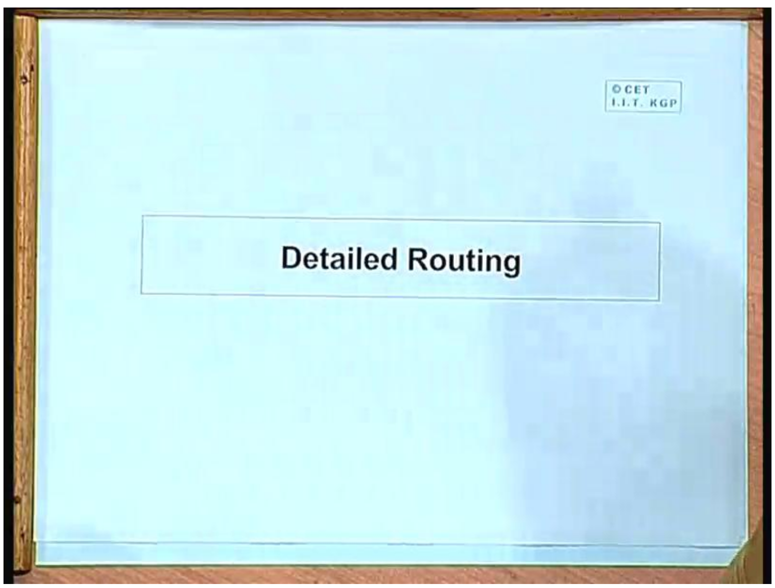
Electronic Design Automation
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture No #25
Backend Design: Part – XI

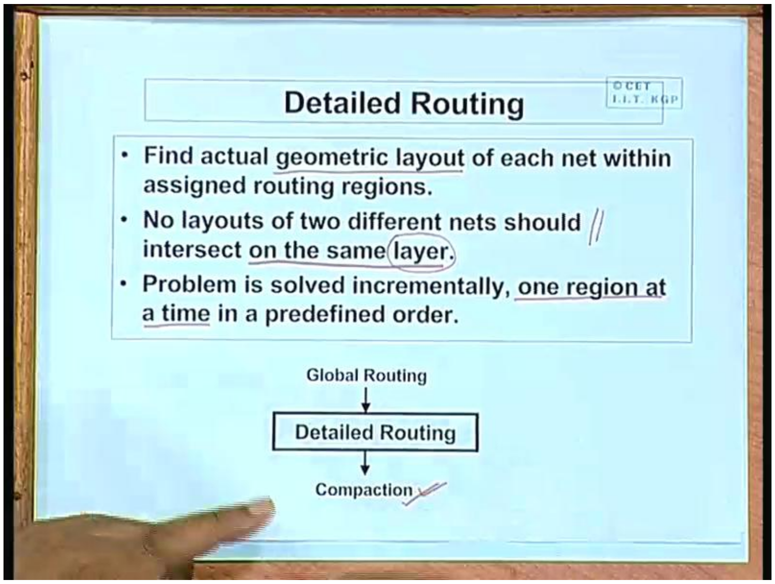
In the last lecture we were talking about the process of global routing. Now we recall during global routing, we are trying, we have tried to find out the rough roughly the sequence of regions through which each of the net has to flow or pass through. And we are also mentioned that subsequent of global routing we have to take up the routing regions 1 at a time and to complete the routing in each of them. Now this phase of completing the routing individually in each of the regions is called detailed routing.

(Refer Slide Time: 01:47)



And in this class we shall start our discussion on detailed routing. First we shall be looking at what the problem is all about and what are the different kinds of you can say representations we have for tackling or handling this problem. Or subsequently we shall be looking at the actual algorithm which people have proposed to solve this problem. Firstly just about the problem what is detailed routing.

(Refer Slide Time: 02:15)

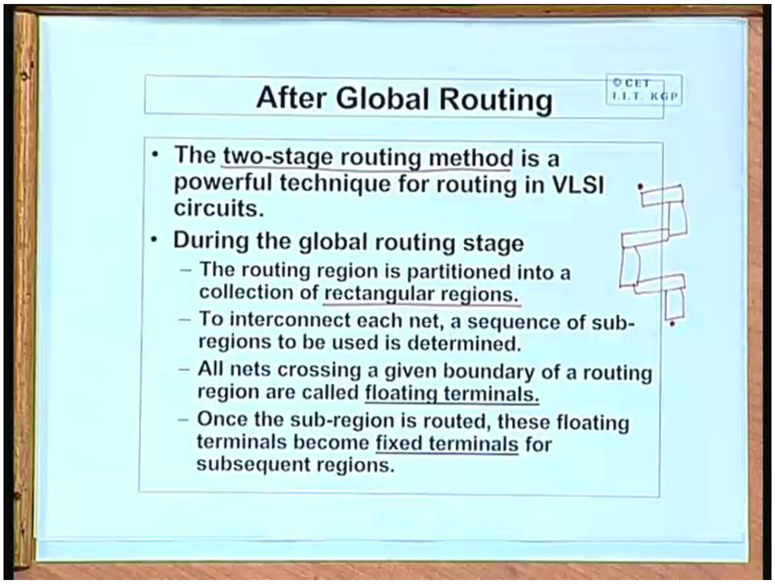


So this I have mentioned. So detailed routing consists of finding the actual geometric layout paths. So you try to find out the actual path on the surface of the floor through which each net will be passing through. And obviously an obvious constraint is that 2 different nets must not intersect on the same layer. Well here a term called layer is coming. Now layer is coming due to the fact that well in a typical VLSI chip the interconnection is not d 1 on a single layer. There are often several layers available to you and you can complete the interconnection by laying out wires on 1 or more of those layers.

Typically at least 2 of those layers are metal layers and the other layer maybe diffusion or polysilicon layer. So on this layers you can complete the interconnection is obviously on a single layer the wires of 2 different nets must not intersect each other otherwise there we short circuit obviously. And as we had said that for detailed routing we consider 1 routing region at a time. So the problem we are solving in an incremental fashion and the ordering of the regions were mentioned something about that. There is a ordering which is created and the regions or you can say routed or handled in that particular order. So to look at this picture this process of detailed routing starts after global routing and after this detailed routing is done.

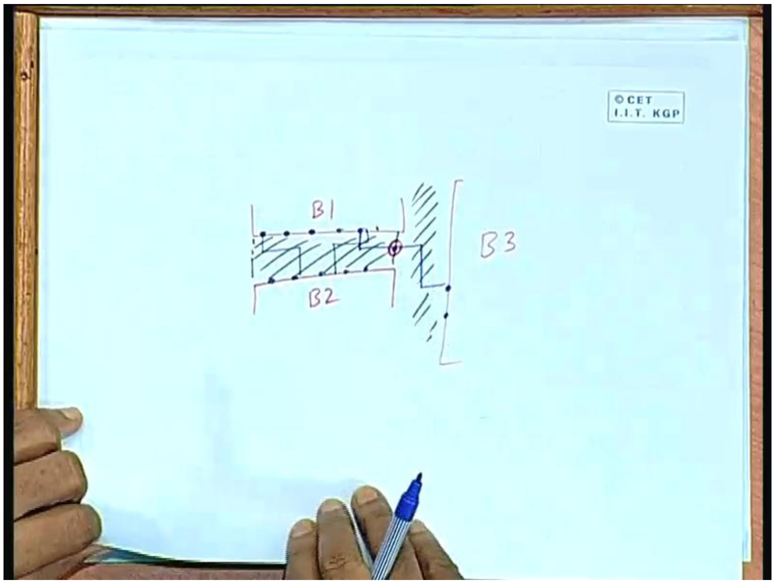
So we have found out the layout of all the wires. Now it is often the case that after placement the area you had kept aside for interconnection. Well area was too much you really do not require all of that area. So you will find if you have if we if we look at a snapshot of the floor plan after routing is complete you will feel that there are some areas where there is some empty space still. So often after detailed routing a process called compaction is carried out. Compaction tries to squeeze the layout as a final optimization phase to reduce the area as much as possible. So this is algorithms of compaction we shall see later after we finish our discussion on detailed routing.

(Refer Slide Time: 05:02)



So global routing followed by detailed routing this is a 2 stage routing process or method ok. Now during the global routing stage you recall once again that the overall routing area that is available on the chip is partitioned or divide into a set of rectangular regions. These rectangular regions are the individual routing regions which we typically call a channel or a switch box. Now to interconnect each net we will have to find out a sequence of such regions. For example if I want to connect a point here to a point here we will have to find out what are the different regions through which this net has to go through ok. There maybe a sequence of such regions ok. Now, one thing to notice that when we are considering global routing, we are only talking about a tentative path. But suppose when we are completing detailed routing or when you are going about the process of detailed routing suppose you have a scenario like this.

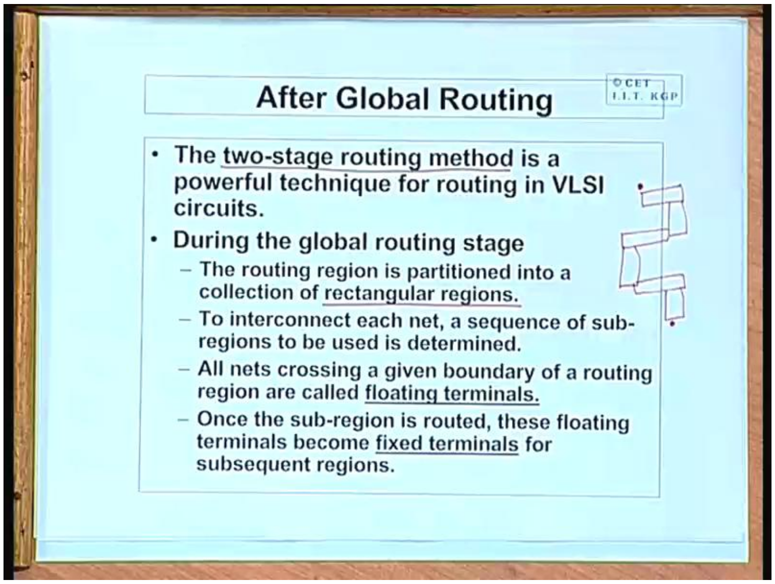
(Refer Slide Time: 06:15)



You have a block out here say b1 you have a block out here b2. So this will be a routing region if you look at this is a routing region. Now on the other side there may be other routing. For example there is another block b3 on this side. So there will be a routing region like this also on the other side. Suppose we are considering the problem of completing the route in this region first. So we will be trying to interconnect the points whatever they are. Now it is not necessarily true that all of these interconnections will connect points of b1 to points of b2 only. There can be some nets which will connect b1 with some point of b3 also. So those nets will possibly be routed like this.

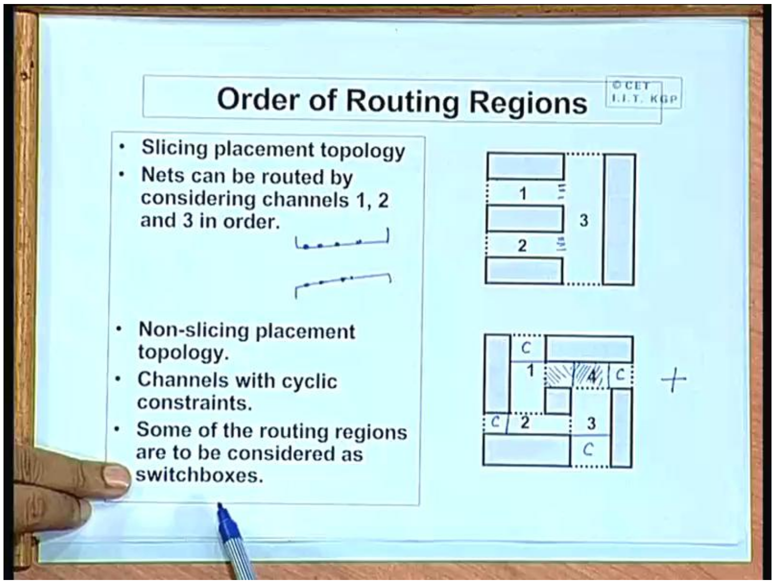
They will have to cross this region and enter the other region. Now the point to notice that when we are starting the problem of routing this particular region you look at this point where this wire is crossing the boundary. So the exact location of this point is not fixed. This wire we can layout on the first track on the second track on the third track. So initially this terminal is called a floating terminal with respect to the second region. So for the second region the location of this terminal is not fixed. But once this region the routing of this is completed the location of this point becomes fixed. So now this becomes more like a fixed terminal with respect to this region. So the concept of floating terminal and fixed terminals comes into the picture like this.

(Refer Slide Time: 08:25)



So what I have said is that all the nets which are crossing a given boundary they are called floating terminals because their exact locations are not fixed till you are completing the routing. So once you have completed routing that particular region these floating terminals their exact locations will get fixed and they will now be termed as fixed terminals. For routing the subsequent regions these will be considered as fixed terminals with fixed co-ordinates ok fine. So the ordering of the channel is important due to this because some of the floating terminals will become fixed terminals and will make the subsequent routing problems easier. If they are fixed then the conventional channel routers can be used to route them ok.

(Refer Slide Time: 09:20)

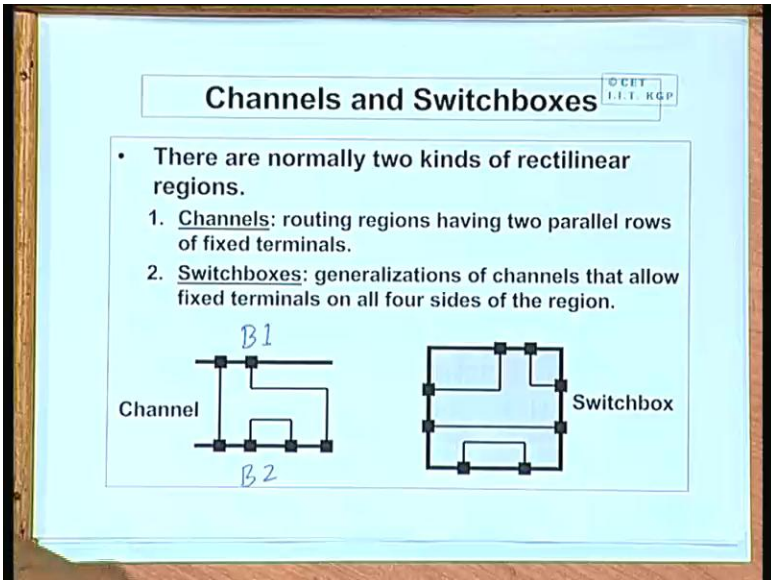


So we had talked about this earlier also but again let us look at a couple of examples related to the order of the routing regions. The first example we are taking a slicing placement topology which can be divided by horizontal and vertical slices. Now in this case we can route the nets in the order 1, 2, followed by 3. See here our assumption is that we are using a channel router because I mentioned earlier a channel router is easier to design and implement as compared to a switch box router. So a channel router will be having terminals on top and bottom with a free space in between for interconnection. So you look at 1 is a channel with pins on top or bottom 2 is a channel with possibly some floating terminals on the right. So once we route 1 and 2 this floating terminals will become fixed terminals. Then you can route this 3 because 3 now becomes a true channel all the terminals are fixed.

But you cannot do it the other way around you cannot start with 3 because these terminals you do not know their location ok. So you cannot start with 3 you will have to start with 1 and 2 and then we will have to handle 3 yes. [Students Noise Time: 10:50] No, the point is that the way channel routing programs work well you can extend some pins to the right or left. But you cannot flexibly change the points on the top or bottom. So if you had taken this region 3 first then these points their locations are not fixed. So you cannot use any of the conventional channel routing algorithms to route this before that we will have to fix this points ok. But if you have a non slicing topology like this a wheel then you can see that the definition of channels is not that straight forward. Like in this example for instance well this is a channel if you divide this portion there are pins here pins here and out here there will be floating terminals.

Similarly this small thing will be a channel this will be a channel and this also will be a channel. But if you look at the other regions say for example you think of this 1 say this region. There will be some regions like this 1 where you can find that there are terminals on all four sides. Similarly this, there are terminals on all four sides. There will be several regions which will be acting as switch boxes. So if we have a non slicing placement topology as another example you have taken earlier the plus shaped junctions. So for those situations you cannot avoid using a switch box router. Because there will be some regions which will be inherently there will be switch boxes and you will have to use a switch box router which is more complex as you will see ok fine. [Students Noise Time: 13:00] Well here normally we will be completing the channels first that the other switch boxes you can do it any 1 ok. So just coming back to channels and switch boxes once again.

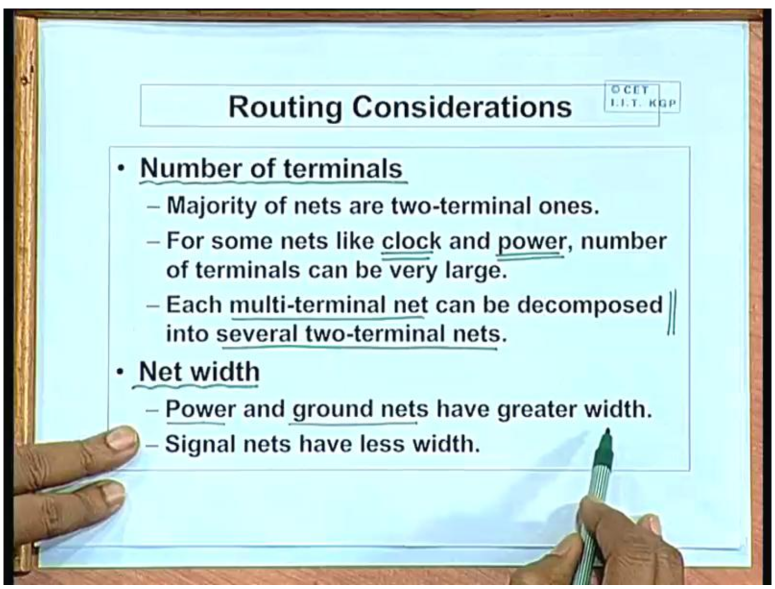
(Refer Slide Time: 13:20)



Well earlier we had categorized channels as horizontal and vertical. But well you can just without loss of generality you can consider all channels as horizontal you can always rotate it by ninety degree and consider the other way around. So a channel is a routing region which is having 2 parallel rows of fixed terminal on top and bottom like this diagram shows. These are the 2 boundaries there is a block say b1 here, there is a block b2 here. These are the terminals which need to be interconnected this is a possible interconnection. So this space in between is kept aside for interconnection there are no obstacles out there.

So a channel is an obstruction less region which is used for routing. And a switch box looks like this where pins maybe located on all four sides of the boundary this is a typical interconnection. This interconnection can be d 1 on a single layer on multiple layers we will see the different approaches later. Before we talk about the routing algorithms and the and well the problem of detailed routing in general first let us look at some of the issues involved. Well when we talk about routing there are a number of different things, we need to look at.

(Refer Slide Time: 14:51)

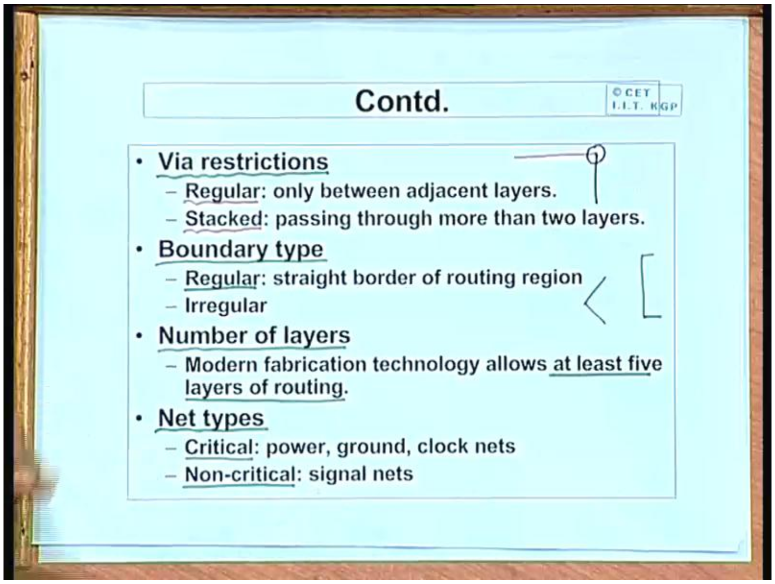


First of course is the number of terminals corresponding to the nets. Now it is true in general that most of the nets are 2 terminal nets. But of course there are nets like clock power some kind of a signal which is going to many different places where the number of terminals can be large. Typically clock and the power nets due to their very high cardinality with respect to the number of terminals they are handled in a different way. So whatever algorithms we are discussing now, they do not apply to clock and power routing they will be handled differently right. But many routing algorithm will see in fact most of the channel routing algorithm we will be discussing.

They either handle in 2 terminal nets or a multi terminal net you can breakup into 2 terminal ones. But some of them can handle multi terminals also. So if necessary the multi terminal nets can be decomposed into several 2 terminal nets. So this may be necessary this is not required for all algorithms, but for some it may be necessary. Second issue is regarding net width. Well we are mentioned in last class that high performance circuits some of the signal lines maybe wider than the others but we are not considering that issue right now. Well even without those constraints for high performance designs. Normally the power and the ground nets have greater width as compared to signal nets, greater width because they will have to carry more current.

Where the thing is that greater width is fine. But the width of the power and ground nets need not be uniform all throughout the chips. The point for it enters the chip it is the widest out there and from there it is getting distributed. So each branch of distribution will be thinner than the main trunk and from their again they are getting distributed. So wires are getting even thinner. So the different segments of for the route will be having different widths depending on the estimated amount of current it is expected to carry right fine. There are some other issues.

(Refer Slide Time: 17:32)



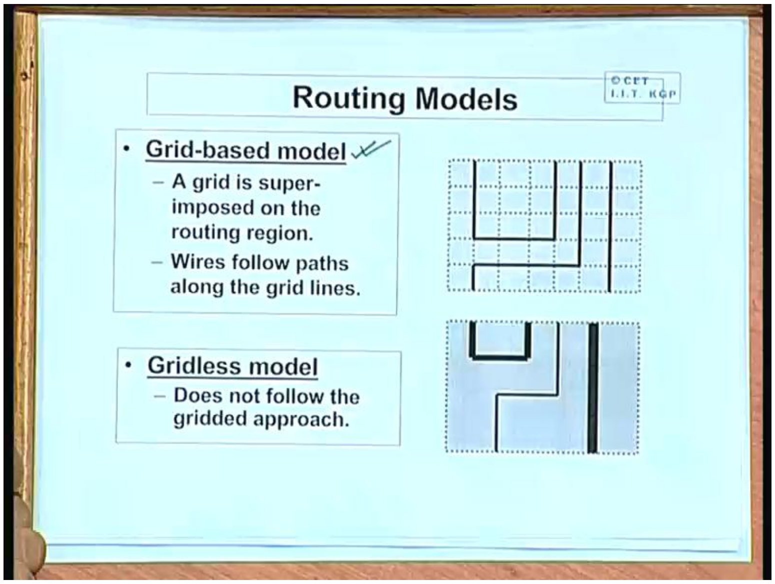
Well there is something called via connections. Well we had mentioned that we have typically more than 1 layer available for interconnection. Suppose we have layer 1 and layer two. There is a wire a segment of the wire which is running on layer 1 and a segment of the same wire net running on layer two. So in order to complete the connection we have to make a connection between these 2 points. So what we are saying is that if a portion of the net is running on layer 1 and another portion of net is running on layer 2 I am showing in different colors. Then at the junction there has to be some kind of a connection made. This is typically d 1 in VLSI fabrication technology by drilling a hole and putting some kind of metallization to make the connection. This is called a via connection.

Via connections are essential if you are using a multilayer router but reducing the number of via connection is also desirable. Because more the number of via connection more will be the area taken and less will be the reliability. So as you connect across layers you reliability and the performance also slightly degrade. And regarding the via there can be 2 kinds. 1 is the regular types of via connection where you only connect between adjacent layers. But you can have a stacked via also where 2 layers are getting connected where there is more than 1 layer in between. So it is a very deep interconnection. So stacked via are more difficult to implement in terms of the fabrication technology and it takes more area to fabricate right. So these are the 2 different kinds of via.

Suppose you are you want to connect a metal like over the sorry diffusion of the polysilicon lying. Then you may have to drill down deeper to get access to the layer beneath down below. So that will be a stacked via. Well there is another issue regarding the boundary type. Boundary type is the boundaries of the blocks. Well typically we have regular boundary types but the borders are all straight lines. But in general it can be lines which are oriented at any arbitrary angles. There are some tool cat tools which allow boundaries at arbitrary angles other than 90 degrees vertical and horizontal. Number of layers is an issue. Before you do the routing you must know that how many layers are available to you. Modern the fabrication technology typically allows at least 5 layers.

Well out of these 5, typically 2 of them are left aside for power ground and clock. So you have 2 or 3 layer available to you. Net types you can categorize the net types. Some of the nets are critical. Power ground clock are absolutely critical. In addition there can be some signal nets also which you can categorize in the critical category. For the non-critical nets are the other ones where well you have some flexibility. There you can you can find a longer route if the need arises for that. So the slight increase in delay is not important for those. Ok and there is another issue this is regarding the routing models. Now most of the algorithm that we will be discussing they will be based on the so called grid based model.

(Refer Slide Time: 21:30)



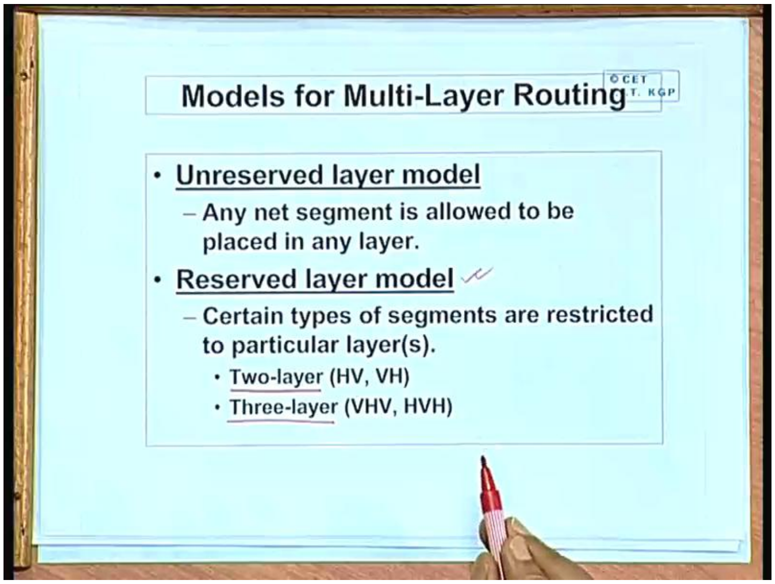
Grid based model says that you have an imaginary matrix of you can say lines horizontal and vertical. So all the interconnecting lines will pass along the junction of these rows and columns. So you cannot have a line from any arbitrary location from this middle you cannot have a line like this. So the lines will be all of this same width and they will have to pass just exactly along the boundaries. This is the grid based model. So actually for signal routing we typically use the gird based model. Then when we talk about a channel we typically talk about the number of rows available the number of columns.

So implicitly we are assuming the grid model. But when we are say routing power. So we are not interested in the exact places from where the lines routed it can be anywhere and the width can be different also. Because typically there is a different metal layer which is available to us for routing the power so we have complete flexibility out there. This is the so called gridless model. So it need not follow the boundaries of the grids and the widths also maybe different right ok. Yes [Students Noise Time: 23:03] Lees algorithms is different there the gird that the interpretational grid was different.

That the whole layout area was divided up into grid cells were each grid cell was considered smallest entity you can talk of suppose that terminal. So a terminal will belong to 1 grid cell. [Students Noise Time: 23:27] Yes [Students Noise Time: 23:30] Yes, yes. [Students Noise Time: 23:35] Yeah, you have seen that in the lees algorithm the way the algorithm we had talked about were we had considered there that everything was laid out on a single layer. But you can easily extend lees algorithm in a 2 dimensional plane we can consider that there are 2 planes 1 on top of the other that corresponds to the 2 different layers.

So you can extend lees algorithm to 2 layer model also. So that is a not an issue but usually lees algorithm is used when some means as you can say as the last resort. When you have you have laid out most of the nets using some other method. Some of the nets you are unable to find a path then we use the lees algorithm or the line search algorithm to find out a path. Now if there are multiple layers available we will have to extend that basic lees model. [Students Noise Time: 24:36] Yes, yes. That model is for a single layer ok. So when you talk about multilayer routing.

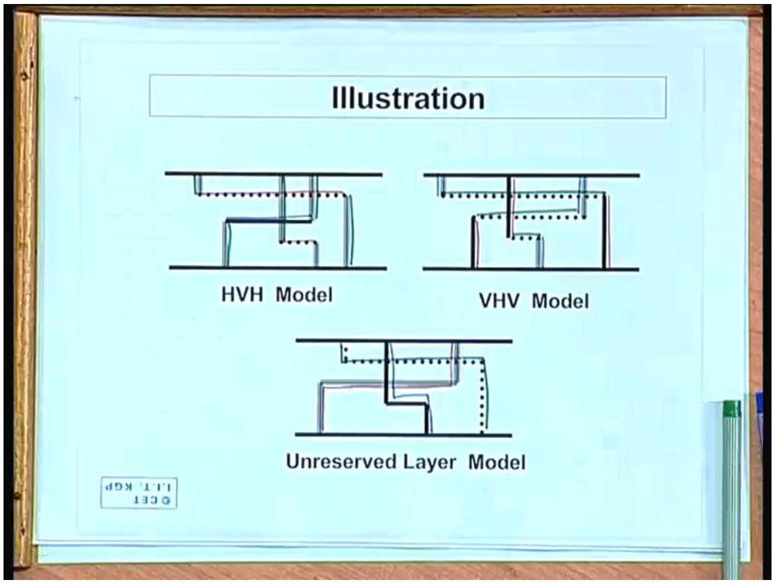
(Refer Slide Time: 24:50)



There are broadly 2 approaches. First says that there is no restriction as such this is called unreserved layer model. This unreserved layer model says, that suppose you have 2 layers of routing. So you can layout wires in any fashion on the 2 layers. Of course if it is a gridded model you will have to lay it in synchronism with the grid rows and columns. But other than that there are no restrictions. This is the unreserved layer model. But in the reserved layer model it says something like this that well in layer 1 you can only lay horizontal tracks on layer 2 you can only lay vertical tracks. Well if you make some restrictions like this your algorithms for routing becomes much easier. Moreover say in adjacent layers if we have 2 parallel tracks running over a long distance the capacitance will be more.

And the delay of the signals and the noise will be more. But since they are crossing across adjacent layers, the effect of noise and the inter layer capacitance is also much less. So typically we use the reserved layer model unless the problem is simple enough to be routed in a single layer. Reserved layer model says that well you have some restriction as for the horizontal and vertical tracks. So if it is a 2 layer routing problem then we normally go for HV or VH. H means in layer 1 only horizontal tracks in layer 2 vertical tracks or the reverse. If it is a 3 layer routing problem you can go for VHV or HVH right. So some simple examples.

(Refer Slide Time: 26:48)

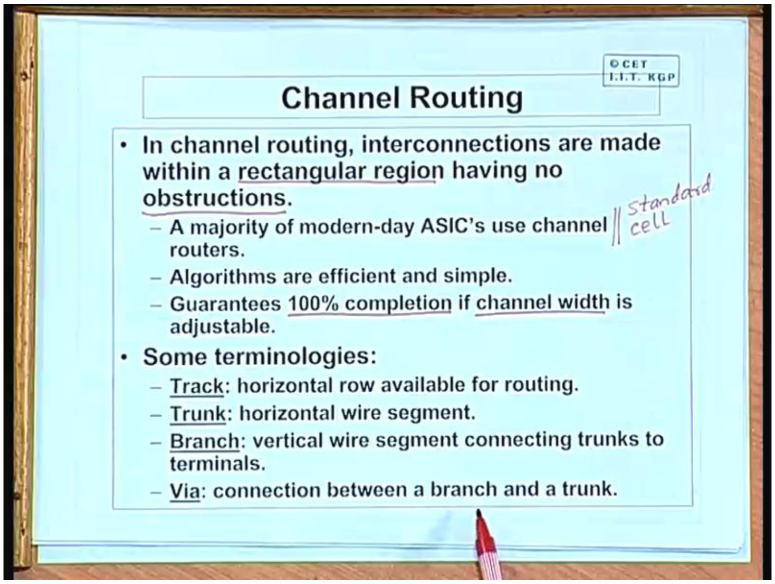


Say this shows a typical illustration for the HVH model. Well I am showing this by color. This is say the first la layer one. Layer 1 is H, layer 2 is this this is layer two. Layer 2 is vertical and say this is layer three. Layer 1 and layer 3 will be horizontal layer 2 will be vertical. So this is a typical example. Similarly VHV model. So the 2 layers which will carry vertical lines see this is 1 this is the other and you can have 1 horizontal line. This example shows an unreserved layer model where we have 3 layers but there is no restriction. You can have 1 layer which can carry both horizontal and vertical line segments. You can have 1 which can carry both horizontal and vertical line segments.

Similarly the third layer can be like this. Typically when you are routing the power and ground that usually follows the unreserved layer model. Because in power and ground since you are trying to carry high current you do not want the wire connections. Because wire connections are the sources of additional impedances there can be voltage drops here also. Now he here 1 thing you absorb here that every time you move from 1 layer to the another. For example here these are all wire connections. So the number of wire connections is also an issue. Some of the algorithms you will see they ignore the wire connections. But there are some algorithms which also take these into account and they try to minimize that ok right.

So now let us come to the channel routing problem and see how people will have try to approach this. So as I have mentioned that in channel routing you have a rectangular region called the channel which is left for interconnections and that region has does not have any obstructions. And these channel routing algorithms which we will talk about they are unlike the grid routing algorithms like lees square, we route the nets 1 at a time and once we route a net it becomes an obstacle for feature nets. So it is not like that we are trying to find out a global solution using an algorithm. So we are considering all the nets at a time.

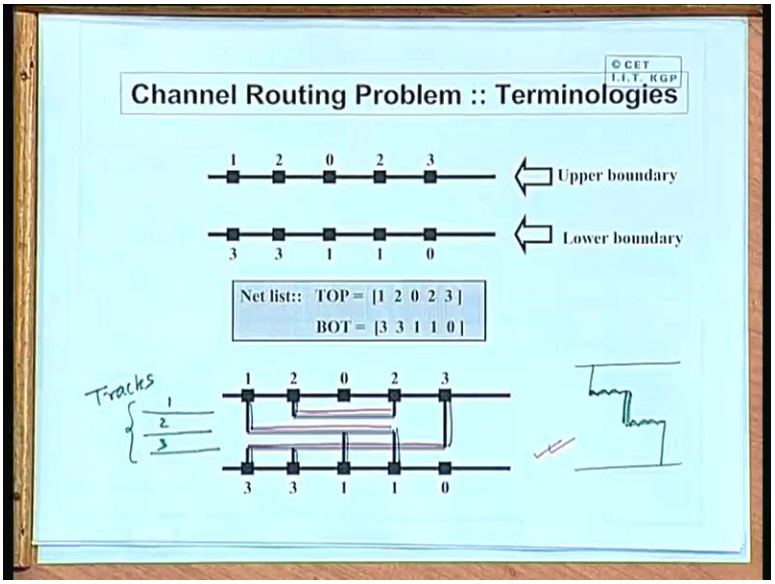
(Refer Slide Time: 30:01)



Channel routing was important because ok. Most of the ASIC's we design nowadays they use channel router because most of them use the standard cell design style and standard cells implicitly they use channels. The algorithms for channel routing are efficient and you will see that there also not very difficult to implement. And the most important thing is that if we have a channel with flexible channel width, then 100 percent routing completion is guaranteed. Well by channel width ok you can mean both ways either the channel is wider or it is or the channel height is more. So you can adjust the channel size in in any direction you want to.

But if we have this flexibility you can guarantee hundred percent completion. Well for channels there are some terminologies. We define something called tracks in the channel. These are the horizontal rows which are available for routing. Trunk is a horizontal wire segment corresponding to a particular net which is laid out on some track. Branch is the other way around branches are the vertical wire segments. They connect the trunks to the terminals and of course via is a connection between a branch and a trunk I am taking an example say I have an example out here.

(Refer Slide Time: 31:43)

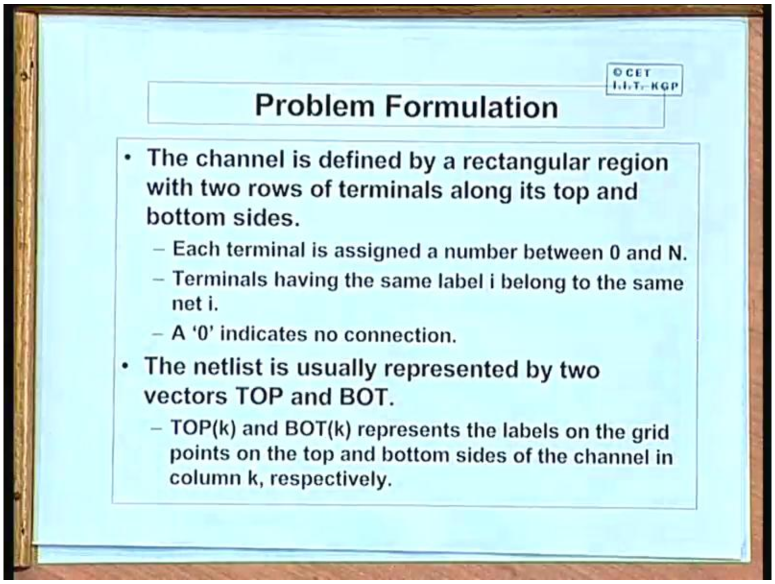


This is an example channel. This is the upper boundary this is the lower boundary. These are the terminals. Terminals are typically arranged in synchronism with the columns of the grid. Each terminal is labeled with a number this number indicates the net. 1, 1, 1, means these 3 belong to the same net and they have to be interconnected. Similarly 2, 3 so on. 0 means no connection. So the channels specification you can express very efficiently by 2 arrays top and bottom. Top will contain these numbers 1, 2, 0, 2, 3 bottom 3, 3, 1, 1, 0. So the kth element of these array arrays will indicate the terminal in the kth column of the channel of this problem I mean.

Now this gives 1 possible solution to this channel routing problem. So let me show this by color. This is a 2 layer routing model say HV 1 layer is horizontal and the other layer carries the vertical connections ok. So now an according to the definitions which we had given earlier these are the tracks. This is 1 track, this is the second track, this is the third track these are tracks. There are 3 tracks on which you can layout the horizontal segments of nets ok. These are the trunks segments of the net. For example for net 1 here to here there is a horizontal segment which has been laid out on track number 2. Suppose 1, 2, 3. This is a trunk.

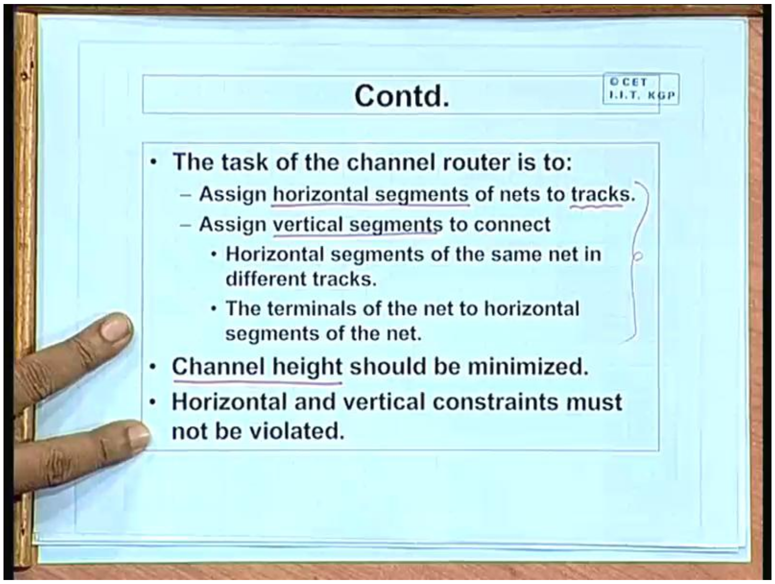
Similarly for net number 3, this is the trunk for net number 2 this is the trunk. The vertical segments are the branches they connect a trunk to a terminal trunk to a terminal. Sometimes this example does not show you a branch can also connect 1 trunk to another trunk. For example you can have a connection like this also. There a branch will connect this trunk with this trunk. You will see later that it become necessary sometimes to break a particular net in more than 1 tracks of course this example does not require that. Our particular net can be completely laid in a single track in terms of the horizontal trunk right ok.

(Refer Slide Time: 35:10)



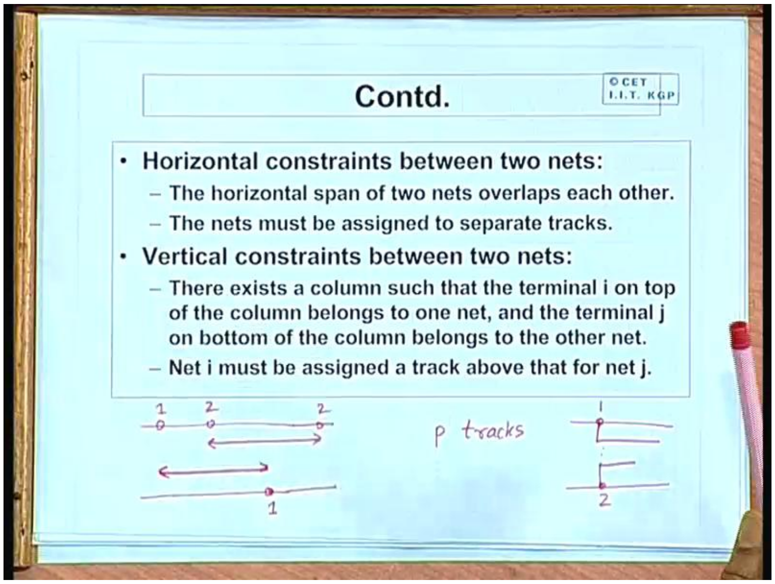
So first let us go in to the problem formulation. The channel is defined by rectangle region just as I mentioned with 2 rows of terminal along top and bottom. Each terminal is assigned a number between 0 and N, where N is the number of nets. Terminal having the same label I they belong to the same net, 0 means no connection. The net list represented by 2 arrays as I had mentioned top and bottom. The k th element of the array represent the labels on the k th column of the channel top and bottom ok. So this is the input to the problem.

(Refer Slide Time: 35:55)



Now the channel router will be taking that input specification and will try to assign the horizontal segments of the nets to tracks, assign a vertical segments to connect either horizontal segments of the same net in different tracks, just I have mentioned zigzag kind of a thing or the terminals to the horizontal segments. So the channel router will make these assignments to complete the routing. The objective is obvious the channel height should be minimized means number of tracks you require that has to be minimized. This is the primary goal. Of course we have not talked about wire connections as yet. We will see it later that how we can minimize wire also. But while doing it there are some constraints that must not be violated. These are called horizontal and vertical constraint let us see what these constraints are? So any channel router must not violate these constraints. Now this constraints are failure obvious.

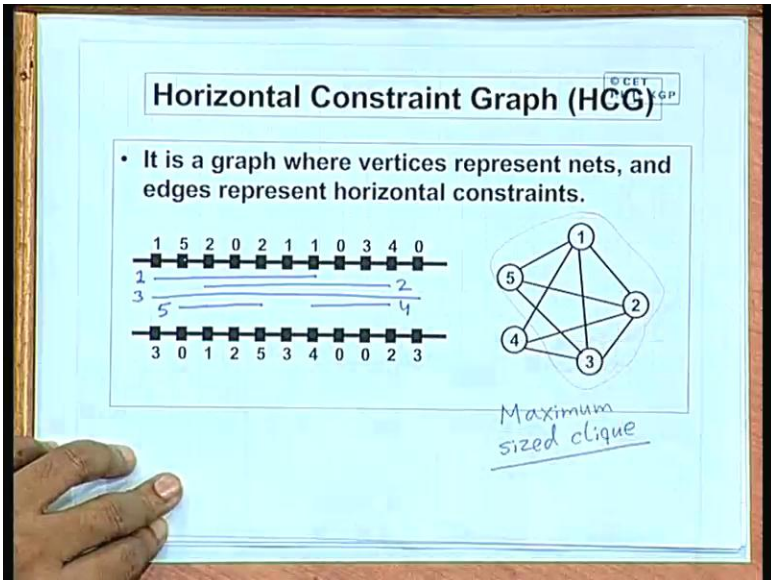
(Refer Slide Time: 37:02)



Say a horizontal constraint says suppose I have a net 1 say with 1 out here and 1 out here and a net 2 with 2 here and 2 here. Well if you look at the horizontal spans of the 2 nets the net 1 spans from here to here net 2 spans from here to here. So there is an overlap between the spans. Now since they overlap you cannot layout these 2 nets on a single track you will require minimum 2 tracks for these. So if there is a horizontal constraint like this then the 2 nets cannot be laid on the same track. So in general if there are p number of nets which mutually they have horizontal constraints with each other then minimum p number of tracks are required for routing. This will give you a load bound to the number of tracks required ok.

Similarly you can have something called vertical constraints. Vertical constraints says that suppose you have a channel with a terminal level 1 out here and exactly down below you have a terminal level two. This says that if you layout the horizontal segment of net 1 say out here then the corresponding horizontal segment of net 2 must be below this it cannot above this. If you try to be above this there will be a short circuit here it has to be something like this. There exists a column such that the terminal on top belong to 1 net and the terminal j on bottom belong to other net. So at least on this column net I must be assigned a track above that for j. Because if you try to do the other way around in terms of the branches there will be a short circuit ok. These are the constraints. Now this constraints are expressed in the form of a graph.

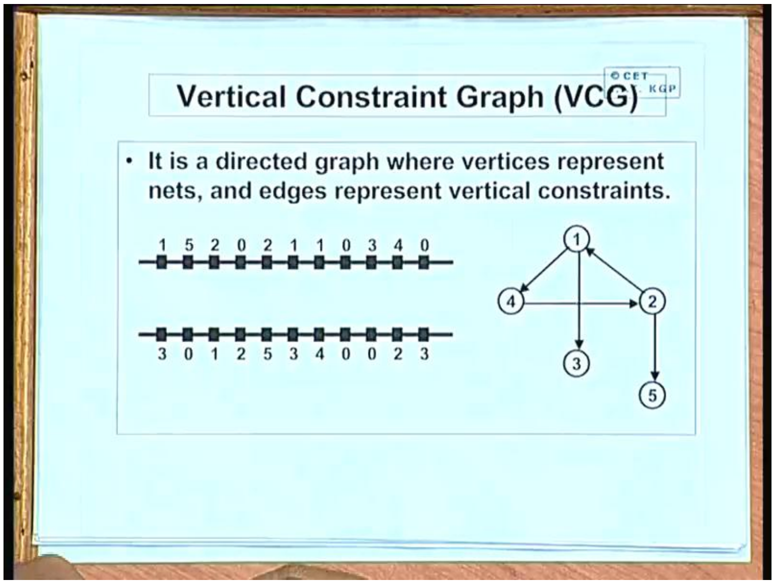
(Refer Slide Time: 39:30)



First for the horizontal constraint. Let us take an example. This is a more elaborate example of a channel with the points noted. Now a horizontal constraint graph is a graph where the vert that there is a horizontal constraint between them. Now drawing the HCG is very easy you just absorb the horizontal span of the different nets. Say for 1 this is the span this is for net one. Net 2 it starts with here and ends here. Net 2 this span is this. Net 3 it starts here ends here so the entire span. Net four from here to here. Net five from here to here. These are the horizontal spans. Now if you look at the horizontal spans you see that other than four and five the others are all mutually conflicting.

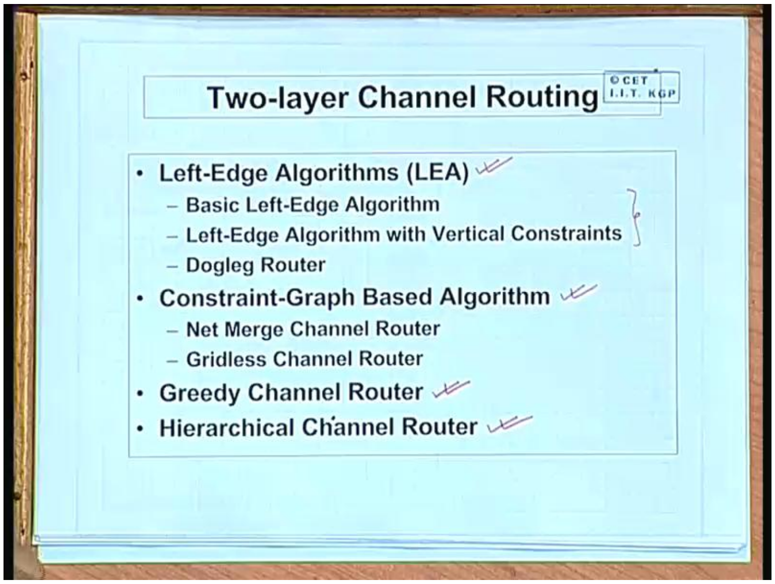
So the corresponding graph is an almost complete graph without this edge between four and five ok. Now in this graph you can find out the maximum sized clique. Now clique is a complete graph a complete sub graph. Now in this graph of five vertices you can find out that the maximum clique you can have is a sub graph of four vertices. If you take this sub graph this is a complete graph. This will mean that any solution to this problem will require minimum four tracks. Because there are four nets which are mutually conflicting horizontally this is the lower bound ok fine. Similarly you can have a vertical constraint graph VCG.

(Refer Slide Time: 41:53)



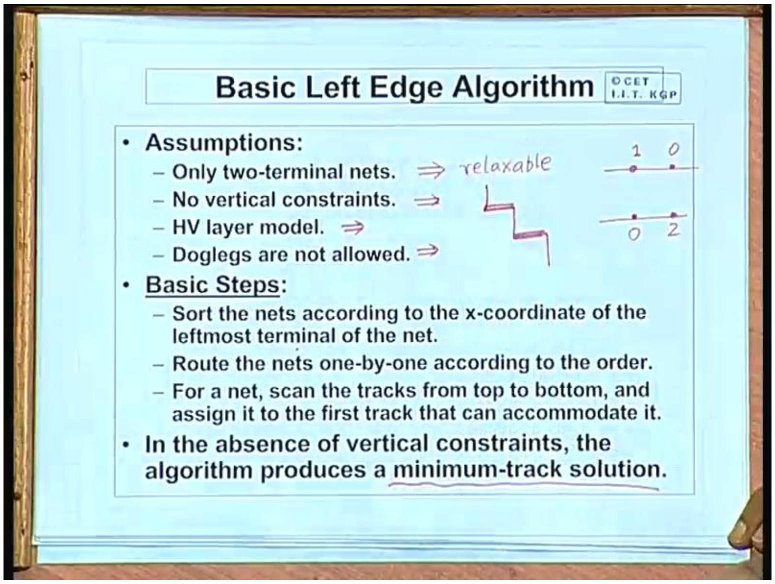
VCG is a directed graph because here you keep track of who is above whom. Here 1 is above 3 in column 1 so there is H from 1 to 3. 2 is above 1, so from 2 to 1 there is an edge. 2 is above 5, 2 to 5, 1 is above 3, 1 to 3, 1 is above 4 and 4 is above 2. Rest are no connection zeros are no connections. So from a given specification you can have a vertical constrain graph. Vertical constrain graph will give you some information regarding which, in which order you will have to do the routing or you will have to assign the because the segment of the nets to tracks this we will see subsequently. But 1 thing let me tell you that the problem case is when the VCG contains a cycle as this example also contains a directed cycle. So if a VCG contains a cycle this is a problem case you will have to have some kind of a zigzag kind of a connection we will see later ok. So we first look at the different 2 layer channel routing algorithm.

(Refer Slide Time: 43:19)



Because they are the most widely explored. There are multilayer 3 layer channel al channel routing algorithms which have been proposed but they are mere extensions to this. So first we will have to understand very nicely or very clearly that the main or the important 2 layer channel routing algorithm 2 layer means h and v on 1 layer horizontal tracks other layer vertical tracks. The first important class of algorithms they are based on a very simple concept called left edge algorithm. The basic version an extension of it with vertical constraints and something called a dogleg router. Then there are some algorithms which are based on something called constraint graph there are couple of algorithm there. Then greedy channel router and hierarchical channel router this we shall be looking at. To start with we look at the basic left edge algorithm and its extension. The basic left edge algorithm is very simple in concept.

(Refer Slide Time: 44:38)

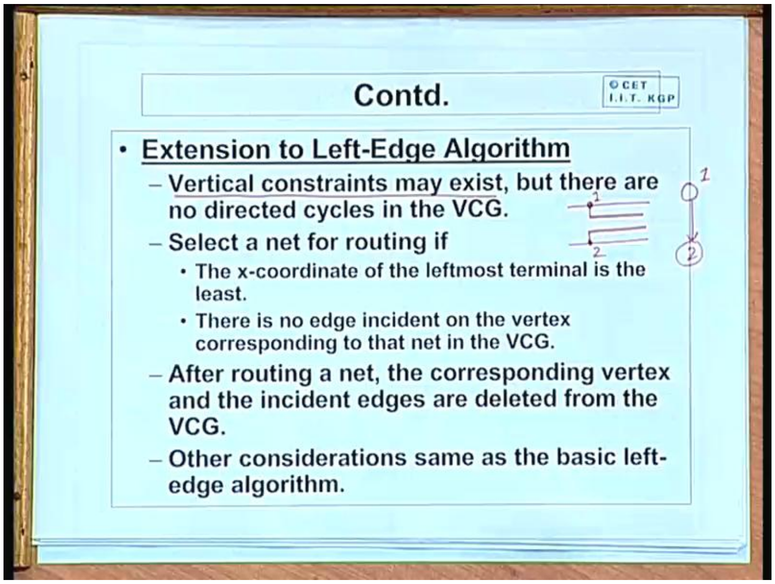


Well the basic left edge algorithm says that you have only 2 terminal nets. But you will see later that this assumption is not mandatory this can be relaxed. This is a relaxable assumption. So you need not restrict yourself only to 2 terminal nets you can have multi terminal nets also. Well the basic version assumes that there are no vertical constraints which means that a pin belonging to 1 net cannot be above a pin belonging to some other net. This means that if you have a signal pin label 1 here the 1 below it must be label zero. Similarly if you have a pin level 2 here the 1 above that should be label zero. This is of course a very restrictive assumption. This is the basic version of the algorithm but you will see later that we can just relax that also. So this this is a 2 layer model horizontal vertical. Doglegs are not allowed. Well I do not know why they are call it dogleg I have never seen a dog with a leg like this.

Say a net like this called a dogleg where the horizontal segment is broken into more than 1 track. This kind of a connection is called a dogleg. There are 2 horizontal tracks on which the total horizontal segment is assigned. Now in with these assumptions you will never need a dogleg. The basic steps are very simple. They says that for each net you look at the horizontal spans. And you sort the net according to the x coordinate of the leftmost terminal. The net with the first terminal on the left in terms of the horizontal span you keep it first. So in that order you sort them. Route the nets 1 by 1 according to this order and you place the net in the first track you can place them scanning from the top to bottom. For a particular net which you are trying to route scan the tracks from top to bottom and assign it to the first track that can accommodate it. So the algorithm is very trivial.

You look at the horizontal spans of the net the 1 which is coming left most with respect to left terminal you assign it to the first track take. The next one you see that whether it can be accommodate on the first track if not put it on the second track, take the third one you again scan from the top you find out the first track for you can put it you put it there in that way you go on putting them. Now in the absence of these constraints it can be shown that this produces a minimum track solution. For obvious reasons you cannot do any better you can just a few look at the algorithm though it works. Now as I had mentioned that this no vertical constraint assumption is true restrictive. So first let us try to avoid this assumption to make it slightly more general. So the way we can do this is very simple.

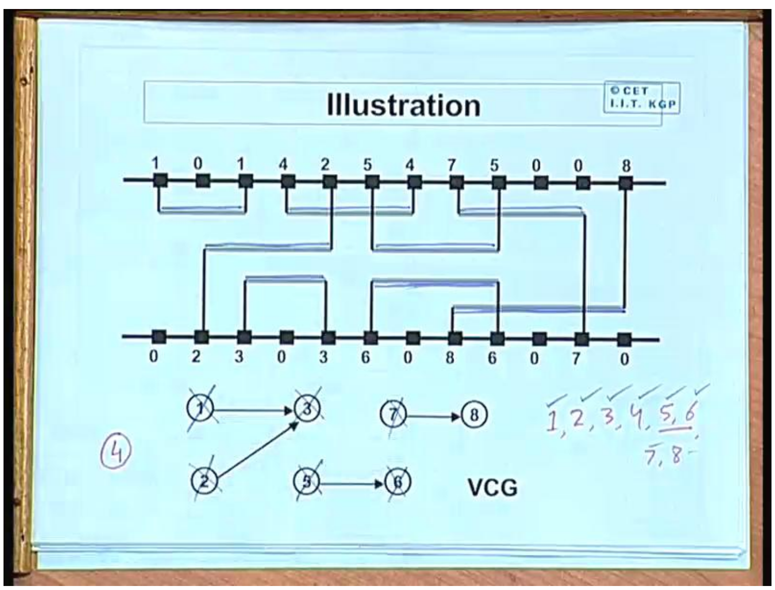
(Refer Slide Time: 48:17)



There can be vertical constraints which means that the VCG is not a null graph there are edges in the graph. Now VCG will dictate that which node should be routed first. You take a simple example. Suppose net 1 just below it net 2. So with respect to the VCG from 1 there is an edge to 2. Now in terms of assignments to the tracks you know that well net 1 must be placed on the track first net 2 must be placed on a track below it. So now what I say is that I would be I would be following or honoring the ordering in the VCG. Well I will be sorting the nets as earlier but I will be picking up a net of routing provided in the corresponding graph there are no incoming edges in the corresponding node. If the graph is like this then 2 is not a candidate I cannot take 2 for routing before 1 I will have to take 1 first.

First I will route 1 then I will remove this 1 from the graph. In the residual graph again I will do the same thing. I will only pickup nodes which do not have edges coming from some other nodes ok. So now the algorithm will be like this that we can select a net for routing if the x-coordinate of the leftmost terminal is the least. This is like the basic algorithm. And secondly there is no edge incident on the vertex corresponding to the net in the VCG. So if you follow this then you can get a solution. But of course after the routing of that particular net is completed you will have to delete that vertex from the graph that vertex and the associated edges. Ok we have an example to illustrate this. Let us take this example.

(Refer Slide Time: 50:45)

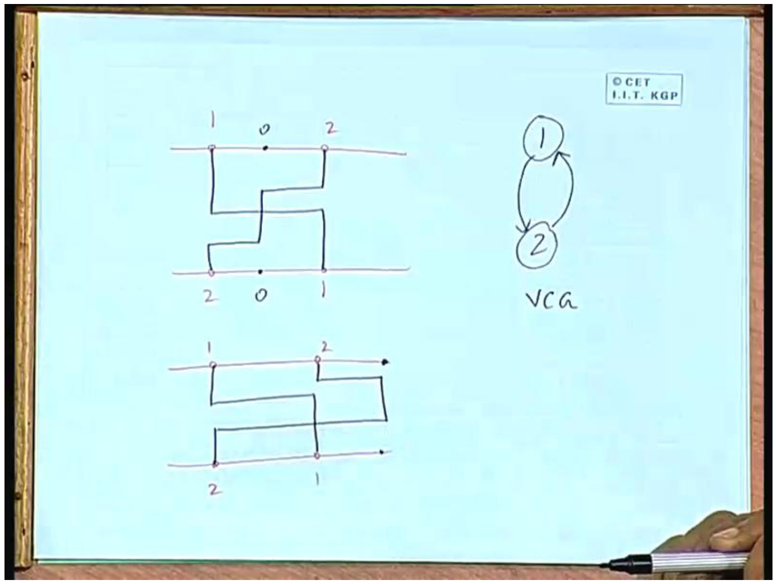


This is slightly most complex example with nets 1, 2, 3, well 4, 5, 6, 7, 8, there are 8 nets ok. Now in terms of the vertical constraints you see 1 is above 3. So 1, 2, 3. 2 is above 3, 5, is above 6 and 7 is above 8. Well there is a vertex four also but four does not have an edge, edge with anybody else 4,0 4,0 there is no vertical constraint. So if you look at the left most boundaries if you try to sort say left most boundaries net 1. So first we will have net 1 then 2, 2 then 1 have taken 3 then 4, 5 and 6 they start from the same column. So 5, 6 both in any order we can have this. Then 7, 8 again starts from the same column. So this is the sorted order. Now we would be laying out nets in this order subject to the constraints in this graph. First let us take 1.

Well we can take 1 because this node 1 does not have any incoming edges. So we take 1. We this is the span of 1 we complete this assignment. You route you the assign the horizontal segment in track 1 remove this node from the graph then take 2. You assign there is a horizontal constraint between 1 and 2. So you assign it to the next available track. Remove it from the graph. So now 3 is also candidate now in this list 3 comes next. You assign 3 to the next available 3 cannot be put on row 1 or row 2 it has to be put on row 3 track number 3 remove 3. Then comes 4. 4 you can put in the first track without any there is no horizontal conflict between 1 and 4. I have not shown the HCG we will also have to have that. So four can be laid out on track 1 4 is d 1 then 5 no constraint 5.

5 you cannot put on track 1 it has to be put on track 2 remove this. So now 6 is also ready you take 6. 6 cannot be put on track 1 and track 2 this has to be put on track 3, 6 is d 1 then 7. 7 can be put on track 1 this is the horizontal segment span 7 is d 1 then 8. Well eight starts from here and it has a horizontal conflict to the 7, 5 and 6. So it has to be put on the fourth track this is the final solution. So basically the left edge algorithm it assigns a net to a track just the horizontal lines. Now vertical lines you can complete after that. The way these are assigned there will be no conflict in assigning the vertical lines. Yes [Students Noise Time: 54:15] Ok, ok. I was just about to say that.

(Refer Slide Time: 54:27)



You are saying that you have a problem where 1, 2, 2, 1. So here you are saying that in the VCG there is a cycle. (Student Noise Time: 54:39) You have a situation like this in the vertical constraint graph. Now if we have a cycle you must have a dogleg. There are 2 ways you can do it either you see you need an additional free column. If the free column is in the middle then you will have a dogleg like this or if we have a free column on 1 side of it. Suppose there is a free column out here ok. So you complete or on the other side. So there what you do? You complete this routing the net to you will be going to this column then again you will have to come back on another row. So you will be needing addition say 3 tracks anyway you will be needing 3 tracks in both cases. So the idea is that if you have a cycle in the VCG you will have to break the horizontal segment of at least 1 net into more than 1 track. So this doglegging algorithm will be discussing in detail in our next lecture. Thank you.