INTERNATIONAL JOURNAL OF RECENT TRENDS IN ENGINEERING & RESEARCH
Impact Factor 4.101 ISSN -2455-1457
RECENT ADVANCES IN ELECTRONICS & COMMUNICATION ENGINEERING (RAECE-2K17)

AN OPTIMAL SOLUTION FOR FLOOR-PLANNING AND MINIMIZATION OF CONNECTIVITY

G. UMADEVI
Assistant Professor, ECE
NNRG, Hyd-500088

P. ABHISHEK
(M. Tech) VLSI &ES
NNRG, Hyd-500088

B. VENKATESHWARA RAO
(M. Tech) VLSI &ES
NNRG, Hyd-500088

***Abstract*-The Minimization of the Connectivity plays the important Role in the Physical Design automation of very large scale integration (VLSI) chips. The Connectivity Minimization can be achieved by finding the optimal solution for VLSI physical design components like Partitioning and floor-planning. In VLSI Problem of obtaining minimum delay in partitioning has prime importance. Reducing the minimum delay in partitioning and the area in the floor plan helps to minimize the connectivity. And performing this we are using different algorithms like Interative-improvement techniques and Rectangular dual graph based are used in Partitioning and floor-plan.**

I. INTRODUCTION

Partitioning and floor-planning (PF) has become an active area of research for at least a quarter of a century. The main reason that partitioning has become an central and critical design task today is due to the enormous increase of system complexity in the past and the expected further advances of microelectronic system design and fabrication. Synthesis and simulation tools often cannot go with the complexity of the entire system under development, and also designers want to concentrate on critical parts of a system to speed up the design cycle. Thus the present state of design technology often requires a partitioning of the system For speed and effective optimization. So the fabrication technology requires partitioning of a system into components by arranging the blocks without wasting free space. The direct implementation of large circuit will occupy more area. Hence the large circuit is to be split into small sub circuit. This will minimize the area of the system and the complexity of the system. When they are partitioned, the connection between two modules should be minimum (or the number net cut by the partition). This is known as cut size and hence this plays a major role in partitioning.

The process for determining blocks shapes and positions with area minimization objective is referred as floor-planning. A common method for blocks floor-planning is to determine in the first phase and then the relative location of the blocks to each other based on Fly lines criteria. And the second step, block sizing is performed with the help of minimizing the overall chip area and the location of each block is finalized. When the partitioning and floor-planning are both come in to picture (PF), the criteria like power, cost, and clock speed of each module are the sub objective to be optimized.

In the early stages, many interchanging methods have been used which resulted in local optimum solution. And later some of the mathematical methods are followed. Some heuristics are also used which resulted in better result but it has its own advantages and disadvantages. Since there may be many solutions possible for this problem, stochastic optimization techniques are utilized and until now many techniques have been known like Simulated Annealing Algorithm (SA) which combines the Local Search Algorithm with the Metropolis algorithm.

Simulated Annealing is a simple algorithm and does not need much memory, but it takes a long time to reach the desired solution. Kernighan and Lin proposed a two-way graph partitioning algorithm which was become the basis for most of the subsequent partitioning algorithms. Fiduccia and Mattheyses modified the K-L algorithm to a more efficient algorithm by suggesting moving one cell at a time and by designing a new data structure separately. As a kind of global optimization technique Genetic Algorithm (GA) which borrows the concept of generation from biological system had been used for physical design problems like circuit partitioning, floor-planning, and so forth. This technique has been applied to several problems, most of which are graph related because the genetic metaphor can be most easily applied to these types of problems. GA requires more memory but it takes less time than Simulated Annealing.

II. PARTITIONING

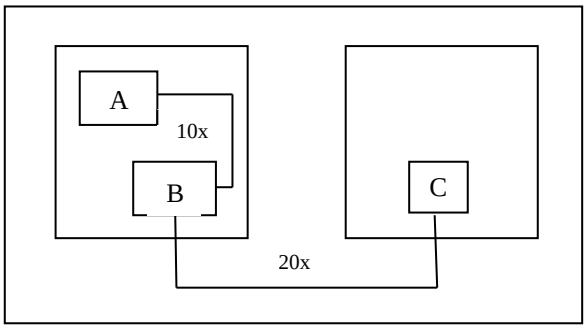
A better circuit partitioning will reduce connections among sub-circuits as the result better routing area for the layout. The Partitioning can be done in different levels such as

1. System level
2. Board level
3. Chip level

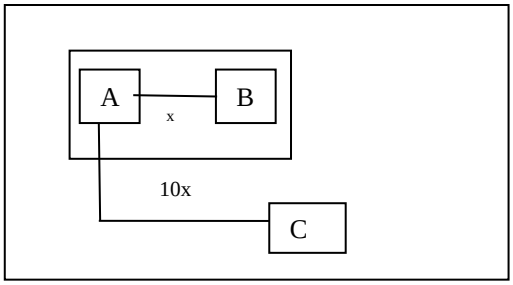
But the delay implications are different at each and every level. Which means that when we are connecting with the Intra chip (we are connecting two points inside the chip) The Delay may be for example X. And for the Intra Board (we are connecting two chips in a single board) The Delay may be for example 10X. And for the Inter Board (we are going across the board) The Delay may be for example 20X.

A. Problem Formulation

Partitioning a given net-list in to smaller net-lists such that Interconnection between Partitions are minimized. And Delay due to the Partitioning is also minimized. And the number of the terminals is less than a predetermined maximum value. The area of each partition remains with IN specified bounds. And the number of partitions also remains within specified bounds. They are different techniques are used for the partitioning. They are constructive and the Interative-improvement.



Total delay = 10x+20x = 30x



TOTAL DELAY = X+10X = 11X

B. Constructive Techniques

In constructive technique we will take the small partition and we will construct in to the bigger one. They are three types of the constructive techniques are available

1. Random Selections
2. Cluster Growth
3. Hierarchical clustering

Random selection

In random selection we will select the nodes randomly one at a time and place in to clusters of a fixed size until the proper size is reached. While doing the random selection we will not see any parameters such as size, quality.

Cluster Growth

In the cluster growth selection starting with a single node we are going to add the other nodes to form the partitions based on the connectivity. Number of clusters used as an input parameter.

Hierarchical Clustering

In the Hierarchical clustering we are going to the group the set of the objects depending on some measures and the closeness. This means that two closest objects are clustered first and consider being a single object further partitioning. By doing this we can make the two or more clusters. In this we cut the tree in to two trees and make in to the cluster when they are heavily connected.

C. Iterative-Improvement techniques

In this technique we are having the several partitions to start we will start by taking one partition we will improve the quality of algorithm by blocks.

Min-cut Algorithm

The min-cut Algorithm is also called as the Kernighan-lin algorithm. The main aim of this algorithm is that when we cut the cluster in to the two partitions we must see the signals passing to the partitions should be minimize. Here input graph is partitioned in to two subsets of equal sizes. Till cutsets keep improving vertex pairs which give the largest decrease in cut size are exchanged. Then the vertices are the locked. If no improvement is possible and the some vertices are still unlocked then the vertices which give the smallest increase are exchanged.

III. FLOOR-PLANNING

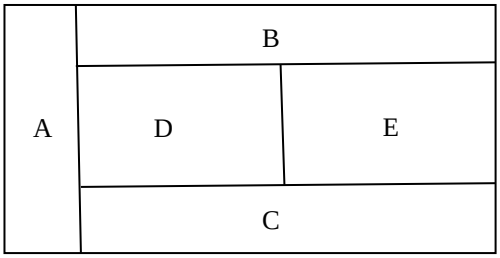
To design a chip we must place Memories nothing but macros and the standard cells should be placed on the chip this is done by the floor-planning. Normally a chip consist of the core and the die. Inside the core we are going to place the logic and the core area is surrounded by the die area here die area is use to protect the core area. And the die which consist of core which is small semiconductor material specimen on which fundamental circuit is fabricated. If the logic cells cover totally which means that core is 100 percent utilization.

$$\text{Utilization factor} = \frac{\text{area occupied by the net-list}}{\text{Total area of the core}}$$

When it is 100 percent utilization we cant place any logic inside the core. Whenever Aspect ratio is one which means that the chip is square shape. If the Aspect ratio is less than one which means the chip is the Rectangle shape which means chip is totally not utilized it is utilized about 60 to 70 percent that means we can place standard cells.

To avoid the fluctuations between the blocks we are placing the die around the core because to avoid the DRC and the shorts between the blocks. And must not keep the input and output ports at corners because there will be conjunctions between the ports. And between two input and output ports

there must be 2x spacing must be given to avoid the conjunctions.



A. Macros

Macros are nothing but memories. And the macros are power functionality and target some area. And the macros should be placed at the corners of the core. And the power pins are used inside the macros. They are two types of macros are present they are

1. Soft macros
2. Hard macro

Soft macro

Soft macro uses the RTL and the synthesis codes and changes the macros what we require in the design. And they are used in the design for the proper functionality.

Hard macro

In the hard macro we cant modify anything before the macro is given they will harden the macro so that we cant modify the macro.

B. Macro placement

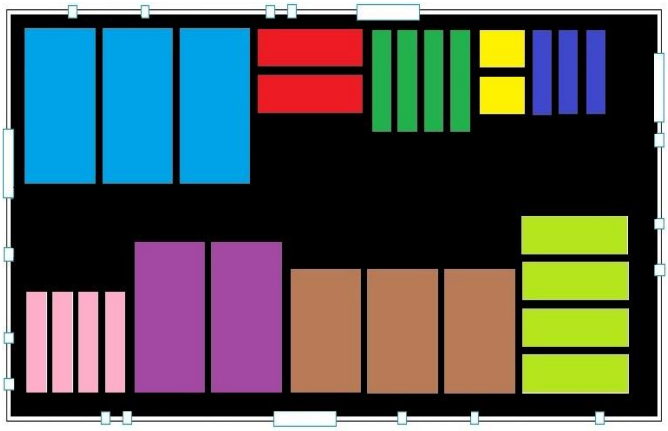


Figure.1 Macro placement

When we are going to do the macro placement on the chip see that the macros are placed at the corners of the core boundary. While placing the macros we must see the fly-lines between the macros. By seeing the fly-lines between the macros we must connect the macros we must see fly-lines should be going in straight way there must be not zigzag in fly-lines. And we must see that macro pins should be at small distance and the pins must be face each other. So that the delay can be reduced in over all chip. We are going to group macros in to modules. Macros should be never placed at the middle or the center because it will mainly effects the power because from top power must travel till middle because of that we are placing macros at the sides. When we place macro at middle or center then they are timing critical only. Place macros where less number of input ports and output ports and no input port and output ports are present. If you have no option then place macros near I/O ports provide enough routing space.

C. Estimating cost of a floor-plan

The number of the feasible solutions for the floor-plan is very large. Several criteria used to measure the quality of the floor-plans.

a. Minimize the area
b. Minimize total length of the wire
c. Maximum routing
d. Minimize the delay
e. Any combination above

We can find the area by reducing the block size. And by taking the width and height of the block. We can find the area. And to determine the wire length. There will be manhattan distance will present on the wire length. The space which is present after the floor-plan is known as the dead space or unoccupied space.

D. Rectangular dual graph based algorithm

In this algorithm the output of the partitioning is represented by the graph. The floor-plan obtained by converting the graph in to the rectangular dual. Here graph is converted in the floor-plan it is known has the rectangular dual. Without loss of generality we assume that a rectangular floor-plan contains no cross junctions. Under this assumption the dual graph of a rectangular floor-plan is a planar triangulated graph. Every dual graph without the cross junctions is a planar triangulated graph.

However not every planar triangulated graph is not a rectangular floor-plan.

E. Integer linear programming formation algorithm

In this algorithm we use linear equation 0/1 integer variables This algorithm used to obtain optimal solution based on defined cost function. This algorithm used for small problem instance only because of high computational complexity

REFERENCES

[1] I. H. Shanavas and R. K. Gnanamurthy, "Wirelength minimization in partitioning and floor-planning using evolutionary algorithms," VLSI Design, vol. 2011, Article ID 896241, 9 pages, 2011.

[2] M. J. S. Smith, Application Specific Integrated Circuits, Pearson Education, Asia, 2004.

[3] W. Theodore Manikas and T. James Cain, "Genetic algorithms vs simulated annealing: a comparison of approaches for solving the circuit partitioning problem," Tech. Rep. 96-101,The University of Pittsburgh, 1996, http://www.ee.utulsa.edu/~tmanikas / Pubs/gasa-TR-96-101.pdf.

[4] G. Ch. Sipakoulis, I. Karafyllidis, and A. Thanailkis, "Genetic partition and placement for VLSI circuits," in Proceedings of 6th IEEE conference on Electronics Circuits and Systems, vol. 3, pp. 1647–1650, 1999.

[5] M. R. Garey and Johnson, Computers and Intractability: A Guide to Theory of NP-Completeness, Freeman, San Francisco, Calif, USA.

[6] C. J. Augeri and H. H. Ali, "New graph-based algorithms for partitioning VLSI circuits," in Proceedings of the IEEE International Symposium on Circuits and Systems, vol. 6, pp. V521–V-524, May 2004.

