in the analysis of the more complex layout in Fig. 5. In this case FASTNET memory allocation and CPU time remain almost the same, while SIMPLEX requires much more resources: the CPU time grows exponentially and also the memory allocation is about three time larger. Note that in both cases FASTNET is even more accurate ( $< 5\%$ ) than SIMPLEX ( $< 10\%$ ). *In general FASTNET CPU time remains almost independent of the dimensions of the circuit layout thus becoming extremely efficient in handling parasitic extraction in realistic industrial layouts.*

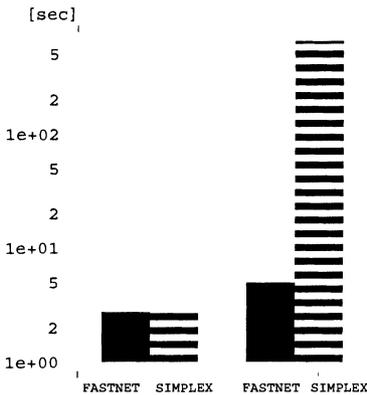


Fig. 3. CPU time required by FASTNET and SIMPLEX for computing the equivalent lumped models of the interconnects of a simple cell embedded in the layout of Fig. 5 (left two bars) and of the more complex layout of Fig. 5 (right two bars).

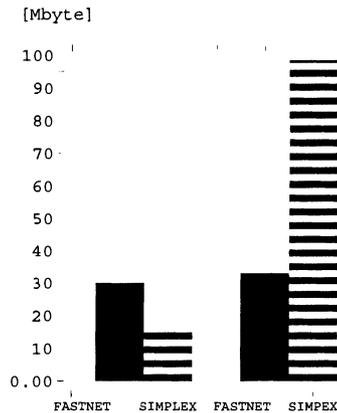


Fig. 4. Memory required by FASTNET and SIMPLEX for extracting the capacitance of nets of a simple cell embedded in the layout of Fig. 5 (left two bars) and of the layout in Fig. 5 (right two bars).

Fig. 6 compares the number of RC cells needed by SIMPLEX and FASTNET to model an interconnect and to reach the same accuracy in the delay evaluation. Note that since geometry of layout has not been partitioned by FASTNET the equivalent RC network is always composed of a few number of cells independently from line length  $L$ . Despite compactness this equivalent model has an accuracy comparable to that composed of 100 cells of SIMPLEX as shown in fig. 7.

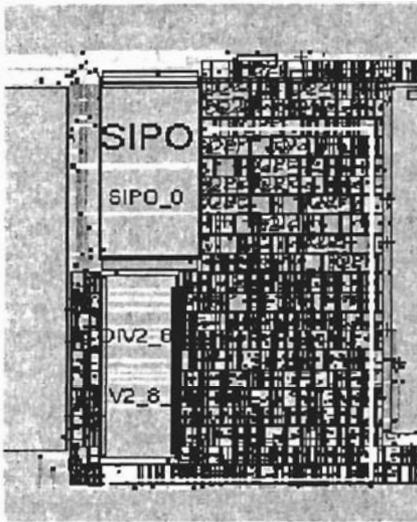


Fig. 5. Typical interconnect (highlighted in white) in a  $0.25\mu\text{m}$  6 metal level technology layout.

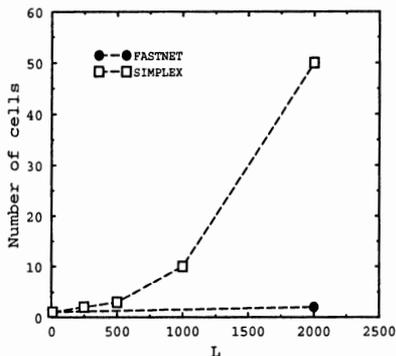


Fig. 6. Number of lumped cells required by FASTNET and SIMPLEX to get an accuracy  $< 1\%$  versus the  $L$  line length.

## References

- [1] M. S. Ghauri, J.J. Kelly, "Introduction to Distributed-Parameter Networks with Application to Integrated Circuits", Holt, Rinehart and Winston, New York 1968.
- [2] G. M. Royer, "A Monte Carlo Procedure for Potential Theory Problems", IEEE Trans. on MTT, vol. MTT-19. No. 10, pp. 813-818, October 1971.
- [3] SIMPLEX reference manual, (1999).

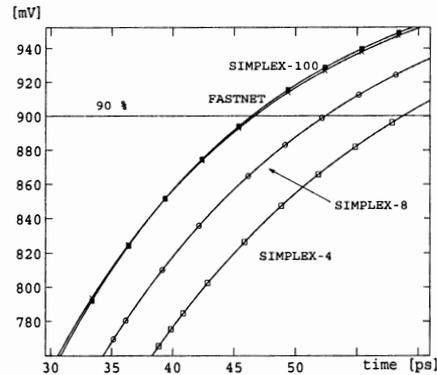


Fig. 7. Leading edge of the voltage pulse at the end of the line in Fig. 5, as derived by SIMPLEX when using 100 RC cells (crosses) and by forcing SIMPLEX to use only 8 (circles) and 4 cells (empty squares). Result obtained with the two cell equivalent network (filled squares) extracted by (FASTNET).

## 5 Conclusions

A fast and accurate method to compute and reduce the lumped RC model of a distributed net has been presented. It is based on an extension of the Floating Random Walk method and demonstrate that statistical methods are attractive to these applications since they need limited memory resources and are able to deal with large 3D industrial layouts.

## 6 Acknowledgments

The work has been supported by CNR (Italian National Research Council) in the frame of the project MADESS-II.