Speed Indicators for Circuit Optimization

Alexandre Verle, Alexis Landrault, Philippe Maurine, Nadine Azemard

Speed Indicators for Circuit Optimization

Alexandre Verle, A. Landrault, Philippe Maurine, and Nadine Azémard

LIRMM, UMR CNRS/Université de Montpellier II, (C5506),
161 rue Ada, 34392 Montpellier, France
{pmaurine,azemard}@lirmm.fr

Abstract. Due to the development of high performance portable applications associated to the high-density integration allowed by deep submicron processes, circuit optimization under delay constraints has emerged as a critical issue for VLSI designers. The objective of this work is to avoid the use of random mathematical methods (very CPU time expensive), by defining simple, fast and deterministic indicators allowing easy and fast implementation of circuits at the required speed. We propose to extend the method of equal sensitivity, previously developed for combinatorial paths [1], to circuit sizing in order to solve the circuit convergence branch problem. We propose a coefficient based approach to solve the divergence branch problem. Validation is given by comparing with an industrial tool the performance of different benchmarks implemented in a standard 180nm CMOS process.

1 Introduction

Delay bound determination and sizing under constraints for a complete circuit is one of the most difficult task to be achieved. A lot of solutions has been proposed for paths, but, because a circuit can be assimilated to overlapping paths, the resulting interdependence between delays and input capacitances of the different circuit paths imposes the solution of a NP complete problem [2].

The only solution to optimize a circuit in an optimal or quasi-optimal way, consists in using mathematical approaches [3-6]. Unfortunately, these approaches are very CPU time expensive and are quickly limited by the circuit size. An another approach consists in evaluating and then in sizing the slowest path of the circuit [7]. This procedure is repeated until constraint satisfaction on all the paths. However, the path overlap requires many iterations. In this case, the convergence of the algorithm is not guaranteed and the risk of infinite loops is very high [8].

To effectively reduce the number of loops, we extend, in this paper**,** the equal sensitivity method defined on a path [1] to the sizing of all the circuit paths. To solve the path sizing problem two particular structures must be considered, the divergence and re-convergence branches. For that, we study these two structures in order to propose indicators allowing to define an accurate and deterministic circuit sizing methodology for reducing the number of optimization loops.

In section 2, we define the delay bounds of a circuit. For that, we study the problem of re-convergences and determine, from an initial sizing, the most probable real critical path [9] to get an abstraction of the circuit delay performance. In section 3, we treat the transistor sizing problem under delay constraint. We study the problems induced by the divergence branches. Finally we apply and validate the proposed approach to full circuits before concluding in section 4.

2   Critical Path Evaluation: Convergence Problem

Before sizing a circuit under delay constraint, it is necessary to be able to estimate the feasibility of its constraint. For that we define bounds from physical indicators.

- For the maximum delay (T_{MAX}), we set all transistors of the circuit at minimum size. This technique was already used to obtain a pseudo-maximum delay of a path.
- For the minimum delay (T_{MIN}), the problem is different, because this minimum delay depends on divergences and path overlaps: it is a NP complete problem.

The problem is the determination of the critical path of the circuit (i.e. the path with the longest delay for a given sizing) and the value of its minimum delay. The lower delay bound of the circuit (T_{MIN}) and the feasibility of circuit delay constraint are determined from the first path identified as a critical path. Thus, if this path is badly defined at the beginning, iterations are necessary. The most frequently used technique consists in determining the critical path starting from a minimum sizing but it presents an obvious lack of effectiveness. Indeed, this approach does not reflect the path ability to be sized.

2.1   Critical Path Problem Illustration

To illustrate this problem, let us consider three independent paths (path1, path2 and path3 described in Fig.1). For these paths, gate type and parasitic capacitances (C_P units) of each node are given, inv, nrx and ndx for inverters, nor and nand x inputs, respectively. First we define the delay bounds.

For the upper bound, a realistic approach consists in setting all transistors at the minimum size.

For the lower bound, we apply the method of equal sensitivity previously developed for combinatorial paths [1]. So, for a path, the inferior delay bound is easily obtained by canceling the derivatives of the path delay with respect to the input capacitance of its gates. This results in a set of linked equations where the size of a gate depends on the sizes of its previous and next gates. Instead of solving this linked equations, we use an iterative approach starting from any local solution (equal to the local minimum on Fig.1) in order to save CPU time. With this approach, we reach quickly the minimum delay. The evolution of these iterations is illustrated in Fig.1.

We can compare the minimum and maximum delays of these paths. The curves of Fig.1 represent for each path, the delay versus area with an implementation at minimum area (T_{MAX}) and an implementation at global minimum (T_{MIN}). In this figure, each point represents an iteration, starting from the 1st iteration for the local solution, to the last iteration for the delay min.

In Fig.1, we note that for the implementation at minimum area, the critical path of the circuit (the slowest path) is path1. This path must represent the circuit performance when it is implemented at minimum size, i.e. the value of the minimum delay, T_{MIN1}, for the circuit.

However if we compare the delay evolution with minimum sizing of the other paths, we can note that path2 limits the delay performance of the circuit because parth1 is faster.
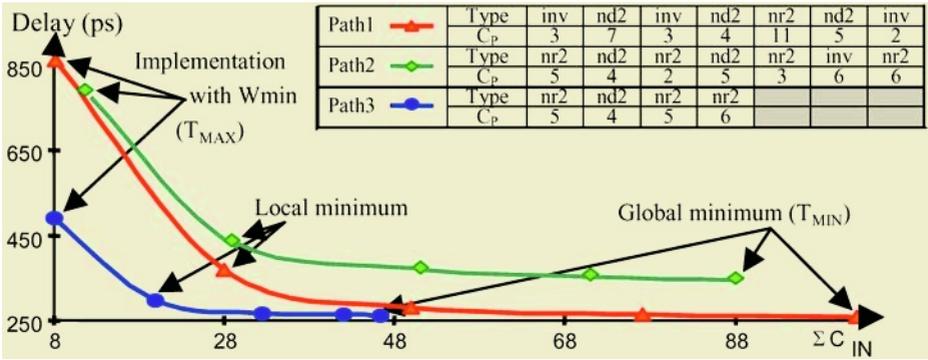
Fig. 1. Design space exploration for three paths of a circuit

Thus, to extract the critical path of a circuit, the implementation at minimum area is not sufficient, because a defined critical path (path1) can have better delay performances than a defined sub-critical path. Only the global implementation allows to exactly evaluate the circuit critical path: this is not easy to realize because an exhaustive research is CPU time expensive.

To avoid these problems, we propose to use a circuit implementation at local minimum. As we can verify in Fig.1, an evaluation at local minimum also allows to have a good idea of the path delay performances.

Tab.1 gives the delay values obtained for path1, path2 and path3 with implementations at minimum area, at local minimum and finally at global minimum (the exact technique). For each implementation, the critical path is represented by a gray box. The implementation at local minimum limits the risk of error in the determination of the critical path and gives a satisfactory solution.

Table 1. Critical path identification for different implementations

	T_{MAX} (ps) for Minimum area	T (ps) for Local minimum	T_{MIN} (ps) for Global minimum
Path 1	855	369	274
Path2	794	440	352
Path 3	494	299	264

In the following, we define an approach, allowing to evaluate the minimum delay bound of a circuit, by using a path classification obtained with an implementation at local minimum.

2.2 Local Minimum Definition

The definition of a local minimum for a circuit requires to define sizing rules taking into account re-convergences.

For a simplified delay model [10], based on the transition time of a gate, the local minimum of an array of two gates (Fig.2) is obtained by sizing the input capacitance of each gate, following:

$$C_{INk} = \sqrt{\frac{S_k}{S_{k-1}} \cdot C_L \cdot C_{REF}} \qquad (1)$$

where S_k represents the current possibilities of the gate K, C_L the load of this gate and C_{REF} a reference capacitance value, defined from the minimum value available in the process.
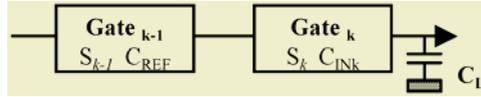

Fig. 2. A cell with two gates

To generalize this approach to a circuit, it is necessary to manage the re-convergence sizing, in order to obtain a correct local minimum evaluation.

To formalize the input capacitance of a re-convergent gate, we consider in Fig.3, a reconvergence of N gates.
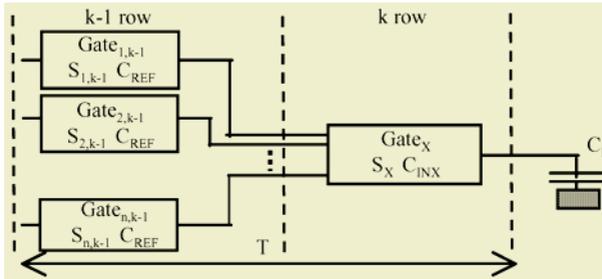

Fig. 3. A generic re-convergence

Considering (1) it is obvious that the sizing of the re-convergent gate must be defined with respect to the slowest preceding gate:

$$C_{INx} = \sqrt{\frac{S_X}{\max\left\{S_{1,k-1}; S_{2,k-1}; ..., S_{n,k-1}\right\}} \cdot C_L \cdot C_{REF}} \qquad (2)$$

This C_{INx} value allows to determine the maximum value of the input capacitance of the re-convergent gate, with no slowing down of the delay of the gates of the k-1 row.

2.3 Validation

We validate this critical path search approach by implementing at minimum global, minimum local and minimum sizing the circuit represented in Fig.4.

For this circuit, gate type and parasitic capacitances (C_P units) of each node are given, inv, nrx and ndx for inverters, nor and nand x inputs, respectively.

In order to determine the exact critical path for an implementation at global minimum, we apply an exhaustive research by sizing each path of the circuit (in$_i$ → OUT) at global minimum. For each case, the re-convergence branches not belonging to the

path under study will be also sized at delay global minimum. Tab.2 compares the delays obtained according to the selected path (in$_i$ → OUT sized at global minimum) and according to the sensitized input of convergence branch (in$_i$). For each path, we also give the sum of circuit transistor sizes ($\sum$W) in µm.
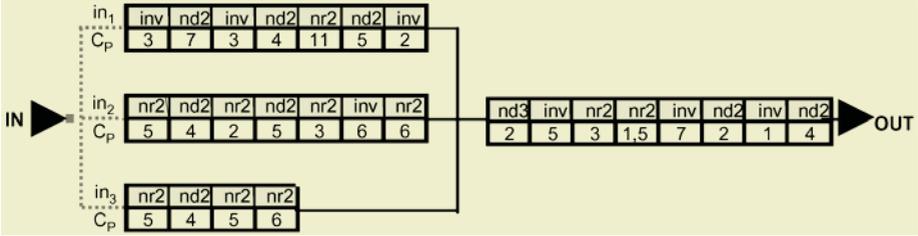

Fig. 4. Combinatorial circuit with re-convergence branches

Table 2. Exhaustive research of critical path for an implementation at global minimum

		Path delay (in$_i$ →OUT) / sensitized input (ps)			ΣW (µm)
		in$_1$	in$_2$	in$_3$	
Path sized at global minimum	in$_1$→OUT	448	541	459	370
	in$_2$→OUT	448	540	459	374
	in$_3$→OUT	448	541	459	370

For each path, the critical delay is obtained when the convergence branch input, in2, is sensitized (gray box in the table). The critical delay of the circuit is the lowest path critical delay: it is the delay lower bound of the circuit. As a result, the exact critical path of the circuit is the path in$_2$ → OUT. It exhibits the lowest sizing sensitivity.

Tab.3 compares the approach with an implementation at minimum sizing and the approach with an implementation at local minimum, together with the delay and area (ΣW). It gives the critical delay and path (gray box) of these implementations.

Table 3. Delay and area comparison for different implementations

		Circuit delay / sensitive input (ps)			ΣW (µm)
		in$_1$→OUT	in$_2$→OUT	in$_3$→OUT	
Implementation at	Minimum sizing	1556	1487	1190	37
	Local minimum	693	762	625	102

The critical path obtained for an implementation at global minimum (Tab.2), shows that the implementation at minimum sizing does not give the good critical path. On the other hand, the implementation at local minimum gives the same critical path as the approach with an implementation at global minimum.

After determination of the critical path, the next step consists in sizing the circuit to respect a delay constraint. For that, it is essential to take into account and to manage divergence branches.

3   Sizing Under Constraint: Divergence Problem

The sizing under constraint of a path with divergence branches can involve loop problems in the sizing algorithm. Indeed, the re-sizing of a gate with a size modified at a preceding iteration can strongly increase the CPU time and put the algorithm in failure. So, to solve this problem, for the path under optimization, it is essential to fix the input capacitance size of each divergence branch. The proposed approach consists in evaluating these input capacitance sizes by using an analytically calculated coefficient.

3.1   Description of the Approach with Coefficient

The proposed technique consists in determining a γ coefficient for each input of divergence branch of the critical path. This coefficient allows to calculate and to fix the input capacitance of each divergence branch. Let us detail the principle of this approach with the circuit of Fig.5.
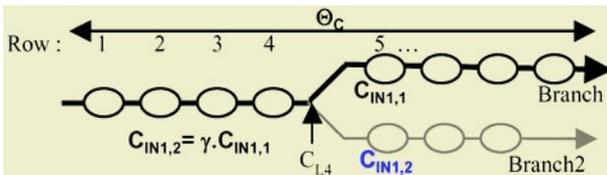

Fig. 5. Illustration of the approach with divergence coefficient

In Fig.5, the critical path is represented in bold and it has a divergence branch (Branch2). To fix the input capacitance of the divergence branch, we use a γ coefficient defined by the ratio of the input capacitance of the divergence branch (Branch2) to the input capacitance of the critical path gate of Branch1: $\gamma = C_{IN1,2}/C_{IN1,1}$. Then path sizing is processed backward from the path output to the path input. As an example, the load seen by the gate located at row 4 (C_{L4}) is then $C_{L4} = C_{IN1,1} \cdot (1 + \gamma)$. This approach allows to size easily the circuit. Let us define, and evaluate now, the coefficients of divergence branches.

3.2   Coefficient Definition and Evaluation

Circuit sizing problems are due to the circuit complexity and the significant number of unknown parameters. To calculate divergence coefficient, instead of using slow and blind mathematical approaches, we propose to define a simple indicator γ. To determine the value of the γ coefficient, let us consider the circuit of Fig.6.
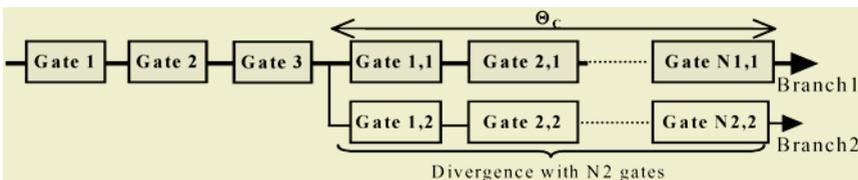

Fig. 6. Circuit with a critical path and its divergence branch

The objective is to extract a metric to obtain a ratio between the input capacitances of gate1,2 and gate1,1. At this level, any sizing information exists. So, some assumptions are made.

– Initially, on the critical path, each gate is sized at a reference value, C_{REF}, equal to the minimum size allowed by the process or the library.
– To reduce the CPU time, to obtain a fast evaluation of the path delays and to propose a simple heuristic, we use a first order delay model [10], based on the gate transition time, as

$$T_{HL,LH} = \tau_{ST} \cdot S_{HL,LH} \cdot \frac{C_L}{C_{IN}} \tag{3}$$

where, τ_{ST} is a time unit that characterizes the process, $S_{HL,LH}$, $S_{HL,LH}$ represent the symmetry factor of the falling, rising edges. C_L, and C_{IN} represent, respectively, the output load including the cell parasitic capacitance and the gate input capacitance.

The principle of the proposed approach is to equalize the propagation delay of the two branches of the circuit of Fig.6, in order to impose the reduced delay value of the critical path branch, Θ_C, on the divergence branch. The propagation delay of the critical path branch (Branch1) is equal to

$$\Theta_C = \frac{T_C}{\tau_{ST}} = \sum_{i=1}^{i=N_1} \Theta_{i,1} = \sum_{i=1}^{i=N_1} \frac{C_{Li,1}}{C_{INi,1}} = \sum_{i=1}^{i=N_1} S_{i,1} \cdot \frac{C_{Pi,1} + C_{INi+1,1}}{C_{INi,1}} \tag{4}$$

where $C_{Pi,1} + C_{INi+1,1} = C_{Li,1}$ is the output capacitance of the Gatei,1 of Branch1.

Now we apply this delay constraint, Θ_C, on the divergence branch (Branch2). The goal is to size the gates of Branch2 to obtain the value of the input capacitance of the divergence branch ($C_{IN1,2}$), and then the value of the γ coefficient. Consequently, we cannot apply the constant sensitivity method, because with this method the input capacitance of the branch is fixed [1] in order to have a convex delay function. So, to obtain an approximate but fast sizing, we use the heuristic approach of [11].

We apply the equal repartition of delays on Branch2 and we obtain the analytical expression of the γ coefficient ($\gamma = C_{IN1,2}/C_{IN1,1}$) to apply the input gate of each divergence branch.

$$\Theta_C = \sum_{i=1}^{i=N_2} \Theta_{i,2} = N_2 . \Theta_{1,2} = N_2 . \frac{S_{1,2} . C_{L1,2}}{C_{IN1,2}} \Rightarrow \gamma = \frac{S_{1,2} . C_{L1,2} . N_2}{C_{IN1,1} \sum_{i=1}^{i=N_1} S_{i,1} . \frac{C_{Li,1}}{C_{INi,1}}} \tag{5}$$

Now we compare this analytical coefficient to an experimental optimal coefficient.

3.3 Coefficient Validation

We validate the analytical expression allowing to calculate the γ coefficient (5), by comparison with an optimal coefficient, experimentally determined on two simple circuits. These circuits represent the two configurations of divergence branches:

– balanced divergences: divergence branches with same topology (circuit test1).
– unbalanced divergences: divergence branches with different characteristics (circuit test 2).

Fig.7 presents the characteristics of the two test circuits. L_{depth} represents the path logical depth, i.e. the number of gates in each path. *Gate1* is the input gate type of path under study and θ_{MAX} is the propagation delay for an implementation at minimum area (W_{MIN}).
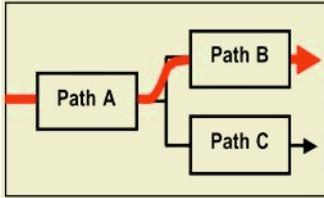
	Path characteristics	Balanced divergence branches circuit Test1			Unbalanced divergence branches circuit Test2		
Path B Path A Path C	Path	A	B	C	A	B	C
	Gate$_1$	NR3	NR3	NR2	NR3	NR2	ND2
	L$_{depth}$	10	11	11	4	17	11
	Θ_{MAX} (ps)	993	1400	1076	347	2046	1054

Fig. 7. Principal characteristics of the two test circuits

First we validate the approach on circuit Test1 (balanced divergence branches) and then on circuit Test2 (unbalanced divergence branches).

a) Validation on Balanced Divergence Branches (Circuit Test1)
Circuit test1, presented on Fig.7, allows to check that the computed value of the γ coefficient (5) is not under evaluated and is equivalent to its experimental value.

For that, Tab.4 shows the area evolution (ΣC_{IN}) of circuit Test1 for different delay constraints (Θ_C) and for different values of the γ coefficient. The γ experimental coefficient allowing to reach the delay constraint at minimum area cost is $\gamma=0,43$.

Table 4. Experimental results for different γ coefficients on the circuit Test1

Θ_C(ps)	ΣC_{IN} for different γ Coefficients						
	$\gamma=0,65$	$\gamma=0,55$	$\gamma=0,48$	$\gamma=0,43$	$\gamma=0,38$	$\gamma=0,3$	$\gamma=0,2$
2393	70	70	70	70	70	70	70
2000	79,6	78,81	78,81	78,81	78,81	78,81	78,81
1500	118,7	117,93	117,81	117,81	117,9	117,9	117,9
800	592,54	582,14	579,96	579,14	579,29	583,06	605,86
Computed coefficient : $\gamma=0,51$				Experimental coefficient : $\gamma=0,43$			

We can note that the computed γ value is equivalent to the experimental value. Now let us validate this approach on a circuit unbalanced divergence branches.

b) Validation on Unbalanced Divergence Branches (Circuit Test2)
We apply the same procedure on the circuit Test2 presented in Fig.7. We want to check that the computed value of γ coefficient (5) is not overestimated and is equivalent to its experimental value.

In Tab.5, we can see the area evolution (ΣC_{IN}) of circuit Test2 for different delay constraints (Θ_C) and for different values of the γ coefficient.

Table 5. Experimental results for different γ coefficients on circuit Test2

Θ_C(ps)	ΣC_{IN} for different γ Coefficients				
	γ=0,35	γ=0,3	γ=0,2	γ=0,15	γ=0,09
2393	71	71	71	71	71
1100	188,17	187,54	186,94	186,94	186,94
1000	236	235	233,71	233,63	233,6
750	786,01	776,5	759,92	752,5	748,95

Computed coefficient : γ=0,298   Experimental coefficient   γ=0,09

The γ experimental coefficient allowing to reach the delay constraint at minimum area cost is γ =0,09. So, the γ computed coefficient is quite equivalent to its experimental value. We also note that the approach with computed γ coefficient request an area (ΣC_{IN}) slightly higher than the experimental approach.

After validation of the γ coefficient calculated by an analytical expression, we now compare this approach to that of an industrial tool (AMPS).

3.4 Comparison with AMPS

In order to check the effectiveness of the coefficient based approach, we compare this approach to an implementation given by an industrial physical synthesis tool (AMPS tool from Synopsys company). We have just seen on the two previous circuits (circuit Test1 and circuit Test2) that the approach with coefficient gives satisfying results. These two circuits, with a critical path and a divergence branch, have allowed to apply the sizing protocol on simple examples. Thus, comparison with AMPS will be applied on a circuit with several divergence branches (balanced and unbalanced divergence branches); we used the circuit Test3, illustrated in Fig.8. This circuit is constituted of a critical path (in bold) with the a, b, c and d sub-circuits and three divergence branches (E, F and G). Each sub-circuit is represented by a box with its gate number. g_{in} indicates the input gate type of sub-circuit.
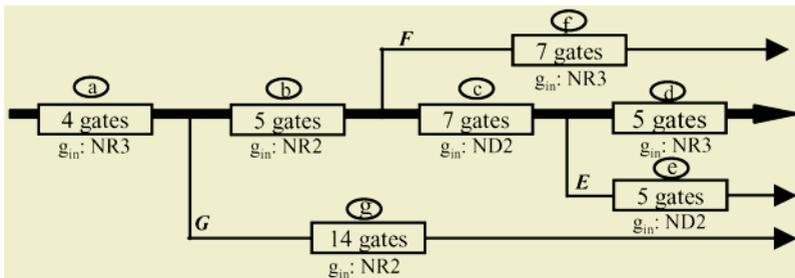

Fig. 8. Circuit Test 3

So, circuit Test3 is a mixture of the two previous test circuits (balanced and unbalanced divergences). This circuit is used to evaluate the effectiveness of the approach with coefficient on an example with multi-divergence branches. Contrary to the two preceding circuits (Fig.7), it is very difficult to obtain an experimental coefficient on a general circuit, due to problems of CPU time. Thus circuit Test3 may give evidence

of the interest in using the coefficient based approach. For illustration, we compare in Tab.6 the γ coefficients calculated using the approach with coefficient (5), for each divergence branches (E, F and G) of the circuit Test3.

Table 6. γ coefficients for divergence branches of circuit Test3

γ coefficients		
Divergence branch G	Divergence branch F	Divergence branch E
0.32	0.42	0.48

The curves of Fig.9. represent the delay evolution versus the area (ΣWi), for the circuit test3, using the approach with coefficient and the tool AMPS. These results are obtained with Hspice for a standard 0,18μm process.

On Fig.9, we can note that the implementation obtained by the approach with coefficient, gives better results than AMPS, at once for the evaluation of the minimum delay and for area saving under delay constraint. We noticed that the proposed approach enable to reach the best results.
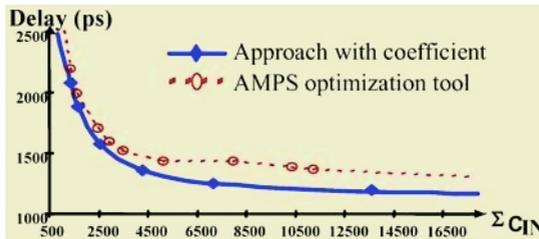

Fig. 9. Space design exploration for the circuit test 3 with AMPS

4   Conclusion

In this paper, we have presented an indicator based solution for circuit sizing under delay constraint. We have given solutions to the sizing of re-convergence (determination of delay bounds) and divergence branches (sizing under delay constraint).

We have defined a methodology in which we firstly propose to determine the circuit delay bounds from critical path search (the path with the worst sizing sensitivity). Then we have demonstrated that an implementation at minimum area does not reflect the real performances of a circuit, and proposed an implementation at local minimum to obtain a better idea of the circuit performance and a robust path classification. To determine this local minimum, that defines the circuit performance, we have proposed and validated an approach allowing to manage the re-convergence branches.

In the second time, for an achievable constraint, we have proposed a complete circuit sizing solution. Highlighting the divergence branch problem, this solution allows to fix the size of the input capacitances of divergence branches in a deterministic way, allowing to avoid iterations.

This circuit sizing protocol is under implementation for validation on ISCAS benchmarks.

References

1. A.Verle, X. Michel, N. Azémard, P. Maurine, D. Auvergne, "Low Power Oriented CMOS Circuit *Optimization Protocol*", DATE'05, Munich, Germany, 7-11 March 2005, pp 640-645.
2. R. Murgai, "On the Global Fanout Optimization Problem", In IWLS, Granlibakken, USA, 1999.
3. D. Marple, "*Transistor Size Optimization in Tailor Layout System*", Proc of the 26th IEEE/ACM Design Automation Conference, pp43-48, June 1989.
4. V. Sundarajan, S. S. Sapatnekar and, K. K. Parhi, "*Fast and Exact Transistor Sizing Based on Iterative Relaxation*," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, Vol. 21, N° 5, pp568-581, May 2002.
5. A. R. Conn and, *al*, "JiffyTune: Circuit Optimization Using Time-Domain Sensitivities," IEEE Transactions on Computer Aided Design of Integrated Circuits an Systems, Vol. 17, NO. 12, pp1292-1308, December 1998.
6. H. Tennakoon and, C. Sechen, "Gate Sizing Using Lagrangian Relaxation Combined with a Fast Gradient-Based Pre-Processing Step," Proc of the IEEE/ACM International Conference on Computer-Aided Design, ICCAD'02, pp395-402, November 2002.
7. Y. Y. Yu, V. Oklobdzija and, W. W. Walker," An Efficient Transistor Optimizer for Custom Circuits," Proc of IEEE International Symposium on Circuits and Systems, ISCAS'03, Vol. 5, pp197-200, May 2003.
8. R. H. J. M Otten and, R.K. Brayton, "Planning for Performance," Proc of the 35th Design Automation Conference, pp121-126, June 1998.
9. H.C. Chen, D.H.C. Du and L.R. Liu, "Critical Path Selection for Performance Optimization", IEEE trans. On CAD of Integrated Circuits and Systems, vol. 12, n°2, pp. 185-195, February 1995
10. A.Verle, X. Michel, N. Azémard, P. Maurine, D. Auvergne, "Low Power Oriented CMOS Circuit *Optimization Protocol*", DATE'05, Munich, Germany, 7-11 March 2005, pp 640-645.
11. I.E. Sutherland, B. Sproull, D. Harris, "Logical effort, designing fast cmos circuits", Morgan Kaufmann Publishers, Inc., 1999.