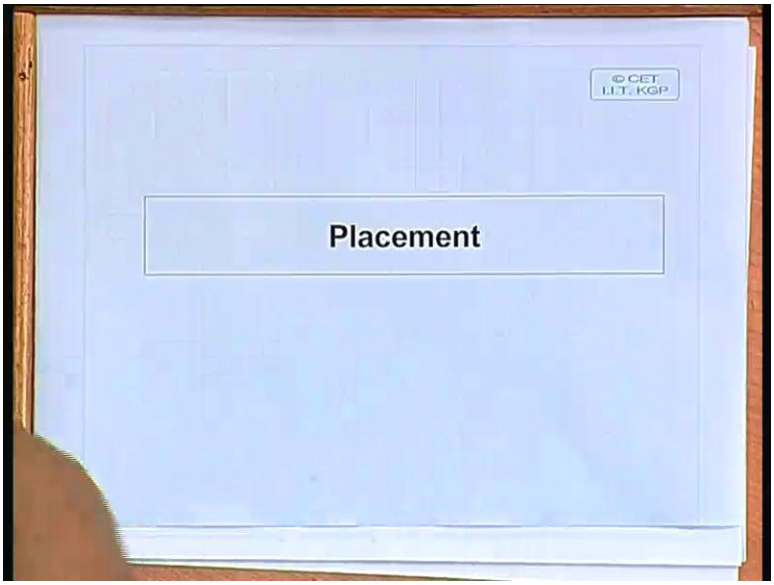
Electronic Design Automation
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture No #20
Backend Design Part-VI

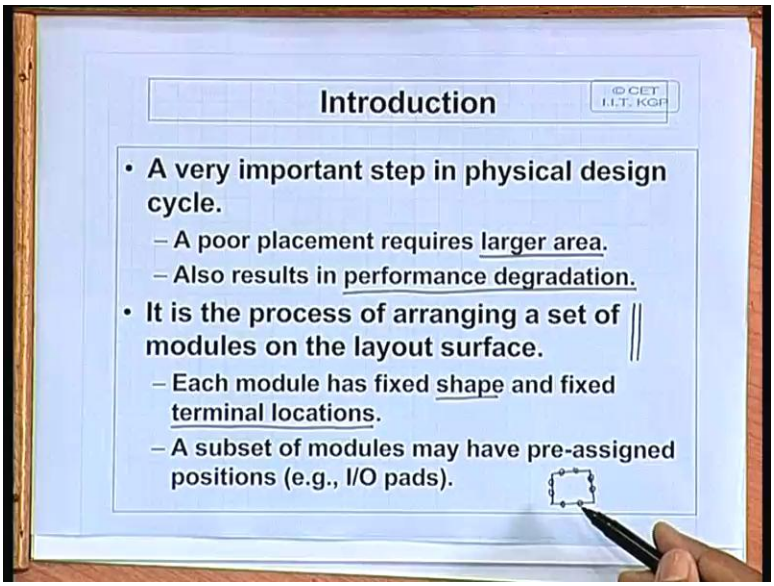
Good morning in this lecture continuing with our discussion on physical design automation techniques. We would be looking at the placement problem.

(Refer Slide Time: 01:18)



Now I had mentioned in my earlier lecture, that the process of floor planning and placement are somewhat similar in concept and in many cases they are done together. So when you are doing floor planning, you are merely trying to find out a relative ordering of the blocks. But while in placement in addition to the ordering you also try to find out the actual amount of space that you must leave in between for the routing or interconnection of a well, the nets you mean say. So the interconnecting nets some space has to be left between the modules so as to interconnect them. So our topic of discussion today is placement.

(Refer Slide Time: 02:13)



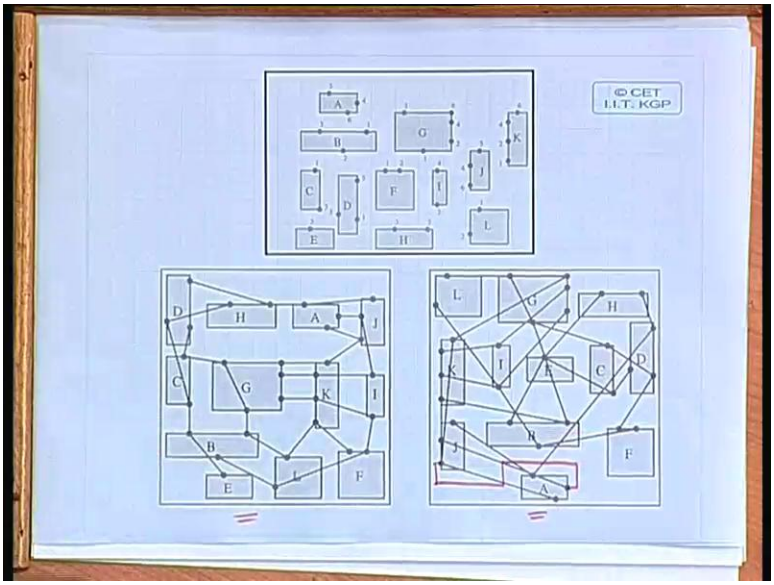
Now placement clearly is a very important step in the design cycle of a VLSI chip. The reason is very clear if you have a poor placement then typically the length of the nets would be pretty long and moreover they will be congestion in the routing regions ok. So it will require larger area in general this larger area will primarily arise out of increased area required for interconnection. And since some of the nets may become longer they may be critical nets also so in general the performance will degrade.

Now essentially placement means you have a set of modules you want to you want to arrange them on a layout surface in this sense this is very similar to floor planning. But I had mentioned that in case of a floor planning you are trying to find out a tentative arrangement but here it is more like an exact arrangement or finding out the exact location of the modules. Now at the level of placement unlike floor planning we assume that all the modules have already being finalized in terms of the shapes and in terms of the terminal locations. Well in floor planning some of the modules may also be flexible.

But when we are dealing at the level of placement there is fixed and moreover some of the modules may already have some pre assigned positions in the chip for example the

input output pads. Well if your chip is of a rectangular shape then the position of this I/O pads may already be fixed because I/O pads also involve some circuitry. So the circuitry can be treated as modules and the position of those modules in the chip may be fixed. So some of the modules may be fixed in terms of location and some of the other modules you will have to place them suitably ok. So I am just showing 1 example in order to illustrate what is a difference between a good placement and a bad placement.

(Refer Slide Time: 04:30)

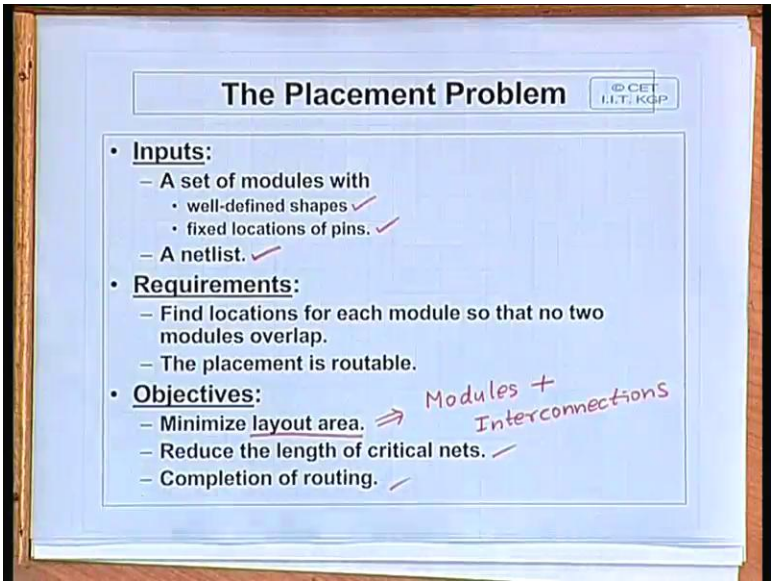


Just have a look at this diagram the top diagram shows some of the blocks which are called A, B, C, D, E, F, G, H, etc., with some points showing the net connection numbers like 1 1 and 1 they have to be connected and so on. Now you see the two extreme alternatives. This of course denotes a good placement for you see that the interconnection nets are very regular and local in nature. But while in placement like this you see that you require long interconnecting lines ok. So the lines will become long in general there will be some criss-crossing of the lines.

So 1 thing you note this is typically the output of a placement algorithm these 2 diagrams I am showing here. Now here you see that I am still not finding out or determining the

exact path through which this nets will have to be routed for instance just you think of a line which goes from here to here for example. But in practice the routing re line may be like this. But at the level of placement where approximating them as a straight line connecting these 2 points. Well we will see that they are our ways of estimating the length of the interconnecting line. But this will give you a fair idea regarding how difficult the routing will be in the next step ok.

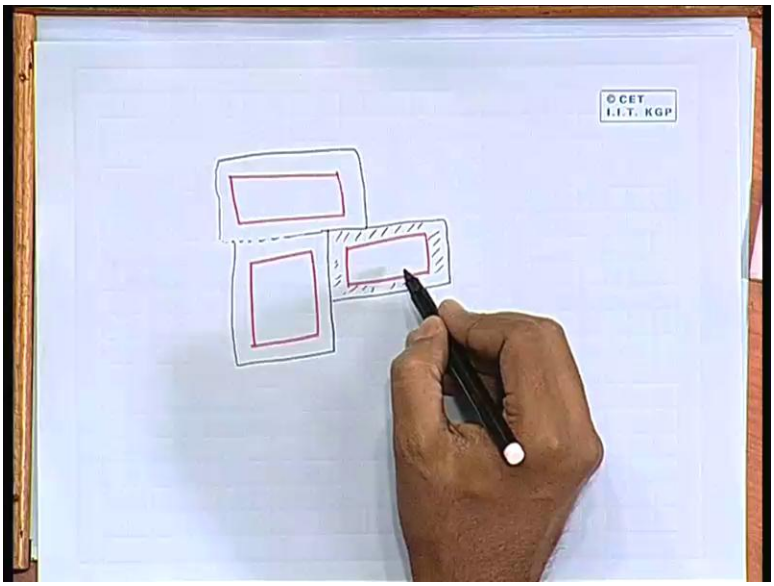
(Refer Slide Time: 06:14)



So I had mentioned the inputs to the placement problem will be a set of modules with of course well defined shapes they are fixed and the locations of the pins are also fixed. And in addition a net list which will tell you how this modules have to be interconnected. Requirement is obvious find the location of each module so that no 2 modules overlap because overlapping modules are not acceptable for obvious reasons we will have to find out disjoint locations for them on the floor of the silicon. And secondly you will have to keep aside sufficient space between the modules to take care of routing in the next step so the placement has to be routable.

So in terms of the objective of course you will have to minimize the layout area. Now layout area will consist of 2 things 1 of course will be the area of the basic modules plus the area occupied by the interconnections. Yes. (()) (07:20) Area of the module is fixed. Yes. (()) (07:25) Minimized means the area of the module is fixed you cannot do anything about it. But 1 thing the way you place it you may find that some of these space in the floor area is the they are lying un utilized. So it is just not the sum total of the module areas was reducing length of critical nets and completion of routing are some idea.

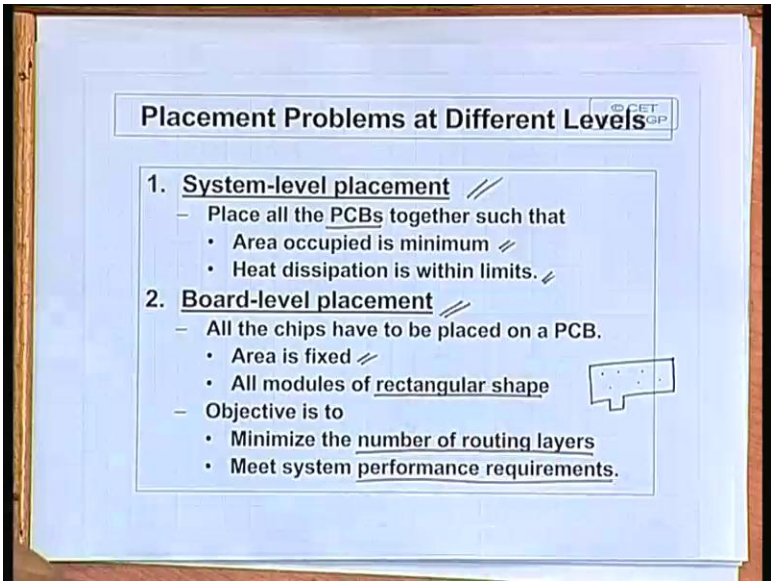
(Refer Slide Time: 07:48)



Well 1 thing I may also mention here that well in terms of the placement problem what we are trying as I had mentioned what we are trying to find out the location of the blocks and also this space between them which will be sufficient for doing the routing. Well some of the tools what they do they before the placement step starts they expand the size of the basic modules for example this module will become this big. This module will become this big this module will become this big.

The reason is that the way they expand the modules is that the additional area you are keeping aside in the boundary of this modules this additional space these are actually the space you are keeping for routing. So once you do this then the placement and the floor planning problem becomes very similar. So there are some tools where floor planning and placement are done together they are some technique like this is also adapted ok fine. Now in fact in practice you can carry out placement at several different levels.

(Refer Slide Time: 09:18)



Suppose that your objective is to design a large system, say a large system it may be something like a computer system or any dedicated application where there will be several printed circuit boards which will be going into a cabinet or a cache and inside the PCBs there will be IC's and inside the IC's of course there will be blocks or modules. So at the highest level you can talk about system level placement. System level placement means I have the system I have the cabinet. Now inside the cabinet I have to place the printed circuit boards together.

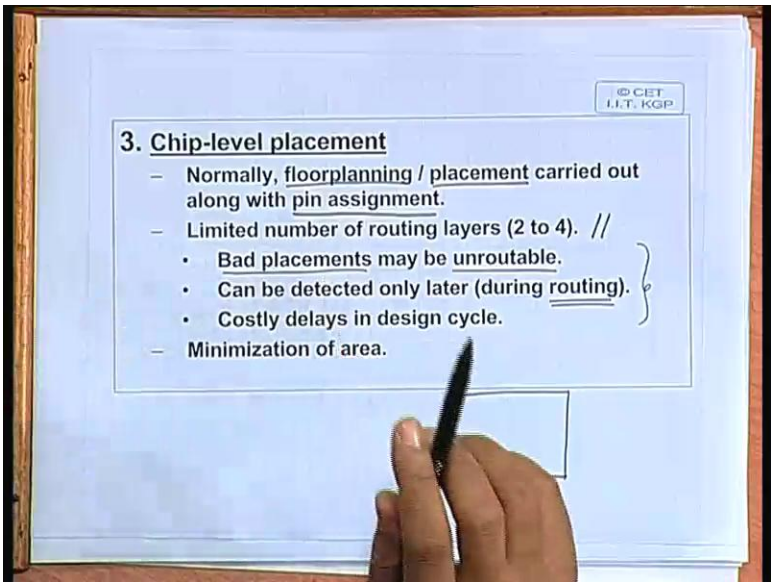
Now the way I would be putting the PCBs together is that of course the total area occupied inside is minimized. And secondly the heat dissipation should be uniformly

distributed so there should not be any concentration of heating in 1 portion inside the cabinet so that there will be some circuit malfunction due to increased temperature. So heat dissipation should also be controlled in a suitable way that how will be spacing out PCBs which PCBs will consume more power they will have to spaced apart or something like that. So at the level of system, we are treating with PCBs as the basic modules and how to put them together it is very high level thing. But once we go inside the PCBs we have what is called board level placement.

So here the PCBs are given and we are given the task of assigning the chips or the components inside that PCB. Now since my overall design of the system is fixed so I have a choice that which chips I need to put in here well I am not really talking about a system like a pc. But the chords are very well defined and the functionality of the chords is known. But rather I am talking about a large dedicated system where what you put on the chord is up to you ok. So here the chips have to be placed on a PCB the requirements in this case are. See once you have completed a system level placement or the top level or means you have done a system level design the size of the boards are fixed. So the total area is fixed and the chips and the other components they are all typically of rectangular shapes so there is no irregular shape involved.

An objective is minimize the number of routing layers because, you know today the PCBs are all multi-layer PCBs and inter connection between the chip pins they are carried out not on 1 or 2 layers. But several layers the layers are all sandwiched together 1 after the other with some through holes which connect 1 layer with the other ok. So if you have multi-layer PCBs then the interconnection area becomes very less. So you will see in todays PCB the chips are so close together ok. So of course more the number of routing layers more will be the cost of the PCB and less will be the reliability. So, 1 objective is to have a design which will minimize the number of routing layers. This will bring down the cost as well as increase the reliability. Of course performance requirement is 1 issue will have to look at the critical paths and other things. This is what you call board level placement.

(Refer Slide Time: 12:49)



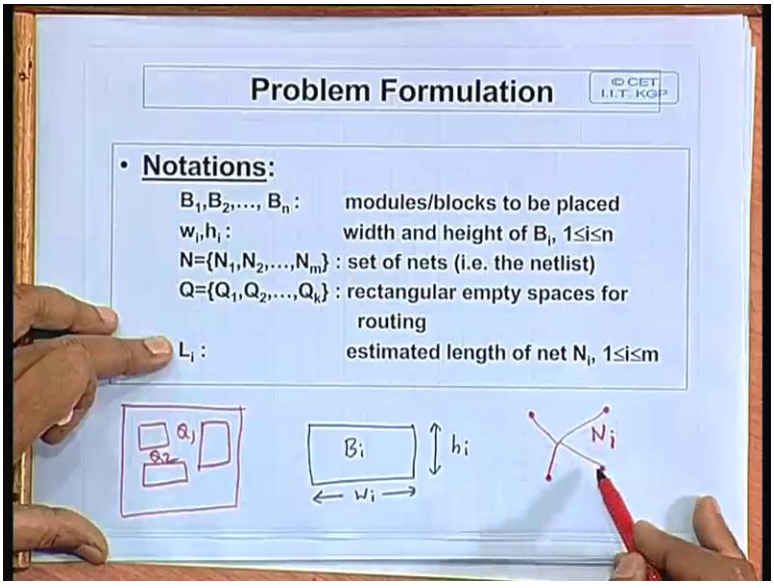
Then finally we come inside the chips this is the chip level placement so now inside the chip level placement we have the complete flexibility. We have the floor of the silicon we can place the blocks and modules inside this. So all the conventional floor planning and now the placement tools and algorithms will be talking about they may be applied to this case. Pin assignment is also an option if you are going for full custom design. Now in in case of chip level also depending on the fabrication technology the number of routing layers are limited. Well in PCBs if you need more number of routing layers you can have you can sandwich more number of layers.

But in an IC depending on the fabrication technology you can have well anything from 2 to 4 layers which are typically on metal which are available to you for interconnection. So this is a given thing you can say some kind of a constraint which is available to you. And since the numbers of layers are limited for routing so if your placement is bad you may have a design where you cannot complete the routing. You say somewhere there is a conflict you cannot find a route for the last few nets. But the problem is that this that the placement is un-routable this can be found out only much later while you are doing the

routing. So if you found out find out during routing that the placement is unroutable you will again have to go back and change the placement and do it again.

So this is a costly delay in the overall design cycle but you will have to live with this. So a good placement is very important if the placement is good you may not have to iterate in this way. Yes. (()) (14:44) Board level well typically the way they are manufactured if you say that I want 8 layers they will provide you 8 layers. But typically the layers are kept between 4 and 6 due to concerns of reliability but the number of layers can be increased if you really want.(()) (15:03) Chips, chips and the other components (()) (15:08) placement of the circuit modules inside a chip ok. Of course here 1 criteria is minimization of area that is important ok fine. So now let us try to formulate the problem mathematically.

(Refer Slide Time: 15:31)



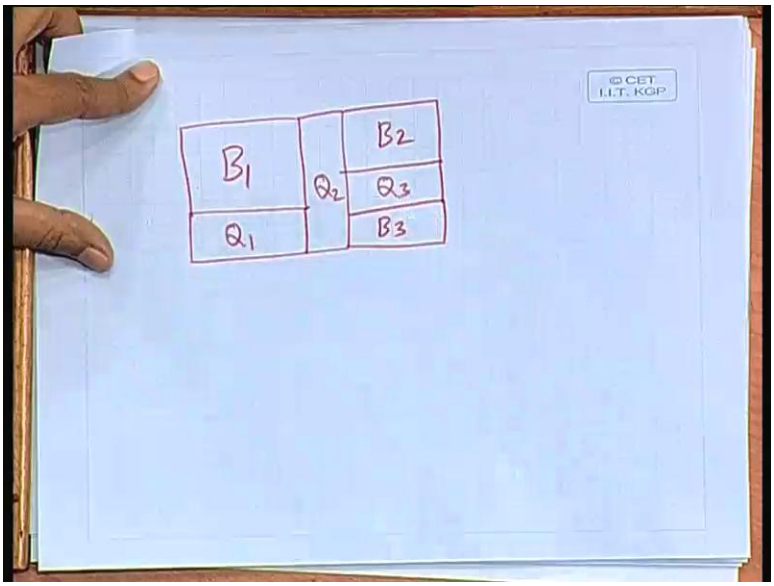
First introduce some notations. Let B 1 through Bn denote the modules or blocks which are to be placed there are n, such blocks. And each block say block number I Bi will be having a width of wi and a height of hi. So wi, hi are the width and height of the blocks which in the placement problem is also given and fixed. And N is the set of net lists. Well

N 1 into Nm I am assuming there are m such nets each net means a set of pins. This will constitute a net for example the problem of connecting these 4 pins will constitute my net number i. There are, m such nets.

Similarly in the placement when you place the blocks you will be having some vacant places in between these vacant places you call Q's, Q 1, Q 2, etc. So this Q's are the rectangular empty spaces which are kept aside for routing. And Li denote the estimated total length of net Ni. Total length of the wire you need to route this Ni this is the estimated length. Because at this level you are not doing the exact routing you are only trying to find out a good estimate this Li denote the length right yes. (()) (17:14)Yes.

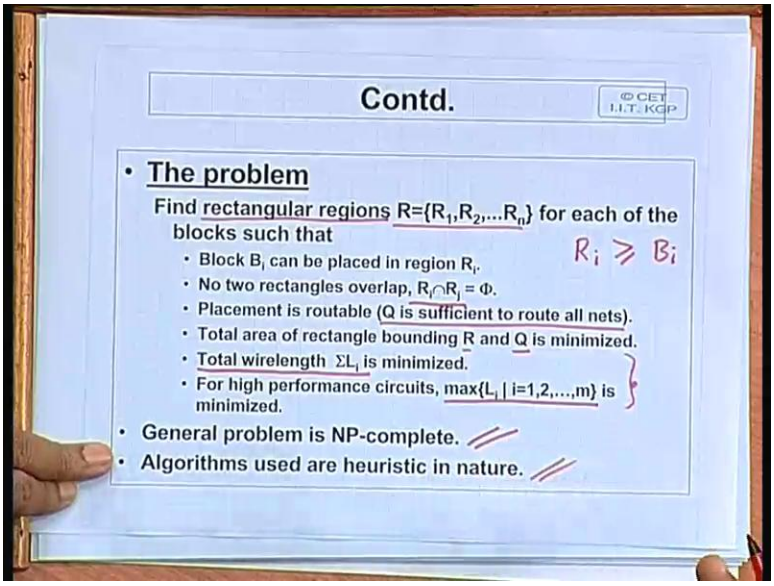
I will come to that. There are number of ways of estimating this length. Well a some kind of a straight line, some kind of an approximation to the reality ok will see that yes. (()) (17:28) It is a connected space but you can partition it into several rectangles because some of the space may be L shaped they may be cross junctions that you can divide into a number of rectangles. (()) (17:44) Now what I am saying yes I am just coming.

(Refer Slide Time: 17:52)



Suppose you have a free space like this this is a free space. There is 1 block out here say this is B 1 there is a block out here B 2 there is another block out here B 3 say. Now what you do this is also free space so you either divide it like this or divide it like this its up to you suppose I divide it like this. So this I call as Q 1, this I call as Q 2, this I call as Q 3. (()) (18:23) See, this we shall come a little later when we go into the problem of global routing because this concept is not really required in placement. This is just a notation where introduced now. But exactly how this this regions are defined that we will see in detail later when we talk about global routing ok. Given the overall placement how do we define this rectangular regions the next sub problem is that for each net, which is the sequence of regions, that need to be traversed ok. So this problem will be coming into later.

(Refer Slide Time: 19:25)



Well the placement problem is well the blocks B 1, B 2 through Bn are given. The placement problem says that you find rectangular regions R 1, R 2 to Rn such that a block can be placed into a region. Now the size of the region may not exactly match with the size of a block. Because you may not be able to fit exactly well what you really require is that Ri should be greater than or equal to the size of Bi. This Ri can be bigger than Bi

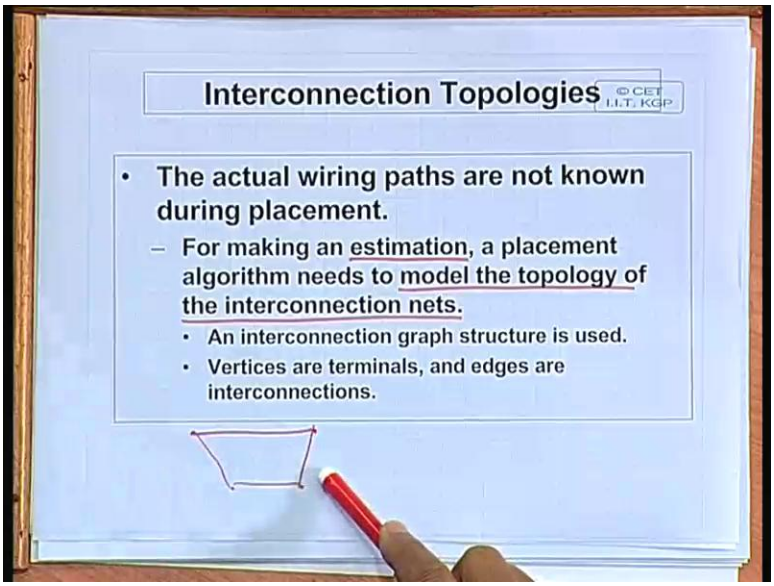
also you can place it there. So block Bi can be placed in region Ri. No 2 such rectangular regions will overlap obviously.

Placement is routable so the rectangular empty spaces which are available they are sufficient for routing. Total bounding area of R and Q R represents the place where putting the modules Q represents the space which are which are kept as for interconnection the total area is minimized. Now in some cases where you are looking at the total area you may want to minimize the total wire length sigma of Li or in cases where you want to minimize the critical paths max of Li may be minimized. The maximum length of a net you may choose to minimize.

Well it depends on your requirement either you chose to select choose to minimize the overall wire length where the total wire length will be minimized or you may choose to minimize the maximum of the wire lengths so that the total length of the wires will remain within bounds. Well the point to note is that the general problem to solve this kind of a situation this has been shown to be NP complete. So we cannot go for an optimal solution and all the algorithms and techniques we would be discussing they are heuristic or they are some kind of randomized in nature ok fine.

But the first thing we discuss before we go into the algorithm is that well when we talk about placement we have seen we are saying that we want to talk about minimizing the wire length. Well either the some of the wire lengths or the maximum of the wire lengths. Now at this level at the level of placement we are not exactly finding out the paths through which the wires will be laid out in terms of the individual nets. So we will have to find out a very good estimate regarding the length of the nets means how much silicon area will be consumed for each individual net.

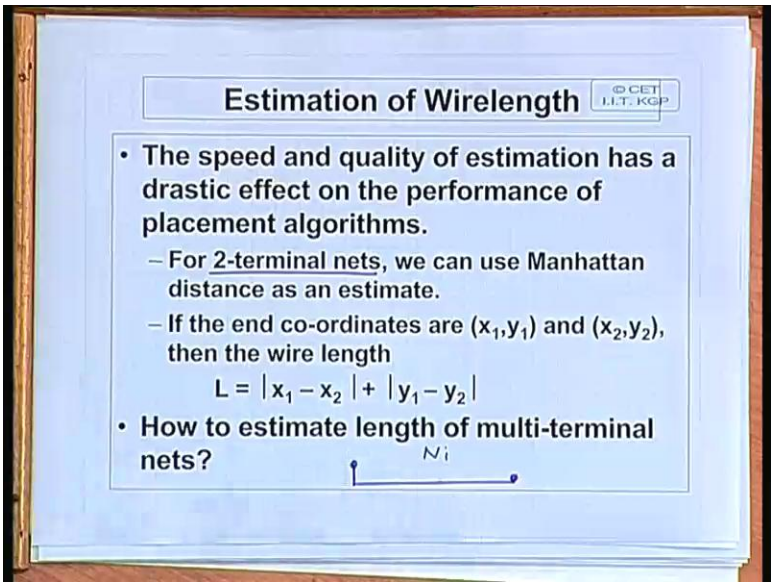
(Refer Slide Time: 22:01)



So these are the so called interconnection topologies and these are very important in case of placement you recall in floor planning we had used some simple estimate like Manhattan distance to compute the cost of interconnection. But here we will be refining the concept we will be having some estimate which is more accurate. So let us see how we do this. Fine so in order to make the estimate so means we do not know the actual wiring path we want to make a good estimate. So in order to make the estimate we model the topology of the interconnection nets this is important. Interconnection nets means again I am talking I am telling you it is just a sequence of pins on the surface of the blocks.

This will constitute a net number say I you will have to connect them. So you will have to model this topology some how in the form of a graph typically and with respect to the graph we will have to find out the cost ok. In that graph vertices are the terminals you want to interconnect and edges are interconnection. Suppose this is 1 example of a graph edges are interconnections vertices are the terminals. Ok. We will be looking at the different graph structures which have been proposed and how the estimate has been done. We will be seeing that. But first, before that let us look at some basic things.

(Refer Slide Time: 23:41)

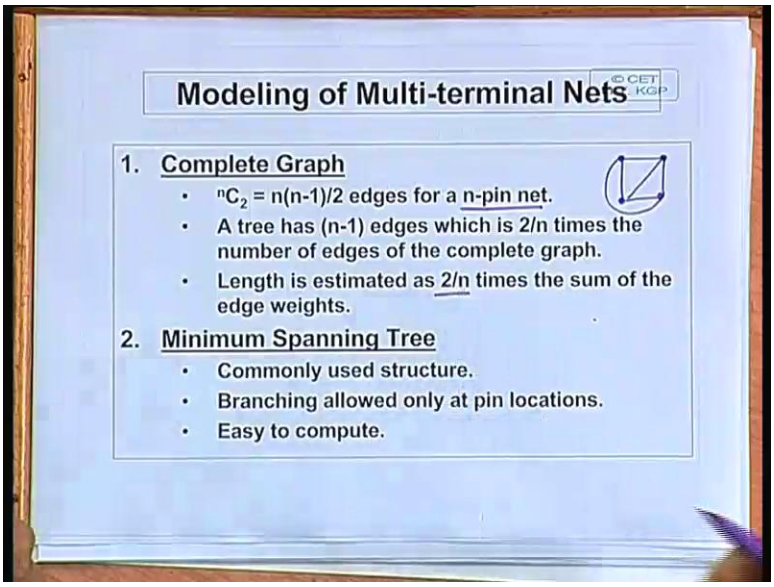


Well first is the estimation of well means of course we are modeling the interconnection topology with the objective of estimating wire length. Now this speed and quality of estimation how fast we can estimate this and how good we can estimate these things will indirectly depend or means will indirectly control the performance of the placement algorithm because if you have a bad estimation then the placement may not be very good. Well if you have good estimation which is which is also running very fast then may be the placement can be done very fast in a good way.

Well for estimating wire length if it is a 2 terminal len net the problem is simple. If there are only 2 terminals to be interconnected this is your net number I say then we simply compute the Manhattan distance. Manhattan distance you recall this is the rectangular distance like this. So if you have the x-y coordinates of these 2 points the Manhattan distance will be the difference of the x coordinates plus the difference of the y coordinates. So the Manhattan distance of 2 points those coordinates are x 1, y 1 and x 2, y 2 can be written as difference between x 1 and x 2 plus difference between y 1 and y two.

Well Manhattan distance is used very very widely in routing because that is normally how we try to interconnect. We do not connect by diagonal lines you connect by horizontal and vertical lines only. But the problem arises if the nets are multi terminal in nature. If instead of 2 I have to connect 4 or 5 points in a net. So, in that case how to estimate the length? Okay. There are in yes (()) (25:36) We can take them in groups of 2 yes we can do that but there are other simpler ways also. Well so now let us see the different alternate ways in which people have talked about estimating length of multi terminal nets. Well first we will explain this then we will just have a look at an example.

(Refer Slide Time: 26:00)



Complete graph is a very simple way to look at the problem. Well here essentially it is the same as what you are talking about you take each pair of them you connect them as a 2 terminal one. Complete graph says that if there is an n pin net that you model the topology where every pair of pin is connected by an edge it is a complete graph. Suppose if it is 4 nodes you connect them like this it is a complete graph so if there are n pins the number of edges will be n choose 2 or n into n minus 1 by 2 right in this graph.

But you recall well in practice what happens in practice if I have n number of pins to connect my interconnection topology will be a tree there is no point in having a ring or a cycle in my interconnection because I have to connect them anyway that will be a tree typically. A tree connecting n nodes will having will be having n minus 1 edge. But in this case when you have a complete graph instead on n minus 1 edges I am having n into n minus 1 by 2 edges which is well such which is you can say n by 2 times more than the that corresponding to a tree.

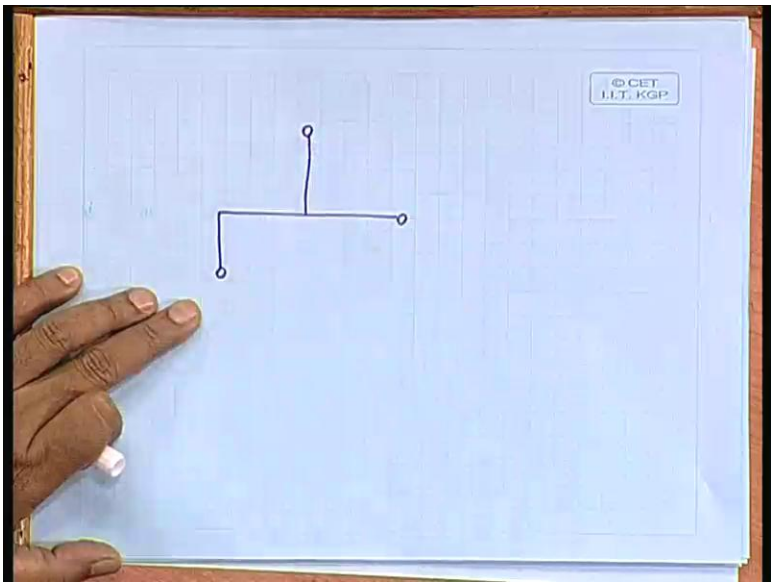
So what we do here in terms of the estimate is that we find out the sum total of the edge length of the complete graph all n into n minus 1 by 2 edges then multiply it by 2 by n. Because in the complete graph we have n into n minus by 2 edges but what we really want is n minus 1 edge. So if I multiply this estimate by 2 by n we will get something close to that that is how we do it we find out the sum total of all the edges multiplied by 2 by n. That will give me the estimate of the average wire length for this net ok fine. But as you can see that here com competition complexity is quite large.

We will have to compute the cost for all n choose 2 edges and then do this things so we had so means it is an expensive way of estimating. The second alternative is to have a minimum spanning tree. So if you have several points you try to connect them in a minimum spanning tree and try to find out the cost. The cost which you get that is the actual cost because spanning tree are the tree is a structure which is close to reality ok so the you do not have to multiply it with any weighting factor finally. Yes (()) (28:55) Yes (()) (28:58)

We will take an example. These distances are all computed as a Manhattan distances not straight line distances. Minimum spanning tree is a commonly used structure here branching is allowed only at pin location just like a tree when you construct a tree well edges can only come out from the vertices we cannot take an edge out from the middle of an edge. This is the characteristic of a spanning tree. So there are many algorithms for finding minimum spanning tree you can just use any 1 of them. So you will get this is

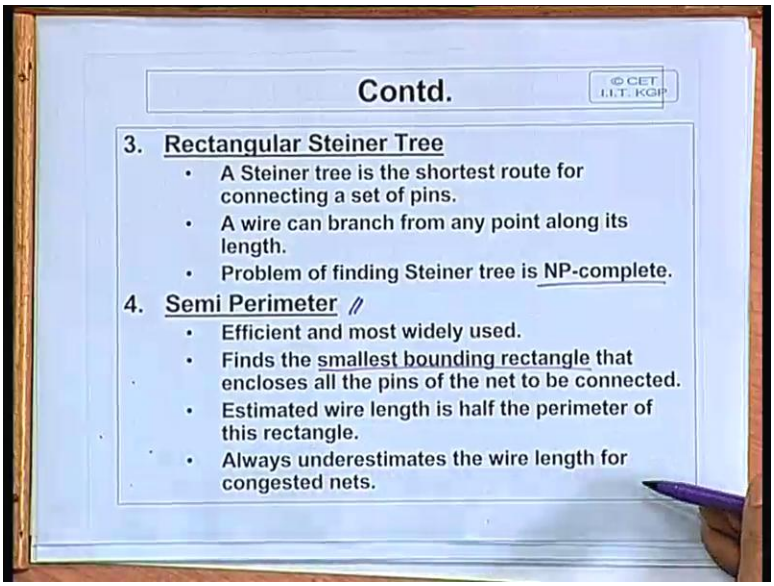
also easy to compute (()) (29:38) No (()) (29:42) No this I am mentioning because you may have a situation like this. For instance where you want to connect.

(Refer Slide Time: 29:50)



Say a pin here, a pin here and a pin here. Now in this case the optimal interconnection path will be like this in terms of a graph this edge is branching out from the middle. This is not vertex that is why I explicitly mentioned that points ok. Now a graph topology like this where an edge can come out from the middle this is called a Steiner tree.

(Refer Slide Time: 30:28)

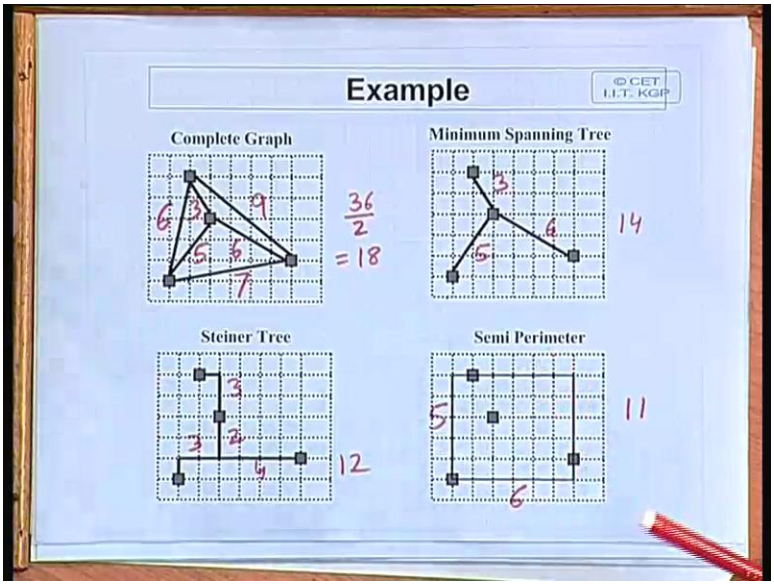


In fact the third alternative I mention is that a Steiner tree rectangular means the edges are all horizontal and vertical rectangular Steiner tree. Now a Steiner tree is the closest to reality well it is a tree alright but Steiner tree is what be really do while interconnecting. Because if it were not a Steiner tree then in this example my interconnection would have been something like this say I connect this like this I again connect this like this. So in this case a wire can branch from any point along its length. But the difficulty here is that the problem of finding the minimal Steiner tree this has been prove to be NP complete.

So although this is extremely desirable finding the best Steiner tree is not computationally feasible right ok. So the measures which most people use today is something called semi parameter semi perimeter. Semi perimeter concept is very simple you have a number of points to interconnect. You imagine the smallest bounding rectangle which encloses all those points. Smallest bounding rectangle and you find the semi perimeter of that rectangle. Find the perimeter divide by 2 that semi perimeter is taken as the estimate of the wire length. Now the advantage of this method is that well this gives a good estimate. Well in case of non congested areas this will give you the exact estimate. In case of congested areas where there are some ob obstacles in between.

May be you cannot draw the lines straight you will have to go through around then it always under estimates the wire length under estimates means it gives an estimate which is less than the actual. This is unlike some other estimate where sometimes it can give an over estimate sometimes it can give an under estimate. But this always estimates in 1 direction so that your decision will never be wrong right. In a set when you comparing 2 solutions if both of them are estimating in the same direction, it will be easier for you to compare otherwise your decision may be come wrong at any stage fine ok. So these are the 4 means ways in which people have talked about modeling the interconnection topologies.

(Refer Slide Time: 33:17)



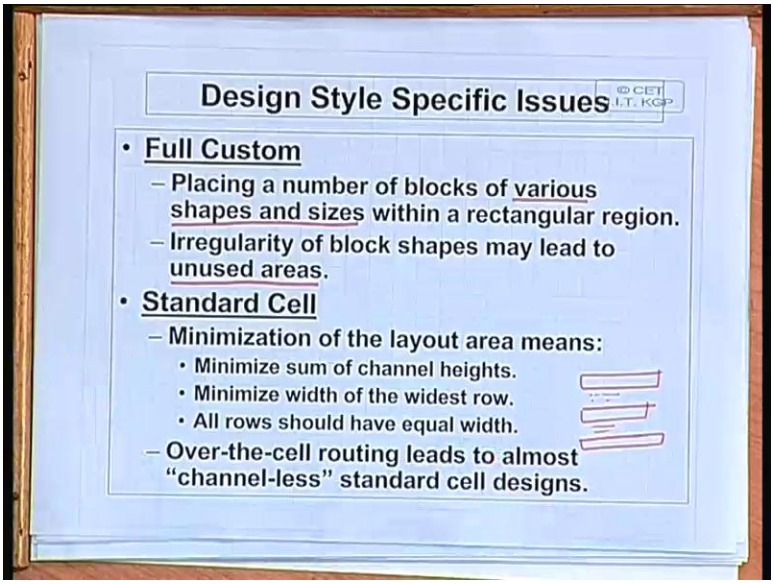
Let us take an example and try to see that how this estimate can be made. So this example shows the problem of connecting these 4 pins ok. First we talk about the complete graph. So there is an edge connecting every pair of nodes. You try to find out the Manhattan distance of each of the edges for example this 1, 2, 3, 4, 5, 6 and 1, 7, so the weight of this edge is 7. This 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, this is 9, 1, 2, 3, 4, 5 and 1, 6, 1, 2, 3, this is

3. 1, 2, 3, 4, 5, this is 5 and 1, 2, 3, 4, 5, 6, this is 6. So if you take the sum total of the edges it becomes 15, 20, 30, and 36, 36.

This you will have to multiply by 2 by n here n is 4, 2 by n is half. So 36 by 2 your estimate of wire length in this case is eighteen ok. In case of minimum spanning tree in this case this will be the minimum spanning tree. Now again estimate the length of the edges 1, 2 and 3 all are Manhattan distances 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 14. This will give the best solution this Steiner tree 1, 2, 3, this segment gives 3 this segment is 2 this segment is 3, 1, 2, 3, this segment is 4. So 7, 10 and to twelve this gives a cost of 12.

In the semi perimeter method you just imagine the bounding rectangle and take this semi perimeter 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 6 and 5, 11. So as you can see for complete graph you need a lot of computation. For minimum spanning tree you need to construct this spanning tree. Steiner tree is same but semi perimeter is very fast. Well you basically need not compute anything because defining this rectangle is very easy computationally. See the minimum x coordinate and the minimum y coordinate maximum x maximum y you can find out from that fine.

(Refer Slide Time: 36:04)



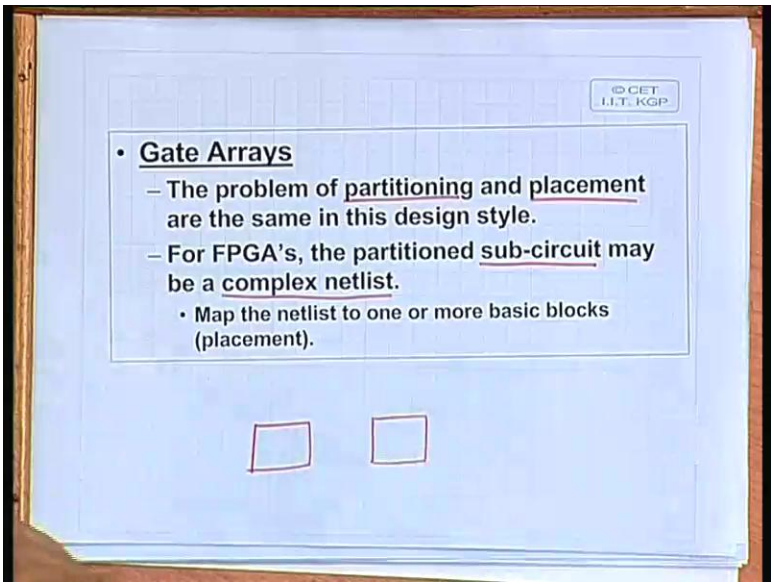
Now in addition to these there are some other issues which also need to be looked at when you are doing a placement because there are a few issues which are very much dependent on the design style. For example in a full custom the blocks may be of various shapes and sizes. And since the block size are irregular it may not all is be possible to pack them in a very uniform and compact way there may be some unused spaces in the corners. So there may be unused areas. So in a full custom design when you talk about minimization of area you are also talking about minimization of unused areas in addition to minimizing interconnection.

Now in a standard cell where all circuit elements are placed along rows like this. There first thing is that you will have to minimize this sum of the channel heights because channels are the regions between the cells which are left aside for interconnection. Now if your placement is good you will be requiring less number of tracks in the channels for interconnection your channel height will be less. So a good placement will be minimizing the sum of the channel heights. Moreover so a placement where some row is small in terms of widths some row is wider that is not a very good way of filling up the rows.

Well in the ideal case all rows should be of equal width. Minimize width of the widest row that is 1 objective. So all rows should have equal width these are desirable objective. Well now in standard cell there is a new concept which has come up normally in the conventional standard cell this space between the rows was kept aside for routing. But in a technique called over the cell routing what happens is that well you have the rows of cells. This space which is above that just above that cell there are some layers of metal above that.

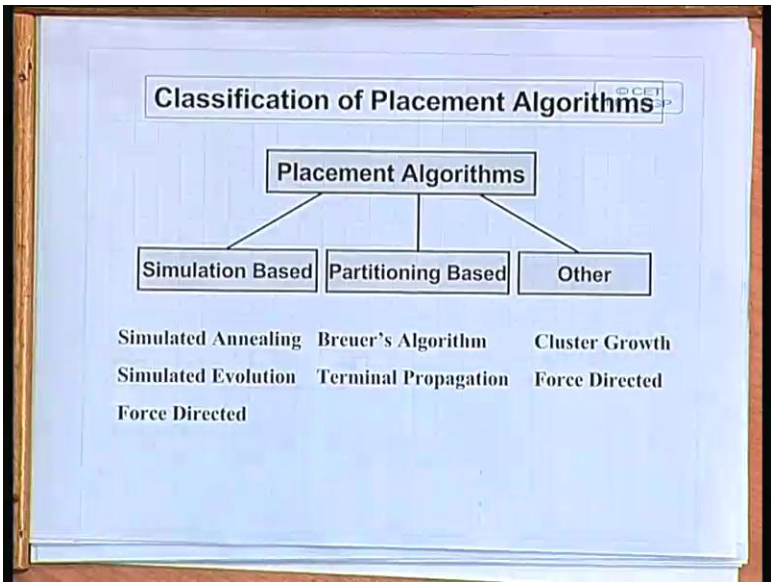
So those layers are used for interconnection rather than the space between 2 rows. This is called over the cell routing. So if you can utilize this space right above the cells for routing purposes then the channel size may become minimal. It is actually you can get some designs which are which are almost channel less the cells are very close together and compact the cell rows ok so this we will see later what is over the cell routing.

(Refer Slide Time: 38:55)



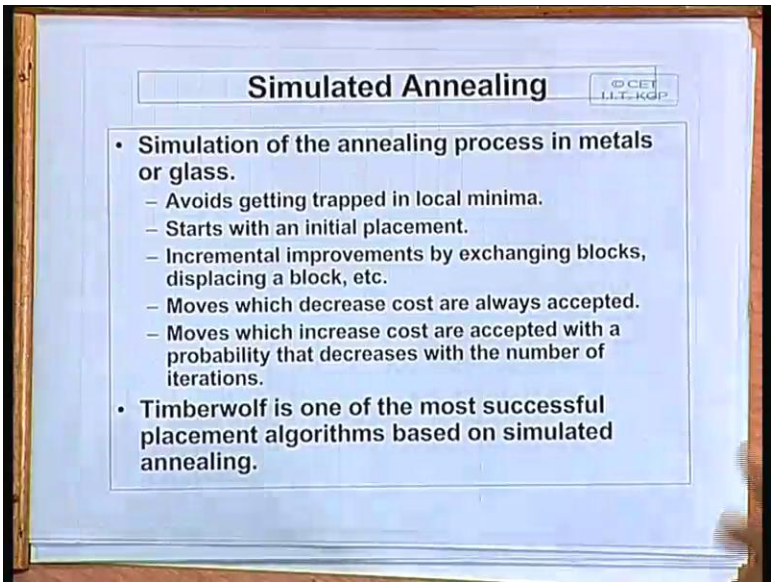
Gate arrays partitioning and placement are almost the same. Because when you are partitioning a circuit your objective is to partition and place them on the surface of the silicon in terms of the gates. So for gate array is partitioning and placement problems are the same there is no difference. But for FPGA'S since the basic blocks on the chip they are more complex than a gate. So the partition sub circuit it can be a complex net list that complex net list can map into 1 or more basic blocks. So the problem for greater than FPGA'S slightly different slightly simpler ok. So now let us come to the actual algorithms which people use.

(Refer Slide Time: 39:54)



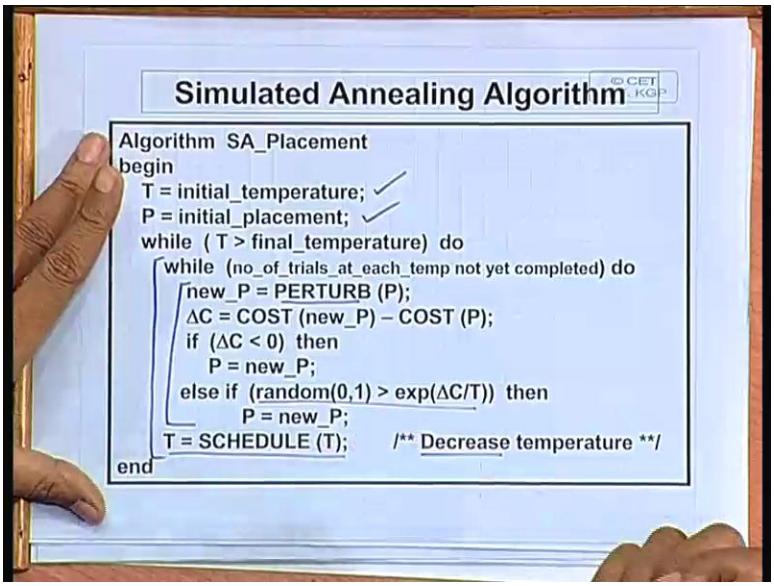
Well broadly speaking the placement algorithms can be categorized as simulation based. Well again simulated annealing and simulated evaluation are the better candidates here. There is a third method called force directed there is a method which is based on partition partitioning and placements are going on together. There is a breuers algorithm and terminal propagation algorithm and some other techniques cluster growth. This force directed can also be treated from a slightly different angle we will be looking at this. So we start by talking about these simulation based algorithms simulated annealing in fact. Well in the placement problem the simulated annealing approach to solving the problem has been most widely accepted because it gives very excellent result.

(Refer Slide Time: 40:50)



In fact very large designs have also been solved in simulated annealing for placement. Well what is simulated annealing we have already seen so I need not spend much time on expressing explaining what simulated annealing is. If you recall you start with a solution you define some moves through which you can modify the solution there is a way of estimating the cost function. You make a move evaluate the cost and see that if it is going to a better solution or a worse solution if it is better you always accept if it is worse you accept it with a certain probability. Now in the initial stages of the algorithm that probability is high, now as you go ahead that probability goes low and low. Now the algorithm this simulated annealing applied to placement this is called timber wolf this is the name of the algorithm which is given ok.

(Refer Slide Time: 42:00)



So the first let us look at broad structure of the algorithm. Well in this algorithm I am using the nomenclature which was used in the timber wolf documentation. They call this temperature tem temper in terms of reality it is just a variable with values this is just a value but they call it temperature capital T is the initial temperature and P is an initial placement this can be done randomly or using some heuristic you can choose this. There is another temperature called the final temperature so until you reach the final temperature you repeat the inner loop this is the inner loop.

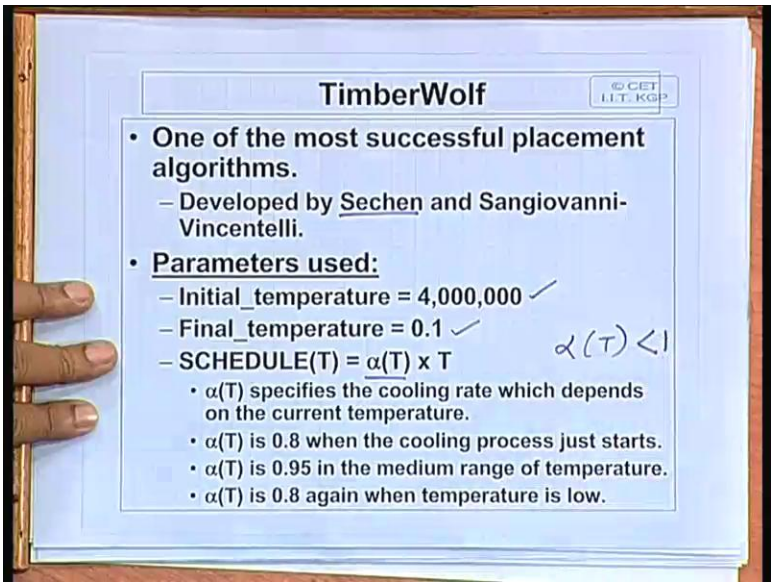
Now in the inner loop you iterate a certain number of times every time while number of trials at each temperature not yet completed this is defined as a fixed number that at each temperature you iterate the inner loop certain n number of times. So again you call a function perturb to modify the solution P to get a new solution new P. You determine the incremental cost increase delta C. If the cost is negative then this is a better solution you take this as a new solution. Else you again based on that random number if based on that probability you accept it or else you do not accept it. And at the end of which well this is the inner loop right. So this inner loop is executed certain number of times.

So after that you modify the temperature using a function called schedule. You apply this function on the temperature you get the new temperature this is actually you are decreasing the temperature. This way you repeat this till the final temperature is reached. Now in terms of this algorithm let us look at the individual function this perturb this cost this schedule. Because the goodness or the badness of this method will depend on these functions or this measures of cost for example. Yes (()) (44:32) See your question is that the temperature cannot decrease until we find out a better solution.

See the way simulated annealing algorithm works is that say at any particular temperature means the outer loop you are making certain number of n trial moves the inner loop is moving n times. Now in this trial moves it is always possible that none of these trial moves are accepted ok. So in that case in that particular temperature you really do not get any better solution you again change the temperature and repeat. But in general what will happen some of these moves will be accepted some will be not accepted. Now the accepted does not always mean that you are getting a better solution.

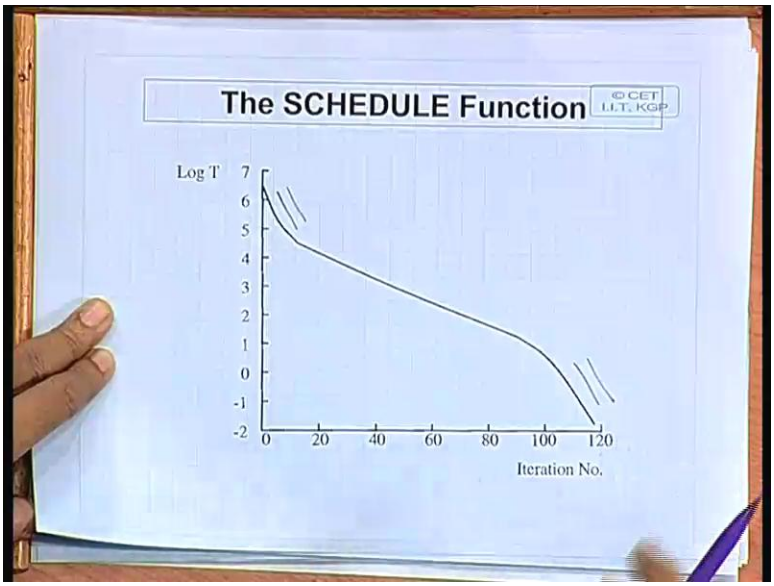
May be you are also getting a worse solution that you are accepting. But you will always be memorizing the current best solution you have seen. Because that will be the final output of your program. Because what you get finally as P that may not be the best solution you have seen so far may be in some earlier iteration you had seen a better solution. Since this is a randomized algorithm and is not deterministic you cannot say that finally you will always get a better solution. So it must also keep track of the best solution you have seen so far ok.

(Refer Slide Time: 46:11)



So this timber wolf, some details about that while of course this was developed by sechen and sangio sangiovanni vencentelli this was as part of a PhD thesis of sechen. Ok here the initial temperature was defined as a number 4 million. The final temperature was point 1 and this schedule function using which the temperature was brought down this was not defined as a linear function this was some function alpha T multiplied by the temperature. Well alpha T is a number alpha T is a number which is less than 1 but the why it was selected is like this. Alpha T was taken to be point 8 when the cooling process just starts. This alpha T was point 95 in the medium range. Again towards the end it was brought back to point 8. So the way you are doing the cooling there are also you are making some adjustment in terms of a plot it looks something like this. Right. We have this.

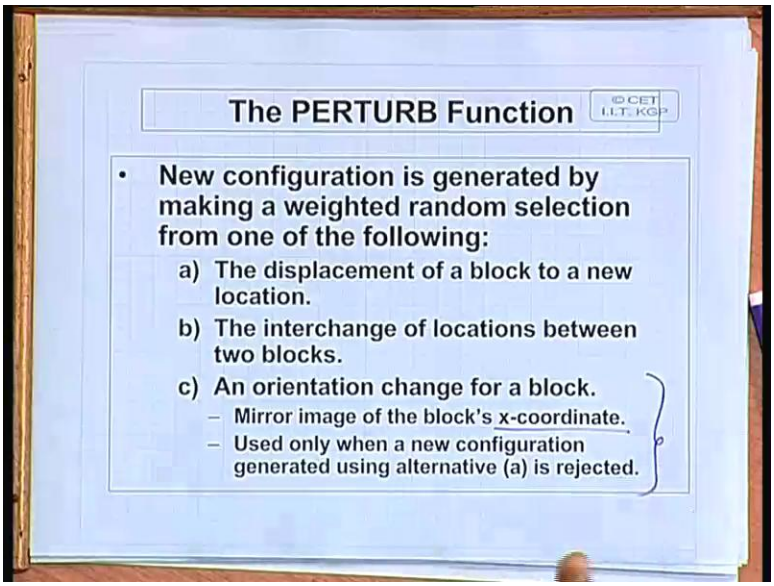
(Refer Slide Time: 47:30)



This is the iteration number this is the temperature on a logarithmic scale. So this this initial slope and the final slope are the same in the middle the slope becomes different. So through extensive experimentation just found that if you follow this slope the results obtained are better. So there is no technical justification why this is chosen this, this cooling curve has been selected through extensive experimentation. And other found that this cooling curve gives you (()) (48:02) Yes, yes.

You are exploring more values in the medium range. Because initially may be you are choosing here we have starting with a very junk solution you are making some changes going towards a better 1 after that you are exploring on that better one. And again towards the end we have already got some good solution you are a little less well interested to make changes there and this is the intuitive justification. And then the perturb solution the way the solutions are modified.

(Refer Slide Time: 48:43)

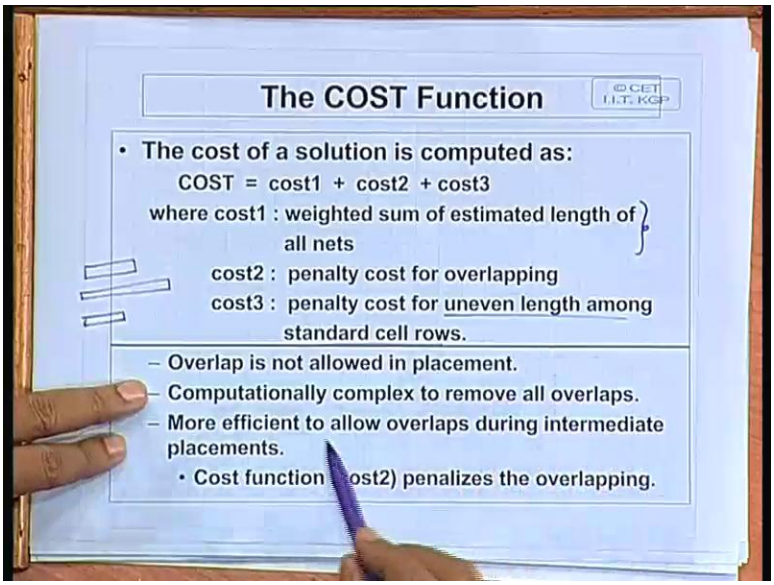


Well there are 3 different moves which are defined. These moves are very simple to understand. The first is that you move a block to a new location simple move. The interchange of location between 2 blocks 2 blocks you exchange the positions. And third you change the orientation ok orientation in general may mean rotation or mirror imaging. But in this case they used only mirror imaging with respect to the x coordinate just mirror image of the block with respect to x. But the way this move 3 was applied is that if this a was rejected then only we apply c suppose you select a move a and you find out that by the probability criteria this was getting rejected then you just try and apply c. So c was not an independent choice you will first apply a or b if a gets rejected then you may apply c right. Ok now 1 thing to note out here is that.

Well when you are talking about displacement of a block to a new location or interchange of 2 blocks see a feasible placement or a valid placement will be 1 but obviously no 2 blocks will overlap. But in this kind of a algorithm when you are saying that I am moving a block to a new block position then I will have to check whether that movement will create an overlap with all other blocks. So this simple checking itself will be of order and

complexity because there are n minus 1 other blocks. So what this algorithm does is that it does not check this at every step. So in the intermediate solutions there may be some overlap between 2 or more blocks. So how this algorithm is tackling that situation well this is allowing block overlap. But it will try to find a way out by defining the cost function in a suitable way how. If there is an overlap and negative or means you can say some kind of penalty is imposed that is a part of the cost function. So let us see how the cost function was defined here.

(Refer Slide Time: 51:30)



So the cost of the solution was computed as the sum of 3 factors cost 1 cost 2 and cost three. Cost 1 is the weighted sum of the estimated length of all nets we have have seen already we can do this. So you are finding out the estimated sum of the net lengths. Now the thing to notice that you need not compute this afresh for the new solution because you are making a move just the portion which is affected by the move you can recompute that part. So this can be an incremental computation not necessarily you are computing the entire thing for the new solution from scratch. Cost 2 is the penalty cost for overlapping so if there is an overlap you incur a cost. (()) (52:20) Yes, yes. Here for computing the cost you have to check that yes.

But while making the moves you do not check that. (()) (52:33) In a sense you can say you are cal. Yeah. In a sense you can say that a for every move you have to the calculate the cost you are check checking this anyway. But if you are say means the idea that if you are excluding block overlap at the intermediate stages then possibly you are excluding some of the better solutions because may be some blocks will overlap. But since the cost will increasing some of the other block will move out to some other place may be in some later generation. That is the possibility we are keeping open. That is not exactly what I had mentioned earlier that you are not computing cost every at. Yes, you are computing the cost.

But we are also opening the possibility that there can be overlap. But the other block may be moving out possibly leading to a better solution because we are also including that in the cost penalty. And cost 3 is that see this method was initially or primarily applied for standard cell placement because standard cell placement is the most commonly and widely used placement to use today. Cost 3 is a penalty cost which is used only for standard cell kind of a placement for un even length among standard cell rows like 1 of the row may be in narrow 1 row may be longer 1 row may be this size. So there is a cost depending on the unevenness of the rows so they defined an expression for this also.

These are the 3 components it is the sum total of these three. So here as I mentioned overlap is not allowed in placement but it is computationally complex to remove all overlaps. So we allow overlaps to occur but to incorporate this in the cost function so that most likely with respect to the cost will moving towards a better solution which will remove that overlap ok. So today in this lecture we had talked about 1 very popular algorithm or method which is used in placement. Now in our means in our next lecture we will be looking at several other techniques which are used for placement simulated evaluation force directed scheduling and some other partitioning based and cluster growth kind of technique. You will see that some of those methods are good as stand alone ones some of them can be used to generate the initial solution for the simulated annealing kind of algorithms ok. So this will be seeing in the next class.