Timing Driven Floorplanning on Programmable Hierarchical Targets

S.A Senouci, A. Amoura, H. Krupnova, G. Saucier.

Institut National Polytechnique de Grenoble/CSI

46, Avenue Felix Viallet 38031 Grenoble cedex, FRANCE

Aadil.Amoura@atarax.imag.fr

Abstract– The goal of this paper is to perform a timing optimization of a circuit described by a network of cells on a target structure whose connection delays have discrete values following its hierarchy. The circuits is modelled by a set of timed cones whose delay histograms allow their classification into critical, potential critical and neutral cones according to predicted delays. The floorplanning is then guided by this cone structuring and has two innovative features: first, it is shown that the placement of the elements of the neutral cones has no impact on timing results, thus a significant reduction is obtained; second, despite a greedy approach, a near optimal floorplan is achieved in a large number of examples.

1 Introduction

The problem of incorporating performance objectives in the physical design of integrated circuits has been widely addressed in the past relating to placement, floorplanning, and chip partitioning.

In [BurYou85] an approach to the automatic layout design for VLSI chips was proposed. It incorporates timing information to influence the placement and wiring processes. Placement is based on a successive partitioning algorithm. Weighting nets according to their timing criticality biases the gain computation of the FM partitioning algorithm.

In [ShiKuTsay92] system partitioning algorithm for MCM is proposed under timing and capacity constraints. They use a divide-and-conquer strategy: fisrtly, clustering is applied to insure timing correctness; then K-Way packing is applied to obtain an initial solution satisfying capacity constraints, and after that K&L algorithm tries to minimize net crossings.

In [KatsKoWaYo95] partitioning method under performance, area and IO pins constraints was proposed for MCM systems. The method firstly uses clustering of all nodes that cause the timing violations. After that the iterative improvement with mathematical programming is applied to minimize the number of cuts.

In [RajWong93] the problem of circuit clustering for delay minimization was considered under any monotone constraint. Proposed algorithm is timing-optimal, but the penalty is a high degree of replication. In [YaWo95] pin constraint is also taken into consideration.

In [SwaSe95] path-based timing driven placement algorithm is presented. The difference with the preceding path-based approaches is that it may handle very large circuits. The hierarchical methodology is applied through condensing netlist and applying simulated annealing with different temperatures to netlist with different degrees of condensing. Timing penalty is incorporated into the annealing cost function.

In [YousSait95] timing driven floorplanning approach is presented for general cell layouts. The approach incorporates timing criteria into the objective function of the greedy force-directed block placement algorithm.

In [RoySe94] present a multi-FPGA partitioning algorithm handling timing constraints. Method considers the geometric aspects - relative positions of partitions with respect to each other, subdividing each partition into bins. Timing penalty function is incorporated into the simulated annealing cost function.

In [SawTho95] a constructive set cover based approach is proposed to minimize the number of chip crossings in multi-way partitioning for FPGAs. In [SauBra93] cone-based clustering and clusters merge are applied to contain critical paths inside cones.

No one of the previous approaches addressed the multi-level target. In this paper, we present a timing-driven floorplanning approach for hierachically structured programmable targets. We propose an algorithm which defines partial assignement of the design cells to the target struc-

ture nodes. This assignement is performed only for timing critical cells and guarantes the timing predictability of the final placement.

The paper is organized as follows. In section 2, we give basic definitions, terminology and notations used thereafter. Our desing modelling is presented in section 3 nd 4. Timing modelling and the floorplanning problem are described in section 5. Algorithm for timing invariant partial floorplan is discussed in section 6. In section 6, we show how to translate constraints floorplan to the place and route tool. Experimental results and conclusion are given in sections 8 and 9.

2 Hierarchical target

2.1 Physical characteristics

A target architecture is characterized by a set of basic modules/cells and interconnection resources. A hierarchical target is defined in addition by a hierarchy tree diagram with a depth corresponding to the number of levels of hierarchy. At each level the chip is organized into a subset regions, called quadrants, containing a fixed number of modules. Like for any hierarchy, the subsets at a given level are included in a subset associated with a higher level. In this paper we make a basic assumption that connections have a discrete delay at different hierarchy levels. Usually the interconnect delay grows at higher levels. Figure 1 gives an example of a hierarchical device structure represented by a hierarchy tree and a chip layout. Each hierarchy tree node corresponds to a chip region, or quadrant, and may be weighted by a number of parameters (size in terms of the number of basic cells, IO pins number, etc.). In Figure 1a the "hierarchy tree" diagram represents the target with three levels of hierarchy. In Figure 1b the corresponding chip structure is shown. At each level of hierarchy the interconnect channels are available. We suppose that there is no limitation on the interconnect resources.

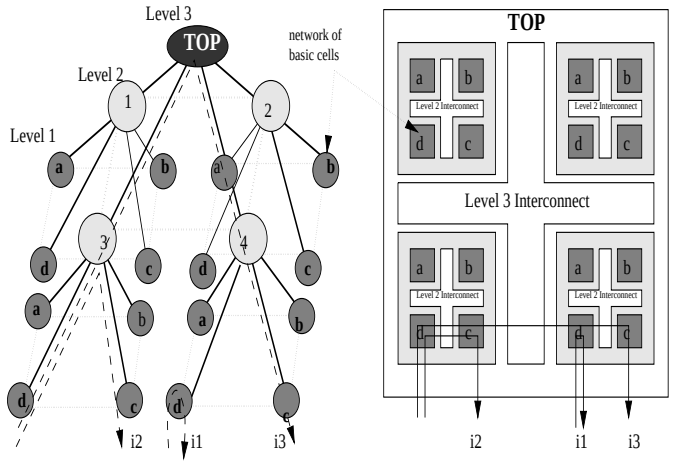

Figure 1: Hierarchical Target: a) Hierarchy tree; b) chip structure.

Notations

- Let $N_{I/O}^{Q}$ be the number of I/O of a quadrant.
- Let N_{c}^{Q} be the number of cells of a quadrant.
- Let N_{q}^{S} be the number of quadrants in a segment.

2.2 Timing characteristics

As was said above, each level of hierarchy in the target architecture is characterized by an interconnect delay added when traversing this level. For example $i1, i2$ and $i3$ correspond to the interconnect delay of the Levels 1,2 and 3 correspondingly in Figure 1.

In the following, we suppose that the target hierarchy has 3 levels as shown in Figure 1. We call the first level nodes as quadrants and the second level nodes as segments.

- Traversal time of a cell c_i is denoted $\triangle c_i$ and may vary from a cell to another.
- The interconnect delay between two cells in a quadrant is constant which implies a fanout independant delay.
- The interconnect delay between two cells in two differents quadrants is constant and denoted d_q.
- The interconnect delay between two cells in two differents segments is constant and denoted d_s.

3 Design modelling

3.1 Boolean network

The digital circuit is modelized as boolean network. This network is represented as a directed bipartite graph $G = (V_1, V_2, E)$ where the node set V_1 represents the circuit elements and node set V_2 represents the nets.

- A node $N_1 \in V_1$ is said predecessor of a node $N_2 \in V_2$ if there exists a directed edge $e \in E$ from $N_1 \in V_1$ to $N_2 \in V_2$. In other words,if a net N_2 is connected to the output pin of module represented by N_1. The node N_2 is called successor of the node N_1.
- A node $N_1 \in V_1$ is said successor of a node $N_2 \in V_2$ if there exists a directed edge $e \in E$ from $N_2 \in V_2$ to $N_1 \in V_1$. In other words,if a net N_2 is connected to an input pin of module represented by N_1. The node N_2 is called predecessor of the node N_1.
- Nodes from the set V_1 correspond to combinatorial and sequential elements of the circuit.
- The inputs/outputs of the sequential elements are called secondary inputs/outputs.

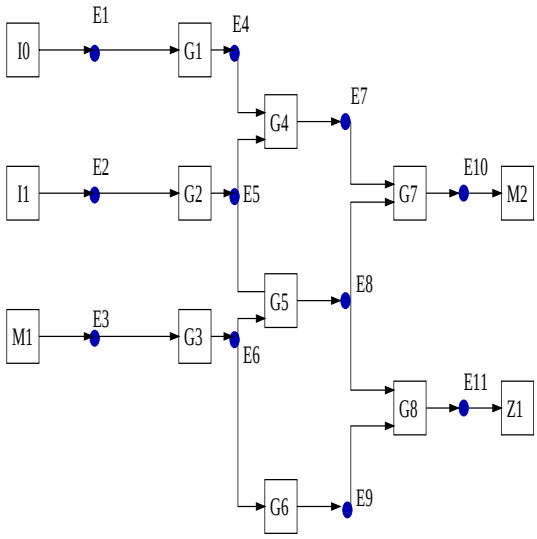

Figure 2: Circuit representation.

3.2 Predecessor cone

We define the predecessor cone of a node of the set V_2 by the set of paths connecting that node to primary or secondary inputs without traversing any node corresponding to a sequntial element.

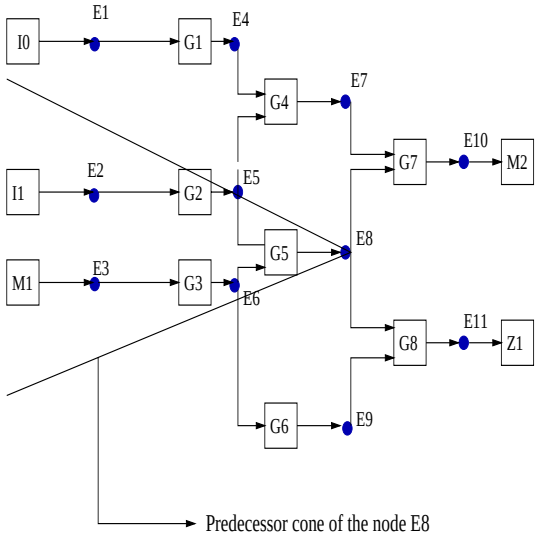

Figure 3: Predecessor cones.

3.3 Paths in a boolean network

Definition 1 : Logic delay of a path
A path P traversing n cells has a logic delay $T_L(P) = \sum_{i=1}^{n}(\triangle c_i)$.
Definition 2 : Interconnect delay of a path
The interconnect delay of a path P is determined after the floorplanning process. It takes in account the number of traversed hierarchy levels.
Definition 3 : Physical delay of a path
The physical delay of a path P is defined as follow : $T_P(P) = T_L(P) + T_I(P)$. Where $T_L(P)$ is the logic delay of the path P and $T_I(P)$ is the Interconnect delay of the path P.

3.4 Prime cones of a design

A prime cone in the network is a predecessor cone of any primary /secondary output.

One node may belong to one or more cones, which forms the cone intersections. An example of circuit containing two intersecting cones is given in Figure 4.

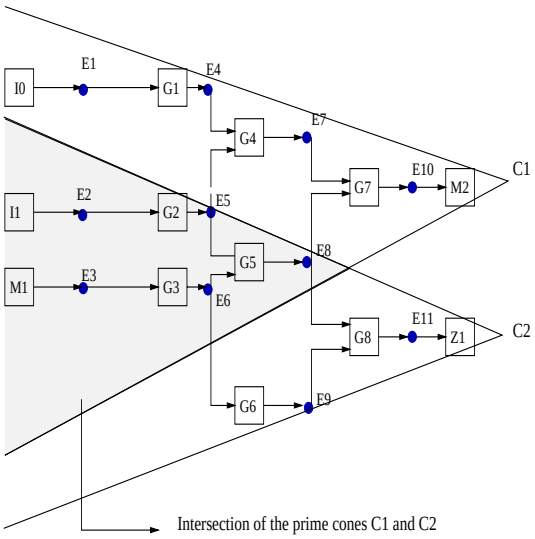

Figure 4: Intersecting prime cones.

3.5 Design profile

In Figure 5 are given statistics of the number of prime cones in different industrial circuits.

Figure 6 shows the saturation in term of cells of the prime cones. In fact, the size of the cones in term of cells will be a criteria to choose an appropriate algorithm to perform the floorplanning, thus we will consider in the following two kinds of cones, the wide ones and the narrow ones.

4 Timing modelling for prime cones

It is considered here to use the prime cones as basic constituants. The prime cones are classified according to timing criticality. This will allow timing driven assignement of prime cones to quadrants later on.

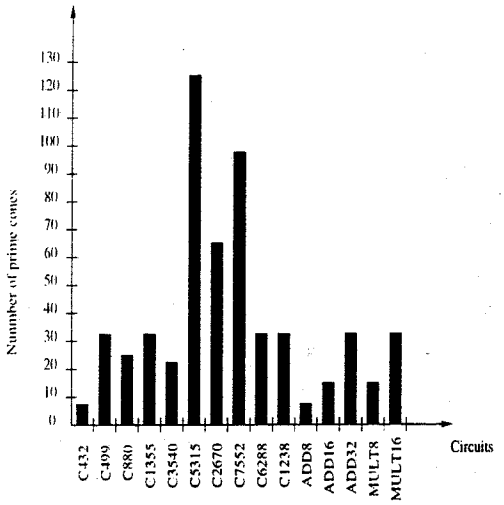

Figure 5: Prime cones .

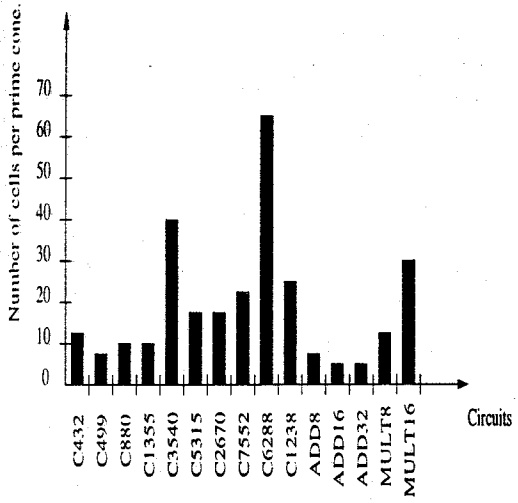

Figure 6: Cells saturation of the prime cones .

4.1 Predicted arrival time

The arrival time of a cone C_i is the arrival time at the output of the top cell of this cone.
With each node is associated a predicted arrival time and a predecessor cone. For a signal to reach the root of the cone, it has to cross potentially quadrants, segments and basic cells.

Without taking in account any interconnection delay, the arrival time at a node N_i is at least equal to $Max_{p \in C_i}(T_L(P))$, maximal logic time of all paths ending at node N_i .

To predict the arrival time at a node, it is also required to predict the number of quadrants or segments traversed before reaching this node. This number of quadrants depends both on the I/O and size saturation.

Quadrant I/O constraints
Let $N_{I/O}(C_i)$ be the number of I/O of the predecessor cone C_i of a node N_i. The minimum number of traversed quadrants required according to the I/O constraints is given by the following formula :

$$N^Q_{I/O} = \lceil \frac{N_{I/O}(C_i)}{N_{I/O}(Q)} \rceil.$$

Quadrant cell saturation constraints
Let $N_c(C_i)$ is the number of cells of the predecessor cone C_i of a node N_i. The minimum number of traversed quadrants required according to the cell constraints is given by the following formula :

$$N^Q_c = \lceil \frac{N_c(C_i)}{N_c(Q)} \rceil.$$

$N_Q(C_i)$ is the minimum number of traversed quadrants required to implement the predecessor cone C_i of the node N_i and defined as follow :

$$N_Q(C_i) = Max\{N^Q_{I/O}, N^Q_c\}.$$

Segment saturation constraints
$N_S(C_i)$ is the minimal number of traversed segments required by the assignement of the predecessor cone C_i of the node N_i and is defined as follows :

$$N_S(C_i) = \lceil \frac{N_Q(C_i)}{N_Q(S)} \rceil - 1.$$

Definition 5 :
The *upper bound* predicted arrival time of the cone C_i is defined as follows :

$$\overline{AT}(C_i) = Max_{P \in C_i}(T_L(P)) + (m-1) \cdot d_s.$$
Where m is the number of basic cells belonging to the path P.

Definition 6 :
The *lower bound* predicted arrival time of the cone C_i is

defined as follow :

$\underline{AT}(C_i) = Max_{P \in C_i}(T_L(P)) + N_Q(C_i) \cdot d_q + N_S(C_i) \cdot d_S.$

Definition 7 :
The physical delay of a cone is equal to the physical delay of its longest path.
$T_P(C_i) = T_P(P)$, where P is the longest path in the cone C_i.

Property :
Let $T_P(C_i)$ be the physical delay of the prime cone C_i, then we have :
$T_p(C_i) \leq \overline{AT}(C_i)$.

Proof :
Let C_i be a prime cone, its physical delay after placement is at worst equal to $\overline{AT}(C_i)$, because in the worst case, each cell of the longest path delay in the cone C_i is assigned to a different segment.

5 Design timing profile and classifying prime cones of a circuit

5.1 Classification of prime cones

At the begining, the Floorplanning process consists in assignment of basic cells to quadrants. Prime cones are evaluated by computing thier lower and upper predicted arrival times. These predicted times allow us to classify the cones into three different sets.

Set1: ϵ **– Critical prime cone.** Prime cone whose root has a lower bound predicted time $\underline{AT}(C_i)$ such that : $\underline{AT}(C_i) \geq Max(\underline{AT}) - \epsilon$.

Set2: Neutral cones. Prime cone whose root has an upper bound predicted time $\overline{AT}(C_i)$ such that : $\overline{AT}(C_i) \leq Max(\underline{AT}) - \epsilon$.

Set3: Potential critical cones. Prime cone whose root has an upper bound predicted time $\overline{AT}(C_i)$ such that :$\overline{AT}(C_i) \succ Max(\underline{AT}) - \epsilon$.

5.2 Design timing profile

In Table 1, we present the results of timing analysis performed on the MCNC benchmark C880. It may be seen that the number of $\epsilon - critical$ cones is small (equal to 4), and the number of potential critical cones is smaller (equal to 3) than the number of neutral cones (equal to 19).
The value of ϵ is fixed here to almost 10% of the maximum lower bound predicted time of the whole prime cones.

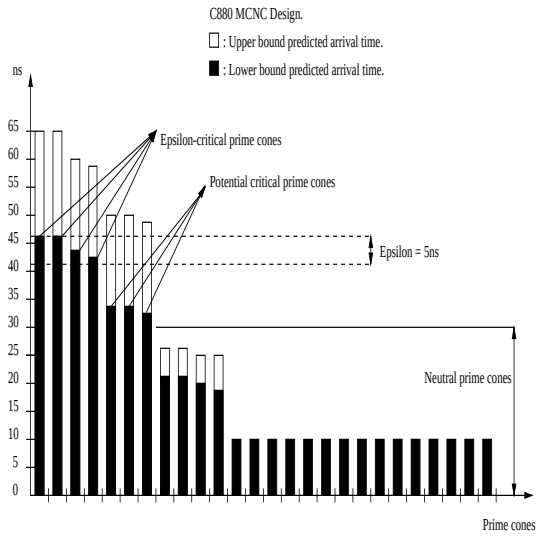

Figure 7: Cone classification results.

5.3 Floorplanning problem

The Floorplanning process consits on assigning cells to the quadrants of the hierarchical target.

Theorem
The physical delay of a set of prime cones is independent of the assignement of the cells of neutral cones.

Proof
Let C_i be a neutral cone and $T_P(C_i)$ the physical delay of C_i .
According to the definition of neutral cones, we know that $\overline{AT}(C_i) \leq Max(\underline{AT})$. (I)
In another hand, even in the case of the worst assignement of the cells of the cone C_i, we have $T_P(C_i) \leq \overline{AT}(C_i)$. (II)
(I) and (II) implie that $T_P(C_i) \leq Max(\underline{AT})$.
Or $Max(\underline{AT})$ is the lower bound predicted arrival time of the set of prime cones, this implies that the assignement of the cells of the neutral cone C_i has no influence on the physical delay of the set of prime cones.

5.4 Experimental results on complexity reduction

The complexity reduction of the floorplanning problem is about (Number of neutral prime cone / Number of prime cones).
In Figure 7 is presented an experimental evaluation of the complexity reduction due to the elimination of neutral cones. We observe that in average, the complexity was reduced by more than 47%.

Notations
Pot-critical :Potential Critical

Designs	Total prime cones	$\epsilon - critical$ prime cones	Pot-critical prime cones	Neutral prime cones	% Neutral prime cones
C1355	32	32	0	0	0%
C3540	22	1	4	17	77%
C432	7	1	3	3	43%
C499	32	8	24	0	0%
C5315	123	2	31	90	73%
C880	26	2	5	19	73%
C6288	32	6	15	11	34%
s1238	31	4	18	9	29%
ADD8	8	1	0	7	87%
ADD16	16	1	5	10	62%
ADD32	32	2	10	20	62%
Mult8	16	1	8	7	44%
Mult16	32	3	21	8	25%
Total Average Gain					47%

Table 1: Complexity reduction

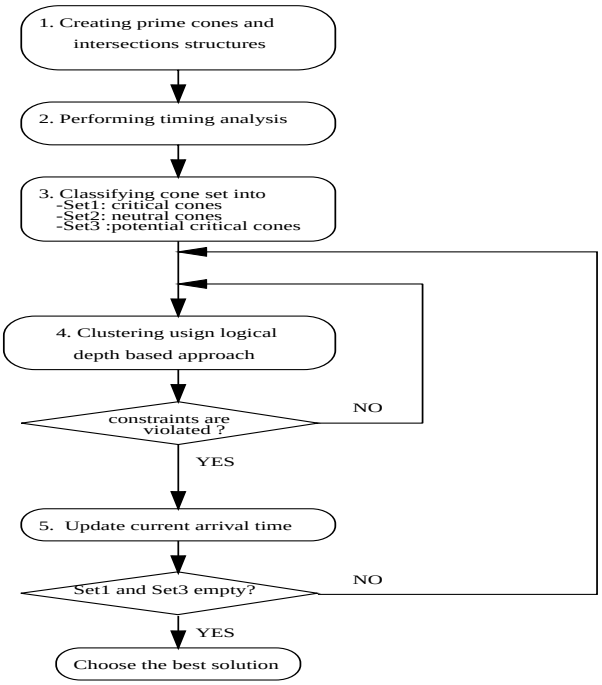

Figure 8: Timing-driven placement algorithm.

6 Algorithm for timing invariant partial floorplan

In the first step, the prime cones and their intersections are created, then we perform the timing analysis by computing the different delays ($\overline{AT}$, $\underline{AT}$,...). This timing analysis allows us to classify the prime cones into three sets :$\epsilon - critical$ prime cones, potential critical prime cones and neutral prime cones.
During the floorplanning process, the current arrival time is updated as follows :
Consider the floorplanning performed on the elements of a cone C_i whose root is a node N_i. At each step, a cell c_j of the cone is assigned to a quadrant Q. If at a given time, the number of quadrants used is greater than the minimum number of quadrants required to implement the cone C_i whish is estimated to $N_Q(C_i)$, the current arrival time is then updated.To update the current arrival time, interconnect delay between quadrants is then taken in account. If two quadrants belong to the same segment and connected to each other, then an interconnect delay d_q is added. Otherwise, if two quadrants belong to two different segments and connected to each other, then an interconnet delay d_s is then added. All the paths which cross the cone C_i are also updated, but we have to distinguish two cases :
The first case is when C_i belongs to an intersection between two different prime cones, then all the paths which cross the cone C_i and may belong to the two prime cones have to be updated. Otherwise, only the paths which cross the cone C_i and the selected prime cone are currently updated.

6.1 Logic depth based approach

Input : Classified cones.
Local variables :
N_c^Q : Number of cells in a quadrant.
N_q^S : Number of quadrants in a segment.
indice_segment : integer = 0.
Output : Set of constraints.

while(set of $\epsilon - critical$ cones is not empty)
Get a cone C_i in the set of $\epsilon - critical$ cones;

while(all the nodes of C_i are not assigned)
Select The longest path tree of C_i;
Let N_i be the top of the path tree selected;
/*Grouping the elements of N_i predecessor cone PC_i.*/
/*The grouping is performed from the leaves to the top*/
of the cone PC_i. */
while (Number_of_quadrant_used $\prec N_Q(PC_i)$)
indice_segment++;
$S_{indice\_segment}$ = New Set of quadrants;
while ($|S_{indice\_segment}| \prec N_q^S$)
Elements are assigned to a quadrant Q;
Add the quadrant Q to the set $S_{indice\_segment}$;
end while
end while
Update the current arrival time;
end while
end while

7 Connection to place and route tool

The resulting informations of the floorplanning algorithm are stored as constraints in a file for the place and route tool. These facilities to propagate floorplan constraints exist in mostly for all FPGA design (Xilinx, ORCA, Altera...). In this paper, we take as illustration a hierarchical target namely called AMD MACH5. In the corresponding software environnement, these constraints will be propagated to a special file called PI (physical information) and passed to the AMD filter MACHXL .
The same work can be done for Xilinx using RLOC constraints and PPR tool.

Example :
The positions of the macrocells S2 and S1 are passed to the place and route tool as follows :

Use PIfile for constraints :
{
Section Target 'S0Ba';
S1,S2;
end Section;
}
This constraint means that the place and route tool has to assign the cells S1 and S2 to the quadrant a of the segment S0 (see Figure 1).

8 Experimental results

The floorplanning algorithm described in this paper has been implemented in the C language on Sun SPARC workstations, and tested on a set of industrial examples. We compared the results with those obtained by the placement tool without floorplanning constraints. The experimental results (Table 2 and Table 3 and table 4) show a reduction of 57%,79.44%,61% on the delay due to the interconnections in the circuits and 15%,19.22%,15% on the critical path delay of the circuits.

Notations
C.P : Critical Path
Int : Interconnect

9 Conclusion

Timing predictable layout is one of the most difficult problems in the electronic circuit design world. The target addressed here makes the timing prediction easier. In addition to the complecity reduction of the floorplanning problem, we focused on designs where a logic structuring contributes also to the problem simplification for the sufficient condition track. Within this framework, it was shown that the timing predictable layout becomes a tractable problem. Pratical

Circuits	Without Containment		With Containment		Average Gain	
	C.P Delay	Int Delay	C.P Delay	Int Delay	C.P Delay	Int Delay
C1355	48.3	18	36.8	6.5	24%	64%
C3540	166.1	32	150.1	14	6%	56%
C432	78.1	13	69.6	8	11%	38%
C499	83.1	15	74.6	6.5	10%	57%
C5315	87	24	72.5	9.5	17%	60%
C880	49.3	14.5	39.3	4.5	20%	69%
Total Average Gain					15%	57%

Table 2: Experimental results on MCNC benchs.

Circuits Width (N)	Without Containment		With Containment		Average Gain	
	C.P Delay	Int Delay	C.P Delay	Int Delay	C.P Delay	Int Delay
2	22.5	0.0	22.5	0.0	0%	0%
4	32.1	1.5	30.6	0.0	5%	100%
6	60.1	14.5	45.6	0.0	25%	100%
8	76.9	14.5	62.4	1.5	20%	90%
10	85.2	21.0	64.2	3.0	25%	85%
12	94.2	22.5	71.7	3.0	24%	85%
14	94.5	22.5	72.0	3.0	24%	85%
16	108.5	27.5	81.0	4.5	25%	85%
18	108.8	27.5	81.3	4.5	25%	85%
Total Average Gain					19.22%	79.44%

Table 3: Experimental results on mutiplier.

Circuits width (N)	Without Containment		With Containment		Average Gain	
	C.P Delay	Int Delay	C.P Delay	Int Delay	C.P Delay	Int Delay
5	16.5	1.5	15	0	9%	100%
6	16.8	1.5	15.3	0	9%	100%
8	24.3	1.5	22.8	0	6%	100%
12	24.6	1.5	24.6	1.5	0%	0%
15	40.6	10	32.1	1.5	21%	85%
19	40.4	10	33.9	3	17%	70%
22	49.4	11.5	41.4	3	17%	74%
26	50.2	11.5	43.2	4.5	14%	61%
29	64.5	18	51	4.5	21%	75%
33	63.0	16.5	56	9.5	11%	42%
36	70.6	18	63.6	9.5	12%	47%
40	72.3	18	65.3	11	10%	39%
43	84.8	23	72.8	11	14%	52%
46	84.8	23	74.3	12.5	14%	46%
50	94.1	24.5	82.1	12.5	13%	49%
53	97.6	28	83.6	14	14%	50%
57	110.4	33	91.4	14	17%	56%
60	113.6	38	94.6	19	50%	17%
Total Average Gain					15%	61%

Table 4: Experimental results on adder

results demonstrated the efficiency of these approaches on a typical hierarchical target namely the last MACH5 CPLD family. The approaches proposed here can be extended to all FPGA/CPLD families.

10 References

[*SawTho*95] P. Sawkar, D. Thomas "Multi-Way Partitioning For Minimum Delay For Look-Up Table Based FPGAs", *Proc. 32nd Design Automation Conference*, 1995: 201-205.

[*SauBra*93] G. Saucier, D. Brasen, J-P. Hiol "Partitionning with cone structures", *Proc. ICCAD* , 1993

[*KatsKoWaYo*95] Y. Katsura, T. Koide, S. Wakabayashi, N.Yoshida "A New System Partitioning Method under Performance and Physical Constraints dor Multi-Chip Modules", *Proc. ICCAD*, 1995

[*YousSait*95] H. Youssef, S. M. Sait, K. J. Al-Farra, "Timing Influenced Force Directed Floorplanning", *Proc. of IEEE Int. Conf. on Comput.-Aided Design*, 1995: 156-161.

[*SwaSe*95] W. Swartz, C. Sechen, "Timing driven placement for large standard cell circuits", *Proc. 32nd Design Automation Conference*, 1995: 211-215.

[*YanWon*95] H. Yang, D. F. Wong "Circuit clustering for delay minimization under area and pin constraints", *Proc. of IEEE Int. Conf. on Comput.-Aided Design*, 1995: 65-70.

[*RoySe*94] K. Roy-Neogi, C. Sechen, "Multiple FPGA partitioning with performance optimization", *Proc. ACM/SIGDA Int. Symp. on Field Programmable Gate Arrays*, 1994: 146-152.

[*SauBra*93] G. Saucier, D. Brasen, J-P. Hiol "Partitioning with cone structures", *Proc. ICCAD*, 1993.

[*RajWong*93] R. Rajaraman, D. F. Wong, "Optimal clustering for delay minimization", *Proc. 30th Design Automation Conference*, 1993: 309-314.

[*ShiKuTsay*92] M. Shih, E. S. Kuh, R.-S. Tsay, "Performance-driven system partitioning on multi-chip modules", *Proc. 29th Design Automation Conference*, 1992: 53-56.

[*HauNaiYof*87] P.S. Hauge, R. Nair, E.J. Yoffa, "Circuit placement for predictable performance", *Proc. of IEEE Int. Conf. on Comput.-Aided Design*, 1987: 88-91.

[*BurYou*85] M. Burstein, M. N. Youssef, "Timing influenced layout design", *Proc. 22nd Design Automation Conference*, 1985: 124-130.