

A New Area and Shape Function Estimation Technique for VLSI Layouts

Gerhard Zimmerman

FB Infomatik
Universitaet Kaiserslautern
675 Kaiserslautern, FRG

Abstract

Area estimation of IC layouts has become an important requirement for early design and top-down chip planning tools. Especially the relation of area and aspect ratio (shape function) is necessary for chip planning. Statistical models have been published with good results for standard cell blocks with near unity aspect ratios. This paper describes a new model for the prediction of shape functions for aspect ratios up to 1:5. The model is based on the shape and connectivity of adjacent cells. It can be used for many different design styles and has been tested for standard cell blocks and for the placement of general cells.

Categories: 6, 9

Introduction

With increasing complexities and design costs for VLSI circuits, accurate area predictions become more and more important. Total area estimates are necessary during architectural and logic design in order to determine the number of necessary chips and their functionality. Finding a feasible physical partitioning between and on chips is another problem where accurate estimates are necessary.

Also, top-down chip planning relies on the quality of area estimates. A top-down chip planner places rectangular modules of unknown layout and at the same time determines the shapes of the modules. Placement and shapes can only be optimized if the areas of the modules as a function of the shapes (shape function) can be estimated. These module shapes are the frames for the next lower level layouts. Any failure in the prediction may lead to very expensive design iterations.

Finally, the layout of standard cell blocks can be directed by a good shape function to the row number which results in minimal area without design iterations. Therefore, any improvement over the current area prediction methods can lead to significant savings in design cost and a better quality of the final products.

So far published area and shape function estimations have been based on statistical wiring length models [KuP86] or have been tuned to specific design tools [UKH85]. Although both models seem to be accurate for aspect ratios near unity, no experiments have been presented for a larger range of shapes. Our own measurements of standard cell modules show typical area

increases of 50% for rectangles with aspect ratios of 1:5, compared with square layouts. Also, both models are restricted to standard cell layout-styles. In [UKH85] floorplan sizes can only be calculated after the placement of blocks. Here we present an improved sizing model for a wide range of design styles, including chip planning. The model is applicable to all geometries which can be represented by slicing [SzO80] or come close to it. Since the model only provides estimations, deviations like jogs in slicing lines or a small number of pinwheels will not severely affect the results. Very good results have been achieved for standard cell designs.

Model Assumptions

The main assumption of the new model is a slicing geometry which can be represented by a slicing tree [SzO80]. No assumption is made about the direction of outlines or the orientation of rectangles with respect to the outlines. It is assumed that a structural hierarchy exists and that shape functions are known for the leafnodes. The level of the leafnodes is unimportant. Because of the latter, the model can be used at any level of refinement during the architectural and logic design. The model requires average routing grid dimensions as the only technology parameters. The model uses a set of track demand factors which are design style dependent. These parameters can be determined by statistical wiring models or from sample designs with a specific layout tool. Default values will be given in this paper. The track demand factors differ for routing around the cells (channels), through free routing tracks in the cells (built-in feedthroughs), or through tracks that are generated by stretching the cells. Therefore the model requires knowledge about built-in feedthroughs of the leafcells which can be used to reduce the necessary wiring space.

Providing shape functions and feedthrough counts for leafcells is a smaller problem than it appears. If, for instance, the structural design is mapped onto a standard cell library, the known dimensions of the standard cells represent the shape functions. Feedthrough numbers are also part of standard cell libraries. If the structure is only known at a higher level in the hierarchy, e.g. the RT-level, typical structural realizations of RT-modules can be found in libraries and shape functions can be based on transistor or gate counts. A statistical model as described in [RRZ84, Zim85] can be used. The accuracy of the predicted shape function depends on the accuracy of the shape functions of the leafcells. In the case of standard cells or macro cells the shape functions are known precisely. Therefore, the presented experiments with standard cell blocks show the uncertainty which is introduced by the model itself.

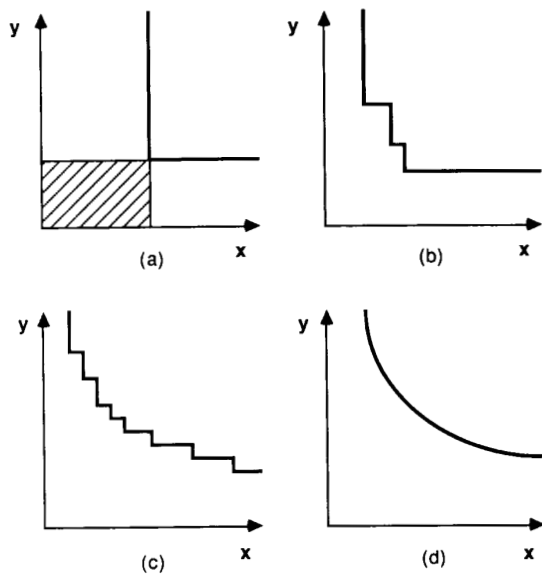


Fig.1. Shape functions for a (a) macrocell, (b) multi-shape cell, (c) standard cell block, and (d) flexible cell.

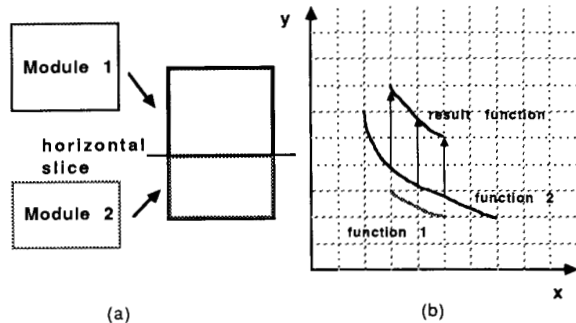


Fig.2. Geometric summation of the shape functions of modules 1 and 2 for horizontal cut.

The Sizing Model

The shape function [Ott83] of a cell is defined as the lower area bound of all possible rectangles of the cell. It is expressed as the x and y dimensions of the rectangles. If we assume that each rectangle can be extended in either the x or y dimension by empty space, we achieve Figure 1a for a cell with one fixed shape. Figure 1 also shows examples of a cell with several fixed shapes, a cell composed of standard cell rows and a totally flexible cell. More precisely, the shape function is a bounding curve [WOL86] that divides the first quadrant of the x,y plane into a dimension of feasible rectangles (bounded area) and infeasible ones. The bounding curve is part of the bounded area and represents the solutions with minimal area for a given x or y dimension. Piecewise linear bounding curves as in Figure 1c can be uniquely represented by the coordinates of the corners.

As it was observed by [Ott83], the combined shape function of two or more siblings in the slicing tree can be easily calculated, especially if the functions are piecewise linear. If the slicing line is horizontal as in Figure 2a, the x dimension is considered to be equal for all siblings and the resulting combined cell. The y dimension is calculated by adding the individual y dimensions of all siblings at each valid x dimension. Figure 2b shows the result. This process can be applied throughout the whole slicing tree, starting with the leaf nodes.

This method requires the knowledge of the slicing tree and of the orientations of the cut lines. The slicing tree is generated in many different ways throughout placement. Mincut [LaD86], clustering [RRZ84], or force-directed methods are some possible examples. One observation is that the results of the different methods are very similar in total area, although the individual geometry may be totally different [ScZ85, Pre87].

We choose the mincut algorithm [FiM82] as a fast means to generate an acceptable slicing tree. The inputs to this algorithm are the module netlist for which a shape function is required and an area value for each of its leaf cells. Since the shape functions for all leaf cells are required anyway, the minimum area of each cell is extracted and used to balance the tree during the mincut procedure.

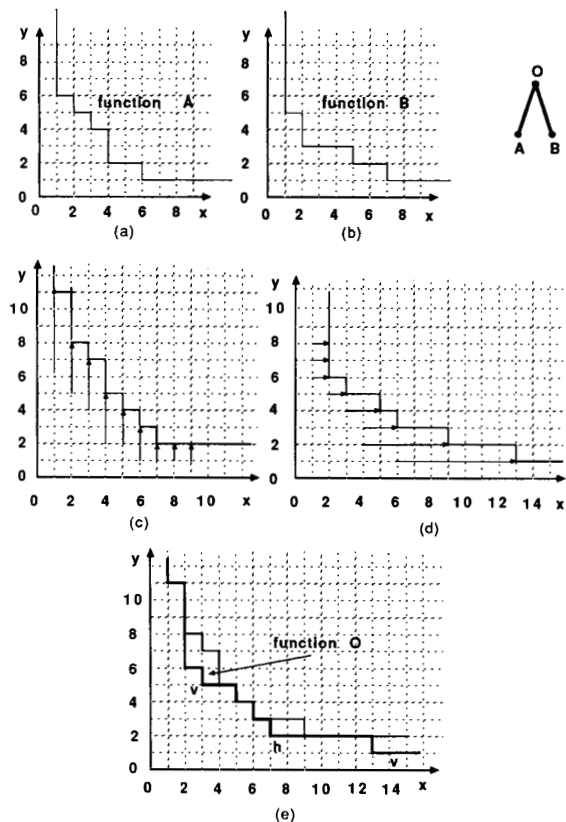


Fig.3. Calculation of the optimal shape function O and of the cut orientations. (a,b) Shape functions of the siblings A and B.(c) Shape function of the parent for horizontal cut. Arrows indicate the share of B. (d) Same as (c) for vertical cut. (e) Heavy line indicates the optimal shape function, v and h the corresponding orientations.

Mincut generates a binary slicing tree without outline orientation. This seems to be a problem during the computation of the shape function for all nodes of the tree. Different shape functions for horizontal and vertical orientation would be generated in every node, resulting in a large number of functions for the root node. The problem is solved in a very simple manner. First, when two siblings are combined, the shape function for horizontal and vertical cut are computed as shown in Figures 3c and d. Then, going back to the definition of the shape function, the lower bound of both functions is chosen at each x coordinate. This results in only one shape function for the parent of both siblings (Figure 3e). Segments of the shape function can be marked to represent the chosen outline orientation. This orientation is the optimal choice at each node of the tree and can be used in placement tools. It has been shown in [KLZ87] that all these orientations also represent the solution for the global optimum. Thus the shape function of the root cell represents the minimum area for all aspect ratios of the root for a given tree.

Wiring Estimation

The model so far has totally neglected the necessary wiring space within the root cell. In the case of standard cells, the result would be the total cell area. Instead of adding space based on an average wiring length estimate at the root cell [KuP86], wiring is added at each node of the slicing tree. We assume that each net is contained in the smallest bounding box of all connected leaf cells. Since we know the size of the bounding box, we can estimate the length of each net. Within this bounding box, a net will occupy more or less of horizontal and vertical routing tracks as shown in Figure 4a. We model this statistical behavior by the average percentage of a track (track demand factor) that is blocked by one net. Track demand factor depend on the design style, the direction relative to the cut line and relative to the row orientation in the case of standard cells. Table 1 shows typical values. If we multiply the number of nets in the bounding box with the appropriate track demand factor, and the track width, we get the wiring spaces xw and yw . These dimensions have to be added to the cell area as Figure 4c shows. If we further assume that the track demand factor and thus xw and yw are independent of the shape of the bounding box, as indicated in Figures 4b and d, the shape function of the corresponding node is simply shifted right by xw and up by yw as in Figure 4e. The numbers of tracks are reduced by available feedthroughs. Nonused feedthroughs are propagated up in the tree. For all nets that do not connect adjacent leaf cells, wiring space is added to the nearest common ancestor node. Multiterminal nets are handled respectively. The only nets not accounted for so far are nets leaving the root node. For these, additional routing space can be added to the root.

Another distinction has to be made between cells with pins at the perimeter (called "red" cells) and cells with area pins (called "pink" cells). Red cells are for example standard or macro cells. Pink cells are typically all nodes in the slicing tree, as long as nets leaving the node have not been accounted for. This addition will make a pink cell red, as described above for the root cell. Pink cells need more wiring space for additional nets, because these penetrate the cell. Wiring to red cells is only connected to the perimeter. Therefore, different track demand factors are used for red and pink cells.

There is a range of cells between red and pink. They are parameterized by their "redness". The compound of two "red" standard cells is still very red, whereas the assembly of many standard cells is nearly pink. The redness determines the

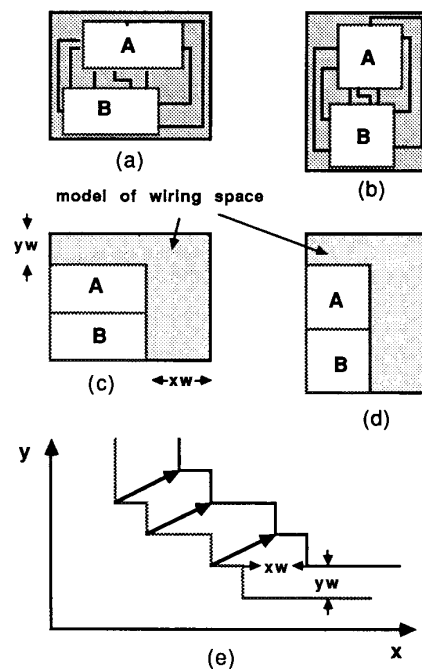


Fig.4. Wiring space correction of the shape function.

Table 1
Track Demand Factors for General and Standard Cells

Type	Cut Orientation	Net-to-Cut Orientation	Track Demand Factor	
			General Cells	Stand. Cells
pink	horizontal	parallel	0.5	0.5
red	horizontal	parallel	0.2	0.5
pink	horizontal	orthogonal	0.6	0.1
red	horizontal	orthogonal	0.4	0.0
pink	vertical	parallel	0.5	0.1
red	vertical	parallel	0.2	0.0
pink	vertical	orthogonal	0.6	0.4
red	vertical	orthogonal	0.4	1.0

weight of both track demand factors. The redness is automatically decreased by the distance of nodes from red cells.

Track demand factors and routing grids may be different for horizontal and vertical cuts. For example, standard cells in the same row can only be connected in a channel parallel to the row. Thus no wiring space is added in the direction of the row. For this reason, wiring is added to the shape functions for horizontal and vertical cut independently and the minimum of both is calculated afterwards.

All these distinctions result in a set of eight track demand factors. Table 1 gives default values for standard cells (horizontal rows only) and flexible cells (general cells). For specific tools these values can be adapted to yield higher model precision by examining layout experiments.

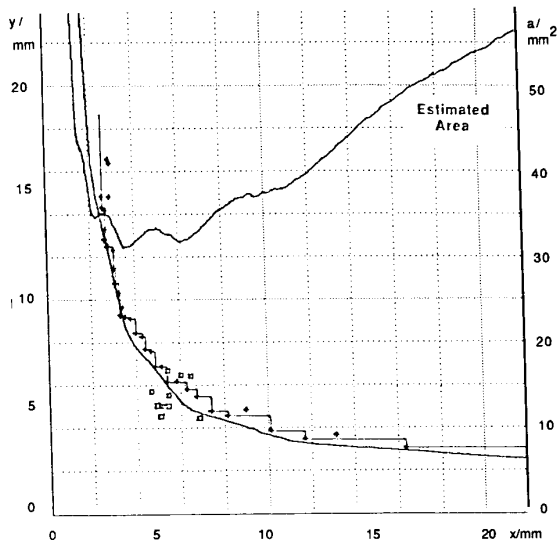


Fig.5. Shape function and measurements for the standard cell benchmark Primary1. + indicate measured layout dimensions for different aspect ratios. □ indicate measured layout dimensions for different layout tools.

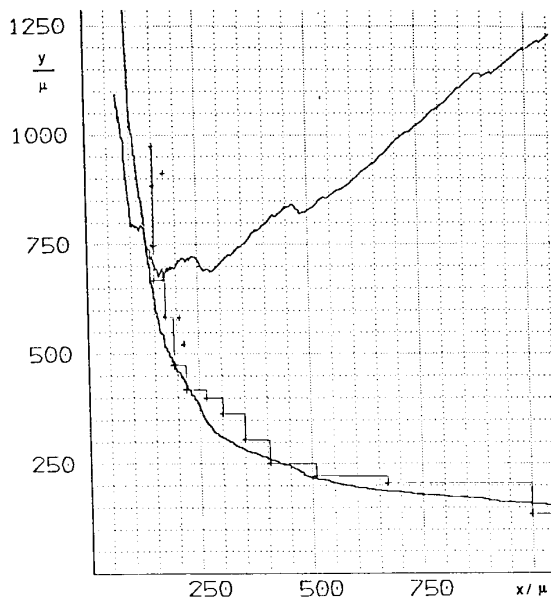


Fig.6. Standard cell example S6F.

Experimental Results

The quality of the shape function estimation was tested against experimental layouts. Accurate measurements can be achieved with standard cell layout systems. The default track demand factors in Table 1 were estimated before the measurements were performed and have been used without corrections. Adjustments are possible in order to tune the parameters to a

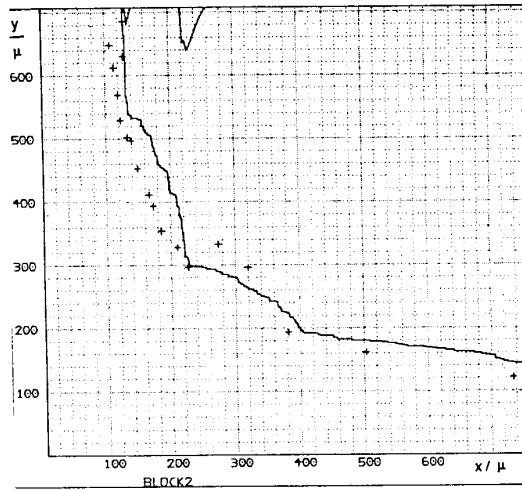


Fig.7. Standard cell example BLOCK 2.

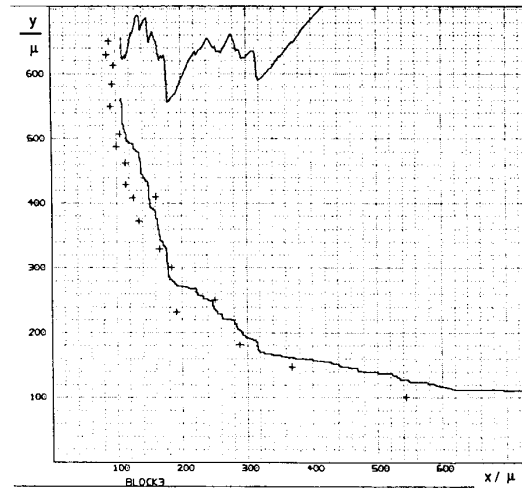


Fig.8. Standard cell example BLOCK 3.

specific layout system. Tuning was not necessary for the conducted experiments. Figure 5 shows the results for the benchmark example Primary1 [PDW87]. Squares show the results of layouts as presented by a number of participants at the workshop. The variations result from differences in the placement algorithms used. The results connected by the step function were obtained by J. Rose [Ros87] with the Altor standard cell place and route program by varying the number of rows between 5 and 40. The estimated shape function predicts these measurements astonishingly well. The estimated area curve shows a strong dependency of the total area on the x dimension or aspect ratio.

Figures 6 to 10 have been achieved in a similar manner with the industrial standard cell layout system VENUS III [HNS86]. The fit between the estimates and the measured values is excellent. The difference in the step width is explained

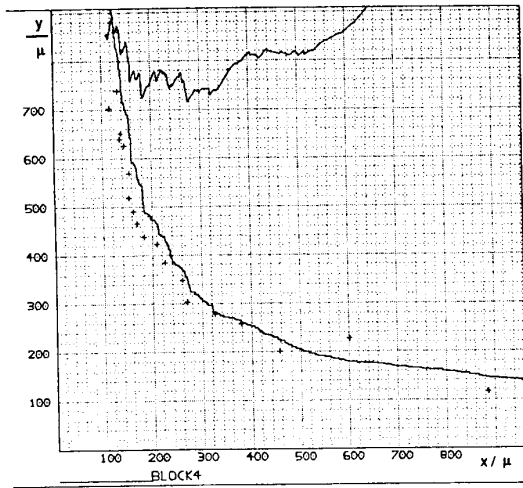


Fig.9. Standard cell example BLOCK 4.

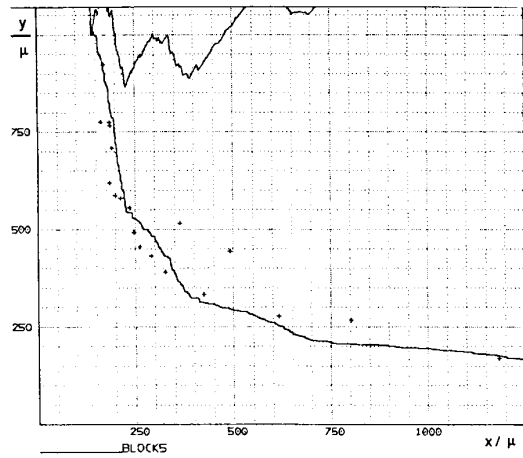


Fig.10. Standard cell example BLOCK 5.

by the fact that the model knows nothing about rows and is free to place cells anywhere. It is therefore surprising that all plots of the estimated placements resemble row placement strongly (see Figure 11). This gave us additional confidence in the applicability of the model.

The results of a quantitative error analysis is summarized in Table 2. For every significant measurement, the area of the nearest point on the shape function was computed and compared with the measured area. Significant means that there exists no other measurement with smaller x and y dimensions. Insignificant measurements show up above the measured step function (see Figure 5). The average error is the mean value of all area deviations. Table 2 also shows the aspect ratio range of each experiment. It can be concluded that the area of standard cell blocks can be estimated with an accuracy of 5 to 10% in the range of aspect ratios between 5:1 and 1:5.

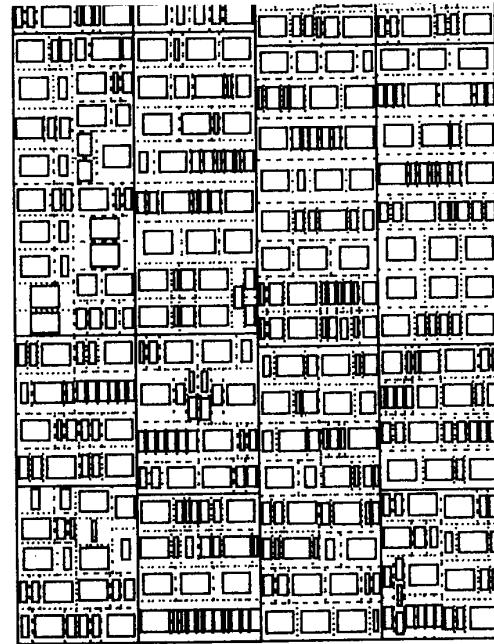


Fig.11. Section of a Primary1 placement, produced by the shape function generator.

Table 2

Average Area Estimation Error for Standard Cell Experiments

Cellname	Number of Std. Cells	Aver. Error %	Aspect Ratio Range	
			(x/y)max	(y/x)max
Primary 1	800	5.8	5.3	5.7
S6F	496	7.4	7.6	4.5
Block 1	113	6.4	2.6	4.8
Block 2	186	8.4	6.1	5.4
Block 3	117	8.8	5.5	6.1
Block 4	193	8.2	7.7	5.6
Block 5	318	7.3	6.8	5.5

Since layouts for different aspect ratios of complex VLSI chips could not be obtained, results of a floorplanning tool [RRZ84] were employed. This tool is a preliminary version and is probably not optimal in all cases. Figure 12 shows an example with 13 cells of fixed shape (macrocells). Floorplans were generated for different aspect ratios. Floorplanning is done by doing automatic placement, channel estimation, loose global routing and final channel sizing on the basis of nets per channel. An uncertainty about the channel widths remains, because detailed routing could not be done. In Figure 12 the floorplanning results are indicated by squares and the step function is the shape function. The shape function at least predicts the "good" floorplans (small total area). Currently, the differences can not be explained, but a closer match is expected as the floorplanning tool is improved. Also, the experimental results will be more precise as soon as chip assembly is available.

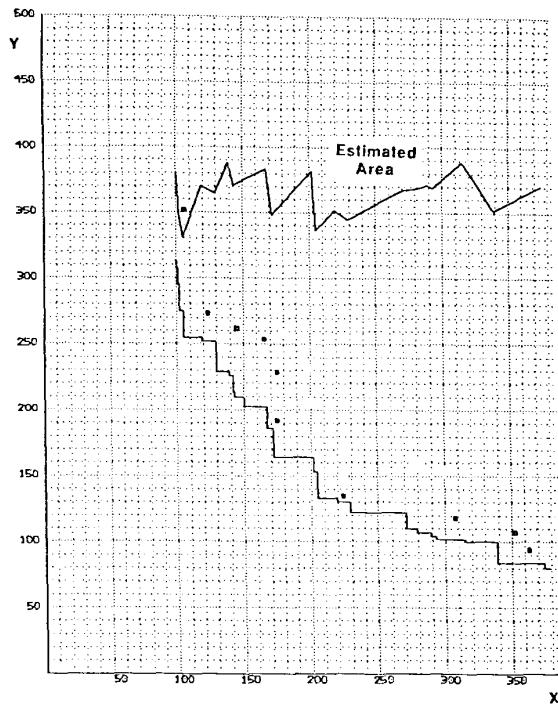


Fig.12. Shape function for general cell example with 13 macrocells. Squares show best chip planning results after global routing.

Conclusion

The presented shape function estimation method is an improvement over published statistical models, because it takes more knowledge about the geometry and about neighborhoods into account. Therefore it can predict the area of flexible cells for a large range of aspect ratios with a sufficient accuracy of 5 to 10%. More precision can not be expected, because no layout tool can produce consistently good results. The only assumption of the model is that the used layout tools try to obtain area minimization and that the resulting geometries resemble slicing geometries. This is true for a large range of design styles. The model precision is sufficient for most applications. The estimation tool can be applied hierarchically, because it accepts shape functions as inputs for the leaf cells at all levels of the hierarchy. Therefore, it supports hierarchical VLSI design styles.

This work was supported by the Deutsche Forschungsgemeinschaft and by Siemens AG Munich. It is the result of a combined effort of many scientists in my research group.

References

- [FiM82] C.M. Fiduccia and R.M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partitions", Proc. of the 19th Design Automation Conference, 1982, pp 175-181.
- [HNS86] E. Hörbst, M. Nett, and H. Schwärzel, "VENUS: Entwurf von VLSI Schaltungen," Springer Verlag Berlin, 1986.
- [KuP86] F.J. Kurdahi and A.C. Parker, "PLEST: A Program for Area Estimation of VLSI Integrated Circuits", Proc. of the 23rd Design Automation Conference, 1986, pp 467-473.
- [LaD86] D.P. La Potin and S.W. Director, "Mason: A Global Floorplanning Approach for VLSI Design", IEEE Trans. on Computer-Aided Design, Vol CAD-5, No. 4, 1986, pp 477-489.
- [Ot83] R. Otten, "Efficient Floorplan Optimization", Proc. ICCD 83, 1983, pp 499-502.
- [PDW87] Physical Design Workshop, Hilton Head, Hilton Head, 1987.
- [Pre87] B. Preas, "Benchmarks For Cell-Based Layout Systems", Proc. of the 24th Design Automation Conference, 1987, pp 319-320 (Results presented at the Physical Design Workshop in Hilton Head, 1987).
- [Ros87] J. Rose, Stanford University, private communication 1987.
- [RRZ84] P. Rao, R. Ramnarayan and G. Zimmermann, "SPIDER, A Chip Planner for ISL Technology", Trans. of the 21st Design Automation Conference, 1984, pp. 665-666.
- [ScZ85] F. Schreiner and G. Zimmermann "Comparison of Different Placement Algorithms for Chip Planning", (in German), Report of the SFB 124, University of Kaiserslautern FRG, 1985.
- [SzO80] A.A. Szeplieniec and R. Otten, "The Genealogical Approach to the Layout Problem", Proc. of the 17th Design Automation Conference, 1980, pp 535-542.
- [UKH85] K. Ueda, H. Kitzawa and I. Harada, "CHAMP: Chip Floor Plan for Hierarchical VLSI Layout Design", IEEE Trans. on Computer-Aided Design, vol. CAD-4, No.1, 1985, pp 12-22.
- [WoL86] D.F. Wong and C.L. Liu, "A New Algorithm for Floorplan Design", Proc. of the 23rd Design Automation Conference, 1986, pp 101-107.
- [KIZ87] A. Klein and G. Zimmermann, "Area Estimation of VLSI Chips", to be published.
- [Zim85] G. Zimmermann, "A VLSI Cell Sizing Model for Chip Planning", Report of the SFB 124, No. 15/85, University of Kaiserslautern, FRG 1985.