Fast Interconnect Parasitic Extraction in Deep Submicron Using Bin-Based Algorithm

Fabio Ferreira and Ricardo Reis
UFRGS – Instituto de Informática
Av. Bento Gonçalves, 9500 – Bloco IV
Caixa Postal 15064 – 91501-971 - Porto Alegre – Brazil
e-mail: {fklein, reis}@inf.ufrgs.br

Fernando Moraes
PUCRS – Faculdade de Informática
Av. Ipiranga, 6681 - Prédio 30
90619-900 - Porto Alegre - Brazil
e-mail: moraes@inf.pucrs.br

Abstract **- In this paper we present a fast interconnect capacitance extraction tool (wire extractor) LASCA, which considers the lateral coupling, ground and crossovers capacitances. Using the bin-based algorithm to extract connectivity, empirical formulation and a simple 2½D methodology, we provide a good trade-off between accuracy and efficiency when compared to traditional layout extraction. Comparisons between our wire extractor with the Diva Extractor (Cadence design Systems), give an average difference of only 5% in the final delay of some benchmarks, being up to 30 times faster than Diva.**

I. INTRODUCTION

The performance of integrated circuits has been increasing exponentially due to the device and interconnection scaling, new micro-architectures and design methodologies. As the device physical size decreases, the time spent by the signals to travel between gates is equivalent or greater than the gate delay for deep-submicron technology [1].

These parasitic effects and the number of interconnection layers (5 to 6) represent a new challenge for IC design tool developers. Cell library pre-characterization is no more sufficient to predict delay at higher abstraction levels. New tools must be developed to quickly predict parasitic capacitances, feed backing, e.g., the logical synthesis tool.

Currently, analysis and verification procedures in the design flow consume the same amount of time as synthesis, placement and routing. As the number of parasitic elements to consider during design verification increases in deep submicron technologies (coupling capacitances and more metal layers), fast and accurate estimation tools must be developed to keep a reasonable CPU time spent by CAD tools (minutes or few hours instead of days or weeks).

We present in this paper a fast and accurate wire extraction tool, called LASCA [2]. We take a physical description of each net, computing the coupling capacitance and substrate capacitance. Using a bin-based algorithm [3] to extract the connectivity, empirical formulation and a simple 2½D methodology, we provide a good trade-off between accuracy and efficiency when compared to traditional layout extraction (2D extraction).

This wire extractor is part of a physical synthesis environment. A layout synthesis tool, TROPIC [4], coupled with Synopsys, generates the layout without cell libraries from behavioral VHDL. LASCA extracts the parasitic capacitances and resistances, allowing in this way a fast power/delay estimation of the macro-cell.

This paper is organized as follows. Section II presents the connectivity extraction algorithm. Section III introduces the parasitic capacitance extraction method. Section IV presents the preliminary results and Section V our conclusions.

II. CONNECTIVITY EXTRACTION

Connectivity analysis is the first task executed by an electric extractor tool. Usually, this verification is executed over a flattened layout. The problem is how to handle a layout data structure containing up to 10^6 polygons. Some algorithms can handle such huge complexity, such as scanline, corner-stitching, quad-trees and bin-based [3].

The bin-based algorithm presents the best trade-off between memory consumption and CPU time when compared to the other ones. Even it consumes more memory than the scanline algorithm, since the complete circuit is stored in memory, it allows a fast connectivity analysis. The implemented bin-based algorithm can treat 100,000 polygons in less than 10 seconds.

The circuit is mapped over a virtual grid in the bin-based algorithm. This grid split the circuit into rectangles, called *bins*. The set of *bins* composes a NxM bidimensional matrix. Each bin will contain all polygons intersecting it, storing then in linked lists (Fig 1).

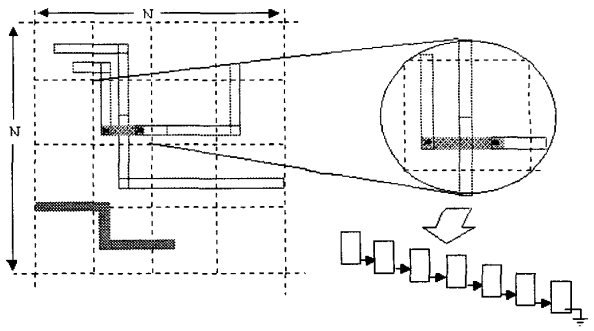

Fig 1. Bin-Based Algorithm

If a polygon intersects more than one bin, it will be stored in each bin it intersects. To reduce the memory usage and to

Proc. 43rd IEEE Midwest Symp. on Circuits and Systems, Lansing MI, Aug 8-11, 2000
0-7803-6475-9/00/$10.00 ©IEEE 2000

avoid redundant information, only pointers to polygons are stored into the bins. There are two main data structures: (*i*) the list of all polygons, (*ii*) the bin matrix, storing polygon pointers.

The space complexity of the bin-based algorithm is $O(b.n)$, where b is the total number of bins, and n is the number of polygons.

After the data structure creation, it is verified the connectivity between all polygons. This task is performed as follows: each polygon not analyzed starts a new net. This starting polygon is the first component of a new net. All non-visited polygons connected to the starting polygon are set as "*new starting polygons*", and added to the net. Recursively, all "*new starting polygons*" are used as "*new polygons*", until no more polygons can be added to the net.

The performance of this algorithm is a function of the bin size. If the bins are too large, the linked lists will contain a huge number of polygons, increasing the CPU time. If we have small bins the memory usage is too important, since a lot of replicate pointers will be required.

From our experimental data, using a 0.25 μm technology, we fixed the bin size as a 2μm x 2μm square. This size results in short linked lists, containing 1 to 20 polygons (average), and consequently in a small CPU time for wire extraction.

To improve the performance of the algorithm, the matrix size is a function of the circuit size. If the grid size is fixed, we can have an important number of empty bins for small circuits or bins containing hundreds of polygons for large circuits, decreasing the algorithm performance.

Fig. 2 shows the polygon distribution histogram for the C7552 circuit (ISCAS85 benchmark). The C7552 benchmark was generated using the TROPIC [4] layout synthesis tool, with three metal layers, in a 0.25 μm technology. It has 14,736 transistors and 265,563 polygons. The resulting circuit size is 933.8 μm x 364.2 μm, corresponding to a matrix size with 466 x 182 units (84812 bins).

In Table 1 we can observe that 86.8% of bins has less than 16 polygons, due to the choice of the bin size (2μm x 2μm) and the matrix size proportional to the circuit size.

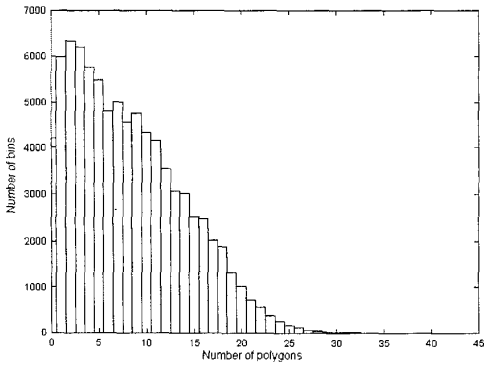

Fig. 2. Polygon distribution histogram (C7552 benchmark).

TABLE 1
POLYGONS DISTRIBUTION FOR THE C7552 BENCHMARK
(TOTAL NUMBER OF BINS = 466 X 182 =84812)

Number of polygons	Number of bins	% (bins/ total nb bins)
0 (bins empty)	4199	5.0%
1-5	29716	35.0%
6-10	23399	27.6%
11-15	16282	19.2%
≥ 16	11216	13.2%

Then, when we need to find connected polygons, we must search in a list containing less than 16 polygons (typical case), instead of 265563 polygons (circuit C7552).

The other analysis concerns the correct choice of bin size, i.e., the average number of bins used by each polygon. In the C7552 benchmark circuit, we have 97% of nets using 1-5 bins. The other nets are long wires, using up to 100 bins (like clock and reset nets).

As showed in Fig. 2 and Table 1, to find connected polygons in one bin we search in a list containing less than 16 polygons (typical case). Consequently, to find all connected polygons to a given polygon, the search space is 5 times 16, or 80 polygons instead of all layout polygons. Observe that this is the typical search space, there are long nets where the search space can be very huge, but these longs nets represents less than 3% off all nets.

Fig 3 presents the CPU time as a function of the number of polygons. These results were obtained from the ISCAS85 benchmarks, in a SUN Ultra Sparc 10.

The solid line represents $O(n)$ and $O(\sqrt{n})$ complexity of 100 polygons/second, respectively. This CPU time consider the connectivity analysis and capacitances extraction.

These results confirm the efficiency of bin-based algorithm implemented.

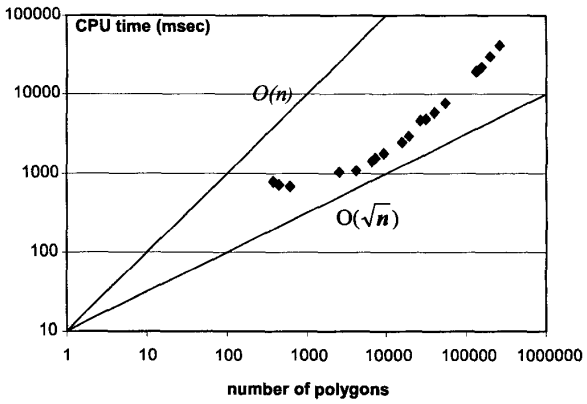

Fig. 3. Extraction time versus number of polygons, in a SUN Ultra Sparc 10.

III. CAPACITANCE EXTRACTION

Once the connectivity extraction is finished, we extract the parasitic capacitances. There are three capacitance components at each node:

- Overlap capacitance (C_{over}) - due to the overlap between two conductors in different planes. They are C_{21a} and C_{23a} in the Fig. 4.
- Lateral capacitance (C_{lat}) - is the capacitance between two conductors in the same plane, C_{22lat} in Fig. 4.
- Fringing capacitance (C_{fr}) - due to the coupling between two conductors of different planes, C_{23fr} and C_{21fr} in Fig. 4.

The LASCA tool compute the line-to-ground, line-to-line and crossover capacitances, which are defined by the components above described.

The line-to-ground and crossover capacitances are calculated by the traditional formulation based in the area and perimeter of the interconnections.

However, the line-to-line capacitance is a function of the distance between two connections and the thickness of the layer. While the thickness is a function of technology and the layer considered, the separation between wires is the main factor, depending of the layout. As the space between two wires can have a large variation, we can not use the capacitance given by the technology rules, because it is restrict to the minimum distance of separation. We use the empirical formulation described in [5].

The capacitances Cij (i,j=1...n, where n is the number of conductors) in the Fig. 4, are calculated by an empirical formulation. It was used the 3D field solver FastCap [6] to validate the equations.

It was also used a simple and accurate 2½D methodology described in [7]. It is based on five assumptions validated through detailed experiments with a 3D field solver over 0.50μm, 0.35μm and 0.18μm processes.

Briefly, for each connection in the layer *i*, we analyze the immediate neighbor in the same layer, all crossunders in the layer i-1 and all crossovers in the layer i+1, treating the layers i±2 like ground planes.

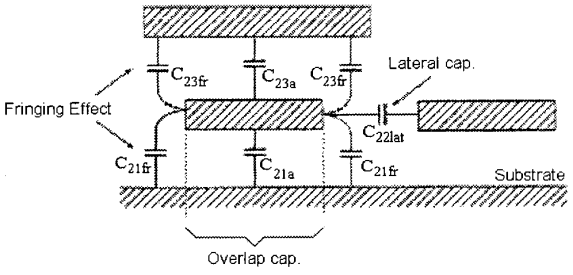

Fig. 4. Capacitance Model

IV. RESULTS

To verify the accuracy of the implemented wire extractor, we first generate some benchmarks using the TROPIC layout generator [4]. The resulting layout (cif file) was extracted with our tool, LASCA, and with Diva[1]. Both extracted netlists were simulated using the HSPICE simulator. Fig. 5 gives an overview of the proposed method to validate the LASCA tool. The layout synthesis tool computes the transistor area and perimeter.

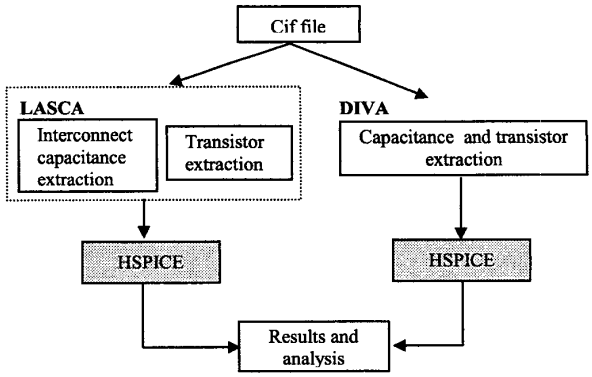

Fig. 5. Design Flow to validate the LASCA tool.

Table 2 shows in the first four columns the circuit name, the number of transistors, nets and polygons. The fifth and sixth column presents the circuit delays with the parasitic capacitances obtained with LASCA, considering first only capacitance to ground (Cground) and after both ground and coupling capacitances (Cground+Ccoupling). The last column shows the delay obtained with the extracted capacitances with Diva.

Comparing the first column (ground capacitance) to Diva (our reference), we get pessimistic delay evaluation, with an average error of 8.25% (worst case: 15%).

When considering all capacitances (coupling and ground) the average error is 3.63%, and the maximum error is below to 10%. In Fig. 6 we plotted the results of Table 2.

TABLE 2
DELAY FOR CIRCUITS EXTRACTED WITH LASCA AND DIVA

Circuit	xtors	Nets	Polygons	Delay(ns)		
				C_{ground}	$C_{ground}+C_{coupl}$	Diva
Adder	28	13	449	0.3624	0.3847	0.36783
Addergate	40	15	635	0.4042	0.4304	0.4247
Alu	260	94	4291	0.985	1.1167	1.1056
Alugate	432	117	6503	1.0984	1.2401	1.2208
Rip 16bits	448	163	7270	3.3713	3.7872	3.8571
Cla 16bits	528	215	9066	2.5539	2.8737	2.6234
Rip 32 bits	896	323	15503	7.0158	8.2248	8.2730
Cla 32bits	1056	427	19051	5.3084	6.2971	5.7966

Technology: 0.25μm, 3-metal layers with stacked contacts. Transistors size: w=2μm, l=0.25μm.

[1] Diva is a registered trademark of Cadence Design System, Inc.

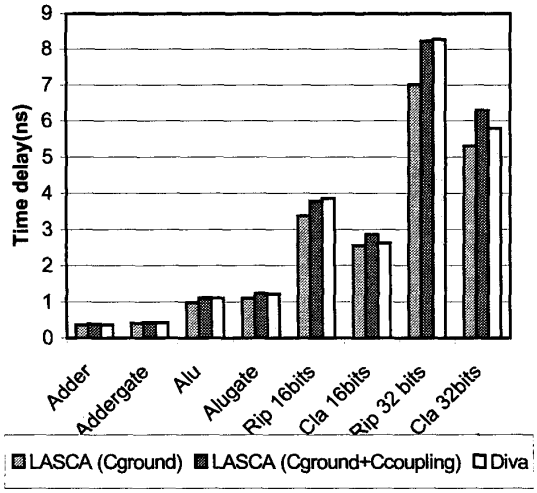

Fig. 6. Delay comparison for circuits extracted by Diva and LASCA.

Table 3 compares the CPU time of the LASCA extractor to the Diva extractor. Again, CPU time for ground capacitance and ground plus coupling capacitance was considered. The CPU time to extract capacitances in circuits containing more than 100,000 polygons are in average 25 times faster with our wire extractor. For a circuit containing 265,563 polygons we extract all capacitances in less 45 seconds, while the Diva extractor takes 19 minutes[2].

Usually, the technology file for commercial extractors is too complicate, since they must consider all technology parameters. The technology description used by LASCA is very simple. It is required only area and peripheral capacitance, thickness and distance to the ground plane of each layer, and the maximum distance between two connections to calculate the lateral coupling capacitance. The layout generator tool gives the diffusion area, needed to compute the diffusion capacitance.

TABLE 3
CPU TIME FOR LASCA AND DIVA (SUN ULTRA SPARC 10)

Circuit	Xtors	Polygons	CPU time (sec)		
			C_{ground}	$C_{ground} + C_{coupl}$	Diva
C17	24	386	0.752	0.767	12
C432	150	2564	0.928	1.036	19
C1355	2244	38841	4.005	5.824	132
C3540	7154	129918	10.886	19.542	481
C6288	10112	155885	12.906	21.852	595
C7552	14376	265563	23.791	42.506	1160

[2] It was computed the time to convert the CIF file to Cadence database, extraction and netlist generate.

V. CONCLUSIONS AND FUTURE WORKS

Our primary goal is to show that it is possible to quickly extract the coupling and ground capacitances with a reasonable accuracy. This extractor is integrated with an automatic layout generator, TROPIC [4], resulting in an integrated environment for fast on-the-fly implementation of macro-cells and for accurate parasitic evaluation.

We are working now in resistance extraction tool. The goal is to provide to the user different files, each one containing the extracted resistances with one different RC model: pi, L and T.

REFERENCES

[1] Semiconductor Industry Association, "The National technology roadmap for semiconductor," Available in http://notes.sematech.org/ntrs/PublNTRS.nsf, 1997.

[2] F. Ferreira, F. Moraes and R. Reis, "LASCA – Interconnect parasitic extraction tool for deep submicron IC design," in SBCCI XIII Symposium on Integrated Circuits and Systems Design, 2000, Manaus, in press.

[3] N. Sherwani, Algorithms for VLSI Physical Design Automation, Kluwer Academic Publisher, 1993.

[4] F. Moraes, M. Robert and D. Auvergne, "A Virtual CMOS library approach for fast layout synthesis," in: VLSI, 1999.

[5] J. H. Chern, J. Huang, L. Arledge, P. C. Li and P. Yang, "Multilevel metal capacitances models for CAD design synthesis systems," IEEE Electron Devices Letters, v.13, n.1, pp. 32-34, February 1992.

[6] K. Nabors and J. White, "FastCap: A Multipole accelerated 3D capacitance extraction program," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol 15, n 1, pp.58-66, January 1996.

[7] J. Cong, A. B. Kahng, D. Noice, N. Shirali and S. H. Yen, "Analysis and justification of a simple, practical 2 1/2D capacitance extraction methodology," UCLA Computer Science Technical Report 970013, 1996.

[8] CADENCE™, Diva Interactive Verification Reference, 1998.