DISPLAY PACKAGE FOR GENERAL FLOORPLAN AND GLOBAL ROUTING IN STAIR-CASE CHANNEL

A dissertation submitted in partial fulfilment of
the requirements for M.Tech(Computer Science)
of the
Indian Statistical Institute

By

JOYDEB DAS BISWAS

Under The Supervision of

Dr(Mrs). Susmita Sur Kolay
Electronics Unit
INDIAN STATISTICAL INSTITUTE
203,Barrackpore Road,
Calcutta- 700035

31 JUlY,1997

Certificate of Approval

This is to certify that the thesis titled Display Package for General Floorplan and Global Routing in Stair-case Channel submitted by Mr. Joydeb Das Biswas , towards partial fulfilment of the requirements for the degree of M.Tech in Computer Science at the Indian Statistical Institute, Calcutta, embodies the work done under my supervision.

Susmita Sur-Kolay

Dr(Mrs). Susmita Sur Kolay.
Electronices Unit
Indian Statistical Institute
Calcutta -700035.

Acknowledgements

I am ever greateful to my supervisor Dr. (Mrs.) S.SUR.KOLAY for her helpful guidence , constant support and inspirations during this work. She has been with me in each and every part of this work.

Apart from , I am also greateful to Dr. P.S.DASGUPTA of IIM , calcutta , from him I get my 1st lessons on VLSI Design and in subsequent time he helped me for giving suggestions and constant support.

I would also thank to Dr. S.C.NANDY for his valuable advice and constant inspiartions for preparing this report. I am also indebted to Prof B.B.BHATTACHARYA for his valuable suggestions.

I would also like to thank to my classmates Mr. Nirmalya Chatterjee , Mr. Sandip Dutta , Miss. Rekha Menon , Miss. Padma Mahalingam and Mr. Gautam Das for helping me to prepare the final phases of this report.

Joydeb Das Biswas

July 31 , 1997 (Joydeb Das Biswas)

Abstract

This dissertation addresses of two problem in VLSI layout design. The first one is , the implementation of graphics modulo package for general floorplan considering the unified approach to topology generation and area optimization of general floorplans[3] , and the 2nd one is , the formulation of an heuristic algorithm for two-terminal net and multi-terminal net in stair-case channels on the floorplan.

Keyword : Global routing , slicible and non-slicible floorplan , algorithm , stair-case , channel , cut.

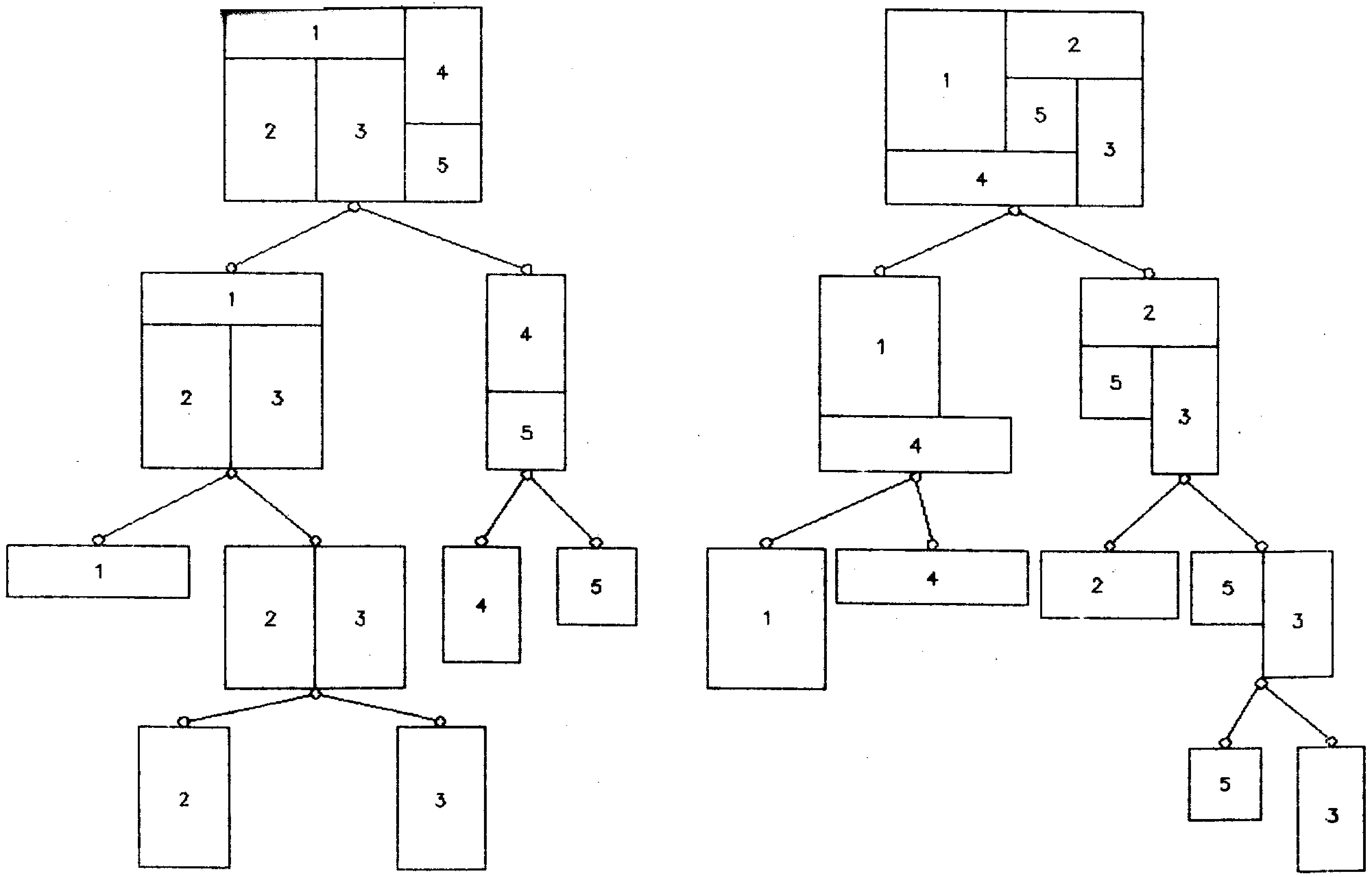
Contents

Chapter 1

Introduction

The VlSI physical design cycle consists in several stages such as Partitioning , Floorplanning , Placement , Routing , and Compaction , of which the floorplanning is important. It determines the relative positions of logical blocks on a chip based on the interconection requirements of the circuit dimensions and estimetes for area,wirelength,etc. In the problem of floorplan generation,a suitable topology and dimensions and aspect ratios of the modules are obtained which satisfies certain optimization criteria such as area , perimeter of the bounding rectangle, total wire length , etc. In the sizing problem , for a given topology , a floorplan implementation of minimum area is sought using a given set of dimensions of the constituent blocks. Time complexity of the sizing problem is polynomial for slicing floorplan and NP-complete for general(non-slicing) floorplan [1]. The existing floorplan methods can be classified into two types (a) Constructive (b) Iterative . A well known constructive method is based on graph dualization ; Algorithm in [2,1] generate only a feasible topology for a given adjacency graph in polynomial time without consideration any optimization criteria. Iterative methods assume a given topology and try to find an optimum implementation,given a set of aspect ratios. A unified approach to topology generation and area optimization of general floorplan is found in [3]. The unified approach consists of two phases , In the first phase , a top down AND-OR graph search technique is used which uses heuristic estimates desired from the aspect ratios of basic modules. In the 2nd phase, the actual values of the dimension are used to find an implementation with minimum wasted area. The output of the unified approach consists of a set of strings where each string represents a block or symbols like h , v , lt , rt , etc and the set of estimated aspect ratios for blocks.In this dissertation , we implement a graphic package which displays the area optimal floorplan obtained by taking the topology and the selected aspect ratios for each block as input.

Another important problem in VLSI design is the Global Routing which follows the placement phase of the design cycle. In the placement phase,the position of the different circuit components on the floorplan is determined. For each circuit component, a net list is provided. Space not occupied by the blocks are used as Routing region. In the global routing , the main objective is to connect the terminals attached to different modules which belong to the same net. In [4] , the channel definition is generated to stair-case(monotone) channels. The advantage of using stair-case channels is that the routing order can be easily solved by identifying suitable stair-case channels hierarchically(fig-1). In this case stair-case channels are guranted to



(a) Slicible Floorplan.

(b) Non-slicible Floorplan.

Fig.1. HIERARCHIAL POSITIONING OF FLOORPLANS.

generate acyclic digraph. Generally , the global routing for multi-terminal net is NP-complete. In this dissertation , we try to formulate heuristic algorithms for two-terminal net and multi-terminal net in stair-case channels.

The dissertation consists of two parts , the 1st part contains the graphical modulo packages for optimal floorplans and the 2nd part conatains the global routing for two-terminal nets and multi-terminal nets in staircase channels on the floorplan. In chapter 2 and chapter 3, we discuss the graphic modulo package for slicible and non slicible floorplans and in the subsequent chapters , we discuss the heuristic global routing for two-terminal and multi-terminal nets in stair-case channels.

Chapter 2

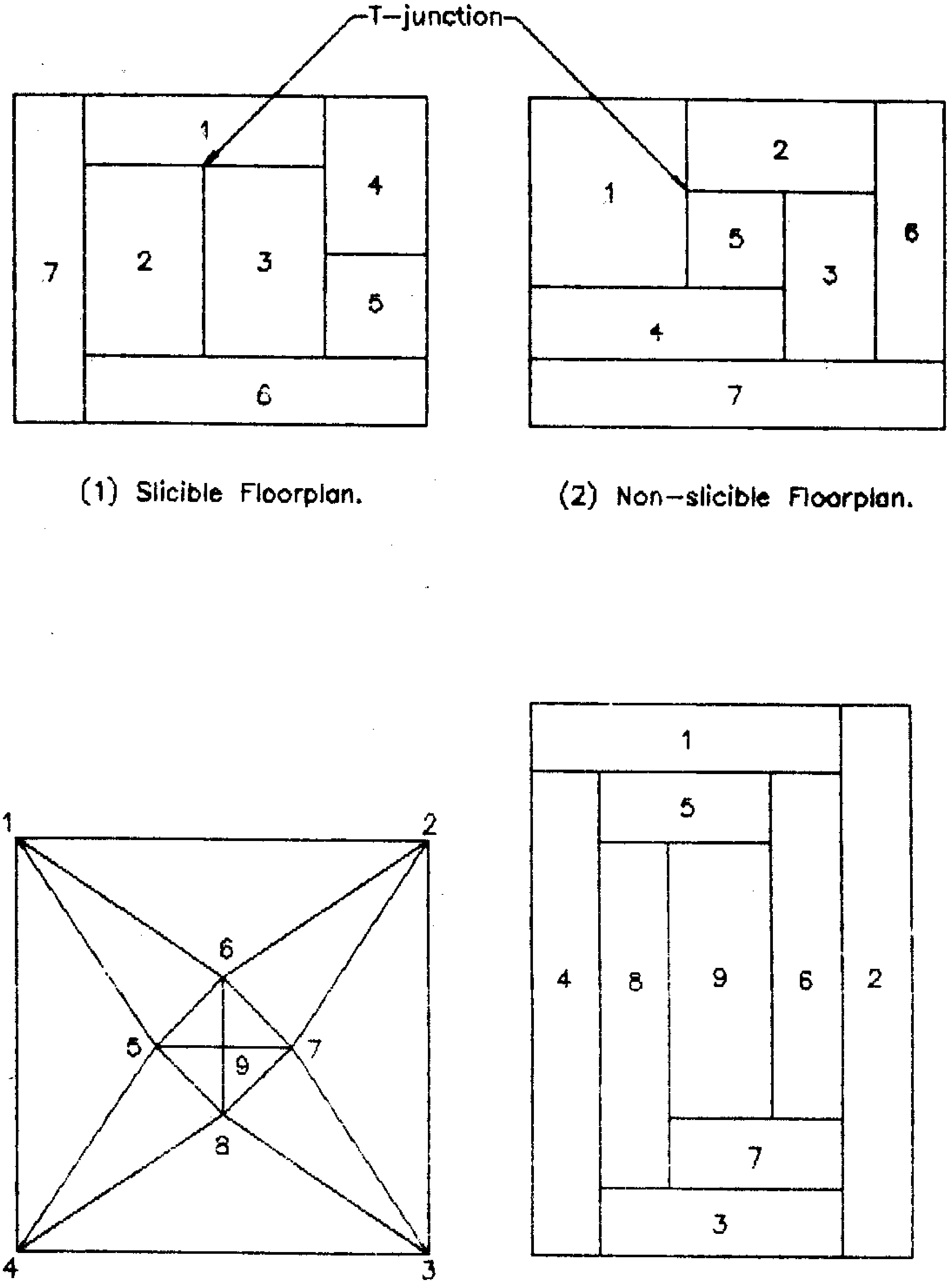
Graphic Modulo Package For Optimal slicible Floorplan

2.1 Introduction

Floorplanning is a rectangular dissection of the bounding rectangle with isothetic cut lines, called cuts into a finite number of indivisible non-overlapping rectangles known as modules or blocks. A floorplan is said to be slicible if it is obtained by recursively by using through cuts only, otherwise,it is non-slicible. A floorplan may or may not be slicible(fig 2.1,2.2). The inner core of a floorplan is obtained by removing the modules touching its boundary and the junctiors among them are T-junctions(fig 2.1) Formally a floorplan is a plane graph Gd with each edge either vertical or horizontal. The four 2-degree vertices on the outer most cycle are denoted by NW , NE , SE , SW ; all other vertices are of degree 3.

The interconection requirements among the different function modules in a floorplan are generally represented by an adjacency graph G=(V,E), where the vertex set V represents the modules in the floorplan , and an edge (u,v) in E implies the adjacency of the modules u and v. Fig 2.2 illustrates an adjacency graph and its corresponding floorplan. It is very easy to see that the geometric dual of the of an adjacency graph corresponds to a rectangular floorplan.By definition , a rectangular floorplan is embedded in a plane which implies that its adjacency graph is also a plane graph. A plane graph has a rectangular floorplan realization only under certain necessary and sufficient conditions and such an adjacency graph is called a rectangular graph. The adjacency graph given in fig 2.2 are rectangular graphs. A given rectangular graph may have more than one floorplan realization whereas the rectangular graph corresponding to a given floorplan is unique. In the formar cases , the different floorplan realizations may be slicible or non slicible. There exits a class of rectangular graphs which do not have any slicible floorplan realization , such graphs are known as Inherently non-slicible(INS) [3].

A cut in the floorplan is a path in G_d between either an exterior vertex of North(West) side and one of South(East) side.The corresponding cut set of edges in G decomposes it into exactly two non-empty subgraphs G_l and G_r.The two sequences of vertices on the two sides of these cut edges of G_d called the two boundary path P_l and P_r of the cut; are essentially the new boundaries of the subfloorplan on the two sides of the cut , corresponding to G_l and G_r. The definition of chord free path and its properties are found in [3]. If both of its boundary paths of a cut are chord free, then



(1) Slicible Floorplan.

(2) Non–slicible Floorplan.

(1) INS Graph.

(2) Non–slicible Floorplan.

Fig 2.2

it is called a slice in a floorplan and both G_l and G_r are rectangular. A vertical(horizontal) cut is between North and South(East and West) sides. The cut-set in G corresponding to a slice in a floorplan is referred to as a slice in G.

In this chapter, we take the output of unified approach to topology generation and area optimization of slicible floorplans which consists of a set of strings and a set of aspect ratios for blocks and then using it , we draw the corresponding optimal slicible floorplan.

2.2 Floorplanning Using AI Search

The unified approach to topology generation and area optimization of sliceble floorplan is described in [3] This approach consists in two phases. In the 1st phase , a floorplan topology can be constructed by recursivley partitioning the adjacency graph untill all the subgraphs produced by the sequence of cuts are indivisivle.Each partition of the adjacency graph in the dual space corresponds to a slice of the floorplan. The top-down search for finding a topology can be reprasented by an AND-OR graph , where each AND node corresponds to an actual bipartition of a rectangular section and each OR node corresponds to the possible slices. The search space is then an AND-OR graph with AND nodes and OR nodes alternating along the depth of the graph. Fig 2.3 shows a simple adjacency graph G and a part of an AND-OR graph used to obtain a topology from G.An AND-OR graph search method of AI can be used to generate a potential topology

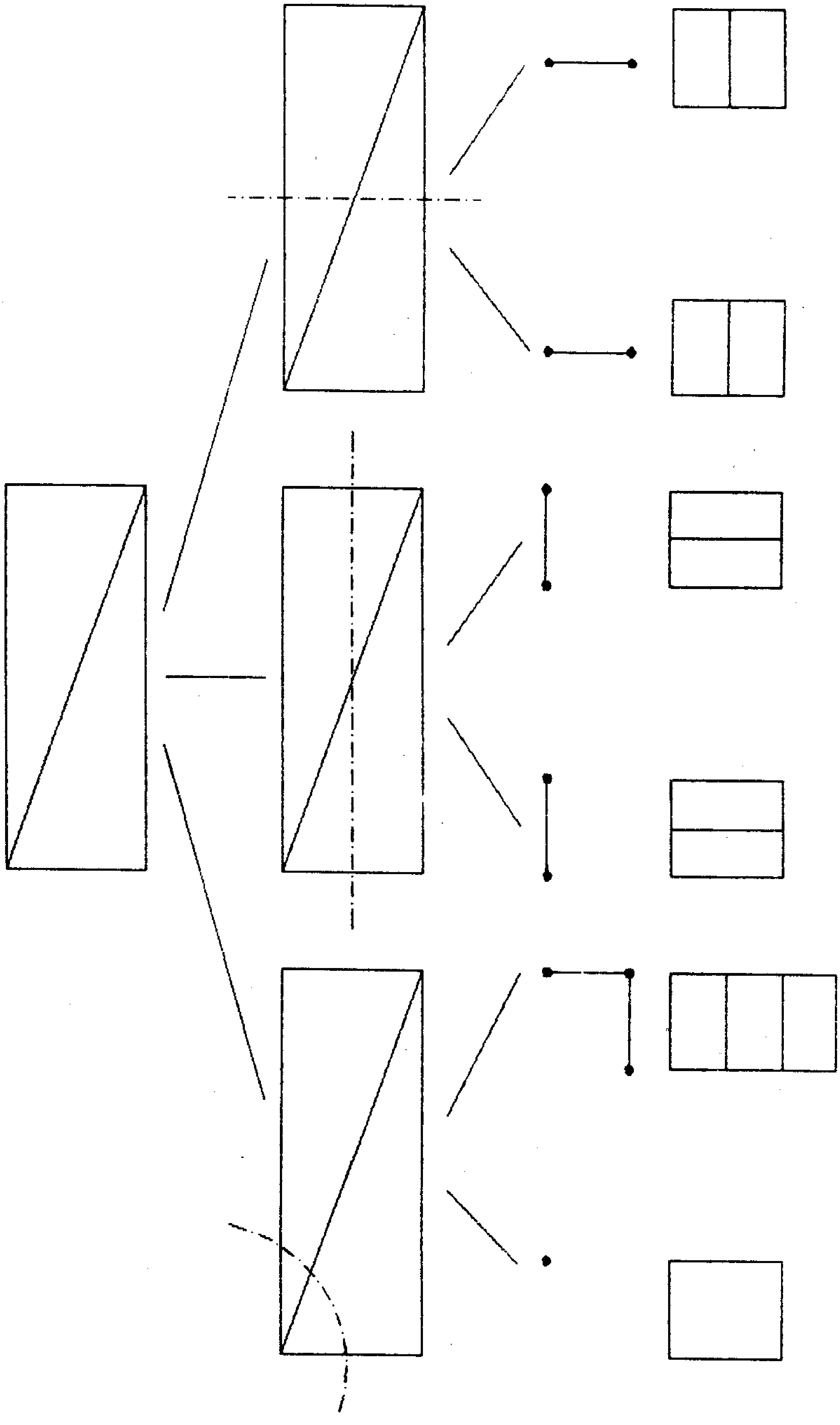
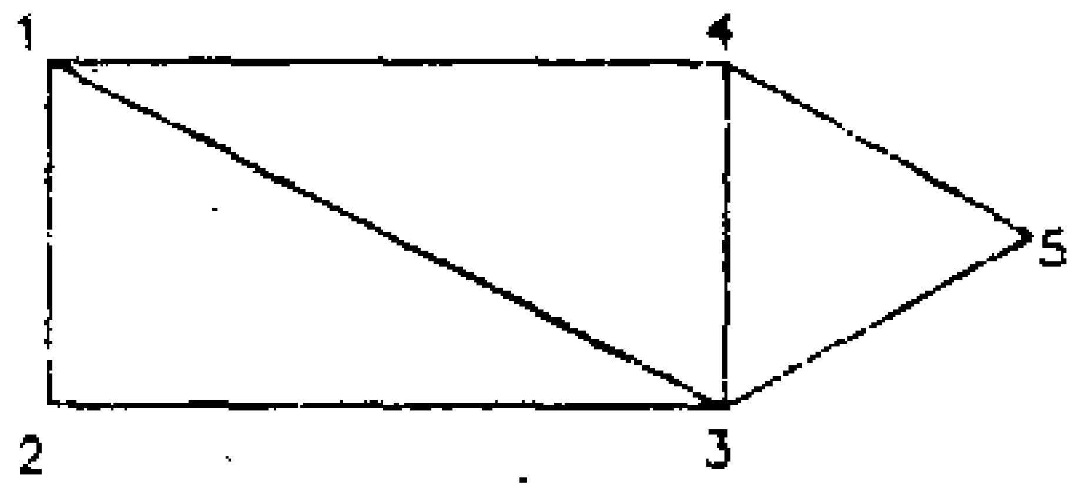


Fig 2.3

A solution is basically a binary tree , where each AND node has two successors and each OR node has one successor only.The terminal leaves of the graph represent the individual rectangular blocks. In the two phase method ,taking the output of the phase 1 which is a floorplan tree F. For F , using the given set of all possible implementations of the blocks, the implementations of the root node can be obtained by a bottom-up traversal of F in post-order and combining the implementation list of children of a node to form that of their parent in a way[3]. An optimal implementation can then be obtained by selecting a minimum cost implementation from the list of implementations for the root

For the two phase method, an adjacency graph and the set of dimensions of blocks are given as an input. The output will be a floorplan tree(given in the form of a set of strings) and the actual set of dimensions of blocks. For example,

INPUT :



Adjacency Graph

Block No	Aspect Ratio
1	(1,2), (2,1)
2	(1,2),(2,1)
3	(1,2),(2,1)
4	(1,2),(2,1)
5	(1,1)

OUTPUT :

The floorplan tree is 1 2 3 v h 4 5 h v and the actual set of aspect ratios

Block No	Aspect Ratio
1	(1,2)
2	(2,1)
3	(2,1)
4	(2,1)
5	(1,1)

2.3 Main Results

The output of the two phase method is a set of strings which is obtained by postorder traversal of a floorplan tree. Fig 2.4 shows an example of a floorplan tree and its postorder traversal and the set of aspect ratios for for blocks.The set of strings consists of integers which represent blocks and the literals v,h represent vertical or horoizontal partitions.

We take the set of strings and the set of aspect ratios for blocks as input and the output will be an optimal floorplan. The mazor steps to obtain floorplan topology are given in the following Alogorithms.

Algorithm :

Input : A set of strings of integers and literals h or v and the set of aspect ratios.

Output: An Graphical representation of an optimal slicible floorplan.

Step1:

Read the set of strings and the set of aspect ratios. Find the maximum between the largest integer in the set of strings and the number of strings.

Let r be equal to the above maximum value plus one.

Step2:

Scan the set of strings from the left side and at a time one string is considered.

Step3:

If the first character of the string is a digit , i.e , an integer , push it into the stack and go to step2 , otherwise, go to the step4.

Step4:

If the first character of the string is 'h', go to the step5 , else if it is 'v', go to step6 , otherwise , the input string is not a valid string.

Step5:

5.1: Pop two elements x and y are popped from the stack.

5.2: If it is not 1st entry , then , draw the rectangles corresponding to blocks x and y with dead space. Let original be equal to r , push r into the the stack and increase r by one and go to step2 , otherwise , go to the step5.3.

5.3: If x is equal to r-1 , then go to step5.4 , otherwise , go to step5.5

5.4: If x is original and y is not equal to original ,i.e , y is a vertex , then , draw the composite rectangle(original) corresponding to the rectangle x and the vertex y using coordinates for x and aspect ratio for with dead space. Let original be equal to r , push r into the stack and increase r by one and go to step2, else if , if x is not original and y is original , i.e , x is a composite rectangle(not original) , then , draw the composite rectangle(original) corresponding to blocks x and y using their coordinates with dead space. Let original be equal to r , push r into the stack and increase r by one and go to step2. else, i.e, x is not original and y is not original , go to step5.6.

5.5: If y is equal to original ,i.e , x is a vertex and y is a composite rectangle(original) , then , draw , the composite rectangle(original) corresponding to the vertex x and the rectangle y using aspect ratio for x and coordinates for y with dead space. Let original be equal to r , push r into the stack and increase r by one and go to step2, else , i.e , y is not equal to original and x is a vertex , go to step5.7.

5.6: If y is greater than the maximum(obtained previously in the step1), then , draw the composite rectangle(not original) corresponding to rectangles x and y using their coordinates with dead space. Push r into the stack and increase r by one and go to step2 , else , i.e , y is less or equal to maximum i.e , y is a vertex , then , draw the composite rectangle(not original) corresponding to composite rectangle(not original) x and the vertex y using coordinates for x and the aspect ratio for y. Push r into the stack and increase r by one and go to step2.

5.7: If y is greater than the maximum(obtained previously in the step1), then , draw the composite rectangle(not original) corresponding to vertex x and the rectangle y using aspect ratio for x and the coordinates with dead space. Push r into the stack and increase r by one and go to step2 , else , i.e , y is less or equal to maximum i.e , y is less equal to maximum , i.e , x and y are vertices , then , draw the rectangle(not original) corresponding to vertices x and y using aspect ratios for them with dead space. Push r into the stack and increase r by one and go to step2.

Step6:

The same steps as in the previous step5 except that in this case , blocks are joined vertically.

Chapter 3

Graphic Modulo Package For Optimal Non-Slicible Floorplan

In this chapter , we implement a graphic modulo package for optimal non-slicible floorplan using a floorplan topology which is given in the from of a ste of strings and a selected set of aspect ratios for blocks.

3.1 Z Cuts

Let F be a non-slicible floorplan and G be the corresponding rectangular graph.

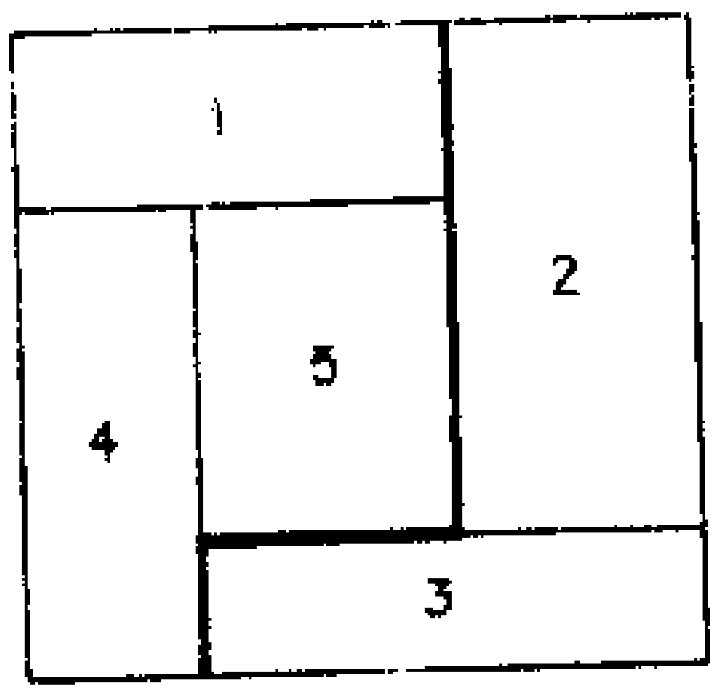
Definition1 :

A generalized k-cut in F is a cut for which the maximal chord[3] on either of its boundary paths is at most k.

Definition2 :

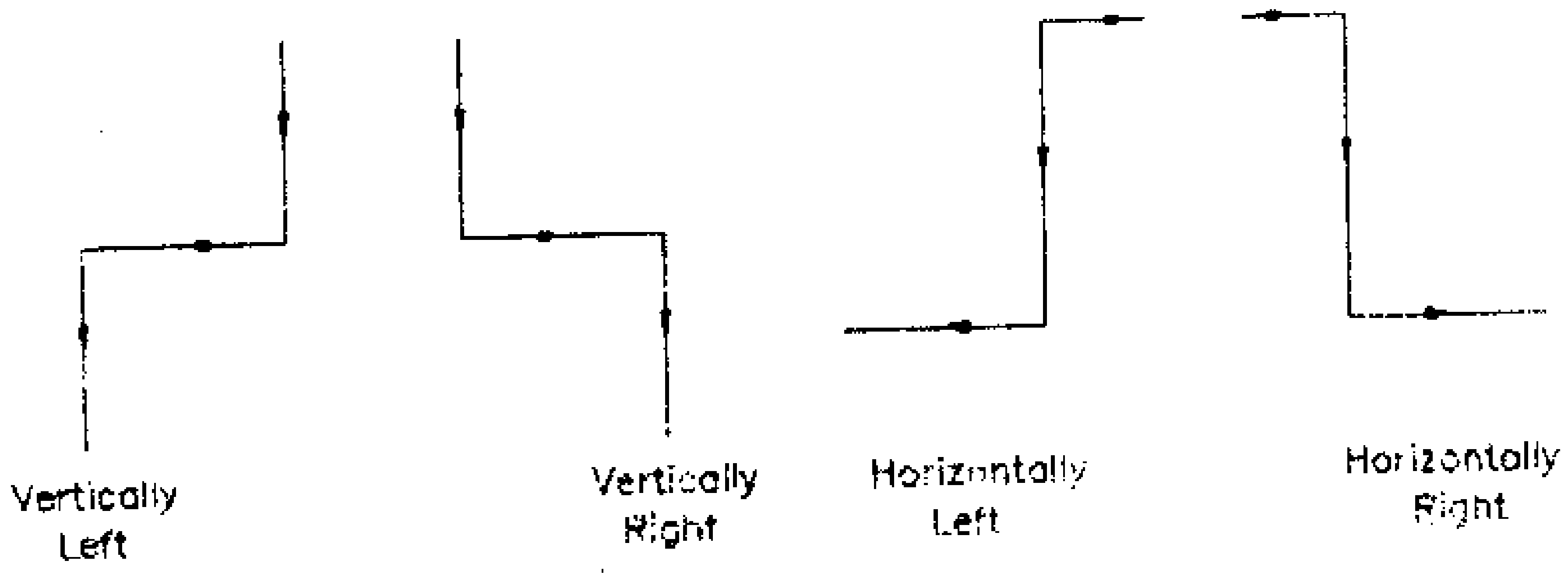
A Z-cut in F is a generalised k-cut with k= 1.

Example ,



Z – cut

A cut-set in G for a generalized k-cut in F is referred to as a k-cut of G. A slice is a generalized k-cut with k=0.Any generalized k-cut in F has a rectilinear embedding with 2k bends or intermediate corners , decomposes F into two sub floorplans each having a rectilinear polygonal bounding with 2k+4 corners. It is easy to see that any generalized k-cut can always be rectilinearly embedded with $2k_1$ bends for any k_1 > k. Floorplan with more than 4 corners are called rectilinear floorplans . Both the sub-floorplans for a slice and a Z-cut have rectangular and L-shape boundaries respectivly. An L-module has one concave and five convex corners . Four of these convex corners are denoted by NW , NE , SE and SW in a clockwise manner , where the NW corner is convevtionally the left most one with the maximum height ; the remaining two corners are labled as MW(mid-west) and ME(mid-east). The Z-cut which divides a floorplan into two sub-floorplans have the orientations Vertically Left , Verticall Right , Horizontally Left and Horizontally Right. The follwing figure shows the various orientations of Z-cut.



3.2 Main Results

The output of two phase method for non-slicible case consists of a set of strings, again these strings consists of integers which represent blocks and strings like , v , h , rt , lt , and they represent vertical and horizontal partitions and right turn and left turn respectivly , and aspect ratios for blocks. The set of string is obtained by postorder traversal of a floorplqn tree. Fig 3.2 shows an example of a floorplan tree.

Algorithm :

Input : A set of strings of integers and h , v , rt , lt and a selected set of aspect ratios.

Output : A graphical representation of non-slicible floorplan.

Step1:

Read the set of strings and the selected set of aspect ratios for blocks. Find the maximum between the largest integer in the set of strings and the number of strings in the set of strings.

Let us introduce two variables , say r , and first and one flag variable . say , start. Let r be equal to the maximum plus one and , start be equal to one , and first is equal to zero.

Step2:

Scan the set of strings from the left side and one at time one string is considered.

Step3:

If the 1st character of the string is l or r , go to step4 , otherwise , go to step2.

Step4:

If the 1st character of the string is l , go to step5 , else if it is r , go to step6 , otherwise , it is an invalid string.

Step5: If the 1st character of the previous string is v , go to step7 , else , if it is h , then go to stcp8.

Step6:

If the 1st character of the previous string is v , go to step9 , else if it is h , go to step10.

Step7:

Call the step VER-LFT-TURN which is described at the end of the step10.

Step8:

Call the step HOR-LFT-TURN which is the same as the VER-LFT-TURN except that in this case , vertical case is replaced by horizontal case.

Step9:

Call the step VER-RGT-TURN which the same as the VER-LFT-TURN except that in this case , right turn is considerd.

Step10:

Call the step HOR-RGT-TURN which is the same as HOR-LFT-TURN except that in this case , right turn is considered.

Step VER-LFT-TURN:

If the start is equal to one , i.e , it is the 1st entry , then , set start is equal to zero and call procedure Determine which determines two set of sub strings and they are joined vertically with left turn , for example , the set of strings 12 13 v 9 h 11 10 h v lt breaks into sets of sub strings 12 13 v

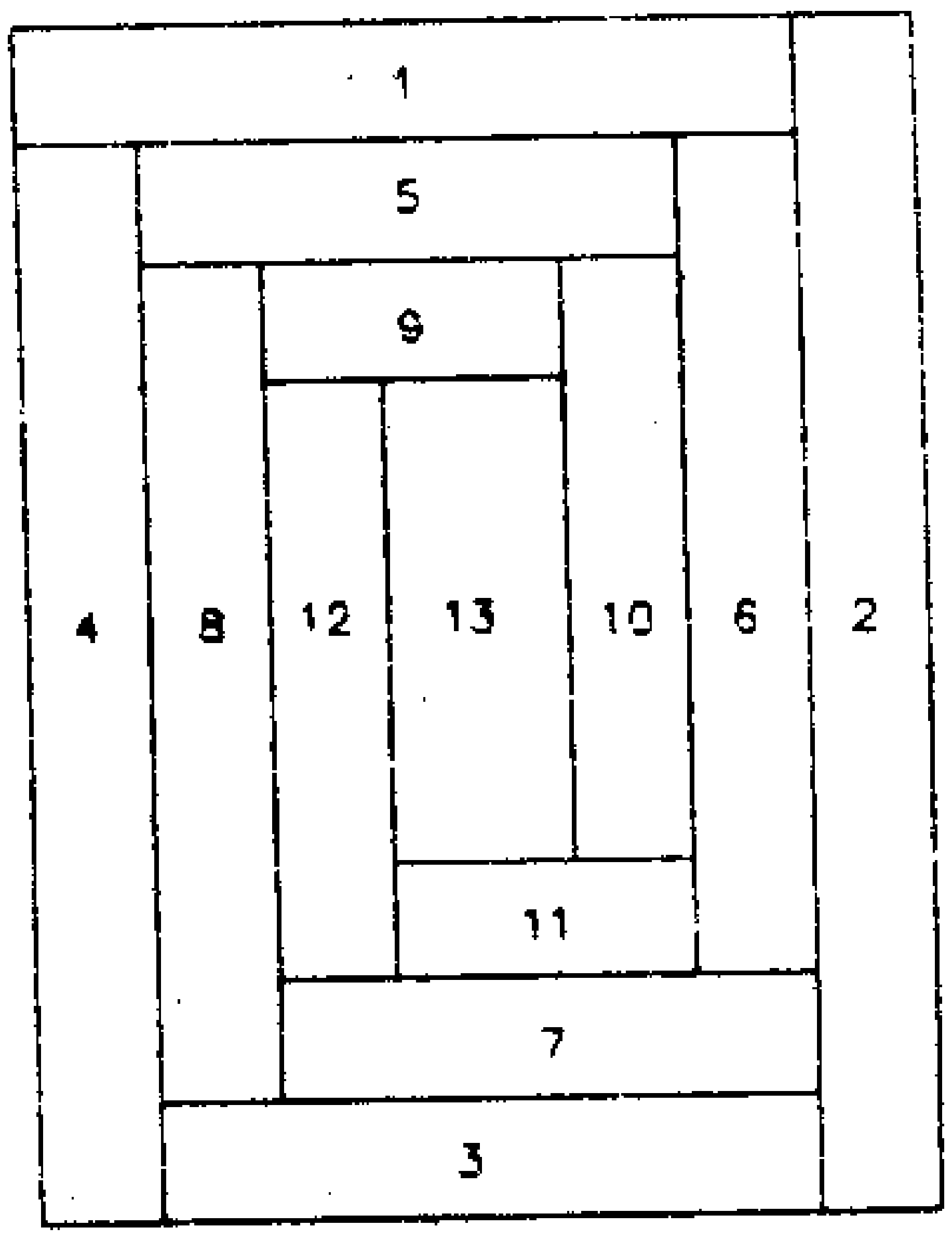
9 h and 11 10 h.They joined vertically with left turn. Four variables ind1 , ind2 , index1 , index2 , are introduced to represent begin and end of of two sub strings. In the above example , values of ind1 , ind2 , index1 and index2 are 1 , 5 , 1 , 3. Now using the values of ind1 , ind2 , index1 and index2 , two sets of sub strings and aspect ratios for blocks , we draw the rectangles corresponding to two sub strings as in the sliceble case and then join them vertical with left turn. In the above example , we 1st draw rectangles corresponding to two sub strings 12 13 v 9 h and 11 10 h separately and then join them vertically with left turn. Push r in to the stack and increas r by one. If start is not equal to one , i.e , start is equal to zero , then , we proceed in same way as in the above case and in the slicible case in chapter2.

Step11: Call the step slicible which is described in chapter2.

Example , Input: 4 8 12 13 v 9 h 11 10 h v lt v 5 h 7 6 h v lt v 1 h 3 2 h v lt.

Block No	Aspect Ratio
1	(6,1)
2	(1,6)
3	(6,1)
4	(1,6)
5	(4,1)
6	(1,4)
7	(4,1)
8	(1,4)
9	(2.1)
10	(1,2)
11	(1,2)
12	(1,2)
13	(1,1).

output



Non-Slicible Floorplan.

Chapter 4

Global Routing in stair-case channels

4.1 Introduction

In the placement phase , the exact locations of circuit blocks and pins are determined A net list is also generated which specifies the required interconections. Space not occupied by blocks can be viewed as a collection of regions. These regions are used for routing and are called as routing regions. Each routing region has a capacity , which is the maximum number of nets that can pass through that region. The process of finding the geometric layouts of all the nets is routing. Nets must be routed within the routing region and must not violet the capacity of any routing. In addition , nets must short circuit , that is , nets must not intersect each other. The objective is to connect each net such that the total wire length as well as the longest wire length will be minimum.

There are two types of routings a) Channels b) switchboxes. A switchbox is a rectangular area bounded on all sides. A channel is a rectangular area bounded by two opposite sides by the blocks. In this case , we consider stair-case channels [4] , we may call this channel as rectilinear channels. Fig 4.1 shows an example of rectilinear channel. Capacity of a channel is a function of the no of of layers , wire width and wire separation. In this case , we consider stair-case (monotone) channels on the floorplan. The advantage of choosing stair-case channel is that a suitable routing order can be easily solved by identifying suitable stair-case channels hierarchically. It can be easily shown that such stair case are guaranted to generate acyclivc channel digraph. after dividing the floorplan hierarchically by using stair-case channels , the channels are routed in bottom up fashion. Thus , the longer channels which appear in the top of hierarchy will be routed at the end and hence should be preferbly have less congestion.

In this chapter , Global routing using stair-case channels for two-terminal net and multi-terminal net will be considered. In the case of two-terminal net , the global routing problem will be the shortest path problem and in case of a multi-terminal net , the global routing problem wil find a minimum steiner tree which can not be solved in polynomial time , i.e , NP-complete.

4.2 Formulation

we are given a floorplan which consists of finite number of blocks. Geometric positions of each blocks are known , i.e , two opposite corners of each block are known and from these , other two corners of block are easily obtained.

We are also given stair-case channels corresponding to a floorplan where each channel consists of a sequence of +ve integers (Integers lie on the boundary of the blocks) ending with the integer -1 . Fig 4.1 shows an example ,

Stair-case channels for the given example , $I_1 = <1 , 2 , 12 , 9 , 10 , 6 , -1>$, $I_2 = <1 , 8 , 9 , 10 , 6 , -1>$, $I_3 = <2 , 12 , 11 , 10 , 6 , -1>$, and $I_4 = <2 , 12 , 11 , 4 , 5 , -1>$.

we now draw a channel intersection graph which is obtained from a set of channels for a given floorplan and it shows intersections of channels. Algorithm for obtaining channel intersection graph is given below. Note that two channels intersect with each other at most two times as they are monotone.

Algorithm:

Input : A set of channels for a floorplan.

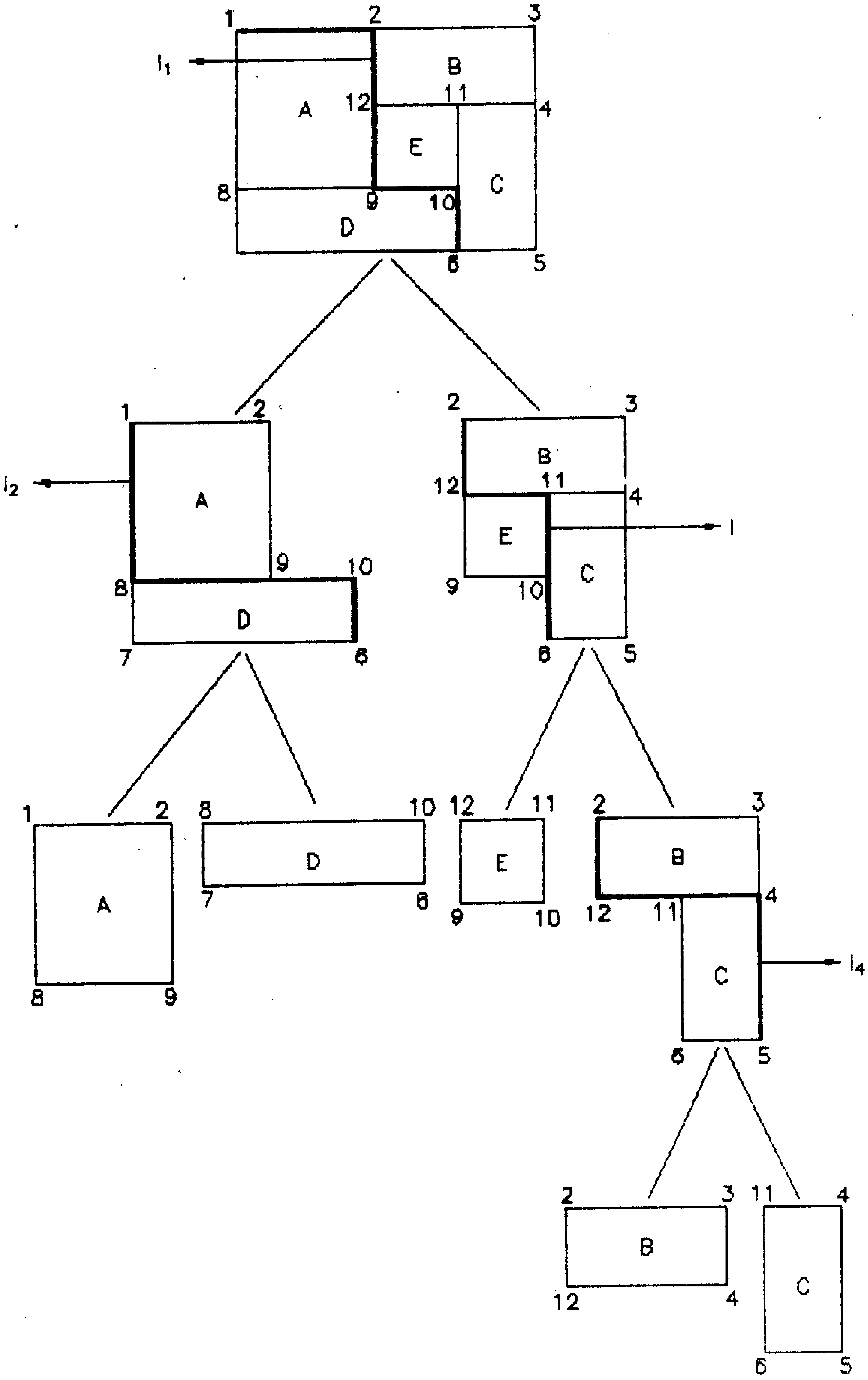

1
2
3
I1
B
12
11
4
A
E
C
8
9
10
D
6
5
1
2
I2
A
9
10
8
D
7
6
2
3
B
12
11
4
I
E
C
9
10
6
5
1
2
A
8
9
8
10
D
7
6
12
11
E
9
10
2
3
B
4
12
11
I4
C
6
5
2
3
B
12
4
11
4
C
6
5

Fig. 4·1

Output : Adjacency Matrix or adjacency list of channel intersections.

Step1 : Read a set of channels and let n be the number of channels.

Step2 : Scan the set of channels and at a time one channel is considered , say , the ith channel. If i is less or equal to n-1 , then go to step3 , else , exit.

Step3 :Increase i by one , if i is less or equal to n , go to step4 , else , exit.

Step4 : Scan the (i-1)th channel and at a time one channel vertex is considered , if the vertex is equal to -1 , go to step5 , else , go to , step3.

Step5 : Scan the ith channel and at a time one vertex is considered , if it is not equal to -1 , go to step6 , else , go to step4.

Step6 : If scan vertices are equal , go to step7 , else , go to the next vertex of the ith channel and go to step5.

Step7 : Introduce a flag variable , say , IN and it is set to equal to 1. If previous vertices are not equal and IN is equal to one , then , we have an intersection , IN is set to 0 , go to the next intersection position and the next vertex in the (i-1)th and ith channels. else , if the current vertices are equal , IN is equal to one and vertices are not end vertyces , then , we have an intersection and go to the next intersection position. Next go to the next vertex of the ith channel. If the vertex in the (i-1)th channel is -1 , then , exit , else , go to the next vertex of the (i-1)th channel.

Step8 : If is less is equal to n-1 , go to step3 , else go to step2.

Example ,

Input :

A set of channels

1: 1 12 9 10 6 -1

2: 1 8 9 10 6 -1

3: 2 12 11 10 6 -1

4: 2 12 11 4 5 -1

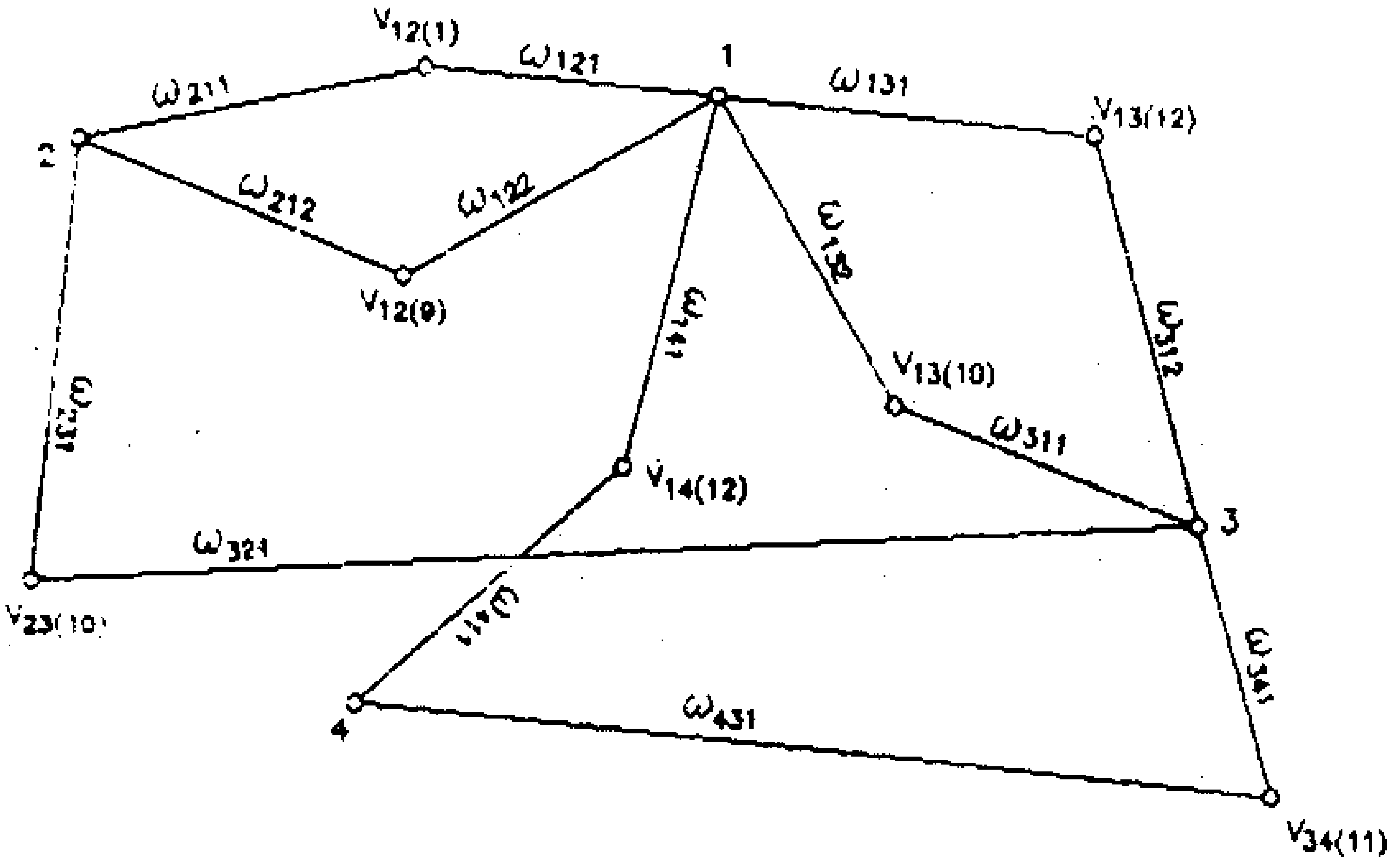
Output:

1: 0 0 1 9 12 10 12 0

2: 0 0 0 0 10 0 0 0

3: 0 0 0 0 0 0 0 0

Channel Intersection Graph



Let x_1 , y_2 and x_2 , y_2 be begin and end coordinates of a channel. The , the coordinate of mid position of the channel is $(x_1 + x_2)/2$, $(y_1 + y_2$ $)/2$. Then , in the channel intersection graph , the manhattan distances between intersection points on the channel and mid position of the channel are considered as weights.

4.3 Main Results

Algorithm(Two Terminal Net) :

Input : Channel Intersection Graph(c.i.g) and a set nets

Output : A Set of Paths for each Net.

Step1: Find first on which channel a terminal lies. Rpeat this for every terminal in the net.

Step2: Find the nearest channel intersections of terminals in a net.

Step3: Find the shorest distance between the nearest channel intersection using DIJKSTRA's algorithm[8 , 9]. Once , a route is obtained , the associated weights are deleted from CIG.

Step4: Repeat this Process for every net in the given set of nets.

For multi-terminal net , we repeat the same 1st two steps as in the the two-terminal net case and then , we construct a basis set where each such set consists of each channel intersection. Following this initialization and applying the genersalized. Dijkstra's algorithm between any pair of vertices in the basis set , we compute a minimum weight path. This path is then output as a fragment of steiner tree and two group containing the end points are merged.

Bibliography

[1] T.Lengauer , Combinatorial Algorithms for Integrated circuit layout design , wily , 1990

[2] ——— ,VLSI Floorplan Generation and Area optimization using AND-OR Graph Search , Proc 8th Intl Cont on VlSI Design , 1995 , 370-375.

[3] P.S.Dasgupta , S.S.Kolay and B.B.Bhattacharya , A unified Approach to Topology Geneartion Area and Area optimization of General Floorplans , IEEE , 1995 , 712-715.

[4] S. Mazumder , Routing Driven floorplan Partitioning using Stair-case Channels , M.tech Dissertation , I.S.I , Calcutta.

[5] Ronald L. Rivest , The " PI "(Placement and interconnect) System , MIT Laboratory for Computer Science , Cambridge , Mass. 02139 , March 1982.

[6] Ronald L. Rivest and Charles M. Fiduccia, A " GREEDY " CHANNEL ROUTER , MIT Laboratory for Computer Science , Cambridge , Mass. 02139 and GE Research and Development Center , Schenectady ,New York 12301 , March 1981.

[7] N.Sherwani , Algorithms for VLSI Physical design Automation , Kluwer Academic Publishers , 1993.

[8] A.V.Aho , J.E.Hopcropt and J.D.Ullman , The design and analysis of computer algorithms , Addision-Wesley Publishing Company , 1974.

[9] E.Horowitz and S.Sahni , Computer Algorithms , Galgotia Publications Pvt ltd , 1985.