Engineering Journal of Qatar University, Vol. 3, 1990.

"HIERARCHICAL FLOOR PLAN FOR VLSI INTEGRATED CIRCUITS BASED ON ANALYTICAL TECHNIQUES"

By

Ahmed L. Al-Mohannadi
Assistant Professor
Electrical Engineering Department,
Faculty of Engineering, Qatar University,
Doha, Qatar - Arabian Gulf.

ABSTRACT

The physical design of electronic circuits consists of transforming a design specification into a representation to be used in the manufacture of the physical circuits. At Very Large Scale Integration (VLSI) level it becomes diffcult to position or place all circuit elements on the layout surface and to interconnect them together in a one step process. Hierarchical approaches make the placement problem simpler and more easily solved. By partitioning the placement problem into several subproblems, the solution of each subproblem can be obtained and verified in short time.

The technique presented supports a design hierarchy whilst maintaining knowledge of global relations between the various components in the floor plan. The paper introduces a new technique where virtual I/O pins are placed on the periphery of each hierarchical block to represent the inter-block wiring. The technique allows for variable sized blocks so that the area requirement for any block is allowed to be adjusted during the floor plan optimisation process. It offers all of the benefit of a hierarchical approach whilst maintaining a global view of the wiring and geometrical requirements without the need to smash the hierarchy. A detailed derivation of the technique is given and its performance is demostrated through an example.

1. INTRODUCTION

Very Large Scale Integration (VLSI) circuit technology requires the use of hundreds or thousands of transistors on a single Integrated Circuit (IC) chip, making system layout difficult. The field of Computer-Aided-Design (CAD) or

Design Automation (DA) has evolved to assist digital designers in manipulating the large amount of data associated with complex designs. One level where CAD tools are involved is in floorplanning. It is the process of dividing the available silicon area into a number of typically rectagular partitions that best satisfy the area given, wiring connectivity, and performance constraints imposed by the design specification. The goal of the algorithm is to handle full-custom VLSI chip design by providing features that allow the user to specify the terminal position around the chip periphery, the aspect ratio of the layout area (eg. 1:1, 2:1, etc), the input/output port position, and the area allowed for the chip. From the given specifications, consideration is given to reducing the interconnection length between different functions and the overall chip area.

Several placement algorightms have been implemented but each imposes some restrictions on component sizes (eg. all points or same sizes) (Tsay 88) [1] which is not usually true in the field of VLSI designs. Another algorithm uses analytical techniques to solve the placement problem [3], [5]. This algorithm is based on the modified rectangle technique (Fig. 1) which states: for a given rectangular B with size $w_i.h_i(w_i > h_i)$ its corresponding modified rectangular is the area enclosed by all the points whose distances to a segment P_{i_1} (X_{i_1}, Y_{i_1}) P_{i_2} (X_{i_2}, Y_{i_2}) of length l_i and radius equal to r_i. The segment $P_{i_1} P_{i_2}$ is called the axis of the rectangle, and r_i is called the radius of the rectangle, where

$$L_i = W_i - h_i \quad r_i = \tfrac{1}{2} h_i$$

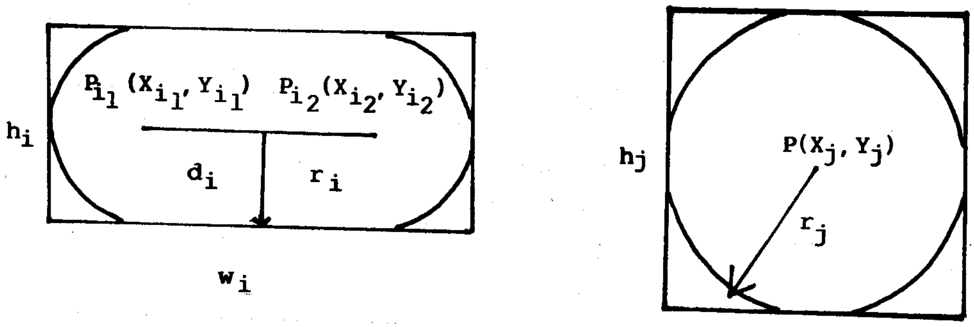


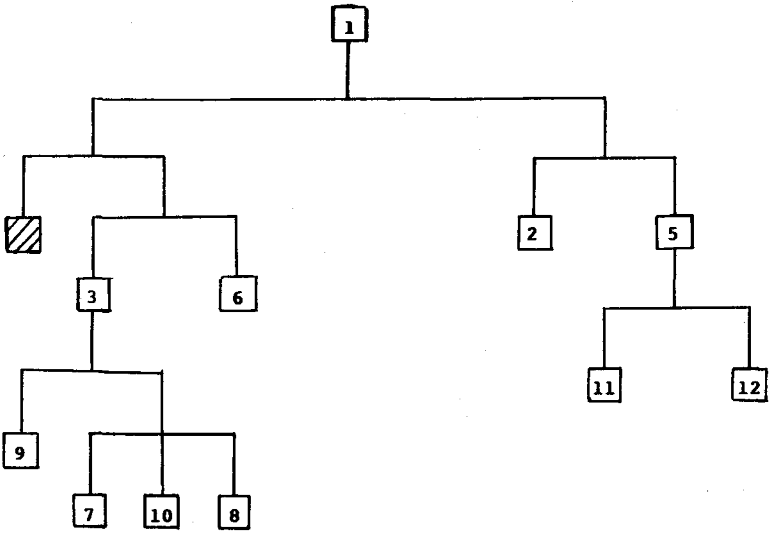
Fig. 1 : Modified Rectangle

If $W_i = h_i$ then $L_i = 0$ and the rectangle becomes a circle with radius r.

The model is feasible and continuous in term of mathematical behavior and can be solved easily. It gives good results. The model also supports a hierarchical floorplan in the case of a very large number of blocks.

A floorplan may consist of several levels (Fig. 2), and in each level there may be two types of modules. One level can be defined as a soft module or slice and the other as a hard module or block [2]. The slices represent circuit components, the topology of which is to be defined. Hard modules are those with dimensions available in the system library or modules whose layout has been specified by the user.

Due to the fact that all units are blocks in any floor plan and are connected physically by nets, two types of connectors can be used; fixed pads on the edges and floating pins or pinouts. The placement of the fixed pads around the bounding box at the highest level (chip) of the hierarchy is well defined since the user can initially specify the locations of input/output pads. However, there are two approaches to improving the choice of fixed pad locations. The first is by letting the pads move freely on any side of the chip. The second is by restricting the movement of pads to specific places on any of the four sides.



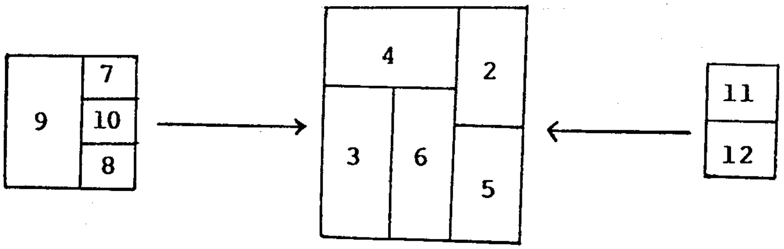


Fig 2 : Hierarchical Floor Plan

Pinouts are virtual objects added to the slice boundaries. They represent imaginary points where different levels in the hierarchy are interconnected and places where wires from external cells enter the slice. The pinout locations directly affect the length of global wiring in the floorplan, and it is very important that this length be minimized. The relations between pinout on different levels in the floor plan are very important and led to this research on the implementation of a pinout adjusted algorithm.

In this paper a new algorithm is introduced for the placement of several blocks on any one level (one placement problem), and for guaranteeing the preservation of the globality of wiring between different levels.

2. PINOUT ADJUSTMENT MODEL

The technique developed is an optimization procedure based on an objective function describing the wire length connecting the pinout within one level of the hierarchy to any other external level. A number of constraints were developed confining the pinouts to slice boundaries of any level of hierarchy. These constraints must be satisfied when minimizing the objective function.

3. THE OBJECTIVE FUNCTION

The technique described aims to minimize the wiring length described by an objective function. This objective function represents squared distances between the external modules or blocks and the contents of each module. This choice is made because it makes the function quadratic and continuous as discussed by

(Rao 78) [6]. If there are M slices in the design, N nets connecting each slices to the outside blocks or slices and L blocks connected by each net outside the slice, then the objective function may be expressed as follows:

$$F = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{L} (X_{p_{ij}} - X_k)^2 + (y_{i_{ij}} - Y_k)^2 \tag{1}$$

where,

$X_{p_{ij}}$, $Y_{p_{ij}}$ are the coordinates of pin p_{i_j} in slice i.

X_k, Y_k are the coordinates of the center of block k connected to the pinouts by net j.

4. THE BOUNDARY CONSTRAINTS

Pinouts are initially located inside the slice and should move freely within its boundaries. In order to prevent pinouts from leaving the slice area under the influence of forces exerted by the fixed blocks connected to them, a number of constraints must be imposed during the minimization procedure.

The constraints are developed based on the modified rectangle technique described by (Sha 85) [3], where any block is modified to be represented by a radius and segment in the case of a rectangle or just a radius when it is square. Three sets of constraints are required, one when the slice is square, one when the slice is rectangular but vertical, and one when the slice is retangular but horizontal.

The square case (Fig. 3a) has the following set of constraints;

$$\begin{aligned} G_1\,(i,j) &= x(i) - r(i) - X_p(j) \\ G_2\,(i,j) &= x_p(j) - x(i) - r(i) \\ G_3\,(i,j) &= y(i) - r(i) - y_p(j) \\ G_4\,(i,j) &= y_p(j) - y(i) - r(i) \end{aligned} \tag{2}$$

When the case is rectangular and vertical (Fig. 3b) the following constraints are required:

$$\begin{aligned}
G_1\ (i,j) &= x(i) - r(i) - x_p(j) \\
G_2\ (i,j) &= x_p(j) - x(i) - r(i) \\
G_3\ (i,j) &= y(i) - \frac{\text{length (i)}}{2} - r(i) - y_p(j) \\
G_4\ (i,j) &= y_p\ (j) - (y(i) + \frac{\text{length (i)}}{2} + r(i))
\end{aligned} \tag{3}$$

When the case is rectangular and horizontal (Fig. 3c) the following constraints are required:

$$\begin{aligned}
G_1\ (i,j) &= x_p(j) - (x(i) + \frac{\text{length (i)}}{2} + r(i) \\
G_2\ (i,j) &= x(i) - \frac{\text{length (i)}}{2} - r(i)) - x_p(j) \\
G_3\ (i,j) &= y(i) - r(i) - y_p(j) \\
G_4\ (i,j) &= y_p(j) - y(i) + r(i))
\end{aligned} \tag{4}$$

For all the above cases:

i= 1,2, ..., M slices and j = j,2, ..., N nets
x(i) , y(i) are the coordinates of the center of slice i
$x_p(j)$, $y_p(j)$ are the coordinates of pin j
r(i) is the radius of slice i
length (i) is the length of slice i

All the above constrainsts are in the form of

$$G_p\ (i,j) \leqslant 0$$

where, p = 1,2,3,4

5. SOLUTION TECHNIQUES

The objective function described in Equation (1) will be minimized using the penalty or indirect optimization which involves adding the boundary constraints as penalty terms to the function (RAO 78) [6]. These penalty terms should be reduced to zero in the final solution. The resulting unconstrained objective function can be written as follows:

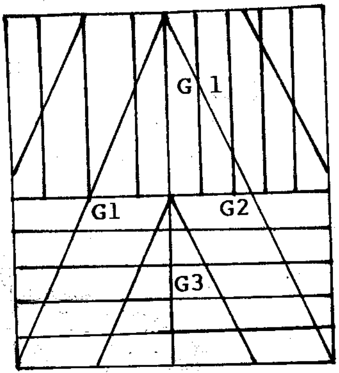
$$W(x,y,r_k) - \sum_{i=l}^{M} \sum_{j=l}^{N} \sum_{k=l}^{L} (x-x)^2 + (y-y)^2 + r_k$$

$$\sum_{i=l}^{m} \sum_{j=l}^{n} \sum_{k=l}^{4} [MAX (O,G_k (I,J))]^2$$

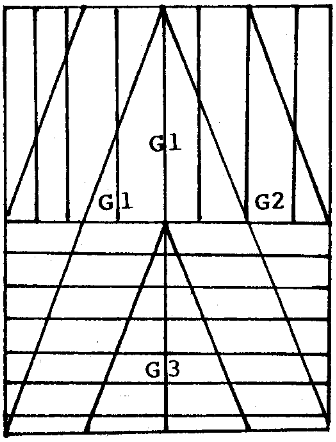
m = number of slice
n = number of pins

Where r_k is an incremental penalty factor and tends to ∞.

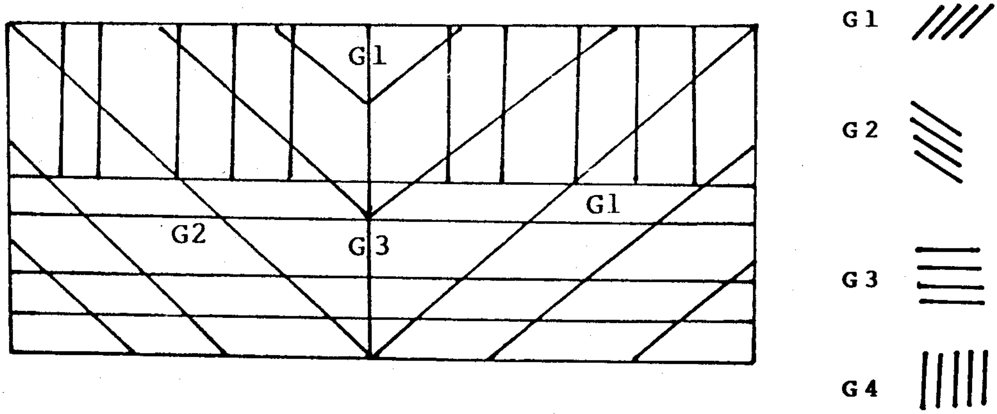
An example run is used to verify the validity of constraints (Fig. 4,5,6). There are four pinouts to external blocks.



(a) Square



(b) Vertical



(c) Horizontal

Fig. 3 : Constraints Mapping for different Slices.

Net number 1 is connected to block number 3, net 2 is connected to block number 2, net number 3 is connected to block number 4 and net number 4 is connected to block 5. The example is used for three cases; the first one is when the slice is square. The final locations for the pinouts (pin 1, pin 2, pin 3 and pin 4) correspond to the shortest distance between each pinout and the particular external block it is connected to. The second is when the slice is vertical. The final positions of the pinouts correspond to the minimum total wire length are as shown in Fig. 5. Finally, the third is when the slice is horizontal. The developed constraints for the horizontal case are used and prove its correctness (Fig. 6). Table 1 shows the initial location and final location of the pinouts for the three cases.

Table 1 : Initial and Final Location of Pinouts.

CASE SQUARE		
Pin No.	Initial Location	Final Location
1	7.96,11	8,8
2	12.95,50	8,5
3	3.04,20	4,4
4	3,11.92	4,8
CASE VERTICAL		
Pin No.	Initial Location	Final Location
1	7.98.11	8,8
2	12.95,50	8,5
3	3.04,20	6,3
4	3,11.91	6,8
CASE HORIZONTAL		
Pin No.	Initial Location	Final Location
1	7,95.11	8,7
2	13,5.0198	10,5
3	3.069,20	5,5
4	3.019,11.93	5,7

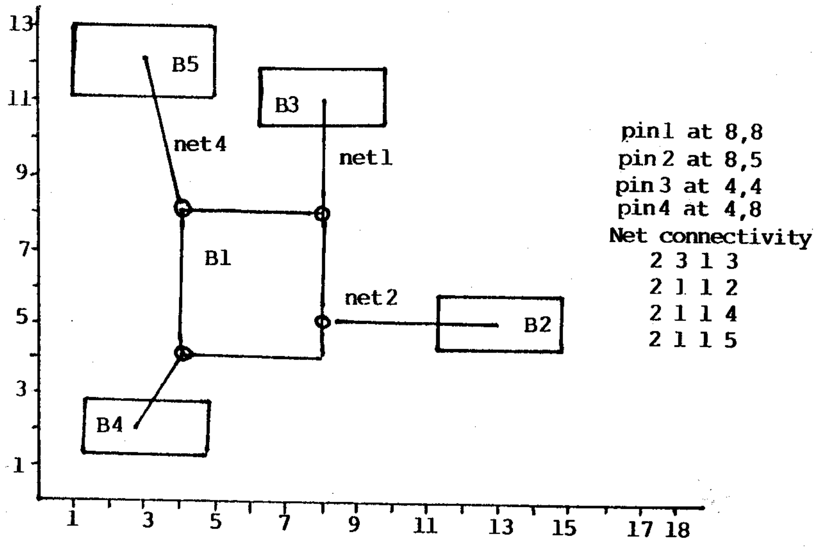


Fig. 4 : Final Locations for Pinouts (Square)

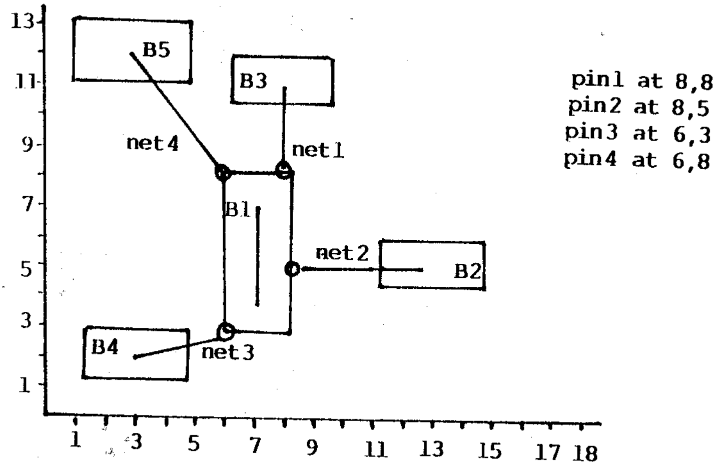


Fig. 5 : Final Locations for Pinouts (Vertical)

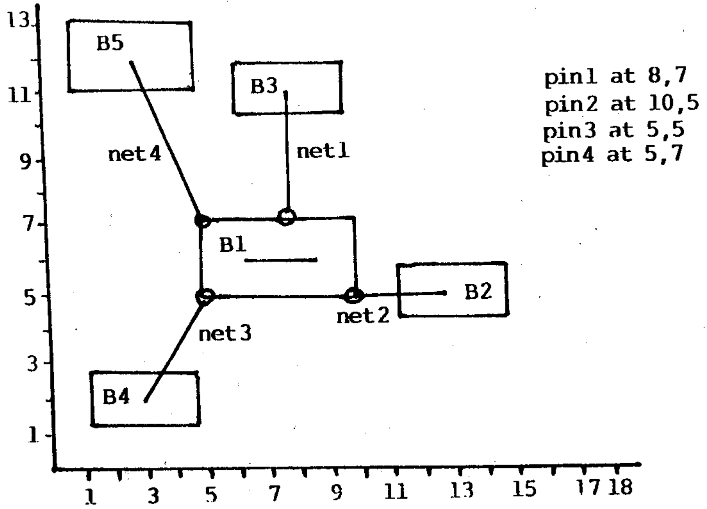


Fig. 6 : Final Locations for Pinouts (Horizontal)

6. TWO LEVEL HIERARCHY

During placement calculations which are carried out at different levels, the location of blocks and slices depends on the connectivities between them. Since the floor plan is one entity, which means physically on one large piece of silicon, there must be a global relation between different levels on the hierarchy. This globality can be preserved by pinout adjustment routine.

The following example will demonstrate the effectiveness of reducing the wiring length in the floor plan. It consists of two slices (variable size and shape). The content of each slice is shown in (Fig. 7). The connectivity is also shown between the content of each slice and also the connectivity to other slices in the floor plan. The program developed considers a random start where sizes and location of blocks within slices and location of slices and other blocks within the floorplan are not defined. Initially the program will consider the slices and the

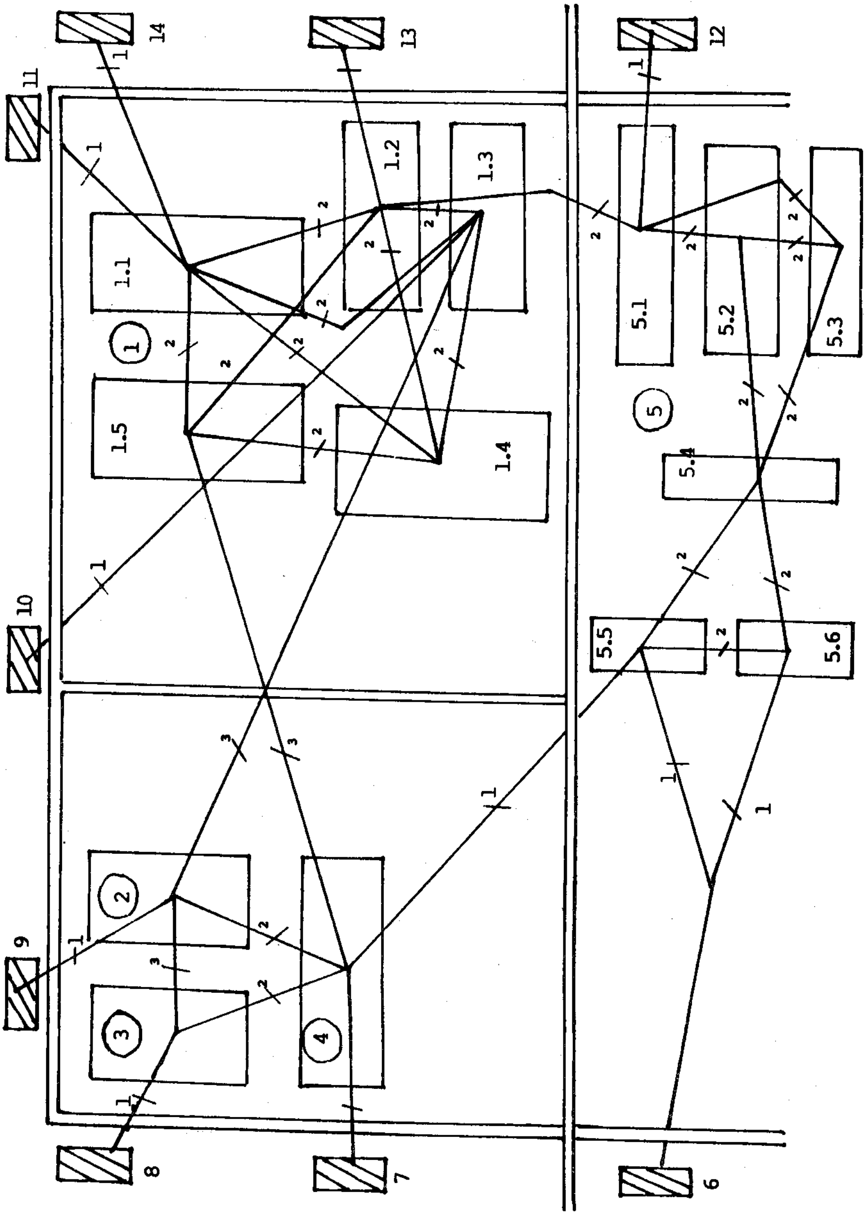
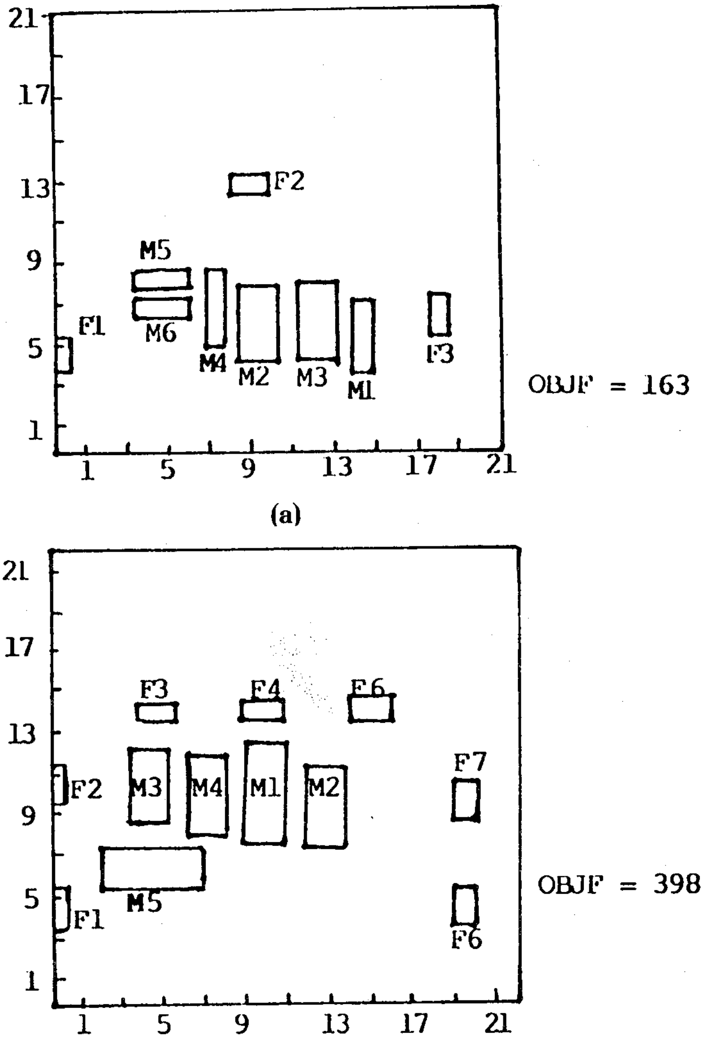


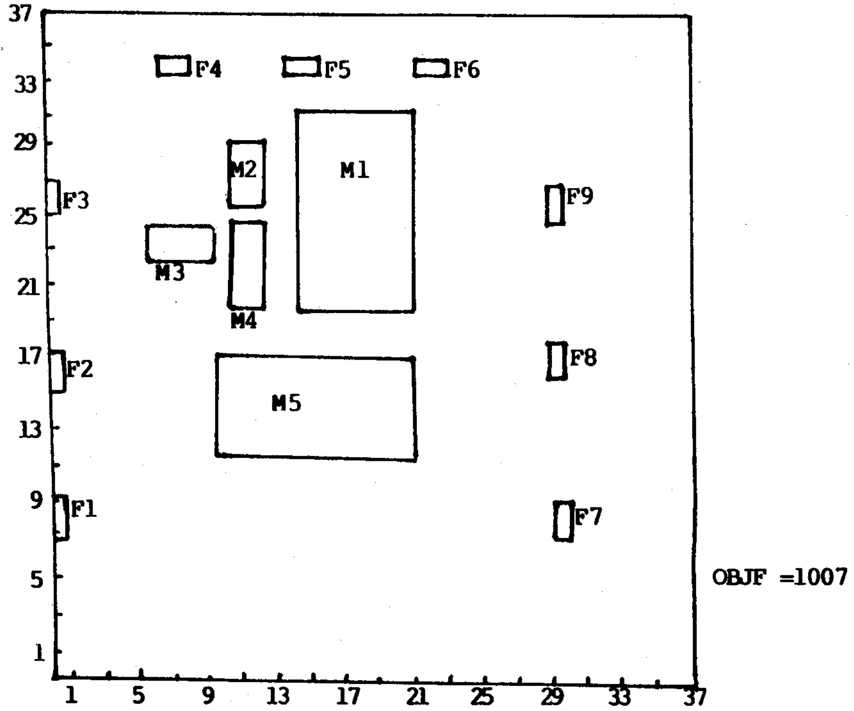
Fig. 7 : Hierarchical Example.

pins around them. It will place the content of each slice based on the connectivity given with the initial location of pinouts. Fig. 8 and Fig. 9 shows the placement program results. Initially the program places the content within each slice based on the connectivity given (Fig. 8a,b), then the floor plan should be placed based on the sizes obtained for each slice. The placement program is used again for the top level and the result is shown in (Fig. 8c).



(a)

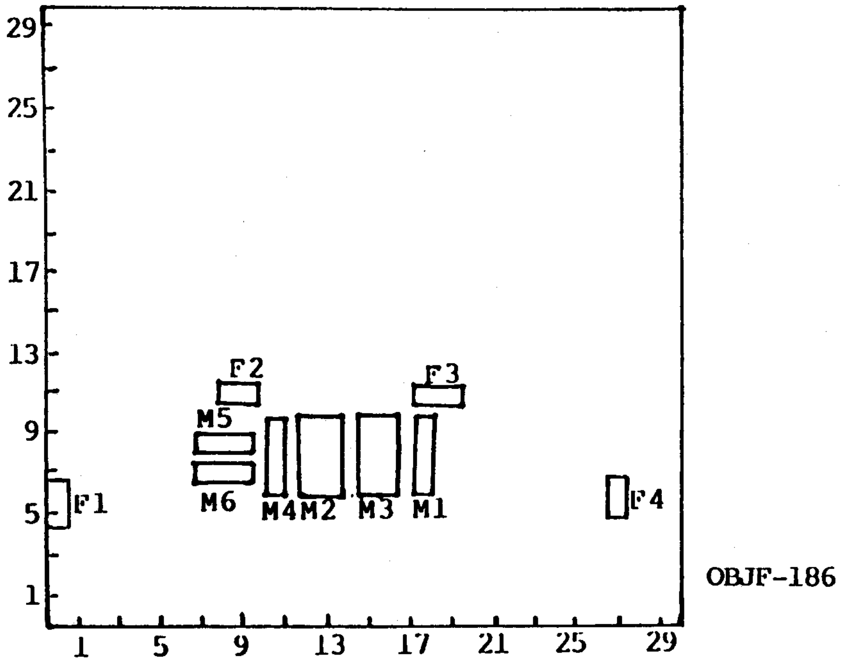
(b)



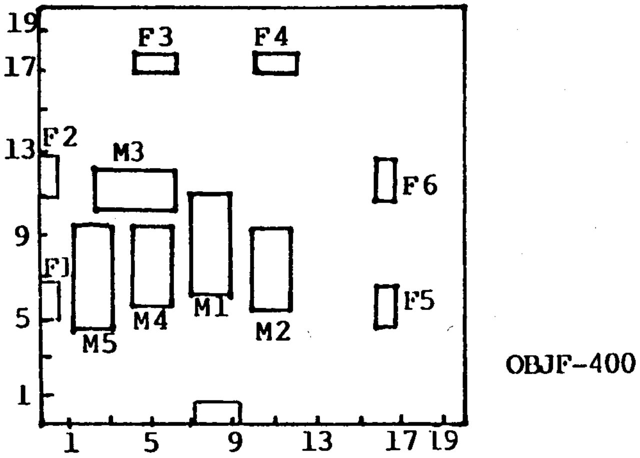
(c)

Fig. 8 : Initial Placement for Different Levels

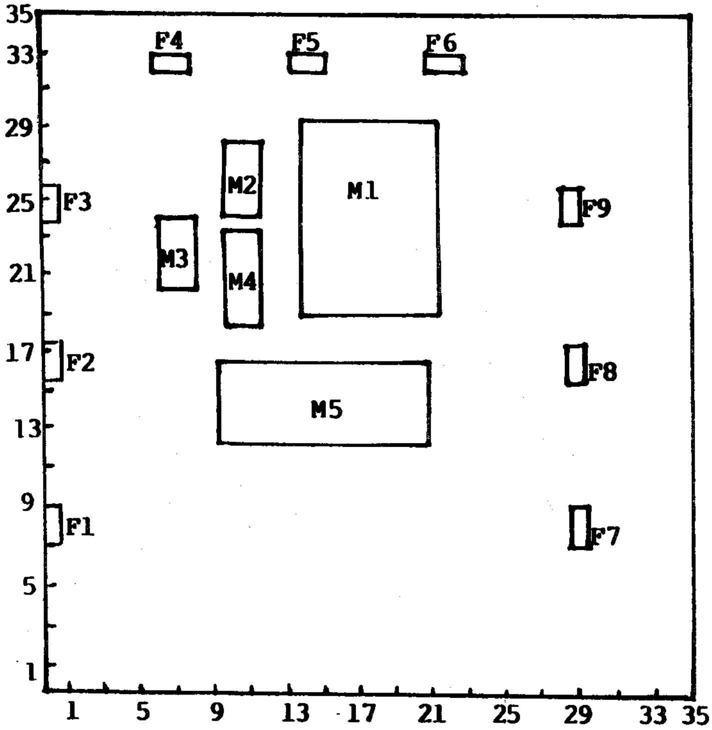
Since the size of each slice is based on the initial pinout locations, now pinout must be adjusted based on the result or the location of the slices. The pinout procedure must be used to adjust pinouts around each slice. Results are given in (Fig. 9a,b), where pinouts are adjusted. The placement program is used again to place the content of each slice based on the new adjustment of pinouts. New sizes of the slices are used to run the placement program for the top floor plan and the result is shown in (Fig. 9c).


29
25
21
17
13
9
5
1
1
5
9
13
17
21
25
29
F2
F3
M5
M6
M4
M2
M3
M1
F1
F4
OBJF-186

(a)


19
17
13
9
5
1
1
5
9
13
17
19
F3
F4
F2
M3
F6
F1
M5
M4
M1
M2
F5
OBJF-400

(b)



OBJF–944

(c)

Fig. 9 : Final Placement for Different Levels

7. CONCLUSION

The algorithm developed for pinout adjustment shows promising results based on a new objective function that takes into account the global wiring and relation between different hierarchy levels. A comparison of our algorithm with the algorithm described in (Yau 88) [4], shows that the modified rectangle in these cases is a circle which actually does not correspond to the actual area, while the algorithm used modifies the rectangle to be a modified rectangle with smooth corners. The location of pinouts around the block in our case is acceptable while in their case they should use another procedure to locate pinouts.

A further study involving an optimization model for IC design is going to be implemented at the microelectronic lab of the Electrical Engineering Department this year. The aim is to design chips and produce layout which can then be fabricated at International Fabrication Centers and tested at Qatar University. Such design can be useful to train students and give them some knowledge about high technologies in electronics.

REFERENCES

1. **Tsay, R. and Kuh, E.,** "Module Placement for Large Chips based on Sparse Linear Equations".

2. **Chaisemartin, P.** "FADDADZ A New Interactive Floorplan Design System". INT Integrated Circuit Technology Conference - Paper - Limerick, pp 361-372 (1986).

3. **Sha, L. and Dulton, R.W.,** "An Analytical Algorithms for Placement of Arbitrary Sized Rectangular Blocks". 22nd IEEE Design Automation Conference, pp 602-608 (1985).

4. **Yao, Xianjin, Yamada, Massaki and Lin, C.L.,** A New Approach to the Pin Assignment Problem. 25th ACM/IEEE Design Automation Conference pp. 566-572, (1988).

5. **Ahmed Al Mohannadi,** "Application of Analytical Placement Techniques for Floorplanning in VLSI Design". Ph.D Thesis submitted to the Faculty of Electronic Engineering at The University of Kent, U.K., (1989).

6. **Rao, S.S.** "Optimization Theory and Application" New Delhi, Wiley Eastern (1978).