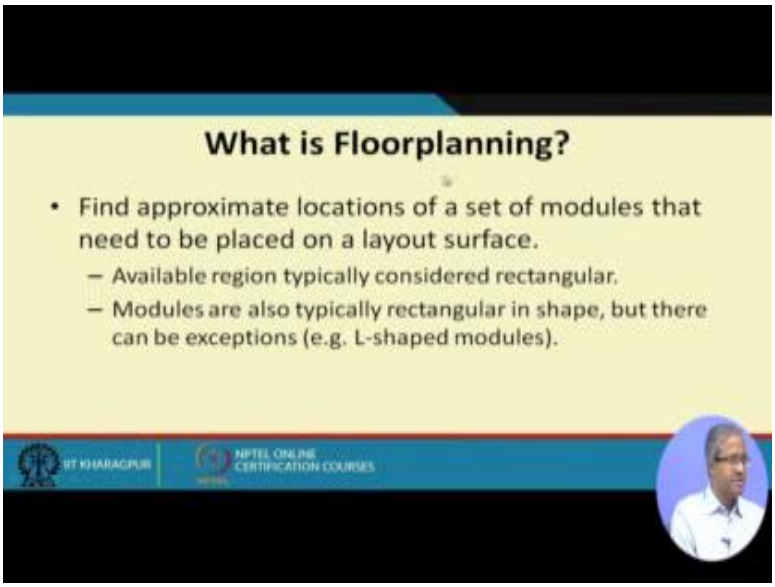
VSLI Physical Design
Prof. Indranil Sengupta
Department of Computer Science and engineering
Indian Institute of Technology, Kharagpur

Lecture - 08
Floor Planning

So, we consider now the next sub problem namely floor planning. So, floor planning let us first see what this problem is all about. Now see in typical wheel is a physical design what we are trying to do. We have our play ground which is the silicon floor, where we have to lay out all the circuit components. Floor planning means just to recall the analogy I gave earlier it is like building the plan of a house.

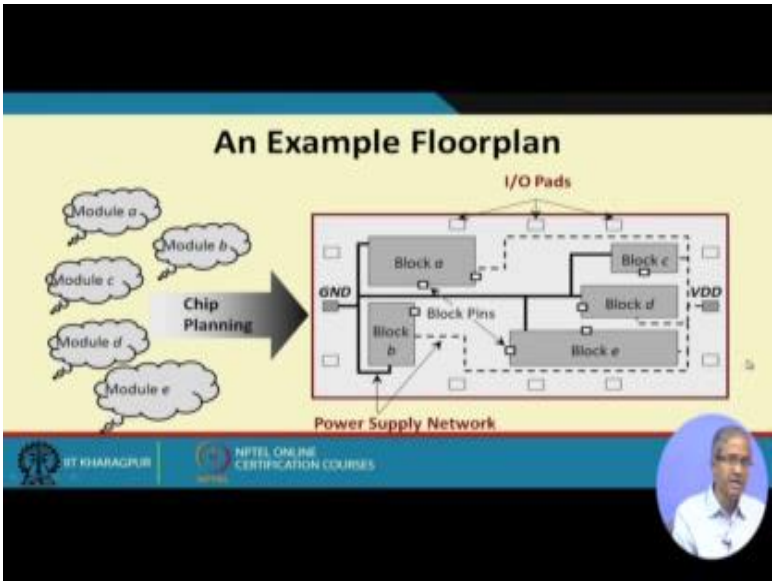
So, I have my circuit blocks, I have to make a planning where to put which circuit blocks on my silicon flow floor. So, once I do this planning in a nice way, the task of lying out and handling the physical design process in subsequent stages becomes much easier. Because one thing we remember in physical design we are tackling circuit or netlist containing millions of millions of components. So, once we do this kind of partitioning, floor planning, we are going for something like a divide and conquer approach. We are reducing the complexity of the problem to a great extent by dividing or splitting the problem into simpler sub problems, and then handling them just by one by one independently.

(Refer Slide Time: 01:55)



So, you can say floor planning concerns finding the approximate locations of the modules on the layout surface. Now some assumptions are typically met the available region for example, the silicon floor is considered rectangular in shapes, the modules that were trying to place they are also typically concerns to a rectangular, but sometimes you can have other module shapes like L-shaped modules we shall see this later.

(Refer Slide Time: 02:34)

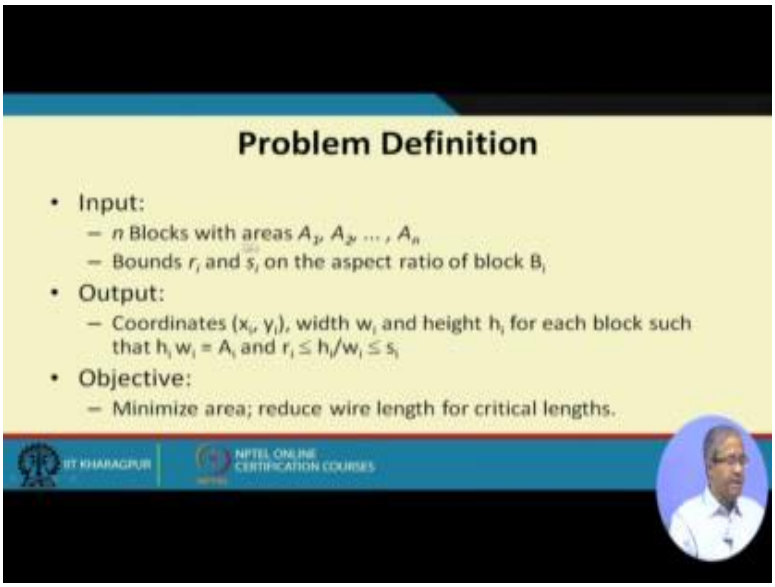


So, let us look at a diagram which will just apprise you about the overall scope of the problem that we are talking about. On the left we show some modules that we are trying to place, and this rectangular region on the right is our silicon floor. So, these 5 blocks we place like this block a b c d e, these are our tentative floor plan. Now this another thing which we are also showing here, which may or may not be done in the floor planning stage, now in fact, in our lecture sequence we shall be considering the sub problem latter.

Like in addition to creating the floor plan, we are also tentatively fixing up the position of the external pins or the I O pads, these are the positions from where our external connections will go out, and also the power supply lines the solid edge indicates the ground line, this doted edge indicates the power supply line VDD. So, as you can see this is the non actually ground and VDD lines are typically laid out on the same layer metal layer. So, these ground and VDD these 2 nets are not intersecting each other you can see. So, this is a typical solution to the floor planning problem including power and ground

planning. This is the overall chip planning I am showing, but if we do not consider VDD and ground planning here, our floor plan will only consist of the blocks and their relative positions in the floor we shall see some examples later also.

(Refer Slide Time: 04:37)



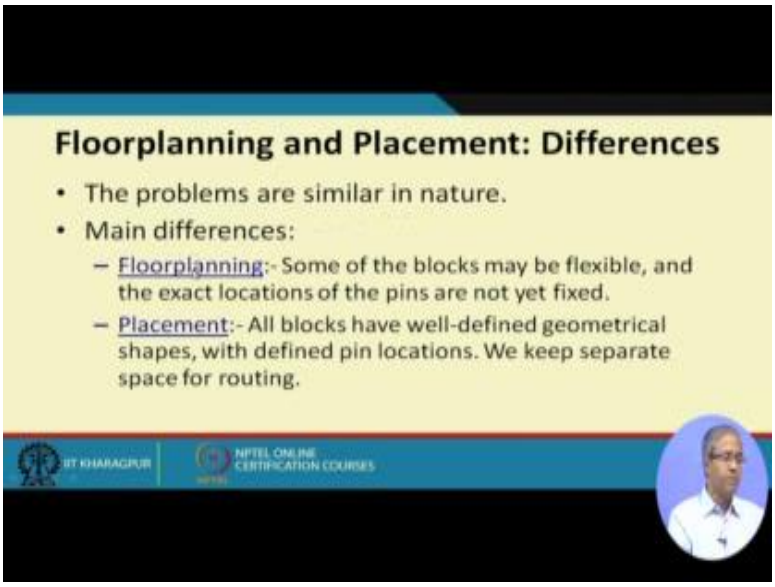
So, the problem definition goes like this; suppose we have n number of blocks, whose areas are given and there is another thing some of these blocks can be fixed, some of these blocks can be flexible. Fixed means I know the width and the height of the blocks, but there are some other blocks I know how many gates or transistors are there, which means I know the area, but my height and width is still flexible. So, I can adjust that. So, my area is given those are called flexible blocks.

So, and we can have some bounds on the aspect ratio; aspect ratio means height divide by width. So, ideally speaking I would like a block to be square in shape, but it can be thin either vertically aligned or horizontally aligned, but how thin that aspect ratio you can possibly specify as a constraint, that it cannot be thinner than this the height to width ratio cannot be more then or less then this.

So, the output will be the exact location of each block, their widths and heights and for the fixed blocks width and heights are already known, for the flexible block you will have to fix up. Their product will be equal to their area and of course, they will be lying within the limits or the bounds of the aspect ratio r i and s i. The objective of will be of course, to minimize the total layout area, and also for the critical wires or the critical nets

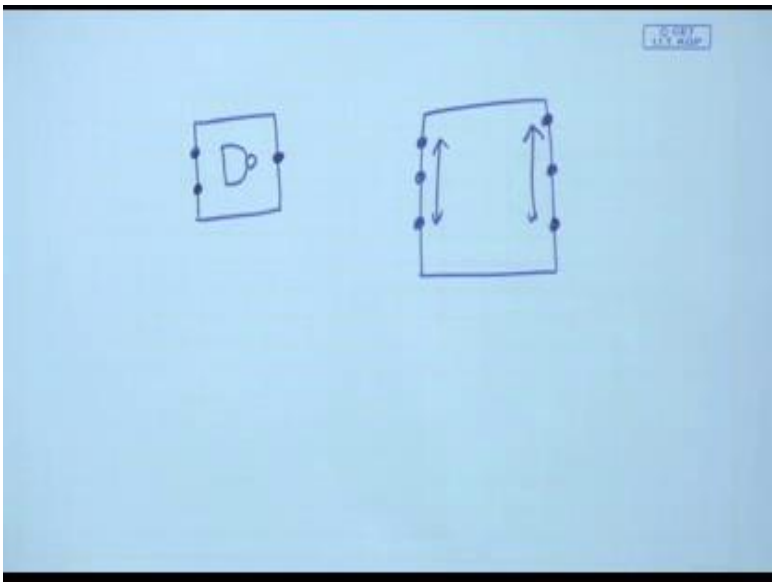
to reduce the wire length, but one thing you remember here although we are talking about reducing wire length, we are only talking about a tentative placement of the blocks. So, exact wiring of the wires we have not done yet, we can only make a very coarse estimate at this stage. So, even with that rough estimate we can say that well my critical net is becoming too long, let us bring these 2 blocks together keep them side by side something like that.

(Refer Slide Time: 07:05)



Now one thing we shall be talking about the placement problem later. Floor planning and placement are quite similar, but there are some precise differences also let see what the differences are. Now in floor planning as I had said some of the block may be flexible in nature, in the sense that their area is known, but there heights and widths are not yet decided and also in each block the pins are there.

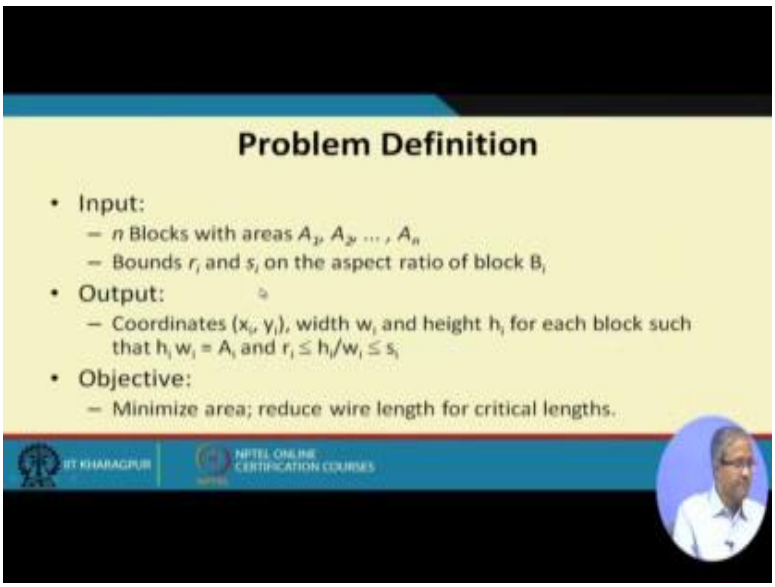
(Refer Slide Time: 07:40)



Like for example, suppose I have a fixed block like this. So, let say this corresponds to nand gate let say. So, my 2 inputs can be here and here my output can be here, these are fixed pin locations.

But if I have a flexible block where I know that there are 6 six pins I need, but their exact locations are not fixed they can move around, the pin locations are also flexible here.

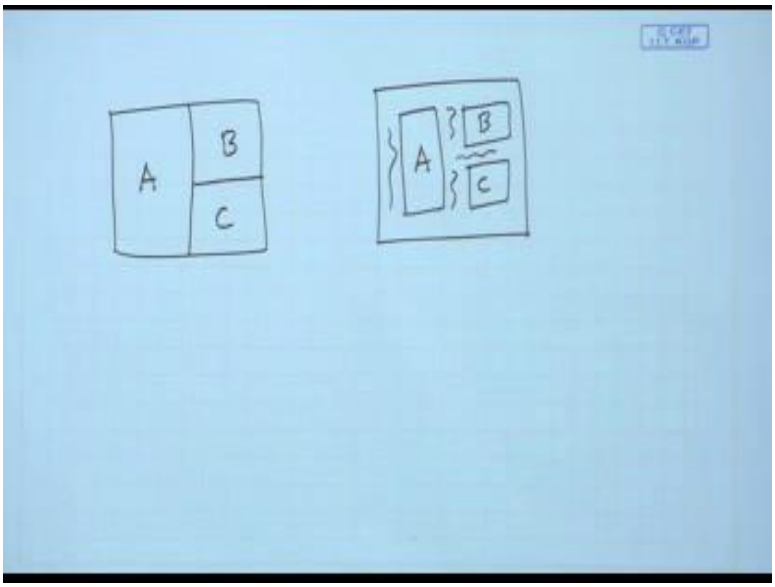
(Refer Slide Time: 08:14)



So, in floor planning the exact location of pins they also may not be fixed, you also need to fix the position of the pins. But in the placement problem we assume that all the

blocks have well defined geometrical shapes and also their pin locations are fixed and although in floor planning we do not keep space for interconnections, but in placement we explicitly keep space for interconnections like an example.

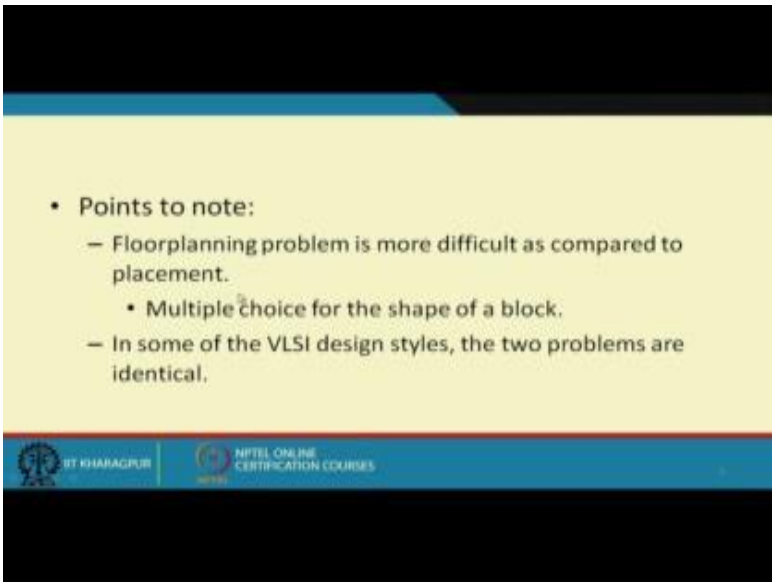
(Refer Slide Time: 09:48)



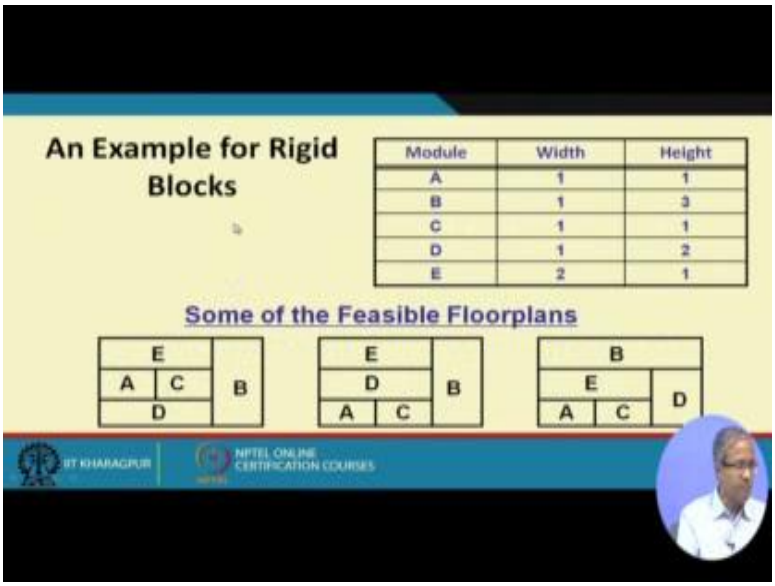
Now, in floor planning I can say that I have 3 blocks A B and C which has to be placed like this; B on top of C and A to the left of B and C. But in placement here I will specify like this, this is my block A, this is my block B, this is my block C and there is some space in between I have kept for interconnections; this is the output of the placement problem right. So, this is the main difference between floor planning and placement; and the point to notice that the floor planning problem is more difficult to handle because you have the concept of flexible blocks. So, you have the area given for a block. So, there can be so many different possible shape of the block, we have to find out which particular shape can give a more compact in a better layout.

So, this is not a easy problem, but in placement because the shapes and sizes of the all the blocks are fixed the problem is much easier; but in floor planning in addition to placing them you also have to find out the aspect ratio of the blocks, the exact shapes of the blocks that are flexible right, but we shall see in some of the design styles the 2 problems are not different they are the same.

(Refer Slide Time: 09:33)



(Refer Slide Time: 10:25)

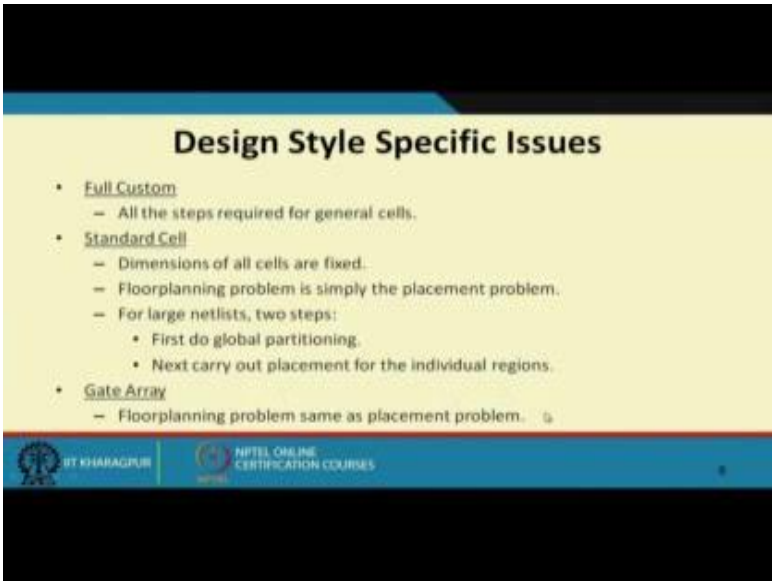


So, let us take one example floor planning problem. So, we consider that there are 5 modules, width and heights are given. So, these are all fixed module none of them are flexible. So, A is a square module 1 by 1, B is mean 1 by 3, C is 1 by 1, D is 1 by 2, E is 2 by 1; these are 3 possible floor plans. Now one thing you see although I know that B is 1 by 3, but I can lay it out either vertically or horizontally that flexibility I have.

Similarly D, I can lay it out either horizontally or vertically because I know the shape of the block, but there is no problem if I rotate it by 90 degrees. So, a horizontal block I

make it vertically by rotating that flexibility I have. So, by doing that, these 5 blocks can be placed in several ways; so you can have several alternate floor plans. So, the point arises here is that, so once you have many floor plans for a given problem, you should have a mechanism to determine which of these is better than the other, we will have to chose the best floor plan among all the alternatives, this is something we have to do.

(Refer Slide Time: 12:00)



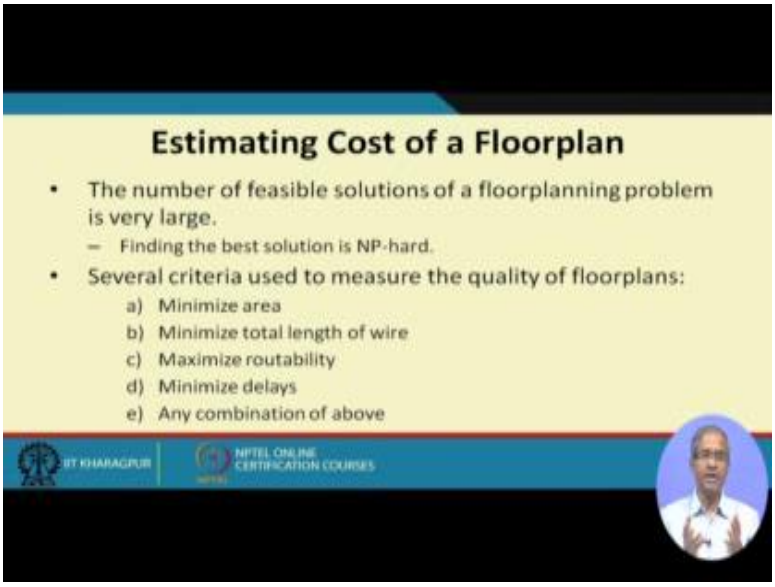
Now coming to the design style specific issues let us have a quick look at this. For the full custom design style you recall in the full custom design style the blocks can have any arbitrate shapes and size. They can be small, they can be big, they can be square they can be rectangular. So, both floor planning and placement problems will come into the picture for the full custom designs style, and here you need all the steps that are required. But for standard cell, but the dimension of all these cells are fixed because you know the cells are picked from a library, and else at the heights of the cells are fixed, but the width can be different; but once I choose a cell the shape and size is fixed and it has to be placed in this way only, you are not also allowed to rotate these cells in a standard cell layout.

So, for standard cell floor planning and placement problems are the same, they are not different sub problems. So, when you have a large netlist you go through 2 steps, first you do a global partitioning, you partition the whole netlist into a smaller pieces, and each of the piece you possibly map into individual regions; may be one raw of the

standard cell, you lay them out into one row of the standard cell and carryout the placement.

Similarly, for gate arrays you have so many gates which are spread on the layout floor there are so many gates. So, when you say floor planning, you are planning where to place the gates. When you talk about placements you are talking about the same thing how to place the gates. So, floor planning and placement even for a gate array is identical they are not different. So, these 2 problems are the same.

(Refer Slide Time: 14:07)

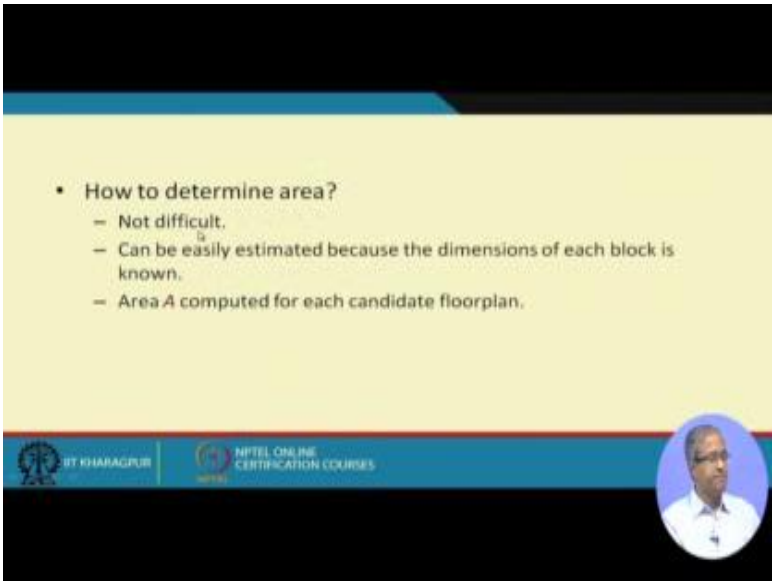


Now, I had said that when you have a number of alternate floor plans to compare, you should be able to just evaluate them with respect to some cost. So, let us see how we can evaluate the cost. So, this is required as I had said that the number of feasible solutions of a floor planning problem can be very large, and determining the best solution can this has been proved to be computationally complex NP-hard.

Now for a new compare more than one floor plans, there are several ways in which you can make this comparison, you can look at the total area, you can look at the total wire length, you can evaluate how easy or how difficult will be interconnect the blocks; how interconnection complexity is different like for example, you may place in the floor plan 2 blocks which are little far away, but the number of connections are pretty large between them. So, you can say that your interconnection complexity will be larger there.

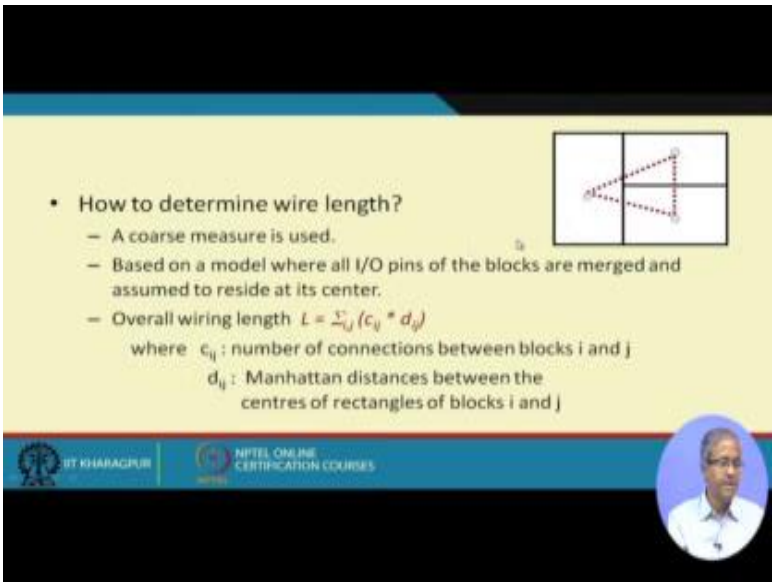
Minimize delays for the critical parts, so you see that how far they are the points you want to connect for the any combination of these.

(Refer Slide Time: 15:34)



Determining area is not difficult; because you see your total silicon floor is rectangular in nature. So, if you can have a better floor plan your that size of the rectangle will reduce. So, just height multiplied by width will be your area. So, you can measure the size of your area very easily by looking at the width and height of the total overall floor plan and taking the product. So, area can be easily estimated; for each candidate floor plan you can measure the area very easily.

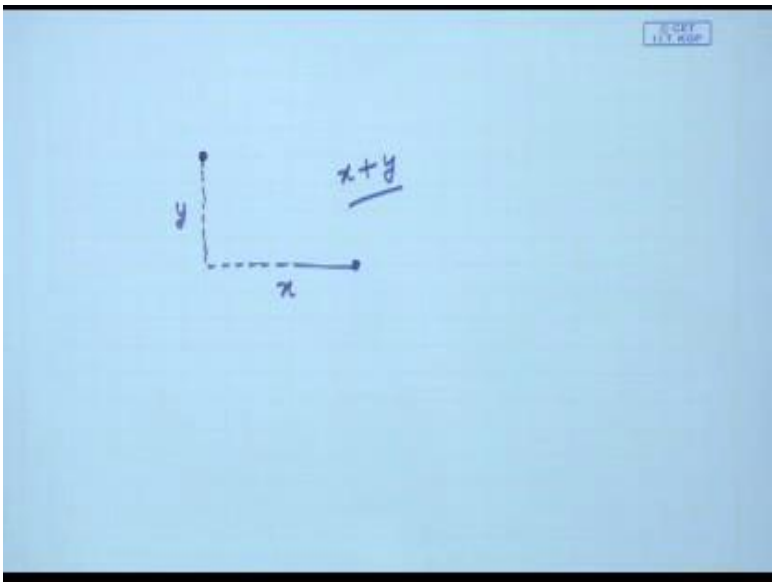
(Refer Slide Time: 16:15)



But wire length. So, I had said at this stage you are only allowed to make a very rough estimate; because here at this stage you have not even kept separate spaces for interconnection, you do not know how the wires will be laid out you can only make an approximation.

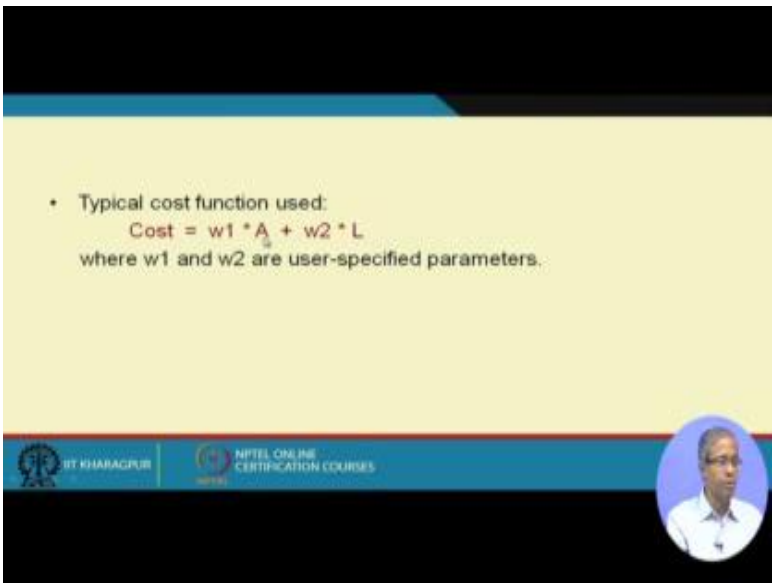
So, one popular method which is used is that I am just explaining with help of this example, suppose I have 3 blocks shown by these 3 rectangles. So, what I; here assume is that in each of these blocks there can be several pins, which need to be connected to several other pins in other 2 blocks. So, we assume that all the pins are merged and residing at the centers of the blocks. Suppose these are the block centers, I assume that the pins are residing at the centers of blocks; and I calculate how many interconnection lines are there between this block and this block that is c ij between block i and block j how many interconnections are there? And d ij is the distance; distance I am calculated as a Manhattan distance.

(Refer Slide Time: 17:43)



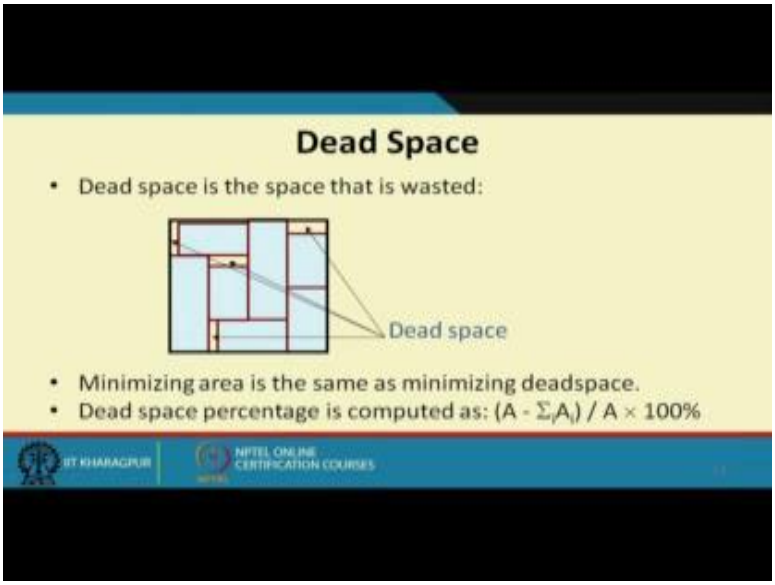
So, what is Manhattan distance? Suppose I want to connect a point this with this, Manhattan distance means you go vertically then you go horizontally. So, this y plus x Manhattan distance is this x plus y. So, you can easily calculate the Manhattan distance given the coordinates of these points, number of interconnection wires you know. So, in this way you can have a very coarse estimate. Across every pairs of blocks you see how many connections are there, multiplied by, the Manhattan distance between the 2 blocks this will be a very rough estimate of the wiring length total wiring length.

(Refer Slide Time: 18:32)



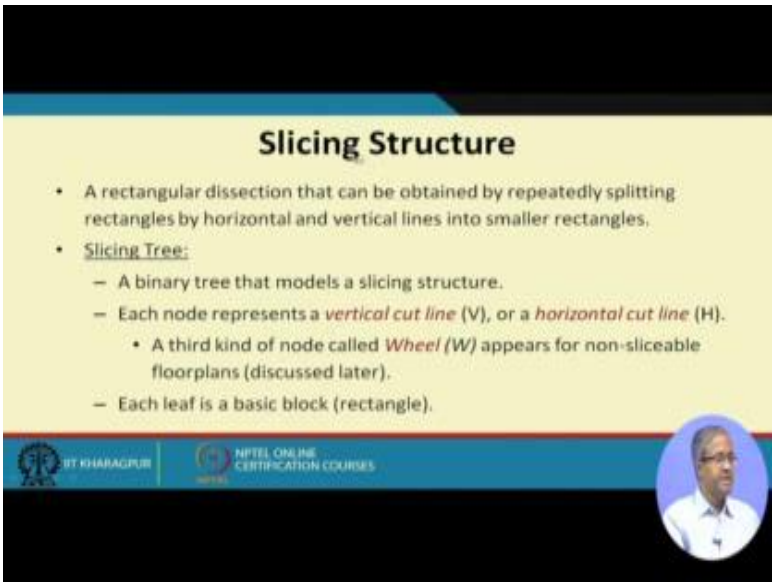
And the total cost you can estimate as some weighted sum of the area and the weighted length. Now it depends in some cases your area may be more important, you may give higher weightage to w 1, but your interconnection length minimization is more important you can give a higher length to w 2 higher weights. So, this w 1 w 2 can be specified by the user.

(Refer Slide Time: 19:01)



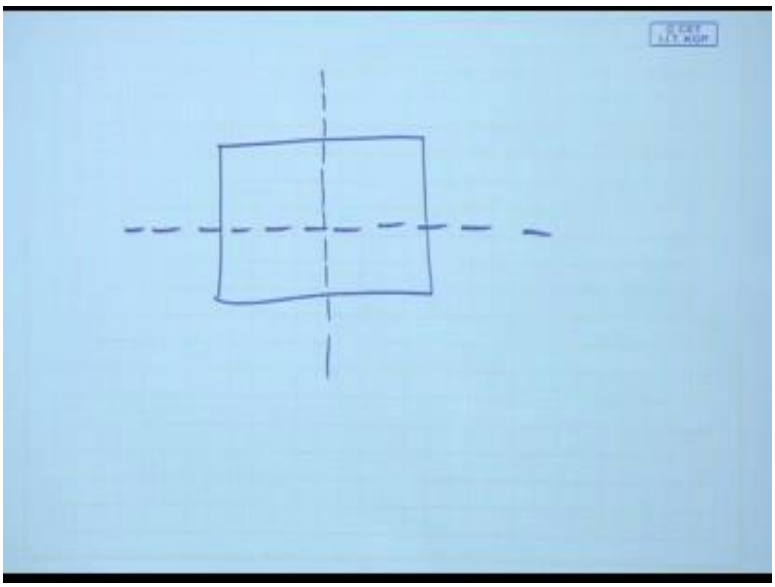
Now, there is another way to estimate the goodness of a floor plan, this is by measuring the dead space. Dead space means the space within a floor plan which is wasted, like you know the total area of this floor plan you can estimate by multiplying the height by the width and the individual blocks are there. So, you take the sum total of the areas of the blocks. So, you get the areas of the blocks, if you subtract that from A you get the area of the so called unoccupied space which is sometimes called dead space. So, that divide by A, if you calculate the percentage this sometimes you call it as dead space percentage. Sometimes instead of area we can say that we want to minimize the dead space percentage, reducing dead space percentage means you are reducing the area it means the same, so the problems are similar.

(Refer Slide Time: 20:09)



Now let see how we can represent a floor plan. Slicing structure is a very common way of doing that; let us try to see what it is.

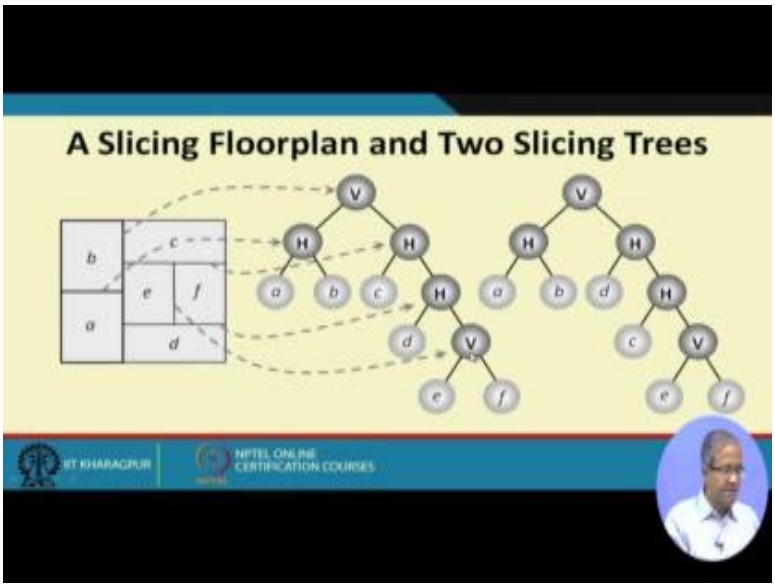
(Refer Slide Time: 20:28)



Slicing structure means it is like a rectangular dissection like let say suppose I have a floor plan, say there are many blocks. Dissection means I can cut the floor plan either by a vertical line or by a horizontal line like this, I can cut it into 2 parts this are called dissections. So once I do a dissection I divide the floor plan into 2 smaller floor plans right.

So, slicing structure actually tells you how to slice a floor plan repeatedly until you get the basic blocks. So, that slicing tree is something which specifies that slicing structure, I shall show some examples; slicing tree is a nothing, but a binary tree, at every stage you are doing one slice, horizontal slice or a vertical slice. In this binary tree the vertex or a node represents either a vertical cut represent by V, or a horizontal cut represented by H. Later on we will see that a third kind of vertex can be also there called a wheel, which appears for some floor plans which cannot be sliced this we shall see a little later and the leaf of the tree indicates a basic block, the blocks that you want to place in the floor plan.

(Refer Slide Time: 21:59)

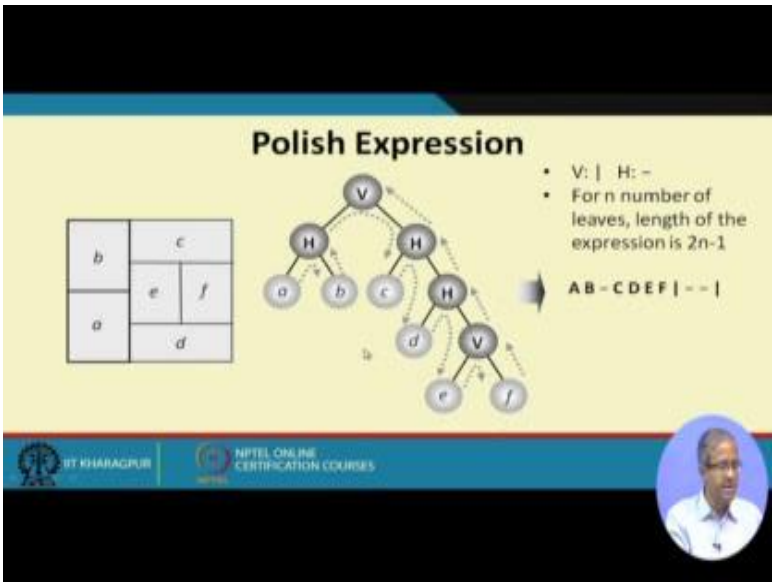


Let us take an example; suppose I have a floor plan like this a b c d e f there are 6 blocks, which are placed on the floor like this. Now this tree on the left represents one possible slicing tree; let us see how it is done lets starts from leaf side, e f connected by a vertex V which means e and f are partitioned by a vertical line, this whole thing indicates this entire rectangle e f, d and this entire rectangle is connected by a horizontal line, d and this entire rectangle is connected by a horizontal line this one. This whole thing and c is connected by another horizontal line, this whole thing and c is connected by another horizontal line and the other side a b is connected by horizontal line, a b by horizontal line this whole thing and this whole thing connected by a vertical line this vertical line.

So, this floor plan can be represented by a slicing tree like this, you can check this is also a slicing tree which represents the same floor plan. So, this c and d are actually

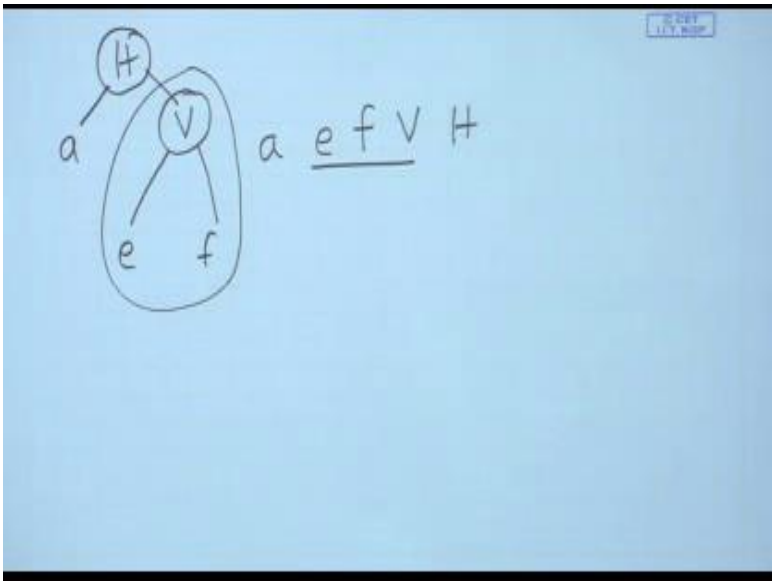
interchanged because you can see there are 2 horizontal slices, so either you do this slice first or this slice first does not matter; so either you do this d first and then c, or c first and then d does not matter. So, for a given floor plan there can be more than one slicing trees possible.

(Refer Slide Time: 23:49)



Now, a slicing tree can be represented in a compact way by a something call a polish expression also known as a postfix expression. Now what is polish expression? We shall explain it later, but here let me just briefly tell you about.

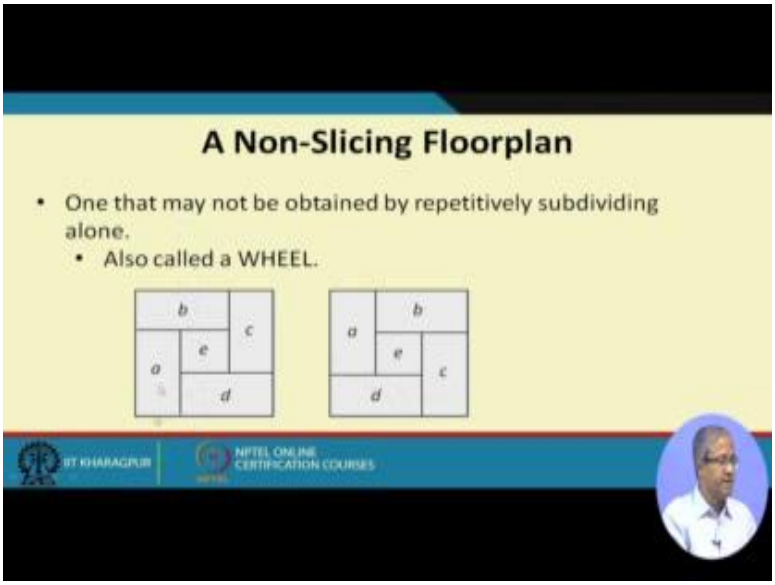
(Refer Slide Time: 24:15)



When you have a node like this, let say V where you have e and f. In the polish notation we express it as first we write the leaf 2 blocks followed by the root V.

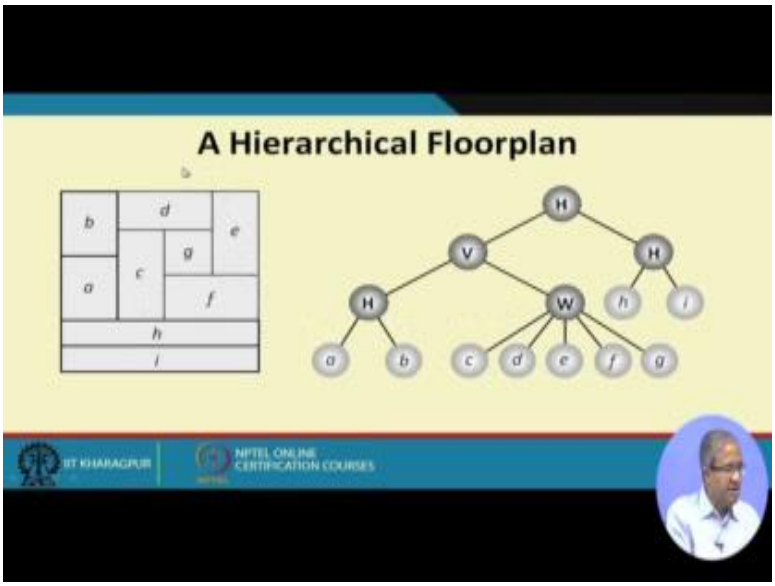
Suppose I have anther this H here this connects here and I have another block a connected here, then I will write a and this whole thing this whole thing represents this whole thing, H after this a whole thing H. So, this is the basic idea behind writing a polish expression. So, when you have any tree like this, you can write down the equivalent polish expression I will show you this well; V I am showing here by a vertical line, and H I am showing by a by a horizontal line a b horizontal line, A B horizontal line here on the other side e f vertical line, you see e f followed by a vertical line, d this thing followed by horizontal line, d this thing followed by horizontal line, c this whole thing followed by horizontal line, c this whole thing followed by horizontal line; this whole thing, this whole thing vertical line. So, this we shall again explain later, but this is how we can express a layout in the polish notation.

(Refer Slide Time: 26:00)



Now, these are the example of a floor plan you can see; here also we have some blocks which are very nicely placed, but you cannot slice them. As you can see you cannot have a continuous vertical or a horizontal slice that can partition. This floor plan this is something which is called a non sliceable floor plan and a topology like these 5 blocks which are oriented in this fashion this is called a wheel.

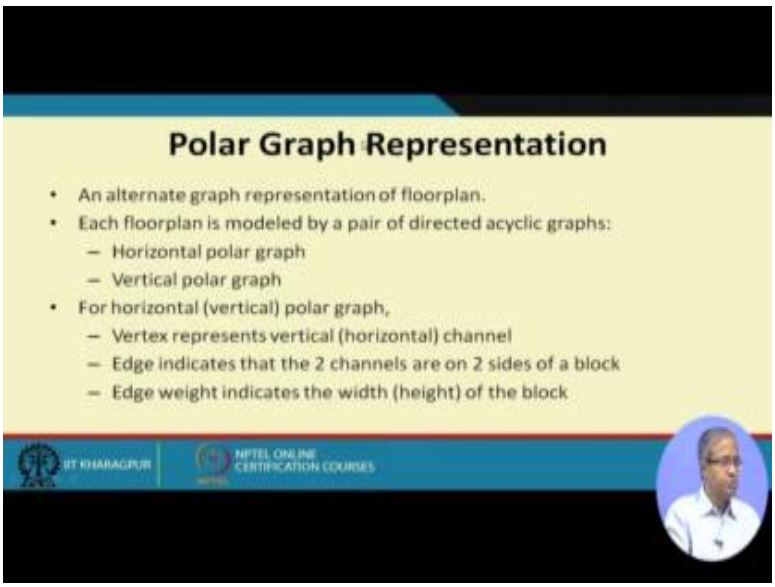
(Refer Slide Time: 26:33)



So, a wheel in a floor plan is represented by a vertex with 5 child nodes, like you say this is represents of wheel in this power plan, d c f e g c d e f g. So, c d e f g represent of a wheel, but once you defined it as a wheel, but rest can be sliced you can have a slice here you can have a slice here and you can slice this and this.

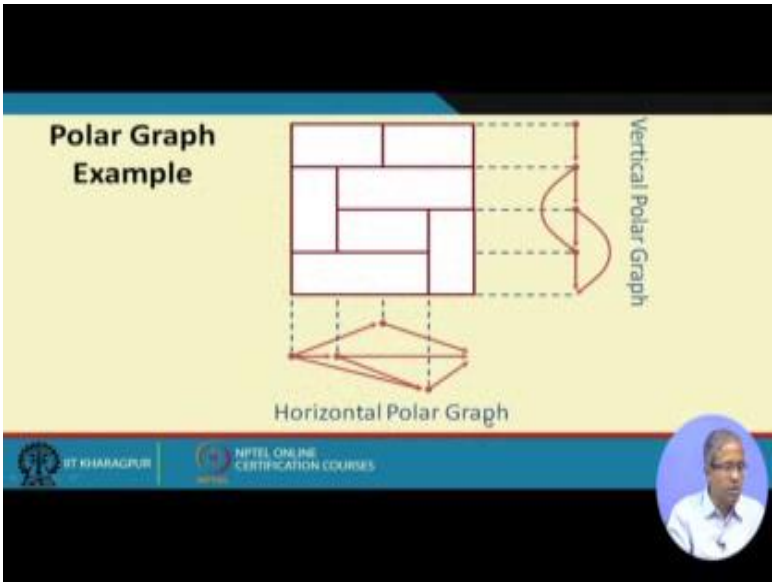
So, you can have a b you can have a slice, vertical this and this you can have a vertical slice, then you can have a horizontal slice horizontal slice. So, you can have a, this is a composite tree which consists of horizontal and vertical slices as well as this wheel.

(Refer Slide Time: 27:24)



And lastly I shall look at anther representation of a floor plan which is also a graph representation. So, here the floor plan is represented by not one, but 2 graphs directed acyclic graphs means that the edges have some arrows. So, one of the graph is called horizontal polar graph other is called vertical polar graph. So, for the horizontal polar graph for example, vertex a vertex will represent a vertical channel, and edge will indicate that the 2 channels are on 2 opposite sides of a block, an edge weight will indicate the width of the block. So, let take an example to explain this.

(Refer Slide Time: 28:17)



Let us take this example. So, this is a layout or a floor plan, this is the horizontal polar graph. In the horizontal direction you see these are the vertex, what does the vertices indicates? The vertices will indicate in each place where you have a vertical line, here I have a vertical line here, next I have a vertical line here in this block here I have a vertical line here in this block, here I have a vertical line here, and at the end I have a vertical line here.

Similarly, in the other direction I can have a vertical polar graph, where I look at the horizontal lines. I have one horizontal line here, 1 here, 1 here, 1 here and the one this whole thing; and this weights like the weights like see this and this there are connected by an edge this edge, and this edge are connected by a block, and the height of this block that will be the weight of this edge, this node and this node this is the block in between. So, the height of this block will be the weight of this edge, this and this; the height of the

this block will be the weight of this, similarly in this direction this and this the whole line is there.

So, this will be the weight the whole width. So, these 2 graphs you can check that if you are given these 2 graphs, from these 2 graphs you can reconstruct this floor plan. So, this is also one way to construct the polar graph.

Now next we shall be seeing some of the floor planning algorithms, how floor planning can actually be carried out. So, just we end this lecture 8 now. So, now, next lecture we shall be looking at the algorithms for floor planning.

Thank you.