Review Paper on Placement Algorithms

Mitali
Academic & Consultancy Services Division
C-DAC, Mohali, India
mitali.garg6@gmail.com

Deepak Bhati
Academic & Consultancy Services Division
C-DAC, Mohali, India
deepakbhati35@gmail.com

ABSTRACT- Placement is a step considered after floor planning in FPGA flow in ASIC. It defines the location of logic cells within functional blocks and to minimize the routing length. We adjust the logic cells in this way so that minimum interconnect length, area and density are used. In this paper we will discuss some of methods of placement of logic cells.

Keywords- Placement Algorithms, FPGA.

1. INTRODUCTION

Some of the architectures have rows of logic cells that's why it is called row based ASICs. In normal ASICs the interconnects run in both horizontal and vertical directions. The interconnects in horizontal direction are running parallel to the channel spine and in vertical direction it goes by crossing the cells which is called over the cell routing, but the routing in vertical direction is in such a way that the cells can't block. Both horizontal and vertical layers are of different metals to distinguish the layers.

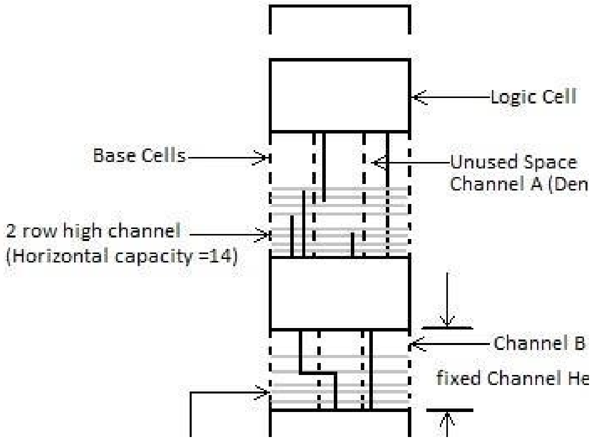


Fig1: Channel Routing showing channel density

1.1 Some Common Use Terms and Definitions

1.1.1 ***Channel capacity:*** The total number of interconnects that are passing through any of logic cell is called channel capacity.

1.1.2 ***Feed through:*** Vertical interconnects uses some of strips to cross the logic cells are called feed through.

1.1.3 ***Uncommitted feed through:*** The vertical track which are not used in a logic cell are called uncommitted feed through.

1.1.4 ***Jumper:*** A vertical track that has no connections inside the logic cell are called jumpers.

1.1.5 ***Spacer cells:*** The cells that are used to fill the spaces in a logic cell is called spacer cell.

2. PLACEMENT ALGORITHMS:

Placement algorithms have two classes:

1. Constructive placement algorithm
2. Iterative placement algorithm

(a) The most commonly used methods in constructive algorithm are:

1. Eigen value method
2. Min cut algorithm

In min cut algorithm there are following steps which are performed:

1. Firstly we are going to cut the placement area into different pieces.
2. Then swapping of logic cells is there.
3. Again repeating the above steps.

The logic cells after dividing the placement area are called "Bins".

Volume 4, Issue 2, Pages 49-52, June 2016, ISSN: 2347-470X

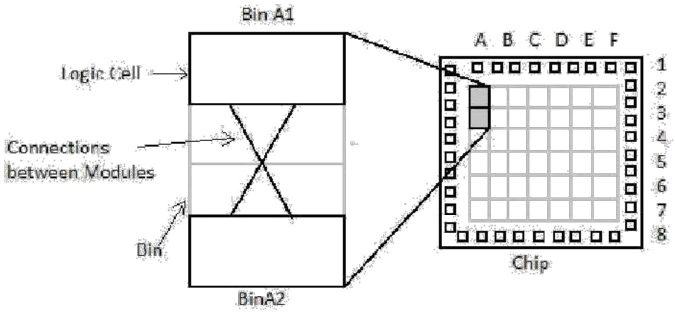


Fig 2: (a) Divide the chip into bins using a grid

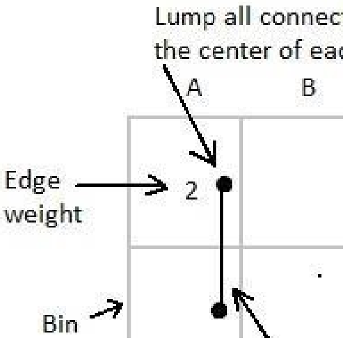


Fig 2: (b) Merge all connections to the center of each bin

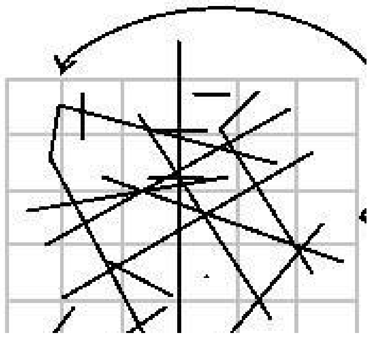


Fig 2: (c) Make a cut and swap logic cells between bins to minimize the cost of the cut

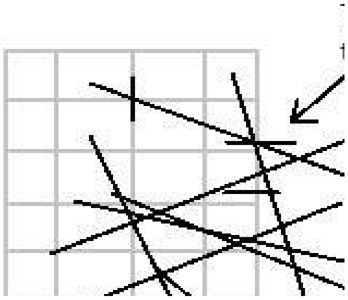


Fig 2: (d) Take the cut pieces and throw out all the edges that are not inside the piece

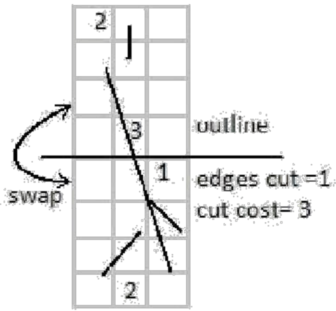


Fig 2: (e) Repeat the process with a new cut and continue until we reach the individual bins

Eigen value placement algorithm

In Eigen value placement algorithm we use connectivity matrix and cost weighted matrix.

The cost function is denoted by f and is given by:

1 n

$f = S\ c_{ij}\ d_{ij}$^2 $2_i = 1$

Where c_{ij} is connectivity matrix.

d_{ij}= Euclidean distance between the centers of logic cell I and logic cell j.

In Eigen value consider the following equations in which c is connectivity matrix and b is disconnection matrix

B= d-c

c=0001
0011
0100
1100

$$B = \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{array} \quad - \quad \begin{array}{cccc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{array} \quad = \quad \begin{array}{cccc} 1 & 0 & 0 & -1 \\ 0 & 2 & -1 & -1 \\ 0 & -1 & 1 & 0 \\ -1 & -1 & 0 & 2 \end{array}$$

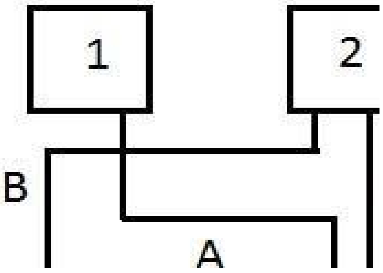


Fig 3: An example network

Iterative placement algorithm

In this algorithm we always take an existing placement logic cells and then we improve it by moving the logic cells.

There are mainly two parts of this algorithm:

1 .we are going to select which logic cell we have to move.
2. Whether the logic cell selected should be moved or not it will depend upon the gain.

Several interchange methods are there which are differing in then selection criteria:

a) Pair wise interchange
b) Force directed interchange
c) Force directed relaxation
d) Force directed pair wise relaxation

In pair wise interchange following steps are there to perform:

1. Firstly take any of the logic cell.
2. Then select any of destination for that logic cell to place so that we can improve the gain.
3. Where the gain will be maximum that destination is our final destination to move that logic cell.

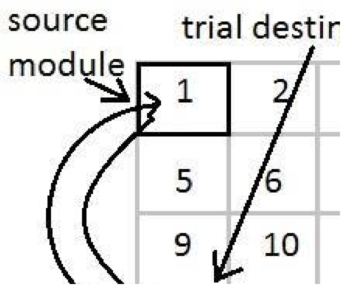


Fig 4: (a) changing the position of source logic cell with destination logic cell with pair wise interchange

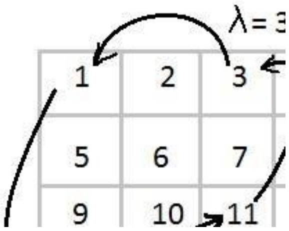


Fig 4: (b) Sometimes we have to swap more than two logic cells at a time to reach an optimum placement, but this is expensive in Computation time. Limiting the search to neighborhoods reduces the search time. Logic cells within a distance of a logic cell form an e-neighborhood

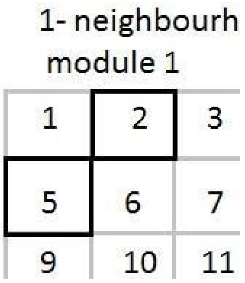


Fig 4: (c) one neighborhood logic cell

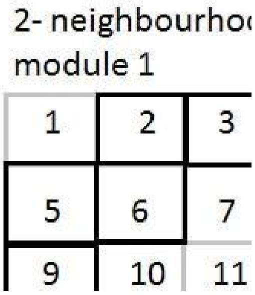


Fig 4: (d) a two neighborhood cell

Force directed placement: Neighborhood logic cells are also used in force directed placement algorithm in which we take identical springs which are connecting all the logic cells together. The number of springs is equal to no. of connections to all the logic cells. The more highly the logic cells are connected, stronger the pull of strings .It is actually based on the HOOK'S law in which

$$F_{ij} = - c_{ij}$$

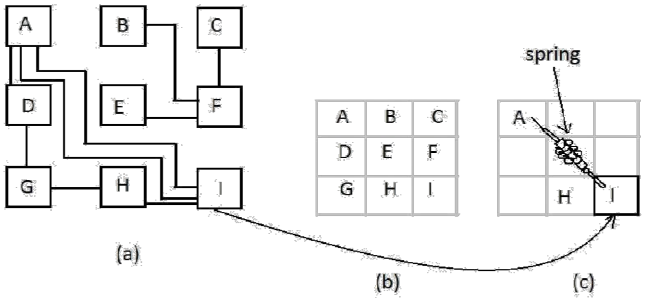


Fig 5: (a) A network with nine logic cells
(b) We make a grid (one logic cell per bin)
(c) Forces are calculated as if springs were attached to the centers of each logic cell for each connection. The two nets connecting logic cells A and I correspond to two springs

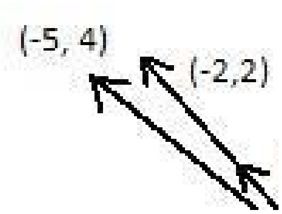
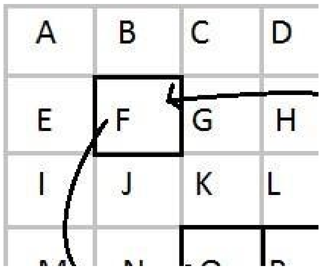


Fig 5: (d) The forces are proportional to the spring extensions

Force directed pair wise relaxation: In this algorithm it swaps one pair of logic cells at a time.



Placement using stimulated annealing: The stimulated annealing algorithm is least preferred because it has so many iterations and is not fast to calculate. In it we are using half perimeter measure for minimizing the length of the interconnect.

There are following steps for stimulated annealing algorithm to perform:
1. Firstly select the logic cells randomly to interchange.
2. Calculate the energy function E for the new placement.
3. The value of E should be negative or zero, if E is negative or zero, then exchange the logic cells.
4. If E is positive then there is probability of exp(-E/T) to exchange the logic cells.
5. Again repeat the steps fixed no. of times.

3. CONCLUSION

The criterion for optimization may be minimum interconnect area, minimum total interconnect length, or performance. There are two main types of placement algorithms: based on min-cut or eigenvector methods. Because interconnect delay in a submicron CMOS process dominates logic-cell delay, planning of interconnect will become more and more important. Instead of completing synthesis before starting floor planning and placement, we will have to use synthesis and floor planning/placement tools together to achieve an accurate estimate of timing.

REFERENCES

[1] Q. Ma, E. F. Y. Young, and K. P. Pun, "Analog placement with common centroid constraints," Proc. ICCAD, 2007, pp. 579-585.

[2] Min Pan, Chris Chu "IPR: An Integrated Placement and Routing Algorithm", Proc. Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE, San Diego, CA.

[3] Application-specific-integrated-circuits-MJ-smith.

[4] Hanan, M., P. K. Wolff Sr., and B. J. Agule. 1973. _Some experimental results on placement techniques._ In Proceedings of the 13th Design Automation Conference. Reference to complete graph wire measure.

[5] Hartoog, M. R., 1986. _Analysis of placement procedures for VLSI standard cell layout._ In Proceedings of the 23rd Design Automation Conference.