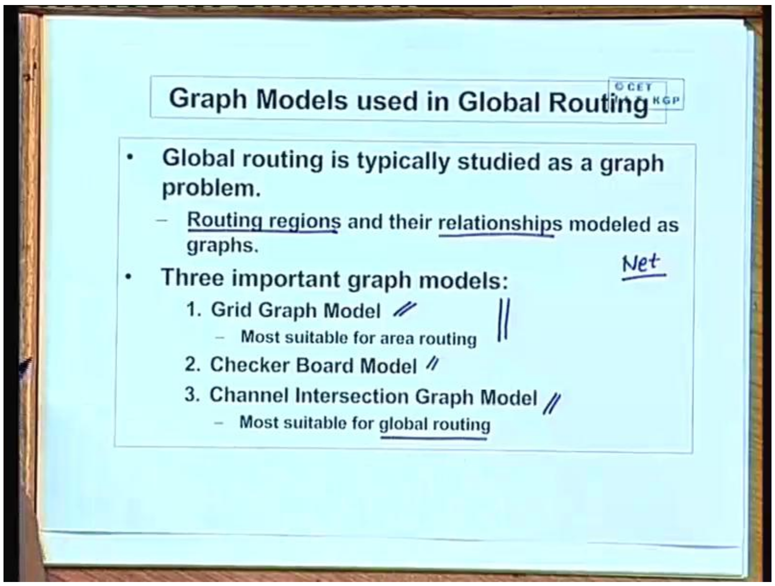
Electronic Design Automation
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture No #24
Backend Design Part-X

In our last lecture we had introduced the problem of global routing. Now in this class what we will look at? We look at the different ways and techniques people use to tackle the global routing problem. So how you represent the problem and how you process that representation in order to get a solution to the global routing problem.

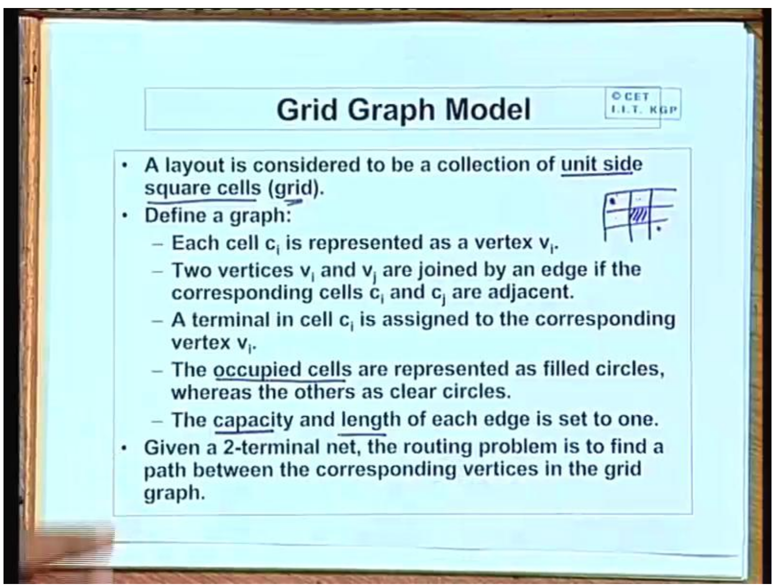
(Refer Slide Time: 01:38)



Now most of the global routing approaches they are based on some kind of graph theoretic approaches. There are various graph models which have been proposed and are used. Now the idea behind use of graphs is that well we had talked about the routing regions. Now in global routing we have the routing regions, we have the inter relationships that means how this routed regions are connected among themselves and we will have to find out that for each given net what are the sequence of routing regions we have to traverse through. So in the graph this kind of relationship between the routing regions must be captured in some way that is needed. Now three important graph models have been explode one is called the grid graph model.

Well grid graph model is more suited for you can say for grid routing kind of scenarios where you are modeling the graph with the resolution of those grids as in case of lee's algorithm or hadlock algorithm. So we can say that the grid graph model is just another way of representing the two dimensional matrix that grid cell matrix. Now this grid graph model you can use, you can say some kind of a course representation of the same to reduce on the memory you get something called checker board model we will see this thing. And finally the model which is used most widely for global routing that is something called channel intersection graph. So let us see these different models and how these models can be used for the purpose of global routing.

(Refer Slide Time: 03:34)

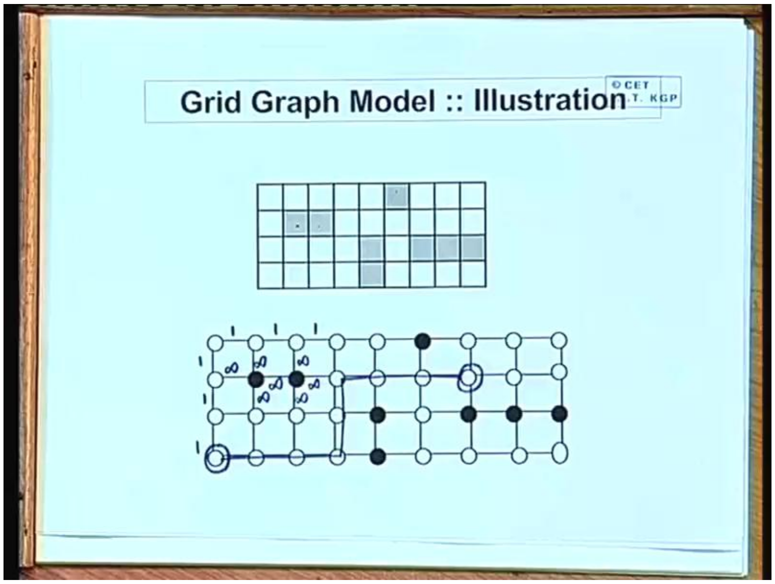


First we talk of the grid graph model. Grid graph model just like the lee's algorithm we had mentioned earlier that we have the layout considered as an array of unit size square cells. These are the so called grids. So if you have a grid like this suppose you have a grid structures like this. Now we define a graph where each of this grid cells represent a vertex. So each of the cells lets call c_i this represents a vertex v_i in the corresponding graph. And two vertices are joined by an edge if the two grid cells are adjacent neighbors. And suppose with respect to the net list, well this is just a representation of the area in terms of grid. Suppose I have a net list where a point which is located somewhere here has to be connected to a point located somewhere here.

So this particular terminal will have to be assigned to the corresponding vertex in the graph if the terminal resides within cell ci then with respect to the graph we will have to consider the vertex v_i. The occupied cells are marked with the special tag just even in the graph there are two kinds of vertices one are the occupied vertices other are the free vertices. Vertices are also colored using two colors using some kind of a bit assignment you can do that. So you have a way of keeping track of the occupied cells. And in this kind of a representation the capacity and length of each edge is set to one. Because since you are having the grid at the lowest possible level you can have only one wire running between adjacent cells that is where the capacity is one.

Length is also between adjacent cells length is also one so length and capacity both are one. Now with respect to the graph if we have a two terminal net given to you then the global routing problem will be to find a path between the two corresponding vertices in the graph overcoming the obstacles. Well the obstacles you can define by having an infinite weight associated with the edges which are coming out of that. So if you use any kind of short shortest path algorithm like the dijkstra's algorithm you will be able to find a path which will be overcoming the obstacles. Just as an illustration.

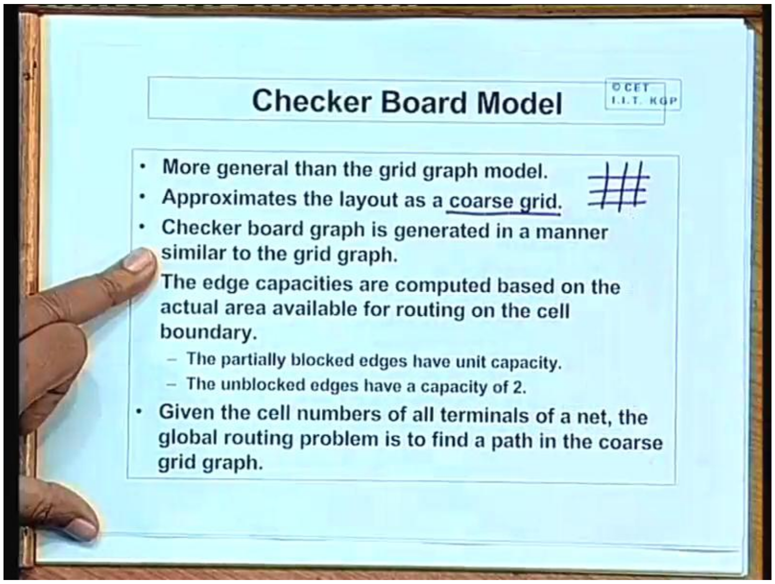
(Refer Slide Time: 06:28)



Suppose we have a layout like this where as you can see some of the cells these are occupied. So this you can straightaway map into a graph where the occupied cells are shown as solid vertices and the other vertices are the free cells. Now in terms of the graph all these edges can have a weight of one. But for the obstacles you can assign a weight of infinity to the edges incident on it. So if you do that, then for example if you want to connect from this source to this target, then it will find out a path avoiding the dark cells because all the weights there are infinite in terms of its value.

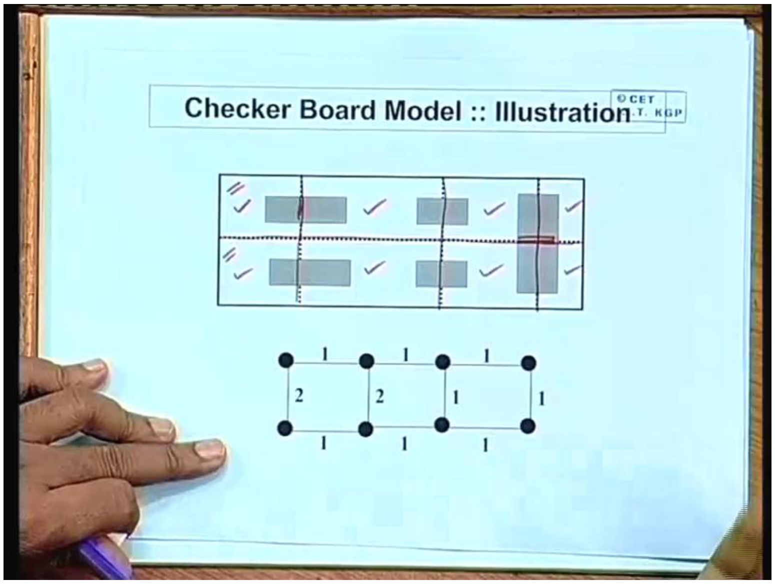
So it will possibly find a path like this. So grid graph model and the approach we have just mentioned this is suitable for finding a path like lee's algorithm. But it is not really very suitable for global routing. Because, in global routing you seldom represent the problem like this in the form of grids at this resolution ok. But in fact these methods have some utility say at some stage of routing we will find that the conventional routers are failing to find out a path then you will have to revert back to methods like this which will find out a path if it exists ok fine. So a modification of the grid graph model which will consume much less memory.

(Refer Slide Time: 08:19)



This is called a checker board model or a checker board graph. Well this is also a grid structure with respect to the layout but it is not a fine uniform size grid as in the earlier case layout is represented as a coarse grid where the grid sizes are also unequal some grid may be thinner some grid may be wider. So we will take an example the graph is generated in a manner which is very similar to the grid graph. But here instead of defining the grid lines in a uniformly spaced manner, here you define grids with respect to the obstacles which are already there on the floor right. So I am giving an example just before proceeding.

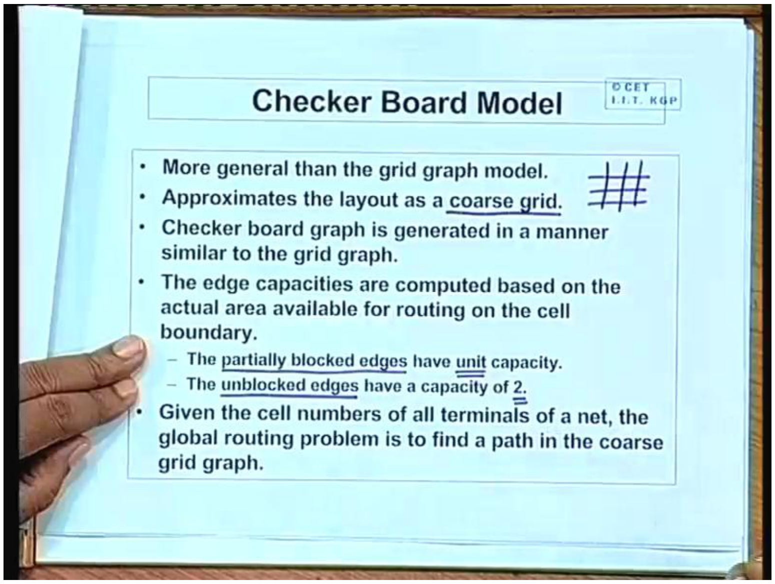
(Refer Slide Time: 09:17)



This is an example this is layout floor these are the obstacles. So I define certain edges on this graph with respect to these obstacles see. Let us consider these imaginary lines see an imaginary line will be cutting an obstacle only once. So you do not cut one obstacle twice with horizontal or with a vertical so means a horizontal line can cut an obstacle only once a vertical line can cut an obstacle only once. So each obstacle must be cut at least once these are the requirements. Now these imaginary lines break up the total layout space into eight regions. Right in the graph these eight regions correspond to the vertices ok. Suppose between these two regions say lets consider this region and this region the boundary between these two regions is not obstructed. There is no obstacle which is partially blocking the boundary which means that the routing capacity of this boundary is maximum.

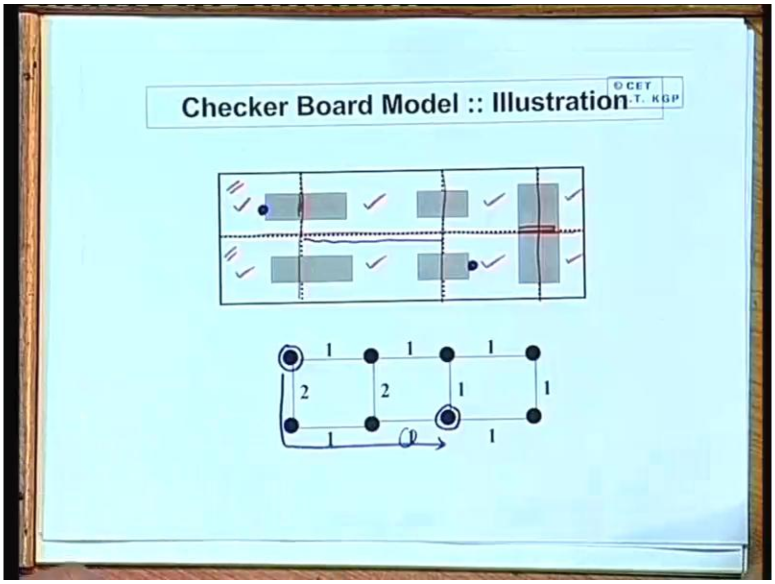
That is why we give a weight two to this edge. Similarly this boundary is also unobstructed this also has a weight two. But you look at the other boundaries say you take this boundary it is partially obstructed by this obstacle so it has a weight of one. This two also it is partially obstructed this has a weight of one you take this and this it is partially obstructed it has a weight of one. So an edge represents a boundary between two regions. Now if the boundary is partially obstructed you give a weight of one if it is unobstructed you give a weight of two. Well of course you can you can refine it further depending on the size of the region you can give other numbers 2, 3, 5, but in simplest case you can assign these two numbers 1 and 2. So let us again come back to the previous slide.

(Refer Slide Time: 11:57)



So the edge capacities in the graph they are computed based on the actual area available. Now in the example we have cited that the partially blocked edges are having a capacity of one the unblocked edges are having a capacity of two. But as I said that but in your case if you want to have some other values you can also have that you can also have the boundary capacity is as the weight not necessarily 1 and 2 you can have any number k in general. Now if we have a model like this then your global routing problem will be to find out the best cost path not necessarily best cost path a path through the coarse grid.

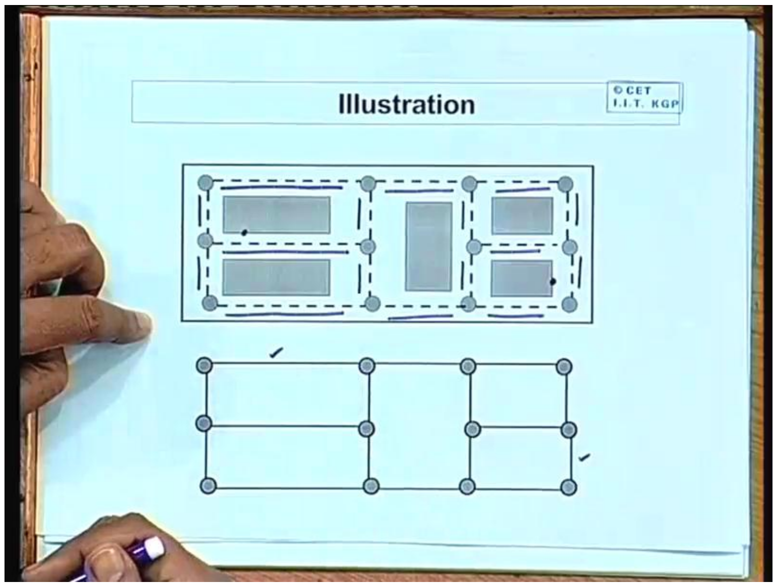
(Refer Slide Time: 12:50)



See suppose you have representation like this now you have to connect a pin say out here to a pin out here. So you know the corresponding vertices you have to find out a path from this vertex to this vertex. Suppose you have you finally find out a path like this. Now the edge weights they do not indicate the cost of the path really. Because in this graph you do not have any cost of the path lengths that you can calculate separately. These are something like the channel capacities. Suppose one wire has been routed through this channel which initially had a capacity of two the residual capacity will decrease. Now well if each line you interconnect that contributes to capacity of one if you assign the numbers in that way. So whenever you route a net you decrease the capacities along that path by one. So if one of them becomes zero in the next iteration you will be not you will not be considering that edge because that edge is having no more the capacity to route any further connections.

So your path finding algorithm should take care of that. So this checker board model is suitable to address the global routing problem because it is not as fine as the grid graph model. So it identifies the obstacles and there boundaries tries to find a path. But the problem with this model or method is that well ultimately when we proceed to do the detailed routing, there we talk about channels and switch boxes typically channels. But in this kind of a graph we are not really using a channel as a basic primitive because you look at some of the boundary say this boundary this boundary will be spanning several different channels ok. So it would be better in terms of our future processing if we can have a data structure but the channels are individually identifiable ok. So the next data structure does just that this is the so called channel intersection graph.

(Refer Slide Time: 15:30)

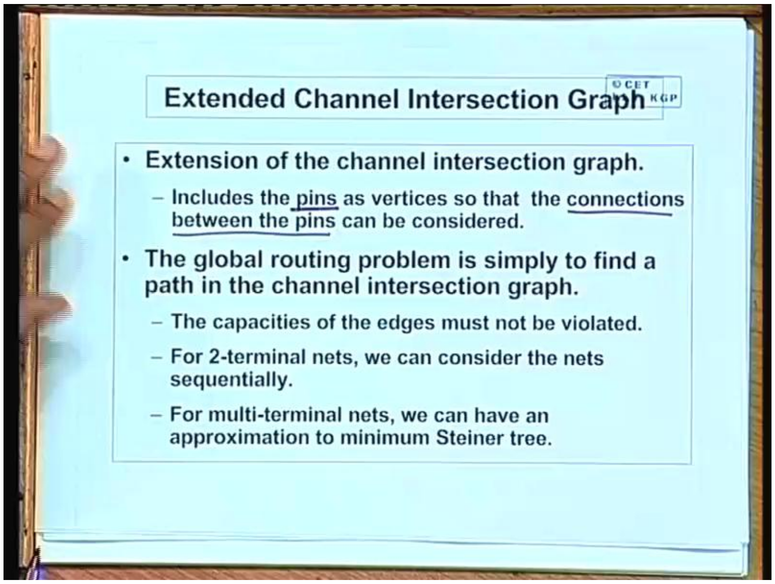


Well here the total layout area is identified by a set of channels. Suppose this is a channel this is another channel this is a channel this is a channel so on. Now in the graph the vertices will represent channel intersections not the channels like say in this example. This will represent a vertex this will represent a vertex this will represent a vertex and of course these are the end points this will represent vertices. So the vertices represent the channel intersections. Now when we talk about a channel intersection it defines intersection of two channels. Two vertices vi and vj are connected by an edge if there exist a channel between ci and cj. For example if two vertices are connected by edge there will be a channel in between them.

So edges represent channel and the vertices represent channel intersections. And of course edge weight will represent channel capacity that means what is the capacity of the channel just an illustration to show. Well in the diagram on the top we have shown some obstacles and on the same diagram I have super impose the graph. Well this graph I also have shown separately down below. See these dotted lines are the channels. This is one channel, this is one channel, this is a channel, this is a channel, and these are all individual channels. These are T junction ok. So we break it up into three channels. These are all channels and the channel intersections are represented by the vertices. So in this data structure this graph we can capture information about the channel and their connections.

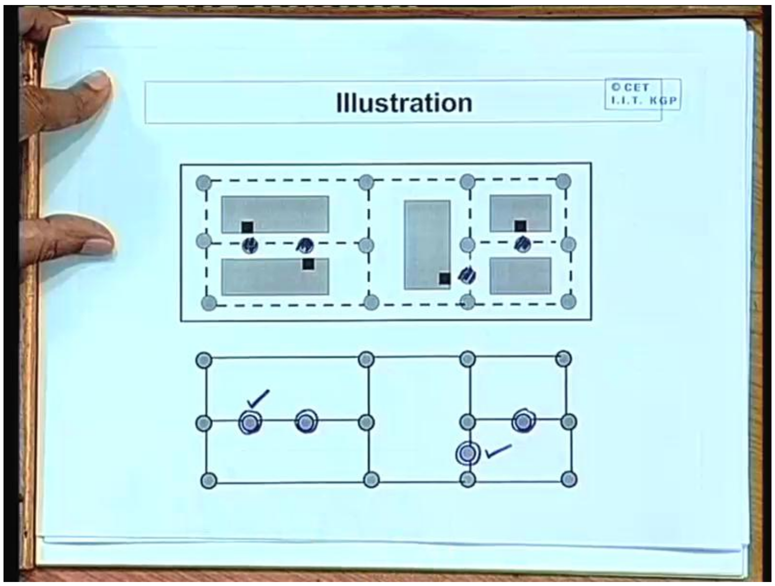
Suppose we want to find out a path from somewhere here to a path say somewhere here then we are talking about channel this to channel this. So somehow we will have to find out a path from here to here. But in this representation it is a it is a little difficult to identify exactly from where we are trying to find out the path. So there is a slightly modified representation where you also have a way of representing the points from where connections are need to be drawn. There is a slight extension to this data structure this is called the extended channel intersection graph. Because in the channel intersection graph the nodes in the graph where simply the channel intersections.

(Refer Slide Time: 18:58)



But in the extended graph what we say we add some new vertices to the graph. This new vertices are the pins from well the pins between which you need to do the interconnections includes the pins as the vertices the reason is that if pins are included as vertices. Then the graph you can say that I want to connect this pin with this pin. So connections between the pins can be very easily captured. Well I am coming to this first let me give an example.

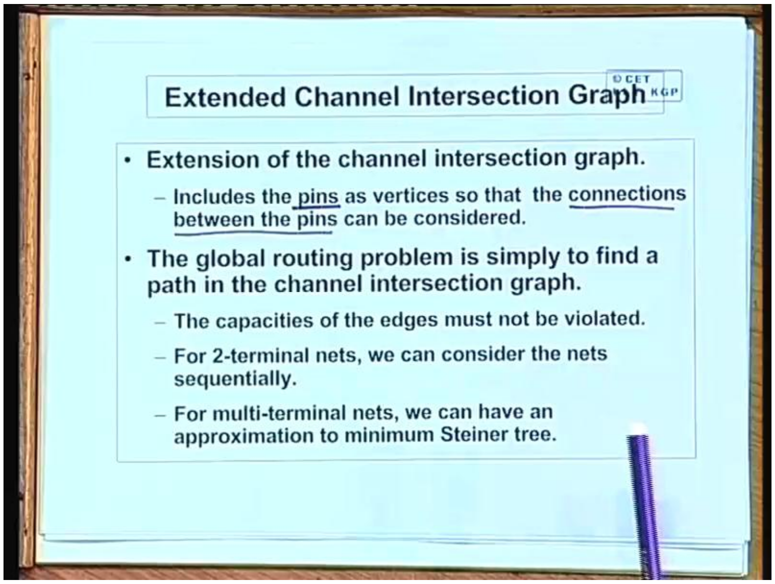
(Refer Slide Time: 19:33)



Well the same example where I have shown four pins this pins are showed by this black dots. Now in terms of the graph we have added some new vertices here, here, here, here, these vertices are very near to the corresponding pins. And if we have the corresponding graph then these four vertices indicate the pins. So now if you say that I want to connect this pin with this pin, well I can find out a path through this graph. Now as I said each edge will represent the channel capacity that also has to be taken care of by the router. So as nets are routed the channel capacity will be going on decreasing ok. Fine yes [Student Noise Time: 20:38] two points these methods all connect two points at a time. Yes [Student Noise Time: 20:44]. Yes [Student Noise Time: 20:46] No. Well, no, no.

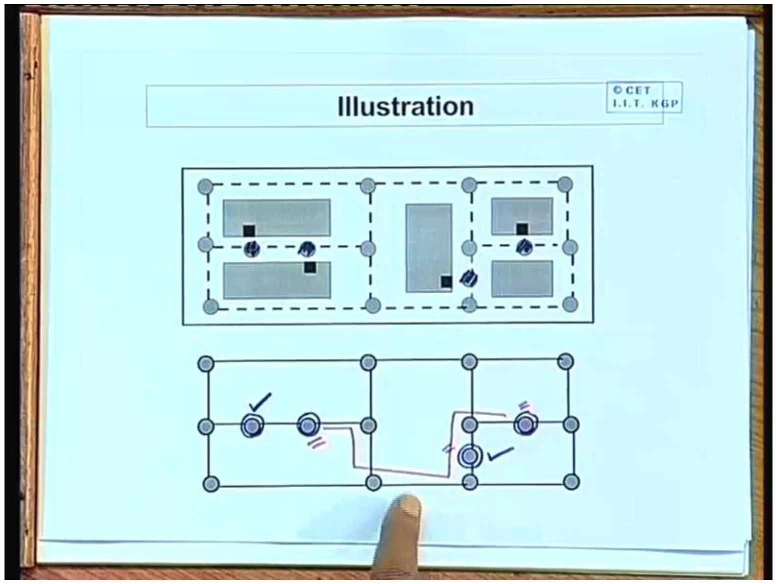
I understand your problem what you are saying is that we are considering two points at a time we are connecting them then we are considering the second set of points. Now your question is that after we connect a pair of points do we have to again change this graph because new obstacles are being put up. [Student Noise Time: 21:15] My answer is no because this is not the detailed routing you are doing you are only finding out the tentative path. But in case of lee's algorithm, for example there the detailed path was found out. But here we are just saying that we will have to pass through this channel, then this channel then this channel you just check whether the channels have sufficient capacity available or not that is all detailed routing will be taken up later ok fine.

(Refer Slide Time: 21:48)



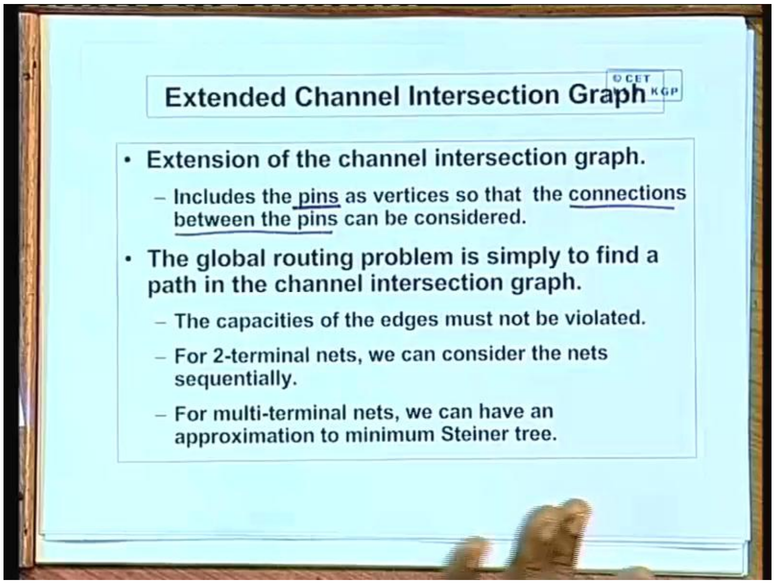
So the global routing problem in this case is very simple to find a path in the graph. Well of course the capacities on the edges must not be violated this is important. And if there are two terminal nets we can consider them sequentially one by one. But for multi terminal nets there are some approximate algorithms which are available there are some Steiner trees which are embedded on the graph. Suppose in this exa.

(Refer Slide Time: 21:48)



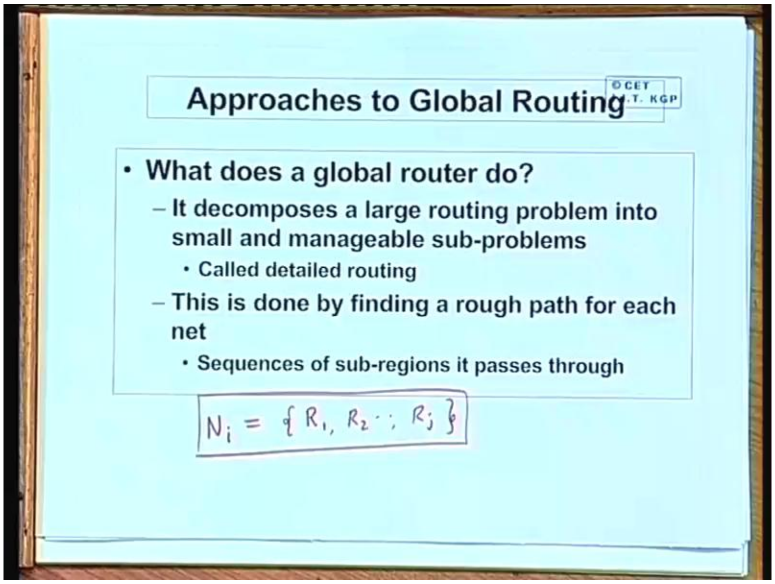
Let us take the same example suppose there is a requirement to connect these three pins say. Then there is an algorithm using which you can embed a Steiner tree say this Steiner tree will be like this. But of course in the Steiner tree you cannot let an edge go out of any other place. So although we are calling it a Steiner tree it is a path in this graph. But here what you do you do this again without violating the constraints, but the algorithms for routing this multiple nets are approximate in nature.

(Refer Slide Time: 22:57)



They will not give you the best solution in general. Because finding out the best multi terminal net solution is a hard problem. But for a pair of pins there are algorithms for doing that ok. So these are the different ways of representing. Well this already we have mentioned the basic approach to the global routing is that.

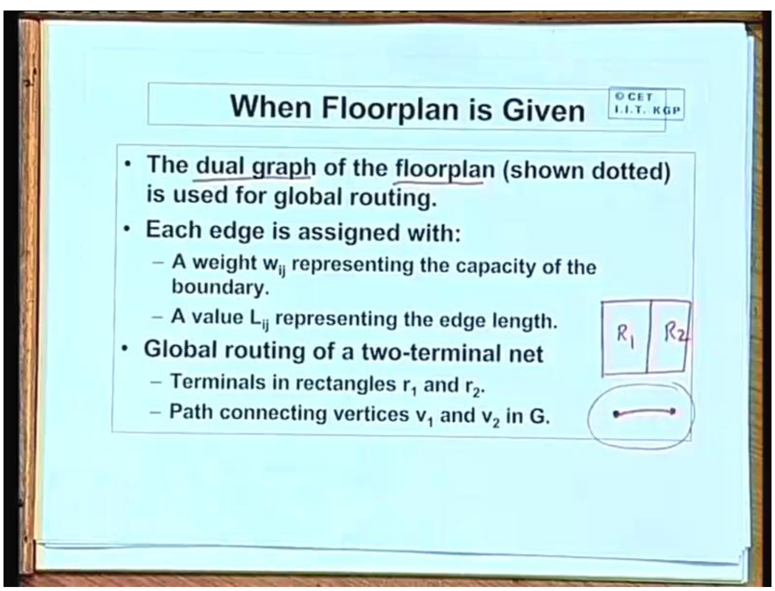
(Refer Slide Time: 23:29)



To decompose a large routing problem into small and manageable sub problems by identifying the routing regions of the channels. And this is done by finding a rough path for each net with respect to the data structures we have just introduced ok fine. So actually we are finding out a sequence of sub regions that are pass through. For example for each net Ni we will have a set of regions say R_1, R_2 to R_j. So each net will have to pass through a set of regions this will be the output of the global router. So for each net a set of such routing regions will be given.

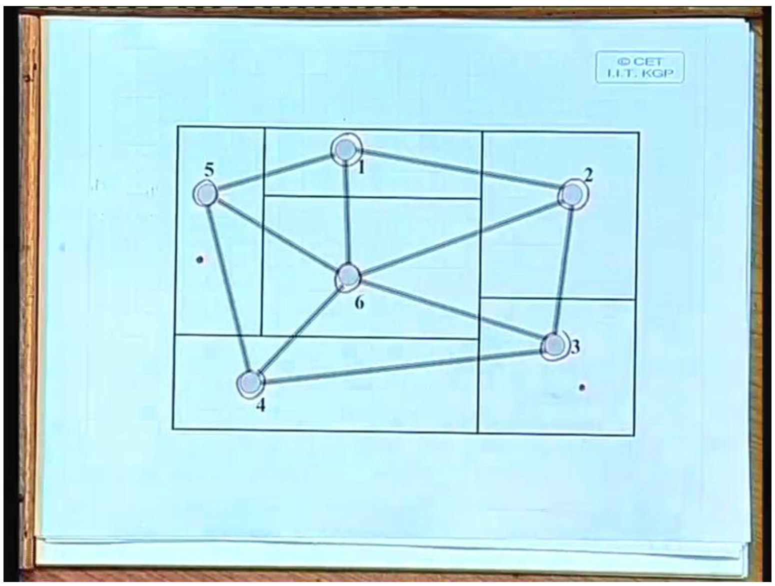
So compiling all of them the requirement for each of the regions will be fixed and after that detailed routing will be done ok. Now global routing although this is the problem. Let us look at it from a slightly different perspective. Suppose if the floor plan is given to us. Well in fact it is normally given well the floor plan may be given. But may not be the placement, but we are trying to do the global routing at the level of floor plan itself that also you can do. So although we have not done the actual placement but we have completed the floor planning. But from that floor planning information we can do the global routing. How?

(Refer Slide Time: 25:07)



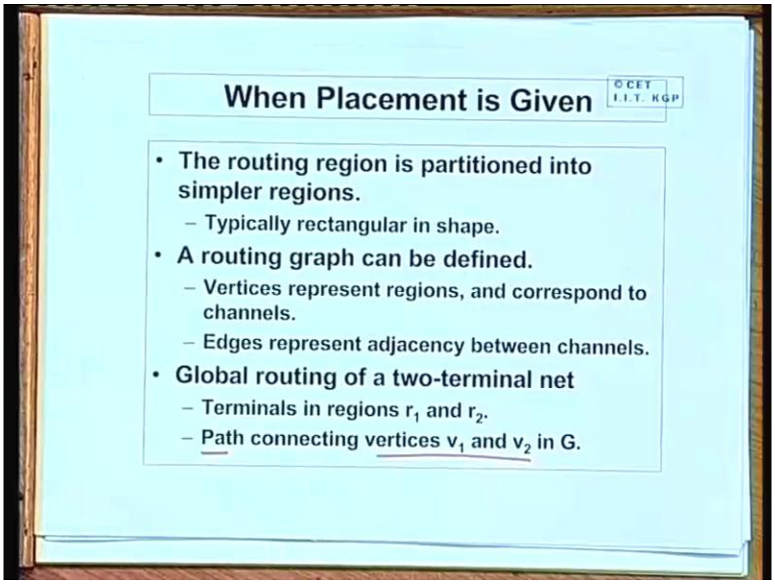
Well, have an example I will show this well you have the floor plan you consider the dual graph of it. Dual graph means suppose in the floor plan if there are two regions R_1 and R_2, then the dual graph the regions will be indicated by vertices and the boundaries will be indicated by edges. So each edge is assigned with a weight which indicates the capacity of the boundary and a value L_{ij} indicating its length these are estimates. So for global routing you will be having a graph like this. You will have to find a path between two vertices. So I am giving an example.

(Refer Slide Time: 25:56)



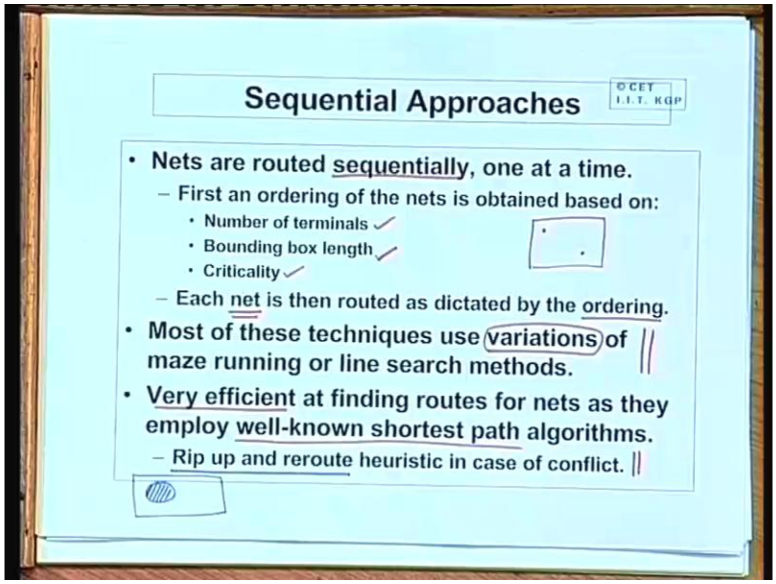
Suppose we have a floor plan like this. There are six regions 1, 2, 3, 4, 5 and 6. So I have shown the dual graph super imposed on this. These are the vertices 1, 2, 3, 4, 5 and 6. And the pair of nodes which are connected by a boundary or connected by an edge. So this is your graph. So if you want to connect a pin say from here to a pin. From here this boils down to connecting node five with node three. So in this graph you try to find out a path. Satisfying the capacity constraint and if you are also worried about the cost also satisfying the minimum cost path. So this is one approach you can do, if the only thing you have is your floor planning. So the idea is that global routing need not wait till the actual placement is done global routing you can do slightly earlier also ok.

(Refer Slide Time: 27:09)



But if the placement is given then we will do exactly what I have just mentioned. So here whatever is written here is exactly the channel intersection graph stuff. So the routing regions are partitioned into channels which are rectangular in shape you define a graph this is a channel intersection graph. And global routing problem boils down to determining the path connecting two vertices in the graph this exactly what we have mentioned ok. Now typically in global routing a typical algorithm.

(Refer Slide Time: 27:46)



They follow sequential approach. Sequential approach means nets are routed sequentially see the only known comprehensive approach which handles all the nets together. That is based on integer linear programming will also mention that problem the weight that is solved. Now in ILP in integer linear problem programming. Well the problem is solved alright but computationally it is quite intensive. So you can only solve small sub problems you cannot solve the overall problem using ILP that is difficulty. So most of the commercial routers they follow some kind of sequential approach. That is why they take nets one at a time and route them sequentially. Some heuristic is used to determine this kind of ordering. Now heuristics may be dependent on the number of terminals. Well larger the number of terminals of a net more problematic it may be to route it. So better give it a higher priority.

So if a net connects ten terminals first route that. Because later on you may see that routing that net may create a problem. So number of terminals may be one issue. Number two may be the bounding box length that means the size of the bounding box. Say if you have two pins, so you just imagine a rectangular bounding box just enclosing them. So if the bounding box size is bigger that means the net is spanning over more distance. So if the bounding box is more, more bigger give it higher priority yes. But of course some nets are critical so the critical nets can be given priority first route the critical nets using the best paths, and then use the other nets. So once you find out this kind of ordering, and then the nets are routed based on that ordering routed means I am basically I am talking about global routing for the time being. Now in global routing they use some variations of maze running or line search.

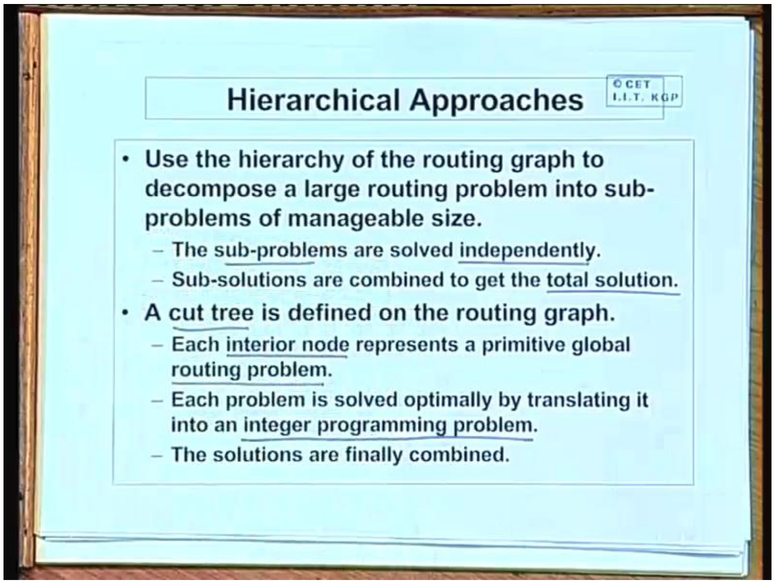
See the graph search algorithm we are talking about there also some kind of maze running or line search. Well the data structure is different. The algorithm we follow is different but conceptually they are doing something similar ok. So here when you say variations of this algorithm they may also include the graph algorithms. Because one variation which people use is that in a typical say graph algorithm for finding the minimum cost path you explode the different paths and find the absolute minimum cost path. But another criterion which is important is that you find a path where the number of bends across the channels are also minimized ok. So that may be a modification to the basic shortest path algorithm you may want to have so all those modifications are there. Now this sequential approach is very efficient because it finds roots using well known shortest path algorithms.

Normally most of these methods, some of this method not most of the some of the methods break up a multi terminal net into several two terminal nets and do them one at a time because routing of two terminal nets is easiest an optimal algorithm exists for that. But in practice you really cannot avoid say a dead end sometimes so means you just fix up some ordering you try to route them in some order. But at some point you find that your routing resources is not allowing you to proceed any further some of the nets it cannot be routed the channel capacity has been exhausted. So one good heuristic which people use and means which also works pretty well is something called rip up and reroute. Rip up means you identify on the chip area which portion is creating the maximum routing congestion. So you identify that region and you withdraw some of the already routed nets from that region.

Give them lower probability and again proceed may be those nets will find some alternate paths of routing. But the nets which are trying to route currently for them you are not finding any alternate path. So rip up and re route works well for most of the time. But if nothing works, you will have to have some manual intervention and complete the last few nets. So the automated

CAD tools also need manual intervention sometimes towards the end. [Student Noise Time: 33:38] Certainly not sequential approach will not give you the optimum. No. [Student Noise Time: 33:42] Yes but the problem is so complex you cannot do anything much better than that. People have talked about hierarchical approach. Let me very briefly talk about the hierarchical approach it says that you have a big routing problem break the routing problem into smaller sub problems.

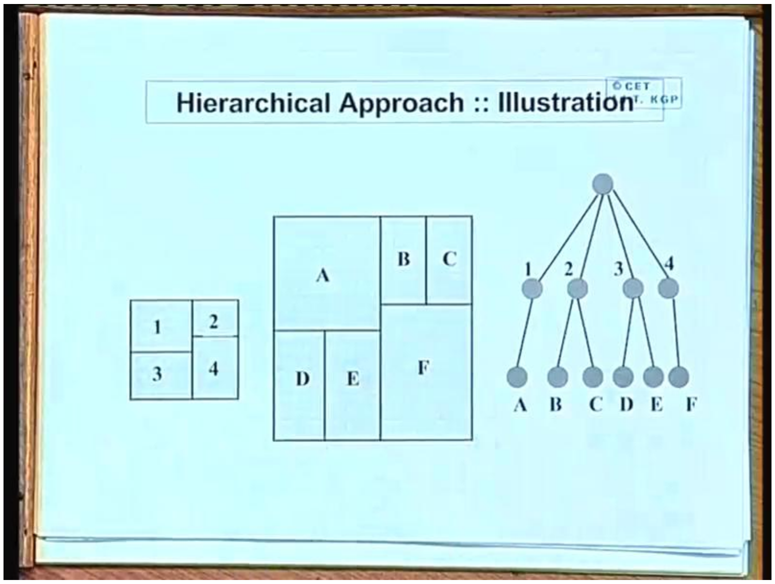
(Refer Slide Time: 34:11)



And solve the sub problems independently. Now once you have solve the sub problems independently you combine them to get the total solution. So the idea is very simple. So, on the overall routing area you define some kind of a cut tree which partitions the total routing area in to several parts and using the cut tree. Well you can define some internal nodes in the tree. So each internal node of the tree will represent a simpler routing problem. So in this case the nodes of the tree indicate the routing problem not I am partitioning the place the blocks I am partitioning the routing problem the tree represents that. Now each problem is solved.

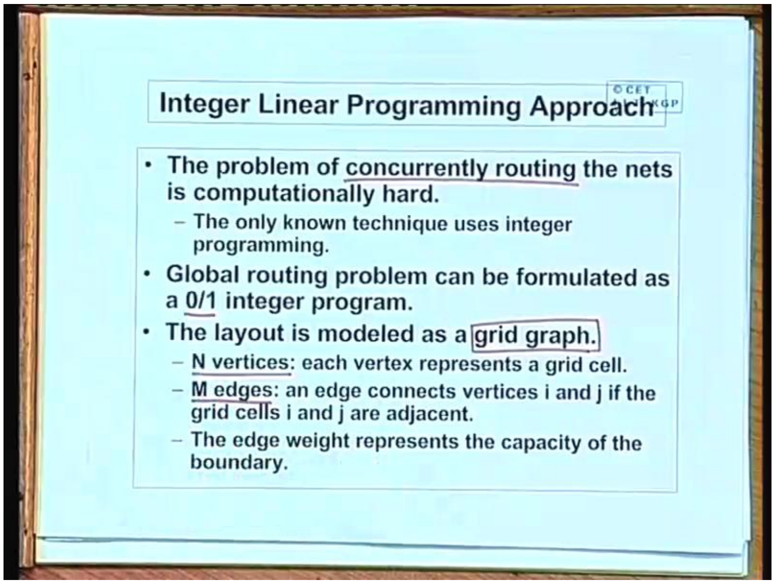
Well the idea behind hierarchical approach is that you divide the total routing problem into smaller routing problems. And you try to solve each of the smaller problems optimally. Now I have just mentioned that integer linear programming is a feasible alternative. But it works only for smaller problems. So you can feed each of the smaller problems may be to an ((word not clear: 35:32)) to get an optimal routing solution at least for that portion. The solutions are finally combined. Well of course one thing you should remember that when you partition it solve the each sub problem optimally and then combine that combination may not give you the globally optimum solution. But at least you are sure that the local routing you have done in the best possible way. So just an illustration.

(Refer Slide Time: 36:08)



Suppose this is a routing problem where this A B C E F G this may be the routing regions and the relationships some kind of. So you can globally divided up into this A is 1, BC is 2, DE is 3 and F is 4. So in this way you can go on partitioning and create a tree. So I am not going into the detail of how you do this partition but I am just trying to tell you that a big routing problem you can divide it up into smaller sub problems ok.

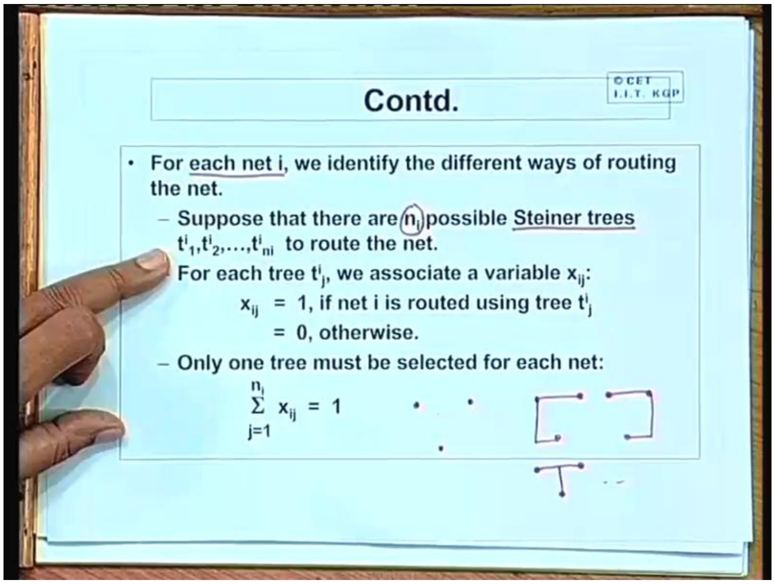
(Refer Slide Time: 36:48)



So now let us briefly have a look at how we can have an integer linear programming formulation to this problem. Because as I had mentioned this ILP is the well in fact the only one feasible approach where you can handle the problem of concurrently routing the nets. So all other approaches they take nets one at a time and route them. But integer linear programming approach since you will be expressing the total problem in terms of a set of constraints. So any solution to that will give you a feasible route for all the nets ok. So as I had mentioned this will be framed as an integer linear programming in fact zero one integer program the variables will have only zero and one values.

Now certain assumptions are here number one is that you are modeling this as a grid graph. That is why the size of the graph is pretty large. So if it is a big problem you cannot handle this. So in the grid graph there are N number of vertices where if you recall the vertices represent the grid cells. There are M numbers of edges so an edge will be connecting to vertices if the cells are adjacent. The edge weight represents the capacity of the boundary. So this may a grid graph or this can also be a checker board model a coarse grid. So it may not be a fine grid just can also be a coarse grid ok. So the graph has N number of vertices and M number of edges.

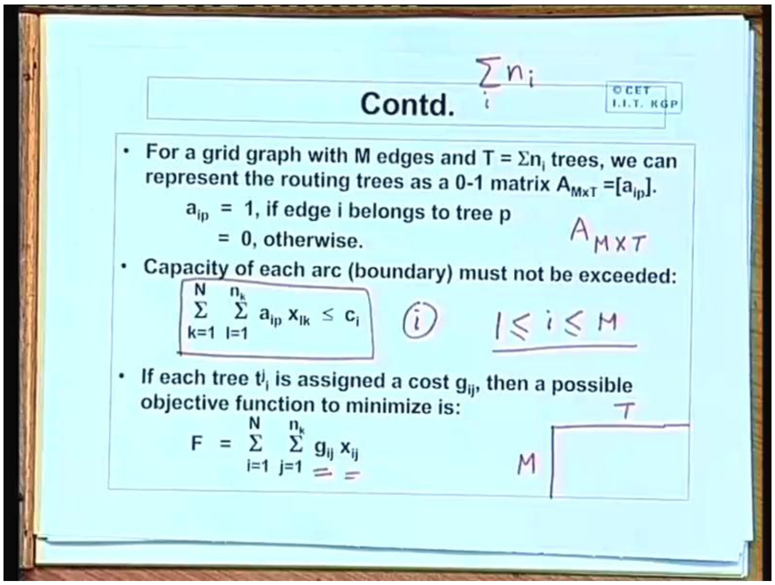
(Refer Slide Time: 38:43)



Now the idea is this. For each net i, suppose net is says I have to connect these three pins. I can say that well there are several different ways of connecting them. I can connect them like this. I can connect them like this. I can connect them like this and so on there may be a number of other methods also. So for each net I we identify the different ways of routing the net different ways means actually we are finding some Steiner tree. Some possible Steiner trees for routing the net. See if the numbers of terminals of the net are large then the number of such trees will also be very large. So actually we will be restricting the number of trees to some feasible number may be there see for each net we will take five not more than five. So for each net I we assume that there are ni possible Steiner trees we are considering. And those Steiner trees we are calling ti 1, ti 2 up to tini.

These are the different alternate trees that are possible for net I. And for each of these nets, we associate a variable xij where i indicates net number and j indicates the corresponding tree number. Xij will be a binary variable 0 or 1, 1 means I have selected tree number j to route net i zero otherwise. So out of these ni trees one of the x values will be one the other all will be zero. Just to indicate that only one of these where choosing, so we can write it like this sum total of this summed over all j’s will be one. So for each I exactly one of the xij value will be one other all will be zero, so that there sum will be one right. So out of these we are selecting only one others we are not selecting. Now xij indicates that we are selecting one others we are not selecting ok fine.

(Refer Slide Time: 41:28)

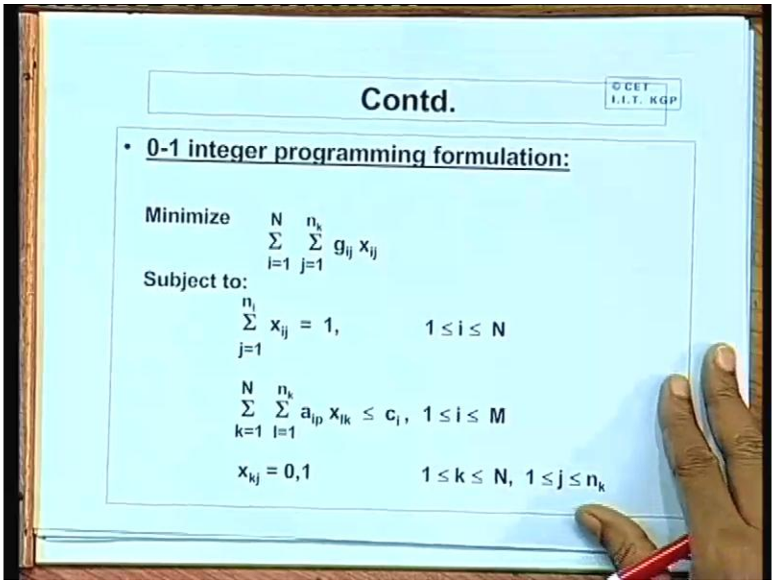


Then well we have a grid graph with M edges and see there are many trees for net I there are ni number of trees right. Now if we sum this over I we will get the total number of trees for all the nets t is that number sigma of ni. So we define a matrix A. A is a matrix which as you can see as a dimension M cross T. So on one side we have the edges of the graph and the other side we have the trees. So actually this matrix will contain which Steiner tree actually uses which edges. So the corresponding aip will be one if edge I belongs to tree p zero. Otherwise so actually the matrix will indicate that. Now the constraints will be put like this say I had mentioned that each arc boundary will be having a constraint.

Capacity of each arc must not be exceeded. Arc means well we are considering a grid graph nodes at the regions and edges are the boundaries of the regions. So arcs will indicate the channel cap the capacity of the boundaries that means how many wires you can route through that boundary. So this you can express like this if you just look at this it will be clear that how we are doing. We are multiplying this matrix with x so x will indicate actually which of the trees we have really taken the others will be made all zeros. So that sum total must be less than equal to say for a given I for net I it must be less than equal to ci.

So this will be true for all the given boundaries I will be showing it in next slide. [Student Noise Time: 44:00] Boundary, boundary yeah, yeah. This in this case I will range from one to M. M is the number of edges in the graph. For each edge the capacity should not cross ci is the capacity of edge I ok. And in terms of the objective function each if each tree is assigned a cost gij well the cost you can assign using a number of ways regarding number of bends you have etc. So there are number of different ways of estimating the goodness of a tree. So if you multiply that with xi's and makes a sum total that will give you the cost function just the way of estimating. So the total problem you can frame like this.

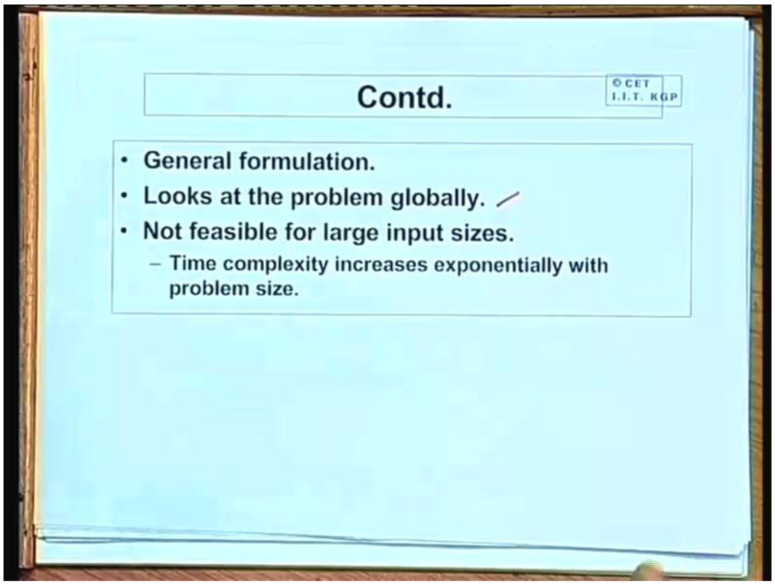
(Refer Slide Time: 44:45)



You minimize this cost function subject to these few constraints one is that sigma of xij must be one and that capacity constraint and each x must be either zero or one. So if you solve this problem then you will be getting the values of this variables xij. [Student Noise Time: 45:07] This one, [Student Noise Time: 45:13] this one see this xij. Well yeah. This xij is what xij indicates that whether we have that means for net i which tree you have taken jth tree. So we are summing over all j's out of the one of the xij will be one. Rest will be zeros and this sigma one to N. N is the nodes you are summing it over all the nodes and each of the trees will be having an associated cost.

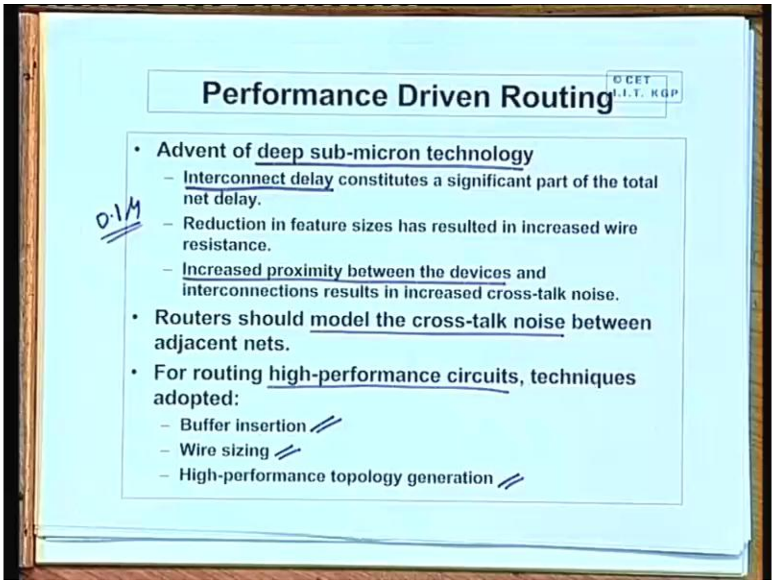
Basic idea is that suppose there are 10 nets. Finally I will be selecting 10 trees each of the trees will be assigned a weight I make a weighted sum of all of them this is this expression. This xij only one of them will be one and the corresponding gij will indicate the goodness or the weight of that solution. This will be summed over all the nets all the fact I equal to one to N all the nodes of the graph. [Student Noise Time: 46:31] Yes [Student Noise Time: 46:33] This one [Student Noise Time: 46:38] This a_{ij}, yeah. This will be ai p p have not yeah. Right, Right, yes. P have not defined ok. [Student Noise Time: 46:57] Yes, yes. Yeah, you are right. So this is how you can formulate it and you can solve it.

(Refer Slide Time: 47:16)



Now the advantage of the ILP formulation is that we are looking at the problem globally and we are trying out trying to get a solution parallel. But as I just mentioned this is not feasible solution for large input sizes. So only for small sub problems you can use this to get an optimal solution or a very good solution ok.

(Refer Slide Time: 47:42)

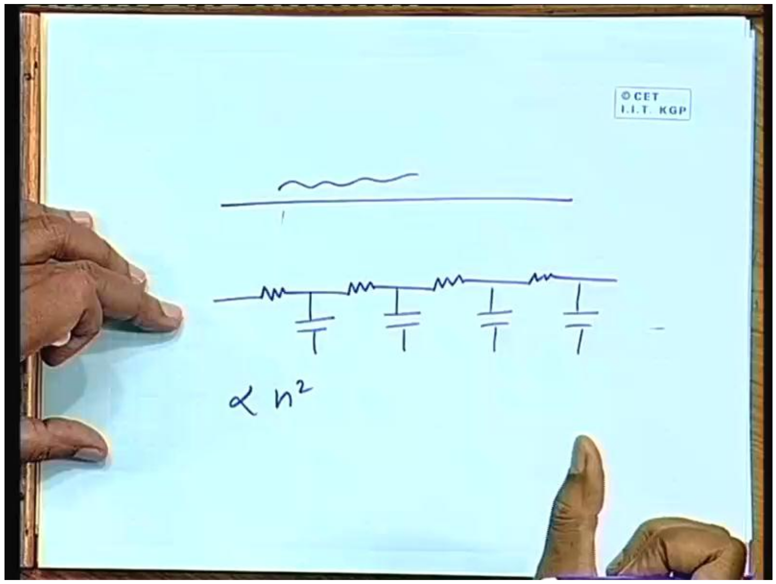


Now let me just also mention something about performance driven routing where which people are talking about nowadays. Now this as become important with the advent of the modern deep sub micron technology which we have now we have technologies in the tune of point one micron people are talking about. So in this kind of a technology that the basic feature sizes of the devices and the interconnections they became very small and are spaced very close together. Now interconnect delay becomes important as well in fact it becomes as important as the component delays in this kind of feature sizes. So if you are not careful the interconnect delays will finally constitute you are, we can say the total overhead in terms of performance and your final chip performance will depend on the interconnection overheads. So the significant part of the total net delay will be your interconnection delay.

So your main criteria here will be to route the nets in a very proper way. So increased proximity between the devices also increases this problem. So the routers should not only just merely model the interconnections. But also should model the cross talk noise between adjacent nets. So if you model the cross talk noise only then you can give a good estimate regarding the delays. Suppose you say this is a critical path so the critical path has to be routed in a proper way so that the delay criteria are satisfied. So the router cannot keep a blind eye to the routing sub problem. The router must also keep an estimate or keep a tag on actually how much delay we are really incurring when you are doing this routing. So for high performance circuit the routers also have to do a number of other things like buffer insertion in order to reduce the interconnection delay. Periodically you can, you will have to insert some buffers to again regenerate the signal.

Because if it is a long wire the capacity of load on it will be much larger. But if you insert a buffer in between the total delay will increase will decrease. Not only that, wire sizing. Now some of the wires may be thinner some of the wires may be thicker depending on the criticality and other issues. All wires of the same size that assumptions no longer hold here and again high performance topology generation which is similar to global routing sub problem. This is a problem of this. You can see this is an open problem of research more or less people are working on it that given some delay and performance constraints how to carry out the global routing. Well global routing cannot be as abstract as it used to be in case of high performance design. Global routing should also very accurately model the cross talk noise and interconnect delays. So as to give a fairly good estimate means regarding the quality of the solution. So actually [Student Noise Time: 51:14] Simple suppose you have well let me briefly talk about it.

(Refer Slide Time: 51:28)



say you have a transmission line. So on this transmission line resistive and capacitive delays are all spread out. Now this are often modeled as distributed resistors and capacitors like this. Now it can be proved by analyzing this that the approximate delay through this is proportional to the square of the number of such rc stages. If we insert a buffer in between this n square will become n by two whole square for each of them. So the total delay of the circuit will become less [Student Noise Time: 52:09] two parts will become independent plus the constant delay of that buffer that will be less than this n square delay. Ok. So in our next class we will be starting our discussion on detailed routing. So now we will see that how this individual regions can be routed one by one ok. Thank you.