

Fig. 3. The distance between the saturation point and the drain ( $\Delta L$ ) versus  $V_D$  of the  $0.025\text{ }\mu\text{m}$  NMOS device biased at  $V_G = 0.8\text{ V}$ – $2.0\text{ V}$ , based on the conventional method and the new approach.

#### IV. CONCLUSION

In this paper, a unified triode/saturation model with an improved continuity in the output conductance suitable for CAD of VLSI circuits using deep sub- $0.1\text{ }\mu\text{m}$  NMOS devices has been reported. As verified by the experimental data, the formula shows an accurate prediction of the output conductance characteristics.

#### ACKNOWLEDGMENT

The authors would like to thank Y. W. Chang and K. W. Su for their help during the final phase of the paper.

#### REFERENCES

- [1] K. Sonoda, K. Taniguchi, and C. Hamaguchi, "Analytical device model for submicrometer MOSFET's," *IEEE Trans. Electron Devices*, vol. 38, pp. 2662–2668, Dec. 1991.
- [2] H. Wong, "A new approach to current-voltage characteristics formation for short-channel MOSFET's," *IEEE Trans. Electron Devices*, vol. 41, pp. 2480–2482, Dec. 1994.
- [3] H. C. Chow, W. S. Feng, and J. B. Kuo, "An improved analytical short-channel MOSFET model valid in all regions of operation for analog/digital circuit simulation," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 1522–1528, Dec. 1992.
- [4] N. D. Arora, R. Rios, C. L. Huang, and K. Raol, "PCIM: A physically based continuous short-channel IGFET model for circuit simulation," *IEEE Trans. Electron Devices*, vol. 41, pp. 988–997, June 1994.
- [5] B. J. Sheu, D. L. Scharfetter, P. K. Ko, and M. C. Jeng, "BSIM: Berkeley short-channel IGFET model for MOS transistors," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 558–566, Aug. 1987.
- [6] M. Fujishima and K. Asada, "A nonpinchoff gradual channel model for deep-submicron MOSFET's," *IEEE Trans. Electron Devices*, vol. 40, pp. 1883–1885, Oct. 1993.
- [7] B. J. Moon, C. K. Park, K. Lee, and M. Shur, "New short-channel n-MOSFET current-voltage model in strong inversion and unified parameter extraction method," *IEEE Trans. Electron Devices*, vol. 38, pp. 592–602, Mar. 1991.
- [8] T. Skotnicki, C. Denat, P. Senn, G. Merckel, and B. Hennion, "A new analog/digital CAD model for sub-halfmicron MOSFET's," San Francisco, CA: *IEDM Dig.*, pp. 165–168, 1994.
- [9] C. K. Park, C. Y. Lee, K. Lee, B. J. Moon, Y. H. Byun, and M. Shur, "A unified current-voltage model for long-channel nMOSFET's," *IEEE Trans. Electron Devices*, vol. 38, pp. 399–406, Feb. 1991.
- [10] P. Antognetti and G. Massobrio, *Semiconductor Device Modeling with SPICE*. New York: McGraw-Hill, 1988, p. 208.

- [11] P. K. Ko, "Approach to Scaling," in *Advanced MOS Device Physics*. N. G. Einspruch and G. Gildenblat, Eds. New York: Academic, vol. 18, 1989, pp. 25–29.
- [12] M. Ono, M. Saito, T. Yoshitomi, C. Fiegna, T. Ohguro, and H. Iwai, "Sub-50 nm gate length N-MOSFET with 10 nm phosphorus source and drain junctions," Washington, DC: *IEDM Dig.*, pp. 119–122, 1993.

### A Method for Area Estimation of Data-Path in High Level Synthesis

Hortensia Mecha, Milagros Fernández, Francisco Tirado,  
Julio Septién, Daniel Mozos, and Katzalin Olcoz

**Abstract**—This paper describes a new method to estimate the area of data paths generated during a High Level Synthesis (HLS) process, when the information concerning the circuit is not yet complete. Our method is more accurate and considers more factors than those used by other HLS systems of which we are aware. Our main concern is the interconnection area, often neglected by HLS systems, which has a strong influence on the final circuit area being optimized, as well as a high dependency on the technology used and on the circuit area itself. Predicting the area of a design layout with accuracy is important because it allows one to foresee whether the design will satisfy the area constraints, and will lead the allocator towards the best design among several possibilities with guarantees. Our estimation of the final standard-cell layout area are similar, or even more accurate, than those obtained following methods used by low-level design systems, which have much more information available. Due to the performance penalty their relatively high complexity will produce, these methods are unusable in an HLS system exploring a wide design space. Our estimation, on the contrary, has a low complexity and can be repeated time and again as the HLS design space is searched.

#### I. INTRODUCTION

While accomplishing a High-Level Synthesis (HLS) process, decisions based on information about the circuit physical features must be often taken. The most significant subtasks of HLS are the scheduling of operations according to different steps of time and the allocation of hardware to perform them. In this paper, we focus our attention on the estimation of the influence of the physical features on the goals of the hardware allocator.

The hardware allocation decides on the modules (Functional Units [FU's], Registers and Multiplexers) going to be used and the interconnections between them. Its input is usually a scheduled graph and its main goal is generating a design with minimum chip area. The final area of an integrated circuit depends on the number and the area of the modules and the interconnections. So, the area of each module and of the interconnections should be previously estimated and included in the global cost function to be optimized.

Some authors assume that module area could easily be known if a module library were available [1], [2]. This could be true for macrocell-based design, but not for standard cells. Though the area of the cells constituting the module will remain constant, due to the

Manuscript received May 10, 1994; revised January 20, 1995 and September 19, 1995. This work has been supported by Spanish Government Research Grant CICYT TIC 94/0725-C03-02 and CEE project HCM CT94-0459. This paper was recommended by Associate Editor R. Camposano.

The authors are with the Departamento de Informática y Automática, Universidad Complutense de Madrid, E-28040 Madrid, Spain.

Publisher Item Identifier S 0278-0070(96)01843-X.

module external interconnections, two factors must be considered when the cells are placed in the circuit.

- 1) The relative positions of the cells may change;
- 2) The width of the channels used for the interconnections grows.

This means that the lengths of the interconnections internal to the modules, and thus the actual module area, can be quite different when placed alone than when it is used as part of a larger design. On the other hand, the area of the internal and external interconnections of a module is highly dependent on the IC design tool and on the technology used in the process. In fact, it remains unknown until the layout of the circuit is generated. This process is very slow and thus, it is not possible to carry it out each time a feasible design is obtained.

Hardware allocation needs a fast and reliable method to estimate the design area (module area and interconnection area) which must be accurate enough to predict whether a certain design is better than another one, or whether it satisfies the user constraints. This does not mean, however, that this method has to be necessarily capable of getting the exact value of the final design area.

These considerations lead us to believe that area estimation is a very interesting research topic in HLS. There exist three main types of hardware allocation algorithms.

1) **Constructive**—allocates hardware while minimizing the final cost. The obtained solution is the best that can be reached by the algorithm.

2) **Iterative**—generates new solutions, each one better than the previous one, starting from an initial solution.

3) **Iterative-constructive**—a combination of the two types above. They create an initial solution of a certain quality with a constructive algorithm and they produce improvements on it. Each of these types of algorithms sets different requirements on the area estimations.

**Constructive** algorithms need to know the area of the interconnections and of every module in order to make decisions during the design process. For example, if a FU output needs to be stored, it has to be decided whether an existing register should be reused (by adding a multiplexer and an interconnection) or a new register should be added to the design. In both cases, the increase in area due to the new interconnection has to be estimated, although the total number of modules in the final design is not known yet. This estimation should be very fast, because it has to be done very often in a design generation. It must be accurate enough to decide which will be the best option for the final design, based only on information about the design part already generated.

For *iterative* algorithms it is necessary to estimate the area of each of the generated designs, when the number and type of modules and interconnections are known. In this way, we get rid of those designs that do not satisfy the users constraints, and are able to decide if a design is better than another. For **iterative-constructive** algorithms, it is necessary to make the estimation of the final area as well as the area increase produced each time a decision is made during the design process.

To sum up two kinds of estimation-related tasks must be carried out during the synthesis process.

- 1) Estimate the area increment when a new interconnection is created during the design process, without knowing the final number of modules and interconnections yet. This area depends on the types of the interconnected modules, on the placement and routing algorithms and on the design technology.
- 2) Estimate the final circuit area without generating the layout and without performing complex computations.

The estimation of the interconnection area is very important because it can reach the same magnitude order than the standard cell area, and has a great influence on the results and cannot be

overlooked. This influence should be extended to the interconnections internal to the modules to refine the rough area estimations available in the libraries obtained for the placement of each separate module.

This paper is structured in seven subsections. The first section summarizes the previous work on these problems. The following one presents the research framework. The third section explains how to compute the average length of a point-to-point interconnection and it concludes by defining two different kinds of interconnections. The next section deals with the problem of the multimodule interconnections, and the following one uses all the previous results to compute the interconnection area during the design process. In Section VI, examples are shown and at the end of the paper the conclusions are presented.

## II. PREVIOUS WORK

Most of the systems performing good estimations of interconnection area need to know everything (all the modules and interconnections) included in the integrated circuit. This means they need to know the RTL structure of the circuit, and the estimation is not of one interconnection area but of the whole circuit area. According to our references, these systems are able to predict the interconnection length and the final area with an error below 10% of the actual layout. They are quite accurate and their computation times are lower than placement and routing algorithms computation times. Nevertheless, they are too slow to be incorporated as part of a HLS tool.

According to [3], we could classify these methods into three categories

1) **Theoretical models**—are based on a mathematical description of the design physical features. They do not care for the particular details of each design. They only require assumptions on the interconnection length distributions. Probabilistic methods [4] and [5] belong to this category.

2) **Empirical models**—produce mathematical expressions of the physical features, extracting information from the design. This class includes the models based on Rent's rule [6], (e.g., the system described in [7]). This rule is a relationship between the number of external connections of one partition inside the circuit, and the number of standard cells existing inside this partition. Another example of this model is [4], already mentioned in the previous category, because it consists of a combination of both models.

3) **Procedural models**—are based on relationships derived from the knowledge of the actual design process interconnection structure of the design, and physical layout rules [3], [8], [9]. Kurdahi in [9] presents an analytical-constructive method for area and shape function prediction. The input design is partitioned recursively until it reaches a user selected level. The analytical model is used to predict the shape functions of the leaf cells. The shape function for the whole design is obtained by traversing the slicing tree bottom-up. Sechen [10] suggests an interconnection length estimator that gives accurate estimations for small designs, but it is only applicable to the sea of gates design style. On [11] the data-path is divided into bit-slices and each of them is assumed to be placed in a standard cell row. The channel length depends on the number of cells of the bit-slice, and its width on the number of tracks needed to connect every net of the bit-slice. It uses the min-cut algorithm to place all the components in a single row and the left-edge algorithm to assign the tracks. In fact, these methods allow us to decide whether a complete circuit design is better than another one, or whether it satisfies the user constraints. However, they do not permit the decision among different choices to be taken during a HLS process, because some circuit elements are not allocated yet and, therefore, it is not possible to predict the area of an interconnection. Besides, these algorithms are too complex to be included in a HLS tool, where the area of a lot of different designs

has to be estimated. On the other hand, all HLS systems need to use estimations of the circuit quality, though only some of them take into account the interconnection influence on the results.

HIS system [12] tries to obtain the maximum performance while using the minimum hardware, so it attempts to reuse the hardware as much as possible. The circuit cost is calculated in terms of the number of elements of each type instead of their area, so the results are not always realistic.

There are some systems that consider the interconnection influence on the final area. For example, Chippe [1] uses a cost function that depends on the number of each type of element, including the interconnections, modulated by parameters chosen by the user. The choosing of these parameters has a great influence on the results, and thus its reliability depends on the expertise of the user.

A similar system is explained in [2], where the cost function is the same as before except for two differences: the factors multiplying the number of each element type are the module areas, and it takes into account the total execution time. In this system, the area of an interconnection is calculated empirically for each layout type. This solution is quite far from reality because it fixes the value for one interconnection and does not consider the particular design characteristics. These features have a great influence on the interconnection area. For example, for a circuit with very few modules, the average distance between two of them (and so its average interconnection length), is smaller than for a circuit with many modules.

Something similar happens in SAW system [13], where the EMUCS allocator uses a fixed value for the area of an interconnection, without bearing in mind the whole circuit size.

Among more recent systems, in [14] a relationship is derived between the number of buses and the area needed for interconnection, multiplexers and tri-state drivers. This system presupposes that all the FU's connected to a bus will be placed together. Thus, the length of a bus is a linear function of the layout width of the FU's directly connected to the bus, while the width of a bus is a linear function of the bus bit-width. These estimations are complemented in [15] where the area of registers is studied.

There are two objections to this method. First, the exact module width cannot be known before generating the final layout due to the effect of the internal interconnections. And second, the placement of a FU in a circuit depends on the modules it is connected to, and it is not possible to predict whether all the FU's connected to a bus are going to be placed together.

Nourani [16] develops a module-level area estimation assuming that interconnections inside a module are placed in different layers from the interconnections outside the modules, and that each module is placed in a single standard-cell row. This latter assumption is often wrong for modules made of several standard cells. The estimation algorithm performs two phases. During the first one, all modules are placed in a row, placing closer those more heavily connected. Then, the number of tracks needed to implement the interconnections is computed. In the second phase, the row is folded to obtain a multirow design to satisfy a certain aspect ratio. If the modules are very large, it is necessary to split them into submodules and to apply them the same algorithm. This algorithm is faster than those using standard cells to compute the estimations, but it is still too complex to be included in a HLS process. Furthermore, it is not possible to use it throughout the design process, when the information related to the circuit is not complete.

In [17], Rim presents an heuristic that combines binding and floorplanning in order to use layout information. For each control step, module and register binding is performed followed by floorplanning. The layout information is used to estimate the interconnection cost

for the binding in the following control step. However, as the circuit information is incomplete, the obtained interconnection cost is not highly realistic. In addition, fulfilling the floorplanning in every control step increases the algorithm complexity.

A technique for lower bound estimation of the circuit area is studied in [18]. Since this is performed before scheduling, it does not take into account the interconnection area.

### III. RESEARCH FRAMEWORK

Our research goal is performing estimations about interconnection influence on the final circuit area, and to use them during the design process of a HLS tool. Thus, the estimations need to have two main features.

- 1) Reliability—in order to make decisions during hardware allocation driving the design process to optimal and realistic results.
- 2) Simplicity—because they must not severely increase the global execution time of the HLS tool.

In order to obtain reliable estimations, it is necessary to connect a HLS tool with a tool for designing integrated circuits, and then to be able to study the working conditions of the design tools as well as to evaluate the quality of the predictions. Carrying out the experiments would only be possible through the generation of a system that fully automates the interactions with the IC design tool, because manual implementation of designs takes too long. On the other hand, in HLS it is useful performing the connection between both tools to complete the integrated circuit design process until layout generation.

Our choice was CADENCE, a design tool using standard cells and macro cells for layout generation. Up to now only standard cells have been considered by our system. The connection with our HLS system is explained next. The output of the HLS tool is a netlist of modules and interconnections. Every module is previously designed and added to a particular system library. The netlist is translated to an EDIF (Electrical Design Interchange Format) format, including the libraries used in the design, the interface of every module in the library, and the list of modules and interconnections. This format is translated into a design made of standard cells and interconnections in an internal CADENCE format.

A command file is generated to automate the placement and routing. This file is what CADENCE uses to process the design and to generate the layout without manual interaction.

Fig. 1 shows the whole design process, from the behavioral description to the layout generation. It should be pointed out that the technological features have to be specifically considered in order to get accurate estimations of the area of both modules and interconnections. The estimations presented below will focus on standard-cell design style. They could also be applied with minimum alterations to macro-cell design style, although they could not be applied, for example, to gate array style.

A design made of standard cells has a series of rows of cells with the same length and height, and channels between the rows to make the interconnection routing. Each channel has several tracks, each one parallel to the standard cell rows (see Fig. 2). One or more interconnections run along each track. Most of the vertical interconnections are routed in a different layer to avoid the interconnection crossing. Using this knowledge, it is possible to get a first estimation of the area of one interconnection, which will be improved later with new experiments.

### IV. AVERAGE INTERCONNECTION LENGTH ESTIMATION

In this section, a method to estimate the average interconnection length value for a given design is shown. This method uses the

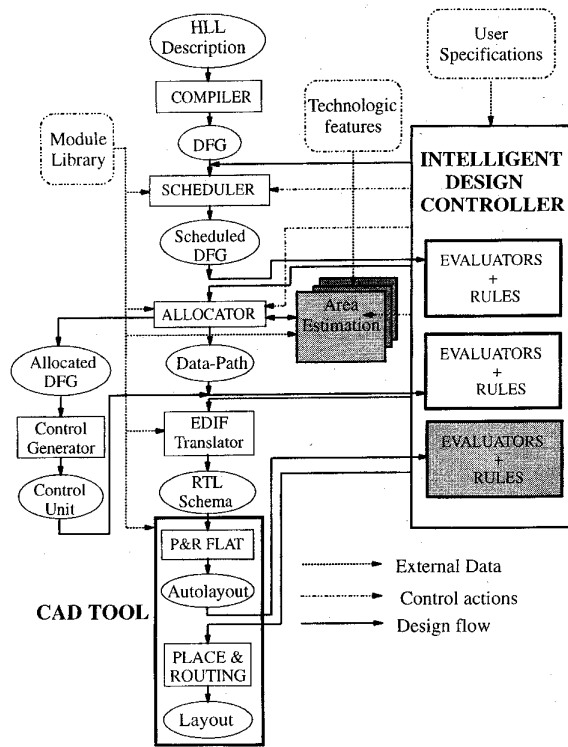


Fig. 1. Design process.

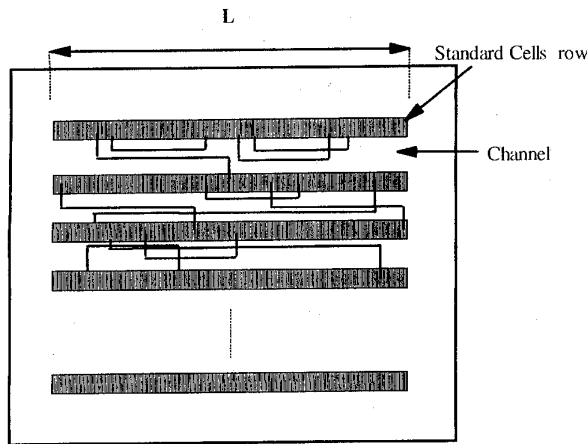


Fig. 2. Standard cell layout model.

total number of modules and interconnections of a complete design, information typically known by a hardware allocator.

Two metal layers are used for making the routing. The first one is used for horizontal interconnections and the second one for vertical ones. The standard-cell rows and the channels for horizontal interconnections are placed in the first layer (see Fig. 2). The second layer is only used for routing, so there is enough room for vertical interconnections, and there is no need to take into account their influence on the final area. Therefore, the estimations of the interconnection length will only consider the horizontal segments. This model is quite realistic and valid for any standard-cell-based CAD tool. Let us assume a design (see Fig. 2) with  $C$  cells in each row. Its standard cell row length is  $L$ , and the average cell length is equal to  $L_{cell}$ . If all the connections between two cells are equally

probable, the average interconnection length ( $L_i$ ) could be calculated by adding the length of all the different interconnections ( $L_{int}$ ) and dividing by the total number of possible interconnections ( $N_{total}$ ):

$$L_i = \frac{\sum_{n=1}^{N_{total}} L_{int_n}}{N_{total}}.$$

That is

$$L_i = \frac{\sum_{j=1}^{C-1} \sum_{h=j+1}^C (h-j) * L_{cell}}{\sum_{j=1}^{C-1} (C-j)}.$$

By developing the formula it will be obtained

$$L_i = L_{cell} * \frac{C+1}{3}.$$

If  $C$  is large enough

$$L_i = L_{cell} * \frac{C}{3}.$$

By replacing  $L = L_{cell} * C$  it will be obtained

$$L_i = \frac{1}{3} L.$$

If the circuit is assumed to be a square with an area  $A_{total}$ , then  $L$  is equal to  $\sqrt{A_{total}}$ , and the average interconnection length is

$$L_i = \frac{1}{3} \sqrt{A_{total}}.$$

The circuit area is the area of the modules plus the area of the interconnections. The module area is the standard cell area plus the area of the interconnections internal to the modules. If the total number of interconnections (internal plus external) is  $N$ , then the average interconnection length is:

$$L_i = \frac{1}{3} \sqrt{A_{cell} + N L_i W_i}.$$

In this formula  $A_{cell}$  is the average cell area, and  $W_i$  is the interconnection width, which is a constant for each technology. This is a second order equation in  $L_i$ , and it is possible to obtain the average interconnection length as a function of the standard-cell area and the total number of interconnections of the circuit.

An area estimation based on this model has been obtained for several designs. The area values estimated by our system, as all those included in this paper, are expressed in gate-equivalent units (ge). Fig. 3 shows how this estimation, if compared to the layout area obtained by CADENCE for each design, gives rather poor results. The estimation gives results even greater than those of the rough estimation made by the design tool previous to placement and routing, which the early design decisions are based upon. To find the reason for such a difference, it would be useful to study the designs after the modules have been placed and the interconnections routed. It can be seen then, that all the modules that are heavily connected are placed relatively closely, and that the interconnections between them have a length shorter than the average value calculated with the formula above. To explain this characteristic, it is necessary to keep in mind the purpose of the placement and routing algorithms: minimize the final chip area. Then, it is necessary to place the connected modules as closely as possible in order to get short interconnections. Therefore, using the same length for all the interconnections produces an overestimation in the total area of the designs. To solve this

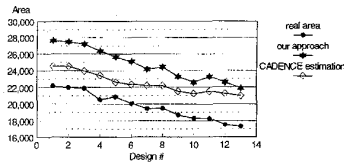


Fig. 3. Our area estimation compared to CADENCE estimation and real chip area.

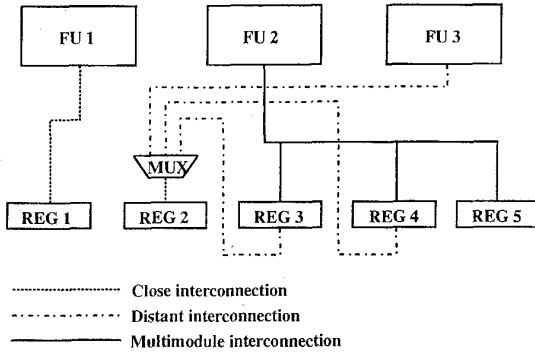


Fig. 4. Register transfer level design fragment.

problem, it becomes necessary to distinguish between two types of point-to-point interconnections.

1) **Close interconnections**—if the output of the source module is the only one connected to the destination module input. An example is a multiplexer connected to the input of a FU (see Fig. 4). This kind of interconnection will be very short because the placement algorithms will try to place the two modules very close (see Fig. 5). It is possible to assign to it a fixed value equal to the average standard cell length. The interconnections inside a bit-slice belong to this group. For example, a 4 to 1 multiplexer of 16 b is made of four 4 to 1 multiplexers of 4 b. These ones are composed of two 2 to 1 multiplexers of 4 b connected to the input of another 2 to 1 multiplexer of 4 b. This module is available in the standard-cell library. These three multiplexers will be placed as close as possible, and the internal interconnections of the 4 to 1 multiplexers of 4 b will be of the close kind. However, the interconnections between the four 4 to 1 multiplexers of 4 b will not belong to this class.

2) **Distant interconnections**—connect the output of a module with a multiplexer attached to the input of another one. An example is several FU's and registers outputs connected to the same register input (see Fig. 4). These interconnections are longer than the previous ones (see Fig. 5), and their average length can be taken equal to the average interconnection length calculated before. The interconnections internal to the modules connecting different bit-slices, also belong to this group.

#### V. MULTIMODULE INTERCONNECTION TREATMENT

All the estimations made so far concern point-to-point interconnections. One of the reasons for the poor results shown in Fig. 3, apart from those explained in the previous section, was that interconnections linking more than two modules (one module output and several module inputs) were in fact split into several point-to-point interconnections. However, in order to get better results, interconnections joining three, four, or more modules together should be treated in a different way. This section will deal with the problem of getting a more accurate estimation for the length of such interconnections.

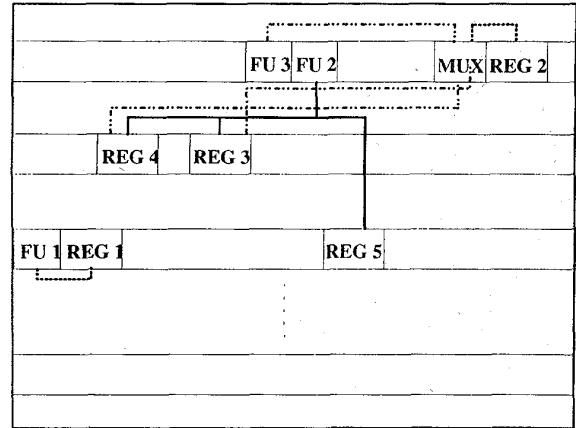


Fig. 5. Layout level design fragment.

Let us imagine a multimodule interconnection joining the output of one module to the input of  $N_{\text{mod}}$  modules, within a square design with a side value equal to  $L$ . The length estimation for such an interconnection should give, for  $N_{\text{mod}} = 1$ , the same value as the point-to-point average interconnection length ( $1/3 * L$ ), obtained above. On the other hand, for large values of  $N_{\text{mod}}$  the estimation should tend towards  $L$ . Also, it is interesting to point out that length increments are larger for small values of  $N_{\text{mod}}$  (2, 3, 4, or 5 modules), and that they become almost insignificant as  $N_{\text{mod}}$  grows. Such requirements are met even by a very simple interpolating function, such as

$$L_{\text{total}} = [1 - (\frac{2}{3})^{N_{\text{mod}}}] * L.$$

If the interconnection links a source to  $N_{\text{mod}} - 1$  modules, when a new one is connected the length increment will be

$$\Delta L_i = (\frac{2}{3})^{N_{\text{mod}}-1} * \frac{1}{3} L.$$

This formula is very easy to compute every time a new module is connected to the other ones, so its execution time does not overload the allocator execution time. Besides, the results obtained are quite suitable and, along with the point-to-point interconnection length estimation, lead to area estimations that very closely match the actual areas of the designs generated by the CAD tool after hours of processing time. This will be shown below with the help of several examples.

#### VI. INTERCONNECTION LENGTH ESTIMATION DURING THE DESIGN PROCESS

A HLS tool [19] converts a high level specification of the behaviour of a circuit into a structural design in terms of an interconnected set of register transfer level components. As in Fig. 1, the first subtask involved in this process is scheduling, which assigns each graph operation a time step, and thus determines the timing features of the generated circuit. Then, hardware allocation is performed with the goal of obtaining a circuit with minimal chip area for the previously scheduled graph. Such area optimization will be successful if the cost function used is actually an area estimation as accurate as possible. In addition, a fast estimation technique is crucial for an allocation algorithm exploring a wide design space, if we want the algorithm to be able to get good designs in reasonable search times. Finally, the estimation will have to be easily integrable in the exploration algorithm. In this way, as allocation decisions are taken or discarded, design area estimations are accordingly

updated and bounding decisions can be taken based upon them, if necessary for search to speed up. In this section, we explain how our interconnection area estimations, meeting the two previous requirements stated above (accuracy and computing speed), are integrated into the allocation design space search accomplished by our HLS system.

Before allocation can take place, a rough area estimation can be done that will allow us to check if the area constraint set by the user can already be met. This estimation is based on several factors. The first one is that, after scheduling is accomplished, information about how many variables have to be stored during hardware allocation is available. Moreover, the value of a close interconnection is constant and is known in advance. Finally, the value of a distant interconnection will change depending on the module area and on the number of interconnections. This value has to be calculated at this stage with only a very small amount of information available (i.e., only the minimal register number and operator parallelism are known, instead of an actual FU's set), and the result is not accurate, though it is a minimal value that can be used for the purposes explained above. The distant interconnection value used at this stage is based on a rough estimation of the future circuit area,  $A_{est}$ :

$$L_i = \frac{1}{3} \sqrt{A_{est}}.$$

Also, as the allocation proceeds, the area increments produced by decisions involving the addition of an interconnection to the design is a problem that must be dealt with. One must bear in mind that some kind of prediction has to be made about the final design area (and thus on its distant interconnection length value), since only a part of it is already known. Our decision has been to use an interconnection area increment for the allocation decisions involving interconnections in order to penalize the design space search times as little as possible. The estimations are based on the module features of the previous final design obtained by the search algorithm.

During the earlier stages of allocation, when the first design satisfying the area constraints has not been found yet, the value used for the distant interconnection length is the rough estimation shown above. Then, when this first design has been reached, the actual number of circuit modules and interconnections is known exactly for this design, and the distant interconnection cost can be recomputed. This value is already quite reliable and allows us to compute again the circuit total area.

This process is repeated time and again as the search proceeds and new designs (each one better than the previous one) are generated. All these designs are reached using an interconnection length value based on the previous design features, and once obtained, their areas are recomputed in order to correct possible errors, and then used to compute the interconnection length value for the next design process.

Even though for the first design the difference between the rough interconnection value used at the start and the recomputed one is significant (about 20%), in later designs this shifting has a very small influence on the results of the estimation, because the number of FU's, registers, and interconnections changes only slightly as the optimization continues.

Fig. 6 shows, using an example, that the error due to this approach is minimum. This picture shows how the area estimation for one distant interconnection changes as the design process of a fifth order filter proceeds. During the first design process, the estimation of the interconnection area is 106 ge; when the estimation is computed again, after the first solution is reached, the obtained value is 136 ge (the error of the first estimation is about 23%). The first value is significantly smaller than the rest obtained during allocation, because

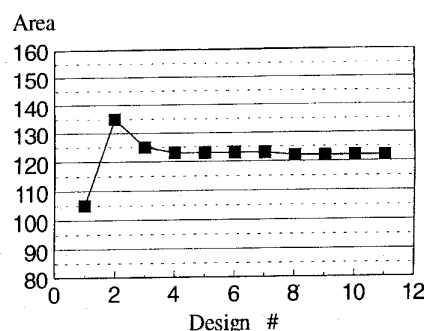


Fig. 6. Interconnection area estimation during design process.

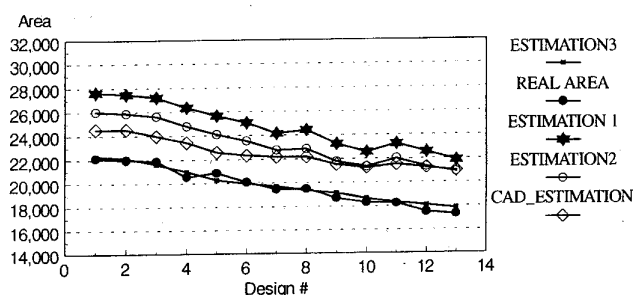


Fig. 7. Different area estimations compared to real design area for 5th order filter.

the number of interconnections and multiplexer area could not be taken into account in the estimation. These two factors have a great influence on the final circuit area, and on the distant interconnection area as well. The value 136 (obtained from design 1) is used as an estimation for design 2. It is computed again after hardware allocation giving a real value of 122 ge. Error now is 11%. Nevertheless, for the rest of the allocation process the errors induced are smaller than 1%. Thus, the use of the estimation of the previous design does not cause loss of accuracy.

## VII. EXAMPLES

The examples below will show the validity of the estimations just explained. The main purpose of our estimations of interconnection area is driving the design process towards better designs. It is also important to obtain good estimations of the whole circuit area, to really know whether a design has an area fitting the constraints specified by the user.

The first example (see Fig. 7) shows the results given by the different estimations presented throughout this paper, for fourteen designs obtained for the fifth order elliptic filter example [20]. CADENCE estimations and the actual design areas are also shown.

There are four estimations presented in the figure.

- 1) ESTIMATION1—is computed using the average interconnection length for every interconnection.
- 2) ESTIMATION2—considers two different types of interconnections: close and distant, but all the interconnections are supposed to be point-to-point interconnections.
- 3) ESTIMATION3—is the same as Estimation 2 but considering multimodule interconnections, also.
- 4) CAD\_ESTIMATION:—is the estimation made by CADENCE. It must be noted that:

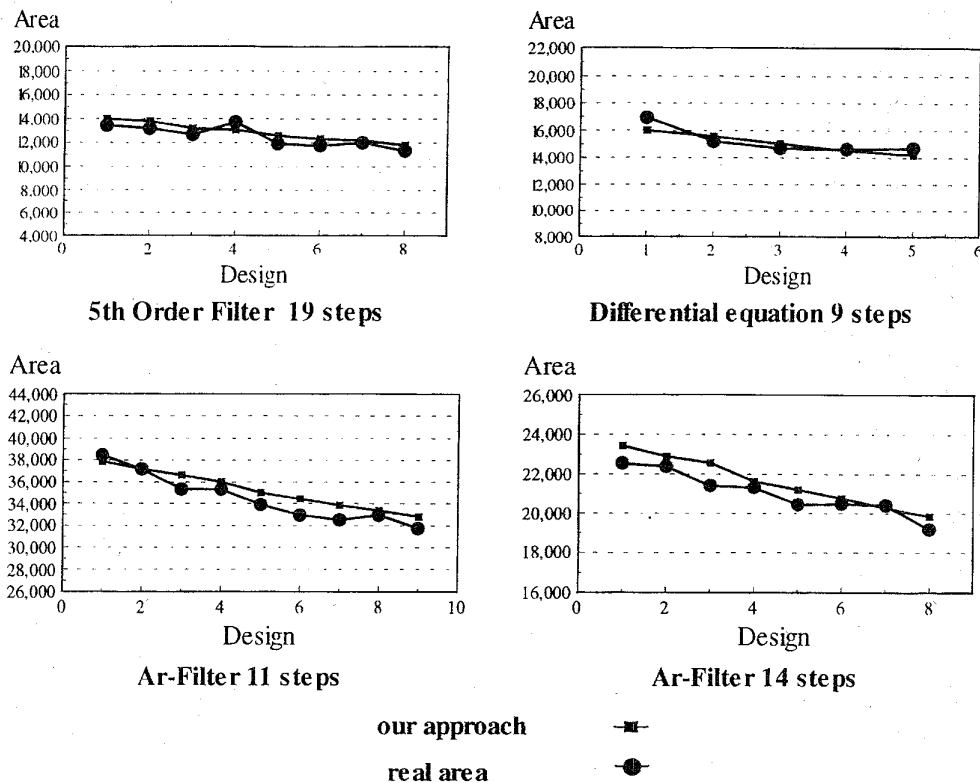


Fig. 8. Our estimation compared to real design area for several examples.

a) ESTIMATION2 is quite close to CADENCE estimations. This estimation is useful to drive the design process, because it permits a differentiation between two designs showing which one is the best. Nevertheless, it is not accurate enough to be used for comparing the design area with the user constraints. For example, if the maximum area constraint is  $M$ , some designs will not be accepted although its real area value might be smaller than  $M$ . It must be noticed that CADENCE generates this estimation only after a flattening process, with much more information available than our HLS tool.

b) ESTIMATION3 is accurate enough to allow good design process driving. Also, the error in the estimations is below 5%, quite a good value compared to other systems. It must be noticed that any design with an area estimation difference respect to the best design lower than the average error could be chosen as a solution of similar quality.

Fig. 8 shows the results given by this estimation for three different examples: the fifth order filter [20], the differential equation solver [20], and the ar-filter [21], the latter with two different schedulings. These designs were obtained through a design process that took few minutes, or even seconds for the differential equation example. These results are compared to the actual design areas generated by the CAD tool after hours of processing (placement and routing performing) on a Sun platform (at least 40 min. for each design). In all cases the error remains below 5%.

### VIII. SUMMARY

In this paper, a method to estimate the area of a circuit during its design process has been described. Since the standard-cell area is known, the attention has been focused on the interconnection area. The interconnections have been classified in two groups: close and

distant. For each of them, a different area estimation has been developed. Multimodule interconnections have been treated separately by using an interpolation function intuitively and experimentally justified. Integration of the estimation technique in a HLS tool has been explained in detail, and the validity of interconnection area prediction based on information about previous designs has been justified. Finally, the complete model has been applied to several known examples, giving evidence of how the estimated chip area has always remained very close to the measured area, in addition to an improvement of the design process driving.

### REFERENCES

- [1] B. M. Pangrle, "Splicer: A heuristic approach to connectivity binding," in *Proc. 25th Design Automation Conf.*, June 1988, pp. 536-541.
- [2] S. Devadas and R. Newton, "Algorithms for hardware allocation in data path synthesis," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 768-781, July 1989.
- [3] M. Pedram and B. Preas, "Accurate prediction of physical design characteristics for random logic," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1989, pp. 100-108.
- [4] S. Sastry and A. C. Parker, "Stochastic models for wireability analysis of gate arrays," *IEEE Trans. Computer-Aided Design*, vol. 5, pp. 52-65, Jan. 1986.
- [5] F. J. Kurdahi and A. C. Parker, "Technique for area estimation of VLSI layouts," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 81-92, Jan. 1989.
- [6] B. Landman and R. Russo, "On a pin versus block relationship for partitions of logic graphs," *IEEE Trans. Comput.*, vol. C-20, pp. 1469-1479, Dec. 1971.
- [7] C. V. Gura and J. A. Abraham, "Average interconnection length and interconnection distribution based on Rent's rule," in *Proc. 26th Design Automation Conf.*, June 1989, pp. 574-577.
- [8] X. Chen, M. L. Bushnell, "A module area estimator for VLSI layout," in *Proc. 25th Design Automation Conf.*, June 1988, pp. 54-59.

- [9] F. J. Kurdahi and Ch. Ramachandran, "Evaluating layout area tradeoffs for high level applications," *IEEE Trans. VLSI Syst.*, vol. 1, pp. 46–55, Mar. 1993.
- [10] C. Sechen, "Average interconnection length estimation for random and optimized placements," in *Proc. IEEE Int. Conf. Computer Aided-Design*, Nov. 1987, pp. 190–193.
- [11] A. C. H. Wu, V. Chaiyakul, and D. Gajski, "Layout-area models for high-level synthesis," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1991, pp. 34–37.
- [12] R. Camposano, R. A. Bergamaschi, C. E. Haynes, M. Payer, and S. M. Wu, *High Level VLSI Synthesis*. Norwell MA: Kluwer Academic, 1991, pp. 79–104.
- [13] D. E. Thomas, E. D. Lagnese, R. A. Walker, J. A. Nestor, J. V. Rajan, and R. L. Blackburn, *Algorithmic and Register-Transfer Level Synthesis: The System Architect's Workbench*. Norwell MA: Kluwer Academic, 1990.
- [14] A. Sharma and R. Jain, "Estimating architectural resources and performance for high-level synthesis applications," *IEEE Trans. VLSI Syst.*, vol. 1, pp. 175–190, June 1993.
- [15] —, "Register estimation from behavioral specifications," in *Proc. IEEE Int. Conf. Computer-Design*, Oct. 1994, pp. 576–580.
- [16] M. Nourani and C. Papachristou, "A layout estimation algorithm for RTL datapaths," in *Proc. 30th Design Automation Conf.*, June 1993, pp. 285–291.
- [17] M. Rim, A. Mujumdar, R. Jain, and R. De Leone, "Optimal and heuristic algorithms for solving the binding problem," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 211–225, June 1994.
- [18] S. Y. Ohm, F. J. Kurdahi, and N. Dutt, "Comprehensive lower bound estimation from behavioral descriptions," in *Proc. Int. Conf. Computer Aided-Design*, Nov. 1994, pp. 182–187.
- [19] J. Septien, D. Mozos, F. Tirado, R. Hermida, and M. Fernández, "Heuristics for branch and bound global allocation," in *Proc. EuroDac 92*, Sept. 1992, pp. 334–340.
- [20] High Level Synthesis Benchmarks, 1991.
- [21] R. Jain, A. C. Parker, and N. Park, "Predicting area-time tradeoffs for pipelined designs," in *Proc. 24th Design Automation Conf.*, June 1987, pp. 35–41.