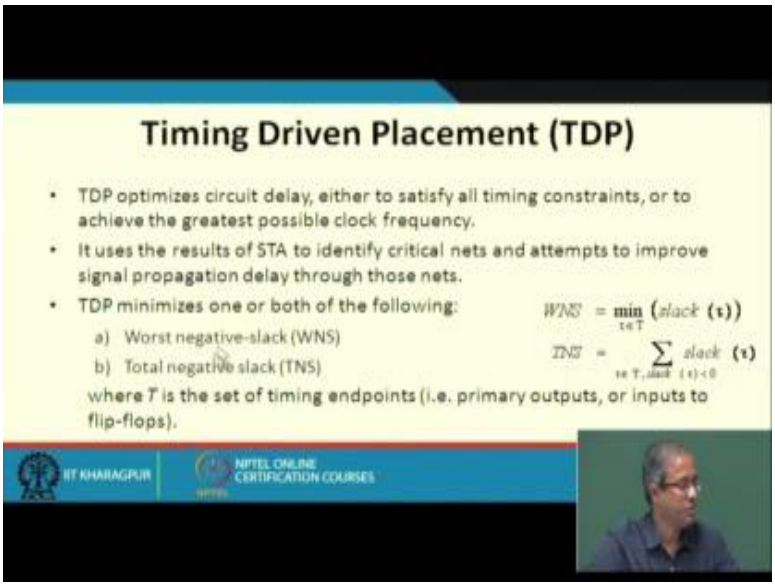
VLSI Physical Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 37
Timing Driven Placement

In this lecture, we shall be talking about Timing Driven Placement. In the earlier lectures we talked about various timing issues where you do static analysis of a circuit to identify false paths and so on. So that some timing related constraints can be satisfied and if there are violations some measures can be taken. Later on we shall see exactly what kind of measures can be taken that we shall see later, when we discuss some physical design or some physical alterations to the layout such that some of this time budgets can be incorporated or the modifications done.

Now, here in this lecture we shall be looking at how we can place the blocks subject to some timing constraints which are given to us. So this is little beyond the timing algorithms or the placement algorithms I mean that we had discussed earlier because at that time we did not consider the timing issues. We just looked at the netlist requirement the sizes of the blocks position of the pins and so on and our objective was to minimize the layout size and minimize the interconnection length or cost some weighted sum of the two, but now a third parameter also comes in some timing constraints we need to place the blocks in such a way certain timing constraints are met. Let see.

(Refer Slide Time: 01:59)



So, timing driven placement TDP, we refer to in short. They try to optimize circuit delay. Not necessarily area and the wire length only. This is done either to satisfy all timing constraints. So we had seen that some of the timing constraints maybe corresponding to false path. So ultimately the bottom line is we want to achieve the greatest possible clock frequency. That is our that is our final objective.

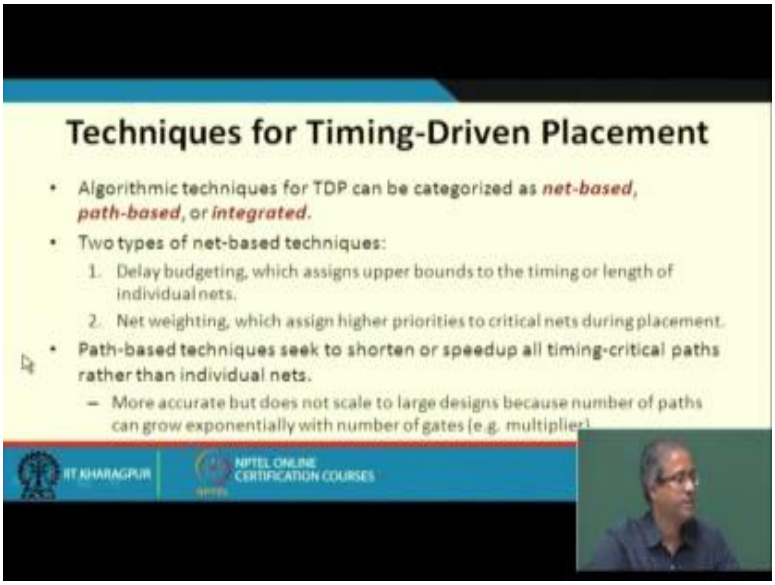
So, for this we need to use static timing analyzer as a tool. So whatever STA gives you that we will be used to identify critical nets, but again as I mentioned earlier, you may be required to do some iteration. Because when you do static analysis you are starting with some estimates of the interconnections. So from where do you get the estimates unless you do a placement? So this can go hand in hand actually. This will be an iterative process.

So, the timing driven placement it aims to minimize means one or both of these parameters. Of course, it will try to eliminate the timing violation in terms of negative slacks, but there are 2 kinds of 2 ways in which you can measure that. You can either measure in terms of worst negative slack; you find the slack value that has the greatest negative value in terms of magnitude. So it is the worst value that is there or you can sum up all the negative slacks and see the total negative value there. So either one you can use.

So, we refer this T to be the set of timing endpoints. So this worst negative slack can be defined as, the slack for every net timing endpoints tau, where tau belongs to T, where T is the set of timing endpoints. So what is the timing endpoints? The places from where, the places into which the outputs are going because whenever your combination circuit, so you either correspond to an output or that output can go to an input of a flip flop. So this refer these are typically referred to as a timing endpoints. That is up to which we need to consider what has to be done within a clock cycle.

So, either the primary outputs or the input to the flip flops, which is the worst negative slack. And the total negative slack is that you see that so whatever slack tau is less than 0 just add them up. These are the 2 parameters you can optimize any one of them or both.

(Refer Slide Time: 05:10)



Now, timing driven placement techniques can be broadly categorized into 3 types. One is based on nets; that means, you look at the nets individually do not look at the total paths. For every gate output you say that if there is a fanout that is a net, for another gate there is a net, for another gate there is another net. So you look at one net at a time, and try to place that particular gate or module in such a way that that particular net value gets optimized in terms of the timing requirement.

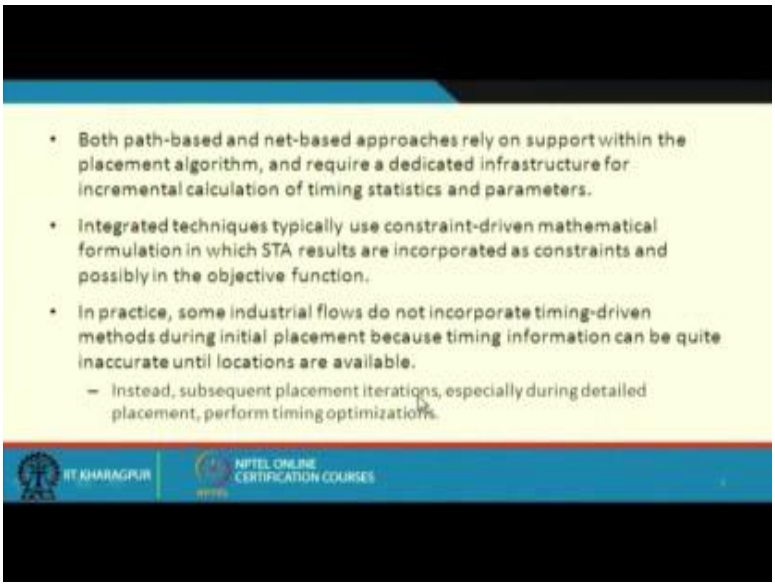
But again, what you need to do here is that you need to have some kind of timing budget for every net. So unless you have this you cannot do this right because someone has to

tell you that this net has to have a maximum delay of say 5 minutes. Then only you can check whether it is within 5 or not right.

The second type is path based where you look at entire path. And you try to optimize the timing of the path. And third one may be integrated you do not look at individually net or path, but integrate the bound whole problem under a single framework. So net based technique as I had said, you have to do some kind of delay budgeting. You have to just assign some delay values to the nets. So you can assign some kind of upper bounds to the timing or lengths of this individual nets. And you can also do some method for net weighting. So you identify which nets are critical. So the critical nets can be given higher priority or higher weights. So you can do a static timing analysis based on an initial placement, then you try to identify the criticality of the nets which of the nets are critical give them higher weights. So whenever you place them give priority to the higher weights of the nets right.

In contrast the path based technique as I had said they seek to handle the all timing critical path; that means, from input to the outputs. They do not look at individual nets. These are more accurate because you are more interested in the input output requirements, not requirement of the individual nets, but again the problem we have mentioned earlier is that the number of paths can grow exponentially for many circuits. So handling individual paths may not be a feasible thing.

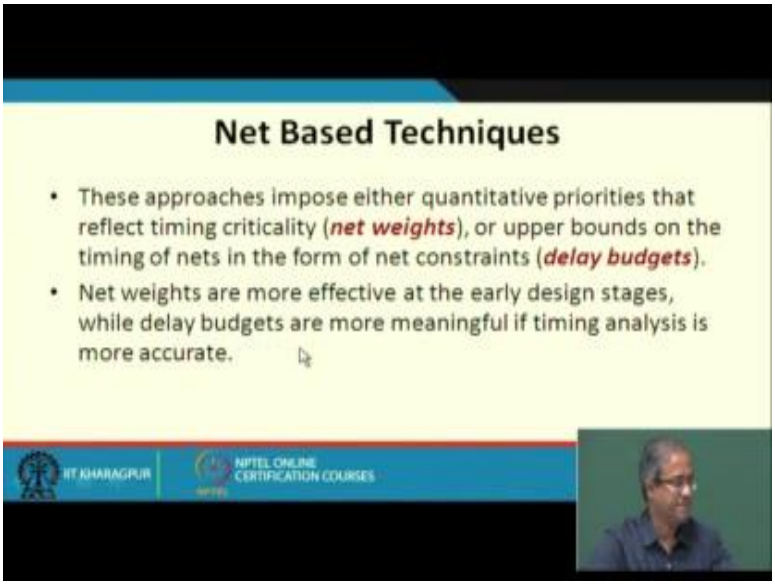
(Refer Slide Time: 07:50)



So, both the approaches either you use the net based or path based they has to go hand in hand with the placement algorithm. So there has to be some kind of a incremental implementation, so that you can incrementally modify the placement and immediately get a feedback with respect to timing and other related things that how you are proceeding, so timing and this placement, these 2 has to go hand in hand. So unlike the earlier case this has to be done first and then this, but again here we see that there is a dilemma, so unless you do placement you cannot get the wire lengths, and unless you do timing you cannot get a good placement. So there has to be iterative step and incremental they has to go hand in hand together right.

So, in contrast the integrated approaches that we have also considered the third approach. They use some kind of a mathematical formulation. They formulate some constraints. In these constraints the static timing analysis can be inbuilt some results of STA. And there is some objective function defined which we are trying to optimize. Now in practice some industrial design flow which handle large high performance designs, they do not incorporate timing issues during initial placement, because as I had said that unless you do placement you do not get accurate information regarding the delays. Therefore, they do not incorporate timing driven methods in the initial stages, but during subsequent iterations they do that so this makes sense, all right, fine.

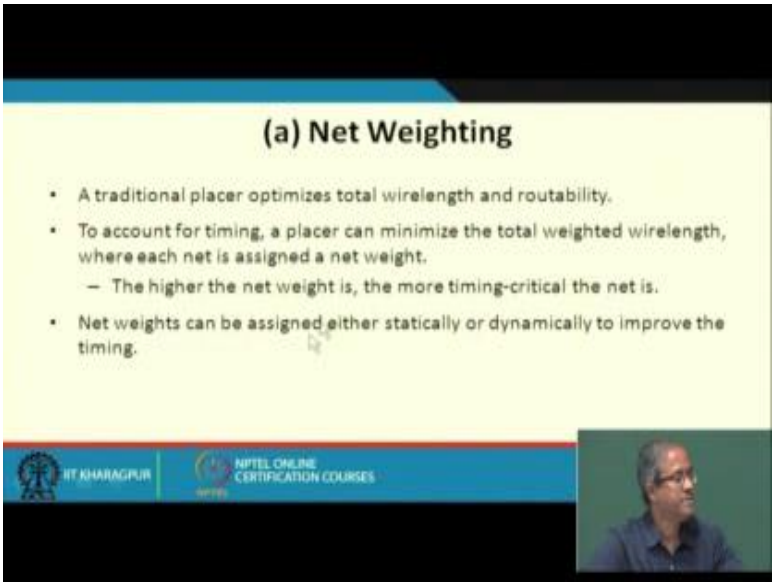
(Refer Slide Time: 09:45)



Now, coming to the net based techniques, these approaches impose quantitative priorities with respect to net weights, which can indicate how critical a particular net is. Or you can give some kind of delay budgets. If you can specify some upper bound of the delay; that means, you can either say that this net is very critical this net is not so critical. This is one way of saying or the other way is that you can give some upper bound that will for this net my maximum delay budget is this. You will have to fit within that budget there are 2 ways of specifying.

Now, the initial stage is net weights can be more effective because delay budgets are very difficult to estimate during that stage. So initially you can just say some weights some nets are more critical than the others, but during subsequent iterations you can go for delay budgets. So once the timing analysis results become more accurate, so only then you can get the delay budget estimates as well.

(Refer Slide Time: 11:05)

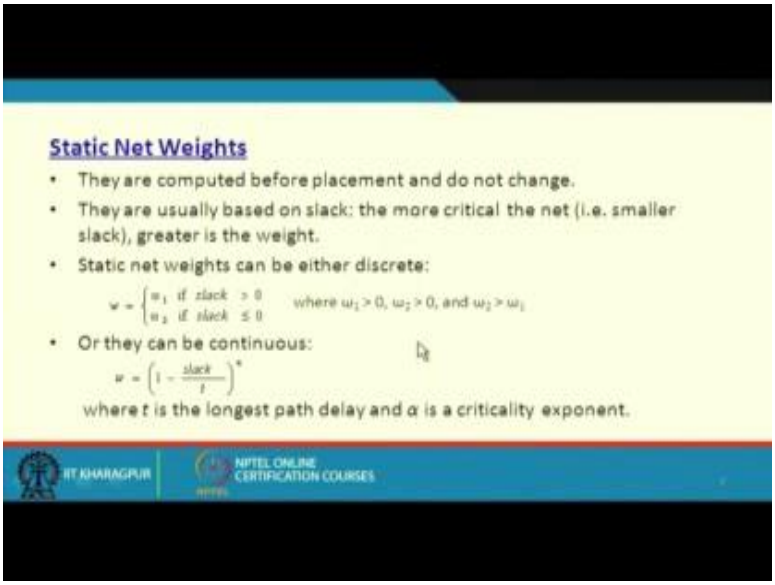


Let us look at some of the existing methods which have been proposed for this net weighting how they make the estimates. So a traditional placement tool that we have seen earlier this optimizes just the total wire length and routability, whether there is sufficient space for routing.

But to account for timing and addition, the tool can also minimize the total weighted wire length where this weight will be a measure of the criticality. So every wire length can be assigned a weight so higher weight means more critical.

So, if you can assign some weights and if you can translate this problem of minimizing wire length to a weighted wire length minimization problem, then this net priority can be implicitly taken into account. Just to modify the cost function for the placement right. Now the way you assign the net weights can be done either statically or dynamically. So we shall see some of the approaches that have been proposed.

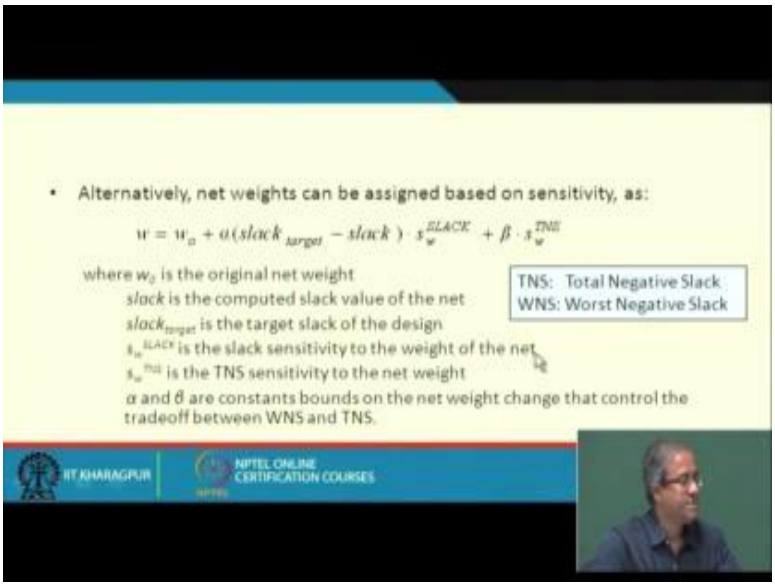
(Refer Slide Time: 12:14)



Static net weights - they are called static because they are computed apriori, before the placement process starts. And do not change during the process. And these are basically estimated or calculated based on the slack values that are calculated through static timing analysis. So the idea is that the more critical the net; that means, smaller slack the weight should be greater. So there are 2 ways that have been proposed. One says that you use discrete values of net weights, 2 values omega 1 and omega 2. Just give omega 1 if the corresponding slack is positive give omega 2 if the slack value is negative. So omega 1 is usually less than omega 2 omega 2 has a higher weight. So if the slack value is negative you give those nets the higher priority.

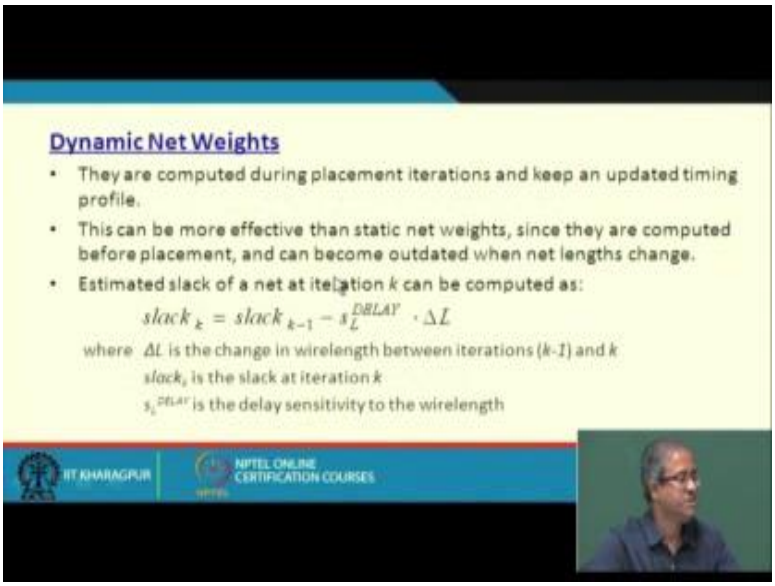
Or there is another method which propose, that you can have a continuous value. You can calculate w as 1 minus slack divide by T to the power alpha, where T is the longest path delay in the circuit and alpha is some critically exponent critical to exponent which you can adjust. You can modify the value of alpha and by that you can also change that how much the variation in slack will impact the value of w.

(Refer Slide Time: 13:50)



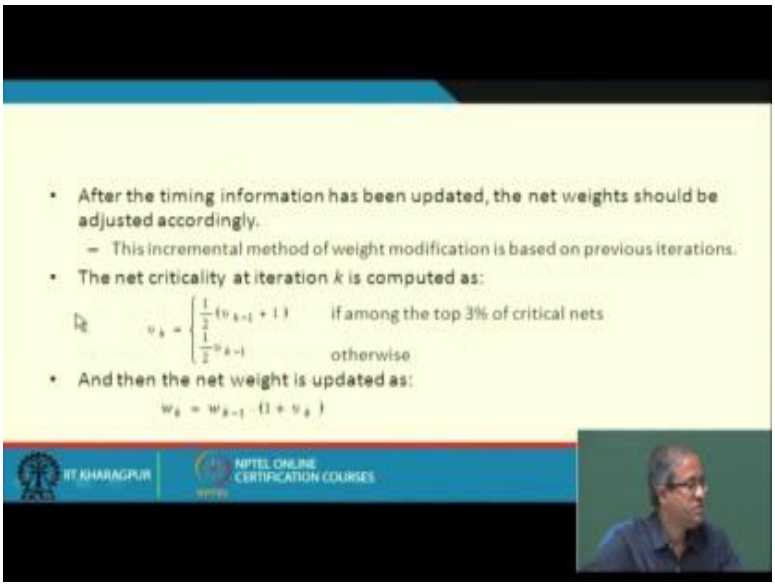
So, there was some other method also, where net weights were based on sensitivity. So this was the expression which was used I am not going to much detail here. So they used some computed value of the slack. And slack target is the target slack how much slack you want; alpha is a parameter, beta is also parameter. And sw slack says, this means, that what is the total negative slack sensitivity to the net weight; that means, if the net weight changes how much impact will it have on the TNS value. Similarly, you can have TNS slack and TNS slack. This is for the slack value and this is for the TNS value. So how much impact a change in net weight will be having on the corresponding slack value. So these are some methods which have been proposed.

(Refer Slide Time: 14:54)



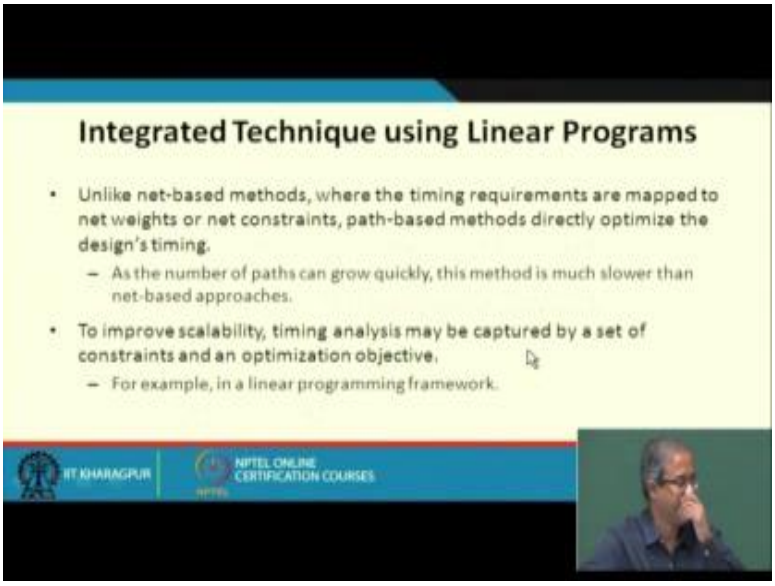
And for the dynamic net weighting, they are modified and changed dynamically during placement iterations. And obviously, this can be more effective because you are changing the weights based on the current scenario. And the old values can become outdated when you make some modifications. So this is an iterative value. So the slack value at an iteration k can be calculated from the previous iteration value by making some changes based on delta l, delta l can be the change in the wire length, that have been taking place between these 2 iterations. And this will be the impact of the change in wire length and the delay, some factor that multiplied by delta l; you subtract this one slack k minus 1 that will be your new slack.

(Refer Slide Time: 15:54)



So, so once you have calculated the slack, then there are ways to calculate wk. So some method have been proposed, that you select or classify some of the top critical nets. So if it is among the top one, you assign a variable vk like this. If otherwise, you assign a lower value. Do not use this plus 1. And once you calculate vk, you can calculate the net weight, as the previous iteration net weight multiplied by 1 plus vk. You just incrementally update it.

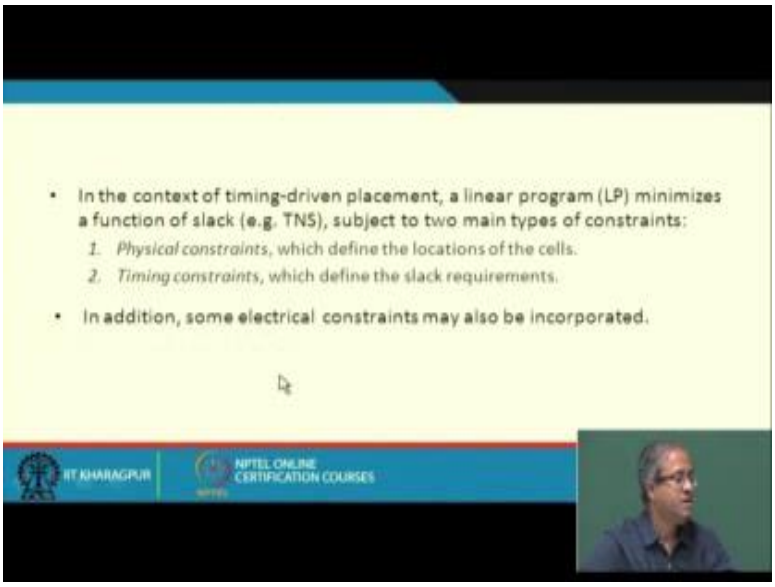
(Refer Slide Time: 16:34)



So, these are some of the methods which have been used, but the integrate approach is interesting here, the entire process of optimization you are expressing mathematically and you are using some kind of a solver like linear programming solver LP solver to get a solution to the problem. That, what will be the placement such that these constraints are met. So let see on that here how this works.

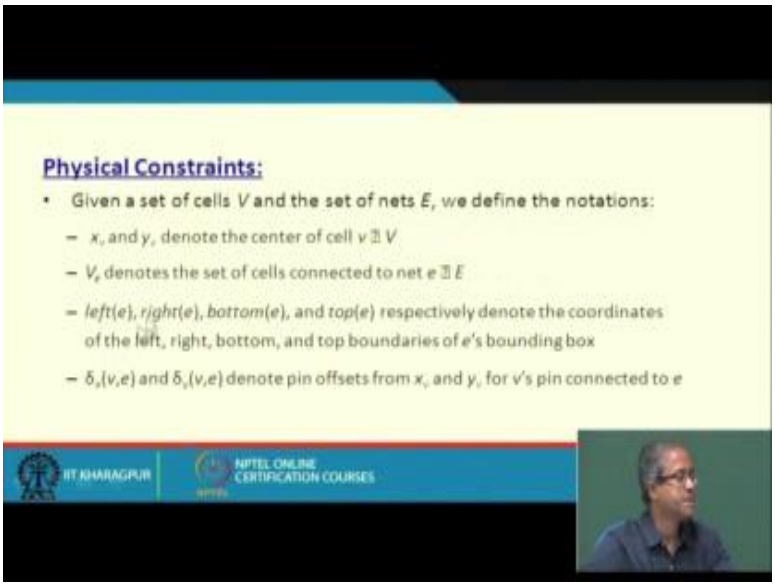
So, in the net based methods, you recall the timing requirements are getting mapped to net weights or constraints. While the path based methods directly look at the timing for the paths, but the path based methods are not scalable. Number of paths can grow very quickly. So for scalability and good performance the timing analysis can be captured by a set of constraints and objective function that can be optimized. For example, the one that we that you present here uses a linear programming formulation.

(Refer Slide Time: 17:45)



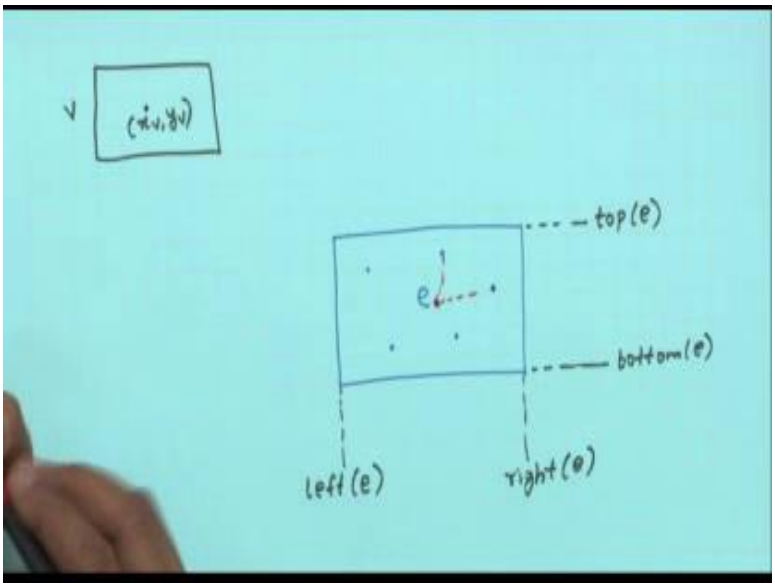
So, how does it work? They actually specify 2 kinds of constraints. One is called the physical constraints those are the placement so where the cells are located and timing constraints that concerns the slack values. So in addition you can also incorporate some additional constraints some electrical constraints if you want, but we only talk about these 2 here physical and timing.

(Refer Slide Time: 18:13)



So, the physical constraint let see what these are. Given a set of cells V in a set of nets E we define xv, yv to be the center of a cell v, there should be, belongs to see this symbol has not come small v belongs to capital V. So what does this mean?

(Refer Slide Time: 18:37)

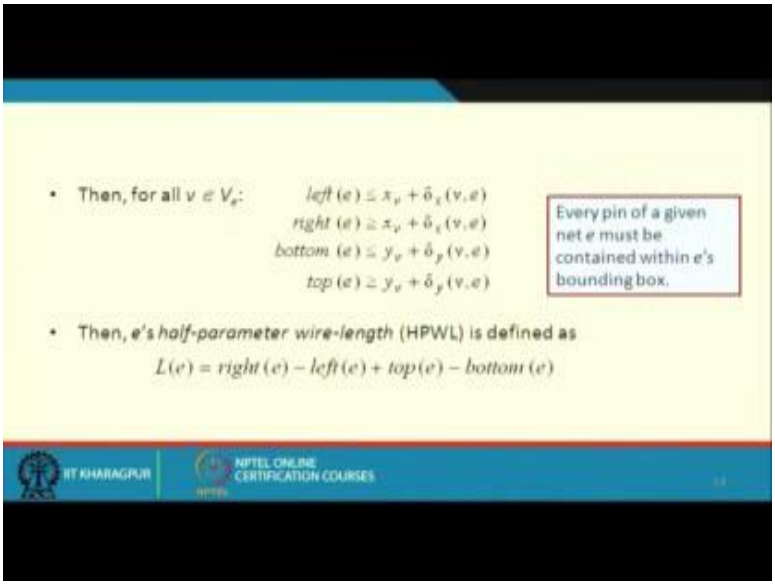


Suppose I have a cell. This block I am placing I call it v. This center point center of this cell, I call it the coordinate xv and yv right. Ve denotes the set of cells which are connected to net E. And left right bottom and top denote the coordinates of the 4 boundaries of es bounding box. So what does this mean? Let us explain.

This e denotes a net right. Suppose this is the bounding box that is represented by e, e denotes a net. Means there are some points to be connected. And I am using a slightly larger bounding box with the assumption that all interconnections will be completed by this bounding box.

Now, we consider some edges left right top bottom. So with respect to this see left means some x coordinate. This is left of say e, this is right of e, left e and right e refers to some x coordinates. And top and bottom refers to some y coordinates. Top of e and bottom of e, they refer to some y coordinates. So with this we will see that how the constraints are and the last one delta x v,e and delta y v,e denote the pin offsets from the center, for vs pin connected to e. So from the center where the pins are located, so you see once we have defined the bounding box we expect that all pins has to be there. So from the center we are measuring delta v delta x as the x and y offsets. So they should not be outside this bonding box that is the constraint, they should all be inside this box fine.

(Refer Slide Time: 21:02)

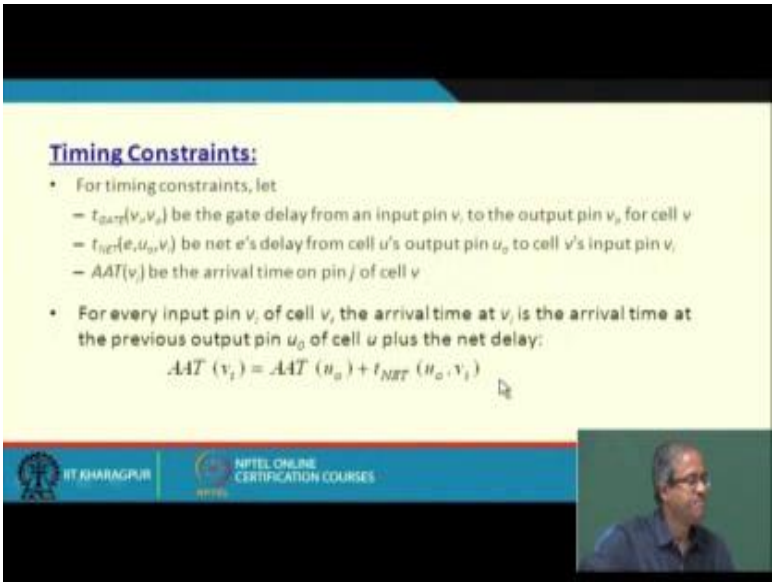


Let us look at the constraints now. So the physical constraints say the first set of constraints they tell you that every pin of the net e must be within the bounding box. These are captured by these 4 constraints. Left e must be less than equal to the x coordinate of the center plus that offset, for that pin e. For some pin for every v belonging to this net the v must be this. Similarly, right one should be greater than equal to this. So it should be within that range. Bottom again should be less than equal to the y

values top should be greater than equal to; that means; this v should be inside that imaginary rectangle.

Secondly the total net length is estimated as the half parameter your recall. This was one of the measures. This is called half parameter wire length. Because it is a rectangle you take the half parameter. So length plus width, so from left right you can say right minus left plus top minus bottom, that will be your you. That will be here half parameter. So your half parameter can be calculated like this - right minus left plus top minus bottom.

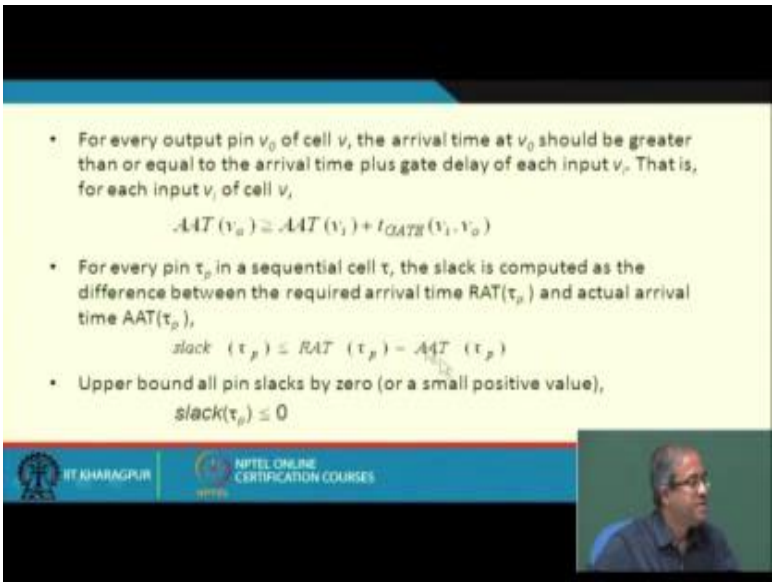
(Refer Slide Time: 22:26)



So, timing constraints you can similarly add like this. So some notations T gate v I, v o is this defined as the gate delay from a input pin v i to the output pin v o of a particular cell. And T net e, u o, v 1 is net is delay, delay of a net from cell u's pin some output pin u 0 it is coming from a cell u, u 0 is the output pin and it is going to the input pin of v, it is called v i. That net delay and AAT you get from static time analysis. This is the actual arrival time on pin j.

So, there are a few timing constraints you can identify. The first one is you are saying that for every input pin of cell v the arrival time at v i will be the arrival time at the previous output pin u o plus the net delay. So AAT of v i should be the arrival time in the previous output plus the delay of this net from u o to v i. It is clear this is one.

(Refer Slide Time: 23:40)

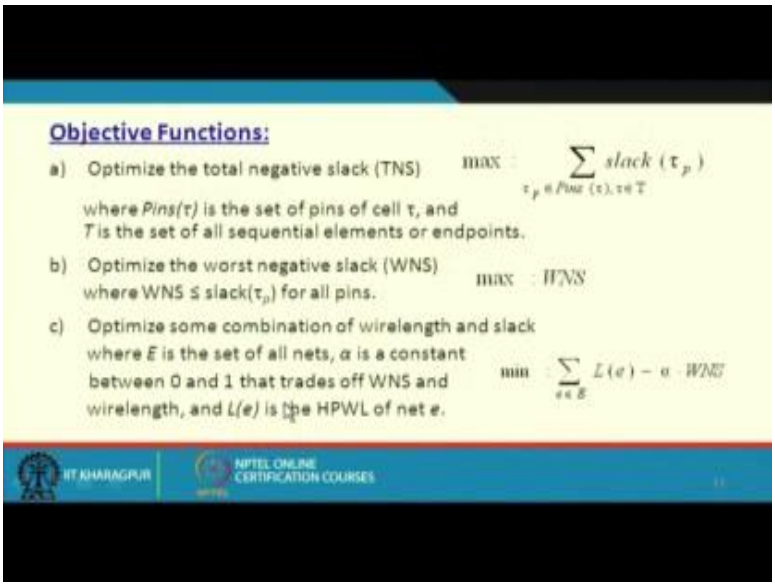


Similarly, second constraint will be for every output pin v o of a cell the arrival time at v o arrival time at v o should be greater than or equal to the arrival time of each of its inputs plus the gate delay. So for a gate the arrival time of the output should be greater than equal to arrival time of the input plus the delay of the gate. The third constraint concerned is slack. It says that for every pin of a sequential cell tau, let us call it the slack is computed as a difference between the RAT minus AAT that is the definition so slack for every timing point we are calculating as difference between RAT and AAT.

So, you see this RAT and AAT calculation the static timing analysis they should be available as some kind of function. You will have to get those values whenever required. So static timing analyzer is available to you we are assuming that.

And of course, the slack values can be upper bounded to 0 because you do not mean want a positive slack value because you seen earlier we use something like 0 slack algorithm. So that we try to bring all slack values close to 0. So either you just upper bound it to 0, or use a small positive value so that the slack values all reach close to 0 within that range.

(Refer Slide Time: 25:16)



So, the objective function can be 3 types, 3 of them are shown here. You can add some more. First one is to optimize the total negative slack. Total negative slack can be expressed like this Pins tau is a set of pins of a cell tau, and T is the set of all sequential elements or endpoints. So you calculate the slack of all system. One thing is not mentioned if it is total negative slack, we will also have to mention another condition that slack tau p should be less than equal to 0, right.

But here since you are assuming in terms of a constraint that slack tau p will less than equal to 0. So it will be either 0 or negative it can never be positive. So under this condition you can assume this. Just summation of the slack tau p this will give you the total negative slack.

Then the second condition maybe you optimize the worst negative slack. Max WNS. What you know the expression you can calculate that, so WNS will be less than equal to slack tau p for all tau p, and you can make some weighted combinations sum of the total length of the net wires and let say the worst negative slack WNS. So you can use this L e as the half per meter wire length for all the nets sum total. And you can use this wire this WNS negative slack. So means alpha can be a parameter between 0 and 1, so using that you can actually control the impact of the area the length of the net and the slack. So these are the typical objective functions which are used here.

So, you see here we looked at a few things. We looked at that how we can do placement taking into account the timing characteristic or the timing parameters in mind. You see earlier when we did placement we ignored timing, we only looked at the location of the blocks and minimization of the wire lengths. Now we have additional parameter that is coming in we are trying to minimize slack, we are trying to minimize delay and so on. So you now require or you now have an environment where the placement tool and the timing analysis tool are working hand in hand. So here we are we just mentioned a couple of methods more advance techniques are also available, but here we have not discussed those.

They really work hand in hand and they are capable of handling large designs as well because you see because the industrial designs are large. We need to have a method whose time complexity is not large of course, we would like to have accuracy, but we would also have to look for scalability. The method should be able to handle lager circuits that a practical in today's scenario and so on and so forth.

So, now that we had talked about timing driven placement, there are other issues that we will be looking at later during our next week's lectures. We shall be looking at for example, timing aware routing, timing driven routing. So how to make the geometry connections for the nets such that some timing estimates or timing budgets are satisfied and finally, whenever we have to satisfy some timing budgets which you get from the timing analysis tool, how to make annotations or modification in your netlist or in your devices such that, those timing analysis or the requirements can be met.

So, with this we come to the end of this lecture.

Thank you.