



**HAL**  
open science

## **P.SIZE: A Sizing aid for Optimized Designs**

Nadine Azemard, Vincent Bonzom, Daniel Auvergne

► **To cite this version:**

Nadine Azemard, Vincent Bonzom, Daniel Auvergne. P.SIZE: A Sizing aid for Optimized Designs. EURODAC 1992 - European Design Automation Conference, Sep 1992, Hamburg, Germany. pp.160-166, 10.1109/EURDAC.1992.246248 . lirmm-00239396

**HAL Id: lirmm-00239396**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00239396>**

Submitted on 19 Mar 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# P.SIZE : a sizing aid for optimized designs

N.AZEMARD, V.BONZOM and D.AUVERGNE

LIRMM : LABORATOIRE D'INFORMATIQUE DE ROBOTIQUE ET DE MICROÉLECTRONIQUE DE MONTPELLIER  
UMR 9928 CNRS / UM2  
UNIVERSITÉ MONTPELLIER II, SCIENCES ET TECHNIQUES DU LANGUEDOC  
Place Eugène Bataillon, 34095 Montpellier cedex 5  
FRANCE  
Telex: 490944F, Fax: (+33) 67 52 14 96, Phone: (+33) 67 14 32 08

## Abstract

Transistor sizing at layout level is necessary to improve the overall performance of integrated circuits. We present in this paper the definition and the validation of a sizing aid, **P.Size**, integrated in a flexible cell generator. Based on a local optimization defined through an explicit formulation of delays, this sizing aid can be used to optimize real data paths, under constraint, with few CPU time requirements. Validations, through comparison with a mathematical optimization procedure and an industrial optimizer, are given.

## 1. Introduction

Improvements in IC fabrication technology, together with an increase in design performance, make the availability of an explicit optimization aid for chip performances essential.

Although semicustom approaches such as standard cell and gate array allow ASIC design with a fast turn around time, they are not suitable for advanced performance design techniques which require customized transistor sizes with respect to physical design issues such as layout, interconnection routing and timing.

As an attempt to overcome these limitations, we have proposed an adaptative cell generator approach. This generator [1], developed as part of the design automation project of the Microelectronics Department (LIRMM) of Montpellier, generates cells for custom design, with post layout evaluation of the performances. This system is able to lay out a transistor netlist into a two dimensional XY plane. The transistor netlist is defined from a post layout extraction or stick input prediction. After evaluation of the circuit performances on the extracted netlist, the next necessary step for this generator is to allow automatic sizing facilities of the transistors belonging to the identified critical path. We

present in this paper the definition and the validation of the local heuristic based sizing aid, **P.SIZE**.

A short review of prior work in this field will be given in part 2. A presentation of the sizing algorithm will be made in part 3. The way we treat the constrained circuit optimization problem is also discussed in this part. Part 4 will give a validation of this method with respect to a global mathematical optimization procedure and an industrial optimizer.

Cell optimization at layout level will be given in part V.

Finally a conclusion with trends for future work will be drawn up in part VI.

## 2. Prior Work

Many researchers have addressed the transistor sizing problem. The primary concern is to optimize the delays of a circuit according to given constraints imposed on area or power. Sometimes, the constraints imposed on delays result in area constraints; the final area gives the power performances and yield.

There is generally three techniques to determine global solutions to a constrained optimization problem. One is the mathematical approach [2,3,4,5,6], applying the classical augmented Lagrangian multiplier, a feasible direction method. This method requires to use the derivatives of delays with respect to the design parameters. If the objective functions and the realistic constraints are convex, this algorithm converges to the global minimum. However it is quickly limited to low complexity structures (without divergence branches) and needs good initial solutions.

The second approach is an heuristic or incremental method [7,8,9] in which sensitive transistors on critical paths are sized until the specifications are met. The third approach [10,11] is a combination of the two preceding ones in which an initial sizing solution is defined from an heuristic algorithm and a global solution is obtained from a mathematical optimization technique.

These methods are usually time consuming, limited to identified data path and have the accuracy of the delay model used to elaborate the cost function. In fact, they are "ad posteriori" methods which never give information for mapping selection or physical layout alternatives.

### 3 – Transistor sizing with P.SIZE

#### a – Sizing law definition

In order to circumvent the limitations due to the number of parameters to be optimized, the convergence difficulties induced by the divergence nodes, prohibitive CPU time, we have developed a local sizing heuristic [16] based on the explicit formulation of delays [12,13,14,15,16]. It gives fast solutions concerning the initial sizing of complex data paths as well as post layout optimization. This local heuristic has been defined as the global optimal solution for simple data paths made up of the structure to be sized and of its controlling inverter, used as a reference (minimum size reference inverter inducing optimal delay for minimum area). This approach allows direct backward processing of the data path (solving the divergence branches) and has been found to give a nearly optimal time area solution. However sizing laws given in [16] have been obtained empirically and found to result in non symmetrical delay values.

We present here the full determination of these laws under the constraint of symmetrical rise and fall times for the devices to be considered. As we already mentioned, for each structure, the solutions of the sizing problem involve the determination of the sizes of the transistors of the controlling device and the size of the N and P transistors of the gate under consideration. Explicit solution to the sizing problem has been obtained on a reduced space by imposing :

- identically sized transistors belonging to N or P serial arrays,
- P transistor sizes proportional to the size of the N transistors, (these two conditions result in regular implementations corresponding to layout solutions with minimum parasitic capacitances),
- symmetrical delays on the individual gates (rise time = fall time) for this sizing solution.

This last condition allows safe timing of data path without consideration of crossed combination of delays, which would impose to master analog complexity in time space.

Solving for the derivatives of the delays, with respect to the size of the N transistors, under the preceding constraints, gives the following explicit sizing laws for complex gates :

$$X_N = \frac{\sqrt{Y^*} \cdot (1 + 6K'(n_2 - 1))}{\left(1 + \frac{\mu_N}{\mu_P} + 6K'[(n_2 - 1) + \frac{\mu_N}{\mu_P}(n_1 - 1)]\right)^{\frac{1}{2}}} \quad (1)$$

$$X_P = \frac{\frac{\mu_N}{\mu_P} \sqrt{Y^*} \cdot (1 + 6K'(n_1 - 1))}{\left(1 + \frac{\mu_N}{\mu_P} + 6K'[(n_2 - 1) + \frac{\mu_N}{\mu_P}(n_1 - 1)]\right)^{\frac{1}{2}}}$$

where :

- $X_N, X_P$  are the respective widths of the N, P transistors normalized with respect to the width of the N transistor of the controlling device,
- $Y^*$ , is the load normalized with respect to the input active capacitance of the N reference transistor,
- $\mu_N/\mu_P$  and  $K'$  characterize the process and represent the technological parameters, ( $K'$  is the propagation constant, as given in [13]),
- $n_1, n_2$  respectively label the number of serial transistors in P and N array.

Equations 1 allow to optimally size any serial parallel combination of transistors organized in one logic level. Note that for  $n_1 = n_2 = 1$ , we obtain the general sizing law for inverters which corresponds to a global solution for a cascade of two stages.

#### b – Objective functions

The local sizing solutions obtained from equ 1, correspond to intermediate solutions between absolute minimum delay (derived from global minimization on the complete data path) and minimum area solutions (derived easily by forcing minimum sizes for all the transistors) These solutions can be used as initial values for global mathematical optimization procedures [2,3,4,5,6], from which global sizing solutions can be obtained.

However, it has been generally observed that around the global minimum, few percents of delay decrease is always paid by a large area increase. As a result local minimum may represent an interesting speed-area tradeoff. In order to explore the area effective space of solutions, we defined a weighted sum of objective functions as:

$$F = \alpha \cdot \Theta + (1 - \alpha) \cdot S \quad (2)$$

where :  $\alpha$ , varying between 0 and 1, represents the weighting or cost factor,

$\Theta$ , is the delay which depends on the ratio of the transistor widths (equation 6 of [16]),

$S$ , is the total active area (expressed as the sum of the widths of the transistors) which is directly correlated to the power dissipated in the device.

Note that :  $\alpha = 0$  corresponds to the minimum area solution, with maximum delay,

$\alpha = 1$  gives the local minimum for delay.

Any delay area trade off stands between these limits. Because minimum delay is very costly in area and power, it appears necessary to define optimal trade off allowing to satisfy constraints on delay with a minimum area.

For that we calculated the set of non inferior solutions  $X_N$ , by minimizing the weighted sum of both objective functions, delay  $\Theta$  and area  $S$  (power). By substituting into equation 2,  $\Theta$  and  $S$  with the explicit values calculated from /14/ and minimizing  $F$ , under the preceding conditions, we get:

$$X_N = \frac{\sqrt{\alpha} \cdot \sqrt{Y^*} \cdot (1 + 6K'(n_2 - 1))}{\left(1 + \frac{\mu_N}{\mu_P} + 6K'[(n_2 - 1) + \frac{\mu_N}{\mu_P}(n_1 - 1)]\right)^{\frac{1}{2}}} \quad (3)$$

$$X_P = \frac{\frac{\mu_N}{\mu_P} \sqrt{\alpha} \sqrt{Y^*} \cdot (1 + 6K'(n_1 - 1))}{\left(1 + \frac{\mu_N}{\mu_P} + 6K'[(n_2 - 1) + \frac{\mu_N}{\mu_P}(n_1 - 1)]\right)^{\frac{1}{2}}}$$

For each value of the  $\alpha$  trade off parameter, these equations define the sizing corresponding to the minimum delay and area available. Or, conversely, for a given constraint on delay, iterative resolution of equations 3 gives best value of  $\alpha$  minimizing the area.

In that way, non inferior local solutions are guaranteed for the delay area trade off.

#### c – P.Size : optimization module

These equations have been implemented in an automatic sizing module : **P.Size**, integrated in the flexible cell generator PRINT [1]. From an input file, defined from a post layout extracted netlist, P.Size generates an equivalent list with the updated sizes of the devices ( both netlists are SPICE equivalent ). As shown in the flow chart of Figure 1, this module is organized into 4 subroutines:

- *syntactic analyser* to build, from the input file, the data base to be used for sizing, (technological parameters are defined at this level),
- *topological analysis* for subcircuit decomposition and loaded conducting path identification,
- *transistor sizing* of the individual conducting paths,
- *generation* of the new SPICE compatible netlist.

Written in C language, P.Size represents 5000 lines of code; it allows the fast optimization of circuits through the superposition of local minimizations of delays. A validation of this heuristic will be done in the next part.

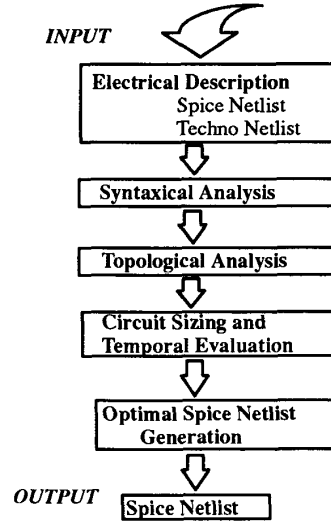


Figure.1: P.Size algorithm flow chart

#### 4. Validation of the local heuristic

To obtain an accurate validation of the local heuristic used in P.Size, we need a reference. For that we used a feasible direction method as a robust algorithm to define a global mathematical solution as an element of reference. We have selected ROSENBROCK's rotating coordinating scheme [17], which allows a direct search for a minimum solution without any calculation of derivatives. We have implemented this method as a prototype for the optimization of single path, in order to get reference values for the local sizing heuristic.

Figure 2 illustrates, for an array of inverters, the delay area trade off versus transistor size, defined in global (ROSENBROCK) and local (P.Size) optimization. Bold line has been obtained from equations 3 for different values of the cost parameter  $\alpha$ . The thin line is calculated from ROSENBROCK's procedure by searching minimum area solutions for given constraints on delay.

As expected, the price to be paid on delay by the simplicity of the local heuristic is nearly negligible, if we compare the local solutions we have obtained (bold line) to the global ones (thin line), which represent the locus of the non inferior solutions.

Note here that the local minimum of delay appears at values slightly different from the global ones, but less than 10% difference in delay trades half of the area.

These curves divide the delay area space into two parts:

- the part A represents the space where no sizing solutions can be obtained satisfying the delay area tradeoff,
- the part B, above the curves, contains all non optimal solutions as we can obtain from gate arrays or in a standard cell approach.

In Table 1, we compare the optimized sizing solutions from P.Size (local heuristic) to the ROSEN BROCK's one, for different cells controlled by an identical inverter (this guarantees real control situations and convexity of the sizing problem [16]). The upper half part of this Table corresponds to solutions obtained for minimum delay, the lower part of this Table corresponds to sizing solutions obtained under delay constraints. As shown, the good level of agreement obtained (less than 5% discrepancy), confirms the validity of our approach in defining sizing under constraint.

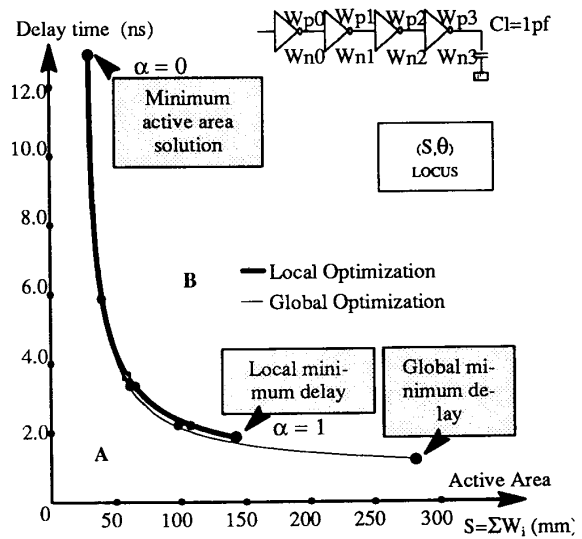


Figure.2: Comparison between sizing solutions, under constraints, obtained with global (thin line) and local (bold line) optimization methods. The thin line represents the locus of non inferior solutions.

#### Optimization without constraints

Structural type	Y	n1	n2	Rosenbrock Optimization				P.Size: Local Optimization				Rosenb / P.Size
				$X_N$	$X_P$	Thl	Thl	$X_N$	$X_P$	Thl	Thl	
Inverter	10	1	1	1.7	4.3	7.4	7.4	1.7	1.2	7.4	7.4	0%
Nand 6	10	1	6	3.9	3.2	16.8	16.8	3.5	2.9	16.8	16.8	0%
Nor 6	10	6	1	1.1	8.2	29.1	29.1	1.0	7.3	29.7	29.0	2%
Andori	10	3	6	3.1	4.7	19.9	19.9	2.4	4.0	21.0	19.3	1%
Inverter	50	1	1	2.9	7.3	16.2	16.2	3.8	9.5	16.3	16.3	0%
Nand 6	50	1	6	8.7	6.8	28.7	28.7	8.0	6.6	29.0	28.0	1%
Nor 6	50	6	1	2.0	15.6	44.8	44.8	2.2	16.3	44.3	44.8	0%
Andori	50	3	6	7.6	11.0	33.8	33.8	5.3	8.9	35.8	32.6	5%

#### Optimization with constraints

Structural Type	Y	n1	n2	Cons-traint	Rosenbrock Optimization				P.Size: Local Optimization				
					Tmin	$X_N$	$X_P$	active area	Tmin	$\alpha$	$X_N$	$X_P$	active area
Inverter	10	1	1	10	7.3	1.0	1.6	2.6	7.4	0.14	1.0	1.6	2.6
Nand 6	10	1	6	20	16.8	1.7	1.3	11.5	16.8	0.22	1.7	1.4	11.6
Nor 6	10	6	1	40	29.0	1.0	2.2	14.2	29.7	0.09	1.0	2.2	14.2
Andori	10	3	3	17	15.4	1.0	2.9	11.7	15.8	0.48	1.2	3.0	12.6
Inverter	50	1	1	20	16.1	1.6	4.0	5.6	16.4	0.18	1.6	4.0	5.6
Nand 6	50	1	6	32	28.7	4.6	3.5	31.1	29.0	0.34	4.6	3.9	32.1
Nor 6	50	6	1	50	44.7	1.1	8.9	54.5	44.8	0.30	1.2	8.9	54.6
Andori	50	3	3	30	27.7	2.6	7.0	28.8	28.3	0.56	2.9	7.2	30.3

Table.1: Comparison of the sizing solutions for optimal delay, obtained with different methods. The upper and lower parts of the table give, respectively, the optimal solutions without and with constraints:  $X_{N,P}$  and Y are, respectively, the reduced values of transistor widths and load as defined above (minimum values of size and capacitance are  $3 \mu m$  and  $5.16 pF$  in the target technology respectively).



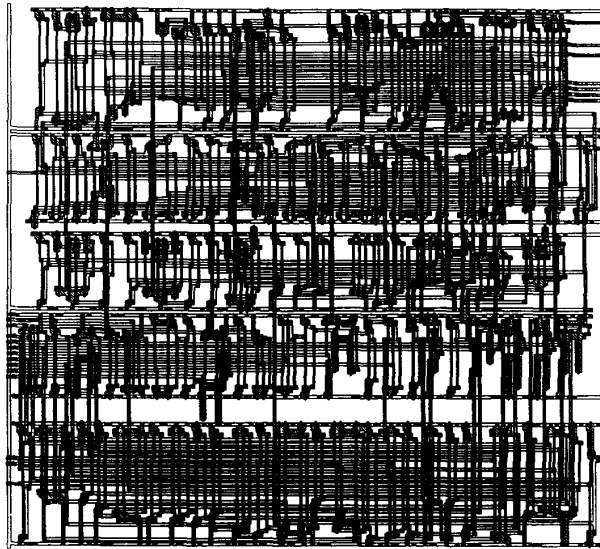


Figure.4: Example of a layout automatically generated, with PRINT and used to define post layout extracted netlist, for the sizing of ALU 4.

#### 4. Conclusion

A big effort, in the design of ICs, is made to evaluate circuit performances and to look for ways to satisfy the constraints.

We have presented in this paper a sizing tool, P.Size, embedded in a flexible cell generator, to optimize real data paths in their full environment. The optimization procedure has been proved to be as efficient as global optimization procedures implemented with mathematical (ROSENBROCK) or incremental (HSPICE) methods. P.Size is able to optimize data paths from a layout representation with very small computational requirements, opening the way to very large circuit automatic sizing.

The integration of a P.Size module in a flexible cell generator gives, to the complete system, the ability to speed circuits up by optimization through automatic layout-evaluation-sizing cycles.

The availability of explicit sizing laws opens the way to predicting the performances of complex structures at the layout level, allowing bottom up design information to be used at the synthesis level.

#### Références

- [1] M. ROBERT, D. DESCHACHT, G. CATHEBRAS, S. PRAVOSSOUDOVITCH, D. AUVERGNE: "PRINT methodology: a compilation approach for cell library generation", 'ISCAS'88, Vol.2, p965-968, FINLAND, 1988.
- [2] C. LEE, M. SOUKUP: "An Algorithm for CMOS Timing and Area Optimization", IEEE Journal of Solid States Circuits, vol. SC-19, n°5, p781-787, October 1984.
- [3] K.S. HEDLUND: "AESOP: A tool for Automated Transistor Sizing", 24<sup>th</sup> ACM/IEEE Design Automation Conference, p43-48, 1987.
- [4] D. MARPLE: "Transistor Size Optimization in the Tailor Layout System", Proceedings of the IEE Design Automation Conference, p43-48, 1989.
- [5] M. R.C.M. BERKELAAR, J. A.G. JESS: "Gate sizing in MOS digital circuits with linear programming", Proceedings of the IEE Design Automation Conference, p217-221, 1990.
- [6] H.Y. CHEN, S.M. KANG: "ICOACH: A circuit optimization aid for CMOS high-performance circuits", INTEGRATION, the VLSI journal 10 (1991), p185-212.
- [7] J.P. FISHBURN, A.E. DUNLOP: "TILOS: a Posynomial Programming Approach to Transistor Sizing", Proceeding IEEE International Conference on CAD, p326-328, nov 1985.
- [8] M. HOFMANN, J.K. KIM: "Delay Optimization of Combinational Static CMOS Logic", 24<sup>th</sup> ACM/IEEE Design Automation Conference, p125-132, 1987.
- [9] J. YUAN, C.H. SVENSSOM: "CMOS Circuits Speed Optimization based on Switch level simulation", IEEE Proceedings International Symposium on Circuits and Systems, (ISCAS'88), p2109-2112, Helsinki, Finland, June 1988.
- [10] J.M., SHYU, A. SANGIOVANNI-VINCENTELLI, J.P. FISHBURN and A.E. DUNLOP: "Optimization based transistor sizing", IEEE J. of Solid State Circuits, vol.23, n°2, p400-409, april 1988.
- [11] F.W. OBERMEIER, R. KATZ: "An Electrical Optimizer that consider Physical Layout", 25<sup>th</sup> ACM/IEEE DAC, p453-459, 1988.
- [12] D. AUVERGNE, N. AZEMARD, D. DESCHACHT, M. ROBERT: "Evaluation dynamique et optimisation des structures CMOS et VLSI", TSI, vol.8, n°6, p593-607, Decembre 1989.
- [13] D. DESCHACHT, P. PINEDE, M. ROBERT, D. AUVERGNE: "PATH-RUNNER: an accurate and fast timing analyser", The European Design Automatic Conference, Glasgow, Scotland, p529-533, 12-15 March 1990.
- [14] D. DESCHACHT, M. ROBERT, D. AUVERGNE: "Explicit Formulation of delays in CMOS data paths", IEEE Journal of Solid State Circuits, vol. 23, n°5, p1257-1264, Oct 1988.
- [15] D. AUVERGNE, N. AZEMARD, D. DESCHACHT, M. ROBERT: "Input waveform slope effects in CMOS delays", IEEE J. of solid state circuits, vol.25, n°6, p1588-1590, dec 1990.
- [16] D. AUVERGNE, N. AZEMARD, V. BONZOM, D. DESCHACHT, M. ROBERT: "Formal Sizing Rules of CMOS Circuits", The European Design Automatic Conference, Amsterdam, The Netherlands, p96-100, 25-28 February 1991.
- [17] D.M. HIMMELBLAU, Applied non linear programming: "Rosenbrock's method", chp.4, Mc Graw Hill ed.1972.