MANAGING THE SHAPE FUNCTION OF ANALOG DEVICES IN A SLICING TREE FLOORPLAN

P. NGUYEN TUONG, M.-M. LOUERAT, A. GREINER
PIERRE ET MARIE CURIE UNIVERSITY, FRANCE

KEYWORDS: Shape Function, Analog IP Cores, Layout, Floorplaning

ABSTRACT: Analog Intellectual Property Cores design is still under study [1, 2]. The precharacterized cell libraries concept cannot be applied because the devices (transistors, capacitors, resistors,...) are electrically sized regarding a specific context. Thus, the trend is to develop reusable generators. To guarantee a reusable and fast placement, slicing tree floorplan can be used. Considering a particular electrical sizing and a specific process, the analog devices may occupy a wide range of shapes because of folding. Finding the correct aspect ratio for each device which optimizes the placement under a specific height and/or width constraint may lead to examine a high number of cases. This paper presents a general placement method to manage rectangular objects with varying aspect ratio. This approach allows hierarchical layout-aware electrical sizing of analog circuits.

INTRODUCTION

An analog function is depicted as a generic module with unsized netlist and layout templates [4]. Each time the module is called, all devices have to be sized to meet a specification or a new process. As a consequence, the resulting device area may widely vary (commonly by a factor ten between two sets of specifications) and may increase or decrease, relatively to other devices, within the same generator. Besides, generators have to be hierarchical, in order to create complex circuits by instantiating existing generators.

The devices are usually multi-shape (flexible) objects: given a process and a specific electrical sizing, the area of a capacitor is constant, but the height/width aspect ratio is not fixed. Transistors have fingers to minimize the drain parasitics : the area is almost constant, but the aspect ratio can vary thanks to the finger number. Therefore the analog circuit layout results in a hierarchical placement problem with a controlled aspect ratio. Considering a process and an electrical sizing, we will show how each device can provide a *shape function* which defines all the feasible layout realizations for this device.

To avoid expensive computing time, it is possible to specify a floorplan for the entire circuit. The slicing tree structure [7] is suitable to consider analog constraints such as symetry. We introduce vertical containers and horizontal containers to divide the space. These containers are able to arrange into a column or into a line their children, which can be other containers or devices. As each device has a shape function, each container must have its own shape function. From bottom up to the top, the shape function can be calculated using a composition operation. Once the feasible aspect ratio of all containers has been computed, it is possible to reduce the solution space by applying a constraint, and to obtain an exact placement.

DEVICE SHAPE FUNCTION

Figure 1 shows an example of a folded transistor [5]. The layout width as a function of the layout length and the layout length as a function of the width are two step functions, where a variation of a finger number leads to a new step.

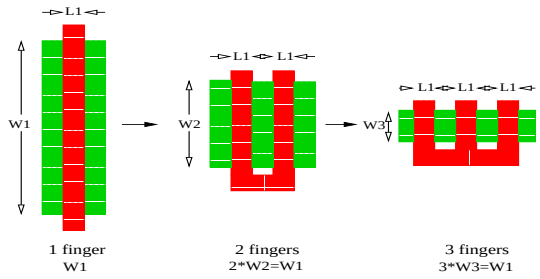

Fig. 1. Transistor folding

The following *shape function* of a device is defined, which is a step function.

Let *DY* be the height of a device and *DX* the width. We use a symbolic fixed grid (lambda unit), thus these numbers are integers : $DY \in \mathbb{N}^{*+} and\ DX \in \mathbb{N}^{*+}$.
The shape function consideringDX as a function of DY, named FF_{DY}, is defined as follows :

$$FF_{DY} : \begin{cases} there\ is\ a\ sequence \\ DY_0 < DY_1 < ... < DY_n < ... < +\infty \\ f\ is\ constant\ on\ [DY_i, DY_{i+1}[\end{cases} \quad (1)$$

and :

$$\forall\ DY \in DY_{[i,i+1[} : FF_{DY}(DY) \longrightarrow DX_i \quad (2)$$

Thus, the device width *DX* is defined by the height constraint *DY*. For an interval height (interval $[DY_i, DY_{i+1}[$) the device width is constant. The graph of this decreasing monotonous function can be drawn (figure 2 on top).

Note the minimum value DY_0, achieved by the shape function.

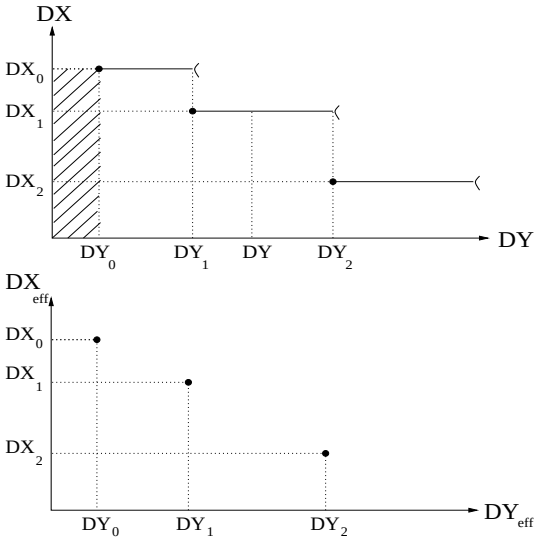

Fig. 2. Shape function regarding DY and reduced shape function regarding DY

Based on the shape function FF_{DY}, the DY_{eff} function is defined which computes the device actual height required to meet the DY height constraint:

$$\forall\, DY \in DY_{[i,i+1[} : DY_{eff}(DY) \longrightarrow DY_i \quad (3)$$

In the same way, the shape function considering DY as a function of DX , named FF_{DX}, is easily defined and gives the device actual height and width required to meet the width constraint:

$$FF_{DX} : \begin{cases} there\ is\ a\ sequence \\ DX_0 < DX_1 < ... < DX_n < ... < +\infty \\ f\ is\ constant\ on\ [DX_i, DX_{i+1}[\end{cases} \quad (4)$$

and :

$$\forall\, DX \in DX_{[i,i+1[} : FF_{DX}(DX) \longrightarrow DY_i \quad (5)$$

The data structure used for the shape function is compact: only the start point of an interval have to be considered. The reduced shape function of the device width (figure 2, bottom) is defined as the set of points : (DY_n, DX_n). It is also a decreasing monotonous function.

RELATIVE PLACEMENT

In order to follow a layout-aware approach for circuit synthesis [6], it is compulsory to be able to predict parasitics resulting from layout, early in the synthesis phase. Thus, we may choose an approach using layout templates [3, 4] with dedicated layout device generators. The space partition of a circuit is achieved using the slicing structure where the horizontal and vertical slices are named *containers*. For each level, an horizontal container must alternate with a vertical container. A *container* is a vertical or horizontal abutment of modules or other containers. The circuit placement problem is solved when, for each instantiated component, a specific point of the shape function is chosen and all the devices are relatively placed.

Figures 3 and 4 show two types of containers, a space partition and the associated container tree.

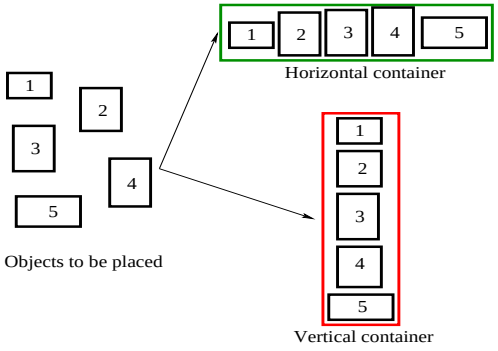

Fig. 3. Containers: object definition

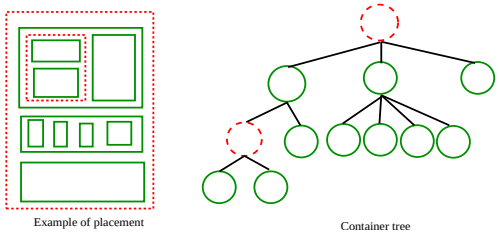

Fig. 4. Containers: floorplan and container tree

SHAPE FUNCTION COMPOSITION

A container is a collection of abutted rectangles. The shape function composition is the computation of a container shape function based on the shape function of the objects instantiated by this container. The composition is unique. Applying recursively this composition to the container tree, from the bottom leafs (i.e. the devices) up to the root, leads to the root module shape function. Keep in mind that the shape function is a way to shortly represent *all* the possible module realizations for a given process and an electrical sizing.

We present now the composition procedure for an horizontal container.

Let an horizontal container with n children (figure 5).

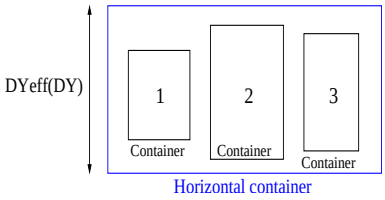

Fig. 5. Horizontal composition

- Let $DY_{eff}(DY)$ (*resp.* $DX_{eff}(DY)$) be the actual height (*resp.* width) of the horizontal container given the DY constraint.

- Let j the index of the child container and $DY^j_{eff}(DY)$ (*resp* $DX^j_{eff}(DY)$) the actual height (*resp.* width) of the j container given the DY constraint.

$DY_{eff}(DY)$ is the maximum of child actual height. The width $DX_{eff}(DY)$ of the horizontal container is the sum of each child width, thus :

$$\begin{aligned} DX_{eff}(DY) &= \sum_j DX^j_{eff}(DY) \\ DY_{eff}(DY) &= \max_j \{DY^j_{eff}(DY)\} \end{aligned} \quad (6)$$

Points DY_i, DX_i of the reduced shape function of the horizontal container are:

$$\left.\begin{array}{l} DX_i = \sum_j DX^j_{eff}(DY_i) \\ DY_i = DY_{eff}(DY) \end{array}\right\} DY \in [DY_i, DY_{i+1}[\quad (7)$$

Considering the associativity of the *sum* and *max* operators, the composition has also the associativity property. The reduced shape function of the children are sequentially examined to retain the significant points as defined in equation 6.

Figure 6 presents the implementation of the shape function composition.

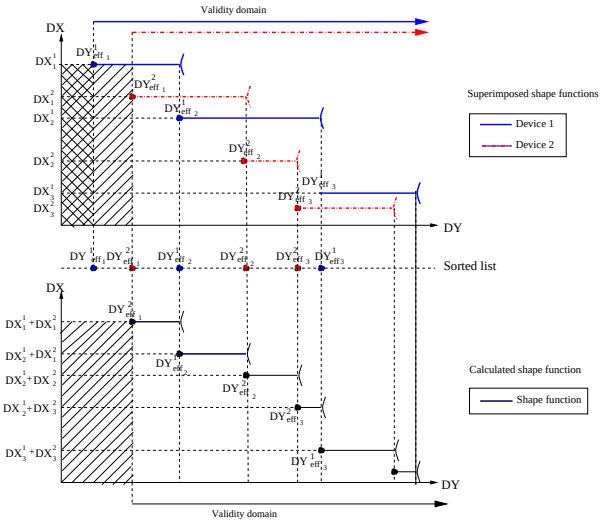

Fig. 6. Shape function composition for an horizontal container

In order to find the shape function of the container, the following steps are performed:

- The shape function point lists are merged to get a sorted list of heights in increasing order. The list is examined in increasing order.
- The first value is $DY^1_{eff_1}$; it does not belong to the device 2 feasible interval, it is thus excluded.
- The next value is $DY^2_{eff_1}$. The height $DY^2_{eff_1}$ gives $DY^1_{eff_1}$ for device 1 and $DY^2_{eff_1}$ for device 2. We keep the larger, $DY^2_{eff_1}$ (equation 7). The total width for this height is (equation 6): $DX^1_1 + DX^2_1$.
- The next height is $DY^1_{eff_2}$ for the two points $DY^2_{eff_1}$ and $DY^2_{eff_2}$. The larger value $DY^2_{eff_1}$ is retained. The width is $DX^1_2 + DX^2_1$.
- The process is repeated until there is no more actual values.

SHAPE FUNCTION SELECTION

Shape functions are computed from bottom to top. They represent all the feasible layout realizations of the associated module, given an electrical sizing. If a constraint (height or width) is applied on the top module, then a specific point of the shape function is chosen. Choosing a point of the top shape function means that a constraint is also applied on each child, so that each child aspect ratio is selected. This constraint propagates from top to bottom; then the *layout* is generated. The given placement is optimal and unique.

Figure 7 shows a height constraint DX applied to the vertical root container VC0.

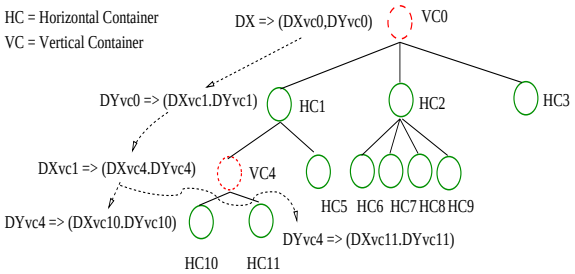

Fig. 7. Shape function selection

From the shape function, its width and height are (DX_{VC_0}, DY_{VC_0}). The left child is an horizontal container, so the height constraint DY_{VC_0} is applied. Its width and height are (DX_{VC_1}, DY_{VC_1}). Then the width constraint DX_{VC_1} is applied to its left child...

EXAMPLES

On figure 8 on the left, a square constraint was applied (process CMOS, 0.6μm) to a five transistor OTA composed by three devices, a simple transistor (top), a differential pair (middle) and a current mirror (bottom). The three devices are abutted in a vertical container. On the right, a 300 μm height constraint was applied. The electrical sizing is the same. Note the unchanged relative placement.
The designed generator can handle a large set of physical constraints with an exact placement, which ensures the portability.

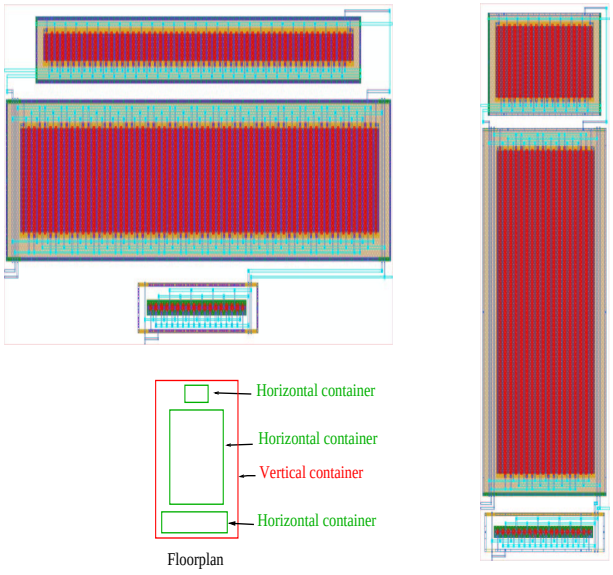

Fig. 8. Constraints: square (left) and 300 μm height (right)

On figure 9 on the left, a 500 μm height was applied (process CMOS, 0.6μm) to a GMC [8] (Only the 24 transistors are shown). The placement is much more complicated. On the right, a 100 μm height constraint was applied.

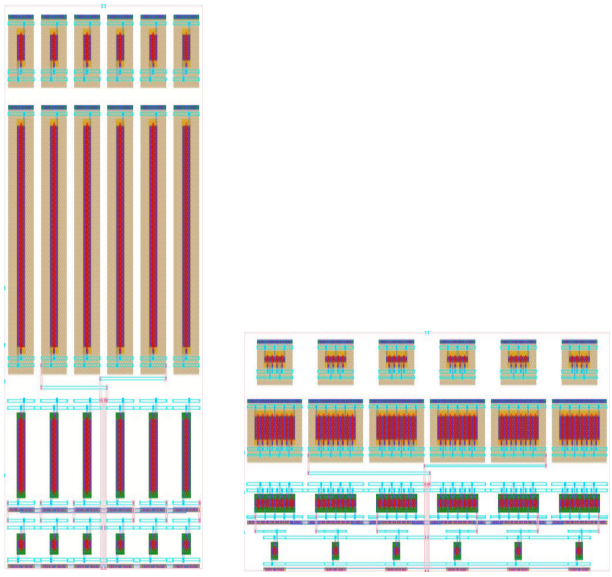

Fig. 9. Constraints: 500 μm (left) and 100 μm height (right)

CONCLUSION

We have shown that the placement based on slicing tree allows to manage the flexible layout of an analog IP, thanks to a shape function. This shape function is calculated at each level of the hierarchy using a simple set of mathematical definitions. Thus the aspect ratio of an analog IP Core can be easily predicted in a context of process and specification migration.
This method is part of the CAIRO+ analog design environment [9], under developpment at the ASIM department of the LIP6 laborartory.

AUTHORS

Pierre Nguyen Tuong is a PhD student in the ASIM Department of the LIP6 Laboratory
Dr. Marie-Minerve Louerat is in charge of the Analog thematic and Prof. Alain Greiner is the head of ASIM department.

Email: Pierre.Nguyen-Tuong@lip6.fr

Pierre et Marie CURIE University
LIP6 Laboratory
ASIM Department
4 place Jussieu
75252 Paris CEDEX 05
FRANCE

REFERENCES

[1] Gielen and Rutenbar, “Computer aided design of analog and mixed-signal integrated circuits,” *Proc IEEE*, vol. 88, pp. 1825–1852, December 2000.

[2] Hershenson, “Efficient description of the design space of analog circuits,” *DAC 2003*, pp. 970–973, June 2003.

[3] M. Dessouky and M.-M. Louerat, “A layout approach for electrical and physical design integration of high-performance analog circuits,” pp. 291–298, ISQED, 2000.

[4] Lampaert, Gielen, and Sansen, *Analog Layout Generation for Performance and Manufacturability*. The Netherlands: Kluwer Academic Publisher, 1999.

[5] Hastings, *The Art of Analog Layout*. Prentice Hall, 2001.

[6] Vancoreland, Van de Plas, Steyaert, Gielen, and Sansen, “A layout-aware synthesis methodology for RF circuits,” *ICCAD*, pp. 358–362, November 2001.

[7] M.Lai, D.F. Wong, “Slicing tree is a complete floorplan representation” *DATE-01*, pp. 228–232, 2001.

[8] H.Aboushady, F.Montaudon, F.Paillardet and M.-M.Louerat, “A 5mW, 100KHz Bandwidth, Current-Mode Continuous-Time Sigma-Delta Modulator with 84dB Dynamic Range”, *ESSCIR’02*, Florence, Italy, septembre 2002.

[9] P.Nguyen Tuong, M.-M,Louerat,”University Booth”, *DATE-04*, 2004.