PRE-LAYOUT PERFORMANCE PREDICTION FOR AUTOMATIC MACRO-CELL SYNTHESIS

F.Moraes[1], R.Reis[1], L.Torres[2], M.Robert[2], D.Auvergne[2]

[1]CPGCC - Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves 9500 - Campus do Vale - BlocoIV
CP 15064 - CEP 91501-970 Porto Alegre - BRASIL
Phone: (+55) 51 336 83 99/Ext. 6828 - Fax: (+55) 51 336 55 76
e-mail: moraes@inf.ufrgs.br

[2]LIRMM - Université de Montpellier II - 161 rue Ada
34392 Montpellier cedex 5 - FRANCE
Phone: (+33) 67 41 85 76 - Fax: (+33) 67 41 85 00

ABSTRACT

In this paper we present an approach allowing the area and delay prediction of macro-cells to be used in automatic layout synthesis tools. These parameters are of great importance in IC design. They allow to guide the floor planning phase of an IC by specifying the number of rows and the aspect ratio of the macro-cell or to explore the power delay trade-off by selecting the size of the transistors (before layout generation) to satisfy the user constraints. We propose here a calibration method based on specific benchmarks to determine post-layout parasitic contribution at pre-layout level. Experimental results are given where we compare predicted and simulated post-layout performances of various circuits for different sizing alternatives.

I. INTRODUCTION

Nowadays the design of digital circuits can start from a behavioral description, giving to the designers a nice flexibility at the architectural level. The standard-cell approach is accepted as a good solution for the layout synthesis problem, but, the technology mapping is limited by cells found in the library and mostly by the sizing possibilities which are restricted to the templates of each cells. For these reasons we have developed a layout macro-cell generator [1], having as main advantages:

- Technology independence. All design rules are input parameters for the generator. We can generate the same circuit for different design rules (for example 0.7 or 0.5 μm), by changing just the technology parameters description file.
- No cell libraries. All gates are generated on-the-fly, considering the specific needs of the circuit.
- Free technology mapping. Possibility to use all complex gates (AOIs) combinations [2]. The advantages of using AOI gates are: area, delay and power reduction (when comparing with standard-cell approach).
- Transistor sizing possibility. This facility allows to meet the user constraints.
- Input at transistor or gate level. This allows an easy integration with high-level synthesis tools.

However, the automatic layout synthesis tools give no information about area and delay before layout generation. This handicap reduces the use of this kind of tools by VLSI designers. In order to circumvent this problem we show that for a given layout style it is possible to make an accurate area and delay prediction before layout generation. In this study we use the TROPIC [1] generator, which implements a circuit layout with a "*linear-matrix multi-row*" layout style.

This paper is organized as follows. In section II we present the area prediction approach. In section III we present two models of delay prediction and finally we discuss about future works in the conclusion.

II. AREA ESTIMATION

To estimate area, we propose an approach based on the knowledge of the layout style (linear-matrix multi-row style). The layout generation is divided in two steps: topological generation and compaction [3]. The first one, topological generation, starts from an electrical description and generates a symbolic layout. The main algorithms used in this phase are: basic cell extraction, partition into rows, cell generation, cell placement and routing. The translation of symbolic layout into real layout is performed by using a compactor.

The area estimation is done between the topological generation and layout compactor steps, since the CPU time expended by layout compactor is very important. After the topological generation we know the number of routing tracks and the width of the rows. With these two parameters, the number of rows (input parameter of the generator) and some design rules, we can use the following equations to predict area (figure 1 shows these parameters in a real layout):

$$\begin{cases} Y = tracks * step + row * (W_n + W_p) + (row+1) * W_{\text{supply}} \\ X = WIDTH * CDM + 2 * W_{\text{supply}} \end{cases}$$

Where:
row = number of rows
step = {contact width} + {metal to metal distance}
*tracks = $\sum$ routing lines + 2 * row*
(given by the topological generation)
W_n, W_p = width of N and P transistors
(for irregular sizes we take the maximum W for each row)
W_{supply} = width of supply lines
WIDTH =the symbolic width of the larger row
(given by the topological generation)
CDM ={contact width}+ {poly width}+{minimum source width}

Table 1 summarizes the results obtained with different kinds of benchmarks, where we compared the estimated area with the real ones (obtained after layout compaction). The obtained results indicate a 10% average accuracy in area prediction for designs with up to 700 transistors. The higher differences between estimated and real area are induced by irregularities in the layout, mainly due to the use of transmission gates.

0-7803-3073-0/96/$5.00 ©1996 IEEE

benchmark (CMOS 1μm)	Tr#	R#	L#	Estimated area	Final Area	ratio
Adder cell	28	1	7	**0.0054**	0.0062	0.88
LS 161	222	3	35	**0.0574**	0.0627	0.92
Ripple carry *	384	4	28	**0.0669**	0.0764	0.88
ALU	424	5	47	**0.0930**	0.1068	0.87
Carry Skip*	426	5	43	**0.1103**	0.1475	0.75
Carry select*	642	6	51	**0.1306**	0.1465	0.93
Carry Lookahead*	696	6	72	**0.1769**	0.1859	0.95

**12 bits adders; Tr# = Number of Transistors,*
R# = Number of rows, L# = Routing lines, Area in mm^2

Table 1 - Comparison between estimated and real area

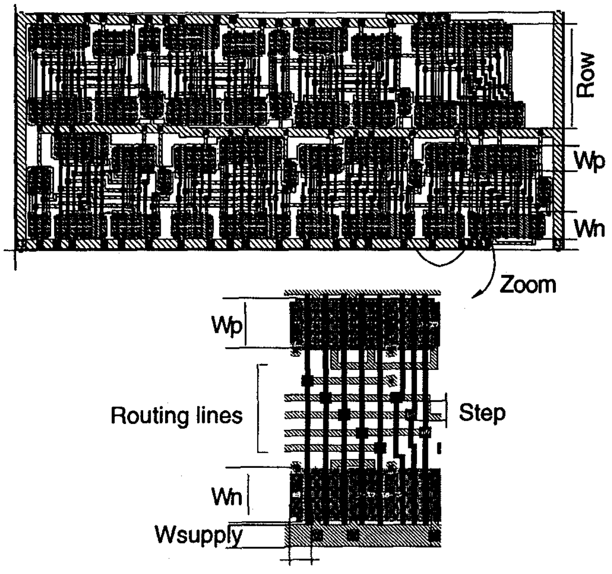


Figure 1 - Area estimation parameters

This information is useful at the floor planning phase of an integrated circuit [4]. The user can have quickly an estimation of the block size and change the number of rows to change its aspect ratio, trying to save area in the final circuit.

If we know the average transistor density (Tr/mm^2) for a given technology, we can have also a good area estimation before topological generation (i.e., before placement and routing). This method is quite accurate (in order of 10 to 20%), but it needs the generation of a set of benchmarks and it doesn't give the width and height of the block, only the area.

III. DELAY ESTIMATION

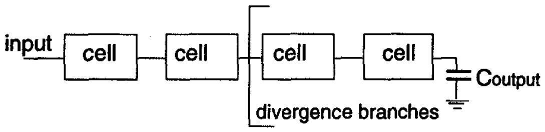
We present in this section two methods to obtain delay estimation:

- Analytical method, based on the explicit formulation of delays [5], using the concept of critical paths [6].
- Global method, based on the generation and simulation of a set of benchmarks, to determine rules for the post-layout delay estimation. The users of the system realize the pre-layout simulation of the circuit. The obtained pre-layout delay is then multiplied by a factor K, determined from a simulation of benchmarks.

The first method doesn't need electrical simulation and input vectors, only the electrical parameters. The difficulty of this method is to find the correct critical path, to have an accurate performance prediction. The second method allows to estimate the parasitic contribution before layout generation, but needs electrical simulation and input vectors.

III.1. Analytical method

Until now, the automatic layout synthesis tools are not widely used due to the difficulty to make an accurate delay prediction before layout generation. We can overcome this problem by using an explicit formulation of delays [5], with a static analysis approach to evaluate the critical paths. The principle of delay prediction (figure 2) is given bellow.



$$t_{critical\ path} = \max(t_{rise}, t_{fall})$$
$$t_{rise} = t_{rise(k)} + t_{fall(k-1)} + \dots + t_{rise(0)}$$
$$t_{fall} = t_{fall(k)} + t_{rise(k-1)} + \dots + t_{fall(0)}$$

Figure 2 - Delay calculus strategy

1) Evaluation of the load at each node of the circuit, from layout modeling of the active and parasitic capacitance. The parasitic capacitance is estimated from the layout style, which is quite regular and allows to calculate the input and output capacitance of each cell.
2) Calculation of the delay of each cell, using design parameters such as the cell structure (number of serial transistors), its environment (input and output capacitance), the size of transistors and the technology parameters.
3) Finding the static critical path of the circuit (DFS and BFS algorithms are used to find the longest static path). As an initial approach we don't take into account the false path problem [7][8].
4) Calculation of the critical paths fall and rise time, using the delays of each cell. The resulting values are multiplied by a factor (1+A), representing the input slope contribution to the overall delay [9].

In table 2 we compare results obtained using this prediction methodology to values measured from Spice simulations. As shown, the prediction is quite good. The values of delays are obtained for each solution of transistors size. In this way it is possible, before layout generation, to explore the delay-area (power) trade-off in order to get initial solution not too far from the user specifications. As an example, figure 3 illustrates the space of solutions for delay as a function of transistor sizes (w) and load (in fF) for a 4-bit ALU. This shows the possibility to choose an initial sizing solution. The CPU time for these examples is lower than 1 second (Sun Sparc 20).

Circuit	Tr#	Delay prediction	Spice simulation	*simulation* / *prediction*
2 bit adder	28	2.8	2.4	0.87
4-bit ALU	260	6.8	7.6	1.11
ripple carry	336	12.7	13.6	1.07
carry select	540	7.3	7.6	1.04
CLA 12 bits	624	14.4	16.8	1.17

Table 2 - Post-layout simulation (Hspice-level 6) versus delay prediction (ns)
(Technology: CMOS 1 μm, parasitic capacitance estimated for delay prediction: 50 fF, output load: 500 fF)

As shown, it appears possible to define a high level performance prediction (or initial solution definition) with a quite good accuracy. However, important limitations of this analytical method are given by the real determination of critical paths and the accurate evaluation of routing capacitances.

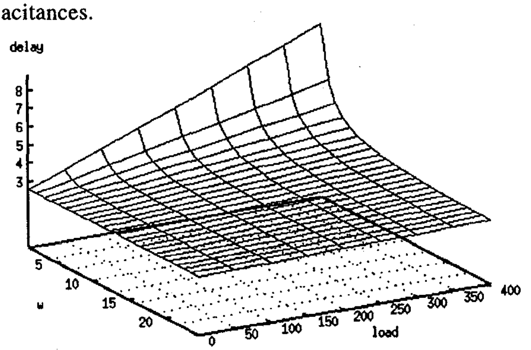


Figure 3 - Trade-off between transistor width, load and delay for a 4-bit ALU

As an alternative, and using these results, we propose in the next part a global method allowing macro-cell performance prediction from a set of parameters defined on specific benchmarks.

III.2. Global method

We propose here to implement a set of benchmarks to determine rules for predicting post-layout delays. This is obtained by characterizing macro-cells generated from the TROPIC layout tool.

We first consider that delays can be written with the following derating law: $T_{hl,lh} = A + B\ C_l$, where T_{lh} and T_{hl} are the rising and falling delay time respectively, C_l represents the output load, B the last stage delay contribution, and A the intrinsic structural or inertial delay, summing up the contribution from all the stages preceding the output one.

A) Methodology:

- For this study, we consider first a set of benchmarks, which represents different implementations of 12-bit adder architecture (Ripple carry, Carry Select, Look Ahead, Brent & Kung adder, Carry skip). These architectures are implemented with different transistor sizing alternatives (W_n=W_p=W=12μm, 8μm and 4μm).
- We determine the critical path delay (Tc) for the pre and post-layout,
- Then, we calculate the average delay (respectively $\overline{X_B}$ and $\overline{X_A}$ for the pre and post-layout) on the twelve outputs as well as on the carry for each adder (for C_l=0).
- Finally, we determine the ratio $K = \overline{X_A} / \overline{X_B}$ (post/pre-layout delay), which is considered as the ratio between the pre and post-layout performances.

Results are given in Table 3.

Wn=Wp= 12μm	pre-layout		post-layout		
	Tc	$\overline{X_B}$	Tc	$\overline{X_A}$	K
Ripple carry	4.07+3.2c_L	2.3	7.66+3.2 C_l	4.42	**1.92**
Carry select	3.1+1.7c_L	2.2	6.24+1.7 C_l	4.85	**2.2**
Look Ahead serie	2.91+3.2c_L	1.93	6.9+3.2 C_l	4.65	**2.4**
Look Ahead //	3+3.2c_L	1.81	6.79+3.2 C_l	4.2	**2.32**
Carry-Skip	2.7+3.2c_L	1.84	6.76+3.2 C_l	4.35	**2.37**

Wn=Wp= 8μm	pre-layout		post-layout		
	Tc	$\overline{X_B}$	Tc	$\overline{X_A}$	K
Ripple carry	4.07+4.8c_L	2.3	8.67+4.8 C_l	5	**2.17**
Carry select	3.1+2.5c_L	2.2	7.37+2.5 C_l	5.34	**2.46**
Look Ahead serie	3+4.8c_L	1.81	8.31+4.8 C_l	5.57	**2.93**
Look Ahead //	2.91+4.8c_L	1.93	8.17+4.8 C_l	5.06	**2.81**
Carry-Skip	2.7+4.8c_L	1.84	7.5+4.8 C_l	5.03	**2.7**

Wn=Wp= 4μm	pre-layout		post-layout		
	Tc	$\overline{X_B}$	Tc	$\overline{X_A}$	K
Ripple carry	4.07+9.6c_L	2.3	11.3+9.6 C_l	6.47	**2.81**
Carry select	3.1+5c_L	2.2	10.34+5 C_l	7.51	**3.41**
Look Ahead serie	3+9.6c_L	1.81	12.3+9.6 C_l	7.09	**3.93**
Look Ahead //	2.91+9.6c_L	1.93	11.5+9.6 C_l	8.47	**4.45**
Carry-Skip	2.7+9.6c_L	1.84	10+9.6 C_l	7	**3.82**

Table 3 - Pre and post-layout delays (1μm technology)

We note that for each transistor sizing alternative the ratio *K* (post/pre-layout delay) is practically constant. This ratio represents the relative effect of the total parasitic capacitances (essentially diffusion and wire length capacitances), and is transistor width dependent.

The main idea is to use these results to determine a mathematical delay prediction model. For this reason, we calculate the average value of *K* over the different benchmarks, for each transistor sizing alternative. This gives:
- W = 12 μm, K_1= (1.92 + 2.2 + 2.32 + 2.4 + 2.37) / 5 = 2.24
- W = 8 μm, K_2 = (2.17 + 2.46 + 2.93 + 2.81 + 2.7) / 5 = 2.61
- W = 4 μm, K_3= (2.81 + 3.41 + 3.93 + 4.45 + 3.82) / 5 = 3.68

In the same way, we processed other examples for transistor widths equal to 2, 6, 10 and 14 μm. The resulting average *K* values are represented in figure 4. As a result, *K* coefficient has a variation with transistor size which can be defined as: $K_w = \frac{A}{W} + B$, where the coefficients A=8.64 and B=1.52, specific of the considered process, can be obtained directly from the curve of figure 4.

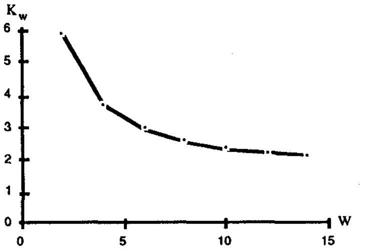


Figure 4 - Average contribution of the total parasitic capacitances

The observed variation gives direct illustration of the parasitic contribution of interconnect and diffusion capacitance. As mentioned in [10] the **B** term represents the transistor width dependent contribution of the diffusion. This term is strongly process dependent and gives a direct evaluation of the placement methodology and layout style used to generate the circuits.

B) Results

The benchmark used to validate the proposed model (for W=12, 8 and 4µm, for C_L =0) is constituted of: two 4x4 bits multipliers (Shift adders and Booth algorithm), an arithmetic unit-LS 181 and an ISCAS C435 circuit. Results are given in table 4.

The 2nd column of these tables gives the values of delay obtained from pre-layout simulation of the circuits. The 3rd column gives the predicted value of the resulting delay obtained by multiplying the data of the preceding column by the K_w value. This results in a prediction of post-layout delays. The 4th column gives the real values of delay as obtained from post-layout simulation of the circuits. These values are given for different regular sizing alternatives.

Wn=Wp=12 µm (C_L=0)

Benchmark	DEL (ns)	PRED (ns)	SIM (ns)	Simulation Prediction	Tr#	Area mm^2
Mult 4x4 dec	1.06	2.37	2.28	3.81 %	416	0.1
Mult 4x4 boo	2.04	4.59	5.09	9.8 %	824	0.25
ALU LS 181	2,41	5.4	6.04	10.6 %	425	0.11
C435	2.86	6.40	6.75	5.1 %	150	0.0417

Wn=Wp=8 µm (C_L=0)

Benchmark	DEL (ns)	PRED (ns)	SIM (ns)	Simulation Prediction	Tr#	Area mm^2
Mult 4x4 dec	1,06	2,77	2.60	6.6 %	416	0.09
Mult 4x4 boo	2.05	5.34	5.83	11.86 %	824	0.247
ALU LS 181	2,41	6.26	7.11	8.4 %	425	0.104
C435	2.86	7.47	9.00	17 %	150	0.0379

Wn=Wp=4 µm (C_L=0)

Benchmark	DEL (ns)	PRED (ns)	SIM (ns)	Simulation Prediction	Tr#	Area mm^2
Mult 4x4 dec	1,06	3.83	3.47	10.4 %	416	0.083
Mult 4x4 boo	2.05	7.54	8.62	15.7 %	824	0.245
ALU LS 181	2,41	8.84	10.5	15.5 %	425	0.100
C435	2.86	10.53	12.2	13.9 %	150	0.033

DEL = pre-layout delay (ns) - **PRED** = delay prediction (ns) - **SIM** = simulate delay post layou (ns)

Table 4 - Post-layout simulation versus delay prediction

As observed in these tables the good agreement obtained between predicted and simulated values confirms the validity of the approach. It appears possible to estimate post-sizing layout delay with an average accuracy of 10% for circuits with complexity up to 700 transistors. This approach gives indications on the contribution of diffusion and routing capacitance for different transistor widths, and constitutes an useful technique for evaluating layout or sizing strategies. It can be noticed that if this prediction may be considered as highly circuit architecture dependent, the wide selection of benchmarks circuits, used to define the K coefficient, allows a quite good evaluation of the post-layout performances.

CONCLUSION

In the era of coming very short channel technologies, automatic layout synthesis of macro-cells could represent an efficient solution to synthesize power-delay optimized circuits. Moreover tools to explore the trade-off space are necessary. We proposed here a method allowing, at pre-layout level, quite accurate prediction of real area-delay performances. For the process under study we show how to define, from calibration on specific benchmarks, a *K* derating coefficient allowing to take into account layout parasitic contribution as well as general rules for area estimation which can be used at higher level to guide the logical synthesis, or to select the best initial solution for area and delay trade-off.

Validation of this approach has been obtained by comparing, on different architectures of circuits, pre-layout predicted area and delay values to post-layout simulated ones, for different sizing alternatives.

This prediction method appears effective in controlling the size of macro-cells for performance driven placement and routing, or, in investigating with good accuracy the optimization space for power and delay, to full fill user constraints without entering the complete layout synthesis phase.

REFERENCES

[1] F.Moraes, N.Azemard, M.Robert, D. Auvergne, "*Flexible macrocell layout generator*", in Proc. of 4th ACM/SIGDA Physical Design Workshop, Los Angeles, 1993, p. 105-116.

[2] A.Reis, M.Robert, D.Auvergne, R.Reis. *Associating CMOS transistors with BDD arcs for technology mapping*. Electronic Letters, Vol. 31, No 14, July 1995.

[3] G.Cathebras, (1990), *Contribution à la compilation structurelle des circuits intégrés CMOS*, Phd Thesis, LIRMM, Montpellier, France, 1990.

[4] R.Reis, (1983), TESS: évaluateur topologique prédictif pour la génération automatique des plans de masse de circuits VLSI, Phd Thesis, INPG, Grenoble, France, 1983.

[5] D.Auvergne, N.Azemard, V. Bonzom, D. Deschacht, M.Robert, "*Formal sizing rules of CMOS circuits*". European Design Automation Conference, Amsterdam, February 1991.

[6] P.McGeer, R.K.Brayton, "*Efficient algorithms for computing the longest viable path in a combinational network*", 26th ACM/IEEE DAC, 1989, pp. 561-567.

[7] J.Benkoski, E.V.Meersch, L.Claesen, H, Man, "*Efficient algorithm for solving the false path problem in timing verification*", ICCAD 1987, pp. 44-47.

[8] D.H.Du, S.H.Yen, S.Ghanta, "*On the general false path problem in timing analysis*", 26th ACM/IEEE DAC, 1989, pp. 555-560.

[9] D.Auvergne, D. Deschacht, M.Robert, "*Input waveform slope effects in CMOS delays*". IEEE Solid State circuits, vol. 25, n. 6, dec. 1990.

[10] D.Auvergne, S.Amat, M.Mellah, N.Azemard, M.Robert. "*Evaluation of speed up strategy from gate performance modeling*". IFIP Workshop on logic synthesis, pp 193-208, Grenoble Dez. 1993.